Fast Stereo Triangulation using Symmetry

Wai Ho Li and Lindsay Kleeman
Intelligent Robotics Research Centre
Monash University, Clayton
Melbourne, Australia
{Wai.Ho.Li, Lindsay.Kleeman}@eng.monash.edu.au

Abstract

This paper proposes a method to use reflectional symmetry as a feature to rapidly triangulate the 3D location of objects using stereo vision. Our approach can triangulate objects under non-uniform lighting conditions. Objects that pose a problem to other stereo methods, such as reflective and transparent objects, as well as objects with rounded surfaces prone to incorrect stereo matching, can also be triangulated using our method. Assuming the object being triangulated is visually symmetric, no *a priori* models are required. The triangulation approach was tested experimentally. The test data contain 24 image pairs of 6 different objects, each placed at four different locations. The triangulation results are compared quantitatively against ground truth locations on a camera calibration checkerboard. The mean error of the triangulation is 10.6mm across the entire test data set. A qualitative comparison show that our algorithm out performs dense stereo methods for a variety of objects. The algorithm can operate on 640x480 images at 5 frame-pairs-per-second on a standard laptop PC.

1 Introduction

This paper proposes a method to triangulate reflectional symmetry lines with a stereo camera pair to find the 3D location of objects. This work can be applied to any robotic platform equipped with a stereo camera pair, but is specifically designed for use on a humanoid robot platform in the authors' research laboratory. The stereo triangulation method will be used to aid the localization and grasping of objects such as boxes, cups, cans and bottles, resting upright on a table. As shape, colour and size vary between our objects, and some objects are transparent, highly reflective or multi-coloured, symmetry provides an elegant way of representing the entire set of objects. As symmetry detection does not require *a priori* object models, no data collection for offline training or manual model construction is needed to triangulate novel symmetric objects in the robot's environment.

The result of our stereo triangulation is the location of an object's symmetry line in 3D, defined by its two end points. For most visually symmetric objects, this line will pass through their centre. This is different from the results returned by other stereo algorithms according to recent surveys [Brown *et al.*, 2003; Scharstein and Szeliski, 2001]. Dense stereo algorithms produce a disparity map of distances to the surface of an object. Sparse feature-based stereo also provides distances to select locations on an object's surface. As such, even though our approach is feature-based, the result cannot be classified as dense or sparse stereo.

In the context of humanoid robotics, having the location of the object centre will benefit grasp planning and object manipulation tasks. Our approach can also be used synergetically with standard stereo methods. The fusion of surface depth with the object centre provided by our approach will provide a richer metric model of a robot's manipulatable environment. Also, the triangulated symmetry line can be used to initialize the object pose to bootstrap model-fitting algorithms.

The paper is partitioned in the following manner. Section 2 provides an overview of our fast symmetry detection algorithm. Our stereo triangulation algorithm is described in section 3. Experiment results are located in section 4. The results include the triangulation accuracy of our method measured against ground truth as well as a qualitative comparison with dense stereo. The ground truth locations were found by triangulating corners of known association on a checkerboard calibration pattern. Where appropriate, a summary of related research is provided at the beginning of major sections.

2 Fast Symmetry Detection

There are several established methods to detect symmetry in digital images. The Generalized Symmetry Trans-

form [Reisfeld *et al.*, 1995] can detect reflectional and radial symmetry at different scales. It has a computational complexity of $O(n^2)$, where n is the total number of pixels in the input image. Levitt first suggested using the Hough transform to find symmetry in point clusters [Levitt, 1984]. A similar method was employed by Yip's [Yip *et al.*, 1994] symmetry detector, which can detect reflectional and skew symmetry. However, as the algorithm uses mid-point pairs, each generated from *two* edge pixel pairs, it has a complexity of $O(n_{edge}^4)$, where n_{edge} is the number of edge pixels in the image. Other approaches include the use of ribbons [Ponce, 1990] and modified versions of the Generalized Symmetry Transform that can perform symmetry detection at specific corner angles [Choi and Chien, 2004].

While radial symmetry has been used in real time applications [Loy and Zelinsky, 2003], reflectional symmetry detectors, due to their high computational costs, have only been used for offline processing in the past. To remedy this, the authors proposed a fast reflectional symmetry algorithm [Li *et al.*, 2005]. An updated version of the fast symmetry detection algorithm, with improved computational efficiency and accuracy, is used in our triangulation method. Section 2.1 below describes the algorithm and implementation details, as well as parameters that are relevant to stereo triangulation.

2.1 Algorithm Description

Symmetry detection is performed using the edge pixels of an image. By doing this, detection indirectly benefits from the noise rejection, edge linking and weak edge retention properties of edge filters. The Canny edge filter with a 3x3 aperture and fixed thresholds is used for edge detection. Edge pixels are grouped into pairs and each pair votes for a single symmetry line in a polar parameter space, as seen in Figure 1. Unlike traditional Hough Transform [Duda and Hart, 1972], which requires multiple votes per edge pixel, our approach only requires a single vote per *edge pixel pair*. Hence, the computational complexity of our algorithm is $O(n_{edge}^2)$, where n_{edge}^2 is the number of edge pixels filtered from our input image. This *convergent* voting scheme is similar to the approach used in Randomized Hough Transform [Xu and Oja, 1993].

Given a range of symmetry line angles, the edge pixels are rotated about the center of the image by a series of discrete angles. The angle discretization is based on the size of the Hough accumulator's angle bins. The rotated edge pixels are then quantized into a 2D array, named *Rot* in Algorithm 1. Edge pixels are placed into the rows of *Rot* based on their *scanline* after rotation, as shown in Figure 2.

Notice that the edge pixels belonging to the same row can only vote for symmetry lines at the current angle of

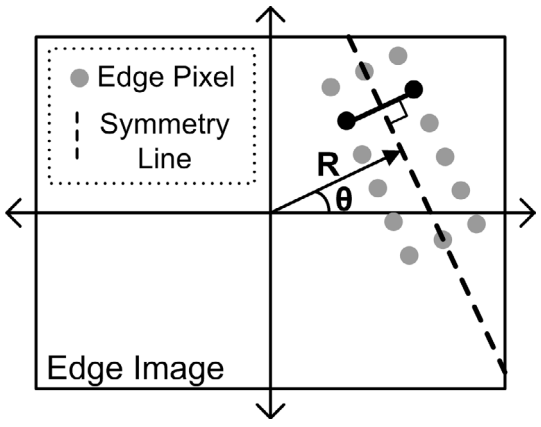


Figure 1: Edge pixels voting for dashed symmetry line

rotation. This corresponds to the dashed symmetry line at angle θ in Figure 2. The line radius R can be found by taking the average of the x coordinates of an edge pixel pair. For example, the [3,1] rows will vote for the dashed symmetry line ($R = 2$). After voting, symmetry lines are found by looking for peaks in the Hough accumulator. An iterative non-maxima suppression algorithm is used for peak finding.

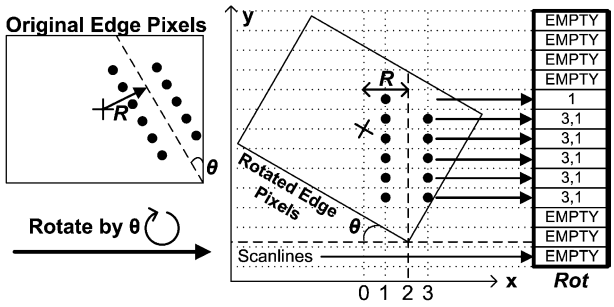


Figure 2: Edge Pixel Rotation and Quantization

The entire fast symmetry detection process is described in Algorithm 1. As edge pixels are sorted into rows, the Hough accumulation has in effect been divided into multiple voting steps, one for each discrete angle. This approach allows angle limits to be placed on the detection process, which can be used to improve computational efficiency of detection. For the purposes of stereo, horizontal symmetry lines cannot be used for triangulation, as their projected planes do not intersect in any meaningful way. As such, in our stereo triangulation experiments, the detection angle is limited to ± 25 degrees of vertical. This reduced the detection time by about 70%.

2.2 Detection Results on Single Images

Symmetry detection results are shown in Figure 3. The images are taken from the test data used in our triangulation experiments. Note that the result have been

Algorithm 1: Angle-Limited Fast Sym. Detection

Input: I – Source Image
Output: sym – Symmetry Line Parameters (R, θ)
Parameters:
D_{min} – Minimum distance threshold
D_{max} – Maximum distance threshold
H – Hough Accumulator
θ_{lower}, θ_{upper} – Detection Angle Range
N_{lines} – Number of symmetry lines returned

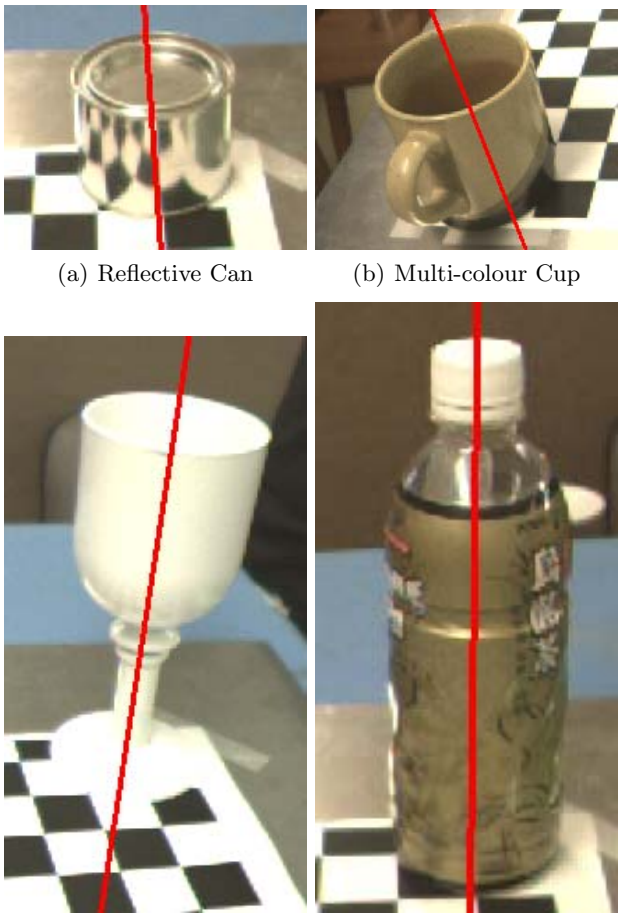
```
edgePixels ← (x,y) locations of edge pixels in I
H[ ][ ] ← 0
for θ_index ← θ_lower to θ_upper do
    θ ← θ_index in radians
    Rot ← Rotate edgePixels by angle θ. See Figure 2
    for each row in Rot do
        for each possible pair (x1, x2) in current row do
            dx ← |x2 − x1|
            if dx < D_min OR dx > D_max then
                continue to next pair
            x0 ← (x2 + x1)/2
            Increment H[x0][θ_index] by 1
for i ← 1 to N_lines do
    sym[i] ← max(R_index, θ_index) ∈ H
    Bins around sym[i] in H ← 0
```

cropped and enlarged for visualization purposes. In all the images shown, the object's symmetry lines are the strongest in terms of their total Hough votes. However, background symmetry may at times over shadow foreground symmetry. As such, in our stereo algorithm, the five strongest symmetry lines from each image are used for triangulation.

Examples of full camera images corresponding to objects in Figure 3 can be seen in Figure 7. Note that the fast symmetry detection algorithm can find lines of symmetry for multi-colour, textured, reflective and even transparent objects. These detection results provides an indication of the robustness and generality of symmetry as an object feature.

3 Stereo Triangulation

While a plethora of stereo algorithms have been developed to date, their algorithmic process can usually be generalized into several steps. First, where possible, the intrinsic and extrinsic parameters of the stereo camera pair are found through a *Calibration* step. After calibration, the next stage of most stereo algorithms can be termed *Correspondence*. This stage tries to match portions of the left and right images that belong to the same 3D location. The 3D location is usually assumed



(a) Reflective Can (b) Multi-colour Cup

(c) White Cup (d) Textured Bottle

(e) White Bottle (f) Transparent Bottle

Figure 3: Symmetry Detection Results

to be a lambertian surface which appears the same in both camera images. Once corresponding portions have been found, its distance from the camera can be triangulated using the intrinsic and extrinsic parameters found during calibration.

In *sparse* or feature-based stereo, more commonly used in wide baseline and uncalibrated systems, a set of feature points are matched. Recent sparse stereo approaches generally make use of scale invariant features such as Lowe's SIFT operator [Lowe, 2004] , or affine transform invariant features such as Maximally-Stable Extrema Regions (MSER) [Matas *et al.*, 2002]. With increasing computing power, recent trends have also gravitated towards matching descriptive features such as SIFT and other Histogram-of-Gradients patches.

In *dense stereo* algorithms, correspondences between every pixel is found. Depending on the time available for processing, dense stereo approaches utilize a variety of optimization methods to find the best correspondences. Local approaches simply find the best patch match along epiploar lines. Global approaches may use dynamic programming or network algorithms such as graph cuts to optimize across multiple pixels.

3.1 Camera Calibration

Our stereo camera pair was calibrated using the MATLAB calibration toolbox [Bouguet, 2006]. Both intrinsic and extrinsic parameters were estimated prior to triangulation. The intrinsic parameters refer to the camera-specific idiosyncrasies, such as focal length, image offset from the optical center and lens distortion. The extrinsic parameters model the physical pose of the two cameras, such as their translation and rotation relative to each other. Note that camera calibration may not be possible in some situations. In fact, many wide baseline stereo approaches are used to recover calibration parameters online.

Figure 4 shows the extrinsics of our stereo cameras, looking downwards from above the cameras. The cameras are verged towards each other to provide a larger overlap between the images. The vergence angle is roughly 15 degrees, and the right camera is rotated slightly about its z axis due to a slight rotation introduced by the mounting bracket. The red triangles indicate the camera's field of view. The origin of the axes is located at the focal point of the left camera.

3.2 Triangulating Symmetry Lines

Due to the reduction from three dimensions down to a pair of 2D image planes, stereo correspondence is not a straightforward problem. Apart from the obvious issue of partial occlusions, where only one camera can see the point being triangulated, other problems can arise. For example, specular reflections and non-lambertian surfaces will cause the same location to appear differently

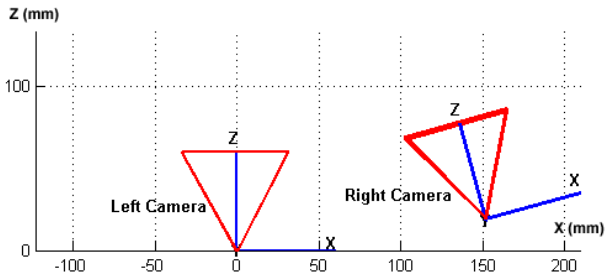


Figure 4: Extrinsics of the Verged Stereo Pair

in the stereo images, which can make correspondence difficult. The proposed method attempts to provide a robust solution by using symmetry lines as the primary feature for stereo matching. By using symmetry, we also show that both reflective and transparent objects can be successfully triangulated.

Figure 5 shows the triangulated symmetry axes of the reflective metal can seen in Figure 7(e), positioned on the four outer corners of the checkerboard. The red lines are the triangulated symmetry lines of the metallic can. The blue dots are the corners of the checkerboard as seen in Figure 6. The stereo camera pair can be seen in the upper left of the figure.

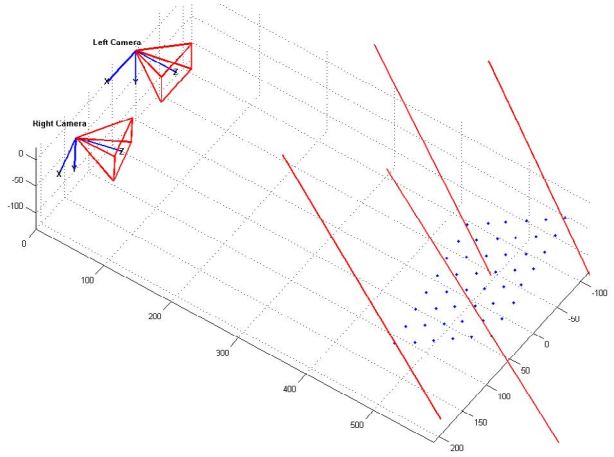
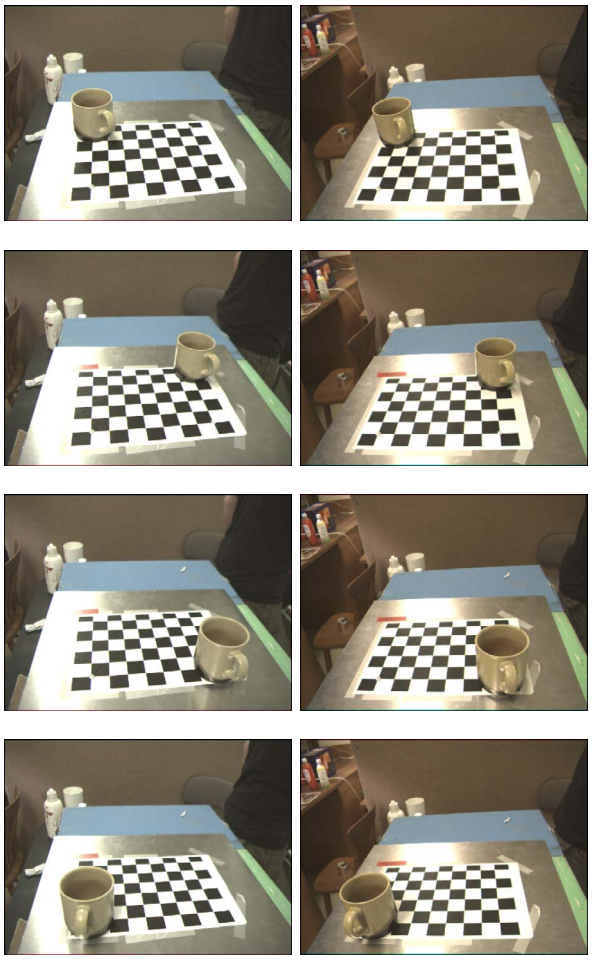


Figure 5: Triangulation Results for the Reflective Metal Can shown in Figure 7(e)

The 3D location of an object's symmetry axis is found using the following method. First, we project the symmetry line out from a camera's focal point. The projection forms a semi-infinite triangular plane in 3D space. This projection is done for both camera images using their respective detected symmetry lines. After this, we search for an intersection between the triangles emanating from each camera. The triangulation results is simply the line of intersection, assuming one exists.

4 Experiment Results

The test data contains six objects, each placed on the 4 outer corners of the checkerboard, giving 24 image pairs in total. One of our six object sets, containing 4 image pairs, is shown in Figure 6. All six objects in the test set can be seen in Figure 7.

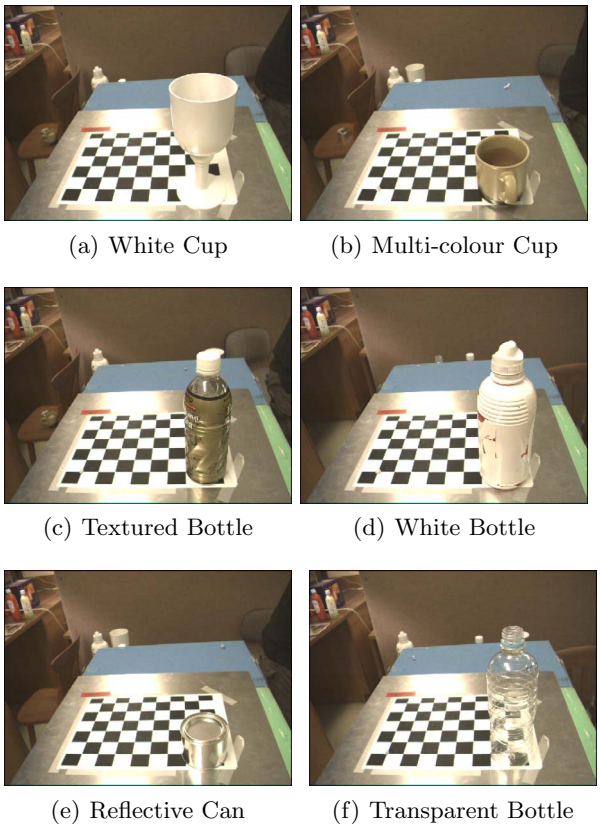


(g) Left Camera Images (h) Right Camera Images

Figure 6: Example Stereo Data Set: Multi-colour Cup

All the images were taken with the cameras located between 400mm to 1200mm from the object being triangulated. This was done to simulate the situation of a humanoid robot interacting with objects at a table using its arm. Each object being triangulated was placed on the four outer corners of the a checkerboard pattern. At each location, 640x480 images were taken using the verged camera pair shown in Figure 4. The object's symmetry line was physically aligned to be directly above the checkerboard corner.

Our algorithm is implemented using C++. Tomas Moller's triangle intersection code [Moller, 1997] is used to find the end points of the triangulated symmetry lines. The compiled binary run at 5 frame-pairs-per-second on a Pentium M 1.73GHz laptop PC when operating on the 640x480 images. This frame rate includes the Canny edge detection and symmetry detection on both stereo images.



(a) White Cup (b) Multi-colour Cup

(c) Textured Bottle (d) White Bottle

(e) Reflective Can (f) Transparent Bottle

Figure 7: Stereo Data Sets used in experiments. Right camera images with object located at bottom right corner of calibration checkboard shown

4.1 Triangulation Accuracy

To obtain ground truth, an additional image pair was taken with nothing on the checkerboard. This was done for each data set, to ensure that subtle movements of the checkerboard will not adversely affect the results. Using MATLAB code based on Jean-Yves Bouguet's calibration toolbox [Bouguet, 2006], the corner locations of the checkerboard were extracted in the stereo images. The locations of these corners in 3D space were found using the calibration toolboxes' triangulation code.

A Hessian model for the checkerboard plane was found using a least squares fit of the triangulated corner points. The plane Hessian provides the 3D location of the table on which our objects were placed. Using the plane Hessian, an intersection between the object's triangulated symmetry line and the table plane is found. This intersection point represents the location of the object's

centre, according to its detected symmetry, on the table. The Euclidean distance between the point of intersection and the ground truth location, found using standard triangulation, is then calculated. This distance is used as the error metric between the results of our algorithm and ground truth.

The following steps was used to measure the triangulation accuracy by applying the error metric. First, the top five symmetry lines was found for each image in a stereo pair. All possible pairings of symmetry lines between the left and right camera images were found. These pairings were triangulated by computing the line of intersection between their projected planes. Triangulation results were ignored if the result was more than 1200mm away from the camera. Triangulated symmetry lines that were not within 5 degrees of the checkerboard's surface normal were also ignored. Intersection points between the remaining valid symmetry lines and the table plane were found.

After obtaining a list of intersection points for all image pairs, the triangulation accuracy was measured using our error metric. In the case where multiple points were found for a ground truth datum, the nearest point was used. If no intersection point was found, the triangulation was considered a failure.

Table 1 shows the average triangulation error for our test objects. All six test objects can be seen in Figure 7. The mean error was calculated for four triangulation attempts, one at each outer corner of the checkerboard pattern, resulting in a total of 24 triangulations. There was only a single failure among the 24 triangulation attempts. The failed triangulation occurred with the multi-colour cup due to self occlusion caused by the cup's handle. The mean error across the successful triangulations was $10.62mm$, with a standard deviation of $7.38mm$. An average of 1.5 symmetry lines were found per object, which is very good considering that high level knowledge has not been applied to reject the non-object symmetry lines.

Table 1: Triangulation Error at Checkerboard Corners

Object	**Mean Error (mm)**
White Cup	13.5
Multi-Color Cup	6.8 †
White Bottle	10.7
Textured Bottle	12.4
Reflective Can	4.5
Transparent Bottle	14.9

†Triangulation failed for 1 of 4 locations

4.2 Qualitative Comparison with Dense Stereo

Dense and sparse stereo approaches provide 3D information about the surface of an object. As mentioned in our introduction, this is different from the results of our triangulation algorithm, which returns an object's symmetry axis. This axis is always inside an object, and usually passes through its centre. Due to this geometric difference between the results, a *qualitative* comparison with dense stereo was performed.

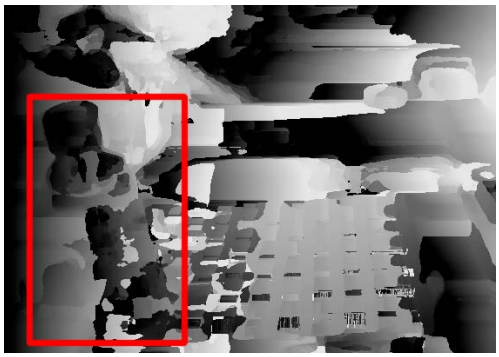
Dense Stereo disparity maps were generated using C++ code from the Middlebury Stereo Research Lab [Scharstein and Szeliski, 2006]. The input images were rectified using the MATLAB calibration toolbox before disparity calculations. After testing multiple stereo cost functions and optimization approaches, Sum-of-Squared-Differences (SSD) of 15x15 windows was found to produce the best results for our test images. Global optimization methods, such as dynamic programming, did not provide any significant improvements. Figure 8 shows disparity results for several objects, corresponding to objects in Figure 7. Darker pixels have lower disparity, that is, the locations they represent are further from the camera. The object's location in the disparity map is marked with a red rectangle in the disparity map.

The three objects shown in Figure 7 can be triangulated using our symmetry-based approach. Note that triangulation results for the reflective can is also displayed in Figure 5. Looking at the disparity maps, it is difficult to imagine any high level algorithm that can recover the object location.

The comparison also highlighted cases where the traditional stereo assumption of the same surface appearing similar in a stereo pair tend to fail. The textured bottle's surface curvature, combined with non-uniform lighting, can cause the same surface to appear differently to each camera. The transparent bottle appears as a distorted version of its background, and its appearance changes from different view points. The reflective can acts as a curved mirror and reflects its surroundings, which also violates the assumption.

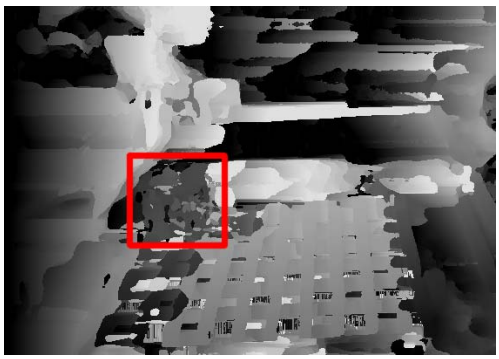
5 Conclusion

A stereo triangulation method that uses object symmetry as its primary feature has been proposed. Experiments were carried out on 24 stereo image pairs of six different objects. The experiment demonstrates that the algorithm can triangulate objects of different colour, shape and size. Also, reflective and transparent objects, which are undetectable using dense stereo, can be triangulated using our method. The triangulation has a mean error of 10.6mm from ground truth across all image pairs, discounting a single failure. The C++ implementation operates at 5 frame-pairs-per-second on 640x480 images.



(a) Textured Bottle

(b) Transparent Bottle



(c) Reflective Can

Figure 8: Dense Stereo Disparity Results. The object is enclosed with a red rectangle

Acknowledgements

The authors would like to thank PIMCE (www.pimce.edu.au) for their financial support.

References

[Bouguet, 2006] Jean-Yves Bouguet. Camera calibration toolbox for matlab. Online, July 2006. http://www.vision.caltech.edu/bouguetj/calib doc/.

[Brown *et al.*, 2003] MZ Brown, D. Burschka, and GD Hager. Advances in computational stereo. *IEEE PAMI*, 25(8):993–1008, 2003.

[Choi and Chien, 2004] I. Choi and S.I. Chien. A generalized symmetry transform with selective attention capability for specific corner angles. *IEEE Signal Processing Letters*, 11(2):255–257, February 2004.

[Duda and Hart, 1972] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[Levitt, 1984] Tod S. Levitt. Domain independent object description and decomposition. In *AAAI*, pages 207–211, 1984.

[Li *et al.*, 2005] Wai Ho Li, Alan M. Zhang, and Lindsay Kleeman. Fast global reflectional symmetry detection for robotic grasping and visual tracking. In *ACRA*, December 2005.

[Lowe, 2004] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[Loy and Zelinsky, 2003] Gareth Loy and Alexander Zelinsky. Fast radial symmetry for detecting points of interest. *IEEE PAMI*, 25(8):959–973, 2003.

[Matas *et al.*, 2002] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *BMVC*, 1:384–393, 2002.

[Moller, 1997] Tomas Moller. A fast triangle-triangle intersection test. *JGT*, 2(2):25–30, 1997.

[Ponce, 1990] J. Ponce. On characterizing ribbons and finding skewed symmetries. *CVGIP*, 52(3):328–340, 1990.

[Reisfeld *et al.*, 1995] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context-free attentional operators: The generalized symmetry transform. *IJCV, Special Issue on Qualitative Vision*, 14(2):119–130, March 1995.

[Scharstein and Szeliski, 2001] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Technical Report MSR-TR-2001-81, Microsoft Research, Microsoft Corporation, November 2001.

[Scharstein and Szeliski, 2006] Daniel Scharstein and Richard Szeliski. Middlebury stereo c++ code. Online, February 2006. www.middlebury.edu/stereo.

[Xu and Oja, 1993] L. Xu and E. Oja. Randomized hough transform (rht): Basic mechanisms, algorithms, and computational complexities. *CVGIP*, 57(2):131–154, March 1993.

[Yip *et al.*, 1994] Raymond K. K. Yip, Wilson C. Y. Lam, Peter K. S. Tam, and Dennis N. K. Leung. A hough transform technique for the detection of rotational symmetry. *Pattern Recogn. Lett.*, 15(9):919–928, 1994.