Disparity Measurement Using Dynamic Programming

Yu-Cheng Fan , Yan-Hong Jiang and Chieh-Lin Chen

Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan
E-mail: skystarfan@ntu.edu.tw

Abstract—**In this paper, we proposed a method of depth map measurement with dynamic programming which calculates the disparity of binocular stereo camera. We scan the intensity horizontally of the two images captured by the stereo camera and record the cumulative error value of each pixel in the scanline to find the minimum cumulative error to get the optimized path. With the optimized path, we can find the reasonable and dense depth map. Furthermore, in order to reach real-time processing, we proposed a simplifying algorithm to reduce the computational complexity of dynamic programming.**

Keywords-3D-TV; Depth Map Measurement; Dynamic Programming; Disparity Estimation

I. Introduction

In recent year, 3D technologies were concerned again. Since the most successful 3D movie - Avatar in 2009, there are more and more 3D movies were played in the theater, in addition, TV manufacturers like Sony, Panasonic, Samsung or LG were also released their 3D-TV one after another in 2010. Except for TVs, 3D technologies were also applied to the projectors, mobile phones, advertisement board and so on. It's really into our live and put the technologies of display in to a new generation.

However, it's still facing some bottlenecks now even though the 3D products are so marketable nowadays. One of the crucial problems of the 3D industry is the content. The traditional video sources captured as single-viewed videos so it can not present any 3D effect on the 3D-TV. ATTEST, an organization of advanced three-dimensional television system technology proposed an effective method to use a single view image and corresponded depth information, instead of transmitting enormous amount information from every views of the images. In this paper, we devoted to the technology of depth map measurement for accelerate the production of 3D content and portable devices with 3D displayers.

In the literature, an algorithm was proposed to taxonomy and evaluation of dense two-frame stereo correspondence by Scharstein [1]. Xu [2] and Gong [3] reconstructed depth image using dynamic programming. Xu [2] uses a binocular stereo camera as their source and down-sampling the images to process images on real-time. After dynamic programming is finished, they then using the block erosion to restore the resolution of the depth map. Gong [3] proposed reliability-based dynamic programming and consistency constraints for fast stereo matching. With the consistency constraints and the reliable matching, the method can provide depth map more precisely.

In this paper, we adopt the method mentioned in [1] to measure and reconstruct depth map with four steps:

- Matching cost computation
- Cost aggregation
- Disparity computation
- Disparity refinement

To implement the four steps, we use different algorithm for each step and we will introduce the detailed algorithms in next section.

II. Proposed Method

We use binocular stereo camera to get the two view image and measure depth information through dynamic programming which produces depth map densely and precisely in four steps mentioned in last section. Fig. 1 shows our system architecture.

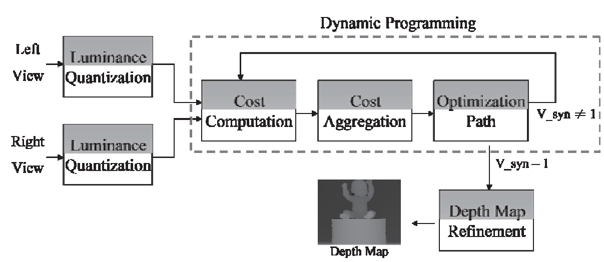

Figure 1. System architecture

Luminance Quantization

We first get the gray-level image of the two view from camera and quantize each pixel into five bits for two reason – first, it can remove small variation of pixels which do not influence the quality of depth deeply and will also reduce the computation during depth map measurement.

Matching Cost Computation

To compute the matching cost of image by efficient way, we scan the image horizontally to calculate the different value between each pixel with SAD, shown in Fig. 2. Besides, there

978-1-4577-1772-7/12/$26.00 ©2012 IEEE

are more than one way of matching cost computation - pixel-based, window-based or single-sided window-based. The first two methods were described as following equations:

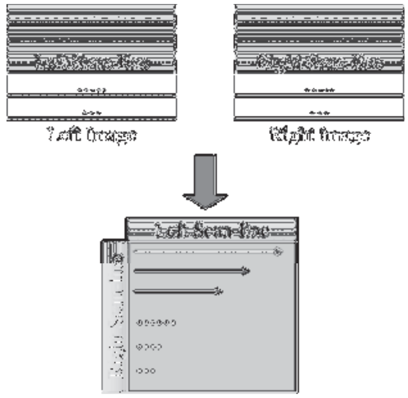

Figure 2. Pixel-based cost computation

$$D_p(xL, xR) = | I(xL, y) - I(xR, y) |, 0 \le xL, xR \le m \tag{1}$$

$$D_w(xL, xR) = \sum_{k=-r}^{r} \sum_{l=-r}^{r} | I(xL + l, y + k) - I(xR + l, y + k) |, 0 \le xL, xR \le m \tag{2}$$

Single-sided window-based method is a variation of window based method to prevent the effect of occlusion. It separates the window into left side and right side and choose the smaller one as the matching cost to ignore the unusual high value caused by the occlusion area. The expressions are as follows:

$$D_{Left}(xL, xR) = \sum_{k=-r}^{r} \sum_{l=-r}^{0} | I(xL + l, y + k) - I(xR + l, y + k) |, \quad 0 \le xL, xR \le m \tag{3}$$

$$D_{Right}(xL, xR) = \sum_{k=-r}^{r} \sum_{l=0}^{r} | I(xL + l, y + k) - I(xR + l, y + k) |, \quad 0 \le xL, xR \le m \tag{4}$$

$$D_{sw} = Min(D_{Left}, D_{Right}) \tag{5}$$

Cost Aggregation

After matching cost computation, we will obtain an array composed of SAD value of each pair of pixel. In this step, we have to accumulate the cost to obtain a path with smallest value. Cost aggregation start with the most northwest and terminate at the most southeast of the cost array. To simplify the computation of finding the path of smallest value, we will choose the smallest value from three direction, north, west and northwest, to accumulate so the value will be the smallest along the path from the starting point as we shown in Fig. 3. At the same time, we will also record the direction that we were chosen like Fig. 3(right). The equation used for determine the value to accumulate is as follow:

$$Aeg(j, i) = Cost(j, i) + Min(UpLeft, Up, Left) \tag{6}$$

$$UpLeft = Cost(j-1, i-1)$$
$$Up = Cost(j-1, i)$$
$$Left = Cost(j, i-1) \tag{7}$$

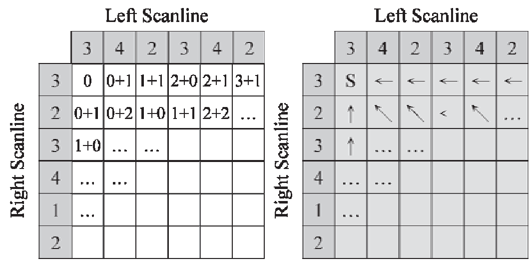

Figure 3. Cost aggregation(left) and possible path recording(right)

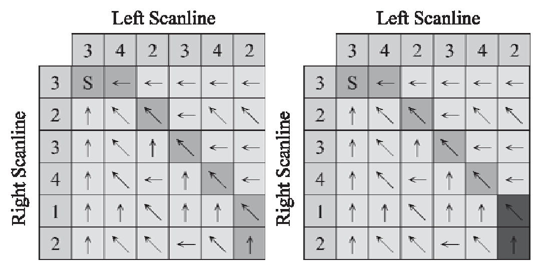

Figure 4. Optimized path(left) and the pixel with occlusion(right)

Disparity Computation

Before computing the disparity, we have to find the optimized path first. The optimized path is also the path with the smallest aggregated cost value we have mentioned in last paragraph. To get the optimized path, we must start with the very last pixel and go along with the direction we have recorded in the previous step. The result of searching optimized path is shown in Fig. 4(left). There will have three directions in the path, each direction have its meaning. The direction of northwest means the pixel is not occluded both in the left scanline and right scanline and the direction of west means the pixel in the left scanline is occluded in the right scanline. On the contrary, the north direction means the pixel in the right scanline is occluded in the left scanline. As long as we got the optimized path, the disparity based on the left scanline then can be computed by subtracting the index of left scanline from index of right scanline.

Disparity Refinement

There are two method were used to refine the disparity we measured in this paper. The first operation is using median filter to improve the problem of discontinuous edge of the disparity map because the median filter is well known as its ability of removing the impulse content in the image with considerably less blurring effect than linear filter. The other refinement is to fill the reasonable value into the occluded area. Because our disparity map is based on the left-view image, so the occlusion is only occur at the direction of the optimized path is north as shown in Fig. 4(right). To fill the occluded disparity of pixel, we used the linear interpolation to generate a reasonable value of disparity.

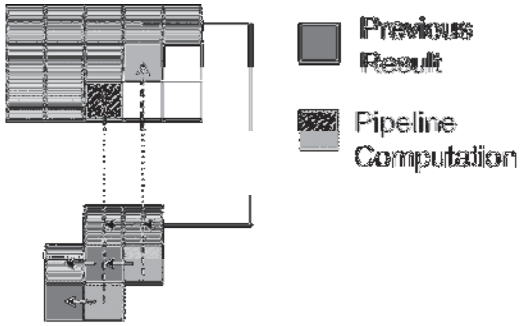

Figure 5. Architecture of parallel processing

Reduced Dynamic Programming

Dynamic programming reduces a n-factorial process into n sub-processes to simplify the computational complexity. However, it still have too plenty of computation to reach real-time process even it reduces lots of computation of searching disparity from two-view image. So we used three reducing algorithm to accelerate our disparity measurement system. The methods will be explain as follow.

1) *Limitation of Disparity:* Cost aggregation of dynamic programming uses scanline of two-view image to find the optimized path. The matrix used for save the aggregated result composed of m by m matrix where m is the width of the image. Each pixel in the matrix represents different disparity value with range –m to m. Nevertheless, we measured disparity based on the left-view image, so the value of measured disparity will always be positive value. Besides, the maximum value of disparity will less than 0.25 times of image width in normal images. Therefore we can limit the range with 0 to 160 when searching disparity in case of VGA to save lots of computation.
2) *Aggregation Matrix Reduce:* Although the computation reduced enormously with limitation of disparity, the usage of memory is still be the same and the utility of memory will be lower than 50 percent. So we convert the available range of disparity into a smaller matrix composed of the disparity and scanline of left image which is 1/4 size compare to original memory.
3) *Parallel Processing:* The computation of dynamic programming is concentrated on the inter computing of scanline of left view and right view. Therefore
4) we use parallel processing when cost aggregation. N parallel processing unit can reduce run time to almost 1/N compared to original.
   We need three direction of data when cost aggregation, north, west and northwest. So we can preserve current result of aggregated value for the next parallel processing unit to compute as shown in Fig. 5. With this method, we not only reduced the computation time but also lower the access of memory.
5) *Down-sampling:* Dynamic programming can produce dense and reasonable disparity information with global cost aggregation. However, the process of cost aggregation cost too much computation to reach real-time processing even we used three reduction method mentioned in this section. Thus we down-sampled the image with the average of a mask which size is 2 by 2. With down-sampling, we can also drop the limitation of disparity to 1/2 compare to original. Hence the reduction is quite effective. Furthermore, the usage of memory can be reduced at the same time.

III. Experimental Result

In this section, we will present the experimental result of disparity measurement using dynamic programming with 6 standard image which size is 640 by 480 to evaluate the quality of disparity image, processing time and compare to other method in the literature.

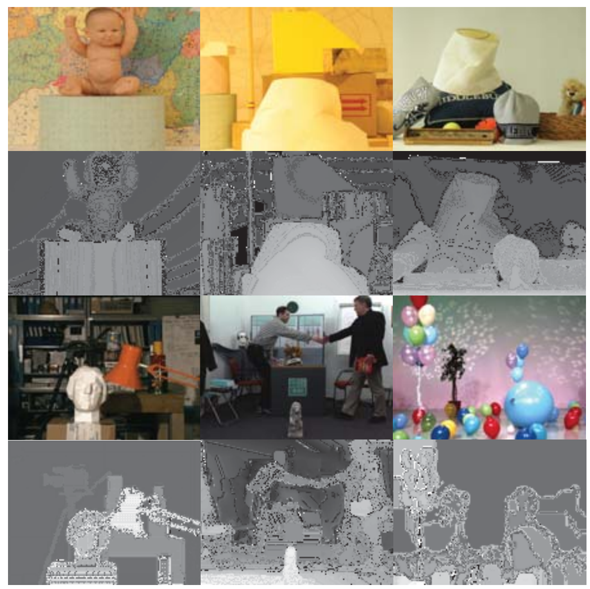

Figure 6. Testing image and the corresponding ground truth

$$MSE = \frac{\| I(x) - R(x) \|}{N} \tag{8}$$

$$PSNR = 10 \times \log_{10}\left(\frac{255^2}{MSE}\right) \tag{9}$$

There are three method when calculating matching cost, in order to evaluate which one is better for our algorithm, we experimented all of the method which are pixel-based, window-based and single-sided window-based method with the same image and evaluate with PSNR shown in (8) and (9), yet the results of PSNR are probably quite low because we use the ground truth, which is absolute depth map, as the reference to compute the PSNR value but the measured disparity is a relative disparity. So the PSNR value is used to compare the effect between different methods.

Table I shows the PSNR of different method of matching cost computation and the PSNR after refinement. Before discussing the result, we must understand that the PSNR value is used to evaluate the correlation between two groups of data. However, the disparity measured from our method, dynamic programming, is a relative result instead of absolute one. So we can discuss the performance of different methods through the change of the PSNR value instead of assess the result from PSNR value directly.

We can learn from table I that the window-based method is better than the pixel-based one in dynamic programming because the pixel-based matching is much harder to distinct the correct-matched pixel and the wrong one. After refine the result image with the method mentioned in the proposed method, we can gain better result from the window-based one.

The experimental result of our proposed method is shown in Fig. 7 and Fig. 8. Fig. 7 is the result using window-based matching cost computation and there are a lot of broken areas in the measured disparity image, so the refinement algorithm is also performed in the Fig. 8. From the figures, we can observe that lots of the broken area in Fig. 7 is repaired in the Fig.8 by using the refinement algorithm mentioned above.

Figure 7. The disparity of window-based method before refinement

Figure 8. The disparity of window-based method after refinement

TABLE I. METHOD COMPARISON OF COST COMPUTATION

Image	**PSNR**		
	Pixel-based	**Window-based**	**After Refinement**
1	15.18 dB	15.74 dB	18.49 dB
2	11.96 dB	12.27 dB	12.47 dB
3	13.34 dB	13.46 dB	14.99 dB
4	13.26 dB	13.50 dB	15.16 dB
5	11.79 dB	12.38 dB	12.77 dB
6	9.52 dB	13.15 dB	13.40 dB
Average	12.51 dB	13.42 dB	14.55 dB

Table 2 presented the comparison result of proposed method and other method from literatures. Three of the methods in the table were made with dynamic programming algorithm and we proposed different reduction method to reach real-time processing. We simulated in different disparity level and resolution, include 256×256, 320×240, 512×512 and 640×480 to compare the processing time. From the table, we can discover that the performance of our method is better than the others methods proposed before.

TABLE II. COMPARISON OF PROCESSING TIME

Method	**Resolution**	**Disparity Level**	**Processing Time**
Proposed	640x480	100	187 ms
		50	141 ms
		48	140 ms
	512x512	100	203 ms
		50	156 ms
		48	156 ms
	320x240	100	31 ms
		50	16 ms
		48	15 ms
	256x256	100	32 ms
		50	16 ms
		48	16 ms
Xu[n-1]	512x512	48	171 ms
	256x256	48	46 ms
Gong[n]	640x480	100	208 ms
		50	106 ms
	320x240	100	80 ms
		50	40 ms

IV. CONCLUSION

In this paper, we analyzed and reduced the method of disparity measurement using dynamic programming to provide a real-time method to get the disparity image from binocular stereo camera. The method of dynamic programming has high computational complexity and the method proposed not only improved the processing time but also save lots of usage of memory. However, there are still something to improve like the resolution can reach real-time process is still quite low, so we plan to implement the method with integrated circuit to improve the processing speed further in the future.

ACKNOWLEDGMENT

This work was supported by the Taiwan E-learning and Digital Archives Programs (TELDAP) sponsored by the National Science Council of Taiwan under Grants NSC 100-2631-H-027-003-. The authors gratefully acknowledge the Chip Implementation Center (CIC), for supplying the technology models used in IC design.

REFERENCES

[1] D. Scharstein, R. Szeliski, and R. Zabih, "A Taxonomy and Evaluation of Dense Two-frame Stereo Correspondence Algorithms," IEEE Workshop on Stereo and Multi-Baseline Vision, Kauai, USA, Dec. 9-10, 2001, pp. 131-140.

[2] X. Xu, X. Xie, and Q. Dai, "Real-Time 3D Video Synthesis from Binocular Stereo Camera," 3DTV Conference on The True Vision - Capture, Transmission and Display of 3D Video, Istanbul, May 28-30, 2008. pp. 133-136.

[3] M. Gong, and Y. H. Yang, "Fast Stereo Matching Using Reliability-based Dynamic Programming and Consistency Constraints," IEEE International Conference on Computer Vision, Nice, France , vol. 1, Oct. 13-16, 2003, pp.610-617.