

A Simple Tree Structure for Dynamic Programming Based on Real Scene

Xing Pang, Yi Zhang, Lianfa Bai, Jing Han

Jiangsu Key Laboratory of Spectral Imaging and Intelligent Sense, Nanjing University of Science and Technology, Nanjing 210094, China

zhy441@sina.com, px117488@163.com

Abstract - A new stereo matching approach based on first order invariant moment is proposed. First order invariant moment is utilized in our algorithm to transform the left and right images from the input space to the feature space, and the improved a simple tree structure dynamic programming based on adaptive parameters is introduced to evaluate the disparity map to reduce the mismatching points on discontinuous and low texture regions. To testify the applicability of the proposed algorithm, it is applied to gray images captured in the real situation, and the results of experiments show that the proposed algorithm can effectively improve the quality of disparity map on interference areas.

Index Terms - binocular stereo matching, invariant moment, dynamic programming, a simple tree

1. Introduction

The robust and accurate computation of stereo depth is an important problem for the visual tasks. In the real scene, because of the influence of illumination and certain types of image distortion, the points of left image are unable to identify the correct matching points in right image.

Two non-parametric transforms which have been proposed for the stereo matching problem are the rank transform[1] and Census transform[2]. But the pix-based construction in image isn't considered.

So the paper proposes an algorithm used first order invariant moment transform due to its insensibility to illumination and amenability to geometry distortion. The process of matching based on image feature can be analyzed and processed more effectively.

Generally speaking, stereo matching can be categorized into two major classes: local methods and global methods. Local methods usually employ winner-take-all (WTA) to select the optimal disparity. But local methods provide poor disparity map if the cost is not aggregated properly, and too much noise also limits its application. While most global methods such as graph-cut(GC)[3], dynamic programming(DP) [4], and belief propagation (BP) [5] skip the aggregation step to maintain an unpolled cost entry. Global methods can provide smooth results but are also time and memory consuming. Dynamic programming optimization is also widely used to achieve real-time stereo matching. It optimizes within separate scanlines and can be implemented in parallel. Now many variations of DP are proposed to solve the well-known streaking effect which is caused by the lack of inter-scanline coherence in vertical direction and acquire a smoother disparity map.

There have been many variations of DP proposed for problem which caused by the by the lack of inter-scanline coherence and smoothness in the disparity map.

Gong and Yang [6] presents a two-pass DP. Cost volume recomputed during the horizontal DP process is incorporated in the following vertical DP optimization. Horizontal streaking artifact is reduced but vertical streaking brought in.

The two-pass DP algorithm is presented [6], which computes the cost volume in the horizontal DP process. Obviously, horizontal streaking artifact is reduced but vertical streaking brought in.

Veksler [7] proposes a method to connect all points on a tree. By performing dynamic optimization on the tree, a global optimization result is achieved. All edges both in horizontal and vertical direction are treated symmetrically during the optimization step, through which streaking artifact is reduced. However, due to pix-based construction and edge losing during tree structuring, the performance is somehow degraded.

The tree structure used in dynamic optimization has been proposed in [7], where the horizontal and vertical directions are both considered, so that streaking artifact can be reduced. The disadvantage of this algorithm is the edge loosed.

The paper proposes a simple tree structure dynamic programming based on adaptive parameters can reduces this artifact and reduces error matches in the repeat texture areas of complex background. If a tree has nodes and the number of possible disparity values is h , then the straightforward dynamic programming takes $O(nh^2)$ time. However for a certain restricted but still quite useful type of energy functions, the running time can be reduced to $O(nh)$, which is the complexity of our method.

2. Invariant Moment

The traditional stereo matching algorithms such as SAD, SSD [8,9] estimate the disparity only based on local intensity between stereo images so that the results may be susceptible to difference and noise from illumination.

From the point of feature space, except the reason of experiment environment, the information of the gray space is relative narrow, which only represents the intensity of the pixel but ignores the position information of other pixels relating to the central one. So the invariant moment space is presented for stereo matching. The paper proposes a method which improves pixel uniqueness information to be satisfied for a correct match.

In this section, the paper firstly describes the equations of first moment invariant, then the paper shows how to use it to reduce influence of illumination and geometry distortion.

I means the left or right image, for every pixel (x,y) and the window size w is started from 3*3, for calculating m_{00}, m_{01}, m_{10} , The descriptors are functions of the moment m_{pq} defined by

$$m_{pq} = \sum_{(x,y) \in w} x^p \cdot y^q \cdot f(x, y) \quad p, q=0,1,2... \quad (1)$$

The two-dimensional (p+q)th order moments of a density distribution function $f(x,y)$ are defined the gray value of the pixels in the window of left and right images in (1).

Put m_{pq} (1) into formula for calculating μ_{pq} , and compute μ_{20} and μ_{02} .

The central moments μ_{pq} that have the property of translation invariance are given by

$$\mu_{pq} = \sum_{(x,y) \in w} (x - \bar{x}) \cdot (y - \bar{y}) \cdot f(x, y) \quad p, q=0,1,2... \quad (2)$$

$$\text{where } (\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}})$$

Put μ_{pq} (2) into formula for calculating n_{pq} , computing n_{20} and n_{02} . n_{pq} is defined as:

$$n_{pq} = \mu_{pq} / \mu_{pq}^r \quad (3)$$

Where $r=(p+q)/2+1$

Φ_1 represent the center pixel in window which sizes is 3*3,

Φ_1 is defined as :

$$\phi_1 = n_{02} + n_{20} \quad (4)$$

The invariant moment calculated by (1)-(4), not only have translation, scaling, rotation invariance, but also response character of the image.

The moment invariant transformation process involves passing the window over the image, and at each point, computing the value of Φ_1 and replacing the value of central pixel in the window.

3. Dynamic Programming on a Tree

In the process of matching, there are two reasons for mismatching: texture repeats in small scale, where image's feature cannot keep uniqueness in disparity range, and geometry distortion or occlusion, where the points in two views are not accordance.

First order invariant moment can reduce error matches, but analyzing the feature of the mismatching points, they always appear in disparity jump of low texture regions. So we consider utilizing the programming as the disparity calculating method.

Let it be the set of all pixels in the transformed image pair and D denotes the set of allowed disparity labels. We formulate the stereo matching task, finding a disparity solution D that maps each pixel $p \in I$ to its own disparity $d_p \in D$. This paper seeks for a disparity map D by minimizing a predefined energy functional $E(D)$, which is typically in the form of

$$E(D) = E_{data}(D) + E_{smooth}(D) \quad (5)$$

Here, the data term E_{data} assesses the agreement of the disparity solution with the input images by computing a match measurement. In addition, the smoothness term E_{smooth} imposes a penalty on spatially neighbouring pixels carrying different disparity. Obviously E_{data} holds for the difference between left and right images and E_{smooth} controls the smooth of the disparity map. When E_{smooth} and E_{smooth} achieve their minimize value in the disparity range, which means the value of disparity map is relative correct and stay similar with the pixel around. The programming problem can be rewritten in Eq.(6).

$$E(D)_{\min} = E_{data}(D) + E_{smooth}(D) \quad (6)$$

$$D(x, y) = [d_{\min}, d_{\max}]$$

In our approach, we benefit from the simple structure of our tree to significantly improve the computational speed.

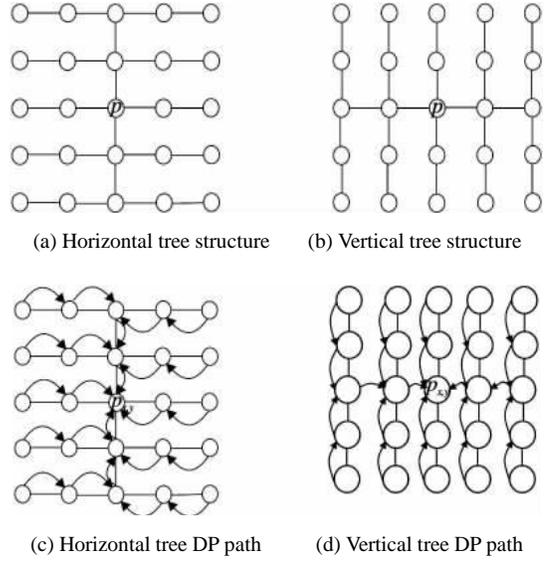


Fig.1 Structures for tree DP

The simple tree algorithm transforms global programming to local problem to optimize the disparity of every pixel in its neighboring area. Construct the tree structure around central point which is root point P. Optimizing process starts from the leaf point at the edge of structure to the root point, and calculates the optimizing value of cost function at each possible disparity of every pixel in the structure and finds out

the minimize value of the cost function at the root point, whose disparity is regarded as the disparity of the root point.

At first, the tree structure and the iteration direction should be defined. Divide the neighboring pixels of the central point P as Fig.1(a)(b). Fig.1(c) shows the scheme of the horizontal tree programming. We compute the value of cost function along with the iteration direction from outside to inside at each horizontal scanline.

Secondly construct the cost function of central point P.

$$E(p_{x,y},d) = m(p_{x,y},d) + \sum_{q \in \text{sub}(p_{x,y})} \min_{i \in D} (\lambda s(d,i) + E(q,i)) \quad (7)$$

$$m(p_{x,y},d) = \sum_{x,y \in W} |L_tex(P_{x,y+d(x,y)}) - R_tex(P_{x,y})| \quad (8)$$

$$s(d,i) = |d - i| \quad (9)$$

Where the $m(p_{x,y},d)$ denotes the difference between P in left image and corresponding point in right image at disparity d, which equals E_{data} in Eq.(5). The latter one,

$\sum_{q \in \text{sub}(p_{x,y})} \min_{i \in D} (\lambda s(d,i) + E(q,i))$, is same with $E_{\text{smooth}}(D)$. The

parameter λ denotes the weight of penalty in the smoothness term. The depth map is smooth with the λ relative large. $\text{sub}(p_{x,y})$ is the set of sub-points of $P, P_{x-1,y}, P_{x+1,y}, P_{x,y-1}, P_{x,y+1}$,

$$d_{p_{x,y}} = \arg \min_{i \in [d_{\min}, d_{\max}]} E(p_{x,y},d) \quad (10)$$

In the experiment, to achieve a good disparity map in different regions, we need different λ . For the region with complex texture, λ is supposed to be relative large, oppositely λ should be small. So we set the value of λ relied on the variance of a window. The reason is simple: the variance can reflects and be along with the value of pixels changing in a window. So the λ can be written as εv_p , where the ε is to have variance v_p to a reasonable level.

$$E(p_{x,y},d) = m(p_{x,y},d) + \sum_{q \in \text{sub}(p_{x,y})} \min_{i \in D} (\varepsilon v_p s(d,i) + E(q,i)) \quad (11)$$

Simply the Eq.(11), the programming process can be divided to horizontal and vertical programming. For horizontal programming shown in Fig.1(c), we compute the values of cost function of points on each scanline and get the optimized values $C(J_{x,y},d)$ of central points $J_{x,y}$ at the disparity d. The whole process is shown in Eq.(9)-(11).

$$C(J_{x,y},d) = m(J_{x,y},d) + \sum_{q \in \text{Hsub}(J_{x,y})} \min_{i \in D} (\varepsilon v_p s(d,i) + C(q,i)) \quad (12)$$

$$C(J_{x,y},d) = F(J_{x,y},d) + B(J_{x,y},d) - m(J_{x,y},d) \quad (13)$$

$$F(J_{x,y},d) = m(J_{x,y},d) + \min_{i \in [d_{\min}, d_{\max}]} (\varepsilon v_p s(d,i) + F(J_{x-1,y},i)) \quad (14)$$

$$B(J_{x,y},d) = m(J_{x,y},d) + \min_{i \in [d_{\min}, d_{\max}]} (\varepsilon v_p s(d,i) + B(J_{x+1,y},i)) \quad (15)$$

Where the $\text{Hsub}(J_{x,y})$ is the set of horizontal sub-points of $J_{x,y}$ and $F(J_{x,y},d)$, $B(J_{x,y},d)$ are the forward and backward points respectively. After finishing the whole horizontal programming, then repeat the process before and calculate the central point's cost value along the vertical direction in central column as shown in (16)

$$H(p_{x,y},d) = C(p_{x,y},d) + \sum_{q \in \text{Vsub}(p_{x,y})} \min_{i \in D} (\varepsilon v_p s(d,i) + H(q,i)) \quad (16)$$

Where the $\text{Vsub}(p_{x,y})$ is the set of vertical sub-points $P_{x,y-1}, P_{x,y+1}$ of $P_{x,y}$

The updated matching scores then represent the input to the calculation of Horizontal Trees. We determine the values of the array H using the modified data costs m' . The final disparity for each pixel p is then selected by $dp = \text{argmin}_{d \in D} H[p,d]$.

$$m'(p_{x,y},d) = m(p_{x,y},d) + \xi (H(p_{x,y},d) - \min_{i \in D} H(p_{x,y},i)) \quad (17)$$

After the horizontal programming, the vertical programming should then be taken. Before that, the values of $m(p_{x,y},d)$ must be refreshed as Eq.(17), which transmits the results of horizontal tree programming to vertical tree. The parameter shows ξ how important the horizontal values are. After all of programming, utilize the Eq.(11) for the real disparity of the root point P.

4. Experimental Results

Results of our algorithm on these stereo images are shown in Fig.2. The stereo matching pairs we made are used for demonstrating the effectiveness of our presented algorithm. All disparity maps have been generated using constant parameters. Our algorithm works well with the same parameters for the image pair we tried.

Figure.2(a)(b) are initial image pair in our experiment. The SAD algorithm is applied based on initial image pair is Fig. 2(c).The disparity map shows that massive noise caused by distinguishing illumination environment or lighting disproportion on plaster model is obvious. But the Fig.2(d) applying SAD algorithm on transformed image pair based on moment invariant is better than the Fig.2(c). Comparing the disparity map Fig.2(e) which applying dynamic programming on tree on the initial image pair with Fig.2(d), we can see the same depth area more smooth. Fig.2(f) shows that our algorithm reduces the error matching rate and effectively proves the advantages of dynamic programming on a tree.



(a) left initial image



(b) right initial image



(c) SAD based on initial image pair



(d) SAD based on transformed image pair



(e) dynamic programming on a tree based on initial image pair



(f) dynamic programming on a tree based on transformed image pair

Fig.2 Disparity map of 4 algorithms

The results show that our algorithm produces relatively accurate disparity map and could properly alleviate the influence of the illumination. Dynamic programming on a tree based moment invariant image pairs are effectively reduced error matches in repeat texture of complicated background.

5. Conclusion

The paper proposes first order moment invariant image pair for stereo matching that not only preserves the image gray level information, but also includes the relative position between the pixel information.

On the calculation of disparity map, this paper puts forward the dynamic programming on a simple tree method based on adaptive parameters, considering the dissimilarity in image pair and the smoothness in disparity map. The paper proposes an adaptive parameter method based on different texture feature area, as a result reduces ambiguous matches caused by jump error between pixels and the use of simple tree method to improve the optimization process speed.

Experiments prove that the appearance of smooth term effectively restrain low texture area, illumination and geometry distortion.

Acknowledgment

This work was supported by the Natural Science Foundations of China (61373061 and 61231014).

References

- [1] J Banks and M Bannamoun, "Reliability analysis of the rank transform for stereo matching," *Imaging & Cognitive Compute*, vol. 31, no. 2, Dept 2001.
- [2] B Froba and A Ernst, "Face detection with the modified census transform," *Automatic Face and Gesture Recognition*, pp: 91-96. May 2004.
- [3] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222-1239, nov 2001.
- [4] Yuichi Ohta and Takeo Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 2, pp. 139-154, march 1985.
- [5] Sun Jian, Zheng Nan-Ning and Shum Heung-Yeung, "Stereo matching using belief propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp.787-800, 2003
- [6] Gong Minglun, Yang Y H, "Near Real-Time Reliable Stereo Matching Using Programmable Graphics Hardware," Proc of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Diego, USA, pp: 924-931, 2005.
- [7] Olga Veksler, "Stereo correspondence by dynamic programming on a tree," *IEEE International Conference on Computer Vision and Pattern Recognition*, San Diego USA, pp: 384-390, 2005.
- [8] V. Kolmogorov. "Convergent tree-reweighted message passing for energy minimization," In Tenth International Workshop on Artificial Intelligence and Statistics, January 2005.
- [9] E. Trucco, V. Roberto, S. Tinonin and M. Corbato, "SSD disparity estimation for dynamic stereo," presented at The British Machine Vision Conference, Edinburgh, England, 1996.