2D-to-3D Conversion Based on Motion and Color Mergence

Feng Xu, Guihua Er*, Xudong Xie*, Qionghai Dai**
*Department of Automation
Tsinghua University
Beijing 10084, China
xufeng07@mails.tsinghua.edu.cn
{ergh,xdxie, qhdai}@mail.tsinghua.edu.cn

ABSTRACT

In this paper, we present an efficient scheme to synthesize stereoscopic video from monoscopic video. We use the improved optical flow method to extract pixel-level motion for each frame. By considering the intensity of the estimated motion, we can classify the moving objects. Then, to achieve more accurate classification, we combine color information in the frame using the method derives from the *minimum discrimination information* (MDI) principle. Finally, constraints-involved flood-fill method is developed to segment the frame and assign depth values for different segmented regions. The experimental results show that our scheme achieves good performances on both segmentation and depth determination.

Index Terms—Stereo vision, 3D display, Motion analysis, Median filters

1. INTRODUCTION

Three-dimensional (3D) visualization has been becoming more and more popular in recent years. 3D display devices, such as autostereoscopic displays, provide us vivid depth cues as we experienced in daily life. However, compared with the growing number of stereoscopic devices, the 3D content for display is obviously insufficient. On the other hand, there exists tremendous amount of 2D video resources. As a result, the 2D to 3D video conversion becomes a common interest of both academic and industrial communities.

Early works in stereoscopic video generation employ 3D geometry [1]. However, it is difficult to obtain 3D models of real world scenes merely from monoscopic videos. In that case, various methods have been proposed to generate stereo views from monoscopic video sequence. Some of them are dedicated to select or synthesis stereo image pair from original image sequence without computing the depth map [2, 3, 4]. They use two neighboring fames as the left and right image based on the analysis of different motions between the camera and objects. Nevertheless, these methods require the captured scene and objects remain almost stationary and the motion of camera should introduce a horizontal parallax, which restricts the practical implementation in certain scenario. Furthermore, these methods can only render binocular stereo image pairs, which is not suitable for multi-view situation. The other schemes to generate 3D video need obtain the depth maps from original 2D video. According to the different methods for acquiring depth information, we can further classify these depth-based methods. The first category is *depth-from-color* [5], which only process single image. The second is *depth-from-motion*. If the motion is caused by the moving camera, the methods can be traced back to an active computer vision area called *structure-from-motion* [6], which requires static scenes. When the motion also involves the scene variation, block matching techniques [7] are helpful, like the MPEG motion [8, 9] which can be extracted directly from the MPEG coder. But this block motion is designed for compression efficiency rather than expressing the real motion, so it can not fully exploit the video's motion information. As a result, some researchers design 2D to 3D video conversion schemes based on the fact that color and motion information are both useful to obtain the depth of the scene. Chang et al. [10] explore the motion by frame difference method, and use K-Means algorithm to realize color segmentation, so the depth map is acquired from both time and spatial information. However, frame difference method needs large number of neighboring frames to achieve good performance, and K-Means algorithm requires manual operation to define the initial groups.

In this situation, we propose our method which using both motion and color information to synthesize stereoscopic video from monoscopic video. The procedure of our method is shown in Fig. 1. Our scheme has the following contributions compared with previous works: *optical flow* is used to estimate the 2D motion on pixel level which enables more detailed result than block-based motion extraction

This work was supported in part by The Distinguished Young Scholars of NSFC, No.60525111 and in part by the 863 Program, 2007AA01Z332 .

The authors are with Tsinghua University, Beijing 100084, China (e-mail: xufeng07@mails.tsinghua.edu.cn).

978-1-4244-1755-1/08/$25.00 ©2008 IEEE

methods. Then the *minimum discrimination information* (MDI) principle is employed to combine the color information with the optical flow result to prepare for accurate segmentation. Further more, it enables an automatic depth decision process. Finally, we add several constraints to the flood-fill method to achieve segmentation and decide the depth value of each segmented region. Unlike deciding the depth only by the motion intensity [8], these constraints are more reasonable for the existent 2D videos. Experiment results show that our scheme can facilitate automatic stereo vision and 3D display.

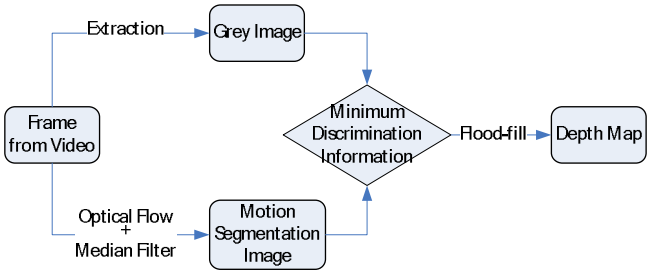

Fig. 1. The whole process of our scheme

2. MOTION EXTRACTION

The idea of using pixels' 2D motion to obtain depth map is based on the fact that objects with different motions usually have different depths. Color information dose not possess this property, because on one depth layer, object may have different colors. To estimate the motion, we use Lucas and Kanade's optical flow method [11]. And in order to settle the strenuous motion problem, we choose the pyramidal method [12], which gives better performance.

As we know, it is difficult for the optical flow method to calculate accurate motions for pixels that located in sharp edges or low texture regions. On the other hand, we need all pixels motions to synthesize depth map. Therefore special improvement need be added to the optical flow method in our application.

1. We run optical flow algorithm in 3×3 blocks which is the best size among all the tested sizes;

2. According with the credibility of pixels' optical flow results, which is decided by the texture of the region, suitable weighted values w_i for the pixels in block Ω are chose. And one weighted average value V_{block} for the block based on the pixels' optical flow values v_i is calculated:

$$V_{block} = \frac{1}{\sum_{i\in\Omega} w_i} \sum_{i\in\Omega} w_i v_i ; \qquad (1)$$

3. If the whole block does not provide enough credibility, we propagate the neighboring block's result into the current one.

After this procedure, we get more reliable optical flow result. At the same time, some details of the motion are lost. But this step will not cause bad affects in the final result for the reason that these details do not have strong relationship with the pixels' depths. Further more, in this procedure, we only use the assumption that neighboring pixels have similar motions which is already used in the optical flow theory [11], so no more empirical assumptions are brought in.

Another improvement in our optical flow method is towards the consistency of the optical flow result. According to the optical flow results shown in Fig. 2 and Fig. 3, we see that the brightness of the object and the background is not that consistent. Therefore we use a median filter to refine these optical flow results. Though the filter solves this problem, it loses some information of the edges which is rather important for segmentation. Therefore, we need a special process to solve this problem.

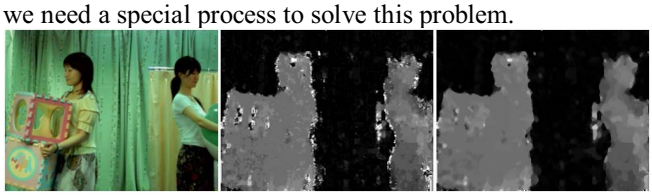

Fig. 2. The "Akko&Koyo" video is about two girls walking in the scene (published by Tanimoto Laboratory, Nagoya University). The left image is one video frame. The middle image is the optical flow intensity map. The right image is the median filtered map.

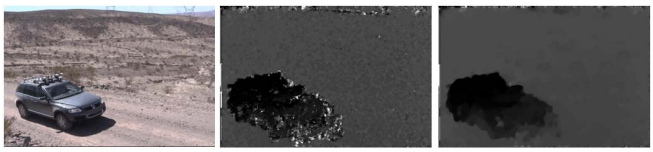

Fig. 3. The "jeep" video is about a jeep running along the road captured by a moving camera (published by Artificial Intelligence Lab, Stanford University). The left image is one video frame. The middle image is the optical flow intensity map. The right image is the median filtered map.

3. COLOR INFORMATION MERGENCE

Since each edge has different color on both of its sides, using color information is an effective method to detect the exact locations of the edges. Conventional color based image segmentation methods attempt to segment the whole image. However, we only need to locate the edges of the objects. Consequently, if we directly bring in color based segmentation method, it will not match the previous method perfectly. In this situation, a minimum discrimination information based method is developed to integrate the color information into the edge areas and prepares for accurate segmentation.

By using the minimum discrimination information principle [13], we can find a probability density q which satisfies all the constraints and has minimum discrimination information with a reference p. In our application, we regard the objective image's brightness as the probability density q, and the segmented image's brightness obtained by the optical flow method as the reference p. To construct a reasonable constraint, we consider the brightness of the corresponding video frame as another function r. After that, we can use the minimum discrimination information

principle to synthesis the objective image by minimizing two things at the same time: first, the error between q and r; second, the discrimination information between q and p.

The details are described as follows:

To describe in a brief way, we call the segmented image obtained by the optical flow method as reference image and the grey image extracted from the corresponding video frame as original image.

Since the functions used in the minimum discrimination information principle are all probability densities, we should transform the images' brightness as follows:

$$p(i)=\frac{k(i)}{\sum_{i=1}^{N^2}k(i)}, q(i)=\frac{f(i)}{\sum_{i=1}^{N^2}f(i)}, r(i)=\frac{g(i)}{\sum_{i=1}^{N^2}g(i)} \quad (2)$$

where $k(i), f(i)$ and $g(i)$ denote the i th pixel's brightness value in the $N \times N$ reference image, objective image and original image.

Without loss of generality, we can normalize the images brightness values to satisfy

$$\sum_{i=1}^{N^2}k(i)=\sum_{i=1}^{N^2}f(i)=\sum_{i=1}^{N^2}g(i). \quad (3)$$

So the discrimination information

$$I(q(x),p(x))=\int q(x)\log\frac{q(x)}{p(x)}dx \quad (4)$$

can be written as

$$I(f,k)=\sum_{i=1}^{N^2}f(i)\log\frac{f(i)}{k(i)}. \quad (5)$$

After adding the constraint that minimizing the error between the objective image and the original image, the final function we should minimize is written as:

$$J(f,\lambda)=\| g-f \|^2+\lambda\sum_{i=1}^{N^2}f(i)\log\frac{f(i)}{k(i)}. \quad (6)$$

Where, λ decides the objective image's degree of dependence on the reference image, which is chose from 300 to 500 experientially.

To minimize $J(f,\lambda)$, we get an equation set

$$\frac{\partial J(f,\lambda)}{\partial f}=2(f-g)+\lambda\begin{pmatrix}1+\log f(1)-\log k(1)\\1+\log f(2)-\log k(2)\\\vdots\\1+\log f(N^2)-\log k(N^2)\end{pmatrix}=0. \quad (7)$$

For certain λ, we have N^2 nonlinear equations, and every equation has its own unknown variable $f(i)$. As we know, $f(i)$ is the value of pixel's brightness, it's an integer between 0 and 255. Hence we can use the bisection method to solve these nonlinear equations by at most 8 iterations. The original, reference and objective images are shown in Fig. 4 and Fig. 5. According to these images, we can conclude that the true edge areas become clear after this color-aided procedure, which will benefit the following procedure.

Fig. 4. The left image is one frame from the "Akko&Koyo" sequence. The middle image is the optical flow intensity map as the reference frame. The right image is the destination image.

Fig. 5. The left image is one frame from the "jeep" sequence. The middle image is the optical flow intensity map as the reference frame. The right image is the destination image.

4. SEGMENTATION AND DEPTH DETERMINATION

The image obtained by the previous procedures has a property that pixels belong to the same object have similar intensities. We can use it to segment, but to synthesize depth map, some reasonable rules are needed to assign segmented regions with suitable depth values. The proposed rules are as follows:

Rule 1. We suppose the pixels at the boundaries of the image belong to the background of the scene while the segmented regions in the center stand for the foreground objects.

Rule 2. One segmented object is assumed to have unique depth value.

Rule 3. Some background object like a wall or a building always has a consistent depth value and this value always implies the farthest distance in the scene. While other background like the land or the ocean usually has gradual depth values with the fact that, from top to bottom of the image, the scene becomes closer to the camera.

Although these rules present some exceptions like far located objects appearing in the center of the image, they are still suitable for most situations. Therefore, following these rules, we design our own segmentation and depth determination algorithm based on the flood-fill method. We first traverse all the pixels in the image obtained in the previous section. Our search starts from the four boundaries of the segmented image and ends at the center as shown in Fig. 6. Based on the brightness consistency of the neighboring pixels, we put similar pixels into one queue. If the number of these pixels is bigger than a threshold, we suppose these pixels to be the background or a new object and assign these pixels with one depth value different from other queues (This value becomes bigger according to the finding of new objects, because the searching order makes the new object probably be in the center of the image).

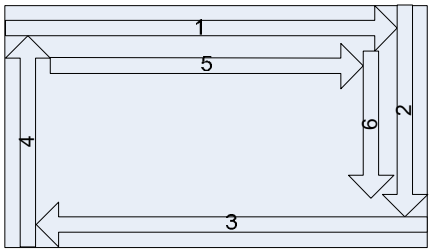

Fig. 6. The order of the pixels traversal: the rectangle stands for the image, the arrows stand for the traversal order. Only the first several arrows are shown in this figure.

Otherwise, these pixels are likely to be in the background or the neighboring object, so we assign these pixels with corresponding depth value.

Until now, our algorithm has followed *Rule 1* and *Rule 2* to segment the image and determine the depth values for all pixels in one frame. Then we use *Rule 3* to update the depth values of the background pixels. At first, we should detect the background region. This can be achieved by testing the pixels' locations in the image. After that, according to *Rule 3*, we change the values of the background pixels depending on the characteristic of the video which is classified manually.

5. EXPERIMENTAL RESULTS

After all the above steps, our final depth map is obtained. For the "jeep" video (Fig. 7), the background region's depth values change gradually according to the real scene. While for the "Akko&Koyo" video (Fig. 8), our algorithm is still verified to give outstanding performance both on object segmentation and depth determination.

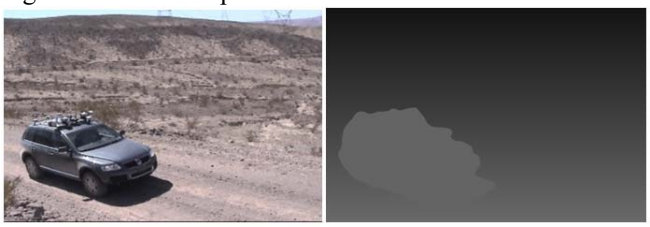
Fig. 7. one frame from the "jeep" video and its depth map


Fig. 8. one frame from the "Akko&Koyo" video and its depth map

From the experimental results we can see that the optical flow method can give pixel-level estimation of the motions in the scene, the motion and color mergence can help us finding accurate edges of the objects and the constraints-involved flood-fill method make the final depth map reasonable to the real scene.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we developed a new scheme to automatically achieve 2D to 3D converting. The proposed method achieves motion and color segmentation and reasonable depth determination which can be verified by the experimental results. Future works should focus on improving the performance of the method. The ambiguity of the segmented edges can be handled by introducing some improved optical method [14] and using anisotropic diffusion to replace the median filter. At the same time, more complicated scenes ask us using more reasonable rules to determining the depth.

7. REFERENCES

[1] D.V. Morland, "Computer-Generated Stereograms: A New Dimension for the Graphic Arts," Proc. SIGGRAPH '76, pp. 19-24, 1976.

[2] M. Kim and et al., "Stereoscopic conversion of monoscopic video by the transformation of vertical-to-horizontal disparity", SPIE Vol. 3295, Photonic West, pp.65-75,Jan. 1998.

[3] K. Man Bae, N. Jeho, B. Woonhak, S. Jungwha, H. Jinwoo, "The adaptation of 3D stereoscopic video in MPEG-21 DIA", Signal Processing: Image Communication Volume: 18(8), pp. 685-697, 2003.

[4] K.Manbae, P.Sanghoon, and C.Youngran "Object-Based Stereoscopic Conversion of MPEG-4 Encoded Data," Lecture Notes in Computer Science, Vol 3, pp. 491-498, Dec. 2004.

[5] S. Battiato, A. Capra, S. Curti, and M La Cascia, "3d stereoscopic image pairs by depth-map generation," Proceedings of the 2nd International Symposium 3D Data Processing, Visualization and Transmission, 3DPVT, pp.124–131, Sept. 2004.

[6] S. Knorr, A. Smolic, T. Sikora, "From 2D- to Stereo- to Multi-view Video", 3DTV07(1-4)

[7] Y. Matsumoto and et. al, "Conversion syslem of monocular image sequence to stereo using motion parallax", SPIE. Vol 3012, pp. 108-115, 1997.

[8] I.A. Ideses, L.P. Yaroslavsky, R. Vistuch, B. Fishbain, "3D video from compressed 2D video", SPIE and IS&T, Proceedings of Stereoscopic Displays and Applications XVIII, San Jose, CA, 2007.

[9] I.A. Ideses, L.P. Yaroslavsky, B. Fishbain, "Real-time 2D to 3D video conversion", RealTimeIP(2), No. 1, pp. 3-9, October 2007.

[10] C. Yu-Lin, F. Chih-Ying, D. Li-Fu, C. Shao-Yi, and C. Liang-Gee, "Depth Map Generation for 2D-to-3D Conversion by Short-Term Motion Assisted Color Segmentation", IEEE International Conference on Multimedia and Expo, 2007.

[11] B. Lucas, and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI), pp. 674-679, 1981.

[12] Jean-Yves Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm" Intel Corporation Microprocessor Research Labs.

[13] Z. Xue-Long, "Fundamentals of applied information theory", Tsinghua University Press, Beijing, 2001.

[14] N.Papenberg, A.Bruhn, T.Brox, S.Didas, and J.Weickert, "Highly Accurate Optic Flow Computation with Theoretically Justified Warping". International Journal of Computer Vision, Vol. 67/2, 141-158, 2006.