Depth and Distance Information from Stereo Vision Using Sum of Absolute Differences Algorithm

Le Thanh Hai **S. H. Cho** **S. Choi** **H. Hwang**

Dept. of Biomechatronic Engineering, Faculty of Life Science & Technology, Sungkyunkwan University

Abstract

This paper presents an area-based stereo algorithm suitable to real time applications. The core of the algorithm depends on the uniqueness constraint and on a matching process that allows for rejecting previous matches. The proposed approach is compared with the left right consistency constraint, being the latter the basic method for detecting unreliable matches in many area-based stereo algorithms. We used the watermelon and tomatoes for experiments.

Keywords: Stereo vision, Sum of Absolute Differences Algorithm

1. Introduction

Depth measurements are required in applications such as teleconferencing, robot navigation and control, exploration and modelingof unstructured environments, virtual reality. According to a recent taxonomy (of D. Scharstein and R. Szeliski), the stereo algorithms that generate depth measurements can be divided into two classes, namely global and local algorithms. Global algorithms rely on iterative schemes that carry out disparity assignments on the basic of the minimization of a global cost function. These algorithms yield accurate and disparity measurements but exhibit a very computational cost that renders them unsuited to real-time applications. Local algorithms also referred to as area-based algorithms, compute the disparity at each pixel on the basic of the photometric properties of the neighboring pixels. Compared to global algorithms, local algorithms yield less accurate disparity maps but can run fast enough to be deployed in many real-time applications.

As far as local matching algorithms are concerned, and considering the more common case of a binocular stereo imaging system, a widely adopted method aimed at detecting unreliable matches, such for example those due to occlusions or photometric distortions, is the so called left right consistency constraint, also referred to as bidirectional matching or left-right check.

The method can be described as follows. Initially, for each point of the left image find the best match into the right image. Then, reverse the role of the two images and for each point of the right image find the best match into the left image. Finally, keep only those matches that turn out to be coherent when matching left to right (direct matching phase) and right to left (reverse matching phase). It is worth observing that in both phases the match associated with each pixel is established independently of those found at neighboring pixels, since the other matching phase will highlight ambiguous matches. The left right check has proven to be particularly effective in detecting and discarding the erroneous matches necessarily yield by area-based algorithms in presence of occlusions. However, this approach is characterized by a significant computational cost.

In fact, it requires two matching phases (direct and reverse) and, although some authors have proposed calculation schemes aimed at reducing the impact of the left right check on the overall stereo execution time, in most implementations this implies doubling the computational complexity of the matching process.

We apply a local algorithm, which enables real-time stereo applications on a standard Personal Computer. The algorithm is based on a matching core that detects unreliable matches during the direct matching phase and therefore does not require a reverse matching phase.

2. Materials and Method

1) Stereo matching algorithm

The left image is the reference, the disparity, d, belongs to the interval [d_{min} ... d_{max}] and the left image is scanned from top to bottom and from left to right during the matching process. The process, starting from one point of the left image, say L(x, y) searches for the best candidate by evaluating function , within the interval [R(x, y) ... R(x+d, y)]. Then for the successive point of reference image L(x+1, y) the procedure is repeated searching for the best matching within [R(x+1, y) ... R(x+1+d, y)]. The process is then iterated for the successive points along the scanline. This will bring one point of the left image into the same point R(x+d, y) of the right image.

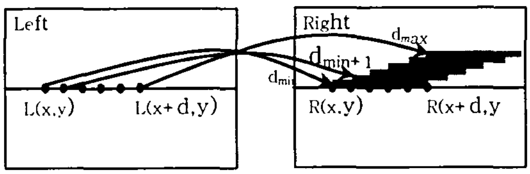

Fig. 1 Matching from left to right.

Now that the best match found for L(x, y) is R(x++d, y) with similarity score (x++d, x, y). We adopt the notation L(x, y) R(x++d, y) to indicate that this match from left to right has been established.

This is area-based stereo algorithm; we use photometric properties, encoded by the error (similarity) function, even though this cue may be ambiguous, due to many causes such as for example photometric distortion, occlusion and signal noise. However, wrong matches expose inconsistencies within the set of matches already established than can be deployed to detect and discard them.

Thus, let suppose that another point of the right image, R(x++d, y), with , has previously matched with L(x, y) with score (x++d, x, y). This situation, that violates the uniqueness constraint, is used to detect wrong matches. Based on the uniqueness constraint that at least one of the two matches, i.e. L(x, y) R(x++d, y) or L(x, y) R(x++d, y), is wrong and retain the match having the better score. So, if the point currently analyzed R(x++d, y) has a better score than L(x++d, y) (i.e. (x++d, x, y) (x++d, x, y)) algorithm will reject the previous match and accept the new one. This implies that, although the proposed approach relies on a direct matching phase only, it allows for recovering from possible previous matching errors.

The capability of the algorithm recovers from previous errors as long as better matches are found during the search.

We notice that greater d values correspond to scores computed more recently while smaller d values to scores computed earlier. With two matches (1): L(x, y) R(x++d, y) and (2): L(x, y) R(x++d, y), the algorithm will discard the old one because (1) is the new one and has the better score with L(x, y).

2) Overall stereo matching

The overall stereo matching consists of three main steps.

- The input images are normalized by subtraction of the mean values of the intensities computed in a small window centered at each pixel (i.e. kernel 3x3). This allows for compensating for different settings of the cameras and different photometric conditions. This information is used to detect regions with lack of texture.
- The normalized images are matched according to the matching approach, selecting a neighborhood of a given square size from the reference image and comparing this neighborhood to a number of neighborhoods in the other image (along the same row), which is independent of the error (similarity) function, using the SAD (Sum of Absolute Differences) error function.

- The final step performs sub-pixel refinement of disparities. Sub-pixel accuracy is achieved detecting the minimum of a second degree curve interpolating the SAD scores in proximity of the minimum found by the matching core and comparing with left image to take a matching image.

3) Computation

The most expensive task performed by the stereo algorithm is the computation of SAD scores, which are needed to carry out the direct matching phase. We show the basic calculation scheme first

Supposing that SAD(x, y, d) is the SAD score between a window of size (2n+1)*(2n+1) centered at coordinates (x, y) in the left image and the corresponding window centered at (x+d, y)

$$SAD(x,y,d)=\sum_{i=-n}^{n}\sum_{j=-n}^{n}|L(x+j,y+i)-R(x+j+d,y+i)|$$

And

SAD(x, y+1, d) = SAD(x, y, d) + U(x, y+1, d)

With U(x, y+1, d) representing the difference between the SADs associated with the lowermost and uppermost rows of the matching window.

$$U(x,y+1,d)=-\sum_{j=-n}^{n}|L(x+j,y-n)-R(x+d+j,y-n)|$$
$$+\sum_{j=-n}^{n}|L(x+j,y+n+1)-R(x+d+j,y+n+1)|$$

We computed U(x, y+1, d) from U(x-1, y+1, d) by simply considering the contributes associated with the four points at four corner of the matching window.

U(x, y+1, d) = U(x-1, y+1, d) + (|L(x+n, y+n+1) – R(x+d+n, y+n+1)| - |L(x+n, y-n) – R(x+d+n, y-n)|) – (|L(x-n-1, y+n+1) – R(x+d-n-1, y+n+1)| - |L(x-n-1, y-n) – R(x+d-n-1, y-n)|)

Thus, we can keep complexity small and independent of the size of the matching window, since only four elementary operations are needed to obtain the SAD score at each new point.

To implement efficiently our matching algorithm, which is based on dis-ambiguity between the collisions occurring while matching left to right along a row, when matching a point of the left image it is necessary to be able to obtain quickly the SAD scores associated with the previous points along a row.

The pre-processing step requires computation of the mean and variance of the two images. For example the left image and N^2 = (2n+1)*(2n+1), the mean is given by:

$$\mu(x,y)=\frac{1}{N^2}\sum_{i,j=-n}^{n}L(x+j,y+i)=\frac{1}{N^2}S_1(x,y)$$

The variance can be expressed:

$$\sigma^2(x,y)=\frac{1}{N^2}\sum_{i,j=-n}^{n}L^2(x+j,y+i)-\mu^2(x,y)$$

$$=\frac{1}{N^2}S_2(x,y)-\mu^2(x,y)$$

In both the matching and pre-processing steps, it is possible to introduce a third level of incremental computation aimed at achieving additional speed up.

$S_1(x, y+1) = S_1(x, y) + U_{S1}(x, y+1)$

$$U_{S_1}(x,y+1)=\sum_{j=-n}^{n}(L(x+j,y+n+1)-L(x+j,y-n))$$

$U_{S1}(x, y+1) = U_{S1}(x-1, y+1)$

$+ (L(x+n, y+n+1) - L(x+n, y-n))$

$- (L(x-n-1, y+n+1) - L(x-n-1, y-n))$

And $U_{S2}(x, y+1)$ is similar to $U_{S1}(x, y+1)$. Both steps use the four pixels at the corners of the correlation window.

In the matching step the third level of incremental computation is applied for each disparity value d $[d_{min}, d_{max}]$; hence, the array T, which is the array of the right term on the right image with each element can be referenced to the index $\widetilde{x}$=x mod (2n+1), grows by one dimension:

$U(x, y+1, d) = U(x-1, y+1, d) + (|L(x+n, y+n+1) - R(x+d+n, y+n+1)| - |L(x+n, y-n) - R(x+d+n, y-n)|) - T(\widetilde{x}, d)$

$T(\widetilde{x}, d) = |L(x-n-1, y+n+1) - R(x+d-n-1, y+n+1)| - |L(x-n-1, y-n) - R(x+d-n-1, y-n)|$

(with $\widetilde{x}$=x mod (2n+1), d $[d_{min}, d_{max}]$)

The described computation can be extended easily to other error (similarity) functions such as Sum of Squared Differences (SSD) and Normalized Cross Correlation (NCC).

3. RESULTS AND DISCUSSION

In this section we show the experimental results using this algorithm.

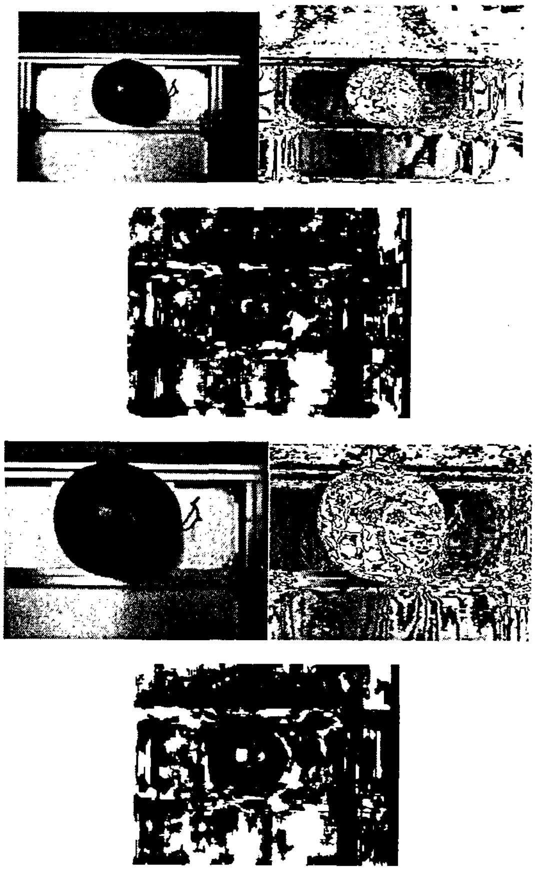

Fig. 2 Reference images and matching images with the different distances from camera to watermelon.

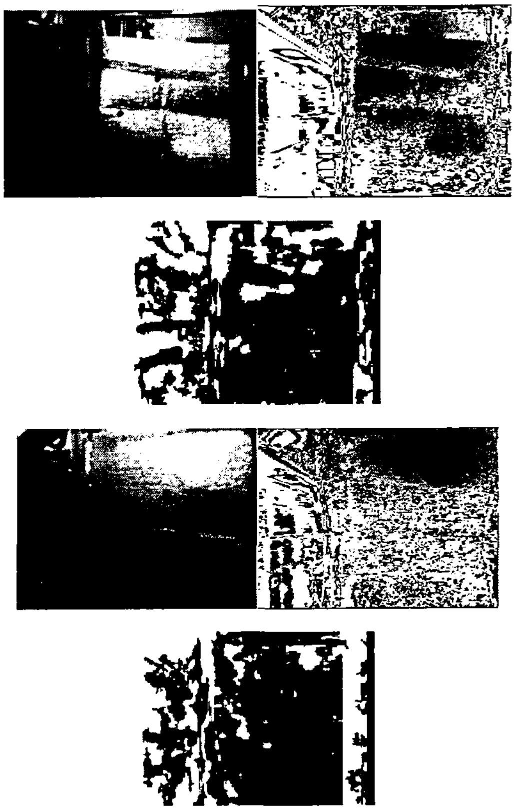

Fig. 3 Reference images and matching images from camera to tomato.

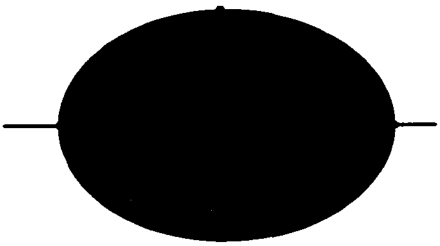

Fig. 4 Radius of watermelon h.

Depending on a disparity image, we calculate the distance from camera to center and boundary of watermelon and then we know the radius of watermelon h.

h= distance from camera to boundary – distance from camera to center

Table 1 Radius of watermelon

Distance of center(m)	Distance of boundary(m)	Radius h(m)
0.666	0.771	0.105
0.597	0.709	0.112
0.506	0.614	0.108

We took the images of tomato to check this algorithm but the illumination is not good so we did not recognized three tomatoes in these images with different distance (0.926; 0.810; 0.721).

However, we still see the shadow of another watermelon in the matching image. This causes clearly an inaccurate fitting of the object. The problem is inherent to local algorithm since it depends on the method adopted to establish correspondences, which relies on the use of a local support area.

4. CONCLUSION

We have presented an area-based stereo matching algorithm, which relies only on a left to right matching phase. This is the algorithm adopted to detect matches in area-based stereo conceived for real-time applications. We propose a further level of incremental calculation, which avoids redundant computations that take place within the correlation window and recover precise object boundaries and smooth surface.

5. REFERENCES

1. L. Di Stefano, M. Marchionni, S. Mattoccia, and G. Neri. A Fast Area-Based Stereo Matching Algorithm. 15th IAPR/ CIPRS International Conference on Vision Interface May 27-29,2002, Calgary,Canada.

2. Stan Birchfield and Carlo Tomasi. Depth Discontinuities by Pixel-to-Pixel Stereo. Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India.

3. Point Grey Research. Triclops Stereo Vision SDK Manual. 2003.

4. Craig Watman, David Austin, Nick Barnes, Gary Overett and Simon Thompson. Fast Sum of Absolute Differences Visual Landmark Detector. Proceedings of IEEE Conference on Robotics and Automation, April, 2004.