DISPARITY ESTIMATION USING BLOCK MATCHING AND DYNAMIC PROGRAMMING

J. Karathanasis[1,2] *D. Kalivas*[2] *J. Vlontzos*[2]

[1]Intracom S.A., Development Programmes Department, Peania, Attica, Greece
[2]Applied Electronics Lab., Dept. of Electrical Engineering and Computer Technology, University of Patras, Rio, Greece

ABSTRACT

In this paper, we present a novel disparity estimation algorithm which is a combination of a block matching part and a dynamic programming part. The block matching gives an initial estimate of the disparity field while the dynamic programming estimates a smooth disparity field. The proposed algorithm is also studied in terms of hardware complexity. The algorithm is tested using real image sequences. The estimated disparity fields are used for the calculation of intermediate views using a simple interpolation/extrapolation algorithm.

1. INTRODUCTION

In a two camera system the images are the 2D projections of a 3D scene in the image planes of the cameras. Because of the relative camera displacement, the 2D projections are different. These differences are called disparities. The knowledge of the disparities is very important for the understanding of a 3D scene. For example if the camera system parameters are known then the disparity field can be used for the calculation of the depth. The problem of disparity estimation has been attracted the interest of many researchers [1, 2]. The disparity problem is similar to the motion estimation problem in the sense that both are correspondence problems between two images. The motion and disparity estimation algorithms can be classified as

- Pixel based algorithms
- Feature based algorithms
- Object based algorithms.

The pixel based techniques are simple but they give correspondence errors when occlusions are present and unreliable estimates in uniform areas. The feature and object based methods are more robust but they assume preprocessing steps like feature extraction and object segmentation which are extremely difficult problems.

In many applications like telepresence, medical imaging, quality control and video production where multicamera systems are used, the synthesis of intermediate views are necessary [3, 4, 5]. In this paper, we propose a novel algorithm for disparity estimation. The estimated disparity fields are being used for the calculation of intermediate views.

The proposed algorithm is a combination of block matching and dynamic programming [6]. The block matching part calculates the matching costs for the correspondings of every pixel in the left image to all pixels in the searching region in the right image. The matching cost is equal to the sum of absolute differences within a block of user selectable size. Each pixel in the left image can be assigned to any pixel of the right image within the searching region with a corresponding matching cost. The common block matching techniques choose the correspondence with the minimum matching cost. However, the derived matches are not reliable in uniform and occluded areas. In order to overcome this problem, we have a postprocessing part where all possible correspondences with their matching costs are fed to a dynamic programming algorithm. In this stage, each combination of matches corresponds to a path in a match space and the path that minimizes the total cost is selected as the array containing the best disparities. The cost function to be minimized takes into account the smoothness of the disparity field. The disparity field is further enhanced by a vertical median filtering utilizing the vertical correlation.

The dynamic programming part assumes horizontal disparity. This is not a severe restriction because in a two-camera system the cameras are usually horizontally aligned. When this is not possible and vertical disparity also appears then a preprocessing step is necessary for the geometric transformation of the images.

The proposed algorithm is also being investigated in terms of hardware complexity. The block matching part can be implemented using existing techniques, which have been applied in motion estimation. For the dynamic programming part, we indicate the analogy of its main processing per pixel to a matrix- vector multiplication.

2. DESCRIPTION OF THE DISPARITY ESTIMATION ALGORITHM

The proposed algorithm is a combination of block matching and dynamic programming. The block diagram of the algorithm is shown in Fig.1.

2.1. Block Matching

The block matching part calculates the matching costs for the correspondences of every pixel of the left image f_L to all pixels in the searching region S of the right image f_R. The searching region is specified by the maximum magnitude of the disparity vectors. The matching cost $mc(i,j;m,n)$ of

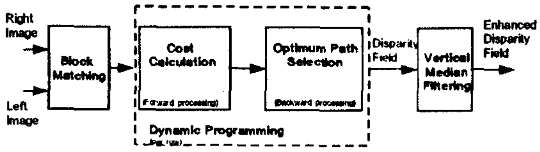

Figure 1. Block diagram of the algorithm

assigning pixel (i, j) of the left image to pixel (m, n) of the right image is equal to

$$\mathrm{mc(i,j;m,n)} = \sum_{\mathrm{k,l}} |\mathrm{f_L(i+k,j+l)} - \mathrm{f_R(m+k,n+l)}|, \quad (1)$$

where $-N \leq k, l \leq N$ and (m, n) belongs to corresponding searching region of (i, j). The matching cost is calculated over blocks of size $(2N+1)*(2N+1)$, where N is user selectable.

2.2. Dynamic Programming Processing

The dynamic programming processing of each row is independent because we assume horizontal disparity. For the ith row, the array $mc(i, j; i, n)$ is derived from the block matching part. Each node (j, n) of the dynamic programming state has an associated cost

$$bc(j,n) = \begin{cases} \mathrm{mc(i,j;i,n)} & j - dmax \leq n \leq j + dmax \\ \infty & \text{otherwise,} \end{cases}$$

where d_{max} is the maximum disparity. The Dynamic Programming algorithm has a forward part (cost calculation) and a backward part (optimum path selection). In the forward part for each possible matching (j, n), a path cost $pc(j, n)$ is calculated by

$$pc(j,n) = \min_{l}[pc(j-1,l) + dc(l,n)], \quad (2)$$

where $j-1-d_{max} \leq l \leq j-1+d_{max}$. The term $dc(l, n)$ represents the cost of assigning the neighboring pixels j and $j-1$ of the left corresponding row to the pixels n and l of the right row. The corresponding disparities are $n-j$ and $l-j+1$. The difference between the two disparities is equal to $(n-l-1)$. Assuming that the disparity field is smooth, the cost $dc(l, n)$ is chosen to be equal to

$$dc(l,n) = c(n-l-1)^2, \quad (3)$$

where c is a smoothing parameter. Therefore, a large difference of disparity of neighboring pixels implies a large matching cost. A more complicated disparity cost function, where discontinuities and occlusions are considered, is being currently studied. The value of l which minimises $pc(j, n)$ in Eq. (2) is kept in the pointer array $p(j, n)$. Thus,

$$p(j,n) = l. \quad (4)$$

The pointer array indicates that if in the backward part the minimum cost path passes through matching point (j, n) then the pixel $j-1$ matches to pixel $p(j, n)$. The backward part starts from the matching of the last pixel. The matching with the minimum path cost is chosen and then the rest matchings back to the first pixel of the row are given by the pointer.

2.3. The vertical median filtering part

Eventhough the estimated disparity field, which is derived by the dynamic programming part, is smooth, a few errors may be present. These errors are sparse and can be easily removed by a vertical median filtering [7] utilizing the vertical correlation of the disparity field.

3. HARDWARE CONSIDERATIONS

3.1. Block Matching

This problem has been extensively examined in the literature. Although many fast algorithms like the logarithmic search of [8] have been developed for the block-matching problem, this problem is usually tackled using full–search procedures which are optimal, need regular memory management, and can be easily parallelised. Several parallel–input array architectures for full–search block–matching have been derived in [9]. Multiple–input array architectures have been also described in [10], [11]. In [10], the input bandwidth problem for the search area has been solved by on–chip line buffers, which allow a low frame–buffer access rate. In [12], a systolic architecture requiring sequential input but being able to perform parallel processing has been introduced. Finally, the problem of flexibility and cascadability of the arrays for performing block–matching with larger reference blocks or search areas has been addressed in [13], [11], [12], and [14].

3.2. Dynamic Programming

Let us consider the computation of the cost of matching pixel (i, j) of the left image to pixel (i, n) of the right image. The forward part of the dynamic programming algorithm computes the cost $pc(j, n)$ (Eq. 2). At stage j, all possible costs of transition from all points l (predecessors) to all points n within the area of possible matches must be taken into account producing a cost vector $\mathbf{C_j}$ containing the minimum cost of matching any point n. Therefore, the processing can be represented by a 2-D operator which operates on vector $\mathbf{C_{j-1}}$ producing vector $\mathbf{C_j}$ in a manner analogous to a matrix-vector multiplication. In our case, the multiplication corresponds to an addition and the addition corresponds to a min operation. It must be noted that in dynamic programming, a "matrix-vector multiplication" needs to be performed for every pixel of a row.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

The proposed algorithm has been tested with real image sequences and the estimated disparity fields were used for the calculation of intermediate views. The original left and right images, the estimated disparity fields and the calculated intermediate views are shown in Figures 2–5 and 6–9. The calculation of the intermediate views were performed by a simple interpolation algorithm. The intensity of each pixel of the intermediate view was estimated as the interpolation of the corresponding pixels from left and right image weighted by the relative distances. In the case of partial occlusion the intensity was extrapolated from one of the images.

The combined algorithm produces a very accurate disparity field but is computationally intensive since it requires both block matching and dynamic programming.

To speed up the algorithm, the following improvements can be considered:

1) Perform the above algorithm at a lower resolution image The estimated disparities will be used as prior constraints when the algorithm runs at full resolution. In the modified algorithm the disparity field is assumed smooth but also close to the disparity values obtained at lower resolution. The algorithm can be further sped up by not using the whole search region but only a small subset of it, since the values obtained at the previous

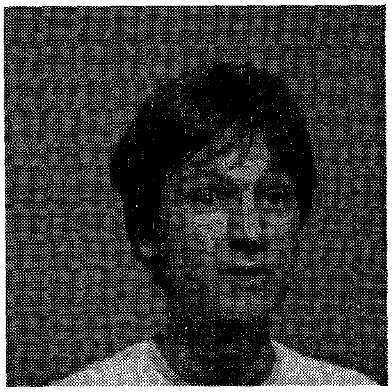

Figure 2. Left view

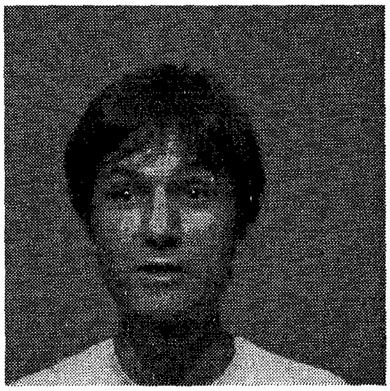

Figure 3. Right view

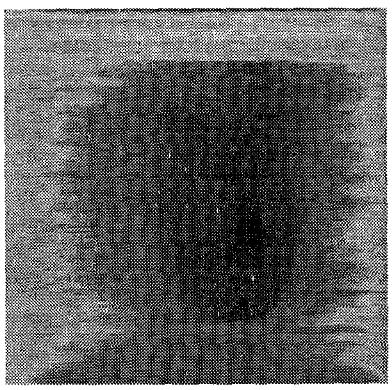

Figure 4. Disparity field

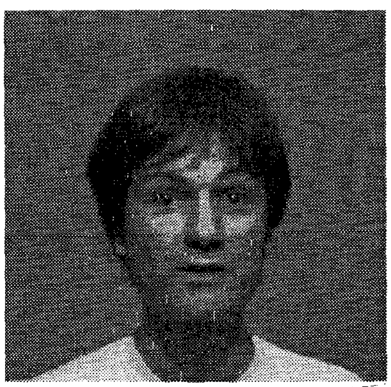

Figure 5. Intermediate view from a virtual camera positioned in the middle between the left and right cameras

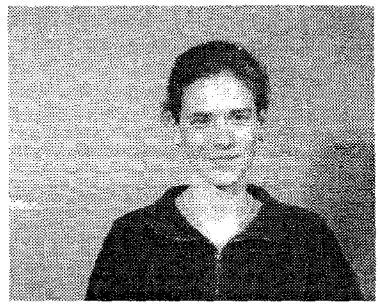

Figure 6. Left view

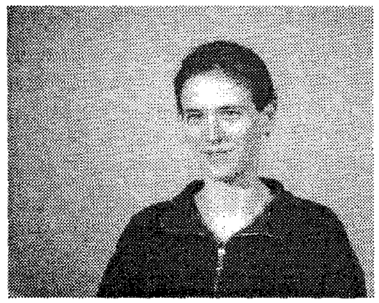

Figure 7. Right view

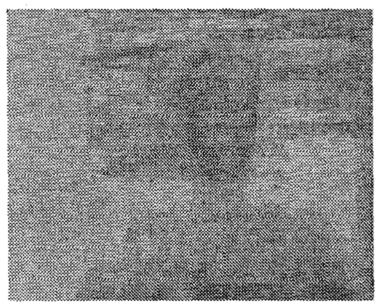

Figure 8. Disparity field

Figure 9. Intermediate view from a virtual camera positioned in the middle between the left and right cameras

step are assumed to be accurate within a certain tolerance.

2) In the case of head and shoulders sequences, the background of the images can be extracted and searching can be restricted on the pixels belonging to the region of interest.

5. CONCLUSIONS

In this paper, a novel disparity estimation algorithm has been presented that is a combination of a block matching part which gives an initial estimate of the disparity and a dynamic programming part which estimates a smooth disparity field. The hardware complexity of the algorithm has been studied. The algorithm has been tested using real image sequences that are representative for videoconferance applications. The estimated disparity fields were used for the calculation of intermediate views. The accuracy of the estimated disparities was proved by the excellent quality of the synthesized views.

6. ACKNOWLEDGMENTS

This work was partialy supported by the EC ACTS Project AC092.

REFERENCES

[1] S. T. Barnard and W. B. Thompson. Disparity analysis of images. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 2:333–340, 1980.

[2] H. D. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, 1982.

[3] B. Chupeau and P. Salmon. Synthesis of intermediate pictures for autostereoscopic multiview displays. *Proceedings of Workshop on HDTV'94*, 1994.

[4] E. Izquierdo and M. Ernst. Motion/disparity analysis and image synthesis for 3dtv. *Proceedings of Workshop on HDTV'94*, 1994.

[5] R. Skerjanc and J. Liu. A three camera approach for calculating disparity and synthesizing intermediate pictures. *Signal Processing: Image Communication*, 4:55–64, 1991.

[6] R. Bellman and S. Dreyfus. *Applied Dynamic Programming*. Princeton University Press, 1962.

[7] T. S. Huang. *Two-Dimensional Digital Signal Processing II*. Springer-Verlag, Berlin, 1981.

[8] J. R. Jain and A. K. Jain. Displacement measurment and its applications in interframe image coding. *IEEE Trans. on Communications*, 29:1799–1808, 1981.

[9] T. Komarek and P. Pirch. Array architectures for block matching algorithms. *IEEE Trans. on Circuits and Systems*, 36:1301–1308, 1989.

[10] L. De Vos and M. Stegherr. Parametrizable vlsi architectures for the full-search block-matching algorithm. *IEEE Trans. on Circuits and Systems*, 36:1309–1316, 1989.

[11] A. Artian and O. Colavin. A chip set core for image compression. *IEEE Trans. on Consumer Electronics.*, 36:395–402, 1990.

[12] C. Hsieh and T. Lin. Vlsi architecture for block-matching motion estimation algorithm. *IEEE Trans. on Circuits and Systems.*, 2:395–402, 1990.

[13] J. Vlontzos H. C. Karathanasis. Vlsi architectures for accurate motion and disparity estimation using full–search block–matching and edge preserving non–linear smoothing. *VLSI Signal Processing VI, Proc. of the 1993 IEEE Workshop on VLSI Processing, L.D.J. Eggermont et al Eds,*, 75–83, IEEE press,1993.

[14] M. T. Sun K. Yang and L. Wu. A family of vlsi designs for the motion compensation block-matching algorithm. *IEEE Trans. on Circuits and Systems.*, 36:1317–1325, 1989.