Analysis of Disparity Map for Stereo Matching Algorithm

Komal D. Bhavsar[1] Prof. Virendrasingh[2]
[1,2]Dept. of Electronics & Communication Engineering
[1,2]Sagar Institute of Research & Technology, Bhopal, Madhya Pradesh, India

Abstract--- This paper presents an analysis to determine disparity map useful for 3D scene reconstruction. Stereo vision systems aim the same by matching two or more images taken from slightly different viewpoints. The main problem that has to be solved is the identification of corresponding pixels, i.e. pixels that represent the same point in the scene. Stereo Matching is not difficult to understand in theory but it is not easy to solve in practice. We describe the quality metrics use for evaluating the performance of stereo correspondence algorithms and the techniques used for acquiring our image data sets and ground truth estimates.
Keywords: Stereo Matching, Disparity, RMS Error

I. INTRODUCTION:

The ability of humans to perceive the three-dimensional world from the two-dimensional projections on the retina is both important and fascinating. Computer vision tries to copy the way how human beings perceive visual information by means of using cameras acting as eyeballs and computers to process the information in an intelligent way as does the human brain.
The ability is partly dependent on perspective effects, that is, the fact that three-dimensional objects look different from different viewpoints. Stereo matching is basic to obtaining depth information from a pair of images and analyzing the 3D structure of objects. Because the process is easily affected by local environments, stereo matching is seemed as an ill-posed problem. In this paper, we compute the Adaptive support-weight approach & Matching (Horizontally Line-Based) algorithm and compare all results with Ground truth using quality matrices RMS Error.

II. STEREO MATCHING

The problem of reconstructing a 3D scene from several viewpoints was first investigated in the fields of aerial photography and human stereopsis. Until relatively recently, the scene reconstruction problem was typically treated as a matching problem where the objective was to match points or features between two or more images. Having obtained a match, the three dimensional position of a point could be determined by triangulation assuming the camera positions were known.
The matching of image points is performed by comparing a region in one image, referred to as the reference image, with potential matching regions in the other image and selecting the most likely match based on some similarity measure. The resulting scene estimate is then invariably represented using a depth-map relative to the reference camera.
As an example of the stereo matching process, consider estimating the three dimensional position of a point P shown in Fig. 1. By correctly matching this point between the two images, the relative shift or displacement of the point can be used to calculate the depth of the point.

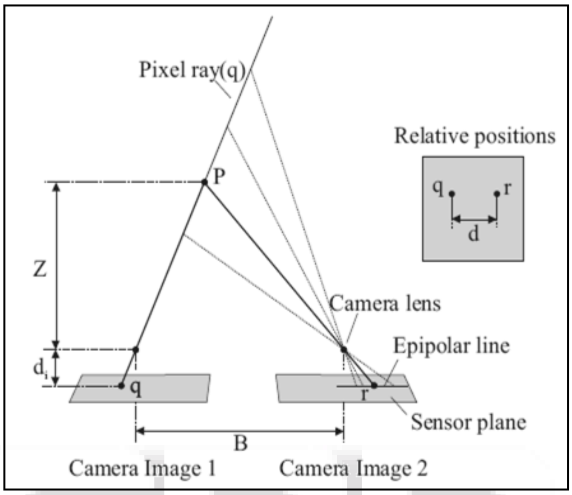

Fig. 1: Demonstration of disparity

As an example consider the point P in Fig. 1. This has image coordinates (x, y) as viewed from camera 1 and image coordinates (x+ d, y) when viewed from camera 2.
By correctly matching this point between the two images the relative shift, or disparity, d, of the point can found. If all cameras have the same focal length, are parallel to each other, and located on the same plane, the magnitude of this disparity is related to the depth, Z,

$$Z = \frac{B \cdot d_i}{d} \quad (1)$$

Where B is the baseline distance between two cameras and di is the distance of the image plane behind the principal point. One problem with this approach is that it is difficult to determine matches reliably because of ambiguities and occlusions. To reduce the number of ambiguities, regions in the image are matched in order to improve the reliability of matching, instead of individual pixels.
However, difficulty with traditional stereo matching is which surfaces that are visible within the reference image may be occluded or hidden from view in one or more of the other images. In this situation false matches will occur as a true match does not exist. To avoid these problems occluded regions must be identified. Matches must then only be formed with images where the corresponding surfaces are visible. Identifying these surfaces is difficult with traditional stereo matching, since the matching is performed directly in 2D image space where occlusions cannot be properly modeled.

A. Solving the Correspondence Problem

The correspondence problem consists in finding correct

Point-to-point correspondences between images or models. If we can identify the same 3D point in both views we can estimate its 3D coordinates. The fundamental hypothesis behind multi-image correspondence is that the appearance of any sufficiently small region in the world changes little from image to image. In general, appearance might emphasize higher-level descriptors over raw intensity values, but in its strongest sense, In other words, if image points p and q are both images of some world point X, then the color values at p and q are equal. This color constancy hypothesis is in fact true with ideal cameras if all visible surfaces in the world are perfectly diffuse (i.e., Lambertian). In practice, given photometric camera calibration and typical scenes, color constancy holds well enough to justify its use by most algorithms for correspondence.

The geometry of the binocular imaging process also significantly prunes the set of Possible correspondences, from lying potentially anywhere within the 2D image, to lying necessarily somewhere along a 1D line embedded in that image[2][3]. Suppose that We are looking for all corresponding image point pairs (p, q) involving a given point q (Figure 2). Then we know that the corresponding world point X, of which q is an image, must lie somewhere along the ray through q from the center of projection Q. The image of this ray Qq in the other camera's image plane ∏ lies on a line l that is the intersection of ∏ with the plane spanned by the points P, Q and q. Because X lies on ray Qq, its projection p on ∏ must lie on the corresponding epipolar line l. This observation, that given one image point, a matching point in the other image must lie on the corresponding epipolar line, is called the epipolar constraint.

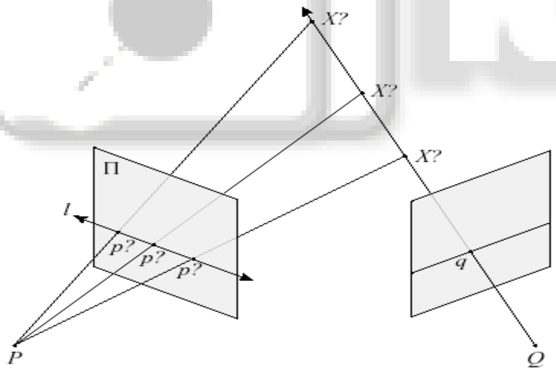

Fig. 2: The geometry of the epipolar constraint:

Based on color constancy and the epipolar constraint, correspondence might proceed by matching every point in one image to every point with exactly the same Color in its corresponding epipolar line. However, this is obviously awed: there would be not only missed matches at the slightest deviation from color constancy, but also potentially many spurious matches from anything else that happens to be the same color. Thus, some additional constraint is needed in order to reconstruct a meaningful three-dimensional model.

Marr and Poggio proposed two such additional rules to guide binocular correspondence [5][6]:

Uniqueness, which states that “each item from each image may be assigned at most one disparity value,” and

Continuity, which states that “disparity varies smoothly almost everywhere."

However, this latter interpretation, that each image location be assigned at most one disparity value, is however very prevalent in practice; only a small number of stereo algorithms attempt to find more than one disparity value per pixel. This common simplification is in fact justifiable, if pixels are regarded as point samples rather than area samples, under the assumption that the scene consists of opaque objects: in that case, each image point receives light from, and is the projection of, only the one closest world point along its optical ray.

In explaining the continuity rule, Marr and Poggio observed that matter is cohesive, it is separated into objects, and the surfaces of objects are generally smooth compared with their distance from the viewer. These smooth surfaces, whose normals vary slowly, generally meet or intersect in smooth edges, whose tangents vary slowly. When projected onto a two-dimensional image plane, these three dimensional features result in smoothly varying disparity values almost everywhere in the image, with only a small fraction of the area of an image composed of boundaries that are discontinuous in depth.

III. STEREO MATCHING ALGORITHMS

Stereo Matching Algorithms like Matching (Horizontally Line-Based), Adaptive Support-Weight Approach for visual correspondence search is studied.

A. *Matching (HORIZONTALLY LINE - Based)*

This algorithm is based on region growing [5][6]. Here, region-growing mechanism comprises of two phases. First phase, finding First point to grow region and the second phase, growing region for a First point corresponding to predefined rule. Rule for associating a point to First point in the growing process is to have lower error energy than a predetermined threshold of error energy (Line Growing Threshold). Being associated to a First point means to have the same disparity by First point. Basic Steps of algorithm are as follows:

Step 1: (First Selection Process) Select a point, which isn’t belonging to any grown region and find its disparity using energy function equation.

$$e(i,j,d) = \frac{1}{3\cdot n\cdot m}\cdot\sum_{x=i}^{i+n}\sum_{y=j}^{j+m}\sum_{k=1}^{3}\left(L(x,y+d,k)-R(x,y,k)\right)^2 \qquad (2)$$

Set it First point and set its disparity to region disparity then go to step 2. If we are not able to find any disparity with lower enough error energy, repeat this step for the next point. If the error energy of selected point is equal or lower than Threshold (T), select it as root point and go to step 2. If not, marked the point idle and do Step 1 for following point in the row.

Step 2: (Region Growing Process) Calculate error energy of neighbor points just for First point disparity, which was called region disparity. If it is equal to or lower than the predetermined error energy threshold T, associate this point to region. Otherwise, left it free. And go back to step 1 to find a new First point.

Step 3: Proceed the Step 2 until region growing any more. In the case that region growing is completed, turn back to step 1 to find out new root point to repeat these steps. When all points in image are processed, stop the algorithm. Grown disparity regions composes disparity map d (i, j)

B. Adaptive Support-Weight Approach for Correspondence Search

This algorithm can be divided into three parts:

1) Adaptive support-weight computation

In this algorithm, the support-weights of the pixels in a given support window is calculated using color similarity and geometric proximity. Visual grouping is very important to form a support window and to compute support-weights and, therefore, the gestalt principles can be used to compute support-weights. Similarity and proximity are the two main grouping concepts in classic gestalt theory.

The gestalt principles of similarity and proximity are also used to compute support-weights. We compute the support-weight of a pixel based on the strength of grouping by similarity and proximity—the support-weight is in proportion to the strength of grouping. They are more similar to the color of a pixel, the larger its support-weight. In addition, the closer the pixel is, the larger the support-weight. The former is related to the grouping by similarity and the latter is related to the grouping by proximity.

2) Support-Weight Based on the Gestalt Grouping

The support-weight of a pixel can be written as, w (p, q) = f (Δcpq, Δgpq) where Δcpq and Δgpq represent the color difference and the spatial distance between pixel p and q, respectively. Here Δcpq and Δgpq can be regarded as independent events and the strength of grouping by similarity and proximity can be measured separately. Then f(Δcpq, Δgpq) can be expressed as

f(Δcpq, Δgpq) = fs(Δcpq) fp(Δgpq)

where fs(Δcpq) and fp(Δgpq) represent the strength of grouping by similarity and proximity, respectively. As shown in (2), the core of the support-weight computation is how to model the strength of grouping by color similarity fs(Δcpq) and the strength of grouping by proximity fp(Δgpq)

3) Strength of Grouping by Proximity

According to the gestalt principle of proximity, the support-weight of a pixel decreases as the spatial distance to the reference pixel increases. Here, as in the color difference, only small spatial distances strongly correlate with the human discrimination performance. Therefore, the strength of grouping by proximity is defined using the Laplacian kernel as

$$f_p\left(\Delta g_{pq}\right)=\exp\left(-\frac{\Delta g_{pq}}{\gamma_p}\right) \quad (3)$$

Where Δgpq is the Euclidean distance between p and q in the image domain and γp is determined according to the size of the support window as γp is proportional to window size. In fact, γp is related to the field-of view of the human visual system.

4) Strength of Grouping by Similarity

The difference between pixel colors is measured in the CIE Lab color space because it provides three-dimensional representation for the perception of color stimuli. As the distance between two points in the CIE Lab color space increases, it is reasonable to assume that the perceived color difference between the stimuli that the points represent increases accordingly. When Δcpq represents the Euclidean distance between two colors, cp = [Lp, ap, bp] and cq = [Lq, aq, bq] in the CIE Lab color space, the perceptual difference between two colors is expressed as

$$D\left(c_p, c_q\right)=1-\exp\left(-\frac{\Delta c_{pq}}{\gamma}\right) \quad (4)$$

Where γ is 14. Based on (3), the strength of grouping by color similarity is defined as

$$f_s\left(\Delta c_{pq}\right)=\exp\left(-\frac{\Delta c_{pq}}{\gamma_c}\right) \quad (5)$$

Support-Weight Based on the Strength of Grouping

According to (3) and (5), w (p, q) = f (Δcpq, Δgpq) can be rewritten as

$$w(p,q)=\exp\left(-\left(\frac{\Delta c_{pq}}{\gamma_c}+\frac{\Delta g_{pq}}{\gamma_p}\right)\right) \quad (6)$$

5) Dissimilarity Computation and Disparity Selection

The dissimilarity between pixels is measured by aggregating raw matching costs with the support weights in both support windows. To minimize the effect of such pixels, we compute the dissimilarity between pixels by combining the support-weights in both support windows. The dissimilarity between pixel p and pd, E (p, pd), can be expressed as

$$E(p,\bar{p}_d)=\frac{\sum_{q\in N_p,\bar{q}_d\in N_{\bar{p}_d}} w(p,q)\cdot w(\bar{p}_d,\bar{q}_d)\cdot e(q,\bar{q}_d)}{\sum_{q\in N_p,\bar{q}_d\in N_{\bar{p}_d}} w(p,q)\cdot w(\bar{p}_d,\bar{q}_d)} \quad (7)$$

Where $\bar{p}_d$ and $\bar{q}_d$ are the corresponding pixels in the target image when the pixel p and q in the reference image have a disparity value d. $e(q,\bar{q}_d)$ represents the pixel-based raw matching cost computed by using the colors of q and qd. When using the truncated AD, it can be expressed as

$$e(q,\bar{q}_d)=\min\left\{\sum_{c\in\{r,g,b\}}\left|I_c(q)-I_c(\bar{q})\right|, T\right\} \quad (8)$$

Where Ic is the intensity of the color band c and T is the truncation value that controls the limit of the matching cost. The disparity of each pixel is simply selected by the WTA method without any global reasoning as

$$d_p=\arg\min_{d\in S_d} E(p,\bar{p}_d) \quad (9)$$

Where Sd = { dmin,….,dmax} is the set of all possible disparities.

IV. EVALUATION METHODOLOGY

I compute the following measures based on known ground truth data: RMS error between the computed disparity map dC(x, y) and the ground truth map in fig (4) dT(x,y),i.e.

$$RMSError=\left(\frac{1}{N}\sum_{(x,y)}\left|d_c(x,y)-d_t(x,y)\right|^2\right)^{\frac{1}{2}} \quad (9)$$

Results of RMS Error is shown in Table-1.

V. OBSERVATION & RESULTS

Results of Algorithms introduced in the paper for the test images were given below.RMS Error is used for comparison of algorithm shown in table 1.

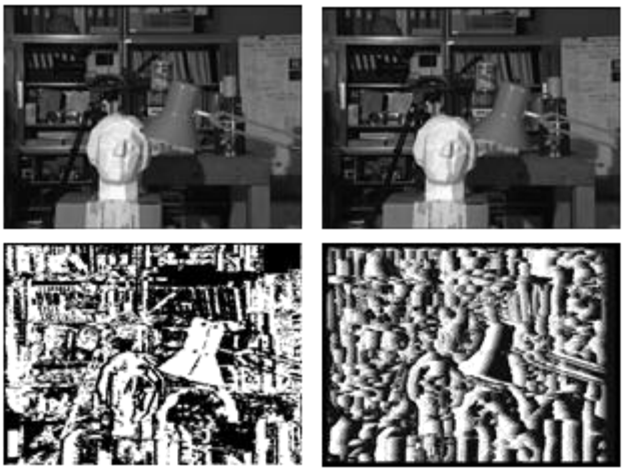

Fig. 3: Stereo Image pair, Disparity Map of Matching, Adaptive Support Weight Approach for Correspondence Search

Sr. No.	Algorithm	RMS Error	Time Taken (Sec)
1	Adaptive Support Weight Approach For Correspondence Search	1.1430e+003	0.0141
2	Matching (Horizontally Line-Based)	1.2366e+003	0.0048

Table 1 RMS Error Comparisons

Fig. 4: Ground Truth Image of Disparity Map

VI. CONCLUSION

Matlab R2007b has been chosen for implementing different Stereo Matching Algorithms. Stereo Matching Algorithms like Matching (Horizontally Line-Based), Adaptive Support-Weight Approach for visual correspondence search have been implemented to generate disparity Map. Matching (Horizontally Line-Based) gives good disparity Map. In Matching (Horizontally – Line Based) Algorithm, as the value of threshold T is decreased, the matching criteria becomes strict and many pixels remains unmatched.RMS error Less in Adaptive Support Weight Approach for Correspondence Search than Matching (Horizontally Line-Based).

VII. FUTURESCOPE

Disparity Maps are successfully generated by implementing Stereo Matching Algorithms, but still there is a scope for improvement. Performance of the Stereo Matching Algorithms is affected by the illumination conditions, shape and the camera characteristics. Effects of these three on Disparity Map. Depth Map can be Generate. Height and Width of an object can also be tried to be calculated. 3D view can also be generated by using Disparity Map and Depth Map.

REFERENCES

[1] R. Gonzales, P.Wintz, woods, Digital Image Processing, Addition-Wesley, 1987
[2] Richard Hartley, Andrew Zisserman, Multiple View Geometry in Vision, 2002
[3] A. Ian H. Witten, Eibe Frank, The Geometry from Multiple Images, MIT Press, 2000.
[4] Adaptive Support-Weight Approach for Correspondence Search by Kuk-Jin-Yoon and In-So Kweon
[5] A Computational Theory of Human Stereo Vision by D. Marr and T.Poggio
[6] Cooperative Computation of Stereo Disparity by D. Marr and T. Poggio
[7] Locally Adaptive Support-Weight Approach for Visual Correspondence Search by Kuk-Jin-Yoon and In-So Kweon
[8] A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms by Daniel Scharstein and Richard Szeliski
[9] A Cooperative Algorithm for Stereo Matching and Occlusion Detection by C. Lawrence and Kanade
[10] A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment by Takeo Kanade and M. Okutomi
[11] www.middlebury.edu