Fast Global Reflectional Symmetry Detection for Robotic Grasping and Visual Tracking

Wai Ho Li, Alan M. Zhang and Lindsay Kleeman
Centre for Perceptive and Intelligent Machines in Complex Environments:
Intelligent Robotics
Monash University, Clayton
Melbourne, Australia
{Wai.Li, Alan.Zhang, Lindsay.Kleeman}@eng.monash.edu.au

Abstract

Many objects found in domestic environments are symmetrical. In this paper, a novel method to detect axis of reflectional symmetry in digital images is presented. The proposed approach utilizes image gradients and edge pixel matching to find lines of symmetry. Preliminary results, comparing the symmetry detector with the *Generalized Symmetry Transform* [Reisfeld *et al.*, 1995] is provided.

1 Introduction

Symmetry is one of many attention mechanisms that humans use in everyday life. Many objects in the domestic environment are symmetrical. In general, cups, cans, bowls and boxes have reflectional symmetry about a straight line. The way objects are grasped and manipulated are also related to their axis of symmetry. For example, in the case of a can, grasping is done by applying force on opposite sides of its axis of symmetry, in a manner perpendicular to the axis. This paper describes a reflectional symmetry detector that locates global mirror lines in 2D digital images.

1.1 Related Research

Symmetry has been viewed as an important attentional operator in Computer Vision for several decades. The Generalized Symmetry Transform [Reisfeld *et al.*, 1995] is a multi-scale approach to detect reflectional and radial symmetry in digital images. This transform produces a symmetry map, which is similar to an edge image. Each point on the map consists of the magnitude and direction of symmetry inferred by all other points in the image, calculated at a given scale. At large scale values, the transform can been seen as a global context-free (low level) symmetry operator. With small scale values, it acts in a similar fashion to interest point detectors, such as Harris Corners [Harris and Stephens, 1988]. This transform has been used in human identification [Hayfron-Acquah *et al.*, 2002] and corner detection [Oh and Chien, 2002] applications. Recently, a focused version of the Generalized Symmetry Transform [Choi and Chien, 2004] has been suggested. This version of the transform can recognize image features, such as polygon corners, at a specific angle.

Other global symmetry detection schemes have also been proposed. A clustering approach was suggested by [Labonte *et al.*, 1993]. An algorithm involving Generalized Complex Moments was developed by [Shen *et al.*, 1999], which is capable of finding both reflectional and radial symmetry in 2D high contrast grayscale images. Its computational complexity and ability to operate on natural images have yet to be tested.

Other robotic applications of symmetry include human gait detection [Hayfron-Acquah *et al.*, 2003], completion of occluded shapes [Zabrodsky *et al.*, 1993] and the use of fast radial symmetry [Loy and Zelinsky, 2003] in the detection and tracking of human eyes.

1.2 The Generalized Symmetry Transform

The Generalized Symmetry Transform uses image gradient orientation, a distance-based Gaussian function and image gradient intensity to generate symmetry maps. The symmetry map is essentially an image of symmetry contributions made by all pixel pairs in the original image. Two symmetry maps are produced by the transform, one containing the magnitude of symmetry contribution and the other the phase.

With p_i and p_j being two pixels in the input image, we define a set $\Gamma_g(p)$ as

$$\Gamma_g(p) = \{ \ (i,j) \mid \frac{p_i + p_j}{2} = p \ \} \tag{1}$$

The following distance weight function is used

$$D_\sigma(i,j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{||p_i - p_j||}{2\sigma}} \tag{2}$$

along with a phase weight function

$$P(i,j) = (1 - \cos(\theta_i + \theta_j - 2\alpha_{ij}))(1 - \cos(\theta_i - \theta_j)) \tag{3}$$

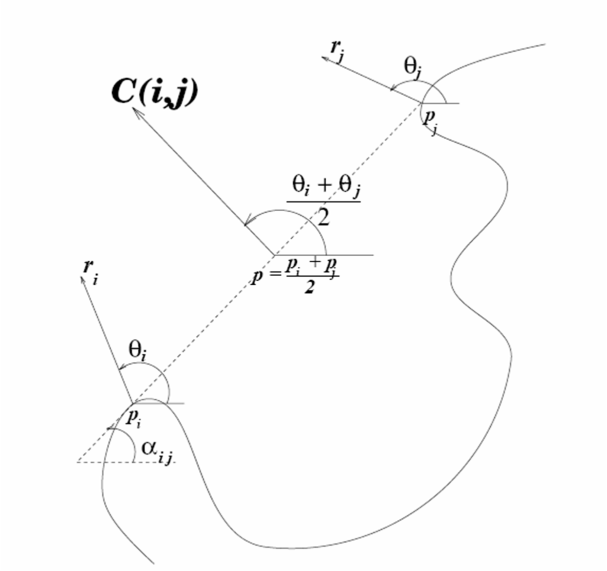


Figure 1: The Generalized Symmetry Transform

The parameters involved are displayed in Figure 1. Note that σ alters the scale at which symmetry calculations take place, thereby allowing for the generation of symmetry maps at different resolutions. The phase function, $P(i,j)$, contains two terms. The first term, $(1-\cos(\theta_i+\theta_j-2\alpha_{ij}))$ is maximized when gradients at p_i and p_j are oriented in a symmetric fashion, similar to that shown in Figure 3(a). The second term prevents the matching of pixels with similar image gradients, such as pixels along a straight line.

The contribution function $C(i,j)$ is defined as

$$C(i,j) = D_\sigma(i,j)P(i,j)r_i r_j \tag{4}$$

r_i and r_j are logarithmic functions of the pixels' gradient intensity.

The contributions of *all* points are used in the generation of the symmetry map. The *symmetry magnitude*, also known as the *isotropic symmetry* is

$$M_\sigma(p) = \sum_{(i,j)\in\Gamma_g(p)} C(i,j) \tag{5}$$

In order to obtain an axis of reflectional symmetry from the map, thresholding followed by a Hough Transform is required.

2 Proposed Method

The proposed approach to symmetry detection is unique in several ways. Firstly, the method was designed with robotic applications in mind. It is not meant to provide a comprehensive coverage of symmetry at all scales, nor will it provide a Gaussian cloud of possible locations

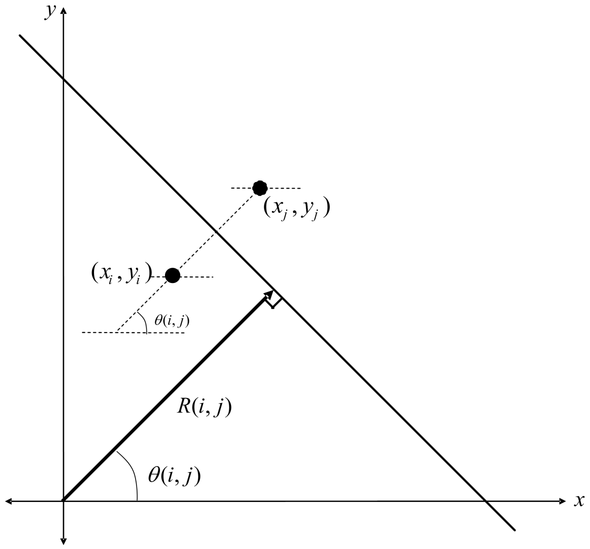


Figure 2: Fast Reflectional Symmetry Detection using Hough Transform. Note that only edge pixels are used by this algorithm

of symmetry. The symmetry detection algorithm utilizes Hough Transform [Duda and Hart, 1972] to quickly process images and identify global lines of reflectional symmetry. The detector returns the axes of reflectional symmetry as straight lines. The algorithm is meant to operate very quickly, and requires less computational resources than the Generalized Symmetry Transform.

2.1 Fast Symmetry Detection using Hough Transform

The input image is first converted into an edge image using an appropriate edge filter. We used the Canny edge filter for this task. Each pair of edge pixels, i and j, votes for a particular r and θ in the Hough transform accumulator. A weight function $W(i,j)$ is used to adjust the Hough accumulation for edge pixels with non-symmetric gradient orientations. The edge points i and j are shown in Figure 2. In order to maximize Hough Space resolution, the origin of the coordinate system is placed at the centre of the image.

The following equation is used to calculate θ. Note that $-\frac{\pi}{2} < \theta(i,j) \leqslant \frac{\pi}{2}$

$$\theta(i,j) = \arctan(\frac{y_j - y_i}{x_j - x_i}) \tag{6}$$

The radius $R(i,j)$ is found using

$$R(i,j) = (\frac{x_i + x_j}{2})\cos[\theta(i,j)] + (\frac{y_i + y_j}{2})\sin[\theta(i,j)] \tag{7}$$

The set Γ_f is the set of edge pixel pairs with the corresponding R and θ shown in Figure 2.

$$\Gamma_f(r,\theta) = \{(i,j) \mid R(i,j) = r,\ \theta(i,j) = \theta\} \qquad (8)$$

Note that only edge pixels are considered, in contrast to the Generalized Symmetry Transform, which processes all image pixels. In order to reject noisy edge pixels and to favour votes made by edge pixels with symmetric image gradients, weighted Hough voting is used

$$H(r,\theta) = \sum_{i,j \in \Gamma_f(r,\theta)} W(i,j) \qquad (9)$$

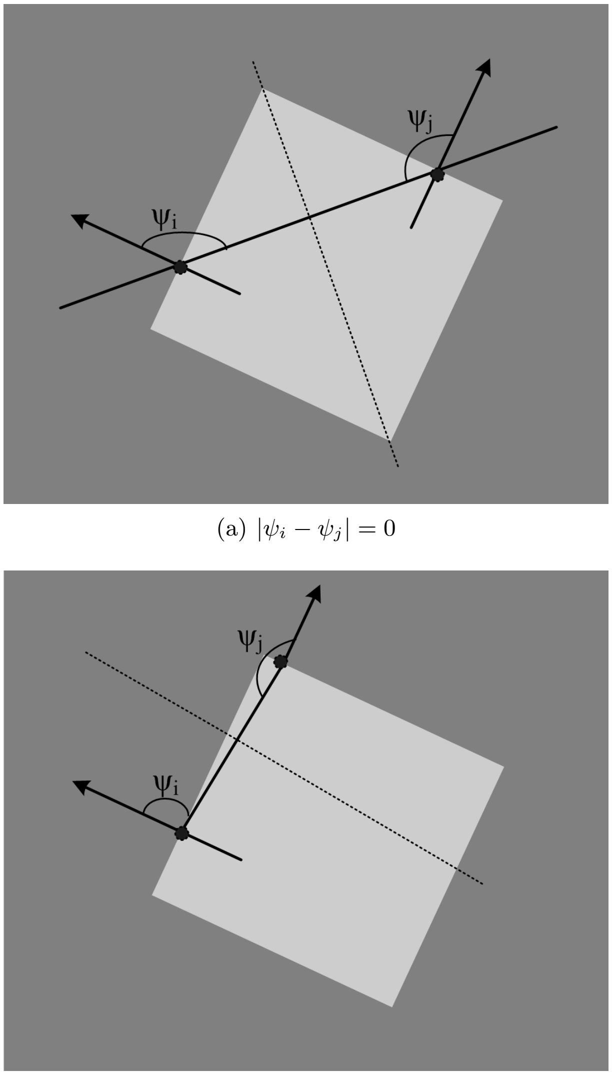
A Gaussian weight function was used in the Hough voting process. The weight $W(i,j)$ is maximized when gradient orientations of pixels are symmetric about their mid point. Figure 3 illustrates the relationship between image gradient angles and the level of symmetry. When angles ψ_i and ψ_j are equal, the level of symmetry is at its maximum. If $|\psi_i - \psi_j| = \frac{\pi}{2}$, there is no symmetry between the edge points. Only the absolute magnitude of angles are used in these calculations. Horizontal and vertical Sobel filtering is used to determine image gradients. The parameter σ in equation 10 is used to adjust the "strictness" of the gradient symmetry criteria. A low sigma will favour edge pairs with high symmetry between the gradient directions of its edge pixels. Using a high sigma has the reverse effect, allowing less symmetric edge pairs to have a greater effect in the voting.

$$W(i,j) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(\psi_i - \psi_j)^2}{2\sigma^2}} \qquad (10)$$

The Hough space is discretized into a fixed number of cells or "bins" in both the R and θ direction. These bins accumulate votes made by the edge pixel pairs. After accumulation, peaks in the Hough accumulator are identified using Non-Maxima suppression. This algorithm locates the Hough bin with the highest value and selects it as a peak. It then *suppresses* all bins around the selected peak to zero. This is repeated until all peaks with a value above a threshold is found. The default suppression neighbourhood is $\frac{1}{10}R_{max}$ by $\frac{1}{10}\theta_{max}$. The suppression threshold is half of the highest peak in the accumulator at the start the suppression, and remains the same through all iterations.

3 Preliminary Results

Generalized Symmetry and Fast Symmetry detection results are presented below. In order to quantify the comparison between these two methods, simple pass-fail trials were used. The test images contained a grey vase with a vertical line of symmetry through its centre point. The location of this line of reflectional symmetry was used as ground truth. If the location of the symmetry line detected with Fast Symmetry was within 1 pixel of ground



(a) $|\psi_i - \psi_j| = 0$

(b) $|\psi_i - \psi_j| \approx \frac{\pi}{2}$

Figure 3: Angles used in weight calculations. Arrows indicate image gradient direction at the edge pixels. Maximum symmetry occurs when $\psi_i = \psi_j$, in (a). (b) will receive a very low voting weight, as $|\psi_i - \psi_j| \approx \frac{\pi}{2}$

truth, the test was successful. Otherwise, the symmetry detection was considered a failure. As Generalized Symmetry does not return a symmetry line, the location with maximum contributed value, that is, the point with maximum *isotropic symmetry*, was used as an indication of detection accuracy. If this point is located within 1 pixel of the real symmetry line, the test was a success.

Three test images were used. The first had a dark shape, the vase, against a light background. The other two images had the same vase set against backgrounds with changing intensity. The tests were repeated after adding Gaussian noise to the input images. All the test images were 64x81 pixels in size.

3.1 Trial Results

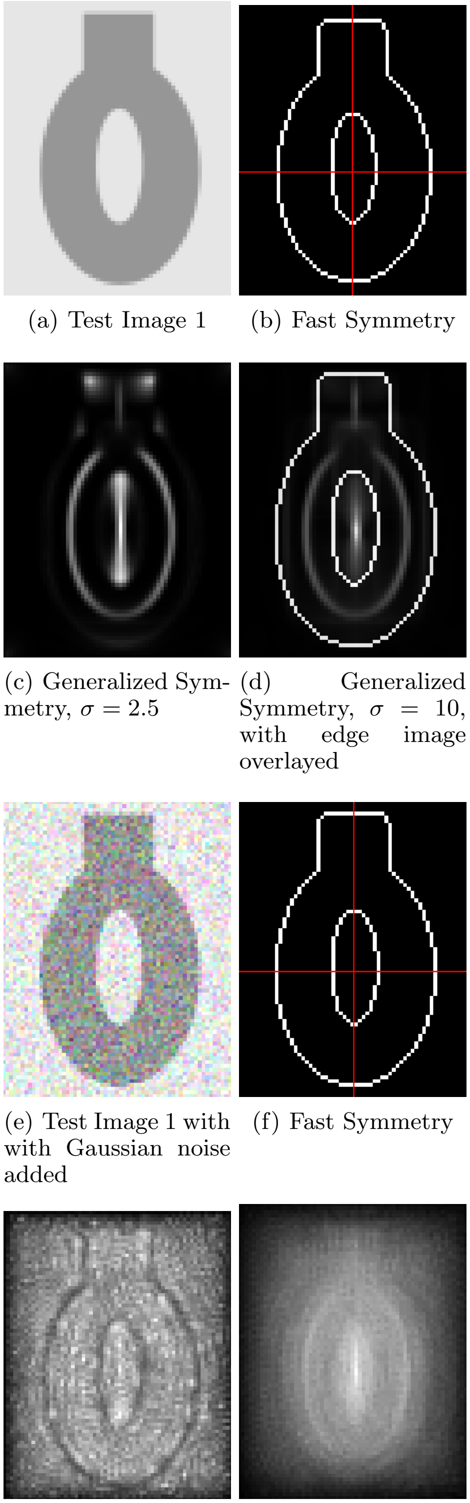
In all test cases, the Fast Symmetry detector was able to find the vertical line of symmetry at the exact, ground truth, location. The Generalized Symmetry Transform was able to find the axis of symmetry in Test Image 1, when no noise was added. Lowering the scale parameter also produced a corner-detecting behaviour, as seen in Figure 4(c) and 7(d). With added noise, the Generalized Symmetry algorithm was only able to detect the axis of symmetry with $\sigma = 10$.

For Test Image 2 and 3, where the background had variations in intensity, the Fast Symmetry detection was sucessful for both images. The Generalized Symmetry algorithm was only able to find the line of symmetry with a σ of 10, for Test Image 3. As seen in Figure 6(d), the line of symmetry found was near the top of the vase, and was not very distinct. With the addition of Gaussian noise, the Generalized approach failed for both images, regardless of the scale factor, σ, used.

Looking at Equation 3, the problem with varying background gradients for the Generalized Symmetry Transform becomes more apparent. The transform was designed to favour opposing image gradients, while rejecting image gradients in the same direction. Basically, the algorithm assumes either light objects on a dark background, or dark objects on a light background. With variations in the background, this leads to zero contributions being made by pixel pairs across the left and right edges of the vase in Test Image 2. The algorithm was still able to find the vertical line of symmetry in Figure 6 as the gradient variations only effected matches that produced symmetry along horizontal lines. That is, the background gradient affected pixel pairs that had pixels on the top and bottom of the vase, not those on the left and right side.

3.2 Comparison Between Fast Symmetry and Generalized Symmetry Detection

In Figure 7, the multi-resolution nature of the Generalized Symmetry Transform can be seen. By varying σ,



(a) Test Image 1

(b) Fast Symmetry

(c) Generalized Symmetry, $\sigma = 2.5$

(d) Generalized Symmetry, $\sigma = 10$, with edge image overlayed

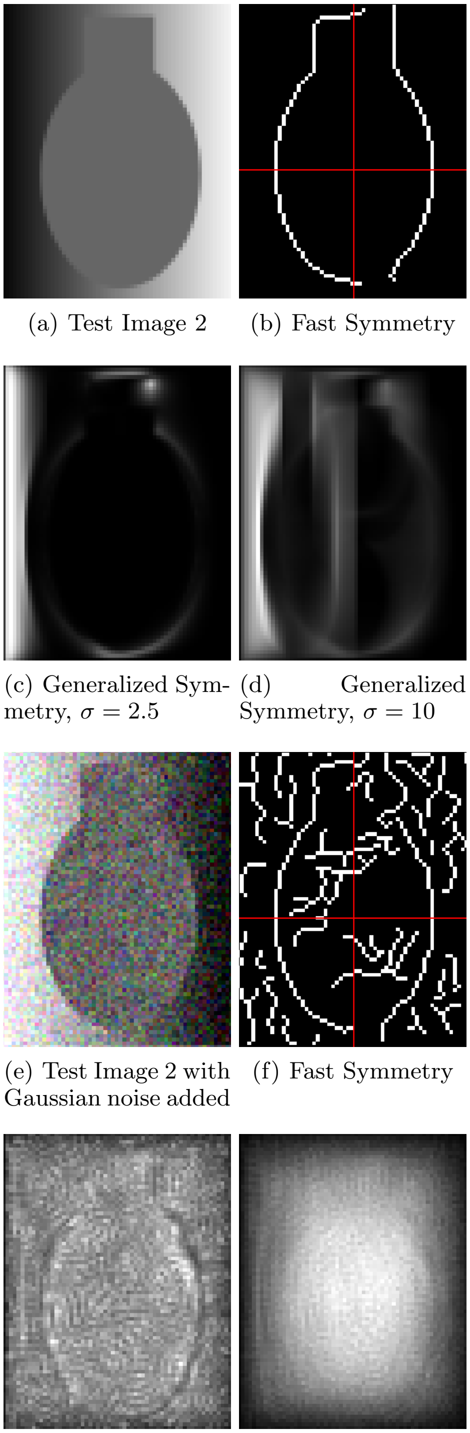
(e) Test Image 1 with with Gaussian noise added

(f) Fast Symmetry

(g) Generalized Symmetry, $\sigma = 2.5$

(h) Generalized Symmetry, $\sigma = 10$

Figure 4: Symmetry detection results for Test Image 1. (b)-(d) contain results for the test image. (e) is Test Image 1 with added Gaussian noise. The noise had $\sigma = 0.1$, with image intensity defined between 0 and 1. (f)-(h) contain detection results for the noisy image. Bright pixels in the Generalized Symmetry results have high levels of detected symmetry



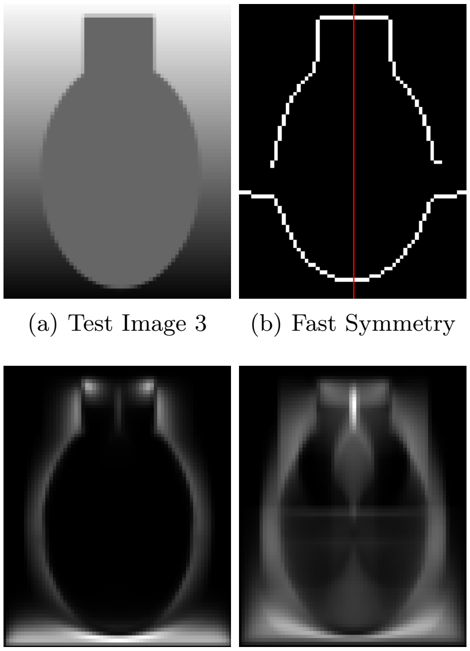
(a) Test Image 2 (b) Fast Symmetry

(c) Generalized Symmetry, $\sigma = 2.5$ (d) Generalized Symmetry, $\sigma = 10$

(e) Test Image 2 with Gaussian noise added (f) Fast Symmetry

(g) Generalized Symmetry, $\sigma = 2.5$ (h) Generalized Symmetry, $\sigma = 10$

Figure 5: Symmetry detection results for Test Image 2. Note the intensity variation in the background of (a). (b)-(d) contain results for the test image. (e) is Test Image 2 with added Gaussian noise. The noise had $\sigma = 0.1$, with image intensity defined between 0 and 1. (f)-(h) contain detection results for the noisy image



(a) Test Image 3 (b) Fast Symmetry

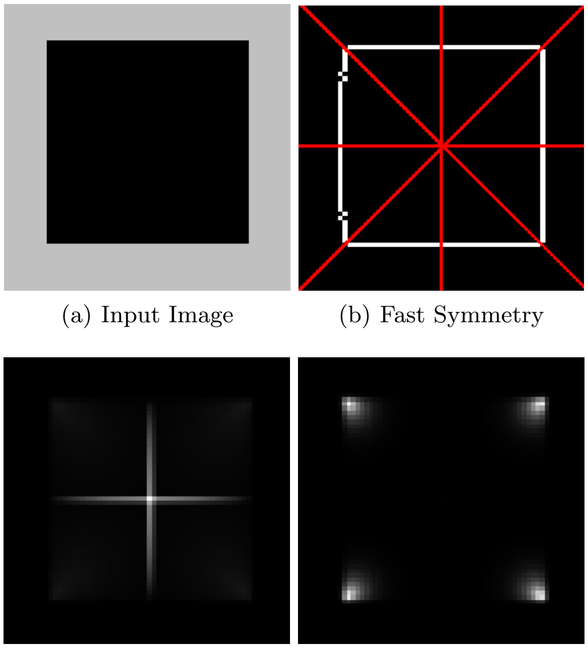
(c) Generalized Symmetry, $\sigma = 2.5$ (d) Generalized Symmetry, $\sigma = 10$

Figure 6: Test image 3 results

corners as well as symmetry lines could be seen on the maps. However, the choice of scale, σ, is not directly obvious from the input image dimensions. Also, diagonal symmetry lines were not found in Figure 7 using the transform unless σ was set to between 9 and 11. Contrary to this, our detector was able to find all four lines of reflectional symmetry. Even though the edge image produced by the Canny filter had several imperfections along the left side of the square, the detector still functioned correctly.

The computational complexity of both symmetry algorithm is $O(n^2)$, with n equal to the number of input pixels. The number of possible pixel pairs is given by $\frac{n(n-1)}{2}$. However, the Fast Symmetry algorithm only operates on edge pixels, while the Generalized Symmetry algorithm operates on all image pixels. The number of edge pixels is generally much smaller than the total number of pixels in an image. Hence, the Fast Symmetry algorithm requires fewer calculations than the Generalized approach. Our approach has combined the post-processing stage of applying Hough Transform to the Symmetry map into the algorithm itself. The calculation of local gradient intensities have been removed by the use of edge images, which removes pixels with low image gradient magnitude.

There was a noticeable difference in the execution time for the two algorithms. The original image in Figure 7 is 64x64 pixels in size. The Generalized Symmetry Trans-



(a) Input Image (b) Fast Symmetry

(c) Gen. Symmetry σ=15 (d) Gen. Symmetry σ=2.5

Figure 7: Symmetry Detection performed on a simple square box. Top Left: Original Image. Top Right: Hough Transform Symmetry Detection Results. Bottom: Generalized Symmetry Transform Results

form has $n = 4096$ for this image. This produced over 8 million unique pixel pairs, a large number for such a small image. With our edge-based detector, the edge image consisted of around 200 pixels. This meant less than 30000 pairs of pixels. Clearly, the generalized method will have serious computational limitations if applied to larger images, especially in time-critical situations, such as visual tracking and robotic grasping.

Both algorithms were implemented as a MATLAB script, not optimized or vertorized. On a host PC with a Pentium4 2.2GHz processor and 1GB of main memory, the Fast Symmetry algorithm only required 5 to 10 seconds to complete for the 64x64 images. The Generalized Symmetry Transform required over three minutes to execute. For larger input images, around 300x300 in size, the Generalized Symmetry Transform took a very long to run under MATLAB, in the order of several hours. The Fast Symmetry algorithm, including Non-Maximal suppression to detect peaks in Hough space, required around 10 to 20 seconds to process these images.

The Fast Symmetry algorithm has also been implemented in C code. This program was tested on images taken with a Firewire (IEEE1394) camera. The scene consisted of symmetrical objects, such as cups and bottles, under non-uniform lighting, with some background clutter. The program was able to perform symmetry detection on these test images in under 1 second. The Canny edge detection as well as the peak finding were included in the timing. By sampling the edge pixels before pairing, similar to the approach used in *Randomized Hough Transform* [Xu and Oja, 1993], the performance of our algorithm can be improved, with some loss in noise robustness. Sampling one-quarter of the edge pixels, the program was able to perform symmetry detection on 640x480 images in roughly 150 milliseconds with no noticable change in detection rate.

3.3 Fast Symmetry Detection on Household Objects

Source images containing objects commonly found in domestic environments were obtained via Google image searches. These images vary in size and complexity. Some of them have poor lighting and produced noisy edge images. Due to their size, symmetry detection using the Generalized Transform was not feasible. As such, only the Fast Symmetry Detection results were computed.

The results are shown in the Appendix.

4 Applications and Future Work

The Fast Symmetry Detection algorithm was designed for use in robotic applications. It will be applied to object tracking and object localization through stereo matching of symmetry lines. The detected symmetry can provide additional information for grasping of objects.

The detector will be employed in a robotic system to locate, manipulate and visually track household objects. The detected symmetry, especially axes of symmetry perpendicular to the table plane, will be used by the robot as a way to identify possible objects in the scene. The experimental platform consists of a pan-tilt capable robot head with colour stereo vision and a PUMA 6-Degree-of-Freedom robot arm with two-fingered end effector. Objects found in the domestic environment, such as cups, bowls, cans and plates will be the test cases in these experiments.

Previously, symmetry has mainly been applied to offline processing domains, such as image registration, 3D model compression, solving for the shape of partially occluded objects and so on. The speed of this algorithm should allow it to be used in real-time applications such as tracking and grasping.

Currently, peak detection in Hough space is a major weakness in the algorithm. As seen in the results when operating on noisy real-world images, too many peaks were found. That led to an excess of symmetry lines, even when σ was decreased to reject noisy matches. Approaches such as hierarachical Hough Transforms, Smoothing of Hough space and Gaussian dispersion of votes may improve the peak finding results.

In terms of pre-processing, better edge filtering will enhance the performance of the algorithm. Fewer noisy edges mean more accurate results and fewer pixels to process, which also leads to faster execution times. The removal of edges before performing the Hough voting is also a possibility. For example, if the robot already knows that there are multiple objects placed on a table, then it can remove all edges oriented parallel to the table, and just look for vertical axes of symmetry using the remaining, non-horizontal edges. Perspective projection also has a negative impact on symmetry detection. As such, the removal of perspective effects is also a valuable pre-processing step. This is only a problem when objects are close to the camera, and of sufficient size, to be affected by perspective effects.

5 Conclusion

A viable alternative to the The Generalized Symmetry Transform has been proposed. The algorithm can detect reflectional lines of symmetry very quickly, even for noisy input images. The algorithm is also able to find lines of symmetry for large images, containing background intensity variations. While the Generalized transform requires a scale factor to be set before use, the Fast Symmetry Detector is able to detect reflectional symmetry across the entire image in a scale free manner. However, our algorithm cannot detect small, locally symmetric features such as the corners of a box and symmetry in fine texture. The proposed algorithm is more computationally efficient than Generalized Symmetry Transform. After further research, it will be applied to time-critical applications, such as visual tracking and robotic grasping.

Acknowledgements

The work described in this paper was supported by the ARC Centre for Perceptive and Intelligent Machines in Complex Environments (PIMCE). We would like to thank Geoff Taylor for his help with locating symmetry-related references and implementation advice.

References

[Choi and Chien, 2004] I. Choi and S.I. Chien. A generalized symmetry transform with selective attention capability for specific corner angles. *SPLetters*, 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. *Commun. ACM*, 15(1):11–15, 1972.

[Harris and Stephens, 1988] Chris Harris and Mike Stephens. A combined corner and edge detector. In M. M. Matthews, editor, *Proceedings of the 4th ALVEY vision conference*, pages 147–151, University of Manchester, England, September 1988.

[Hayfron-Acquah *et al.*, 2002] James B. Hayfron-Acquah, Mark S. Nixon, and John N. Carter. Human identification by spatio-temporal symmetry. In *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 1*, page 10632, Washington, DC, USA, 2002. IEEE Computer Society.

[Hayfron-Acquah *et al.*, 2003] James B. Hayfron-Acquah, Mark S. Nixon, and John N. Carter. Automatic gait recognition by symmetry analysis. *Pattern Recogn. Lett.*, 24(13):2175–2183, 2003.

[Labonte *et al.*, 1993] F. Labonte, Y. Shapira, and P. Cohen. A perceptually plausible model for global symmetry detection. In *ICCV93*, pages 258–263, 1993.

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

[Oh and Chien, 2002] Hyun-Hwa Oh and Sung-Il Chien. Exact corner location using attentional generalized symmetry transform. *Pattern Recogn. Lett.*, 23(11):1361–1372, 2002.

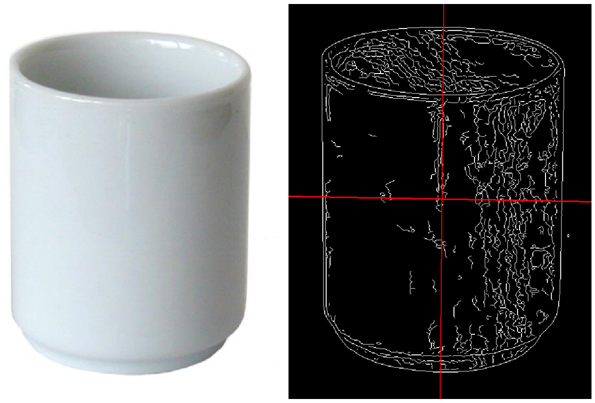
[Reisfeld *et al.*, 1995] Daniel Reisfeld, Haim Wolfson, and Yehezkel Yeshurun. Context-free attentional operators: the generalized symmetry transform. *Int. J. Comput. Vision*, 14(2):119–130, 1995.

[Shen *et al.*, 1999] Dinggang Shen, Horace Ho-Shing Ip, Kent K. T. Cheung, and Eam Khwang Teoh. Symmetry detection by generalized complex (gc) moments: A close-form solution. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):466–476, 1999.

[Xu and Oja, 1993] Lei Xu and Erkki Oja. Randomized hough transform (rht): basic mechanisms, algorithms, and computational complexities. *CVGIP: Image Underst.*, 57(2):131–154, 1993.

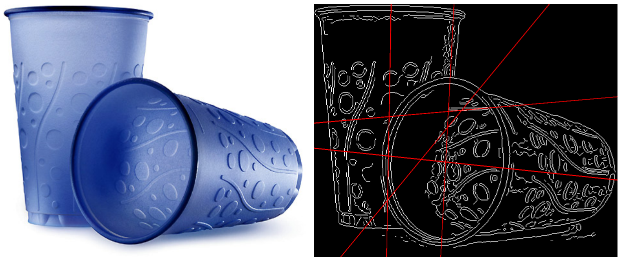
[Zabrodsky *et al.*, 1993] H. Zabrodsky, S. Peleg, and D. Avnir. Completion of Occluded Shapes using Symmetry. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 678–679, New York, June 1993.

Appendix: Fast Symmetry Detection on Household Objects



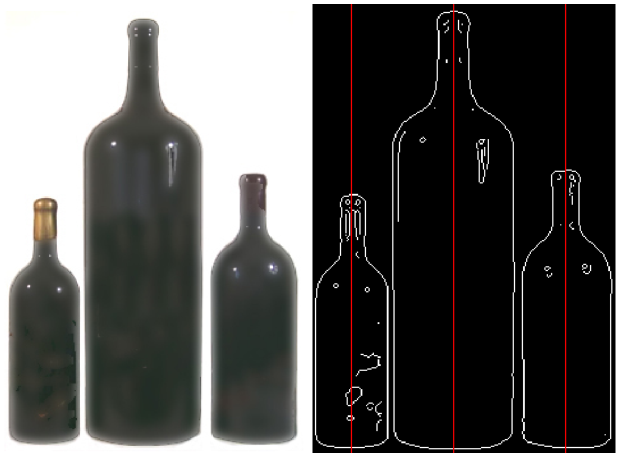
(a) Input image (b) Fast Symmetry

Figure 8: Symmetry Detection on a White Cup. (a) Original Image. (b) Symmetry Lines detected by Fast Symmetry algorithm



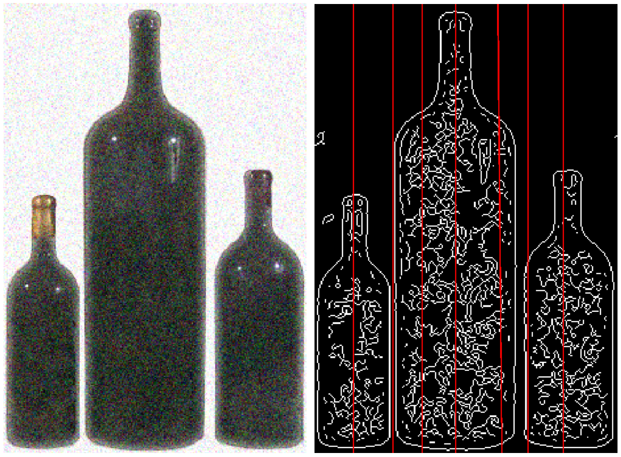
(a) Input image (b) Fast Symmetry

Figure 9: Fast Symmetry Detection on two Highly Textured Objects. The 45 degree line was inter-object symmetry detected between the bottom edge of the right cup, and the left edge of the upright cup



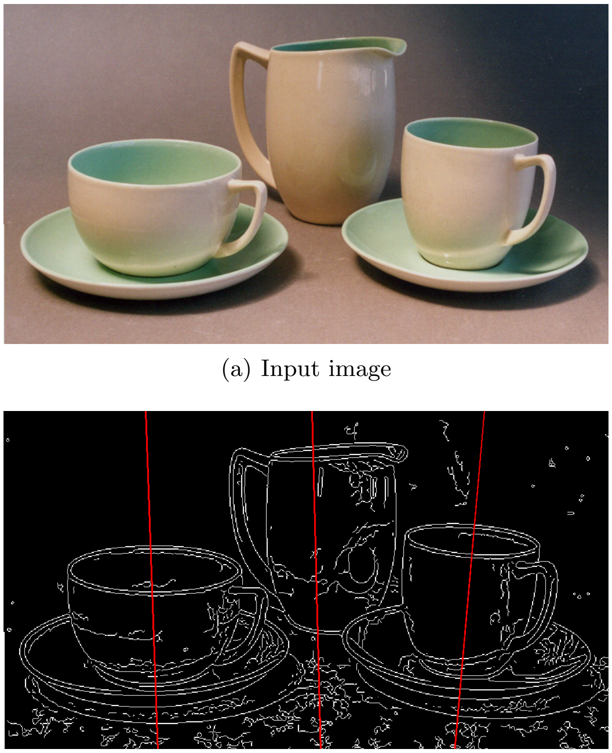
(a) Input image (b) Fast Symmetry

Figure 10: Symmetry Detection on three bottles placed side-by-side. The results was produced from a Hough space peak search with $-5^o < \theta < 5^o$



(a) Input image with Gaussian Noise (b) Fast Symmetry

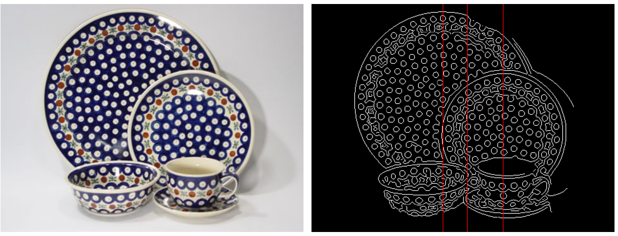
Figure 11: Symmetry Detection on an image with noise. The results were produced from a Hough space peak search with $-5^o < \theta < 5^o$



(a) Input image

(b) Fast Symmetry

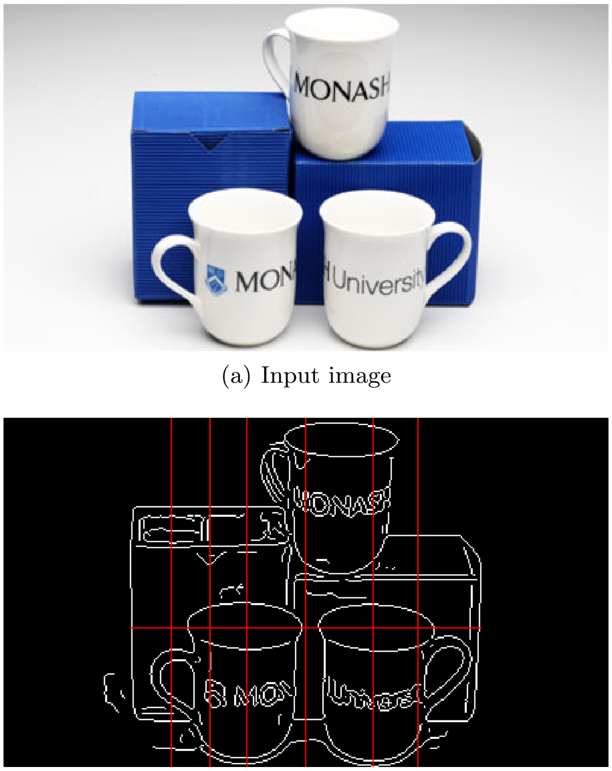
Figure 12: For this image, the suppression size used was doubled to remove some inter-object symmetry.



(a) Input image

(b) Fast Symmetry

Figure 13: Fast Symmetry Detection again on Highly Textured Objects. The results were produced from a Hough space peak search with $-5^o < \theta < 5^o$



(a) Input image

(b) Fast Symmetry

Figure 14: Symmetry Detection on a scene with Multiple cups (a) Original Image. (b) Symmetry Lines detected by Fast Symmetry algorithm