A COMPARISON OF TECHNIQUES FOR ESTIMATING DEPTH USING STEREO PAIRS

Laura Wieme
Rochester Institute of Technology
Rochester, NY
ljw5439@rit.edu

Abstract

Disparity maps can provide a visual estimate of depth information generated by a pair of stereo images. The goal when producing disparity maps is to achieve a smooth and detailed map while minimizing error and maximizing efficiency. Three methods of varying complexity are explored to demonstrate and compare their results and efficiency on the Middlebury stereo test set [5].

1. Background

Computation Stereo refers to the problem of determining structure of a scene as described by images taken at distinct, usually binocular viewpoints. Then through correspondence, it is possible to locate the image locations that equate to the same physical space. However, solving the correspondence problem is difficult due to ambiguous matches caused by occlusion, specularity, and smooth textures [3]. Most stereo matching approaches can be categorized as either local or global. Local methods attempt to match pixels across two images examining a small number of pixels surrounding the pixel of interest [3]. Consequently, local methods can be very efficient, but they lack sophistication to handle the previously described ambiguous matches. Global methods are more computationally expensive, but they try to handle the problem of ambiguous matches, most commonly the problem of occlusion. Previous work done to try and solve the stereo correspondence problem includes the local method of block matching and global correspondence methods such "Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure" as proposed by Klaus, Sormann, and Karner, "A Cooperative Algorithm for Stereo Matching and Occlusion Detection" by Zitnick and Kanade, as well as many more. A more complete list of correspondence methods and their evaluations can be found at the Middlebury stereo website [5]. The first algorithm, block matching, was chosen due to its simplicity and as a comparison of the results and efficiency of local versus global correspondence methods. The remaining methods were chosen due to their rankings according to Middlebury and the quality of their documentation.

2. Introduction to Block Matching

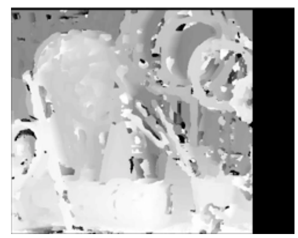
In block matching, a window is used to search one image for the best corresponding template region in the other image [3]. The block is slid over the image and a square-sense error is computed. For this implementation, the sum of squared differences (SSD) metric was used. SSD commonly used for its computational simplicity, but other commonly used metrics include Normalized Cross-Correlation and Sum of Absolute Differences (SAD) [3]. The disparity is then calculated as the areas with minimum error, and therefore where the images are the most similar.

3. Block Matching Implementation

The implementation of the block matching method, as well as all following methods, was realized in Matlab and tested on the Middlebury stereo data sets. The right and left images were read using imread and then processed using rgb2gray and im2double. Also, a blank disparity map was generated to be the size of our original images. Window size, the minimum disparity and the maximum disparity are set in the calling of the Block function. I found that a window size of 9, a minimum disparity of zero, and a max disparity of 70 worked well with most of the Middlebury data set. If the ground truth images are provided with the stereo data set, as they are with Middlebury, a good method to choose a max disparity is to take the maximum of the ground truth image and then divide it by the scale factor of the images being used. The quality of the results produced by the block matching algorithm is highly dependent on the window size and maximum disparity value chosen. Unfortunately, these values vary depending on the input images. Once these values have been chosen, the Block function processes through the image calculating the sum of squared differences and comparing for the most similar matches and filling in the resulting disparity map. The map can then be displayed once the minimum image values have been set to and the maximum values are mapped to the max disparity.

4. Block Matching Results

Ground Truth



Block Matching

The resulting image was computed in 46.957 seconds on an Apple Macbook Pro with a 2.26GHz Intel Core 2 Duo and 2GB 1067 MHz DDR3 memory with an original image size of 370 x 463. The disparity map created from the block matching algorithm achieved reasonable depth indiction in little time, however there are possible areas for improvement. Currently, the disparity values were limited to integers. Using interpolation techniques, sub-pixel precision could be obtained. A large problem with the block matching method is occlusion. When a point is only visible in one of the two images it then lacks a match during the block matching process and results in areas of error. Some of these areas maybe able to be corrected by calculating both a right to left disparity map and a left to right map and then comparing the differences. For now, a simple medfilt2() process helps to mitigate some of the noisy areas containing errors.

5. Introduction to Segment-Based Stereo Matching using Belief Propagation

A stereo matching algorithm was proposed by Klaus, Sormann, and Karner that begins by color segmenting the reference image and then utilizes a matching score to find reliable correspondences. A disparity plane is then assigned to each segment based on the concept of belief propagation. This algorithm assumes that disparity values vary smoothly in regions of homogeneous color and that depth discontinuities will only be found at region boundaries [1]. It is because of this assumed property that the image can be segmented, in this case, using the mean-shift color segmentation method. In order to find a satisfactory number of disparity planes to represent the image geometry, the planes must be estimated using a dissimilarity metric. Commonly squared intensity differences (SD) or absolute intensity differences (AD) could be used, but the paper by Klaus, Sormann, and Karner presents a self-adapting measure that combines the sum of absolute intensity differences and the gradient based measure below [1]:

$$C_{SAD}(x,y,d) = \sum_{(i,j)\in N(x,y)} I_1(i,j) - I_2(i+d,j)$$

and

$$C_{GRAD}(x,y,d) = \sum_{(i,j)\in N_x(x,y)} |\nabla_x I_1(i,j) - \nabla_x I_2(i+d,j)| + \sum_{(i,j)\in N_y(x,y)} |\nabla_y I_1(i,j) - \nabla_y I_2(i+d,j)|,$$

Weighting is then determined by comparing the right to left disparity map and the left to right map and using a winner-take-all optimization which finds the disparity with the lowest matching cost, resulting in the final metric [1]:

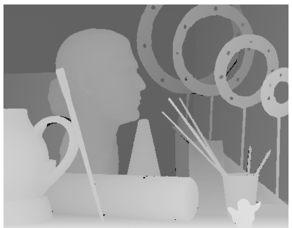
$$C(x,y,d) = (1-\omega)*C_{SAD}(x,y,d) + \omega * C_{GRAD}(x,y,d)$$

The sum of the matching costs of each pixel inside a segment is then used to assign a disparity plane the segment. Lastly, Klaus, Sormann, and Karner propose stereo matching as an energy minimization problem and a new optimal labeling is estimated using Loopy Belief Propagation. Belief propagation is a local message passing technique that solves inference problems on graph models. Since pixels in an image can be thought of as neighboring nodes, belief propagation can also be used make inferences about neighboring pixels in an image.

6. Segment-Based Stereo Matching using Belief Propagation Implementation

Once again, the reference images were read using imread and then preprocessed using rgb2gray and im2double. Color segmentation was then performed using the EDISON [4] mean-shift segmentation function. Next a window based matching technique was used to estimate both the right to left and left to right disparity maps. Then winner-take-all optimization was applied to combine the two previous disparity maps. Finally, the optimized map is used with the segmented image to assign the disparity planes and achieve the final map.

7. Segment-Based Stereo Matching using Belief Propagation Results




Ground Truth Belief Propagation

The resulting image from the segment-based stereo matching method was computed in 12.498 seconds. Much of the speed benefit for this method was achieved through the use of the EDISON mex files for mean-shift color segmentation. Otherwise, the color segmentation tends be the bulk of the computation time. As can be seen above, the resulting depth map is much more well defined that the resulting depth map from the block matching method.

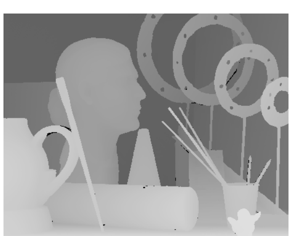
8. Introduction to Cooperative Based Stereo Matching

Cooperative based stereo matching is based on two assumptions initially proposed by Marr and Poggio, uniqueness and continuity. The uniqueness assumption implies that “there can exist only one match within a set of elements that project to the same pixel in an image. [2]” And continuity implies that “neighboring elements have consistent match values. [2]” Unlike Marr and Poggio’s cooperative algorithm, which only produced a 2d array of match values, Zitnick and Kanade propose the use of a 3d array of match values. This opens up the algorithm for use on real life stereo pairs rather than being limited to synthetic random-dot images [2]. The 3d array is used to hold a pixel in the reference image and the estimated match relative to another image. To obtain a smoother map, the function can be iterated to refine the matches. The initial match values can be computed by using the dissimilarity measures discussed previously, such as squared differences. Next, the consistency is improved by averaging neighboring values. In order to do so, a local support area must be determined to calculate the extent in which neighboring pixels should contribute to averaging. Zitnick and Kanade utilize 3D box-shape as their local support area. Sn is then defined as the sum of all match values within the support area. But, due to object overlap and the uniqueness assumption, there is also an inhibition area to take into consideration. Rn denotes the amount of inhibition Sn receives from the set of elements overlapping the current element. This method allows the local area to affect the match values for several iterations while ensuring that details aren’t over-averaged. Also due to the uniqueness assumption, all match values corresponding to occluded pixels are small due to a lack of matches. This allows the occluded pixels to be labeled [2].

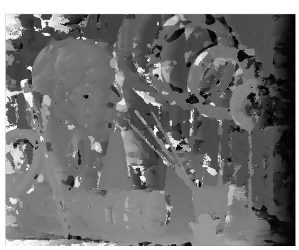
8. Cooperative Based Stereo Matching Implementation

As with the previous methods, the reference images were read using imread and then preprocessed using rgb2gray and im2double. Then the initial disparity values are estimated. Next, the support and inhibition convolution matrices are created using parameters passed to the Cooperative function. For the support area I choose a 7 x 7 x 3 area as recommended by the paper [2]. The inhibition is calculated using the diagonal inhibition function described by Kanade [2]. For the alpha value that affects the power of the inhibition function at each iteration, I found that a value of 1.5 or 2.0 seemed to work best for my test images. The resulting area of support and inhibition values are then used to iteratively average local areas of continuity. Finally, since most areas of occlusion can then be found, I chose to simply set the areas to the max disparity. Unlike the previous methods, this implementation uses drawnow to update the resulting disparity map with each iteration.

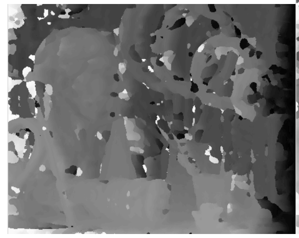
9. Cooperative Based Stereo Matching Results



Ground Truth



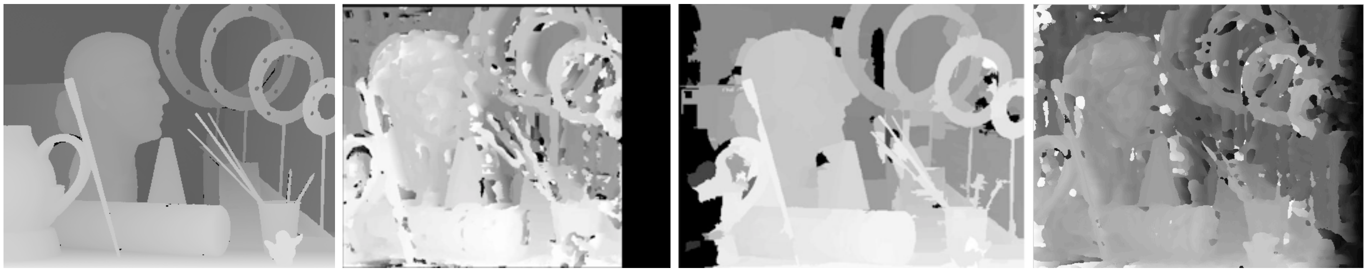
Cooperative iter = 2



Cooperative iter = 3

Cooperative iter = 4

I was slightly disappointed with the results of the implementation of the Cooperative method of stereo matching. As can be seen by the two different iterations above, the iteration is working properly and averaging based on the areas of support and inhibition seem to be working well. However, I believe my disappointment lies with my initial estimate of disparity values. My initial intensions, after the success of color segmentation with the belief propagation method, was to preprocess the image using the same mean-shift segmentation to begin with more consistent regions. I begin my implementation in this fashion and thought I had obtained the desired results; however, I realized that I wasn't properly implementing the 3d support area as described by Zitnick and Kanade was unsuccessful at incorporating color segmentation back into my cooperative based stereo method. Despite my dissatisfaction, cooperative based stereo matching proved to be successful in generating a more consistent disparity map with each iteration. When the method is applied to a better initial estimation of the disparities, it is capable of achieving very good results. However, due to the number of support and inhibition calculations and the need for iteration, the cooperative based stereo matching method proved to be very inefficient. At four iterations and a max disparity of 60, the 370 x 463 test image took 481.73 seconds to complete.



Ground Truth Block Matching Belief Propagation Cooperative

10. Comparison of Methods

The explored methods of estimating depth from stereo pair were all fairly successful at presenting a rough disparity map. The block matching and belief propagation methods were both efficient, with times under a minute for a 370 x 463 test image and a maximum disparity of 70. The cooperative method was the least efficient with a calculation time of 481.73 seconds for four iterations. belief propagation seemed to produce the cleanest disparity map with the clearest areas of continuity. block matching produced a clear depth estimate of the scene, but it was quite noisy primarily due to error caused by occlusion. The cooperative method was very good at smoothing areas of continuity with each iteration, while maintaining areas of detail. If combined with a better form of initial disparity calculation and possibly utilizing color segmentation to preprocess the image, the cooperative method could be quite successful for applications where efficiency is not an issue.

References

[1] A. Klaus, M. Sormann and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. ICPR 2006.

[2] C. Lawrence Zitnick, and Takeo Kanade. A Cooperative Algorithm for Stereo Matching and Occlusion Detection. IEEE 22 May 2000.

[3] Myron Z. Brown, Darius Burschka, and Gregory D. Hager. Advances in Computational Stereo. March 9, 2007.

[4]http://www.caip.rutgers.edu/riul/research/code/EDISON/index.html. November 1, 2009.

[5]Middlebury Stereo. http://vision.middlebury.edu/stereo/. November 1, 2009.