Depth map generation from geometry and motion

Qianmin Li *, Chenyang Ge, Pengju Ren, Huimin Yao

Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Shaanxi 710049, China

liqianmin123@gmail.com

ABSTRACT

As the demand for 3DTV keep increasing these years, the conversion from exist 2D videos to 3D ones becomes a new area of research. Depth map generation plays a key point in the process. Two most important clues of depth are geometry of the scene and motion vector. This paper presents an algorithm of depth map generation, which intends to get the depth map combines two aspects of information. Compared to the previous work, our method is improved in finding vanishing point, detect motion vectors, and depth map generation.

Keywords: depth map, vanishing point, motion detect, 2d to 3d conversion

1. INTRODUCTION

As people's demand for real time and vivid version, three-dimensional (3D) video recently attract lots of attention in market and research. Although we can take new video using 3D camera, convert 2D videos to 3D ones is an easier way. The reason why we experience 3D is that the scenes seen by people's two eyes have a parallax, which can be analysis by our brain so that we can feel the depth information. When it comes to 2D to 3D conversion, we need to make out another view of the scene besides the one already exist in 2D video. Christoph Fehn[1][2] brings out an algorithm which can generate a new view of the scene by the existing view and the depth of it, called Depth Image Based Rendering (DIBR). This makes the generation of depth map becomes a very important step.

In the existing depth map generation algorithm, to generate depth map, two aspects of information of the sequence are mostly used, the geometry information[3] and the motion vectors[4]. Geometry is connected with depth map by vanishing point, which is considered as the furthest point of the image. When it comes to the motion vector, the object near to the camera moves faster than the object far, so the big motion vector represent the object is closer to camera.

This paper also use the two aspects of information. We did some work in improving the existing algorithm to make it more efficiency and exact, so that hardware realization of the algorithm is possible

2. PROPOSED METHOD

We'll give a brief look of our algorithm, first detect the vanishing point to get the geometry information, then detect the motion vector and combine it with graph cut, at last combine two aspects together to get depth map.

2.1 Vanishing point detect

The detection of vanishing point have two main steps, accumulation step and search step. In accumulation step, get some potential vanishing points. In search step, there're some criterion to pick out the vanishing point.

(a) Accumulation step: We only use the luminance information this step. To get edge of the objects in image, 2D 3×3 Sobel operator is used to filter the image. 3×3 operator for Sobel is as this:

Fifth International Conference on Digital Image Processing (ICDIP 2013), edited by Yulin Wang, Xie Yi, Proc. of SPIE
Vol. 8878, 88780G · © 2013 SPIE · CCC code: 0277-786X/13/$18 · doi: 10.1117/12.2030559

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (1)$$

With the edges of original image, hough transform is carried out next to search vanishing lines. PVC Hough first brings this out in 1962[5]. The core of hough transform is the duality between line and point.

With lots of lines, we will select the effective vanishing lines, at the same time, reducing lines will reducing the complex of compute in the same time.

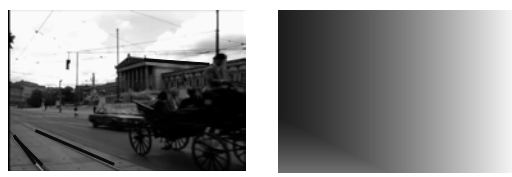
Lines of the similar slope are in the same cluster, We group lines into 12 to 18 clusters, only keep the cluster that contain lines more that 10% of all the lines. Most situation we get less than 6 clusters. Then pick out the longest in the lines that near slope average. At the end of this step, we get about less than 6 lines.

By intersect every pair of the lines, we get some potential vanishing points.

(b)Search step: To detect the final vanishing point, we vote for each point, long line is more reliable and smaller distance between point and line means the point is more reliable. So the vote for each point is the combine of distance and length, smaller value is more reliable:

$$vote(a) = \sum_{l=1}^{n} d_l \times l_l \quad (2)$$

Where d_l is the distance between point a and line l, l_l is the length of line l. The min value of vote is the final vanishing point, Figure 1a is the vanishing point, Figure 1b is the depth map from geometry.



(a)vanishing point (b)depth map

Figure 1. depth map from geometry

2.2 Motion detect and graph cut

In the existing algorithms, motion detect is a huge compute by block matching of every block, at the same time, when there's a big object of the same color moving in the image, block matching often have mistake occur inside the object. So we bring out this algorithm that combine motion detect with graph cut.

(a)Motion detect of the edge: First we get the edge of the image by Sobel operator talked before. Then find the match block in the next frame and the motion vector of it only if the block is at the edge of the image, Figure 2b is the absolute value of the motion vector after this section.

(b)Graph cut: We layer the image by the luminance and the color, after this, the point of similar color and luminance is in the same layer. Then find the connected component and label them. Lifeng He's algorithm[7] is used here to detect the pixels one by one, compare to the pixels near it and label the component segment, the image after this is as Figure 2c shows.

(c) The final motion: We get the segments of the image and the motion at the edge of the image before, computed one segments each time. For the current segment, some pixels is at the edge of the image and already have motion vector, some is inside the segment and not have motion vectors now. We sum the motion vector of the edge and average them as the below:

$$vector(m) = \frac{1}{N} \sum_{n=1}^{n=N} V_n \quad (3)$$

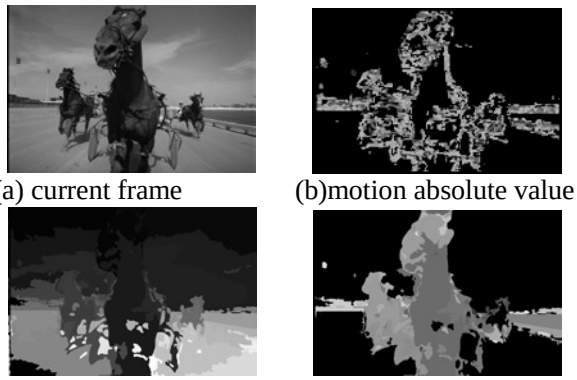
Where m is the number of the current segment, N is the number of pixels already have their motion vectors. We give the vector to all the pixels in the current segment, then we get the final motion vector value in Figure 2d.

2.3 Depth map generate

After all the compute before, we now have the vanishing point of geometry information and the motion vector of

the moving information. In this section, we will combine them together to get the depth map finally.

We take the status of the segment, moving or static, and the size of the segment into consideration. The static one's depth map is determined only by the vanishing point, the vanishing point represents the furthest point in depth map, otherpoints' depth is determined by the distance between vanishing point. The moving one's depth is the plus of geometry depth and the motion depth, for motion depth, the bigger motion vector represent the nearer in depth map. Then in each situation, we distinguish the segment into big one and small one by the number of pixels it contains, the small one will have the same depth all the pixels, which is the average of all the pixels depth, as Figure 3 shows.



(a) current frame (b)motion absolute value

(c)graph cut (d)horse motion at last

Figure 2. depth map from motion

Figure 3. Depth map combine geometry information and motion vector

3. EXPERIMENT AND DISCUSSION

The main compute of our algorithm is vanishing point finding and motion vector detect, so we just give out the result and compare of them separately.

3.1 vanishing point detect

We experiment on three image to detect our algorithm. The result is as below shows.

Table 1. Number of line involved in intersection

Image	Existing algorithm	Our algorithm
Car	19	4
Hall	13	7
Road	16	4

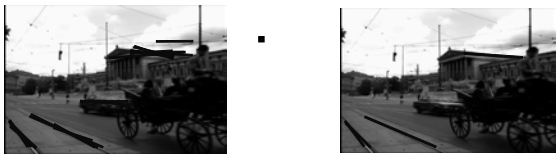
Table 2. vanishing point simulate time(s) in matlab

Image	Existing algorithm	Our algorithm
Car	2.08	2.00
Hall	0.43	0.38
Road	1.78	1.71

Table 1 shows the line involved in final intersection, Table 2 shows the compute time in matlab simulation. We can come to the conclusion that our method is easier and faster.

In the image car, our algorithm is a little exact; in hall, an error occurs in the existing algorithm because of information lost, in our method, that does not happen, in road, both algorithm performs good.

We can see our algorithm is more efficiency and exact.



(a)car in existing algorithm (b)car in our algorithm

(c)hall in existing algorithm (d)hall in our algorithm

(e)road in existing algorithm (f)road in our algorithm

Figure 4. Vanishing point detect in our algorithm and existing ones

3.2 motion detect

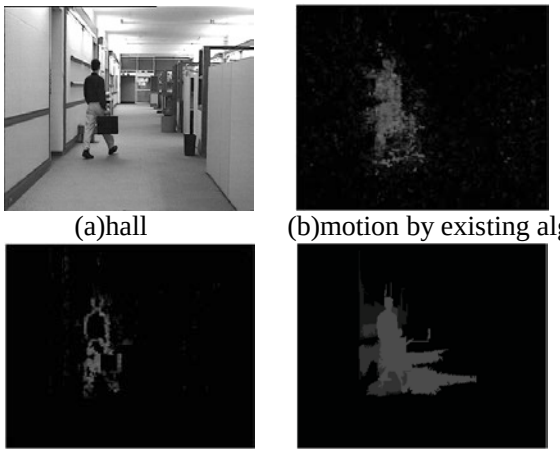
Here we just give out the result of the value as we need. Table 3 is the block involved in block matching for motion detect, the number is much less in our algorithm, the time using is much less, as Table 4 shows, from these two table, we can see our algorithm is much efficiency. And when it comes to the effect, the result is in Figure 5. We can see that our algorithm performs good, and for the background of the image, our method can reduce the interfaces better than the existing ones.

Table 3. Number of block involved

Image	Existing algorithm	Our algorithm
Hall	21301	5095
Horse	21301	6977
Woman	21301	1930

Table 4. Motion detect simulate time(s) in matlab

Image	Existing algorithm	Our algorithm
Hall	14.73	6.40
Horse	15.21	7.72
Woman	15.16	4.40



(a)hall (b)motion by existing algorithm

（c）motion at the edge (d)motion by our algorithm

Figure 5. motion detect and the depth map

3.3 The combine of motion vector and geometry information

Figure 3 is the final depth map of the hall scene, we can see that the depth map is more reasonable.

4. CONCLUSION

A proposed method of depth map generate for 2D to 3D conversion is bring out in this paper, we cut the algorithm into three main steps, find vanishing point for geometry information, detect motion vectors for depth by motion, and finally combine them together. In every step we made some improvement of the algorithm. At the section before, we can see our algorithm is much efficiency and exact. In the future, we can consider about hardware realization of the algorithm.

ACKNOWLEDGEMENTS

This work is supported by the Important National Science & Technology Important Specific Projects of China (Grant No.2009ZX01033-001-010).

REFERENCES

[1] C. Fehn, “Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv,” in Proceedings of SPIE, vol. 5291, 2004, p. 93.

[2] ——, “A 3d-tv approach using depth-image-based rendering (dibr),” in Proc. of VIIP, vol. 3, 2003.

[3] S. Battiato, S. Curti, M. La Cascia, M. Tortora, and E. Scordato, “Depth map generation by image classification,” in Proceedings of SPIE, vol. 5302, 2004, pp. 95–104.

[4] Y. Matsumoto, H. Terasaki, K. Sugimoto, and T. Arakawa, “Conversion system of monocular image sequence to stereo using motion parallax,” in Proceedings of SPIE, vol. 3012, 1997, p. 108.

[5] P. Hough, “Method and means for recognizing complex patterns,” Tech. Rep., 1962.

[6] S. yuan and T. zheng, “New automatic vanishing point detection algorithm,” Infrared and Laser Engineering, vol. 32, no. 005, pp. 479–483, 2003.

[7] L. He, Y. Chao, K. Suzuki, and K. Wu, “Fast connected-component labeling,” Pattern Recognition, vol. 42, no. 9, pp. 1977–1987, 2009.