

Analysis of Disparity Map Output from Similarity Measures

Neil Fordyce ^{1,2} and Warren Ryndycz ^{1,3}

¹School of Computing, University of Dundee, Dundee, UK

²n.fordyce@dundee.ac.uk

³w.ryndycz@dundee.ac.uk

Abstract

The analysis of accurate disparity maps from stereoscopic images is an important cornerstone for generating depth maps with applications in both robotic navigation and scene reconstruction. This paper presents an evaluation of popular correspondence measure using a correlation based method for the purpose of generating a dense disparity map from a set of two pre-rectified stereoscopic images. The first correspondence measure presented is that of the sum of square differences of these block regions, followed by the sum of absolute differences algorithm, and finally correlation. Of these various support and search windows are used to determine the optimal trend in sizing which leads to higher quality disparity maps.

Key-words: Stereoscopic Images, Disparity Map, Scene Reconstruction, Correlation based method, Similarity Measures

1 Introduction

Disparity mapping forms a critical and necessary step for providing accurate spatial information of any scene from a set of rectified images. From this step depth maps of a scene may be computed allowing for accurate three-dimensional reconstruction models to be developed with applications in robotic navigation[3] and scene mapping[4]. Disparity vectors are computed from two rectified images of the same scene where each pixel of the same real world point is offset by a specific distance d . This offset distance allows for the computation of the depth of each pixel from the observer, using the fact that depth is inversely proportional to disparity. Visually this is represented as brighter pixel intensities being closer in distance while darker areas being further away in final disparity maps. The methodology of using stereoscopic image analysis for the determination of the depth of an object from an observer is inspired by the stereoscopic visual system of most animals, however these systems makes use of different visual processing strategies employed by each animal's brain which do not necessarily have relation to a relative of the same genus. Due to the complexity of the visual system

of each animal's brain it is impossible using current scientific analysis to know precisely the method used in computing depth in a robust and practical manner. Thus correspondence measures which determine the distance between pixels in images using the methods described in this paper provide only an estimation at best.

In this paper a correlation based method (CBM), similar to the block matching algorithm used in motion estimation, is used to determine experimentally the accuracy of various correspondence measures for disparity vector calculation. These measures include the sum of square differences (SSD) and the sum of absolute differences (SAD). Other factors which are investigated and which determine the accuracy of the resulting disparity map include the size of support and search windows, which correspond to the neighbourhood around a pixel being searched from and the size of region in the other image that is being scanned for correspondence. Various sizes of each are investigated and their results displayed, with analysis of both the scene objects location and structure being used to gauge the effectiveness of each correspondence measure.

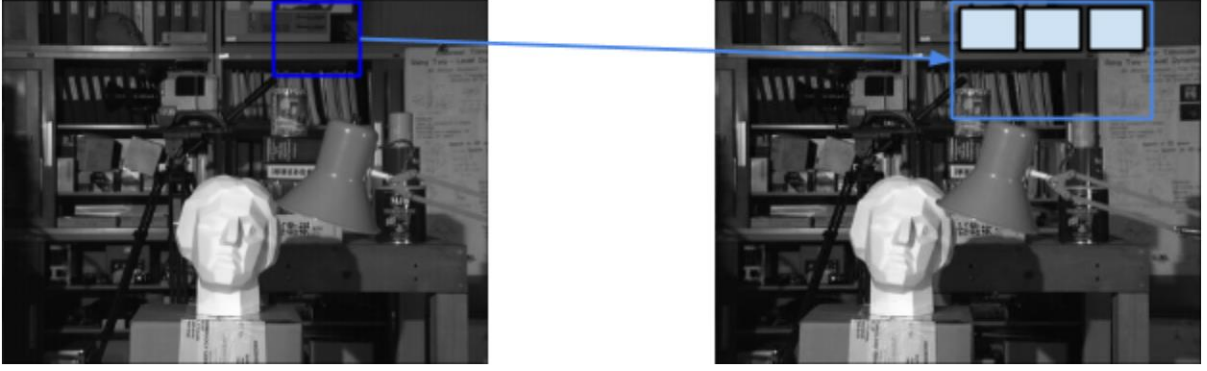


Figure 1 - The support window from the left hand image is compared to regions of the same size along the right hand image's search window.

2 Method

Our system is tested on a provided set of stereoscopic images which are assumed to be rectified. Using these two images as input our correlation based method takes a pre-specified region for both the support (left image) and support window¹ (right image) (In this project a 5x5 support window combined with 15x25, 21x31 and 35x45 search window are used) comparing each pixel and its associated region with that of the entire search window. Our system uses the intuition that rectified images will have corresponding pixels which are aligned horizontally, thus we use a rectangular search window which decreases the run time necessary for our system by limiting the search vertically. Our system is optimised using the intuition that for a subwindow in the left hand image the corresponding subwindow will be to the left in the right hand image. Thus we constrain the search to a single direction on the horizontal axis. This reduces the cost of run time for each image processed and was essential in keeping this cost down during the testing phase. Each image is padded with zeroes before disparity is calculated to allow the support window to centre on the middle pixel of a region without going out of bounds. For each support window location and it's comparison to a region of the search window a correspondence measure is calculated. This process of exhaustive correlation searching is repeated for every pixel within in the support window. It is

from these correspondence measures (SSD, SAD) that the corresponding pixel in the search window is found; the vector between these two pixel locations being the disparity vector for that pixel from which a disparity map is calculated.

The sum of squared distance¹ provides the square difference between the pixel intensity of two regions where a value zero indicates that the two regions are the same and thus an ideal match. For practical applications this value will never be zero and thus the minimum result of SSD provides the corresponding pixel with the highest confidence of being appropriate.

$$SSD = \sum_{i=1}^n (X_i - Y_i)^2$$

Equation 1 - Sum of square differences where X and Y are image matrices.

The sum of absolute differences² takes the absolute difference of each image matrix being examined both this and SSD provide benefits over correlation due to their insensitivity to high values. With SAD the minimum result between two image matrices provides the highest confidence that two pixels are corresponding to one another.

$$SAD = |X - Y|$$

Equation 2 - Sum of absolute difference, where X and Y are image matrices.

Both of the above correspondence metrics are however limited by changes in lighting in both images even if they are rectified, most notably specular highlights. Due to the image set being used to test our system however the lighting did not change between image shots so this limitation was

untested, but is of relevance to practical applications.

From these minimums a matrix can be formed of disparity values that when displayed as an image show a disparity map of the completed scene. This disparity map culls regions that are not visible in both images through padding which allows for an accurate reconstruction.

3 Dynamic Support Window

As discussed in [1], appropriate selection of the support window size parameter has a dramatic impact on the performance of the system. If the support window is too large, then the disparity map appears blurred and the edges are not captured accurately. Furthermore, it is more expensive to compute disparities for a larger support window. If the support window is too small, then the disparity map captures the edges well but the support window will not contain enough information to perform well

on almost solid textures. The resulting disparity map would contain a lot of noise, incorrectly indicating depth variation on flat surfaces. Due to the varying textures in real world images, it is therefore necessary to resize the support window dynamically, altering the size in response to the texture surrounding the current pixel.

Our system implements a dynamic support window by analysing the gradient image of the current support window. If the mean gradient in the support window is below a gradient threshold then we increase the support window size by two pixels. This is repeated until the mean gradient in the image is at least as large as the gradient threshold, or until an upper bound support window size is reached. The gradient threshold used was determined experimentally. Our method is simple, although effective at reducing noise from the output disparity maps while maintaining edge accuracy.

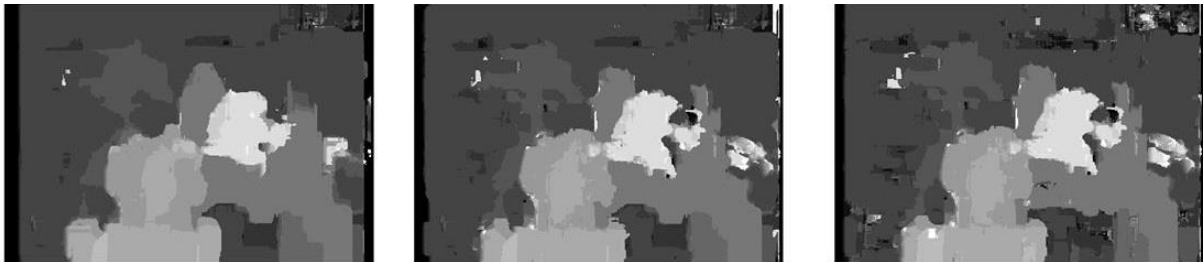


Figure 2 - (Left) output using a fixed 21x21 support window, (middle) output using a dynamically resizing support window in the range of 9x9 to 21x21, (right) output using fixed 9x9 support window. All using a fixed 31x5 search window and no post-processing filters.

4 Results

Disparity map output from using SSD 3 shows some erroneous areas especially around areas of sudden depth change, such as that of the table and the background. With SSD the best results with the takubura image pair set takes place with a search window between 20 to 25x28 to 33 in height and width respectively. Too small a search window and precision is lost which is displayed as artefacts on the disparity map. Too big a search window and objects closer to the observer (head bust) appear darker than objects behind themselves (lamp).

5 Discussion

This paper has explored the difference between generating the disparity vectors that form a disparity map which can then be used for future work developing a depth map. The results of SSD and SAD provide much more legible output than using correlation Figure 5. However between SSD and SAD there is little variation in the outputted results. Further findings found that an optimal fixed size search window must be at least as big as disparity between the closest points in corresponding stereo images. If it is smaller, the correct disparity



Figure 3 - (Left) Output using a 7x7 median filter on a 6x6 support window and a 21x31 search window. (Middle) Output using a 7x7 median filter on a 5x5 support window and a 15x25 search window. (Right) Output using a 7x7 median filter on a 35x45 search window.



Figure 4 - (Left) Output using a 7x7 median filter on a 6x6 support window and a 21x31 search window. (Middle) Output using a 7x7 median filter on a 5x5 support window and a 15x25 search window. (Right) Output using a 7x7 median filter on a 35x45 search window.



Figure 5 - Output using a 5x5 support window and a 15x25 search window using a correlation algorithm.

cannot be found because the corresponding support window is never examined. On the other hand, if the search window too big, then the disparity reported could be much higher than any sensible value. Furthermore, it would be computationally more expensive. We selected the size of our search windows by manually examining the images and approximating the pixel wise distance between the closest objects. As discussed in [1] it would be beneficial to provide a mechanism for dynamically resizing the search window.

All of the images in the report have taken between 56 and 468 seconds to compute depending on the search window

and support window sizes. This could be greatly improved if optimizations were included to exploit the repeated calculations by storing the result of correlation computations on the previous support window at each iteration, as shown in [5].

6 References

- [1] H.-S. Koo and C.-S. Jeong, 'An Area-Based Stereo Matching Using Adaptive Search Range and Window Size', in *Computational Science - ICCS 2001*, V. N. Alexandrov, J. J. Dongarra, B. A. Juliano, R. S. Renner, and C. J. K. Tan, Eds. Springer Berlin Heidelberg, 2001, pp. 44–53.
- [2] Barjatya, Aroh. "Block matching algorithms for motion estimation." *IEEE Transactions Evolution Computation* 8.3 (2004): 225-239.
- [3] C. Hane, C. Zach, J. Lim, A. Ranganathan, and M. Pollefeys, 'Stereo depth map fusion for robot navigation', in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 1618–1625.
- [4] Slabaugh, Greg, et al. "A survey of methods for volumetric scene reconstruction from photographs." *Proceedings of the 2001 Eurographics conference on Volume Graphics*. Eurographics Association, 2001.
- [5] K. Muhlmann, D. Maier, J. Hesser, and R. Männer, 'Calculating dense

disparity maps from color stereo images, an efficient implementation', in *IEEE Workshop on Stereo and Multi-Baseline Vision, 2001. (SMBV 2001). Proceedings*, 2001, pp. 30–36.