A Comparison of Image Data Representations for Local Stereo Matching

André Smith, Amr Abdel-Dayem

Abstract**—**The stereo matching problem, while having been present for several decades, continues to be an active area of research. The goal of this research is to find correspondences between elements found in a set of stereoscopic images. With these pairings, it is possible to infer the distance of objects within a scene, relative to the observer. Advancements in this field have led to experimentations with various techniques, from graph-cut energy minimization to artificial neural networks. At the basis of these techniques is a cost function, which is used to evaluate the likelihood of a particular match between points in each image. While at its core, the cost is based on comparing the image pixel data; there is a general lack of consistency as to what image data representation to use. This paper presents an experimental analysis to compare the effectiveness of more common image data representations. The goal is to determine the effectiveness of these data representations to reduce the cost for the correct correspondence relative to other possible matches.

Keywords**—**Colour data, local stereo matching, stereo correspondence, disparity map.

I.Introduction

THE stereo matching problem has been an active research area for decades. It finds its origins with Marr and Poggio in 1979 [1]. The goal of their research was to develop a recreation of the human visual correspondence system, computationally. This would allow software to reproduce a person's ability to determine the distance separating themselves from visible objects. The approach used to measure depth attempts to match elements visible in both views (i.e. left and right eye). The main challenge faced to perform this matching is that of ambiguity. When multiple elements are considered to be visually similar, it becomes difficult to determine the correct correspondences between them. In an attempt to resolve this, constraints are put in place to reduce the possible matches. The first constraint put in place is that of uniqueness, where every point within the image can be attributed only a single distance value, and therefore a single match. The second assumption is that many regions within a scene are smooth, meaning the distance of continuous surfaces are expected to be similar. With this research, the stereo matching problem took flight, and quickly found its way into the software domain [1].

When taking the theory into practice, certain terms change slightly. The left and right perspectives of human vision are represented as a pair of stereo images. The points being compared are taken as pixels. Instead of measuring depth, as a real-world distance, a disparity measure is taken, representing the pixel separation between both images. The first software implementations to tackle the stereo matching problem are known as area-based approaches [2]. These directly compare pixels, windows or segments in order to determine correspondences. While these methods had certain advantages, they were often prone to errors due to calibration inconsistencies. Factors such as camera positioning and tilt can greatly affect the accuracy of the results. To circumvent this problem, feature-based methods were introduced [3]. Here, features such as objects are first identified, and then matching is performed between these. As the positioning of objects are not expected to be perfectly aligned between the images, calibration errors are avoided. The downside to this approach is that it is highly dependent on both the effectiveness and performance of the feature identification algorithm utilized.

As area-based matching methods have many implementation advantages in their simplicity and low computational complexity, a calibration correction technique is introduced to the problem, known as rectification [4]. As, even in the best circumstances, perfect hardware calibration is nearly impossible to achieve, rectification is a software solution to make adjustments. This can provide compensations for camera position, tilt, and in some cases lens distortions. Without the issue of calibration, new life is brought to research using area-based matching. As a result of this, a survey in 2002 [5] provided classification for area-based stereo correspondence techniques. All methods reviewed are expected to utilize rectified images, and only require a pair of stereo images to generate disparity maps. To classify the proposed algorithms, 4 components are identified of which they comprise. The first of these is a matching cost function. This function (e.g. absolute difference) creates a measure for the similarity of individual pixels. The second component is the support or aggregation method. This represents the manner in which the cost function is applied, such as over a rectangular region of the images. This will result in a cost-volume, where a minimization can then be applied. Local matching techniques typically use a winner-take-all minimization, where the match with the lowest cost is chosen on a per-pixel basis (see Fig. 1). The last step, which is entirely optional, is a refinement, such as error correction and sub-pixel refinement.

André Smith is with the Department of Mathematics and Computer Science, Laurentian University, Sudbury, Ontario, Canada (e-mail: aw_smith@laurentian.ca).

Amr Abdel-Dayem is with the Department of Mathematics and Computer Science, Laurentian University, Sudbury, Ontario, Canada (phone: +1 705-675-1151 extension (2396); fax: +1–705–673-6591; e-mail: aabdeldayem@lcs.laurentian.ca).

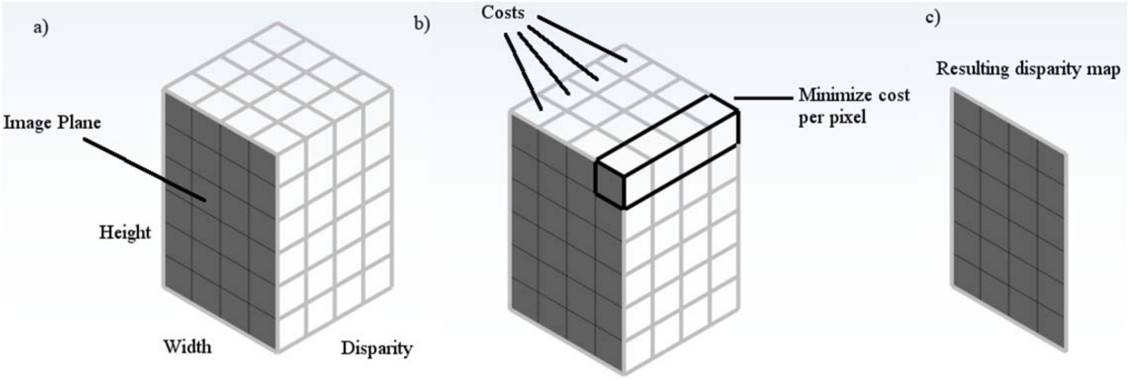

Fig. 1 A visualization of the cost minimization process. a) The image plane represents an image, where each pixel is one square of the grid. With each pixel, multiple candidate disparity values are associated, creating a 3D volume. b) The cost volume is populated, where each pixel has a set of costs associated with every disparity level. From here, the minimization is applied, such as the winner-take-all method, where the minimum cost per pixel is chosen. c) The resulting map contains the disparity value, per pixel of the original image, that had the lowest cost.

Research into stereo correspondence remains active, despite the advancements rectification have allowed. There are still several factors that result in inconsistencies between images, and have less elegant solutions. One such problem, though minor, is noise. Small random variances will result in minute differences between both images. Typically it is corrected using image filters, such as mean filters, however these too have their strengths and weaknesses. A more notable source of inconsistent data is the result of non-Lambertian surfaces [6]. These are regions within a scene that have some reflective properties, such as glass or glossy surfaces. Since these have different appearances depending on viewing angle, measures to match brightness or colours are often ineffective. Alternative matching methods attempt to work around this, where emphasis may be placed on the position, shape or surroundings, as opposed to placing focus on the data directly [7], [8]. Finally, one source of mismatching that has no exact solution is that of occlusion. Due to the change in perspective between left and right images, some segments visible in one image will be hidden in the other, and vice-versa. As the goal of stereo matching is to pair equivalent points between the images, this is not possible to achieve when the point is only visible in one of the images. Solutions to work around this include explicit identification of occluded segments [9], [10], assuming they occur around edges, and interpolation techniques to fill these in with estimated disparity values [11].

II. Research Goal

As the main challenges to stereo matching are well identified, researchers have experimented with many variations and approaches to create their algorithms. One area of inconsistency is the image data representation utilized during the matching process. Older publications, mostly prior to the year 2000, utilize intensity values (i.e. greyscale). After this early generation, colour images (e.g. RGB channels) have been utilized for this process as well. More recently still, researchers have proposed transformations to the existing data [12], [13], though there does not appear to be any consensus on which way the data should be presented for this purpose. Researchers often attempt to introduce new approaches, as opposed to testing or improving upon previous work. Despite such variation, there appears to be seldom any research to demonstrate advantages of any data representation over another, as authors generally focus on their complete stereo matching techniques, and observed results in terms of the accuracy of disparity maps generated. Greyscale image data was used well into the appearance of colour digital images, though it is unclear if this is due to hardware and software limitations, traditions, or the quality of existing results. More recent articles often neglect to explain their choice to use colour image data, or lack thereof. For these reasons, an investigation into the basics of stereo matching is required, and demonstrated here.

As previously described, local stereo matching can be broken down into a few steps. The first of these is a cost function with which to evaluate the likelihood of a correspondence between points within an image set. In related work, Hirschmüller et al. [6] have performed an analysis comparing the effectiveness of different cost functions. The second step, the function chosen is applied to all possible match combinations (or a subset of these) to generate a cost volume. A minimization is then applied (e.g. winner-take-all) to determine the most likely match for each pixel. From this, disparity maps can be generated for each image, which represent the distance separating the matched pixels. This map is inversely proportional to a depth map, as the further the pixels are separated, the closer these points are considered to be relative to the observer. This is similar to how, visually, if an object is closer to an observer, it appears larger; its extremities appear further apart the closer it is to the subject. The accuracy of the matching algorithm is determined by comparing the generated disparity map to one that is 'Ground Truth' (GT), determined by other techniques applied when capturing and generating the images. A match is considered to be correct if the disparity value is within 1 pixel of the GT, or 0.5 if sub-pixel accuracy is used.

III. Method

Instead of comparing cost functions, different data representations will be evaluated using the same function, as this will maintain consistency between the results. While the accuracy of a depth map generated is an important measure, it is not sufficient to determine which data representation is best suited for the matching process. When the minimization step is applied, the majority of the generated data is discarded, much of which may be useful for analysis. As an alternative to traditional measures, the results from the cost function will be evaluated directly, before any minimizations are applied. The accuracy of a match is determined by the percentage of matches whose costs are below that of the expected correspondence. Since it is typical to select the match with the minimal cost, this accuracy measure will indicate the degree of error for a particular match. Along with this error, there is also an ambiguity factor. With matches of an equal cost to the correct correspondence, it is not directly possible to distinguish between the correct match and similar ones. As such, ambiguous matches represent the percentage of matches whose costs are equal to the correct correspondence. To measure the effectiveness of different data formats, the error (better matches), ambiguity (equal matches) and the sum of these will be compared.

For simplicity, and to reduce the data to compare, only rectified images will be used for the analysis. This will allow comparisons between pixels to be restricted to those on the same scan line (i.e. row of pixels). Furthermore, three data formats will be considered. The first of these is greyscale, since it is most commonly utilized in older work, and is equally found in some newer research [11]. Next is RGB data representation. These will be considered both on a per-channel basis, as well as through a linear combination. Finally, the YIQ colour space is considered. This representation is a linear transformation of RGB data, and will also be tested both separate and combined. This representation may prove to be useful since the Y channel represents intensity values, similarly to greyscale. The difference between these two is the numerical precision, as greyscale is typically restricted to a single byte of data, while Y is not.

The cost function to be used is the absolute difference between values per pixel. In the case of combinations (i.e. RGB and YIQ), the sum of absolute differences will be tested. The sample images used for testing are taken from the Middlebury stereo image database [14]. All of the images from each datasets will be considered, with exceptions for those with only greyscale data provided and those without GT data available. Images that have distortions in terms of lighting and exposure will also not be considered. Since all of these images are said to be rectified, matches on the same scan lines will be considered. To perform the analysis, the following steps will be used.

1. In each row, calculate the absolute difference between each pixel in the right image to all pixels in the left, in the same row. Consider the following expression: $V_{rcd} = |L_{rd} + R_{rc}|$, where L and R represent the left and right images, r and c represent the row and column within the images, d the disparity, and V the resulting cost volume.
   - In the case of multi-channel data, the sum of these differences is used. A modified expression can be used: $V_{rcd} = \sum_{i=1}^{h} |L(i)_{rd} + R(i)_{rc}|$
   - With YIQ, each channel is normalized before being tested. This is not required for RGB since the channels are of the same scale.
2. Count the number of pixels in each row who's cost is lower or equal to the expected match, taken from GT data, as represented with: $E_{rc} = \sum_{i=1}^{d} [V_{rcd} < GT_{rc}]$
3. Store the average of these per row, per image

As an example, consider the image set barn1 from the original 2001 dataset available on the Middlebury website (see Fig. 2). Suppose a 15-pixel segment of the images are extracted, with which comparisons will be performed (see Table I). Each of the rows here demonstrates the numerical values (from 0 to 255) for the Red channel of a small segment of the images. In this case, the pixel in question to match in the right image has the value of 213 (highlighted). The goal is to match this point to the most similar point within the other image. As the GT is already known, the expected disparity (distance between matches) in this instance is of 11.6 pixels, demonstrated in Fig. 1, as sub-pixel data is available. Since the distance falls between 11 and 12, both pixels with values 220 and 200 (also highlighted) can be considered acceptable matches. To determine the likelihood, or cost, of possible matches, the absolute difference is calculated between the pixel in question of the right image with all possibilities in the left image. Table II represents the results of these calculations. The costs of the expected matches are 7 and 13 respectively. To measure the error of this match, the costs are placed in ascending order (see Table III). The error is represented by the number of pixels whose costs are lower than the expected match. As in this case both possible matches have lower costs than all other matches, the optimal result is obtained here, as the effective error is 0%. This would suggest the correct match would be selected when choosing the minimal cost.

While the optimal case is ideal, it is generally not obtained. Consider now the same segment of pixels, but using the Blue channel instead of Red (see Table IV). In this instance, the goal is to match the pixel with value 92, and the expected matches are either 130 or 89. Table V depicts the ascending absolute differences for these potential matches. Unlike the previous example, this one demonstrates both error concepts. The first point of interest is the differences calculated. While one of the expected matches has a low cost of 3, which is below even that of the minimum in the previous example, the other possibility is much higher at 38, which is significantly higher than many other matches. For this reason, only the lowest of these costs is considered. Since the minimum cost of 3 is higher than two of the other matches (i.e. values below 3), the error is effectively 13% (2/15 pixels). There is however also an ambiguity here. As there are two matches with costs of 3, only one of which is correct, it is not possible using single cost values alone to determine any preference between these two matches. This adds an ambiguity factor of 6% (1/15 pixels). The sum of these, for a global error measure, is 20% (3/15 pixels). While this may seem quite large, a 15-pixel sample is too low to accurately demonstrate realistic errors.

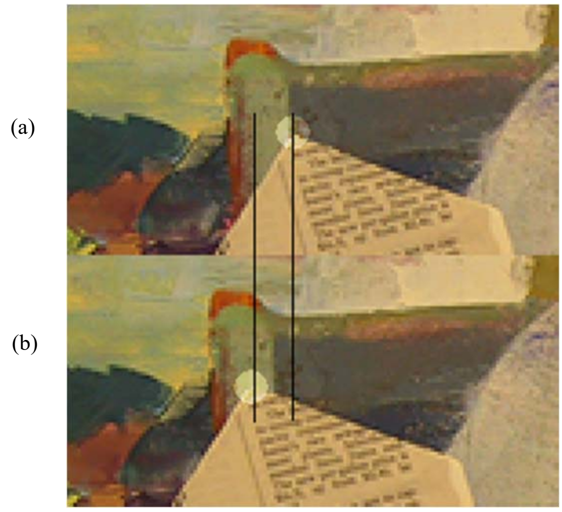

Fig. 2 Comparison between left image (a) and right image (b) of the barn1 set, demonstrating the separation between identical points: The distance between both lines is 11 to 12 pixels

IV. Experiments and Results

This process is performed using all candidate images from the Middlebury database, and allows a row-by-row comparison of the effectiveness for each matching method. Testing is performed on a total of 82 image pairs, which includes both perfect and imperfect calibration sets from 2014. Analysis was performed using MATLAB, on a PC with 8GB of RAM, an Intel Core i5-2400 clocked at 3.1GHz, and an Nvidia GeForce GTX 750 Ti GPU, which has 2GB of VRAM. One thing to note is that pixel-wise matching alone is not an effective method to generate a disparity map, since factors such as ambiguity and calibration errors reduce the likelihood that the minimum cost be the correct match in most cases. Pixel-wise matching is tested since this is generally the underlying fundamental calculation used. The evaluation of the cost volume, rather than the minimum of costs, is done to understand which factors reduce the likelihood of incorrect matches being taken into consideration. In order to avoid any bias, all possible matches are considered in each row, since disparity limits are generally determined by the image capture technique used, as well as any rectification process done thereafter. As the error values are dependent on the width of images, the results are converted into percentages for comparison purposes. This is particularly important since the older images from the database are smaller, having widths less than 1000 pixels. By contrast, the newest images are larger than 2000 pixels wide.

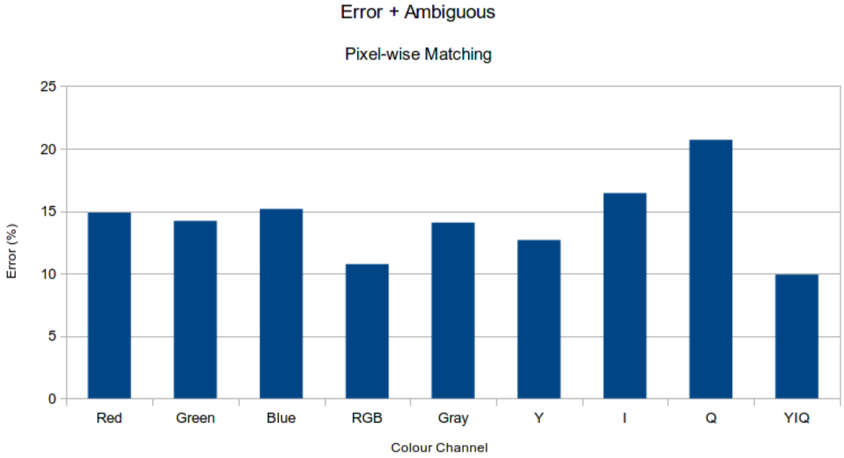

Fig. 3 Results for the average total of error and ambiguity values for matches. These represent the percentage of pixels whose matches have a lower cost than the expected match

As an initial analysis of the results, the average errors, ambiguity and sums of each image are taken. Welch's T-test is used to compare these, with a 95% certainty, since the variances between them are not assumed to be equal. From this a few conclusions can be made. Firstly, there is no statistically significant difference between using Red, Green, Blue or Greyscale on their own (see Fig. 3). Secondly, the I and Q channels have some difference, but since in most cases they have higher averages than the rest, these are considered to be less effective. In many cases, there is a significant difference between these representations, in comparison with Y, RGB and YIQ. Since these have lower error values, a deeper analysis is done to determine the degree of effectiveness. To compare these three, more T-tests are applied per image, instead of the averages per image. In almost all cases, Y is less effective than both RGB and YIQ (see Fig. 4). The exception to this is in terms of the ambiguity, where Y is more effective than RGB, though has no significant difference with YIQ. Between RGB and YIQ, the differences were less obvious. In terms of the error, some cases favoured each, and some cases favoured neither. A Chi-squared proportion test indicates that there is some favouring to YIQ. In terms of ambiguity, YIQ is the clear winner, being significantly lower to RGB in all cases. Finally, the sum of these also favours YIQ in nearly all cases. With this, YIQ takes the lead as the image data representation resulting with the least likelihood to favour incorrect matches.

Since pixel-wise matching is never used on its own, these tests are repeated while matching regions as well. Square windows with sizes of 3, 5 and 7 pixels are used, with Gaussian weights to favour the center pixel (see Fig. 5). To reduce the runtime of this analysis, a Gaussian filter with the window's size is applied to the matrices of differences, as the end result is equivalent. The results for these, much like the single pixel matching, also favour YIQ, both for just the error, as well as the sum values (see Fig. 6). The ambiguity differs with each filter size, and while favouring YIQ with a size 3 filter, shows no significant difference with larger filter sizes. This suggests that, with more complex filtering, the ambiguity becomes insignificant.

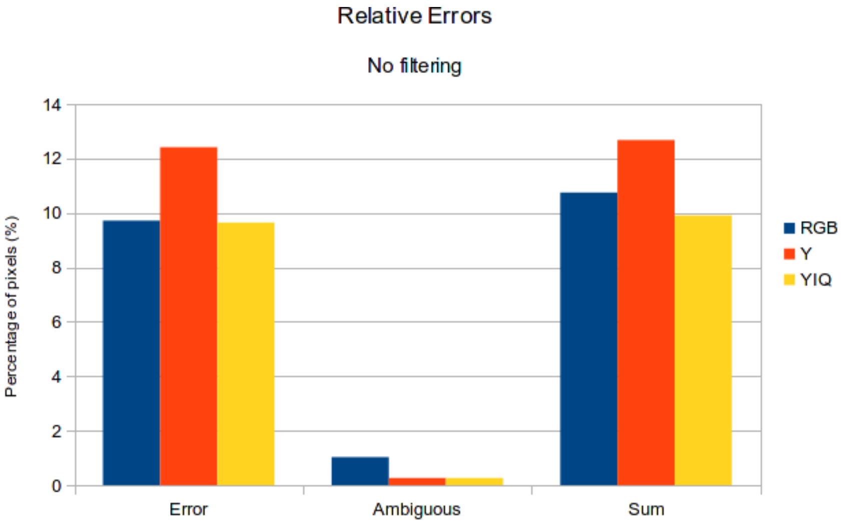

Fig. 4 Comparison of the relative errors for RGB, Y, and YIQ, without filtering

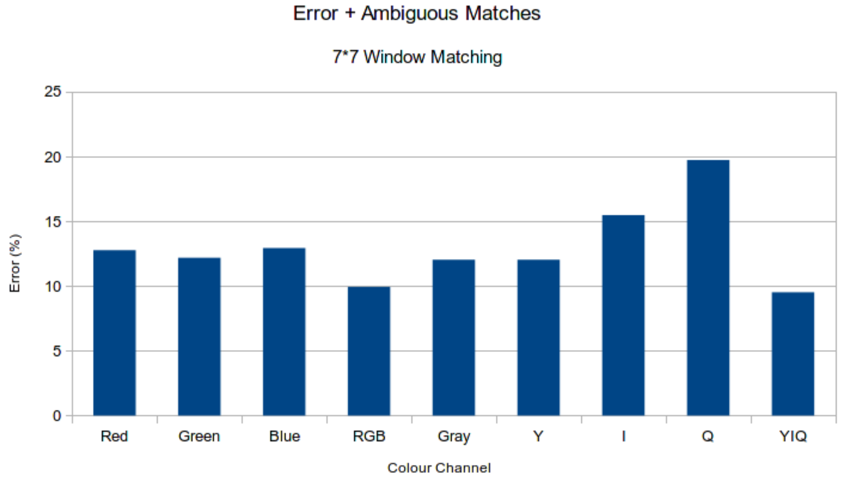

Fig. 5 Results for the average total error and ambiguity values for matches, comparing regions with a 7*7 window

One supplemental test compares YIQ to CEI Lab space, since this one is used in [15]. The largest difference between Lab spaces and others presented here is the size of the colour space. This one has a theoretical infinite number of colours it can present, though not all are visible to humans. The <L> component represents a light value, between 0 and 100, and the <a> and <b> components are for colour. These both have no boundaries, but limits are placed on them to restrict the possible values. In the case for this analysis, since conversions are done from RGB to CEI Lab, these limits are set based on full possible range of values obtained. Furthermore, the channels are normalized based on these limits. As demonstrated in Fig. 7, there is no significant difference between YIQ and CEI Lab space, for the purpose of stereo matching. It's likely this space would only be beneficial if a wider range of colour information could be utilized.

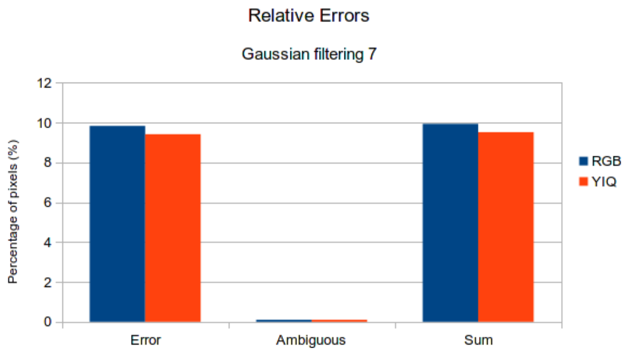

Fig. 6 Comparison of RGB and YIQ, with filtering

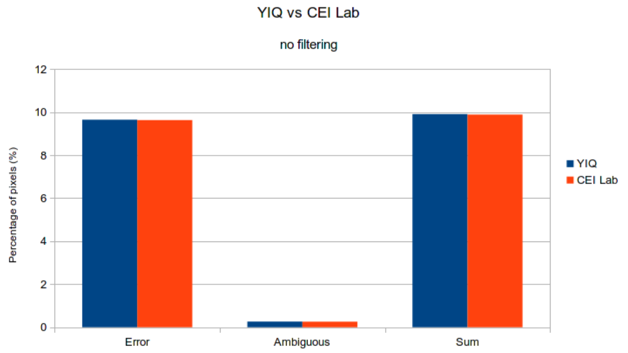

Fig. 7 Comparison of the relative errors between YIQ and CEI Lab colour spaces, without filtering

Overall, it would seem YIQ is the preferable data representation to use to reduce the likelihood that incorrect matches will be preferred over the correct match. This however must be put into perspective. The majority of methods result in average errors of 12-13%. This means that, relative to the width of the image, approximately 1/8th of the pixels are considered to be matches equal or better to the correct correspondence, on average. With RGB and YIQ, this is reduced to less than 10%. The difference between these two methods, while considered to be statistically significant, averages to a 0.4-0.5% difference, favouring YIQ. For the more recent images in the Middlebury dataset, this is equivalent to roughly a 10-pixel difference. While this may be significant for larger images, smaller ones who have widths less than 500 pixels will likely not see any notable benefits between the usage of RGB and YIQ for correspondence, as the difference is too low. Regardless, both of these still provide reduced error rates over their counterparts.

One other notable observation is in reference to the differences with the ambiguous matches. While with direct pixel matching, these values can be fairly significant; this is not the case once regions are taken. With all window sizes, in most cases, matches have on average little to no ambiguity. This is likely due to the increase in numerical precision that results from the summation of more data points than with single pixel matching. This is particularly notable when comparing greyscale and the Y channel. Although direct pixel-wise matching shows an improvement for Y, there does not appear to be any difference between the two in the latter case. Equally, while there is still some favouring towards the ambiguity of YIQ with the window of size 3, this is no longer present with larger window sizes.

V. Conclusion

While there have been a few data representations used for stereo image correspondence, notably greyscale and RGB, there has been little analysis presented to demonstrate the effectiveness of one over the other, or in comparison with other representations such as YIQ. When comparing the cost volumes generated for these, with cases such as direct pixel-wise matching, and using Gaussian weighted windows, it becomes clear that there is an advantage of RGB over greyscale, and a potential small improvement of YIQ over these. Greyscale matches have roughly 3% more pixels whose matches are more or equally similar to the expected correspondence, in comparison with RGB and YIQ. This would suggest, for an image 2000 pixels wide, a single pixel match would have approximately 60 more pixels considered to be better matches when using greyscale. Furthermore, when YIQ is utilized over RGB, an additional 10 pixels may be eliminated. This improvement is likely the result of the increased numerical precision gained when converting standard RGB to YIQ. It is also debatable as to whether or not this benefit is significant enough to merit being used, should it be applied with modern local stereo matching approaches. There is also some potential loss in performance, as there must be data conversion between RGB and YIQ, as well as more memory utilization since each channel in RGB requires a single byte of memory, whereas YIQ (if using double-precision arithmetic) requires four bytes per channel. Further research may explore advantages of yet other image data representations, the impact of different cost functions, and performance comparisons

Appendix

TABLE I
Sample Sequence of Red Pixel Values from the Barn1 Set

Red Left	131	125	123	143	139	138	139	141	149	144	143	220	200	171	184
Red Right	213	170	174	189	189	210	212	217	178	118	95	95	71	77	81

TABLE II
Absolute Differences between Search Pixel and Left Row in Table I

82	88	90	70	74	75	74	72	64	69	70	7	13	42	29

TABLE III
Values from Table II Placed in Ascending Order

7	13	29	42	64	69	70	70	72	74	74	75	82	88	90	122

TABLE IV
Sample Sequence of Blue Pixel Values from the barn1 Set

Blue Left	67	60	71	84	73	85	90	82	84	82	90	130	89	75	89
Blue Right	92	66	80	97	101	116	104	112	88	50	47	56	40	50	63

TABLE V
Ascending Absolute Differences between Search Pixel and Left Row in Table IV

2	2	3	3	7	8	8	10	10	17	19	21	25	32	38

References

[1] D. Marr, and T. Poggio. "A computational theory of human stereo vision." Proceedings of the Royal Society of London B: Biological Sciences 204.1156 (1979): 301-328.

[2] U. R. Dhond, and J. K. Aggarwal, "Structure from stereo-a review." IEEE transactions on systems, man, and cybernetics 19.6 (1989): 1489-1510.

[3] W. Grimson, “Computational Experiments with a Feature Based Stereo Algorithm”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.PAMI-7, No: 1, pp.17 - 34, Jan. 1985.

[4] A. Fusiello, E. Trucco, A. Verri, "Rectification with unconstrained stereo geometry." BMVC. 1997.

[5] D Scharstein, R Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms." International journal of computer vision 47.1-3 (2002): 7-42.

[6] Hirschmüller, Heiko, and Daniel Scharstein. "Evaluation of cost functions for stereo matching." Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE, 2007.

[7] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information." Pattern Analysis and Machine Intelligence, IEEE Transactions on 30.2 (2008): 328-341.

[8] Kim, Junhwan, Vladimir Kolmogorov, and Ramin Zabih. "Visual correspondence using energy minimization and mutual information." Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. IEEE, 2003.

[9] Zitnick, C. Lawrence, and Takeo Kanade. "A cooperative algorithm for stereo matching and occlusion detection." Pattern Analysis and Machine Intelligence, IEEE Transactions on 22.7 (2000): 675-684.

[10] S. Birchfield, C. Tomasi. "Depth discontinuities by pixel-to-pixel stereo." International Journal of Computer Vision 35.3 (1999): 269-293.

[11] Žbontar, Jure, and Yann LeCun. "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches." arXiv preprint arXiv:1510.05970 (2015).

[12] C. C. Pham, J. W. Jeon, "Domain transformation-based efficient cost aggregation for local stereo matching." Circuits and Systems for Video Technology, IEEE Transactions on 23.7 (2013): 1119-1130.

[13] Q. Yang, "A non-local cost aggregation method for stereo matching." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.

[14] http://vision.middlebury.edu/stereo/data

[15] De-Maeztu, Leonardo, Arantxa Villanueva, and Rafael Cabeza. "Stereo matching using gradient similarity and locally adaptive support-weight." Pattern Recognition Letters 32.13 (2011): 1643-1651.