Polar Rectification of Stereo Images Implemented on a GPU

A. L. Nel

***Abstract*—** Polar rectification of stereo image pairs is reviewed and the implementation on a graphics processing unit (GPU) discussed. The rectification process requires the fundamental matrix as the only parameter and its performance will be tested using images with varying SNR. The computational time for the process as implemented will also be discussed.

***Keywords*—** Polar rectification; Image processing; Graphics Processing Unit (GPU)

I. Introduction

THE matching correspondence problem, the problem of matching like pixels from one stereo image to its pair image, has been a pressing issue for decades in the field of computer vision [1,2]. Many algorithms have been proposed to solve this problem [3, 4, 5, 6, 7, 8, 9]. These algorithms use amongst others segmentation based as well as graph cut methods. The computational time required for any of the abovementioned algorithms is however substantial and this has resulted in the solutions not being feasible on standard computers for real-time applications.

In order to achieve a depth map for a stereo image pair, one has to match the pixels in one image to the correct (i.e. same) one in the other image. Two of the main problems that are faced when attempting to match pixels are the large search areas and the impact of image noise. Polar rectification has been used to help decrease the search area to a single line which in turn decreases the computational time required to perform the subsequent depth map calculations on the images.

II. Background

A. Stereo image pairs

A stereo image pair is two images of the same scene taken simultaneously from two different camera positions. In the case of this research the pair of identical cameras were mounted in the same flat horizontal plane and aligned with their focal lines parallel to each other with no vertical tilt.

B. The fundamental matrix

The fundamental matrix is a unique 3 x 3 rank 2 homogeneous matrix which satisfies (1) for all corresponding points PR ↔ PL [10]. In order to calculate the fundamental matrix one requires a minimum of seven point matches between the two images of the stereo pair [11]. The more point matches the better and having these matches spread over the entire image helps improve the estimate of the fundamental matrix [12]. The fundamental matrix must satisfy

$$P_L^T F P_R = 0 \tag{1}$$

Where P_L and P_R are the points in the left and right images respectively and F is the fundamental matrix. These are represented as indicated below.

$$P_L = \begin{bmatrix} x_L \\ y_L \\ 1 \end{bmatrix} \qquad F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \qquad P_R = \begin{bmatrix} x_R \\ y_R \\ 1 \end{bmatrix}$$

In order to obtain the matching line in the other image from the fundamental matrix one calculates either of the following depending on the choice of basis image.

$$l_R = F^T P_L \tag{2}$$

$$l_L = F P_R \tag{3}$$

Where lR and lL are the lines known as epipolar lines. The epipolar lines converge to a single point known as the epipole of the image. Once the epipole has been located one can warp (rectify) the image in order to have all epipolar lines horizontal. This makes searching for matches required in the stereo matching problem simpler as only points on the same horizontal lines need to be considered as potential matches.

C. The Vertex on the GPU

A vertex in the GPU context is a node placed in an image. These vertices can be arranged in a variety of ways including QUADS, TRIANGLE, TRIANGLE_FAN, TRIANGLE_-STRIP, POLYGON, QUAD_STRIP etc [13]. The QUAD configuration was used here – although a TRIANGLE based warping may in theory give a more credible image warp – and a diagram illustrating how the connections are made between the vertices is shown in Figure 1.

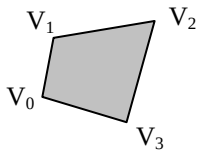
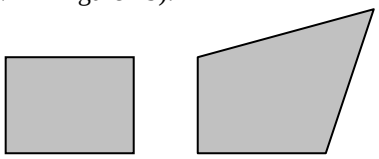


Fig. 1 How the vertices are linked on the GPU.

The vertices are connected from V0 to V1 to V2 to V3 and back to V0. Each vertex takes a texture coordinate (indicating the absolute position in the image of the vertex) and a vertex coordinate (indicating where the vertex should be placed). For example if we had the following 'normal' image shown in Figure 2a) and we moved the top right vertex up and to the right the resulting scaled image would

A. L. Nel is with the Mechanical Engineering Sciences Department at the University of Johannesburg, SA (e-mail: andren@uj.ac.za).

be that shown in Figure 2b).



a) *b)*

Fig. 2 Representations of how vertex movement affects an image.

One can add many QUAD vertices to an image. By moving each respective one to the desired location one is able to warp the images to ensure that all the epipolar lines are horizontal.

III. Implementation

The rectification algorithm comprises of the following steps, each of which will be described in greater detail below.

1. Locate the epipole for both the right and left images.
2. Calculate which area the epipole is located in (see Figure 3).
3. Calculate the maximum and minimum angles.
4. Calculate the maximum and minimum distances from the epipole.
5. Calculate where each vertex must move to.

A. Locating the epipole

In order to locate the coordinates for the epipole two epipolar lines are chosen. For an image of size m by n, one can take the transpose of the fundamental matrix multiplied by the point (0, 0, 1) to get line 1 in the right image (see (2)) and the transpose of the fundamental matrix multiplied by the point (0, n, 1) to get line 2 in the right image. The coordinates where these two intersect is the epipole for the right image. The same is done for the left image instead one uses the fundamental matrix in the multiplication (see (3)).

There are 9 possible regions where the epipole could be located, these are depicted in Figure 3.

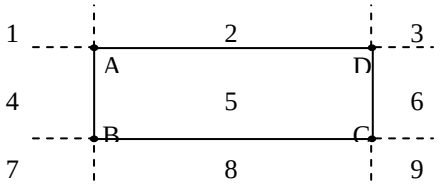


Fig. 3 The nine possible regions where the epipole could be located.

B. Finding the maximum and minimum angles and distances

Depending on the area in which the epipole is found one calculates the maximum and minimum angles and distances. This is shown diagrammatically in Figure 4.

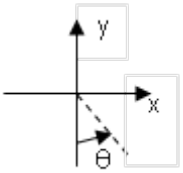


Fig. 4 The convention for positive theta.

For example if one has the epipole in region 4 then the maximum theta will be the value of theta for the line that joins the epipole and vertex A (Figure 5 angle FÊA) and the minimum theta will be the value of theta for the line that joins the epipole and vertex B (Figure 5 angle FÊB).

The maximum distance depends on whether point E (the epipole) is above or below the half way mark of line A-B. If the epipole is above the half way mark then the maximum distance is the distance of the line E-C, otherwise it is the length of the line E-D, as is the case shown in Figure 5. The minimum distance is the perpendicular distance from the epipole to line A-B (distance E-G in Figure 5).

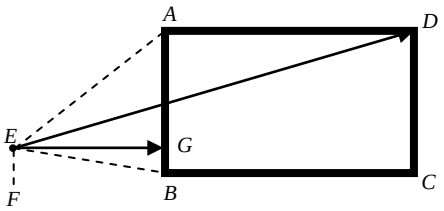


Fig. 5 Method for calculating the extreme values for rectification purposes.

This step is performed for both the left and right images. The smaller of both minimum thetas is the final minimum theta. The same is applicable for the minimum distance. The larger of the maximum thetas is used as the final maximum theta. This also applies to the distance, the larger of the maximum distances is used as the final maximum distance.

C. Calculate vertex movement

The range of theta and of the distance has now been determined. If the epipole is in region 4 then we traverse down the y axis from vertex A towards vertex B in incremental units (Figure 5). For every epipolar line there is an entry point into the image as well as an exit point. Both of these two points will be made a vertex with the respective texture coordinate and the appropriate vertex coordinate. In order to calculate the vertex coordinate the point is moved from the (x, y) domain to the (r, θ) domain, with r comprising the traditional x axis and θ the traditional y axis. The vertex coordinates go from -1 to 1 in the θ direction and from –Ratio to Ratio in the r direction. The value for the Ratio is the image width divided by the image height. Thus one maps the r and θ obtained to a value between the given bounds. The θ is mapped using (4) and the distance is mapped using (5).

$$\left(\frac{\theta-\theta_{\min}}{\theta_{\max}-\theta_{\min}}\right)\times 2-1 \tag{4}$$

$$\left[\left(\frac{R-R_{\min}}{R_{\max}-R_{\min}}\right)\times 2-1\right]\times Ratio \tag{5}$$

Figure 6 shows how the transformation occurs. Note that in Figure 6a) there are only 5 vertices down line A-B whereas in fact the actual number of lines contributing to point E is equal to the number of pixels along line A-B.

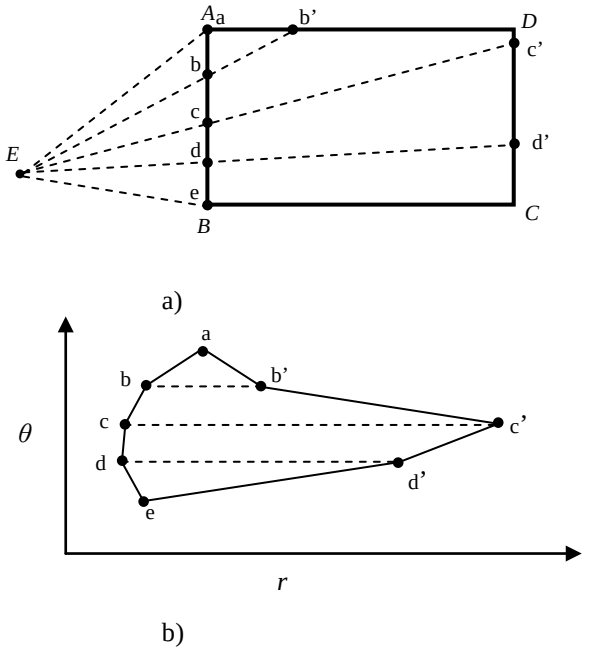


Fig. 6 a) Image before rectification; b) Warped image.

IV. Tests

A single uncalibrated stereo image pair was used as the base image for the tests [14]. Three different noise types were added to the base image. Three noise distributions investigated were Salt and Pepper, Speckle and Gaussian. For every noise type there were ten varying levels of noise added. Due to the fact that the noise added is random, 200 repetitions at the same Signal to Noise Ratio (SNR) were done. This gives a total of 16 000 images evaluated.

The ranges used for the three noise types are indicated in Table I.

TABLE I
THE RANGES USED FOR THE THREE NOISE TYPES.

Noise Type	Lowest Value	Highest Value	Increment Size
Gaussian	0.001	0.01	0.001
Salt And Pepper	0.01	0.1	0.01
Speckle	0.002	0.02	0.002

The SNR is calculated as indicated in (6), where j maps the three channels (red, green and blue) and k is the total number of pixels in the image. Δj, k is shown in (7) and is equal to the intensity of the noisy image minus the intensity of the base image (with zero noise).

$$SNR = 10\log\left(\frac{\sum_{j=1}^{3}\sum_{k=1}^{N}\left[\Delta_{j,k}\right]^2}{\sum_{j=1}^{3}\sum_{k=1}^{N}\left[I_{j,k}(o)\right]^2}\right) \quad (6)$$

$$\Delta_{j,k} = I_{j,k}(Noise) - I_{j,k}(o) \quad (7)$$

The resulting SNR's are indicated in Table II.

TABLE II
THE RANGES USED FOR THE THREE NOISE TYPES.

Intensity Level	Gaussian		Salt And Pepper		Speckle	
	Left	Right	Left	Right	Left	Right
1	-25.82	-25.68	-19.86	-19.72	-27.18	-27.16
2	-22.90	-22.76	-16.84	-16.71	-24.40	-24.39
3	-21.22	-21.08	-15.08	-14.94	-22.83	-22.83
4	-20.05	-19.90	-13.84	-13.69	-21.75	-21.75
5	-19.14	-18.99	-12.87	-12.72	-20.93	-20.93
6	-18.40	-18.25	-12.08	-11.94	-20.26	-20.26
7	-17.77	-17.63	-11.40	-11.26	-19.69	-19.69
8	-17.24	-17.09	-10.82	-10.68	-19.20	-19.20
9	-16.77	-16.62	-10.31	-10.17	-18.77	-18.77
10	-16.35	-16.20	-9.85	-9.71	-18.37	-18.38

The SNR values in Table II are relatively large and modern imaging equipment rarely has such SNR values as used in these experiments. All 16 000 images were processed to find key points and from that the epipole for each image. Normal distributions were fitted to the observed distributions of epipoles using (8).

$$y = \alpha \frac{e^{-\left[\frac{(x-\mu)^2}{2\sigma^2}\right]}}{\sqrt{2\pi\sigma^2}} \quad (8)$$

Where α is the number of samples in the set, σ is the standard deviation and μ is the mean. These are plotted in Figure 7 and Figure 8.

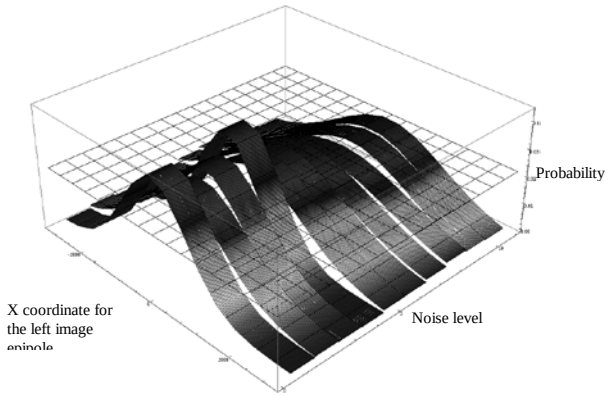


Fig. 7 The distribution of left X coordinate for Speckle noise.

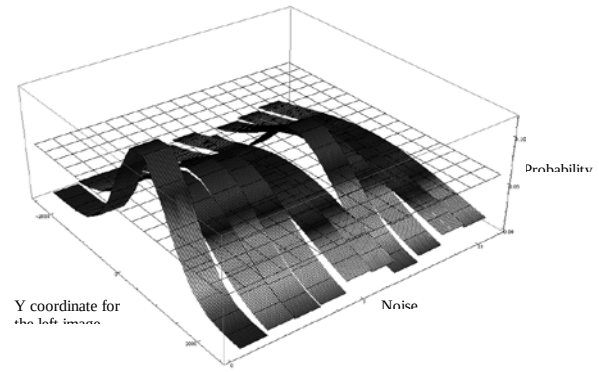


Fig. 8 The distribution of left Y coordinate for Salt and Pepper noise.

For all three of the noise types there were four graphs (Xl epipole for left image, Yl epipole for left image, Xr epipole for right image, Yr epipole for right image). All ten SNR levels per noise type were plotted on the same graph so that it is possible to compare the noise effect on the image epipole.

V. Results

The epipole for the base left and base right images were located at (-252.19, 367.58) and (-183.02, 344.52).

Figure 7 and Figure 8 represent the ten normal distribution curves for the left X coordinate, right X coordinate and the left Y coordinate. Each bell curve represents a different noise level which is increasing from the front to the back of the graph.

Figure 9 to Figure 10 and Figure 11 to Figure 12 show how the average and standard deviation respectively varies as the SNR increases. Figure 13 and Figure 14 show the average and standard deviation varying for changes to the SNR for Salt and Pepper noise. The SNR values of Salt and Pepper are larger than the other noise types and it is for this reason that they are represented on separate figures.

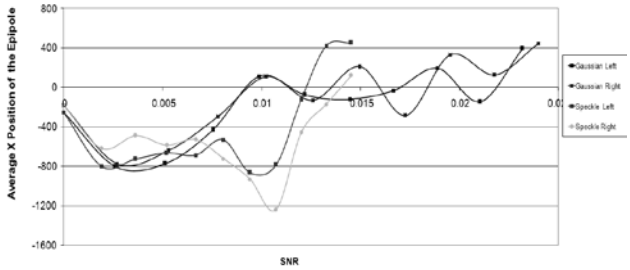


Fig. 9 The AVG as the SNR changes for the X coordinate.

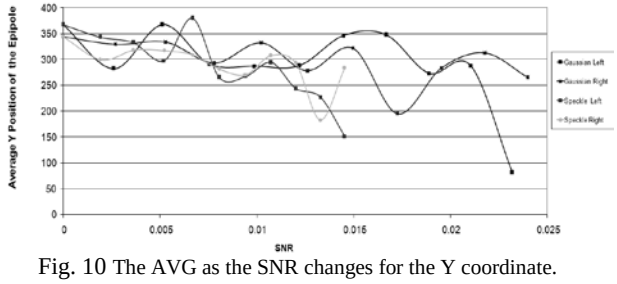


Fig. 10 The AVG as the SNR changes for the Y coordinate.

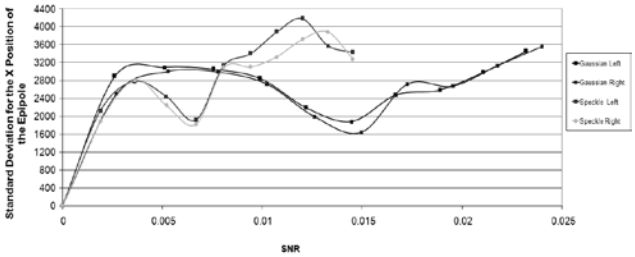


Fig. 11 The STD DEV as the SNR changes for the X coordinate.

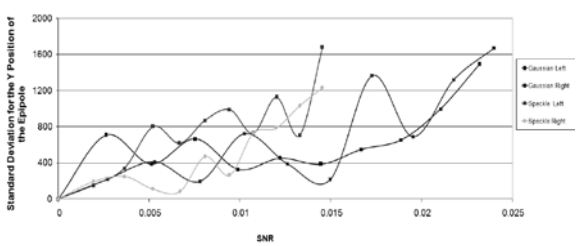


Fig. 12 The STD DEV as the SNR changes for the Y coordinate.

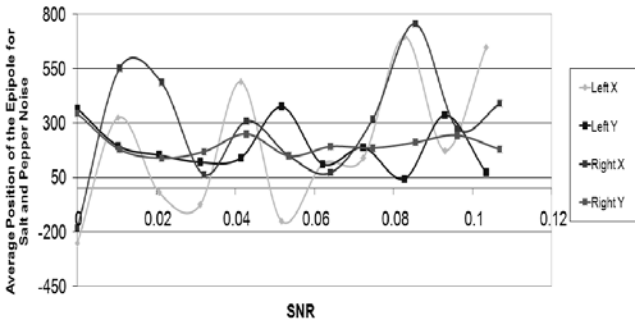


Fig. 13 The AVG as the SNR changes for Salt and Pepper noise.

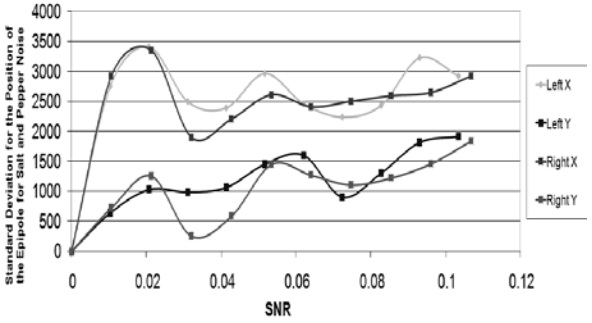


Fig. 14 The STD DEV as the SNR changes for Salt and Pepper noise.

a) b)

Fig. 15 a) The original left image; b) The original right image.

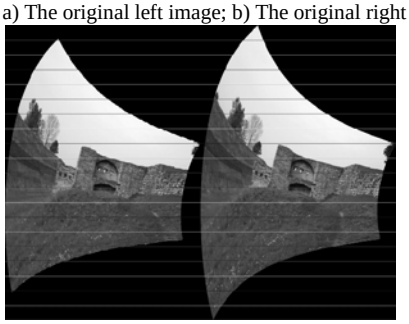


Fig. 16 The left and right images after polar rectification with no noise.

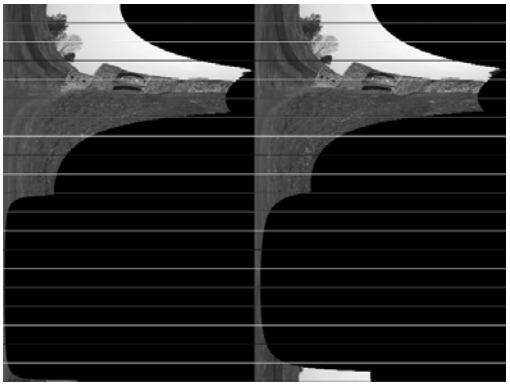


Fig. 17 The left and right images after polar rectification for the Gaussian noise type with a SNR of -16.35 dB.

VI. Discussion of Results

Referring to both Figure 7 and Figure 8 one notices that the bell curves begin to flatten out after the fourth curve. This implies a wider distribution to the location of the epipoles. It is interesting to note that in Figure 8 the seventh noise level rises up compared to the sixth and eighth noise levels. However the spread of the seventh compared to the first noise level is larger. The flatter the bell curve the wider the spread of the location of the epipole coordinate in question. The lower the standard deviation, the tighter the normal distribution becomes. One would expect that as the noise intensity increased so would the standard deviation.

Figure 9 has a trend of the average increasing with an increase to the SNR. This means that the X epipole coordinate is moving from the left to the right of the image. Figure 10 has the Y coordinate for the epipole tending to decrease slightly with an increase to the SNR indicating that the epipole is moving from the bottom to the top of the image.

It is interesting to note that from Figure 10 the average Y coordinate is never negative or greater than 480 (the height of the image), thus implying that the epipole is always either in quadrant 4, 5 or 6 (refer to Figure 3). The average X coordinate in Figure 9 is always less than 640 (the width of the image), with the exception of the Salt and Pepper noise type at noise level 8 and 10 for the left image and noise level 8 for the right image. Thus excluding the three before mentioned exceptions the epipole is always in either quadrant 4 or 5.

Figure 11 and Figure 14 (X coordinates) indicate that the standard deviation tends to remain close to constant with an increase in SNR with the exception of the Speckle noise type which clearly shows an increase to the standard deviation with an increase to the SNR. This seems to indicate the X coordinate of the epipole is not affected by the SNR for the Gaussian and Salt and Pepper noise types and is affected by the Speckle noise type.

Figure 12 and Figure 14 (Y coordinates) seems to show an increase to the standard deviation with an increase to the SNR. This would indicate that the range of Y coordinates for the epipole increases with an increase of SNR. Interestingly the standard deviation values in Figure 12 are far less than those in Figure 10. This is also true for the X coordinates and Y coordinates in Figure 14. This shows that the X coordinate has a much greater spread than the Y counterpart. This is because during the polar rectification process the aim is to get the points on the same horizontal line as their matching points in the other image as opposed to getting the matching points on the same vertical line.

Figure 15 shows the original unwarped images. Figure 16 and Figure 17 show two warped images, one Figure 16 with no noise and Figure 17 where Gaussian noise with a SNR of 0.002616 was added to the original image. The horizontal lines show that the warping has had the desired effect of getting the points in the two images on the same line. The colours of the horizontal lines are merely included for the viewer's convenience. The quadrant found in Figure 16 and Figure 17 are 4 and 5 respectively (refer to Figure 3).

The performance of this implementation on the GPU is reaches rates of over 150 frames per second. The matching of image points has been reduced to matching of points in a single horizontal line with very little computational time required. This will significantly reduce the time required to compute a depth map for a stereo image pair.

VII. Conclusion

Addition of noise, no matter how small, affects the determination of the epipole position. However the magnitude of the SNR has little to no effect for Gaussian and Salt and Pepper. The Speckle noise type showed that the spread of coordinate values increased with an increase to the noise intensity level. This results in the algorithm not always working as desired with the introduction of large SNR values. The SNR values used far exceed those that would be expected in a practical application. Therefore the algorithm would thus be expected to perform at the same or better level in practice to what is described here.

An average rate of 150 frames per second was recorded for the polar rectification process on the GPU. Thus the process of reducing the search space for matching corresponding points in a stereo image pair is achieved with very little computational time required.

References

[1] D. Min and K. Sohn, "Cost aggregation and occlusion handling with wls in stereo matching" TIP, vol. 17, no. 8, pp. 1431-1442, 2008.

[2] D. Scharstein , R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms", International Journal of Computer Vision, 47(1/2/3):7-42, April-June 2002.

[3] S. Forstmann, J. Ohya, Y. Kanou, A. Schmitt, S. Thuering, "Real-time stereo by using dynamic programming", CVPRW2004, vol. III, p. 29, 2004.

[4] J. Sun, Y. Li, S. B. Kang, H. Y. Shum, "Symmetric Stereo Matching for Occlusion Handling", CVPR2005, pp. 1075-1082, 2005

[5] C. L. Zitnick, T. Kanade, "A cooperative algorithm for stereo matching and occlusion detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22(7), pp. 675-684, 2000.

[6] Y. Boykov, O. Veksler, R. Zabih, "Fast approximate energy minimization via graph cuts", IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11), pp. 1222-1239, 2001.

[7] V. Kolmogorov, R. Zabih, "Visual correspondence with occlusions using graph cuts", ICCV2001, pp. 508-515, 2001.

[8] A. F. Bobick, S. S. Intille, "Large occlusion stereo", IJCV, 33(3), pp. 181-200, 1999.

[9] M. Lin, T. Tomasi, "Surfaces with occlusions from layered stereo", CVPR2003, vol. 1, pp. 710-717, 2003.

[10] R. Hartley, A. Zisserman, Multiple View Geometry in computer vision, 2000, ISBN-13 978-0-521-54051-3.

[11] A. W. Fitzgibbon, Y. Wexler, A. Zisserman, "Learning epipolar geometry from image sequences", International Journal of Computer Vision, Volume 75, Issue 3 December 2007

[12] B. Boufama, R. Mohr, "Epipole and Fundamental Matrix Estimation Using the Virtual Parallax Property" Proc. Fifth Int'l Conf. Computer Vision, pp. 1,030-1,036, June 1995.

[13] The Official Guide to Learning OpenGL, Version 1.1 Second Edition Addison-Wesley Publishing Company, January 1997, ISBN: 0201461382.

[14] http://profs.sci.univr.it/~fusiello/demo/rect/, Epi-polar rectification paper with tutorial (Date accessed: 27 July 2010)