

# Fast Radial and Bilateral Symmetry Detection Using Inverted Gradient Hash Maps

R. Gonzalez<sup>1,\*</sup>, and L. Lincoln<sup>1</sup>

<sup>1</sup>School of ICT, Griffith University, Queensland, Australia

## Abstract

This paper presents a fast and novel algorithm for both radial and bilateral symmetry detection based on inverted gradient hash maps (IGHMs). A hash map is an associative array that stores image gradient magnitudes and orientations in the form of an inverted index. This mapping of image gradients to their locations permits points of interest to be located very rapidly without needing to search through the image. Unlike other symmetry operators it is able to detect symmetries without needing the range of the symmetry to be known a priori. It can also easily detect large-scale symmetry. The method is described and experimentally evaluated against existing methods for both radial and bilateral symmetry detection.

**Keywords:** Symmetry Detection, Reflective, Radial, Bilateral, Mirror.

Received on 13 May 2016, accepted on 07 September 2016, published on 06 March 2017

Copyright © 2017 Gonzalez *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/eai.6-3-2017.152336

## 1. Introduction

The detection of symmetry is a fundamental process within many image-processing applications. Playing an important role in visual perception, it has a variety of uses in computer vision applications such as being an attentional operator, and a detector of man made and natural objects such as human faces. It is also useful for shape representation and characterization. While different kinds of symmetry can be detected, the most common are radial (rotational) and bilateral (reflective or mirror) symmetries.

Perhaps the best-known symmetry detector is Reisfeld's generalized symmetry transform [1]. This can detect bright and dark, isotropic or radial symmetries. Its main drawback is its computation complexity, having an order of  $NK^2$ , where  $N$  is the number of pixels in an image and  $K$  is the kernel size. A variety of other bilateral and radial symmetry detectors have been proposed in the literature such as those utilizing the Hough transform [2, 3, 4] which have an order of  $KBN$ , where  $B$  is the number of angular steps. Faster algorithms have been proposed for detecting only radial symmetry such as that of Loy [5] that has an order of  $KN$ .

While Reisfeld's algorithm requires 259 Mflops for a 30x30 kernel and a 521x512 pixel image, the Hough based methods require around 34 Mflops, and Loy's approach requires between 8-19 Mflops [6]. Detecting bilateral or radial symmetry across an entire image requires a kernel that is the same size as the image. In this case the required computing time becomes very large. A major limitation with many of these methods is that the scale or range of the symmetry to be detected needs to be known a priori. If it is not known, an iterative approach is required to find the symmetry using a different kernel size for each pass. This significantly increases computational requirements.

Typical symmetry detection operates by either searching for matching gradients within a kernel using the image's intrinsic Cartesian space or by using a one to many voting scheme using a parameterized 'Hough' accumulator space. In contrast, the method proposed in this paper operates purely in gradient space and avoids searching by using an inverted index. Until recently, inverted indices have only been used in for speeding up retrieval in databases [7]. Yet the concept of an inverted gradient space representation presents certain advantages for algorithms that exploit image gradients. Most importantly, by eliminating the need to

\*Corresponding author. Email: r.gonzalez@griffith.edu.au

search for matching gradients within a kernel, it can reduce processing time by up to two orders of magnitude.

In this paper a method for symmetry detection via an inverted gradient space representation using hash maps is presented in section 2. Experimental results for the performance of this algorithm relative to existing methods are presented in section 3 and conclusions in section 4.

## 2. Method Description

This section begins with a definition of an Inverted Gradient Hash Map (IGHM). Section 2.2 presents the algorithm for detecting general symmetry directly from the IGHM. Section 2.3 discusses the detection of bilateral and radial symmetry. A complexity analysis of the algorithm is presented in Section 2.4.

### 2.1. Inverted Gradient Hash Maps

A typical image  $f$ , is a mapping of image coordinates  $(x,y)$  to pixel intensity values  $i$ :

$$f: (x, y) \rightarrow i \quad \text{where } [x, y] \in \mathbf{N}^2 \quad (1)$$

The derivative of the image intensity at a given coordinate  $(x,y)$  gives rise to the local gradient and is defined by the magnitude and orientation  $\{m, \theta\}$  at that coordinate:

$$m_{x,y} = \sqrt{\left(\frac{\partial}{\partial x} i_{x,y}\right)^2 + \left(\frac{\partial}{\partial y} i_{x,y}\right)^2} \quad (2)$$

$$\theta_{x,y} = \arctan\left(\frac{\partial}{\partial x} i_{x,y}, \frac{\partial}{\partial y} i_{x,y}\right)$$

A mapping of image coordinates to their corresponding gradients is known as the gradient image  $g$ :

$$g: (x, y) \rightarrow \{m, \theta\} \quad (3)$$

A reverse mapping from the image gradients to image coordinates is known as an inverted gradient image  $h$ .

$$h: \{m, \theta\} \rightarrow (x, y) \quad (4)$$

While there is a single gradient for each image coordinate, there may be many image coordinates that have the same gradient. Thus, the inverted gradient image cannot be simply stored as a two dimensional array. Instead the inverted gradient image is best stored as a hash map where collisions in  $\{m, \theta\}$  are resolved via chaining. This hash map is simply a two dimensional array of lists. These lists are indexed by the gradient magnitude and orientation and store the coordinates of the pixels having the indicated local gradient as depicted in Figure 1.

The size of the two-dimensional hash map is determined by the desired angular and scalar resolution used for the values  $(m, \theta)$ . Typically  $m \rightarrow 0..255$  and  $\theta \rightarrow 0..360$  in one degree increments.

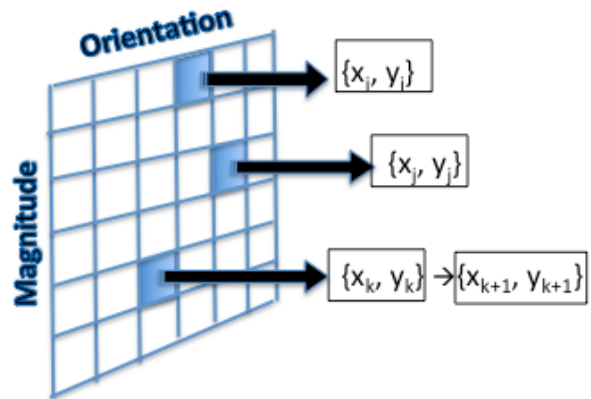


Figure 1. Inverse Gradient Hash Map Data Structure

### 2.2. Symmetry Detection

In Reisfeld's transform a full search of the image is required to find potentially contributing pairs of pixels within a given range defined by the kernel size. Each pixel is checked multiple times. This is avoided using an inverse gradient hash map. A single pass is made through the image visiting each pixel only once in order to populate the hash map. Once the hash map is formed all of the relevant pairs of gradients can be found via a simple lookup and their contribution assessed in a single pass.

The algorithm proceeds by simply looking up all the pairs of pixels having similar gradient magnitudes and opposing orientations (i.e. are rotated by approximately 180 degrees). A margin of up to +/- 45 degrees is used to include gradients that partially contribute to symmetry. This lookup results in two lists containing the coordinates of all the pixels matching the gradient criteria. A simple threshold can be used to filter out pixels below some predetermined noise floor. The contribution of each pair of pixels in these two lists to the symmetry at the given orientation can then be calculated. Thus symmetry in specific directions can be readily determined or if desired, for all directions.

In listing 1, a pseudo code description of the basic algorithm is given. The two outer two loops iterate over the entire hash table using indices  $a$  and  $m$ . Magnitudes below the noise floor  $thres$  are ignored. The next two inner loops using indices  $b$  and  $n$  only iterate respectively over the range of  $(a + 45 + 180)$  and  $(a - 45 + 180)$  degrees and also  $(m - 5)$  to  $(m + 5)$  to allow for noise in the gradient magnitude. For each pair of indices  $(a, m)$  and  $(b, n)$  two lists containing the relevant points are referenced using two pointer variables ptrA and ptrB. For each pair of coordinate points,  $i$  and  $j$  in these lists, the contribution to the symmetry is calculated (as further described below) and accumulated if it is within the target range of influence.

Listing 1. Basic Symmetry Detection Algorithm

```

function symmetry(IGHM, output, maxdist, thres)
Begin
  //-- visit all gradients in hash map --
  Foreach a = gradient orientations from 0...360
    Foreach m = gradient magnitudes > thres
      //-- get contributing gradient points --
      For (b = a-45; b < a+45; b++)
        For (n = m-5; n < m+5; n++)
          var c = (b + 180) % 360
          var ptrA = &(IGHM[a][m])
          var ptrB = &(IGHM[c][n])
          //-- walk coordinate lists --
          For (i=0; i < ptrA->len; i++)
            For (j=0; j < ptrB->len; j++)
              var Dij = Distance(ptrA[i], ptrB[j])
              var Cij = Calc(a,b,ptrA[i], ptrB[j])
              var x = (ptrA[i].x + ptrB[j].x) / 2
              var y = (ptrA[i].y + ptrB[j].y) / 2
              If (Dij < maxdist) Output[x][y] += Cij
            Endfor
          Endfor
        Endfor
      Endfor
    Endforeach
  Endforeach
End

```

The contribution  $C_{ij}$  of each pair of pixels  $p_i$  and  $p_j$  is a function of the magnitude of the nominated gradient  $m_i$  and  $m_j$ , the deviation of the gradient orientation to the suggested axis of symmetry and optionally, the distance between contributing pixels  $D_{ij}$ . This contribution is assigned to the halfway point  $q_{ij}$  between the contributing pixel pair. A two-dimensional accumulator  $C$  is used to store the contribution of each pair at the halfway points.

Given a pair of contributing pixels  $p_i$  and  $p_j$ , as depicted in figure 2, the first step is to determine the orientation of the suggested axis of symmetry  $\phi_{ij}$ . This is calculated as the normal to the vector connecting  $p_i$  and  $p_j$ , thus:

$$\phi_{ij} = \arctan 2 \left( \frac{p_i(y) - p_j(y)}{p_j(x) - p_i(x)} \right) \quad (5)$$

Next the angular difference between the orientation of the suggested axis of symmetry  $\phi_{ij}$  and the orientation of the gradients  $\theta_i$  and  $\theta_j$  at each point is found, using a formulation that avoids costly transcendental functions.

$$\Delta_i = \left| \pi - \left| \phi_{ij} - \theta_i \right| - \pi \right| \quad (6)$$

$$\Delta_j = \left| \pi - \left| \phi_{ij} - \theta_j \right| - \pi \right|$$

The contribution of the gradients  $p_i$  and  $p_j$  becomes

$$C_{ij} = \frac{m_i \cdot m_j}{1 + |\Delta_i - \Delta_j|} \quad (7)$$

To emulate the behavior of traditional symmetry detection algorithms a fixed threshold,  $maxdist$ , can be applied to the Euclidean distance  $D_{ij}$ . This will limit the range of influence that gradients of the contributing pixels can exert by ignoring the contributions of points beyond this range limit.

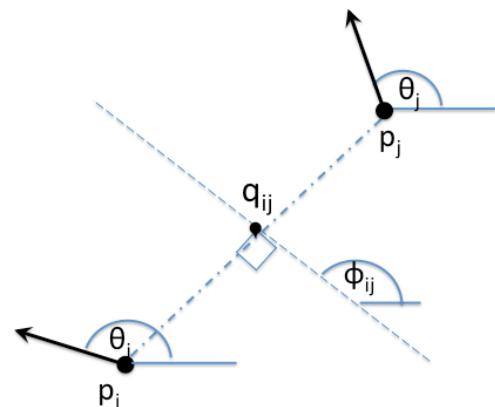


Figure 2. Geometry of points contributing to symmetry.

## 2.3 Bright, Dark Bilateral & Radial, Symmetry

The algorithm presented in the foregoing discussion does not differentiate between the different types of symmetry. It can be further specialized to detect radial and bilateral symmetry as well as bright and dark symmetries.

The basic algorithm detects symmetry by evaluating all potential gradients pairs irrespective of whether their orientation is directed towards or away from each other. It is possible to restrict this to detecting only symmetry that pertains to bright objects on a dark background (bright symmetry) or dark object on a bright background (dark symmetry). To do this the orientation of the gradient relative to the halfway point needs to be determined. First the angle  $A_i$  from each point  $p_i$  to the halfway point  $q_{ij}$  is found and similarly for  $A_j$ .

$$\begin{aligned} A_i &= \arctan\left(\frac{q_{ij}(y) - p_i(y)}{p_i(x) - q_{ij}(x)}\right) \\ A_j &= \arctan\left(\frac{q_{ij}(y) - p_j(y)}{p_j(x) - q_{ij}(x)}\right) \end{aligned} \quad (8)$$

These values are next compared to the gradient orientation  $\theta_i$  and  $\theta_j$  at  $p_i$  and  $p_j$  to see whether the gradients are aligned towards or away from  $A_i$  and  $A_j$ .

$$\varphi_i = |\theta_i - A_i| \quad \text{and} \quad \varphi_j = |\theta_j - A_j| \quad (9)$$

Depending on the value of  $\varphi_i$  and  $\varphi_j$  the contribution  $C_{ij}$  for that pair is retained or set to zero. For bright symmetry the following relationship holds:

$$C_{ij} = \begin{cases} 0 & (\varphi_i < 90 \vee \varphi_i > 270) \wedge (\varphi_j < 90 \vee \varphi_j > 270) \\ C_{ij} & \text{otherwise} \end{cases} \quad (10)$$

For dark symmetry the following relationship is used

$$C_{ij} = \begin{cases} 0 & (\varphi_i > 90 \wedge \varphi_i < 270) \wedge (\varphi_j > 90 \wedge \varphi_j < 270) \\ C_{ij} & \text{otherwise} \end{cases} \quad (11)$$

The differentiating feature between radial and bilateral symmetry is that radial symmetry has a focal point whereas the bilateral case has an axis of symmetry. Accordingly the detection of radial symmetries simply comes down to detecting strong isolated peaks and detecting bilateral symmetries requires finding axes of symmetry by means of line detection.

Most images will have a large number of weak, local symmetries that will be detected. These are insignificant

compared to the detected values from any global symmetry that may be present but result in a cloud of undesirable noise. Additionally, a single image may contain a number of strong overlapping symmetries that will complicate the separate identification of the symmetry.

In the simplest case of radial symmetry detection, the noise can simply be ignored and the value and position of the maxima identifies the amount and location of any symmetry. Once a peak is found a simple neighbourhood search is performed to check its isolation and confirm that it corresponds to radial symmetry. Given a peak at  $(x_p, y_p)$  it is considered isolated if the magnitudes of its neighbours within a given distance are below a given threshold level relative to the magnitude of the peak. If  $I(a, b)$  is the intensity of the symmetry map at location  $a, b$  then for a given distance  $d$  and threshold  $t = 0.1$ , a peak is considered isolated if the following expression is true:

$$x_p, y_p \Leftrightarrow \sum_{i=x-d}^{x+d} \sum_{j=y-d}^{j+d} \begin{cases} i, j; & I(i, j) > t \cdot I(x_p, y_p) \\ 0; & \text{otherwise} \end{cases} \quad (12)$$

If the peak is not isolated, it suggests bilateral symmetry for which it is necessary to discover how the peak is related to other peaks in the image. Suitable line detection algorithms for this purpose include the traditional Hough Transform or the IGHM method for fast line detection [8] that is at least an order of magnitude faster than the Hough approach. Using the IGHM approach pixels contributing to lines of any given orientation are stored within a single column. As individual pixels may contribute to lines with different y-offsets, pixel contributions can be apportioned to the correct potential lines. Potential lines are described in terms of their orientation  $\theta$  and the shortest perpendicular distance  $R$  from the infinite extension of the line to a given origin. The perpendicular distance  $R_i$  from this line to the origin  $(x_0, y_0)$  where  $m = \tan(\theta)$  is given by equation (13):

$$R_i = \frac{|-x_0 m + y_0 - y_i + x_i m|}{\sqrt{m^2 + 1}} \quad (13)$$

All collinear points of any given orientation in the IGHM will share the same distance to the origin. Thus lines are detected as the all the points with common orientation and distance to a given origin. The location, extent and orientation of detected lines give the axes of any bilateral symmetries that may be present in the image. Ideally, only the most significant lines are considered, being those with the greatest support in terms of extent and magnitude.

## 2.4 Complexity Analysis

From the foregoing discussion one can observe that the computational complexity of this method, given the three sets of two loops in the basic algorithm is in the order of:

$$(A * M) * (A/4 * 10) * (La * Lb) = k * A^2 * M * La * Lb \quad (14)$$

Here  $A$  and  $M$  are respectively the number of orientation and magnitude bins used in the hash map and are a function of the angular and scalar resolution used for the values  $(m, \theta)$ . The constant  $k$  for a typical implementation equals 2.5, but could be reduced if desired.  $La$  and  $Lb$  are the lengths of the two coordinate lists and are related to the load factor of the hash map. The load factor will be a function  $A$  and  $M$  relative to the image size in pixels  $N$ . Typically the average load factor  $L_f$  is defined as:

$$L_f = N / (A * M) \quad (15)$$

Since on average the lengths of the two lists  $La$  and  $Lb$  will be equal to the load factor the following substitutions can be made to simplify the calculation of (12):

$$\begin{aligned} A^2 * M * L_f * L_f \\ = A^2 * M * N^2 / (A * M)^2 \\ = N^2 / M \end{aligned} \quad (16)$$

Although the worst-case complexity of this algorithm will never be greater than  $N^2$ , in practice this upper bound will never be approached unless the hash map only a single bin for the gradient magnitude, which defeats the purpose of using a two dimensional hash map. The complexity using a 360 x 256 cell hash map is unlikely to ever exceed  $N^{1.8}$  as can be seen from Table 1, for the different size images:

Table 1. Computational complexity at various sizes.

Image Pixels	Complexity
256 x 256	$N^{1.5}$
512 x 512	$N^{1.56}$
1024 x 1024	$N^{1.6}$
2048 x 2048	$N^{1.65}$
4096 x 4096	$N^{1.67}$
8192 x 8192	$N^{1.69}$
1,000,000 x 1,000,000	$N^{1.8}$

This theoretical complexity is significantly better than the  $NK^2$  complexity of Reisfeld's method and the KBN complexity of the Hough based methods. It is on par with Loy's radial symmetry transform. This result is confirmed experimentally in section 3.3.

In addition to the symmetry calculation the cost of generating the hash map and peak picking needs to be considered. As this is a linear operation with a complexity of  $N$  it makes fairly small contribution on the total complexity.

### 3. Experimental Results

Results were obtained for three categories of experiments. In section 3.1 the performance of the IGHM method for detection of bilateral symmetry is presented and compared

to existing methods. Section 3.2 presents the results of radial symmetry detection and contrasts this with the performance Loy's fast radial symmetry transform. Finally, processing time results are presented in section 3.3.

#### 3.1. Bilateral Symmetry

Bilateral symmetry detection experiments were run using a variety of images. For ease of comparison, a version of the "Cards, keys and hand" image from Reisfeld's paper was used with the resulting symmetry maps shown in figure 3 for Reisfeld's method in the centre and the IGHM method on the right. A kernel size of 16 was used for Reisfeld's method and the range of influence of contributing gradients in the IGHM method was also limited to 16. As can be observed the IGHM approach is better at localizing the symmetry of a wider range of sizes and suppressing noise.

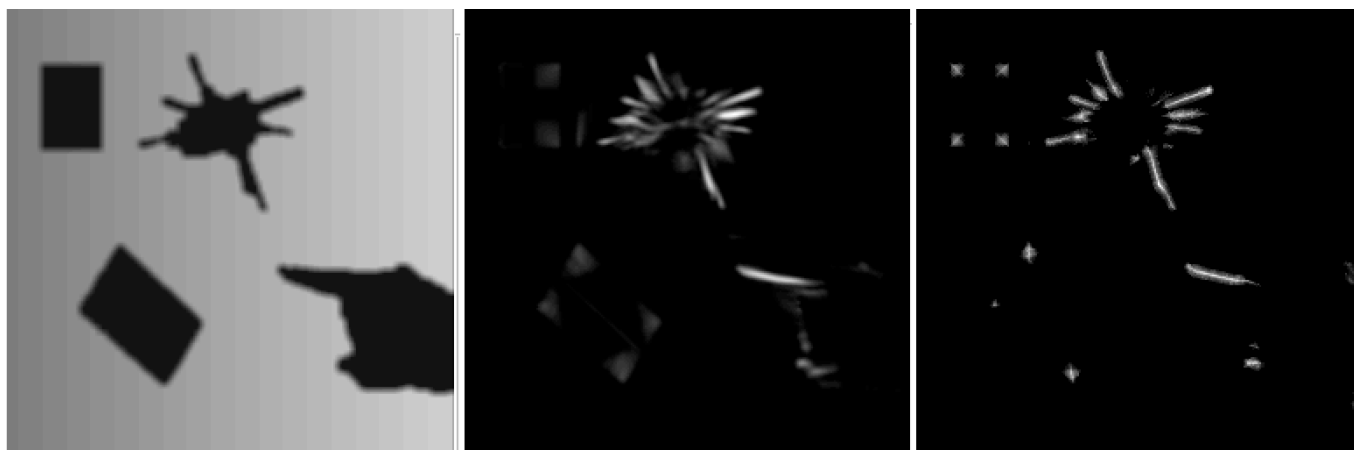
While symmetry detection algorithms typically work well for simple, small-scale symmetry such as that in figure 3, the real challenge is for detecting image wide symmetry. For this experiment the well-known Lena image [9] was first mirrored horizontally and then vertically around a central axis. This is shown in figure 4 with the originals on the left, the negative of the IGHM symmetry maps in the middle and the detected axes of symmetry shown in the right side images. No limits were placed on the range of influence of contributing gradients. Other images of natural and artificial objects such as human faces, a handbag and a geometric drawing are from the CVPR 2013: Symmetry Detection from Real World Images competition dataset [10] were also evaluated with successful detection of the bilateral symmetry as depicted in figure 5.

To compare the ability of the IGHM method to detect image wide symmetry relative to Reisfeld's method, this operator was applied with a kernel size of 256 to the mirrored Lena image used in figure 4. As the results in figure 6 show, the Reisfeld operator (right side) completely fails to identify the global bilateral symmetry in the image.

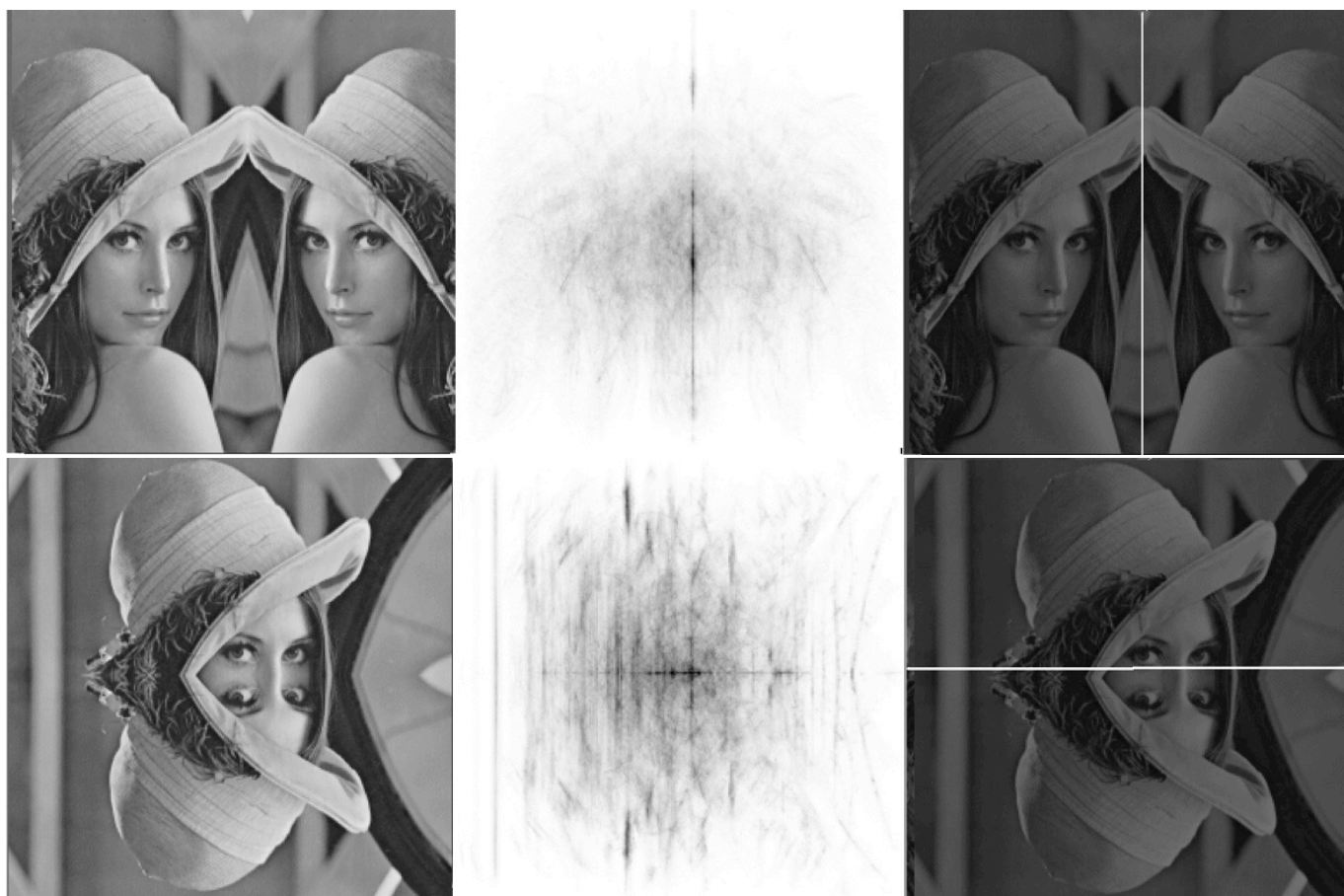
#### 3.2. Radial Symmetry

The accuracy of IGHM based radial symmetry detection was evaluated against a wide range of natural images. These images are from the CVPR 2013: Symmetry Detection from Real World Images competition dataset [10]. Some images were relatively simple and others more complex. Some exhibited poor contrast between the symmetrical objects and their backgrounds. The radial symmetry in some of the images such as the palm tree, the flower, the fireworks, the camouflaged fish in the sand and the gyroscopic monument is not perfectly defined yet is still accurately detected. In other images, there are multiple symmetries both radial and bilateral. These test images and the negatives of their resulting symmetry maps are shown in figure 7. For this experiment no limits were placed on the range of influence of contributing gradients. As can be observed, being a global method, the IGHM approach is by default biased towards detecting large-scale symmetry. It can however be biased to

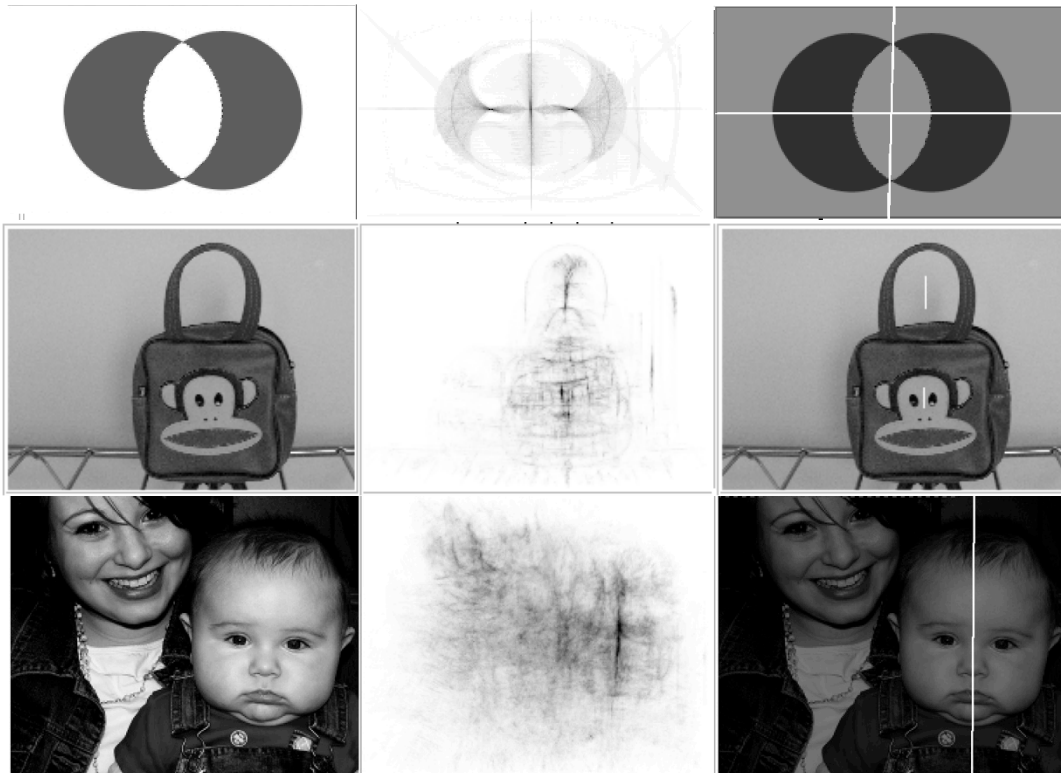
detecting small-scale symmetry by adjusting the influence distance of the image gradients as discussed in section 2.2.



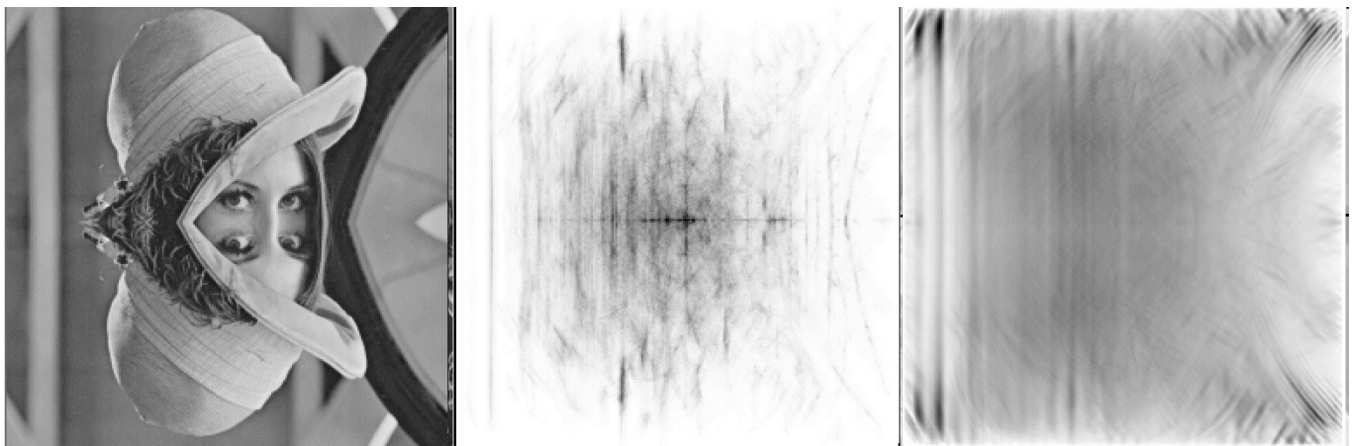
**Figure 3.** Bilateral symmetry with the original image on the left, symmetry map produced by Reisfeld's method using a kernel size of 16 in the middle and the IGHM based method with a Dij of 16.



**Figure 4.** Symmetry of mirrored images with the originals on the left, detected symmetry in the centre and the resulting axes of symmetry superimposed over the original on the right.



**Figure 5.** Bilateral symmetry with the original images on the left, resulting symmetry map in the centre and the resulting axes of symmetry superimposed over the original on the right.



**Figure 6.** Comparison of large field symmetry detection. The original is on the left, the and the symmetry maps produced by the IGHM and Reisfeld's methods respectively in the middle and the right side.

In a second set of experiments the accuracy of the IGHM method was evaluated against that of Loy's state of the art, fast radial symmetry transform. Since that operates over a given range value, a number of different range values were used. These results are shown in figures 8a-c. The original is on the left and then towards the right side are the negatives of the resulting symmetry maps for pixel ranges of 8, 32, 64 and 128. It can be observed that most of the time Loy's method completely fails at detecting the symmetry irrespective of what range is selected. This is generally true

for the more complex images shown in figure 8a. Figure 8b is the only one with correct detection of radial symmetry. However it is the symmetry of the of the bicycle wheel hub using a range of 8 pixels rather than the symmetry of the wheel itself that has been detected. Figure 8c shows that even for simple images when significant peaks are present in the resulting symmetry maps of simpler they are not located at the correct focal points. It can be noted that Loy's fast radial symmetry transform can only reliably detect small circular areas and is hence more of a circular area detector

than a symmetry detector. This is evidenced by Loy's paper that presents results for the detection of small circular regions within images only. While it can do so rapidly, the size of the symmetrical objects to be detected must be known apriori, otherwise their detection will completely fail.

In contrast the IGHM method not only accurately detects the focal point of any radial symmetries present within an image, but does so just as quickly and without any prior knowledge about the sizes of symmetrical objects therein.

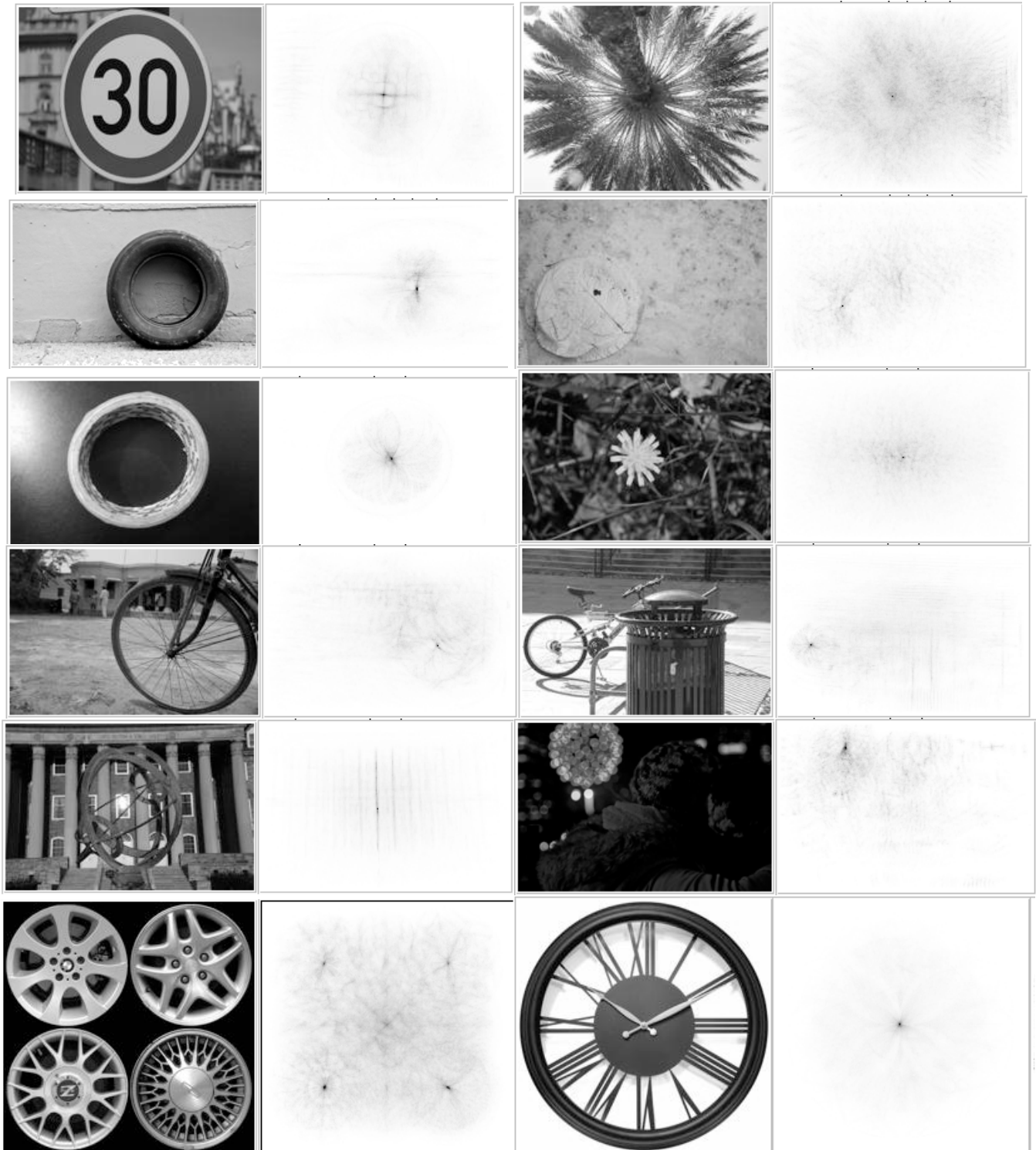
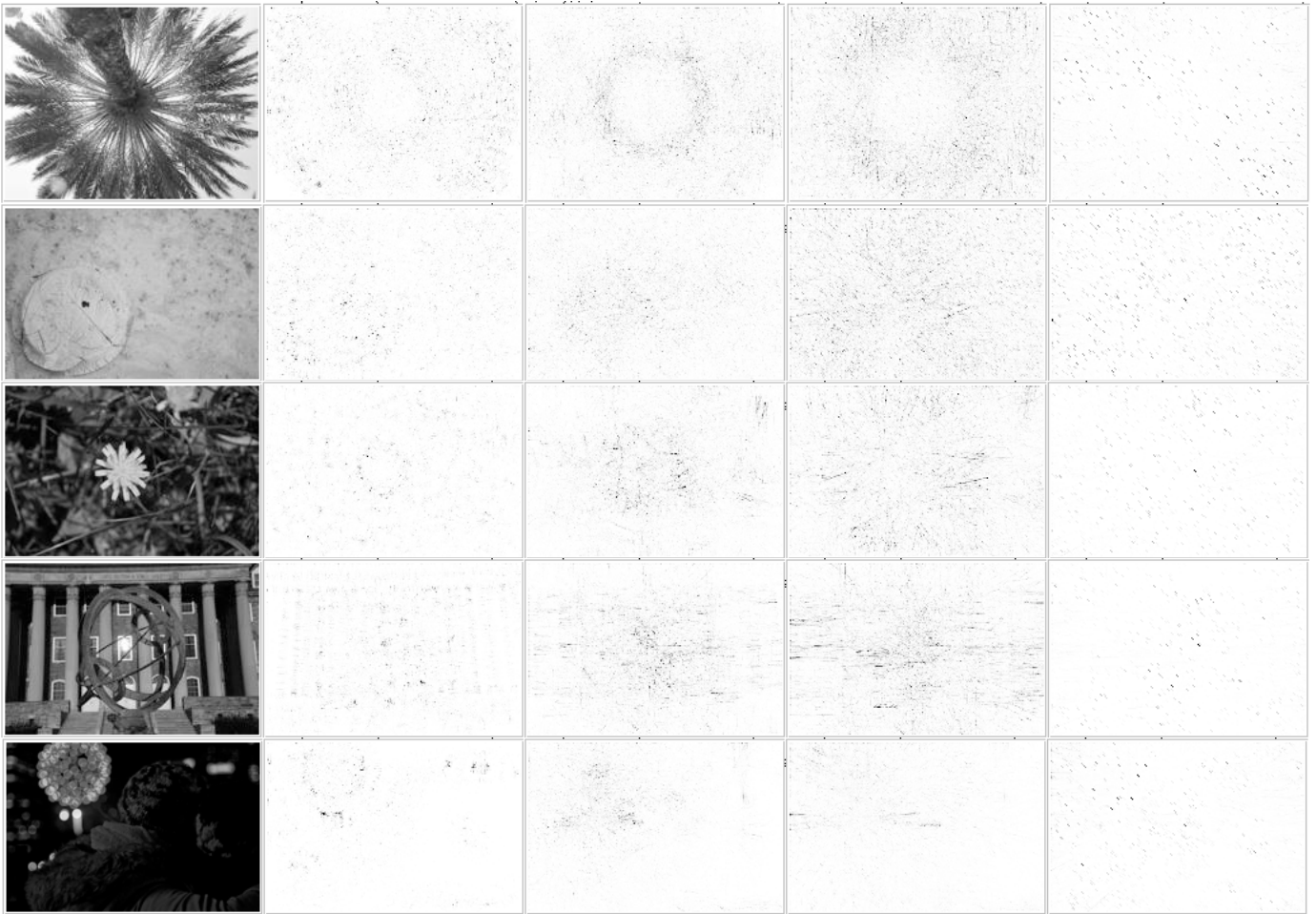
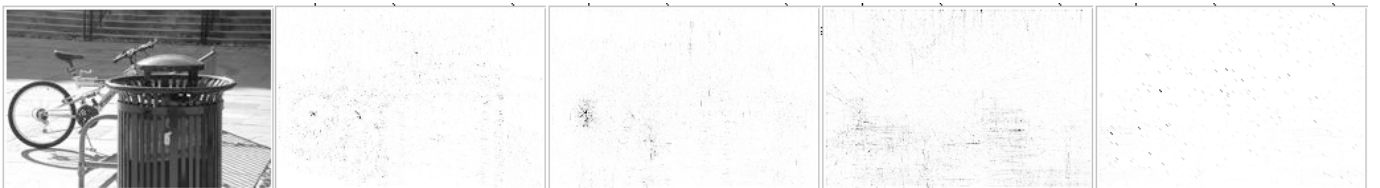


Figure 7. Original images and their corresponding IGHM symmetry maps





**Figure 8a.** Original images and the resulting fast radial symmetry transform maps of ranges 8,32,64,128



**Figure 8b.** Original images and the resulting fast radial symmetry transform maps of ranges 8,32,64,128

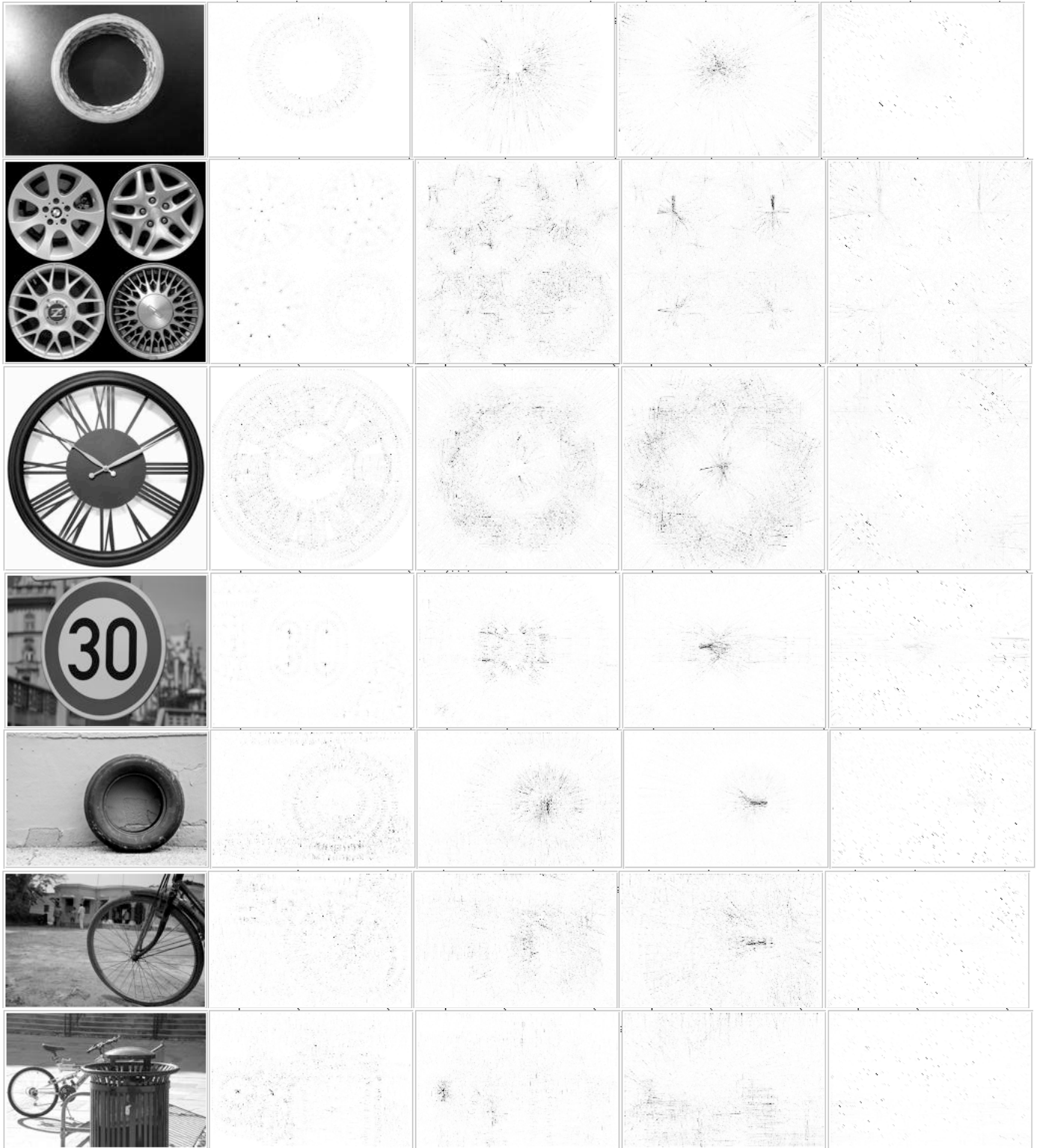


Figure 8c. Original images and the resulting fast radial symmetry transform maps of ranges 8,32,64,128

### 3.3 Processing Time

The time performance of bilateral symmetry detection using the IGHM method was evaluated on a MacBookPro with a 2.6GHz Intel i7 with 16 GB of memory. The running time (in milliseconds) of Reisfeld's general symmetry transform was measured for kernel sizes of 16 and 256 as well as the running time for the proposed method with corresponding maximum gradient influence distance constraints.

Table 2. Bilateral Symmetry Detection Time (ms).

Image	Reisfeld		IGHM	IGHM
	-16	-256		
Card, keys & hand	4,322	100,222	152	483
Lena 256 x 256	4,535	104,097	811	2,527
Elvis 256 x 256	4,716	105,931	898	1,750

These results were compared to those reported in the literature for two Hough based methods by Li et.al [2] and Yip [4]. The results reported by Li et.al. were obtained on a 2.2 GHz Pentium using Matlab and took from 10 to 20 seconds for a 300x300 image. Li et.al. further reported that a native C language implementation, using simple 64x64 pixel images and subsampling was able to run in under one second. R.KK Yip reported taking between 20-50 seconds results using a 500MHz Pentium 2 for simple 256x256 images of polygons, and about 50 minutes for more complex synthetic images. In comparison to the proposed IGHM based method processes simple 256x256 images in about 150ms. While it is half an order of magnitude faster when using small kernel sizes on complex images, the processing time is orders of magnitude less when considering image wide symmetry, as is predicted from the complexity analysis.

Like the proposed method, Patraucean's Hough voting based approach [3] is able to detect image wide mirror symmetries however no performance figures are reported other than that the validation time alone, for each symmetry candidate on an 800x600 image requires 2 seconds on an 2.53GHz, Intel i5 based computer. Since typically between five to ten candidates need to be validated, the total time for the validation alone amounts to 10 to 20 seconds. This however does not include the time required for the selection of the symmetry candidates themselves. In contrast, for similar sized images the proposed IGHM based method completes the task in about 10 seconds.

For radial symmetry the running time of the IGHM method was compared to Loy's state of the art, fast radial symmetry transform. For the experiment a gradient magnitude noise floor of 32 was used, which gave accurate results for all of the test images. To make the comparison fair a range setting of 96 was used with Loy's method as this gave the most accurate detection results. On average the running time in milliseconds of the IGHM

method, was equivalent to Loy's method as shown table 3. Loy's method was also run with a range setting of 16 for comparison.

Table 3. Radial Symmetry Detection Time (ms).

Image	Loy - 16	Loy - 96	IGHM
Card, keys & hand	77	412	219
Clockface 200x200	52	272	718
SpeedSign 200x150	42	201	278
TapeRoll 200x150	38	193	75
Turnbuckle 200x136	32	162	270
Lena 256x256	101	546	299

The running time of Loy's method is directly proportional to the detection range setting as can be noted from the results. Yet if the size of the symmetrical objects to be detected apriori is not known then both Loy's and Reisfeld's methods have to be run multiple times, hunting for the correct setting. This increases the running time by a multiplicative factor. This is not the case for the IGHM method that does not require a range setting. If however the size is known before hand then the distance of gradient influence in the IGHM method can be limited with the effect of reducing the running time of the IGHM method further. Another factor that impacts the running time of the IGHM method is the setting of the gradient magnitude noise floor. In the experiments reducing the noise floor to a magnitude of 16, on average doubles the running time, but does not significantly affect to the detection accuracy.

## 4 Conclusions

Detection of symmetry is a fundamental task in image processing but it has traditionally been computationally expensive. While efficient algorithms have been developed for calculating radial symmetry, algorithms for the detection of bilateral symmetry are still relatively slow. This paper has presented a fast and novel approach to finding both radial and bilateral symmetry based on inverted gradient hash maps. The experimental results demonstrate that not only is it significantly faster than other methods for detecting bilateral symmetry but it can successfully detect such symmetry on large scales. It is also as fast as the fastest methods for radial symmetry detection, but is significantly more successful at finding radial symmetries. Future work will consider methods for automatically reducing the range of influence of gradients to reduce processing both time and symmetry map noise.

## References

- [1] REISFELD, D., WOLFSON, H., YESHURUN, Y., "Context-free attentional operators: The generalized symmetry

- transform”, *International Journal of Computer Vision*, March 1995, **14**(2), pp 119-130
- [2] LI, W. H., ZHANG, A. M., KLEEMAN, L., “Fast Global Reflectional Symmetry Detection for Robotic Grasping and Visual Tracking” in *Proceedings of Australasian Conference on Robotics and Automation*, December 2005.
- [3] PATRAUCEAN, V., VON GIOI, R.G., OVSJANIKOV, M., “Detection of Mirror-Symmetric Image Patches,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* , 23-28 June 2013, pp.211-216
- [4] K.K YIP, R., “A Hough transform technique for the detection of reflectional symmetry and skew-symmetry” *Pattern Recognition Letters*, **21**(2), Feb. 2000, pp. 117-130,
- [5] LOY, G. and ZELINSKY, A. “A Fast Radial Symmetry Transform for Detecting Points of Interest”, in *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Aug. 2003, **25**(8), pp.959-973,
- [6] LOY, G. “Computer Vision to See People: a basis for enhanced human computer interaction” *PhD Thesis, Australian National University*, January 2003
- [7] ZOBEL, J., and MOFFAT, A., “Inverted files for text search engines,” *ACM Computing Surveys*, vol. 38, no. 2, p. 6, July 2006
- [8] GONZALEZ, R., “Fast Line and Circle Detection Using Inverted Gradient Hash Maps, in The 40th International conference on Acoustics, speech and signal processing, proceedings of. (ICASSP2014), IEEE, 19-24 April 2015 Brisbane, Australia, pp.1354-1358
- [9] GONZALEZ, R.C., WOODS, R.E. *Digital Image Processing*, (2nd ed.) Chapter 6.
- [10] CVPR 2013: Symmetry Detection from Real World Images competition dataset. Retrieved from <http://vision.cse.psu.edu/research/symComp13/index.shtml>