International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:6, 2008

Feature Based Dense Stereo Matching using Dynamic Programming and Color

Hajar Sadeghi, Payman Moallem, and S. Amirhassn Monadjemi

Abstract**—**This paper presents a new feature based dense stereo matching algorithm to obtain the dense disparity map via dynamic programming. After extraction of some proper features, we use some matching constraints such as epipolar line, disparity limit, ordering and limit of directional derivative of disparity as well. Also, a coarse-to-fine multiresolution strategy is used to decrease the search space and therefore increase the accuracy and processing speed. The proposed method links the detected feature points into the chains and compares some of the feature points from different chains, to increase the matching speed. We also employ color stereo matching to increase the accuracy of the algorithm. Then after feature matching, we use the dynamic programming to obtain the dense disparity map. It differs from the classical DP methods in the stereo vision, since it employs sparse disparity map obtained from the feature based matching stage. The DP is also performed further on a scan line, between any matched two feature points on that scan line. Thus our algorithm is truly an optimization method. Our algorithm offers a good trade off in terms of accuracy and computational efficiency. Regarding the results of our experiments, the proposed algorithm increases the accuracy from 20 to 70%, and reduces the running time of the algorithm almost 70%.

Keywords**—**Chain Correspondence, Color Stereo Matching, Dynamic Programming, Epipolar Line, Stereo Vision.

I. Introduction

DENSE disparity estimation from stereo images has traditionally been, and continues to be, one of the most active research topics in the field of computer vision [1]–[4]. Correspondence is an essential problem in dense stereo matching. The correspondence is to determine which item in the left image corresponds to which item in the right image. In dense disparity computation, correspondence should be solved for each point in the stereo images.

Many applications require dense measurements, and measurement interpolation is a difficult problem itself. The motivation behind the dense stereo methods is that all, or almost all, image pixels can be matched. To compute reliable dense depth maps, a stereo algorithm must preserve depth discontinuities and avoid gross errors.

Traditional dense matching algorithms fall into two categories: the first approach is based on a local method; the second one is based on a global optimization. Employed in this study, the former compares intensity similarity of pixels within a window between a pair of images to decide whether the centre points of the windows are a pair of corresponding points.

Two main problems in dense stereo are lack of texture and occlusion in the image. We propose a color dynamic programming based algorithm that handles these two problems. Experiment results show that the algorithm compares favorably with other state of the art stereo algorithms.

The proposed algorithm has got three stages: feature extraction, feature matching, and dynamic programming [5]–[9]. In the feature matching stage, we use edge based stereo techniques [10]–[13]. In this paper, we use the concept of the matching feature chains to decrease both the computing time and matching error. In order to make the stereo matching algorithm run efficiently, only some of the features from each chain are tested. We use the MSE (Mean Square Error) in 3×3 windows as the cost function.

In this study, correspondence for a feature point in the first image is obtained by searching a predefined region of the second image, based on the epipolar line and disparity range constraints. Traditionally, the hierarchical multiresolution techniques using Haar wavelet transform are used to decrease the search space and therefore increase the processing speed [14]. In this method, information obtained at a coarse scale is used to guide and limit the search for the matching of finer scale primitives or feature points. Therefore, it increases the matching speed.

Color information could improve the identification of binocular disparities to recover the original three-dimensional scene from two-dimensional images. The color makes the matching less sensitive to occlusion considering the fact that occlusion most often causes color discontinuities. So, in this study we use color stereo images too.

Finally, a dense disparity map is obtained from implementing our algorithm using dynamic programming. The methodology differs considerably from the existing dynamic programming formulation of stereo e.g. [5], [6], and [15], in the way it is performed on points between two subsequent edge points which disparity is obtained for in the previous stages.

Hajar Sadeghi is now M.S. student at the Department of Computer Engineering, University of Isfahan, Isfahan, Iran. (E-mail: hsadeghi@eng.ui.ac.ir).

Payman Moallem is now an assistant professor at the Department of Electrical Engineering, University of Isfahan, Isfahan, Iran. (E-mail: p_moalloem@eng.ui.ac.ir).

S.Amihhassan Monadjemi is now an assistant professor at the Department of Computer Engineering, University of Isfahan, Isfahan, Iran. (E-mail: monadjemi@eng.ui.ac.ir).

This paper is arranged as follow: we begin in Section II by explaining the stereo matching algorithms, and then in Sections III and IV, we show the effect of the color and chaining on the matching algorithms. In Section V, we illustrate the used dynamic programming algorithm. The main algorithm is included in Section VI. In Section VII we present our experimental results. Conclusions are in Section VIII.

II. Stereo Matching

A. The Stereo Matching Algorithms

Most algorithms which used to solve the matching problem can be categorized as either feature based techniques, area based techniques [16]–[18], or pixel based techniques. Feature based stereo is defined as algorithms which perform stereo matching with high level parameterization called image features, these algorithms can be classified by the type of features used in the matching process. In the feature extraction stage, specific feature points such as edges, corners, centroids, and textured areas would be extracted from the left and right images.

In the area based techniques usually a dense disparity map would be produced. According to [1], stereo algorithms that generate dense depth measurements can be roughly divided into two classes, namely global and local algorithms. Global algorithms [19], rely on the iterative schemes that carry out disparity assignments on the basis of the minimization of a global cost function. These algorithms yield accurate and dense disparity measurements but exhibit a very high computational cost that renders them not suitable for real time applications. Local algorithms [20]-[23], also referred to as area based algorithms, calculate the disparity at each pixel on the basis of the photometric properties of the neighboring pixels. In these techniques, the elements to be matched are image windows of fixed or variable sizes, and similarity criterion can be the correlation between the windows in two images. These algorithms can run fast enough to be deployed in many real time applications. Area based stereo matching, compared to the feature based one, delivers more accurate results.

In pixel based techniques, each pixel in each epipolar line in the left image would be compared to every pixel on the same epipolar line in right image and the pixel with minimum matching cost will be picked. This however leaves too much ambiguity. In these methods, a dense disparity map can be obtained.

The correspondence search in stereo images is commonly reduced to important features as computing time is still an important factor in stereo vision. Unfortunately, feature based stereo or edged based stereo, respectively, produce only sparse disparity maps. For a successful reconstruction of the complex surfaces it is, however, essential to compute the dense disparity maps defined for every pixel in the whole image.

B. The Matching Constraints

Various stereo matching constraints [24]–[26], [2] are generated based on the underlying physical principles of world imaging and stereopsis. Some of more common constrains are explained below:

- *Epipolar constraint*: Corresponding points must lie on corresponding epipolar lines.
- *Continuity constraint*: disparity tends to vary slowly across a surface, prefer disparity similar to neighbors.
- *Uniqueness constraint:* a point in one image should have at most one corresponding point in the other image.
- *Ordering constraint:* order of the features along epipolar lines is the same.
- *Occlusion constraint:* a discontinuity in one eye corresponds to an occlusion in the other eye and vice versa.
- *Disparity limit constraint:* regarding the maximum and minimum of depth and geometry of a stereo system, the maximum disparity range can be estimated.
- *Limit of the directional derivative of disparity:* maximum of directional derivative of disparity is limited that the absolute value of the directional derivative of disparity is practically less than 1.2 [25].

C. The Directional Derivate of Disparity

In stereo systems, the directional derivative [24]–[26] possesses some restrictions that it can be used to narrow down the search space. Fig. 1 shows a basic stereo system where the cameras optical axes are parallel and perpendicular to the baseline.

Given two points P^1 and P^2 in the 3D scene, there are two different definitions for the directional derivative of disparity which were shown in (1) and (2) bellow respectively:

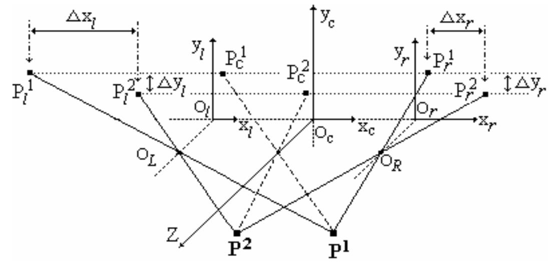

Fig. 1 The 3D camera geometry

$$\delta d = (d_2 - d_1) / \| p_l^2 - p_l^1 \| \tag{1}$$

In (1), ||.|| denotes vector norm. The second definition uses cyclopean separation that is an average distance between (p_l^1, p_l^2) and (p_r^1, p_r^2). Suppose a virtual camera in the middle of the cameras L and R.

$$d_1 = x_l^1 - x_r^1, d_2 = x_l^2 - x_r^2$$
$$p_i^c = (p_i^l + p_i^r)/2 \tag{2}$$
$$\delta d_c = (d_2 - d_1)/\|p_c^2 - p_c^1\|$$

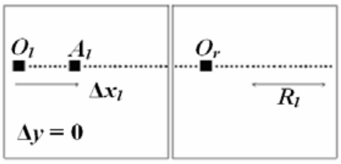

Fig. 2 The search region R$_l$ for the correspondence of A$_l$ in the right image, and Δy=0

TABLE I
THE RELATIONSHIP BETWEEN ΔX$_L$ AND THE SEARCH REGION, WHEN $|\delta d| < 1.2$, ΔY=0 AND ΔX$_L$ IS BETWEEN 1 TO 10 PIXELS.

Δx$_l$	Min(Δx$_l$)	Max(Δx$_l$)
1	0	4
2	0	8
3	0	12
4	1	16
5	1	20
6	1	24
7	1	28
8	2	32
9	2	36
10	2	40

If we consider the proposed equations and the constraints on the directional derivative, and also Fig. 2, we can reduce the search space drastically [26](See Table 1).

III. COLOR STEREO VISION

One of the aspects of an image that has been largely neglected in the stereo algorithms is the color information [2], [27]–[29]. Current investigations have shown that the quality of stereo matching results can be improved using color information. There are several motivations for using chromatic information. Firstly, chromatic information is easily and precisely obtained using a 3-chip CCD camera. Secondly, new psychophysical evidence indicates that color information is widely used in human stereopsis. Thirdly, it is obvious that red pixels can not match the blue pixels even though their intensities are equal or close. Thus, color information can potentially improve the performance matching algorithm.

Amongst several known color spaces, e.g. RGB, HIS, or CIE-Lab, the RGB is chosen in this study. Drumheller and Poggio [2] presented one of the first stereo approaches using color. In color images, we use MSE, as the similarity measure as defined in (3):

$$MSE_{color}(x, y, d) = \frac{1}{n^2}\sum_{i=-k}^{k}\sum_{j=-k}^{k} dist_c(C_R(x+i, y+j), C_L(x+i+d, y+j)) \tag{3}$$

$$dist_c(c^1, c^2) = (R^1 - R^2)^2 + (G^1 - G^2)^2 + (B^1 - B^2)^2 \tag{4}$$

In (3), d is the disparity, and in (4) c_1, c_2 are two points from left and right color images and are defined as below:

$$c^1 = (R^1, G^1, B^1) \ \ , \ \ c^2 = (R^2, G^2, B^2) \tag{5}$$

The MSE is calculated in an n×n window. The left color image C_L and the right color image C_R in RGB color space are expressed as:

$$C_L(x, y) = (R_L(x, y), G_L(x, y), B_L(x, y)) \tag{6}$$

IV. FEATURE CHAINS

In this paper, we reduce the search space for the successive connected features from a predefined disparity to a much smaller range. Therefore, the algorithm can run much faster than former corresponding algorithms [10]. To make the algorithm run more efficiently, we just try the first two feature points of each chain using the MSE similarity measure. If both of the two features have high correlation scores, the tested pair of chains from each image is defined as chains correspondence. This strategy is shown in Fig. 3.

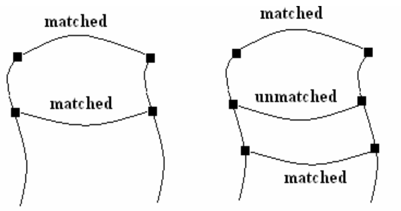

Fig. 3 Feature chains matching strategy

The experimental results show that average 92% of the feature chains have the length less than or equal to 5. This indicates that about 40% of the feature point's correspondences are evaluated.

V. DYNAMIC PROGRAMMING

One class of global correspondence methods are those based on dynamic programming (DP) as it is named by Richard bellman in 1953. Dynamic programming is an effective strategy to compute correspondences for pixels. This method is finding the minimum cost path going monotonically down and right from the top-left corner of the graph to its bottom-right corner. So, the technique of dynamic programming can be used to find the optimal match sequence between the start and end points.

This strategy has a cost matrix with the nodes representing the weight of matching a pixel in the left image with a pixel in the right image. The cost of matching pixel *x* in the left image and pixel *y* in the right image can be computed based on the costs of matching all pixels in the left of these two pixels. If one assumes the ordering constraint, the optimal path computed to match the pixels in left and right images will result in the best set of matches for the pixels in left and right images. Thus, DP yields the optimal path through grid. This is the best set of matches that satisfy the ordering constraint.

Fig. 4(a) demonstrates the search grid for two scan lines with 10 pixels and a maximum disparity of three pixels that is shown with d$_{Max}$. Each (x, y) cell in this grid means a possible match between pixel x in the left image and pixel y in the right image. Our algorithm looks for the best possible path extending from the first column to the last row. In this figure, the matched pixels are shown by the "M". As in the Fig. 4(b), the three moves are allowed between each pixel pairs. The circles represent nodes of the grid.

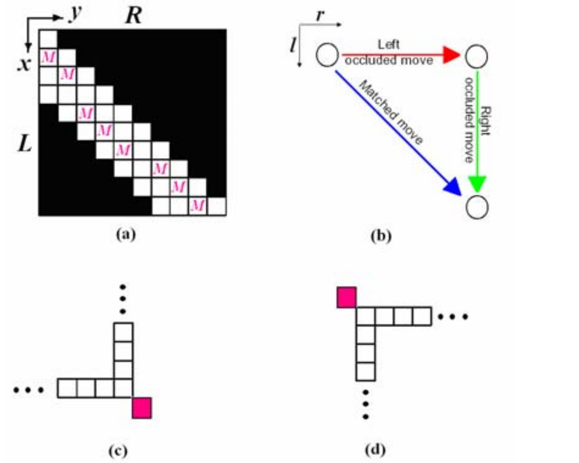

Fig. 4 (a) The search grid and a match sequence ("M" cells),(b) Three allowed moves between two pixels in the grid (c) The immediate preceding matches, (d) The immediate following matches

The immediate preceding matches and immediate following matches of any matches (x_i, y_i), are shown in Figs. 4(c) and 4(d), respectively. Regarding Fig. 4(a), each match has d_{Max} + 1 possible candidate as its immediate preceding matches and d_{Max} + 1 possible candidate as its following matches.

For each white cell of Fig. 4(a), we should record C(x, y) as the cost of the best match sequence so far, and P(x, y) as pointer to the immediate preceding match in that match sequence. In each white cell in Fig. 4(a), we put MSE_{color} of those pixels of left and right images. We call this matrix M, and then we normalize this matrix.

Next we use (7) to compute the cost of the best path to each cell:

$$C(x,y) = d(x,y) + m \qquad where$$

$$m = \left\{ \begin{array}{l} C(x-1,y-1), \\ C(x-2,y-1)+k_{occ}, C(x-3,y-1)+k_{occ},... \; until \;\; x=y \\ C(x-1,y-2)+k_{occ},..., C(\mathrm{d}_{\mathrm{Max}}, x-\mathrm{d}_{\mathrm{Max}}-1)+k_{occ} \end{array} \right\} \tag{7}$$

In equation (7), K_{occ} is the constant occlusion penalty and d(x, y) is the MSE_{color} of pixel x and y. Once the C matrix is filled up, the lowest cost cell from the end row of the matrix M is selected as the final match. Then, starting at this cell, the matrix P is traced to find the optimal match sequence. After trying different values for K_{occ}, we choose K_{occ}=0.2 for our proposed algorithm and K_{occ}=0.05 for the dynamic programming algorithm, respectively.

Dynamic programming matches lines to lines. They can also use the matches found for previous pixels in the same scan line in the computation for the subsequent pixels [6], [15]. We extend this approach to find the matches applying DP between each two edges that their disparity is calculated in the previous stage.

VI. Proposed Algorithm

In this section, we present a novel DP-based color chain stereo matching algorithm. Our algorithm consists of three stages: feature extraction, feature matching, and dynamic programming.

A. Feature extraction

This stage contains identifying non-horizontal thinned edge chains using a 3×3 Sobel filter in the horizontal direction which is applied on both left and right images. The thinned edge points are classified into two groups: positive and negative, depending on the intensity difference between the two sides of the feature points in the horizontal direction in any color channel.

A non-horizontal thinned positive edge in a left image is localized to a pixel that the filter response has to exceed a positive threshold ρ_0^+ and has to obtain a local maximum in the horizontal direction, therefore:

$$\begin{array}{ll} \rho_l(x,y) > \rho_0^+ & \text{Threshold} \\ \left. \begin{array}{l} \rho_l(x,y) > \rho_l(x-1,y) \\ \rho_l(x,y) > \rho_l(x+1,y) \end{array} \right\} & \text{Local Maximum} \end{array} \tag{8}$$

Assume that ρ^+_0 is the mean of positive values of the filter response.

The extraction of non-horizontal negative thinned edge points is similar to the positive one. Now we should attempt to extract feature chains with the length more than 3. Each two sequence feature points in the same chain should be in the sequence scan lines and the disparity in horizontal direction should be less than 2. The chains that their length is less than 3 are ignored.

B. Feature Matching

Once the correspondence between the two images is known the depth information of the objects in the scene can be obtained easily. Meanwhile, the matching feature points should be the same in the positivity or negativity. Thus, our feature matching stage includes two phases itself:

1) Phase I:

a) Do systematic scan from the left to right

b) If the current point is not a feature point, go to (a).

c) If disparity was already computed for the current feature point, store its x value as x_{0l} and go to (a).

d) If the current point is not on feature chain, go to (a), else call the x value of the current feature point as x_{cl}. If there is not any x_{0l} then go to (e), else compute $\Delta x_l = x_{cl} - x_{0l}$ and then compute the search space in the right image recording Table I and then go to (f).

e) Compute the search space based on the disparity range.

f) Find the correspondence point of the current feature point in the right image. If there is not any correspondence point, go to (a) else go to phase II.

2) Phase II: If L(x_{1l}, y) and R(x_{1r}, y) are features correspondences:

a) Choose the next feature points L(x_{2l}, y+1) and R(x_{2r}, y+1) from the same feature chains in the left and right chains separately. Test the similarity between them, if the similarity score is higher than the threshold, record the feature chains correspondence and delete two correspondence chains from the left and right images. Note that the corresponding feature chains should be recorded with the same length and then go to (a) in phase I, else go to (b) in phase II.

b) Use the same method to test the third feature points L(x_{3l}, y+2) and R(x_{3r}, y+2) from the feature chains. If they are features correspondence, record the feature chains as feature chain correspondence and delete two correspondence chains from the left and right images and then go to (a).

The output of this stage is disparity map where each pixel in the map represents the disparity of the matching pixels of two images; otherwise their depth in the scene.

C. Dynamic Programming

In this stage, we use the output of the previous stage and using dynamic programming to calculate the disparity of points between each two subsequent edge in any scan line. As a result, we would have a dense disparity map.

The used cost function is MSE in (3). Also, we use median filter for smoothing the obtained disparity map. This filter computes vertical median with the 6 surround points for each pixel (3 pixels in up and 3 pixels in down) in disparity map. So, the disparity map becomes smooth, and the most singular errors are deleted. This effect is shown in Fig. 5 on a color stereo image, Barn1, and the comparison of color chain stereo dynamic programming with and without median filter is shown in Table II. Since then, in all tables in this paper, the percentage of the number of features which are matched and the error matches in the left image in matching stage are shown in the matched (Match%), and error (Error%) columns respectively.

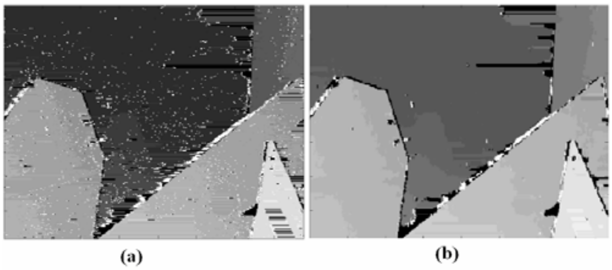

Fig. 5 (a) Color chain stereo dynamic programming without median filter, (b) Color chain stereo dynamic programming with median filter

TABLE II
THE COMPARISON OF COLOR CHAIN STEREO DYNAMIC PROGRAMMING WITH AND WITHOUT MEDIAN FILTER.

	CCSDP without Median filter		CCSDP with Median filter	
Scene	Match%	Error%	Match%	Error%
Ball	97.2%	2.7%	98.1%	1.8%
Barn1	95.3%	4.6%	95.9%	4.0%
Barn2	91.9%	8.0%	93.5%	6.4%
Poster	93.8%	6.1%	94.4%	5.6%
Venus	92.2%	7.7%	94.3%	5.6%

According to this table, using from median filter makes accurate our algorithm from 10 to 40%, however it consumes more time alone about 0.26% that can be ignored. So, since then we always apply median filter to smooth our results.

VII. IMPLEMENTATION RESULTS

Some experiments are arranged to evaluate the new dense matching algorithm. We presented the reduction of the search space in an edge based stereo correspondence, using the context of the maximum disparity gradient and the expansion of the accuracy, using color stereo images, and we obtain a dense disparity map using a dynamic programming algorithm, too. Our algorithm is evaluated on five different stereo scenes from MIDDLEBURY database [30] that are Ball, Barn1, Barn2, Poster, and Venus. All the scenes used are colored, 380×432 and their maximum disparity is less than 20 pixels, therefore we consider d_{Max}=20 for our computations. Their disparity ranges are shown in Table III.

TABLE III
DISPARITY RANGES OF THE TESTED SCENES.

Disparity	Ball	Barn1	Barn2	Poster	Venus
Minimum	3	4	3	3	3
Maximum	19	16	17	20	20

In the proposed algorithm, for the first point of any chain, the hierarchical multiresolution matching strategy is used, and for other points of that chain, we use MSE_{color} with window size of 3×3 as the similarity measure. The hierarchical multiresolution technique uses Haar Wavelet in three levels with the window size of 5×5, 3×3, and 3×3 for the coarse, medium, and fine level respectively while the threshold is 500.

The window size in the correlation based methods is very essential. As the window size decreases, discriminatory power of the window based criterion goes down too, and some local minimums in MSE could have been found in the search region. In contrary, increasing the window size causes the performance to be degraded due to the occlusion of regions and the smoothing of disparity values across the depth boundary. It also increases the speed of the algorithm.

Considering the importance of the first point of chains, we apply a left-right consistency checking technique to reduce the invalid matching, since the matching result of these points is used for matching of the next points on the chain. Just two or at most three points in each chain are matched [26].

Table IV shows comparison of results on color and gray level stereo images. Here, we use CCS for the color chain stereo, and GLCS for the gray level chain stereo. This table indicates that the proposed feature based stereo matching algorithm which uses the color stereo images, can increase the accuracy between 20 to 60% (e.g. (4.4-1.7)/4.4×100=61.3%) while the matching time goes up around 10%. Therefore with using color rather than gray level stereo images, reduces the error considerably in cost of a rather slight increase in the computing time. This claims that the proposed algorithm is advisable. So, we use color stereo images in our fundamental algorithm, Color Chain Stereo Dynamic Programming or CCSDP.

TABLE IV
THE COMPARISON OF THE GLCS AND CCS ALGORITHMS.

	GLCS		CCS	
Scene	Match%	Error%	Match%	Error%
Ball	96.5%	3.4%	97.1%	2.8%
Barn1	96.6%	3.3%	98.4%	1.5%
Barn2	95.5%	4.4%	98.2%	1.7%
Poster	95.0%	4.9%	97.6%	2.3%
Venus	94.2%	5.7%	96.5%	3.4%

As mentioned before, we used a new dynamic programming algorithm to obtain the dense disparity map. In this section, we apply this algorithm. In Table V, we compare the results of our algorithm (CCSDP) with the case that only using dynamic programming is used on color stereo image.

TABLE V
THE COMPARISON OF THE CCSDP AND DP ALGORITHMS.

	CCSDP Algorithm			DP Algorithm	
Scene	Match%	Error%	Reduced Time%	Match%	Error%
Ball	98.4%	1.5%	72.8%	94.5%	5.4%
Barn1	95.9%	4.0%	73.9%	94.9%	5.0%
Barn2	93.5%	6.4%	73.8%	83.6%	16.3%
Poster	94.4%	5.6%	71.6%	80.2%	19.7%
Venus	94.3%	5.6%	71.2%	89.9%	10.0%

The reduced time column in that table shows the reduction percentage of the utilized time in CCSDP algorithm with respect to the DP algorithm.

These results show that the composition of dynamic programming with our restricted search feature based dense stereo algorithm, will improve the outcome of algorithm from 20 to70%, and also the running time decrease about 70% which makes the proposed algorithm applicable and feasible in real time applications.

Table VI, represents the absolute average number of features which are matched and the error matches in both DP and CCSDP algorithms. This table shows that CCSDP algorithm in contrast to DP algorithm has matched more pixels correctly and has got less error matched pixels.

TABLE VI
THE COMPARISON OF THE NUMBER OF THE MATCHED PIXELS AND THE ERROR IN THE CCSDP AND DP ALGORITHMS.

	CCSDP Algorithm		DP Algorithm	
Scene	Match	Error	Match	Error
Ball	1295450	2000	125349	7258
Barn1	125950	5370	121333	6493
Barn2	121215	8376	107707	21014
Poster	123255	7319	94285	23222
Venus	124892	7535	115287	12813

Fig. 6 shows the implementation results on Ball, Barn1, and Poster color stereo scenes. The Fig. 6(a) shows the source left image of each scene and Figs. 6(b) and 6(c) illustrate the disparity map using GLCS and CCS, respectively. Again, the Fig. 6(d) in the each row exposes the obtained dense disparity map from our new algorithm. The images in b and c sections of Fig. 6 are depicted in different colors, where the black color shows the not matching feature points and the red color represents the error matched feature points, and the yellow, and the dark blue colors show the feature points with minimum and maximum disparity respectively. In addition, Fig. 6(d) shows the dense disparity map in intensity range where the black color shows the errors in this disparity map. Regarding these figures, the most errors in the output disparity map are due to occlusions and real errors are very little.

VIII. CONCLUSION

Stereo matching is an important issue in the computer vision. Traditionally, the problem of stereo matching has been addressed either by a sparse feature based approach or a dense area based approach. We present a new dense stereo matching based on a path computation in disparity space using dynamic programming. Application of dense stereo vision in intelligent vehicles requires accurate and robust disparity estimation algorithms that for instance can be run on real time systems.

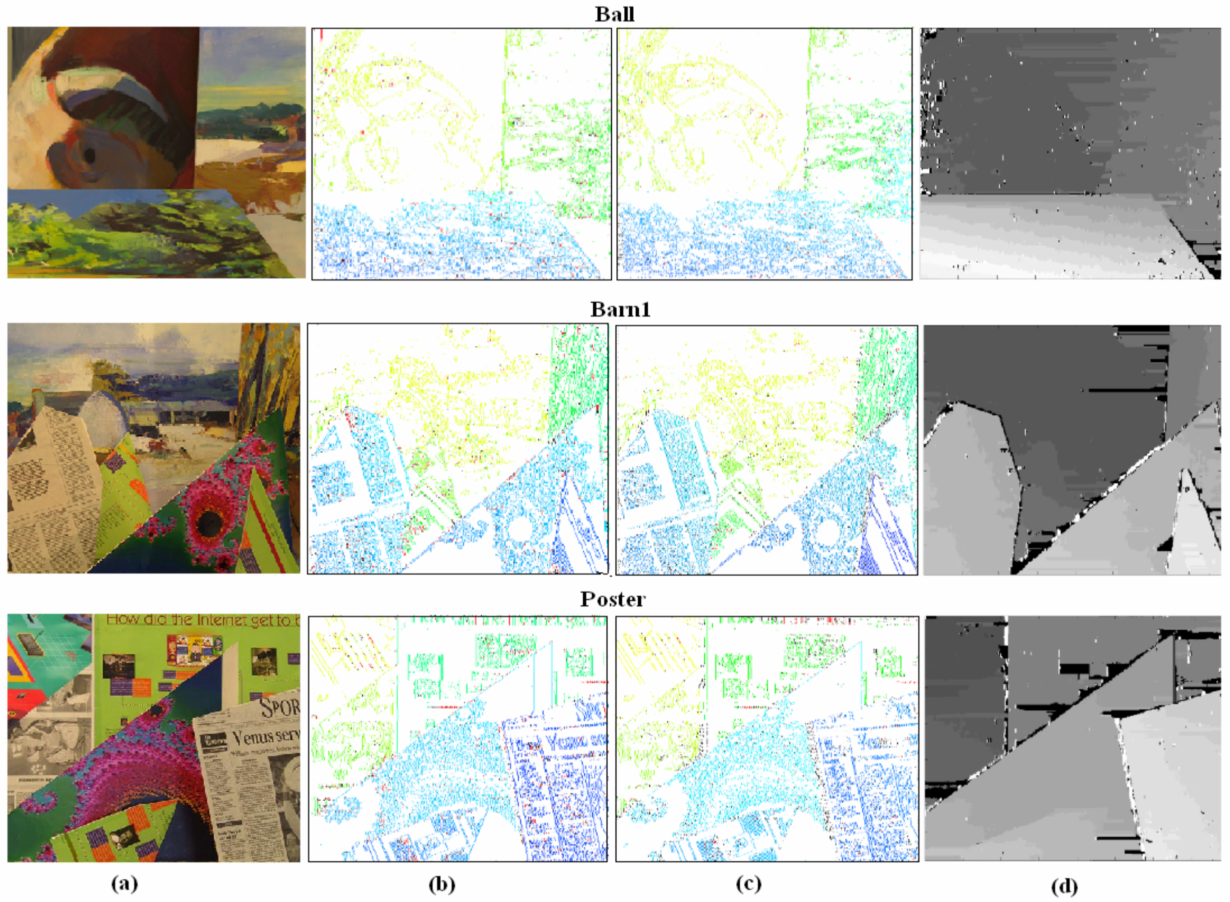

Fig. 6 The experimental results. (a) The left stereo image; (b) The output of the GLCS algorithm; (c) The output of the CCS algorithm; (d) The output of the CCSDP algorithm.

A large amount of research is focused on dense stereo vision since it is important in a number of applications such as robot navigation, surveillance systems, 3D modeling, augmented reality, and video conferencing.

In this paper, we employ color information on stereo images using a feature based approach to get accuracy, and employ dynamic programming on sparse disparity map to get less computation time and obtain a robust dense disparity map. With applying this composition of area based and feature based stereo algorithms, we will gain the both accuracy and speed.

So, the main emphasis of the paper was on forming very quick and accurate dense disparity map employing dynamic programming. To obtain fast and accurate sparse disparity map from initial two stages, we use restricted search, chain correspondence, hierarchical multiresolution, and left-right consistency checking for the first point in any chain.

Our method is composed of three main stages of feature extraction, feature matching, and dynamic programming. In the feature extraction stage, we use Sobel filter to extract the edges of images. In the second stage, we extract correspondence. The output of this stage is a sparse disparity map. In the third stage, we perform dynamic programming on obtained sparse disparity map, and produce a dense disparity map.

Moreover, we found that the quality of the matching results always improves when color information is inserted. This holds for edge based techniques and for dense techniques.

The experiments on our dense algorithm show that the accuracy of results has increased from 20 to 70%, and the running time has reduced about 70%. So, our dense matching results are good enough to allow a robust 3D scene reconstruction.

References

[1] D. Scharstein, and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms”, International journal of computer vision, 2002, 47(1-3), pp. 7-42.

[2] A. Koschan, “What is New in Computational Stereo Since 1989:A Survey on Current Stereo Papers”, Technische Universität Berlin, Technischer Bericht, August 1993, pp. 93-22.

[3] K.I. Tsutsui, M. Taira, and H. Sakata, “Neural mechanisms of three-dimensional vision”, Neuroscience Research 51, 2005, pp. 221–229.

[4] R. Klette, A. Koschan, K. Schlüns, and V. Rodehorst, “Surface Reconstruction based on Visual Information”, Department of Computer Science, Technical Report 95/6, Perth, Western Australia, July 1995, pp. 1-52.

[5] A. Bensrhair, P. Miche, and R. Debrie, “Fast and automatic stereo vision matching algorithm based on dynamic programming method”, Pattern Recognition Letters, 1996, 17, pp. 457–466.

[6] S. Birchfield and C. Tomasi, “Depth discontinuities by pixel-to-pixel stereo”, International Journal of Computer Vision, 1999, pp. 269–293.

[7] Y. Ohta and T. Kanade, “Stereo by Intra- and Interscanline Search Using Dynamic Programming”, IEEE Transactions on PAMI, 1985, 7, pp. 139–154.

[8] O. Veksler, “Stereo Correspondence by Dynamic Programming on a Tree”, Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Volume 2, 2005, pp. 384-390.

[9] C. V. Jawahar and P. J. Narayanan, “A Multifeature Correspondence Algorithm Using Dynamic Programming”, ACCV2002: The 5th Asian Conference on Computer Vision, January 2002, pp. 23–25.

[10] B. Tang, D. Ait-Boudaoud, B. J. Matuszewski, and L. k Shark, “An Efficient Feature Based Matching Algorithm for Stereo Images”, Proceedings of the Geometric Modeling and Imaging-New Trends (GMAI’06), 2006, pp. 195-202.

[11] R.A.Lane, and N.A.Thacker, “Tutorial: Overview of Stereo Matching Research”, Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester, M13 9PT, 1998, pp. 1-10.

[12] C. J. Taylor, “Surface Reconstruction from Feature Based Stereo”, Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV’03), Vol. 1, 2003, pp. 184-190.

[13] S. S. Tan, and D. P. Hart, “A fast and robust feature-based 3D algorithm using compressed image correlation”, Pattern Recognition Letters 26, 2005, pp. 1620–1631.

[14] S. brandt, and J. Heikkonen, “Multi-Resolution Matching of Uncalibrated images utilizing epipolar geometry and its uncertainty”, IEEE International Conference on image Processing (ICIP), Vol. 2, 2001, pp. 213-216.

[15] B. Tang, D. Ait-Boudaoud, B. J. Matuszewski, and L. k Shark, “An Efficient Feature Based Matching Algorithm for Stereo Images”, Proceedings of the Geometric Modeling and Imaging-New Trends (GMAI’06), 2006, 195-202.

[16] R.A.Lane, and N.A.Thacker, “Tutorial: Overview of Stereo Matching Research”, Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester, M13 9PT, 1998, 1-10.

[17] C. J. Taylor, “Surface Reconstruction from Feature Based Stereo”, Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV’03), Vol. 1, 2003, 184-190.

[18] L. Di Stefano, M. Marchionni, and S. Mattoccia, “A Fast Area-Based Stereo Matching Algorithm”, Image and Vision Computing (JIVC), Vol. 22, No 12, pp 983-1005, October 2004.

[19] V. Kolmogorov, and R. Zabih, “Computing visual corresponding with occlusions using graph cuts”, ICCV 2001. Proceedings. Eighth IEEE International Conference on Computer Vision, 2001, Volume 2, 508-515.

[20] L. Di Stefano, and S. Mattoccia, “Fast stereo matching for the videt system using a general purpose processor with multimedia extensions”, Proceedings of the Fifth IEEE International Workshop on Computer Architectures for Machine Perception (CAMP'00), 2000, 356.

[21] O. Faugeras et al, “Real-time correlation-based stereo: algorithm, implementation and applications”, Technical Report 2013, Unite derecherche INRIA Sophia-Antipolis, France, Aout, 1993.

[22] T. Kanade, H. Kato, S. Kimura, A. Yoshida, and K. Oda, “Development of a video-rate stereo machine”, In Proc. Of International Robotics and Systems Conference (IROS ’95), volume 3, pages 95 – 100, August 1995.

[23] K. Konolige. Small vision systems: Hardware and implementation. In 8th Int. Symposium on Robotics Research, pages 111–116, 1997.

[24] P. Moallem, K. Faez, and J. Haddadnia, “Fast Edge-Based Stereo Matching Algorithms through Search Space Reduction”, IEICE Trans. INF. & SYST, Vol.E85-D, No. 11, November 2002, 1859-1871.

[25] P. Moallem and K. Faez, “Effective Parameters in Search Space Reduction Used in a Fast Edge-Based Stereo Matching”, Journal of Circuits, Systems, and Computers, Vol. 14, No. 2, 2005, 249-266.

[26] P. Moallem, M. Ashorian, B. Mirzaeian, and M.Ataei, “A Novel Fast Feature Based Stereo Matching Algorithm with Low Invalid Matching”, WSEAS Transaction on Computers, Issue 3, Vol. 5,March 2006, pp. 469-477.

[27] X. Hua, M. Yokomichi, and M. Kono, “Stereo Correspondence Using Color Based on Competitive-cooperative Neural Networks”, Proceedings of the Sixth International Conference on Parallel and Distributed Computing Applications and Technologies, 2005, pp. 856-860.

[28] Q. Yang, L. Wang, R. yang, H. Stewenius, and D. nister, “Stereo Matching with Color-Weighted correlation, hierarchical Belief Propagation and occlusion Handling”, Proceedings of the 2006 IEEE Computer society Conference on Computer Vision and Pattern recognition (CVPR’06), 2006, pp. 2347-2354.

[29] I. Cabani, G. Toulminet, and A. Bensrhair, “A Fast and Self-adaptive Color Stereo Vision Matching; a first step for road Obstacle Detection”, Intelligent vehicles symposium, 2006, pp. 13-15.

[30] Stereo data sets with ground truth, Middlebury College, Available: http://cat.middlebury.edu/stereo/data.html.

Hajar Sadeghi is M.S. student in Isfahan, Iran. She has born 1982, in Khorasan, Iran. She got her B.Sc. in computer engineering from University of Shahid Beheshti, Tehran, Iran, in 2005.

Payman Moallem, born 1970, in Tehran, Iran. He is an assistant professor in Electrical Engineering Department of University of Isfahan, Iran. He received his B.S. and M.S. both in Electronic Engineering from Isfahan University of Technology and Amirkabir University of Technology, Iran, in 1992 and 1995 respectively. He also received his PhD in Electrical Engineering from Amirkabir University of Technology in 2002. From 1994 to 2002, he has researched in Iranian Research Organization, Science and Technology (IROST) on the topics like, parallel algorithm and hardware used in image processing, DSP based systems and robot stereo vision. His research interests include fast stereo vision, target tracking, real-time video processing, neural networks and image processing, recognition and analysis.

Seyed Amirhassan Monadjemi, born 1968, in Isfahan, Iran. He got his PhD in computer engineering, pattern recognition and image processing, from University of Bristol, Bristol, England, in 2004. He is now working as a lecturer at the Department of Computer, University of Isfahan, Isfahan, Iran. His research interests include pattern recognition, image processing, and human/machine analogy.