3D Reconstruction Based On Block Matching and Image Pyramiding

[1]Mr. K. Rasool Reddy, [2]Mr. K. M. K. Phani Raja, [3]Mr. Ch. Madhava Rao

[1]Assistant Professor & Dept. of ECE, MRITS, Secunderabad, Telangana, India.

[2]PG Scholar & Dept. of ECE, Sir C. R. Reddy College of Engineering, A.P, India

[3]Assistant Professor & Dept. of ECE, Sir C. R. Reddy College of Engineering, A.P, India

ABSTRACT

Nowadays, there is an increase in the usage of geometric 3D models in movie industry, games and virtual environments. There are several existing solutions like user-driven models, 3D scanners but they are not fully satisfying in either in terms of cost and or in efficiency. To overcome these problems, we have introduced a cost-effective solution based on analyzing image sequences. The main focus of this paper is on stereo vision; it is the process of recovering depth from camera images by comparing two or more views of the same scene. This paper mainly focuses on how to compute the depth map between two rectified stereo images. In this paper we use block matching, which is the standard algorithm for high-speed stereo vision in hardware systems. We first explore basic block matching, and then apply dynamic programming to improve accuracy, and image pyramiding to improve speed.

Keywords: Images, 3D reconstruction, stereo vision, block matching, sub-pixel estimation, dynamic programming, image pyramiding.

1. INTRODUCTION

The word "stereo" comes from the Greek word "stereos", which means firm or solid, with stereo vision, the objects are seen solidly in three dimensions with range. Stereo vision is the process of recovering depth from camera images by comparing two or more views of the same scene. Simple, binocular stereo uses only two images, typically taken with parallel cameras that are separated by a horizontal distance known as the "baseline." The output of the stereo computation is a disparity map, which tells how far each point in the physical scene is from the camera.

There are two large families of stereo algorithms, i.e. local and global ones. Local stereo algorithms can provide high frame rates but their accuracy is low. On the other hand, global algorithms suffer from low frame rates but their results are generally very accurate. Block matching and Dynamic Programming (DP) stands somewhere between those two broad classes, providing good accuracy of results in acceptable frame rates.

This paper is organized into 4 sections. In section 1, a brief description of stereo vision. In section 2 the block diagram is described. In section 3 the experimental results are discussed and finally in section 4 conclusion are provided.

2. BLOCK DIAGRAM

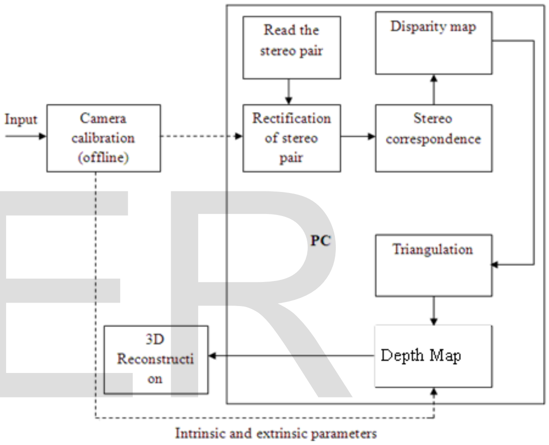

Figure 1: Overview of Stereo vision system

The figure 1 shows the basic block diagram of stereo vision system. It mainly consists of following blocks

1. Camera Calibration
2. Rectification
3. Stereo Correspondence
4. Triangulation
5. Disparity Map
6. Depth Map

2.1. Camera Calibration (Offline) and Epipolar Geometry

Camera calibration

One of the basic tasks in Computer Stereo Vision is to calibrate the stereo camera in order to obtain the parameters that will allow you to calculate 3D information of the scene.

Here the ideal pinhole camera is employed to represent stereo cameras. In this model mapping the 3D points **X** to 2D points **x** in the image plane i.e., **X** = (X, Y, Z) ^{T} maps to **x** = (u, v) ^{T}. This is known as perspective projection which is given by:

$$x = PX \qquad (1)$$

The projection matrix **P** is formed by intrinsic and extrinsic parameters of camera and it is defined as follows:

$$P = K\,[R \mid t] \qquad (2)$$

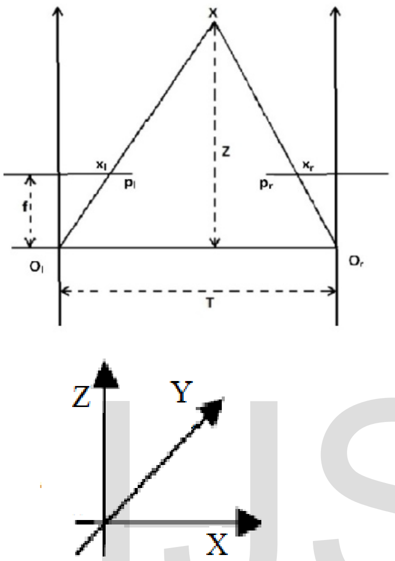

Figure 2: Basic stereo camera model

Where **R (**Rotation**)** and **t** (translation) are the extrinsic parameter. Rotation matrix is a 3×3 matrix which describes the orientation of camera coordinate system with respect to world reference frame and translation vector denotes the position of the camera center in world coordinates and **K** is the matrix consisting of intrinsic camera parameters like focal length 'f', image center ($u_{0,}$ v_0), and skew point's'. Intrinsic camera matrix **K** is given by:

$$K = \begin{bmatrix} fu & s & uo \\ 0 & fv & vo \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Where fu, fv are the focal length measured in width and height of pixels. In this paper K is

$$K = \begin{bmatrix} 409.4433 & 0 & 204.1225 \\ 0 & 416.0865 & 146.4133 \\ 0 & 0 & 1 \end{bmatrix}$$

Epipolar geometry

Epipolar geometry is the geometry of stereo vision. Epipolar geometry is shown in figure 3. X is a point in 3-D space, and E_L and E_R represents the Epipolar poles. The lines that are formed by joining the intersections of E_x on image plane are called Epipolar lines. In other words, Epipolar line of one camera is the projection of a line joining the center of projection of other camera (O_L or O_R) and the corresponding 3-D point (X).

The projection of center of projection of one camera onto the other is called epipole. All the Epipolar lines go through the corresponding camera's epipole. The projections of X on either camera are x_L and x_R, both these points must lie on same Epipolar line. This condition is called Epipolar constraint. This geometry is essential for the processes like rectification, stereo correspondence and triangulation.

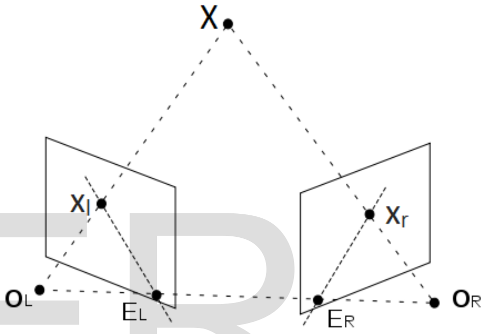

Figure 3: Epipolar geometry

2.2 Rectification

It is the transformation process used to project two or more images on to a common image plane. In most camera configurations, finding correspondences requires a two dimensional search. However, if the two cameras are aligned to be coplanar, the search is simplified to one dimensional.

Generally, rectification is the process of converting general stereo configuration to a simple stereo configuration. By knowing the pair of stereo images, the intrinsic and extrinsic parameters of each camera, image transformation can be computed. This makes Epipolar lines collinear and parallel to the base line as shown in figure 4 by making epipole to infinite. So, by rectifying the images, the matching process becomes faster as searching horizontal Epipolar lines is easier than general Epipolar lines.

Algorithm rectification:

The input is formed by the intrinsic and extrinsic parameters of a stereo system and a set of points in each camera to be rectified. Additional assumptions that to be considered

- The origin if image reference frame is the principle point.
- The focal length is equal to f.

Step 1: Build the matrix R_{rect}.

To carry out the method, we constructed triple of mutually orthogonal unit vectors e_1, e_2 and e_3. The first vector e_1 is given by the epipole; since the image center is in the origin e_1 coincides with the direction of translation,

$$e_{1=}\frac{T}{||T||} \quad (4)$$

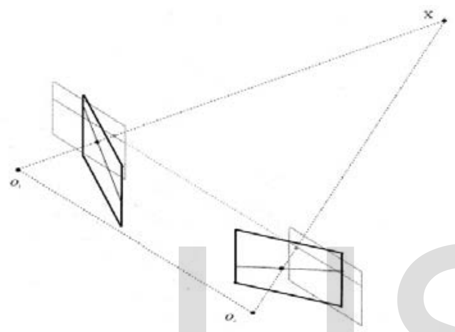

Figure 4: Rectification of stereo pair. The Epipolar lines associated to a 3D point X in the original cameras (dark lines) become collinear in the rectified cameras (light lines)

The only constrained we have on the second vector, $e_{2,}$ is that it must be orthogonal to $e_{1.}$ To this purpose we compute and normalize the cross product of e_1 with the direction vector of the optical axis, to obtain.

$$e_2 = \frac{1}{\sqrt{T_x^2+T_y^2}}[-T_y, T_x, 0]^T \quad (5)$$

The third unit vector is unambiguously determined as

$$e_{3\,=}\, e_{1\,\times}\, e_2 \quad (6)$$

It is easy to check the orthogonal matrix defined as

$$R_{rect}=\begin{pmatrix} e_1^T \\ e_2^T \\ e_3^T \end{pmatrix} \quad (7)$$

Rotates the left camera about the projection center in such a way the Epipolar lines becomes parallel to the horizontal axis. This implements the first step of the algorithm.

Step2: Set $R_L= R_{rect}$ and $R_R= R.\ Rre_{ct}$

Step 3: For each left camera point $X_{\ L}= [x, y, f]^{\ T}$ compute

$$R_L\, X_L= [x^{'}, y^{'}, z^{'}] \ (8)$$

And the coordinates of the corresponding rectified point, $X_L^{\ '}$, as

$$X_L^{\ '} = \frac{f}{z'}[x^{'}, y^{'}, z^{'}] \quad (9)$$

Step 4: Repeat the previous step for the right camera using R_r and x_l.

The output is the pair of transformation to be applied to the two cameras in order to rectify the two input point sets as well as the rectified points.

2.3 Stereo correspondence

Stereo correspondence problem has been one of the most investigated topics in computer vision. The correspondence problem in computer vision concerns the matching of points, or other kinds of primitives, in two or more images such that the matched elements are the projections of the same physical elements in 3D scene, and the resulting displacement of a projected point in one image with respect to the other is termed as disparity.

Similarity is the guiding principle for solving the correspondence problem; however, the stereo correspondence problem is an ill-posed task, in order to make it tractable, it is usually necessary to exploit some additional information or constraints. The most popular constraint is the Epipolar constraint, which can reduce the search to one-dimension rather than two.

The most common method for extracting depth information from intensity images is by means of a pair of synchronized camera-signals, acquired by a stereo rig. The point-by-point matching between the two images from the stereo setup derives the depth images, or the so called disparity maps. This matching is done as a one dimensional search from accurately rectified stereo pairs in which horizontal scan lines reside on the same Epipolar line are assumed, as shown in Figure5. A point P_1 in one image plane may have arisen from any of points in the line C_1P_1, and may appear in the alternate image plane at any point on the so-called Epipolar line E_2.

Thus, the search is theoretically reduced within a scan line, since corresponding pair of points reside on the same Epipolar line. The difference on the horizontal coordinates of these points is the disparity. The disparity map consists of all disparity values of the image.

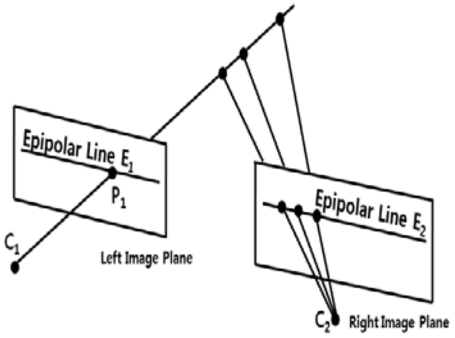

Figure 5: Epipolar constraint for stereo correspondence.

2.4. Disparity Map

Disparity map of rectified images can be achieved by following operations

1. Block Matching
2. Sub-Pixel Estimation
3. Dynamic Programming
4. Image Pyramiding

1. Basic Block Matching:

Basic block matching is one of the local methods in stereo matching algorithms. It is the standard algorithm for high speed stereo-vision in hardware systems. The actual task of block matching is to perform similarity check between two equal sized blocks in the left and right images. For every pixel in the left image we extract a 7-by-7 pixel block around it and search along the same row in the right image as both the images are rectified. Here we use the sum absolute difference (SAD) to compare the image regions.

$$SAD(x,y,d) = \sum(I_L(x,y) - I_R(x,y,d)) \quad (10)$$

Where, I_L, I_R are the intensity values in left and right images, (x, y) are the pixels co-ordinates; d is the disparity value under consideration.

2. Sub Pixel Estimation:

The disparity estimates returned by block matching are all integer-valued, so the above depth map exhibits contouring effects where there are no smooth transitions between regions of different disparity. This can be ameliorated by incorporating sub-pixel computation into the matching metric. Previously we only took the location of the minimum cost as the disparity, but now we take into consideration the minimum cost and the two neighboring cost values. We fit a parabola to these three values, and analytically solve for the minimum to get the sub-pixel correction as shown in figure 6.

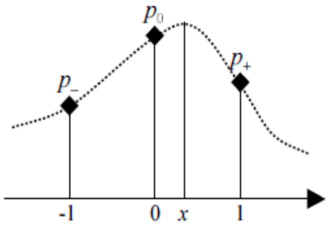

Figure 6: Parabolic Subpixel Estimation

Normal parabola equation is given by,

$$y = ax^2 + bx + c \quad (11)$$

By differentiating above equation w.r.t x and equating to zero

$$\frac{dy}{dx} = 2ax + b = 0 \quad (12)$$

Therefore

$$x = \frac{-b}{2a} \quad (13)$$

Substituting pixel values for each of the 3 known data points into parabola equation,

$$p_- = a - b + c \quad (14)$$

$$p_0 = c \quad (15)$$

$$p_{+=} a + b + c \quad (16)$$

When solved and substituted in second equation results in the estimate of the true point.

$$x = \frac{p_+ - p_-}{4p_0 - 2(p_+ - p_-)} \quad (17)$$

The center point can be either being local minima or local maxima as no assumptions are made. Re-running basic block matching, we achieve the result below where the contouring effects are mostly removed and the disparity estimates are correctly refined. This is especially evident along the walls.

3. Dynamic Programming

Dynamic programming is a semi-global method of stereo matching algorithm. As already mentioned we introduce dynamic programming to improve accuracy. This can be done by introducing a smoothness constraint. Basic block matching chooses the optimal disparity for each pixel based on its own cost function alone. Now we want to allow a pixel to have a disparity with possibly sub-optimal cost for it locally. The problem of finding the optimal disparity estimates for a row of pixels now becomes one of finding the "optimal path" from one side of the image to the other.

Consider two scan lines I_L (i) and I_R (j), 1≤ i,j ≤ N, where N is number of pixels in scan line.

Let d_{ij} be cost associated with matching pixel I_l(i) with pixel I_r(j).here we consider absolute error measure between pixels given by,

$$d_{ij} = \frac{(I_l(i) - I_r(j))}{\sigma^2} \quad (18)$$

Where σ is some measure of pixel noise. The cost of skipping a pixel is given by a constant c_o. For the purpose of analysis here we assume σ = 2 and c_o = 1.

Given this costs we can compute the optimal (minimal cost) alignment of two scan lines as follows.

1. D(0,0) = 0
2. D(i,j) = min(D(i-1,j-1) + d_{ij}, D(i-1,j) + c_0 , D(i,j-1) + c_0)

The intermediate values are stored in an N×N matrix, D. The total cost of matching two scan lines is D (N, N).

Given D, we can find the optimal alignment by back tracking. In particular, starting at (i,j) = (N,N) , we choose the minimal value of D from {(i-1,j-1), (i-1,j) , (i,j-1)}. Selecting (i-1,j) corresponds to skipping a pixel in I_l , while selecting (i,j-1) corresponds to skipping a pixel in I_r. selecting (i-1,j-1) matches pixels (i,j) , and therefore leaves disparity unchanged. Beginning with zero disparity, we can work backwards from (N, N), tallying the disparity until we reach (1, 1).

4. Image Pyramiding

While dynamic programming can improve the accuracy of the stereo image, basic block matching is still an expensive operation, and dynamic programming only adds to the burden. One solution is to use image pyramiding to guide the block matching. Image pyramiding can increase computational efficiency of whole process.

2.4 Triangulation

Triangulation is the process of calculating depth from the disparity map generated by stereo correspondence. The 3D position (X, Y, and Z) of a point X can be reconstructed from the perspective projection of X on the image planes of the cameras, once the relative position and orientation of the two cameras is known.

We choose the 3D world reference system to be the left camera reference system. The right camera is translated and rotated with respect from left one. The optical axes of two cameras are parallel and translation of the right camera is only along the X-axis.

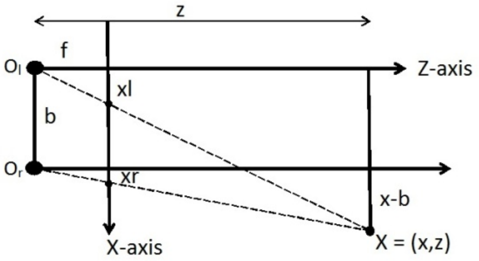

Figure 7: Standard stereo setup

Let us consider a standard model of stereo setup as shown in above figure 7.

From the similar triangles,

$$\frac{z}{f} = \frac{x}{xl} \qquad \frac{z}{f} = \frac{x-b}{xr} \qquad \frac{z}{f} = \frac{y}{yl} = \frac{y}{yr}$$

And disparity,

$$d = (xl\text{-}xr)$$

For stereo cameras with parallel optical axes, focal length f, baseline b, corresponding image points (xl,yl) and (xr,yr), the location of the 3D point can be derived from above equations:

Depth is given by,

$$z = \frac{f \times b}{(xl - xr)} = \frac{f \times b}{d}$$

$$x = \frac{xl \times z}{f} \text{ (or) } \frac{(b + xr) \times z}{f}$$

$$y = \frac{yl \times z}{f} \text{ (or) } \frac{(yr \times z)}{f}$$

3. Results & Discussions

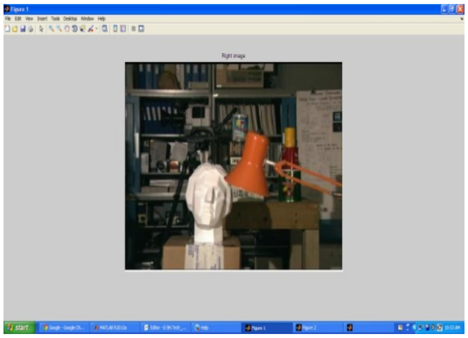

Fig 3.1: Right image

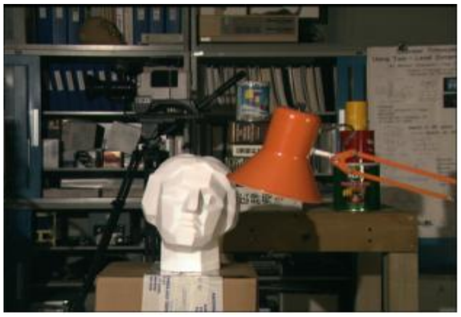

Fig 3.2: Left image

Initially we read the two stereo image pair and convert it in to gray scale for the matching purpose. The above figures show the captured image.

The below figure 3.3 shows the stereo image after converting in to grayscale.

After reading stereo image pair perform the block matching for every pixel in the image for disparity. The resultant image after block matching is shown in figure 3.4.

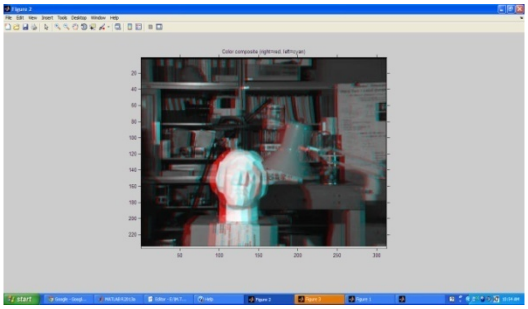

Fig 3.3: Color composite image

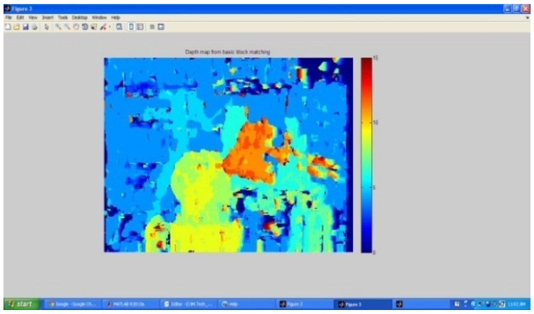

Fig 3.4: Depth map from basic block matching

The disparity estimates returned by block matching are integer values. Due to this depth map exhibits contouring effects these effects can be overcome by sub pixel estimation.

The figure 3.5 shows the estimation of contour effects by sub pixel estimation.

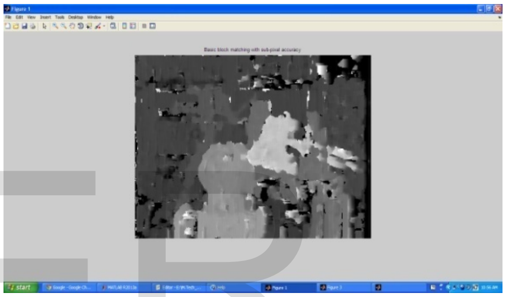

Fig 3.5: Basic block matching with sub-pixel accuracy

The above step creates a noisy disparity image this can be improved by smoothness constraints i.e. Dynamic programming. The figure 3.6 shows enhancement of disparity of block matching.

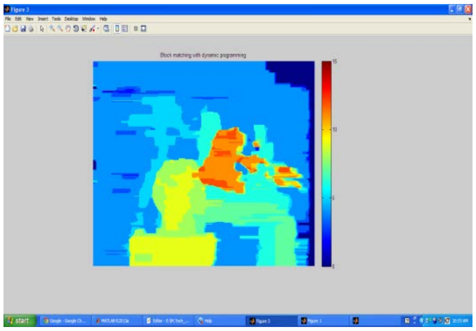

Fig 3.6: Block matching with dynamic programming

To overcome the complexity of block matching technique introduce a new technique i.e. image pyramiding due to this technique the speed of operation is improved.

Finally combine the pyramiding and dynamic programming where dynamic programming is apply on the disparity estimated output by every pyramid level. The figure 3.7 shows the combination of pyramid and block matching.

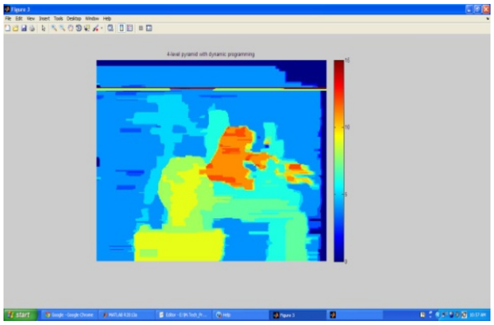

Fig 3.7: 4-level pyramid with dynamic programming

Finally combine all the techniques and apply pyramid with dynamic programming, to reduce the contouring effects.

The figure 3.8 shows the final disparity image.

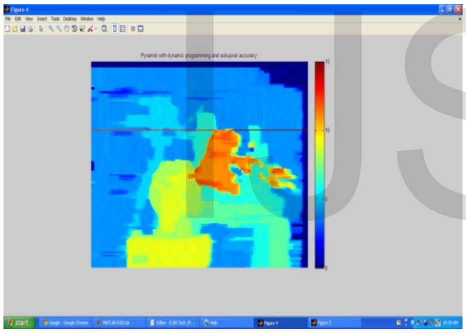

Fig 3.8: Pyramid with dynamic programming and sub-pixel accuracy

The figure 3.9 shows the 3D reconstructed image.

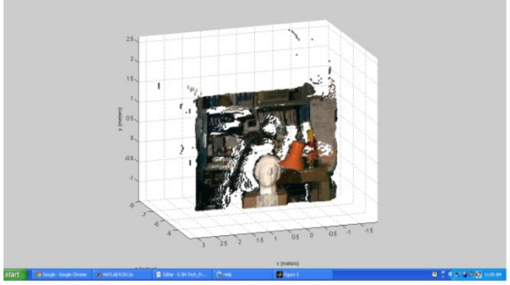

Fig 3.9: 3D reconstructed image

4. Conclusion

The process of stereo matching involves in block matching technique is able to establish a correspondence by matching image pixel intensities. The output of is a disparity mapping which stores the depth or distance of each pixel in an image. Each pixel in the map corresponds to the depth at that point rather than the gray shaded or color.

References

[1] A. Rajeswari, B. Bhuvaneshwari, V. Gowri Priya, “Depth Measurement and 3D Reconstruction of Stereo Images”, IEEE-International Conference on Communication Systems and Network Technologies, 2012.

[2] Zhengyou Zhang, “Flexible Camera Calibration By Viewing a Plane From Unknown Orientations”, IEEE-The Proceedings of Seventh International Conference on Computer Vision, 1999.

[3] Zhengyou Zhang, “Determining the Epipolar Geometry and its Uncertainty: A Review”, International Journal of Computer Vision, 1998.

[4] Andrea Fusiello, Emanuele Trucco, Alessendro Verri, “A Compact Algorithm for Rectification on Stereo pairs”, Machine Vision and Applications, 2000.

[5] Xi Liu, Dahai Li, Xiaoyong Liu, Qionghua Wang, “A Method of Stereo Vision Rectification and its Application in Stereo Vision Measurement”, IEEE-Second IITA International Conference on Geoscience and Remote Sensing, 2010.

[6] Gaurav Gupta and M.S. Rawat, “Region Growing Stereo Matching Method for 3D Building Reconstruction”, International Journal on Computational Vision and Robotics, Volume 2, 2011.

[7] Lazaros Nalpantidis, Georgios Ch, Sirakoulis and Antonios Gasteratos, “Review of Stereo Matching Algorithm for 3D Vision”, 16th International Symposium on Measurement and Control in Robotics, 2007.

[8] T. Ramaprabha Msc MPhill, Dr.M. Mohamed Sathik, “The Role of Block Matching in Stereo Image Compression” International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 6, June 2012.

[9] Andreas Koschan, Volker Rodehorst, Kathrin Spiller, “Color Stereo Vision Using Hierarchical Block Matching and Active Color Illumination, Proceedings of 13th International Conference on Pattern Recognition, August 1996.

[10] Bailey, Donald G, "Sub-pixel estimation oflocal extrema." Proceeding of Image and Vision Computing New Zealand. 2003.

[11] Hajar Sadeghi, Payman Moallem, and S. Amirhassn Monadjemi, “Feature Based Dense Stereo Matching Using Dynamic Programming and Color”, International Journal of information and Mathematical Sciences, 2008.

[12] R. Boutteau, X. Savatier, J.Y. Ertaud, “A Dynamic Programming Algorithm Applied to Omnidirectional Vision for Dense 3D Reconstruction”, IEEE-Second IAPR Asian Conference on Pattern Recognition, 2013