Fast Radial Symmetry for Detecting Points of Interest

Gareth Loy and Alexander Zelinsky, *Senior Member*, *IEEE*

Abstract—A new transform is presented that utilizes local radial symmetry to highlight points of interest within a scene. Its low-computational complexity and fast runtimes makes this method well-suited for real-time vision applications. The performance of the transform is demonstrated on a wide variety of images and compared with leading techniques from the literature. Both as a facial feature detector and as a generic region of interest detector the new transform is seen to offer equal or superior performance to contemporary techniques at a relatively low-computational cost. A real-time implementation of the transform is presented running at over 60 frames per second on a standard Pentium III PC.

Index Terms—Radial symmetry, points of interest, feature detection, face detection, real-time.

1 Introduction

As human beings, when we look at a scene we concentrate on certain points more than others. When looking at a person we tend to pay more attention to the face than the rest of the body and within the face we concentrate on the eyes and mouth more than the cheeks or forehead. People process visual information selectively, because some points contain more interesting information than others. In computer vision these are called *points of interest*.

Automatic detection of points of interest in images is an important topic in computer vision. Point of interest detectors can be used to selectively process images by concentrating effort at key locations in the image, they can identify salient features and compare the prominence of such features, and real-time interest detectors can provide attentional mechanisms for active vision systems [30].

In this paper, a novel point of interest operator is presented. It is a simple and fast gradient-based interest operator that detects points of high radial symmetry. The approach was inspired by the results of the generalized symmetry transform [24], [9], [25], although the final method bares more similarity to the work of Sela and Levine [28] and the circular Hough transform [12], [19]. The approach presented herein determines the contribution each pixel makes to the symmetry of pixels around it, rather than considering the contribution of a local neighborhood to a central pixel. Unlike previous techniques that have used this approach [12], [19], [28], it does not require the gradient to be quantized into angular bins, the contribution of *every* orientation is computed in a single pass over the image. The new method works well with a general fixed parameter set, however, it can also be tuned to exclusively detect particular kinds of features. Computationally, the algorithm is very efficient, being of order $O(KN)$ when considering local radial symmetry in $N \times N$ neighborhoods across an image of K pixels.

This paper presents a substantially revised and extended account of research previously reported in the author's conference publication [18]. Section 2 reviews the published work of other researchers in this area. Section 3 defines the new radial symmetry transform. Section 4 discusses the selection of parameters, and Section 5 describes how the transform can be adapted to different tasks. Section 7 shows the performance of the new transform on a variety of images and compares it to existing techniques, and Section 8 presents the conclusions.

2 Background

A number of context-free attentional operators have been proposed for automatically detecting points of interest in images. These operators have tended to use local radial symmetry as a measure of interest. This correlates well with psychophysical findings on fixation points of the human visual system. It has been observed that visual fixations tend to concentrate along lines of symmetry [17]. Sela and Levine [28] noted that the psychophysical findings of Kaufman and Richards [11] corroborated this, placing the mean eye fixation points at the intersection of lines of symmetry on a number of simple 2D geometric figures. It has also been observed that visual fixations are attracted to centers of mass of objects [27] and that these centers of mass are more readily determined for objects with multiple symmetry axes [23]. Stark and Pritevera [21], [22] compared the responses of a number of artificial region of interest detectors, including Reisfeld's *generalized symmetry transform* [24], with regions of interest detected by human subjects. By using several different algorithms in conjunction with a clustering procedure, they were able to predict the locations of human-detected regions of interest.

One of the best-known point of interest operators is Reisfeld's *generalized symmetry transform* [24]. This transform highlights regions of high contrast and local radial symmetry and has been applied to detecting facial features [24], [9], [25]. It involves analyzing the gradient in a neighborhood about each point. Within this neighborhood, the gradients at pairs of points symmetrically arranged about the central pixel are compared for evidence of radial symmetry, and a contribution to the symmetry measure of the central point is

- *The authors are with the Department of Systems Engineering, Australian National University, ACT 0200, Australia.*
E-mail: {gareth, alex}@syseng.anu.edu.au.

Manuscript received 14 Dec. 2001; revised 20 July 2002; accepted 21 Nov. 2002.
Recommended for acceptance by S. Sarkar.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 115569.

0162-8828/03/$17.00 © 2003 IEEE Published by the IEEE Computer Society

computed. The computational cost is high, being of order $O(KN^2)$, where K is the number of pixels in the image and N is the width of the neighborhood. While a real-time implementation has been attempted [31], it required a massive parallel computer architecture and was only able to achieve processing times of the order of seconds per frame.

Lin and Lin [16] present a symmetry measure specifically for identifying facial features in images. They propose a masking technique to evaluate radial symmetry based on gradient direction. Gradient directions are quantized into eight bins. The masks show which bin the local gradients should fall into for perfect radial symmetry about the center of the neighborhood (for either a dark or light blob). Fig. 1a(i) shows the 3×3 gradient orientation mask for detecting light blobs (gradient pointing from dark to light). Dual-masks are used to accommodate for pixels where the acceptably radially-symmetric gradient orientations span two orientation bins, Fig. 1a(ii) shows the dual mask set for a 5×5 neighborhood. The radial symmetry at each pixel is determined by examining the discrepancy between the gradient orientations in the local neighborhood and the orientation masks that represent perfect radial symmetry. The output of radially symmetric points from this comparison tends to be quite dense. In order to obtain points of radial symmetry useful for facial feature extraction, two additional inhibitory processes are required: An edge map is used to eliminate all interest points which do not occur on edges and regions of uniform gradient distribution are filtered out. The computational cost of the algorithm is stated as "$O(9K)$" for an image of K pixels. However, within the definition of the algorithm the size of the local neighborhood within which symmetry is determined is explicitly set to either 3×3 or 5×5. While the results for these values of $N = 3$ and $N = 5$ are good, it is not clear that this same level of performance will hold for larger neighborhoods. In any case, extending this algorithm to measure symmetry in an $N \times N$ local neighborhood results in a computational cost of order $O(KN^2)$.

Sun et al. [29] modify the symmetry transforms of Reisfeld et al. [24] and Lin and Lin [16] to obtain a symmetry measure which is combined with color information to detect faces in images. An orientation mask is used similar to Lin and Lin's [16], together with a distance-weighting operator similar to Reisfeld et al. [24], and the magnitude of the gradient is also taken into consideration. By using skin color to initially identify potential face regions, the scale of the symmetry operators can be chosen to suit the size of the skin region under consideration.

Sela and Levine [28] present an attention operator based on psychophysical experiments of human gaze fixation. Interest points are defined as the intersection of lines of symmetry within an image. These are detected using a symmetry measure which determines the loci of centers of cocircular edges[1] and requires the initial generation of an edge map. Edge orientations are quantized into a number of angular bins, and *inverted annular templates* are introduced to calculate the symmetry measure in a computationally efficient manner. Fig. 1b shows one such template placed over edge point $\mathbf{p}$. Note that the direction of the gradient $\mathbf{g}(\mathbf{p})$ lies within the angular range of the template, and r_{min} and r_{max} specify the radial range of the template. Separate templates are required for different circle radii and gradient orientations. Convolving one such template, of radius n and a particular angular range, with an image of edges, whose normals lie within with this same angular range, generates an image showing the centers of circles of radius n tangential to these edges. This is repeated for each angular bin and each radius to form images of circle center locations. Cocircular points are then determined by examining common center points for circles of the same radius. The calculation of the final interest measure combines these points with orientation information of the corresponding cocircular tangents. This method can also be readily applied to log-polar images. The technique was shown to run in real-time on a network of parallel processors. The computational cost is of order $O(KBN)$, where B is the number of angular bins used, B is typically at least eight.

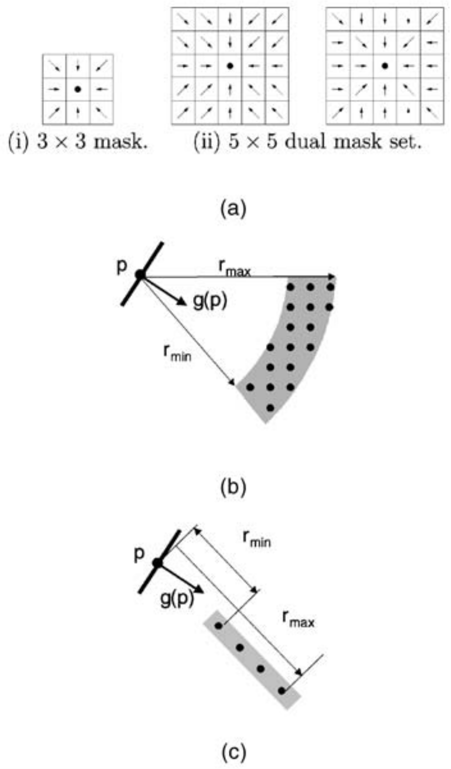

(i) 3×3 mask. (ii) 5×5 dual mask set.

Fig. 1. Techniques of other authors. (a) Gradient orientation masks used by Lin and Lin [16] for detecting light blobs. (b) Inverted annular template as used by Sela and Levine [28]. (c) The spoke filter template proposed by Minor and Sklansky [19].

The approach of Sela and Levine bares some similarity to the circular Hough transform which can also be used to find blobs in images. Duda and Hart [8] showed how the Hough transform could be adapted to detect circles with an appropriate choice of parameter space. They required a three-dimensional parameter space to represent the parameters $a, b,$ and c in the circle equation $(x - a)^2 + (y - b)^2 = c^2$. Kimme et al. [12] noted that on a circle boundary the edge orientation points toward or away from the center of the circle, and used this to refine Duda and Hart's technique and reduce the density of points mapped into the parameter space. Minor and Sklansky [19] further extended the use of edge orientation, introducing a *spoke filter* that plotted a line of points perpendicular to the edge direction (to the nearest 45 degrees) as shown in Fig. 1c. This allowed simultaneous detection of circles over a range of sizes (from r_{min} to r_{max} in Fig. 1c). An

[1] Two edges are said to be *cocircular* if there exists a circle to which both edges are tangent.

8-bit code is generated for each point in the image, one bit for each of the eight 45 degree wide orientation bins. Each bit indicates whether a spoke filter of the appropriate orientation has plotted a point in a 3×3 neighborhood about the point in question. Four discrete output levels are determined from the bit codes: all 8 bits positive, 7 bits positive, 6 adjacent bits positive, and all other cases. This technique was successfully used to detect blobs in infrared images. The computation required for an image of K pixels is of order $O(KBN)$, where B is the number of angular bins used ([19] used 8), and N is the number of radial bins.

Di Gesù and Valenti [6] present another method for measuring image symmetry called the *discrete symmetry transform*. This transform is based on the calculation of local axial moments, and has been applied to eye detection [6], processing astronomical images [7], and as an early vision process in a cooperative object recognition network [4]. The computational load of the transform is of the order $O(KBN)$, where K is the number of pixels in the image, N is the size of the local neighborhoods considered, and B is the number of directions in which the moments are calculated. This load can be reduced by using a fast recursive method for calculating the moments, giving a reduced computational order of either $O(KB)$ [5] or $O(KN)$ [20] depending on how the recursion is implemented. Both these recursive implementations of the discrete symmetry transform have been used to follow the eyes of a human face in video sequences where the scale of the eyes is known a priori. A drawback of this transform as a symmetry-based interest detector is its tendency to highlight lines and regions of high texture in addition to radially symmetric features.

Kovesi [13] presents a technique for determining local symmetry and asymmetry across an image from phase information. He notes that axes of symmetry occur at points where all frequency components are at either the maximum or minimum points in their cycles, and axes of asymmetry occur at the points where all the frequency components are at zero-crossings. Local frequency information is determined via convolution with quadrature log Gabor filters. These convolutions are performed for a full range of filter orientations and a number of scales, with each scale determining the response for a particular frequency bin. This technique is invariant to uniform changes in image intensity and as such is a truer measure of pure symmetry than other approaches which tend to measure a combination of symmetry and contrast. The computational cost of this method is high, and although the convolutions are efficiently performed in the frequency domain, the computation required to transform the image between spatial and frequency domains is costly. This method is not intended as a point of interest operator, however, the resulting continuous symmetry measures it produces strongly corroborate the theory that points of interest lie on lines of symmetry. For a detailed discussion on image phase and its application, see Kovesi [14].

Katahara and Aoki [10] use horizontal and vertical bilateral symmetry determined from the gradient to locate facial features. They require the face to be upright and directly facing the camera. While the simplicity of their approach is attractive, it is inadequate to deal with complex images. In addition to the orientation constraint on the face, this method is only suitable for processing images whose features are well discriminated from the rest of the image by the gradient operator, and it cannot tolerate complex backgrounds. It is therefore better suited to processing the output of other point of interest or feature detectors rather than dealing with raw images.

3 Definition of the Transform

The new transform is calculated at one or more radii $n \in N$, where N is the set of radii of the radially symmetric features to be detected. The value of the transform at radius n indicates the contribution to radial symmetry of the gradients a distance n away from each point. While the transform can be calculated for a continuous set of radii, this is generally unnecessary as a subset of radii is normally sufficient to yield a representative result.

An overview of the algorithm is shown in Fig. 2 along with the key signals (images) involved. These signals and their formation are described in the remainder of this section.

At each radius n, an *orientation projection image* O_n and a *magnitude projection image* M_n are formed. These images are generated by examining the gradient[2] $\mathbf{g}$ at each point $\mathbf{p}$ from which a corresponding *positively-affected pixel* $\mathbf{p}_{+ve}(\mathbf{p})$ and *negatively-affected pixel* $\mathbf{p}_{-ve}(\mathbf{p})$ are determined, as shown in Fig. 3. The *positively-affected pixel* is defined as the pixel that the gradient vector $\mathbf{g}(\mathbf{p})$ is pointing to, a distance n away from $\mathbf{p}$, and the *negatively-affected pixel* is the pixel a distance n away that the gradient is pointing directly away from.

The coordinates of the positively-affected pixel are given by

$$\mathbf{p}_{+ve}(\mathbf{p}) = \mathbf{p} + \text{round}\left(\frac{\mathbf{g}(\mathbf{p})}{\|\mathbf{g}(\mathbf{p})\|}n\right),$$

while those of the negatively-affected pixel are

$$\mathbf{p}_{-ve}(\mathbf{p}) = \mathbf{p} - \text{round}\left(\frac{\mathbf{g}(\mathbf{p})}{\|\mathbf{g}(\mathbf{p})\|}n\right),$$

where "round" rounds each vector element to the nearest integer.

The orientation and magnitude projection images are initially zero. For each pair of affected pixels, the corresponding point $\mathbf{p}_{+ve}$ in the orientation projection image O_n and magnitude projection image M_n is incremented by 1 and $\|\mathbf{g}(\mathbf{p})\|$, respectively, while the point corresponding to $\mathbf{p}_{-ve}$ is decremented by these same quantities in each image. That is,

$$O_n(\mathbf{p}_{+ve}(\mathbf{p})) = O_n(\mathbf{p}_{+ve}(\mathbf{p})) + 1,$$

$$O_n(\mathbf{p}_{-ve}(\mathbf{p})) = O_n(\mathbf{p}_{-ve}(\mathbf{p})) - 1,$$

$$M_n(\mathbf{p}_{+ve}(\mathbf{p})) = M_n(\mathbf{p}_{+ve}(\mathbf{p})) + ||\mathbf{g}(\mathbf{p})||,$$

$$M_n(\mathbf{p}_{-ve}(\mathbf{p})) = M_n(\mathbf{p}_{-ve}(\mathbf{p})) - ||\mathbf{g}(\mathbf{p})||.$$

The radial symmetry contribution at radius n is defined as the convolution

$$S_n = F_n * A_n, \tag{1}$$

[2] All gradients in this paper are determined using the 3×3 Sobel operator.

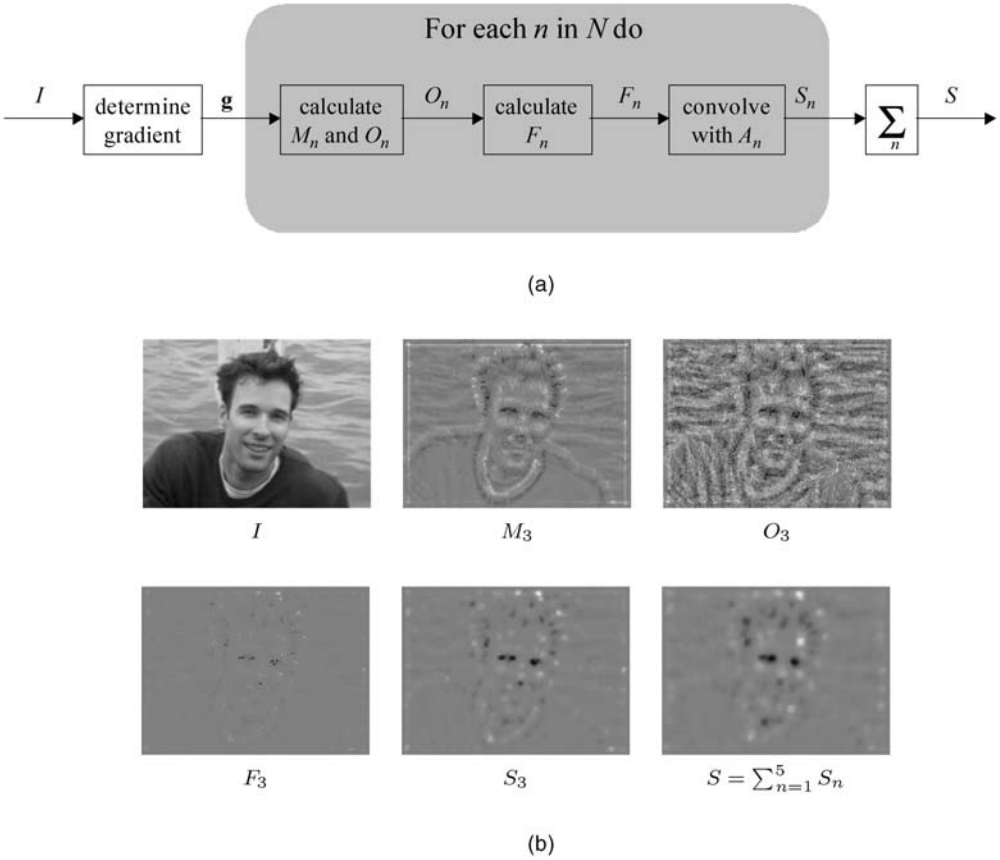


Fig. 2. (a) Block diagram showing the steps involved in computing the transform. (b) Example signals for radius $n = 3$ and with $\alpha = 2$, positive values are shown as light pixels, negatives, as dark and zero as midgray; gradient is assumed to point from dark to light.

where

$$F_n(\mathbf{p}) = \frac{M_n(\mathbf{p})}{k_n}\left(\frac{|\tilde{O}_n(\mathbf{p})|}{k_n}\right)^{\alpha}, \tag{2}$$

and

$$\tilde{O}_n(\mathbf{p}) = \begin{cases} O_n(\mathbf{p}) & \text{if } O_n(\mathbf{p}) < k_n \\ k_n & \text{otherwise.} \end{cases} \tag{3}$$

A_n is a two-dimensional Gaussian, α is the radial strictness parameter, and k_n is a scaling factor that normalizes M_n and O_n across different radii. These parameters are discussed in more detail in Section 4.

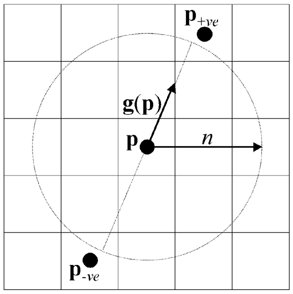


Fig. 3. The locations of pixels $\mathbf{p}_{+ve}(\mathbf{p})$ and $\mathbf{p}_{-ve}(\mathbf{p})$ affected by the gradient element $\mathbf{g}(\mathbf{p})$ for a range of $n = 2$. The dotted circle shows all the pixels that can be affected by the gradient at $\mathbf{p}$ for a radius n.

The full transform is defined as the average of the symmetry contributions over all the radii considered,

$$S = \frac{1}{|N|}\sum_{n \in N} S_n. \tag{4}$$

If the gradient is calculated so it points from dark to light, then the output image S will have positive values corresponding to bright radially symmetric regions and negative values indicating dark symmetric regions as in Fig. 2b.

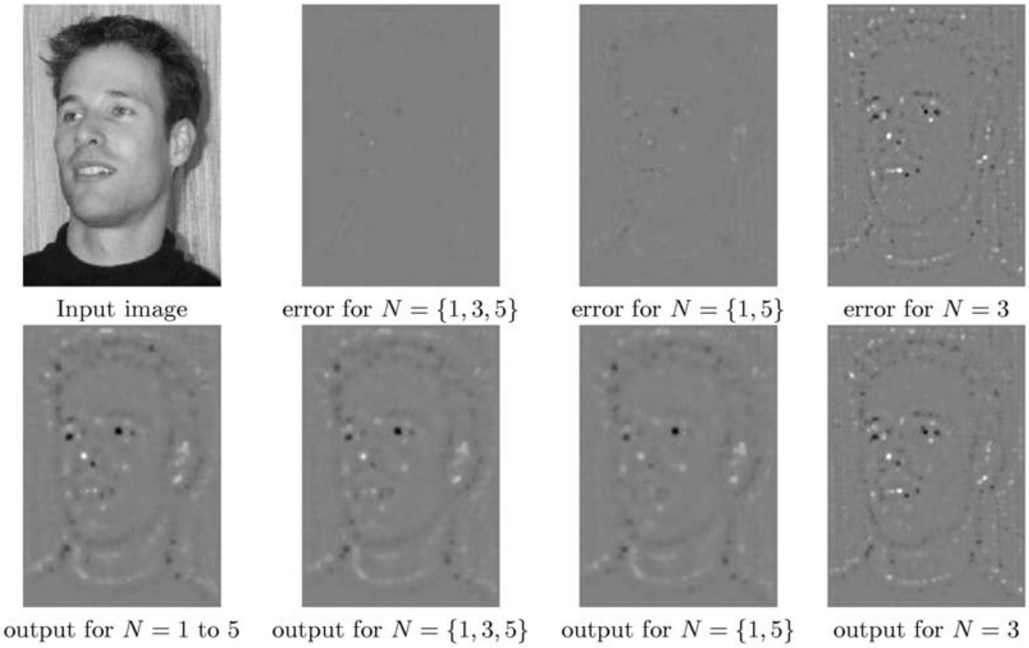
Sometimes it is more useful to consider the gradient orientation exclusively, removing the effect of contrast on the level of interest attributed to points in the image. This leads to an alternate *orientation-based* radial symmetry that is defined by replacing F_n in (1) by

$$\hat{F}_n(\mathbf{p}) = \operatorname{sgn}(\tilde{O}_n(\mathbf{p}))\left(\frac{|\tilde{O}_n(\mathbf{p})|}{k_n}\right)^{\alpha}.$$

This provides a result that is more robust to lighting changes. However, when applying this orientation-based formulation it is generally necessary to ignore very small gradients that tend to add noise to the result; this is discussed in detail in Section 5.1.

4 Choosing the Parameters

The definition of the transform contains a number of parameters that need to be appropriately defined; these are:


Input image error for $N = \{1, 3, 5\}$ error for $N = \{1, 5\}$ error for $N = 3$

output for $N = 1$ to 5 output for $N = \{1, 3, 5\}$ output for $N = \{1, 5\}$ output for $N = 3$

Fig. 4. Effect of varying the set of radii N at which the transform is computed.

- a set of radii $N = \{n_1, n_2, \ldots\}$ at which to calculate S_n,
- the Gaussian kernels A_n,
- the radial strictness parameter α, and
- the normalizing factor k_n.

This section discusses each of these in turn and describes their effect on the output of the transform. A general set of parameters is presented in Section 6.

4.1 Set of Radii N

The traditional approach to local symmetry detection [6], [24], [28] is to calculate the symmetry apparent in a local neighborhood about each point. This can be achieved by calculating S_n for a continuous set of radii $N = \{1, 2, \ldots, n_{max}\}$ and combining using (4). However, since the symmetry contribution is calculated independently for each radius n, it is simple to determine the effects at a single radius or an arbitrary selection of radii that need not be continuous. Furthermore, the results obtained by only examining alternate radii give a good approximation to the output obtained by examining all the radii, while saving on computation.

The effect of choosing sparse sets of radii was quantified experimentally by comparing the output of the transform calculated across all radii from 1 to 5 to that calculated across several sparse sets of radii. An example of this experiment is shown in Fig. 4. The experiment was run over a database of 295 diverse face images and the average power of the error between the sparse and continuous outputs were determined for each set of radii. The results are shown in Table 1 with the power of the error expressed as a percentage of the power of the transform calculated across all five radii.

TABLE 1
Parameter Settings Used for Experimentation

Radii	1,2,3,4,5	1,2,3	1,5	3
Error Power	0%	7.9%	37%	780%

Table 1 shows that taking alternative radii $(1, 3, 5)$ gives a very close approximation to using all the radii with an error of only 7.9 percent between the two outputs. Unsurprisingly, as less radii are included the error increases quite rapidly.

If the scale of a radially symmetric feature is know a priori, then the feature can be efficiently detected by only determining the transform at the appropriate radius(es). For example, the irises in the eyes in the input image in Fig. 4 have a radius of approximately five pixels, so will be well detected using a radius of 5, or radii 1 and 5 as can be seen in Fig. 4.

4.2 Gaussian Kernels A_n

The purpose of the Gaussian kernel A_n is to spread the influence of the positively- and negatively-affected pixels as a function of the radius n. A rotation invariant two-dimensional Gaussian is chosen since it has a consistent effect over all gradient orientations and it is separable so its convolution can be efficiently determined. Fig. 5 shows the contribution for a single gradient element $\mathbf{g}(\mathbf{p})$. By scaling the standard deviation linearly with the radius n, an arc of influence is defined that applies to all affected pixels. The

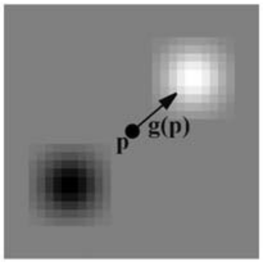
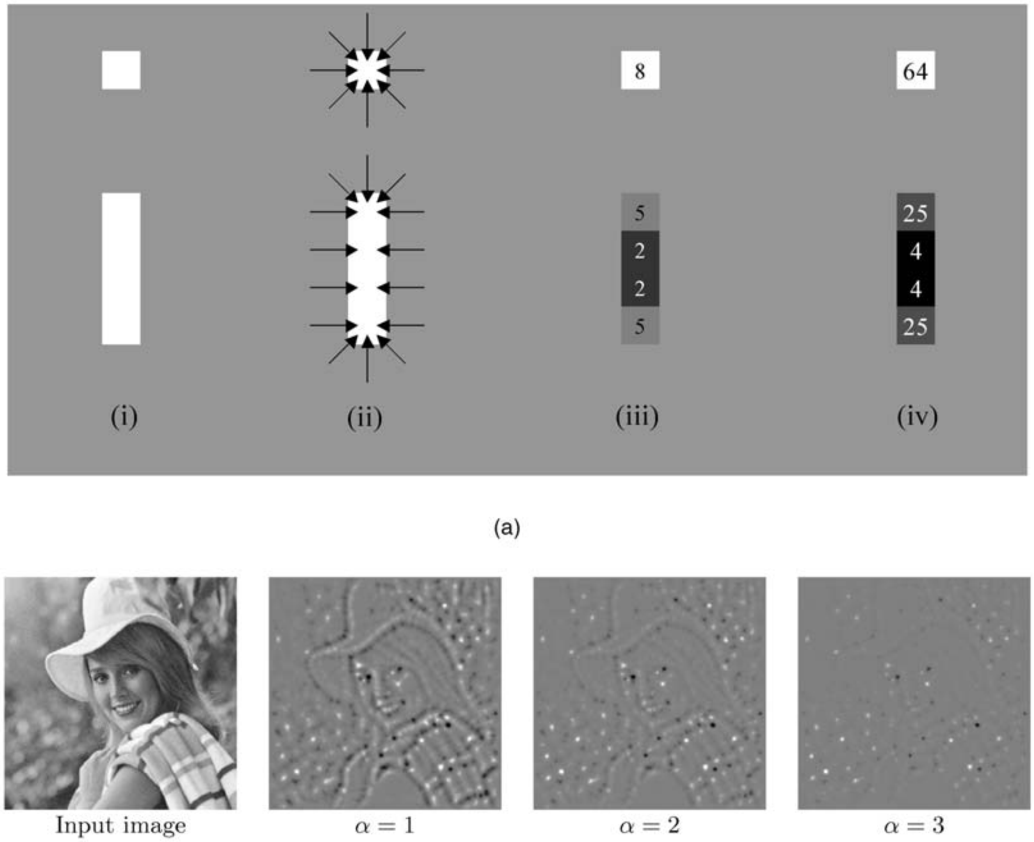


Fig. 5. The contribution of a single gradient element, with A_n chosen to be a 2D Gaussian of size $n \times n$ and standard deviation $\sigma = 0.25n$ and $n = 10$.



(a)

(b)

Fig. 6. Effect of varying α. (a) At the pixel level: 1) Sample arrangement of light pixels on a dark background, 2) gradient from adjacent pixels, 3) number of gradient elements pointing at each pixel O_n, 4) square of the number of gradient elements pointing at each pixel O_n^2. (b) Effect of varying α at the image level. Original image from USC-SIPI Image Database [1].

width of the arc is defined by scaling the standard deviation of A_n with respect to n.

All A_n are defined as two-dimensional Gaussians whose elements sum to n. Convolving with A_n has the result of spreading the effect of each gradient element by an amount proportionate to the standard deviation of the Gaussian and amplifying its magnitude by n. Amplifying the magnitude is necessary to prevent the effect of gradient elements becoming negligible at large radii as a result of being spread too thinly across the image.

Even though the convolution with the Gaussian kernel is separable, depending on the size of the kernel used, this is often still the most time consuming part of the algorithm. This step can be sped up by replacing the Gaussian kernels with uniformly flat kernels whose convolution can be calculated recursively. Surprisingly, this still yields reasonable results in most circumstances, however, uniform (square) kernels are not invariant to rotation so Gaussian kernels are preferred. All results, with the exception of the real-time results in Fig. 10, in this paper are obtained using Gaussian kernels.

4.3 Radial-Strictness Parameter α

The parameter α determines how strictly radial the radial symmetry must be for the transform to return a high interest value. Fig. 6a illustrates the effect of α on O_n at the pixel level. Note how a higher α strongly attenuates the line relative to the dot.

Fig. 6b shows the effect of choosing α to be 1, 2, and 3 on S_1 for an image exhibiting strong radial values around the eyes. Once again, a higher α eliminates nonradially symmetric features such as lines.

A choice of $\alpha = 2$ is suitable for most applications. Choosing a higher α starts attenuating points of interest, while a lower α gives too much emphasis to nonradially symmetric features, however, choosing $\alpha = 1$ minimizes the computation when determining F_n in (2).

4.4 Normalizing Factor k_n

In order to compare or combine the symmetry images calculated for different radii, they must be represented on a similar scale. As the radius increases, so does the number of gradient elements that could potentially affect each pixel, that is, the number of pixels on the perimeter of the circle in Fig. 3.

One way of normalizing across scales is to divide O_n and M_n through by their maximum values as was done in the author's earlier publication [18]. However, this scales the result at each radius relative to itself, and does not provide an absolute measure that can be used to compare between different radii or different images.

It is preferable to scale O_n and M_n by the *expected* maximum value of O_n, and saturate O_n at this value (see (3)). M_n cannot be saturated in the same way, (although it could be averaged using division by O_n) however, large values of M_n do not cause problems, since O_n is raised to an exponential power and, so, is much more significant than M_n at locations where O_n saturates.

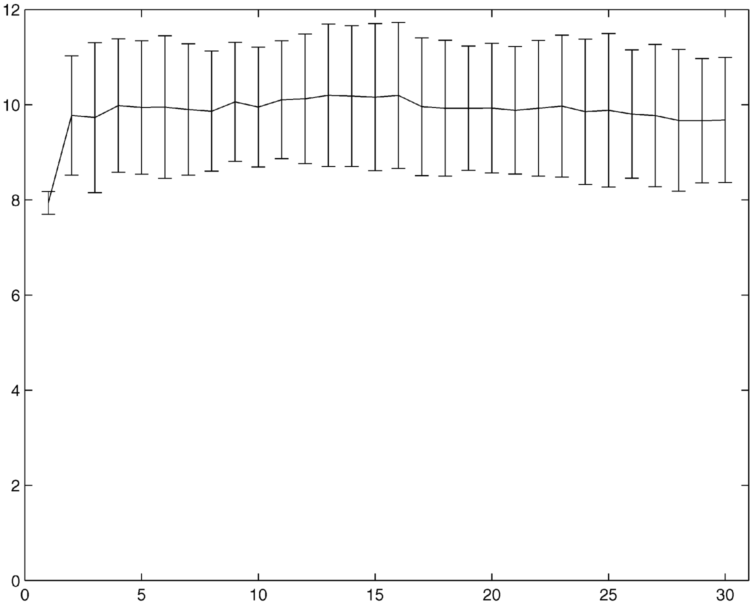


Fig. 7. The mean and standard deviation of the maximum value of the orientation projection images O_n for $n = 1$ to 30, calculated over 295 images.

Determining the expected maximum value of O_n is best done experimentally, since it depends on gradient directions of neighboring pixels and these gradient elements are not probabilistically independent. An experiment was conducted to determine the mean maximum value of O_n for a set of 295 real images for $n = 1$ to 30. The set of test images comprised of photographs of people at a range of scales, and with widely varying backgrounds and lighting conditions. All images were in jpeg format and were obtained off the Internet, image size varied from 108×130 to 405×244 pixels.

The result is shown in Fig. 7. Apart from the value of 8 for $n = 1$ (there are only eight pixels a distance 1 away from any pixel), the expected values for all radii $n \in [2, 30]$ lay within $9.9 \pm 3\%$.

So, a choice of

$$k_n = \begin{cases} 8 & \text{if } n = 1 \\ 9.9 & \text{otherwise} \end{cases}$$

suitably normalizes M_n and O_n in (2).

5 Refining the Transform

The transform can be refined to further increase computational speed and detect particular kinds of features. Refinements include:

- ignoring small gradients when calculating O_n and M_n,
- calculating dark (bright) symmetry by ignoring negatively- (positively-) effected pixels when determining O_n and M_n, and
- choosing a constant A_n.

5.1 Ignoring Small Gradients

Gradient elements with small magnitudes have less reliable orientations, are more easily corrupted by noise, and tend to correspond to features that are not immediately apparent to the human eye. Since the purpose of the transform is to pick out points of interest in the image, it is logical to ignore such elements in our calculation. Reisfeld implemented the generalized symmetry transform to ignored small gradients in his original work [26] and Sela and Levine also ignore small gradient elements; however, they go one step further and binarize the gradient image into an edge map.

We ignore small gradients by introducing a gradient threshold parameter β. When calculating images O_n and M_n, all gradient elements whose magnitudes are below β are ignored. The effect of a small β on M_n is negligible, however, even small values of β start to attenuate O_n in regions of low contrast. This results in an emphasis on interest points with high contrast.

A small value of β that eliminates the lowest 1-2 percent of the gradient will remove the small noisy gradients mentioned

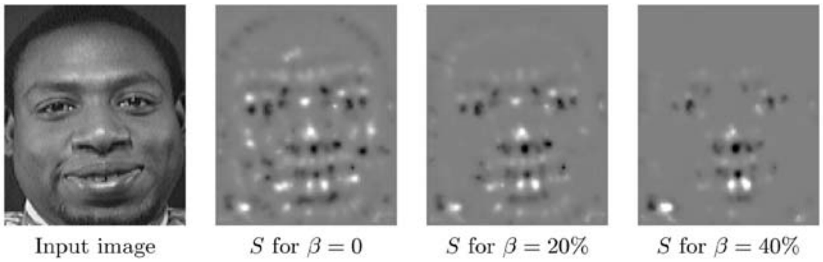


Fig. 8. The effect of different values of β on S. Here, β is measured as a percentage of the maximum possible gradient magnitude and $n = 1$. Original image from Database of Faces, AT&T Laboratories Cambridge [2].

TABLE 2
Parameter Settings Used for Experimentation

Parameter	Setting		
	Full	Fast	Fast Dark
Set of radii N	$\{n : n = 1, 2, ..., 6\}$	$\{n : n = 1, 3, 5\}$	$\{n : n = 1, 3, 5\}$
Gaussian kernel			
Size	n	n	n
Standard deviation	$0.5n$	$0.5n$	$0.5n$
Radial strictness α	2	2	2
Small gradients ignored	0	2% ignored	2% ignored
Dark symmetry	Yes	Yes	Yes
Bright symmetry	Yes	Yes	No

above. However, so long as low contrast features are not important, larger values of β can be chosen to increase the speed of the algorithm by considering fewer gradient elements.

The effect of large values of β is shown in Fig. 8, where β is measured as a percentage of the maximum possible gradient magnitude. In this example, this is beneficial for the detection of eyes and mouth with $\beta = 20\%$, however, too high a value, such as $\beta = 40\%$ starts to attenuate features of interest such as the corners of the mouth.

5.2 Dark and Bright Symmetry

The transform can be tuned to look only for dark or bright regions of symmetry. To look exclusively for dark regions, only the negatively-affected pixels need be considered when determining M_n and O_n. Likewise, to detect bright symmetry only positive effected pixels need be considered. Examples of dark symmetry are shown in Section 7.

Alternatively, dark and bright symmetries can be obtained by thresholding the output image S to eliminate all positive or negative values. This second approach has the advantage that inconsistent dark and light values will cancel each other out as they do when calculating both dark and light symmetries together. However, experimentation has shown that although this alternative approach gives slightly different dark/bright symmetry outputs, the result is no better than simply counting only dark/bright affected pixels. Therefore, since it is not necessary to cancel out inconsistent dark/light values, the first method is preferred as it offers a reduction in computation.

5.3 Choosing a Constant A_n

A faster implementation of the transform can be achieved by choosing the Gaussian kernel to be constant over all radii. The saving in computation comes by avoiding performing the convolution with A_n for each radius. Choosing A_n to be a fixed Gaussian still disperses the influence of the effected pixels and can produce reasonable results. In this case, only one convolution need be performed and (4) reduces to

$$S' = G_\sigma * \sum_n F_n, \quad (5)$$

where G_σ is a 2D Gaussian with standard deviation σ.

6 A General Set of Parameters

As discussed in Sections 4 and 5, there are a number of different parameters and refinements to the basic transform. In Table 2, three general parameter sets suitable for different applications of the transform are presented. The *Full* setting is the best choice when the transform is to be applied in an unsupervised manner; it provides more detail at the expense of requiring more computation than the alternative settings. The *Fast* setting detects both bright and dark symmetry quickly and the *Fast Dark* setting finds only regions of dark symmetry. The performance of each of these settings is presented in Section 7.

7 Performance Evaluation

The performance of the new transform was demonstrated on a range of images and compared with several prominent transforms from the literature.

7.1 Performance of the New Transform

Fig. 9 demonstrates the performance of the transform on faces and other images. These figures were generated using the parameter settings presented in Table 2, and show how the transform can provide a useful cue for the location of facial features—especially eyes—in face images, as well as highlighting generic points of interest that are characterized by high contrast and radial symmetry. Note that the orientation-based symmetry is more sensitive to low-contrast features and texture. This sensitivity can be reduced by using a higher gradient threshold; however, such sensitivity is desirable when considering low-contrast features such as the shadowed side of the face in Fig. 13.

The intuitive notion that facial features are generic points of interest provides a useful benchmark for evaluating point of interest operators. While the application of these operators is by no means limited to facial feature detection [4], [6], [19], this is certainly the most common application area [6], [9], [16], [24], [25], [28], [29]. Facial images provide a useful case study, offering images of widely varying appearances with well-defined sets of interest points, as well as directly addressing the primary application area of point of interest detectors.

The new transform has been implemented in a real-time vision system. The real-time code was written in C++ and made use of the Intel Image Processing Primitives (version 2.05) to achieve a mean processing time of 13.2 ms (standard deviation of 0.08 ms) per 240×320 image frame,

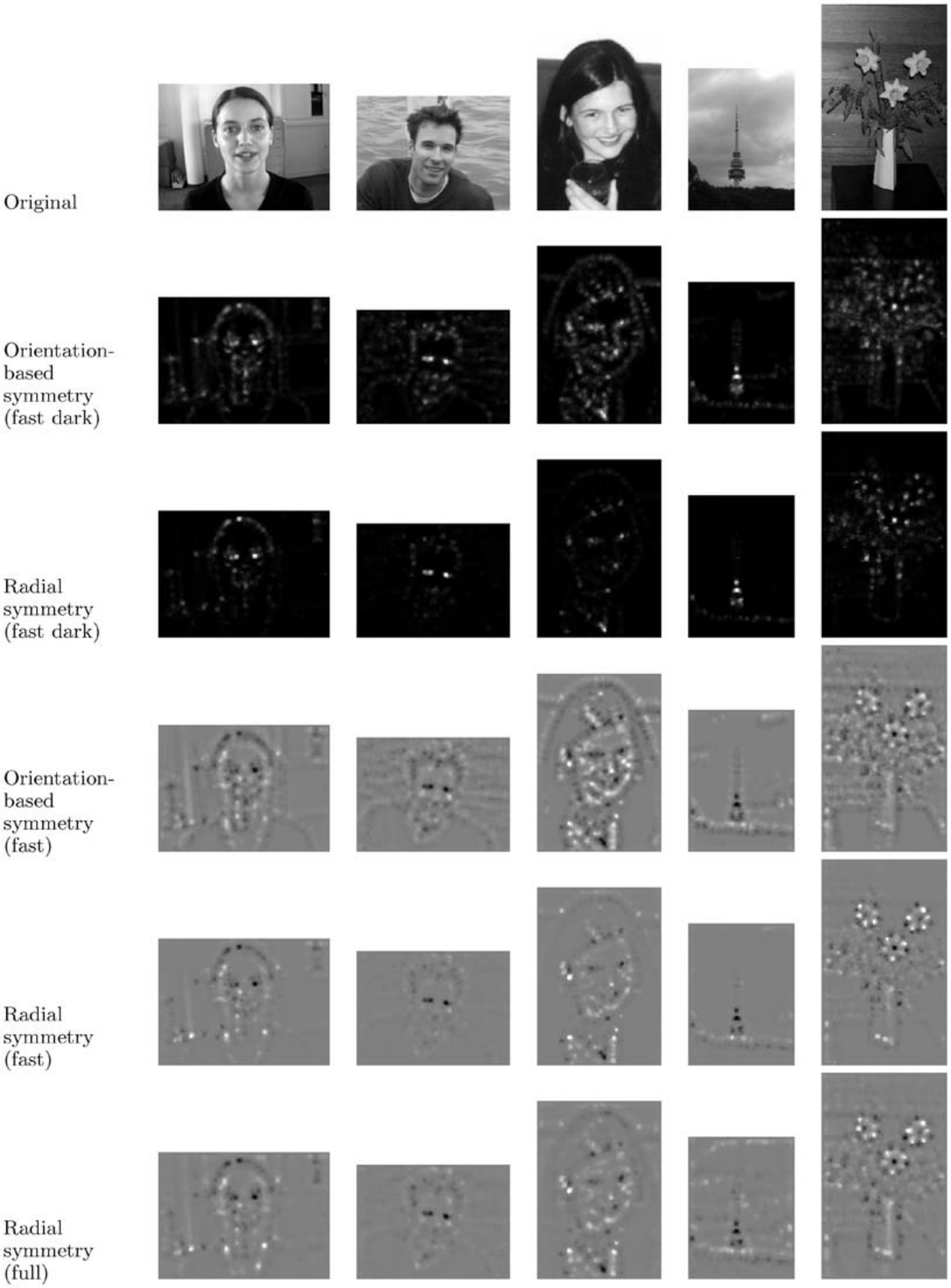

Original
Orientation-based symmetry (fast dark)
Radial symmetry (fast dark)
Orientation-based symmetry (fast)
Radial symmetry (fast)
Radial symmetry (full)

Fig. 9. The new transform applied to face and other images. The form of the transform and the parameter settings used for each row are indicated on the left. The left most image is from the BioID Face Database [3] and has been subsampled to half its original size.

on a 1.4 MHz Pentium III running under Lynux. The real-time system detects orientation-based symmetry online using the *fast dark* settings detailed in Table 2, however, to increase efficiency, uniform square kernels were used rather than Gaussians to blur the response at each radius (1). Fig. 10 shows some snap shots of the system in action. The results highlight the eyes and mouth of the subjects well and there are virtually no noticeable artifacts caused by using uniform square kernels rather than Gaussians.

The real-time code was also tested offline on the 256×256 image in Fig. 12 and timed over 10,000 iterations to determine a mean processing time of 10.8 ms for the calculation of the *fast dark* orientation-based symmetry on this image.

7.2 Comparison with Existing Transforms

The performance of the new fast radial symmetry transform was compared with other prominent transforms from the literature.

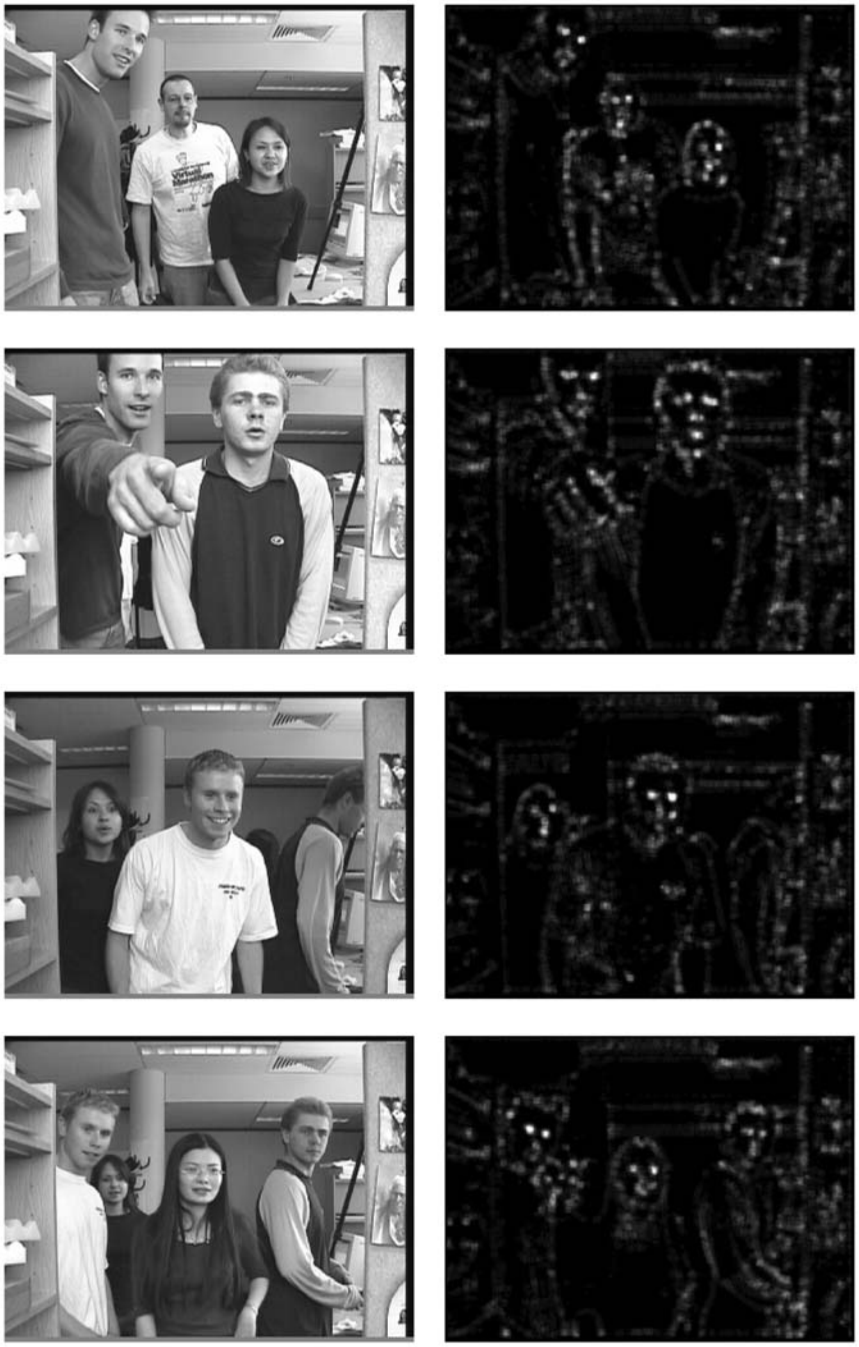


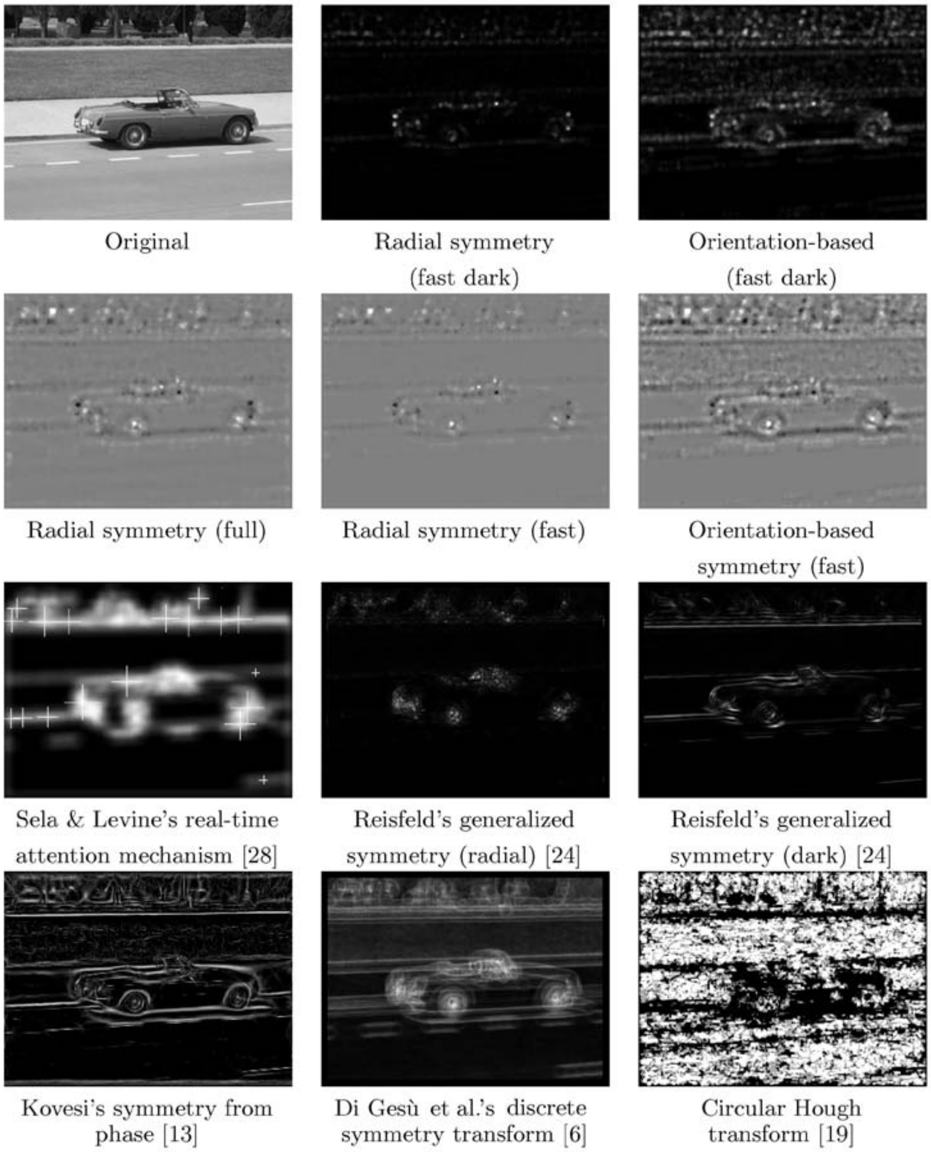
Fig. 10. The orientation-based *fast dark* implementation of the transform being calculated online in real-time. The left column shows sample input images and the right column shows the output, all images are 240×320 pixels.

Other transforms considered were:

- Sela and Levine's real-time attention mechanism [28],
- Reisfeld's generalized symmetry transform for both dark and radial generalized symmetry [24],
- Kovesi's symmetry from phase [13],
- Di Gesù et al.'s discrete symmetry transform [6], and
- Minor and Skalansky's implementation of the Circular Hough transform [19],

Kovesi's symmetry from phase was calculated for six filter orientations and four scales ranging from 2 to 2^4 pixels in diameter. All other methods were implemented with a local neighborhood radius of six pixels, allowing local symmetry to be detected in a neighborhood up to 13×13 pixels about each point. Where the necessary gradient orientation was quantized into eight bins.

Each of the transforms was implemented in Matlab 5.3 (Kovesi's symmetry from phase was implemented using Kovesi's own Matlab code [15]) and the output computed. For the majority of the transforms, an estimate of the approximate number of floating point operations involved was obtained from Matlab, however, for Di Gesù et al.'s discrete symmetry transform and Sela and Levine's real-time attention mechanism this was not the case. These transforms involve optimized low-level processes that were not practical to emulate in Matlab, so the number of operations required is not reported


Original

Radial symmetry (fast dark)

Orientation-based (fast dark)

Radial symmetry (full)

Radial symmetry (fast)

Orientation-based symmetry (fast)

Sela & Levine's real-time attention mechanism [28]

Reisfeld's generalized symmetry (radial) [24]

Reisfeld's generalized symmetry (dark) [24]

Kovesi's symmetry from phase [13]

Di Gesù et al.'s discrete symmetry transform [6]

Circular Hough transform [19]

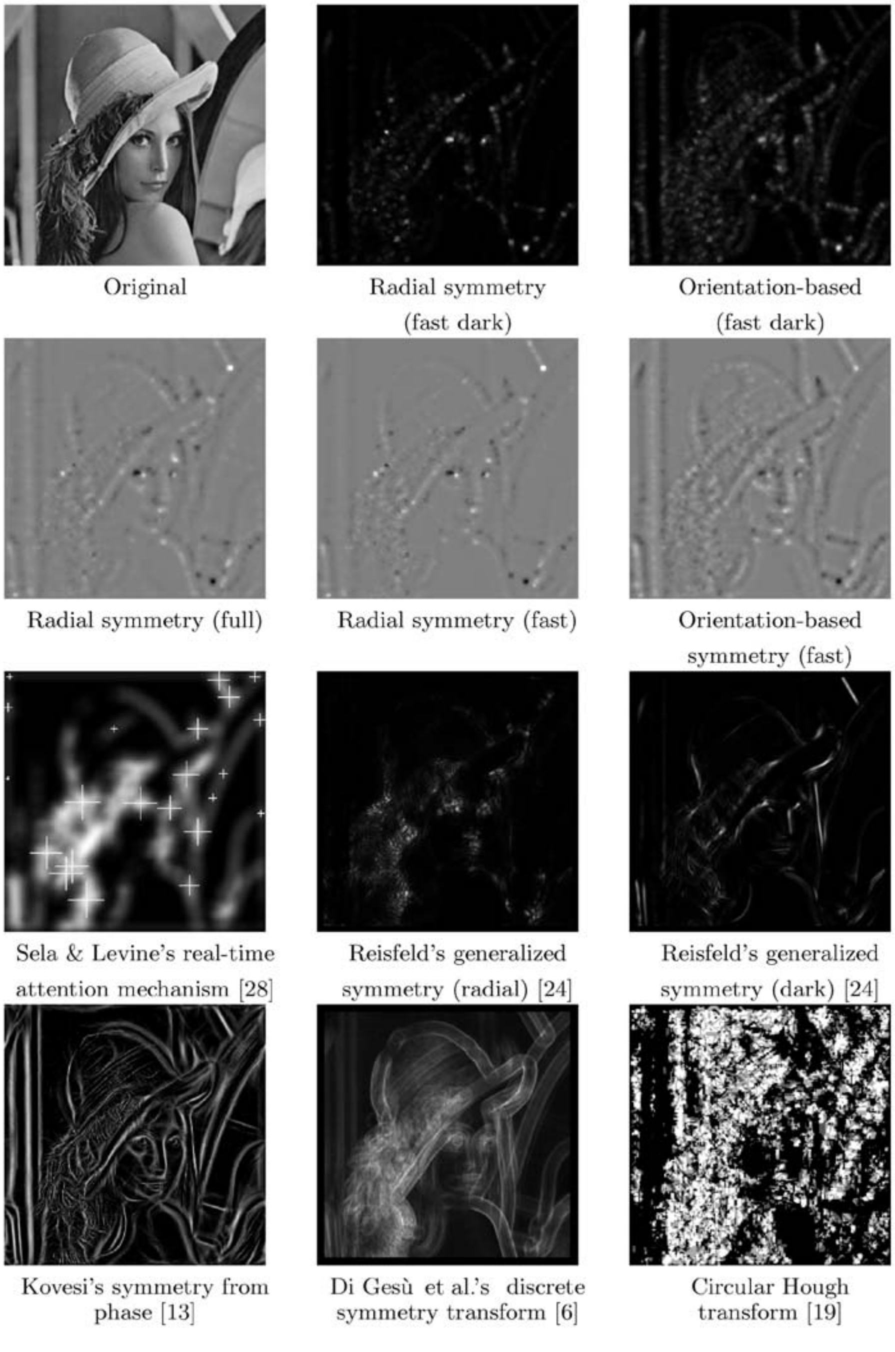
Fig. 11. Comparison of performance on a 320×240 outdoor image. The top two rows show the performance of the new transform, the bottom two rows show the output from other available transforms.

here. (Unsurprisingly, the *nonoptimized* implementations used to generate the visual results shown required computation well in excess of the other methods.)

The results are shown in Figs. 11, 12, and 13 and the computations required are presented in Table 3. These results demonstrate that the new transform can provide comparable or superior results to existing techniques while requiring a relatively low level of computation. As noted in the footnote to Table 3, the *Fast* and *Fast Dark* parameter settings ignore small gradients and are not calculated across all radiuses, however, the transform is still able to provide useful results and the computational efficiency is increased. Other transforms may also benefit from these techniques, indeed Reisfeld initially considered ignoring small gradients [26], however, the effect of these variables on other transforms has not been explored in this paper.

The real-time attention mechanism of Sela and Levine provides cloud-like approximations of interest points. The final step of this transform involves identifying local maximums in this output as points of interest. These have been marked with crosses in Figs. 11, 12, and 13 with the size of the cross corresponding to the value of the transform at the local maximum. The transform detects both eyes in the face in Fig. 12 and the high contrast eye in Fig. 13, but it also awards high interest to nonradially symmetric edges and areas of texture and fails to detect the circular wheels of the car in Fig. 11.

The results from the generalized symmetry transform show good detection of regions of interest by generalized radial symmetry, however, the generalized dark symmetry tends to highlight edges in addition to points of interest. The high-computational load of the generalized symmetry transform (and other methods that consider symmetry in a local neighborhood about each pixel [16]), comes from the computational load scaling with the square of the radius of the neighborhood. The larger the neighborhood, the more pixels that must be considered when calculating the transform at each point in the image. Even for modest sized



Original

Radial symmetry (fast dark)

Orientation-based (fast dark)

Radial symmetry (full)

Radial symmetry (fast)

Orientation-based symmetry (fast)

Sela & Levine's real-time attention mechanism [28]

Reisfeld's generalized symmetry (radial) [24]

Reisfeld's generalized symmetry (dark) [24]

Kovesi's symmetry from phase [13]

Di Gesù et al.'s discrete symmetry transform [6]

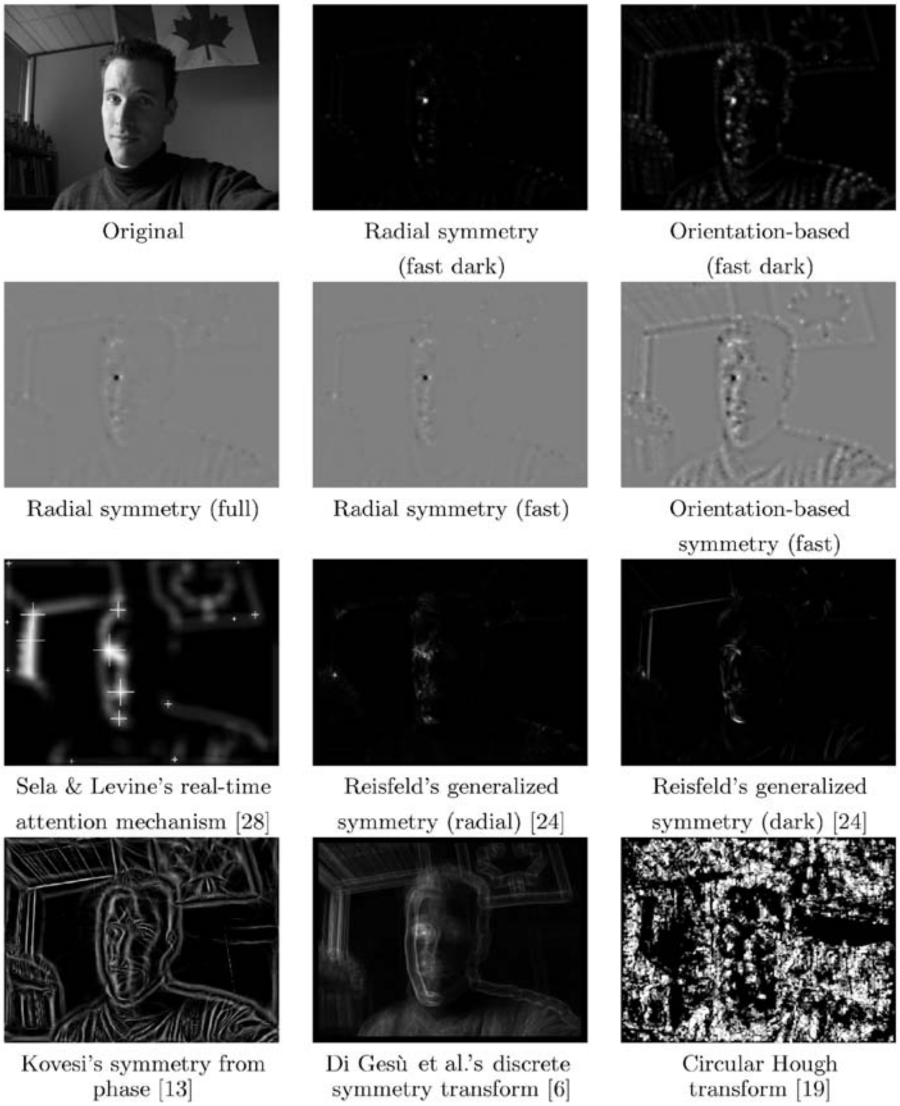
Circular Hough transform [19]

Fig. 12. Comparison of performance on the standard 256×256 Lena image. The top two rows show the performance of the new transform, the bottom two rows show the output from other available transforms.

neighborhoods, such as the 13×13 pixel neighborhood used for the experimentation on the 320×240, 256×256, and 256×341 images in Figs. 11, 12, and 13 the computation is considerable.

Calculating the symmetry from phase detects areas of high bilateral or radial symmetry independently of contrast. This method is not designed to detect points of interest in scenes; however, it provides a detailed map of the underlying symmetries present across the image that is instructive to consider in relation to other "symmetry operators." Comparing the results from this transform with those of Reisfeld's generalized *dark* symmetry, we see that (as noted by Kovesi [13]) the latter is essentially a combined measure of the underlying symmetry *and* the contrast. Furthermore, comparing the lines of bilateral symmetry (from the phase symmetry image) with the points of high radial symmetry from Reisfeld et al.'s generalized *radial* symmetry, confirms that radial, rather than bilateral symmetry is a better detector of points of interest in Figs. 11, 12, and 13.

The discrete symmetry transform tends to highlight either side of high contrast lines, with the result that when such a line forms a ring, such as the wheels of the sports car in Fig. 11, it is strongly highlighted. However, there is also a lot of bold highlighting of noncircular edges and regions of high texture that do not exhibit radially symmetry, while detecting these features may be desirable for some applications they distract from the emphasis placed on radially



Original

Radial symmetry (fast dark)

Orientation-based (fast dark)

Radial symmetry (full)

Radial symmetry (fast)

Orientation-based symmetry (fast)

Sela & Levine's real-time attention mechanism [28]

Reisfeld's generalized symmetry (radial) [24]

Reisfeld's generalized symmetry (dark) [24]

Kovesi's symmetry from phase [13]

Di Gesù et al.'s discrete symmetry transform [6]

Circular Hough transform [19]

Fig. 13. Comparison of performance on a 256×341 image of a face in half shadow. The top two rows show the performance of the new transform, the bottom two rows show the output from other available transforms.

symmetric points, and detract from the performance of the transform as a symmetry-based interest detector.

Minor and Skalansky's implementation of the Circular Hough transform comes closest to rivaling the computational efficiency of the new transform, yet it provides only four levels of output. It was designed for detecting dark blobs in infrared images and when applied as a point of interest detector to the photographs shown here it detects many other points in addition to the primary interest points. In particular, it returns high values along edges, such as the frame of the mirror in Fig. 12, and is easily confused by textured surfaces, such as the grass in Fig. 11.

Table 4 lists the order of computation required to compute the transforms on an image of K pixels, where local symmetry is considered in an $N \times N$ neighborhood and for those methods which require gradient quantization the gradient is quantized into B bins. The complexity $O(KN)$ of the new transform is lower than all other transforms considered, with the exception of Di Gesù et al.'s discrete symmetry transform that has complexity $O(KN)$ or $O(KB)$. When calculating the discrete symmetry transform with complexity $O(KB)$ [5], it is essential to calculate it across four or more angular bins, whereas when calculating the new transform it is not necessary to compute it at all radii $1 \dots N$ (see Section 4.1). Likewise, the order $O(KN)$ implementation of the discrete symmetry transform [20] can be calculated at only a subset of the radii. However, the results from the discrete symmetry transform are quite different from the method presented in this paper, with edges and areas of high texture, in addition to points of radial symmetry, typically being awarded high responses.

The key to the speed of the new transform lies in the use of *affected pixels* to project the effect of gradient elements. This allows an approximation of the effect of each gradient element on the radial symmetry of the pixels around it, without specifically considering neighborhoods about each point like Lin and Lin [16] and Reisfeld et al. [24] or requiring multiple calculations for different gradient orientations, as do many other methods [6], [13], [19], [28].

Unlike other transforms the fast symmetry transform differentiates between dark and bright regions of radial symmetry, while allowing both to be computed simultaneously. Alternatively, just dark (or bright) points of

TABLE 3
Estimated Computation Required for Different Transforms

Transform	Computations (Mflop)		
	Figure 11	Figure 12	Figure 13
New Transform			
Radial Symmetry			
Full	17.9	18.9	23.4
Fast	8.06*	7.26*	8.51*
Fast Dark	6.76*	5.99*	7.4*
Orientation Symmetry			
Fast	7.44*	6.69*	7.87*
Fast Dark	6.45*	5.71*	7.09*
Existing Transforms			
Generalized Symmetry			
Radial [24]	300	259	349
Dark [25]	207	179	239
Circular Hough [19]	30	33.2	43.1
Symmetry from Local Phase [13]	601	196	912

* Note that the Fast and Fast Dark parameter settings ignore small gradients and are not calculated across all radiuses (see Table 2).

TABLE 4
Computational Order of Different Transforms

Transform	Order
New Radial Symmetry Transform	KN
Reisfeld's Generalized Symmetry Transform [24]	KN^2
Lin and Lin's Gradient-based Inhibitory Mechanism [16]	KN^2
Di Gesù et al.'s Discrete Symmetry Transform [5], [20]	KB or KN
Sela and Levine's Real-Time Attentional Mechanism [28]	KBN
Circular Hough Transform [19]	KBN

symmetry can be considered exclusively with an associated reduction in computation.

8 Conclusion

A point of interest detector has been presented that uses image gradient to locate points of high radial symmetry. The method has been demonstrated on a series of face images and other scenes and compared against a number of contemporary techniques from the literature. As a point of interest operator, the new transform provides equal or superior performance on the images tested while offering significant savings in both the computation required and the complexity of the implementation. The efficiency of this transform makes it well-suited to real-time vision applications.

References

[1] The USC-SIPI Image Database, Univ. of Southern Calif. Signal and Image Processing Inst., http://sipi.usc.edu/services/database/Database.html.

[2] Database of Faces, AT&T Laboratories, Cambridge, http://www.cam-orl.co.uk/facedatabase.html.

[3] The BioID Face Database, BioID Technology Research, http://www.bioid.de/bioid-db.zip, 2001.

[4] A. Chella, V. Di Gesù, I. Infantino, D. Intravaia, and C. Valenti, "A Cooperating Strategy for Objects Recognition," *Shape, Contour, and Grouping in Computer Vision,* pp. 264-276, 1999.

[5] V. Di Gesù and R. Palenichka, "A Fast Recursive Algorithm to Compute Local Axial Moments," *Signal Processing,* vol. 81, pp. 265-273, 2001.

[6] V. Di Gesù and C. Valenti, "The Discrete Symmetry Transform in Computer Vision," Technical Report DMA 011 95, Palermo Univ., 1995.

[7] V. Di Gesù and C. Valenti, "Symmetry Operators in Computer Vision," *Proc. First CCMA Workshop Vision Modeling and Information Coding,* Oct. 1995.

[8] R.O. Duda and P.E. Hart, "Use of the Hough Transform to Detect Lines and Curves in Pictures," *Comm. ACM,* vol. 15, no. 1, pp. 11-15, Jan. 1972.

[9] N. Intrator, D. Reisfeld, and Y. Yeshurun, *Extraction of Facial Features for Recognition Using Neural Networks,* 1995.

[10] S. Katahara and M. Aoki, "Face Parts Extraction Window Based on Bilateral Symmetry of Gradient Direction," *Proc. Eighth Int'l Conf. Computer Analysis of Image Patterns,* pp. 489-497, Sept. 1999.

[11] L. Kaufman and W. Richards, "Spontaneous Fixation Tendencies of Visual Forms," *Perception and Psychophysics,* vol. 5, no. 2, pp. 85-88, 1969.

[12] C. Kimme, D. Ballard, and J. Sklansky, "Finding Circles by an Array of Accumulators," *Comm. ACM,* vol. 18, no. 2, pp. 120-122, Feb. 1975.

[13] P. Kovesi, "Symmetry and Asymmetry from Local Phase," *Proc. 10th Australian Joint Conf. Artificial Intelligence,* 1997.

[14] P. Kovesi, "Image Features from Phase Congruency," *Videre: J. Computer Vision Research,* vol. 1, no. 3 1999.

[15] P. Kovesi, "Matlab Code for Calculating Phase Congruency and Phase Symmetry/Asymmetry," http://www.cs.uwa.edu.au/p~k/Research/phasecong.m, 1999.

[16] C.-C. Lin and W.-C. Lin, "Extracting Facial Features by an Inhibitory Mechanism Based on Gradient Distributions," *Pattern Recognition,* vol. 29, no. 12, pp. 2079-2101, 1996.

[17] P. Locher and C. Nodine, "Symmetry Catches the Eye," *Eye Movements: from Physiology to Cognition*, J. O'Regan and A. Levy-Schoen, eds., Elsevier Science Publishers B.V., 1987.

[18] G. Loy and A. Zelinsky, "A Fast Radial Symmetry Operator for Detecting Points of Interest in Images," *Proc. European Conf. Computer Vision,* 2002.

[19] L.G. Minor and J. Sklansky, "Detection and Segmentation of Blobs in Infrared Images," *IEEE Trans. Systems, Man, and Cyberneteics,* vol. 11, no. 3, pp. 194-201, Mar. 1981.

[20] R.M. Palenichka, M.B. Zaremba, and C. Valenti, "A Fast Recursive Algorithm for the Computation of Axial Moments," *Proc. 11th Int'l Conf. Image Analysis and Processing,* pp. 95-100, 2001.

[21] C.M. Privitera and L.W. Stark, "Evaluating Image Processing Algorithms that Predict Regions of Interest," *Pattern Recognition Letters,* vol. 19, pp. 1037-1043, 1998.

[22] C.M. Privitera and L.W. Stark, "Algorithms for Defining Visual Regions-of-Interest: Comparison with Eye Fixations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 9, pp. 970-982, Sept. 2000.

[23] D. Proffit and J. Cutting, "Perceiving the Centroid of Curvilinearly Bounded Rolling Shapes," *Perception and Psychophysics,* vol. 28, no. 5, pp. 484-487, 1980.

[24] D. Reisfeld, H. Wolfson, and Y. Yeshurun, "Context Free Attentional Operators: The Generalized Symmetry Transform," *Int'l J. Computer Vision,* special issue on qualitative vision, vol. 14, pp. 119-130, 1995.

[25] D. Reisfeld and Y. Yeshurun, "Preprocessing of Face Images: Detection of Features and Pose Normalisation," *Computer Vision and Image Understanding,* vol. 71, no. 3, pp. 413-430, Sept. 1998.

[26] D. Reisfeld, "Generalized Symmetry Transforms: Attentional Mechanisms and Face Recognition," PhD thesis, Tel-Aviv Univ., Jan. 1993.

[27] W. Richards and L. Kaufman, "Center-of-Gravity Tendacies for Fixations and Flow Patterns," *Perception and Psychophysics,* vol. 5, no. 2, pp. 81-84, 1969.

[28] G. Sela and M.D. Levine, "Real-Time Attention for Robotic Vision," *Real-Time Imaging,* vol. 3, pp. 173-194, 1997.

[29] Q.B. Sun, W.M. Huang, and J.K. Wu, "Face Detection Based on Color and Local Symmetry Information," *Proc. Third Int'l Conf. Face and Gesture Recognition,* pp. 130-135, 1998.

[30] O. Sutherland, H. Truong, S. Rougeaux, and A. Zelinsky, "Advancing Active Vision Systems by Improved Design and Control," *Proc. Int'l Symp. Experimental Robotics,* Dec. 2000.

[31] H. Yamamoto, Y. Yeshurum, and M. Levine, "An Active Foveated Vision System: Attentional Mechanisms and Scan Path Convergence Measures," *CVGIP: Image Understanding,* 1994.

Gareth Loy received the BSc degree in mathematics from the Australian National University in 1997 and the BE degree in systems engineering in 1999. He is currently completing the PhD degree in robotics at the Australian National University researching computer vision for human computer interaction. During his PhD, he has spent several months on complementary research projects at the University of Western Australia and the Humanoid Interaction Lab at AIST, Japan, has undertaken consulting work as a research scientist for Seeing Machines, and has written an undergraduate lecture course in computer vision.

Alexander Zelinsky received the PhD degree in robotics in 1991. He worked for BHP Information Technology as a computer systems engineer for six years before joining the University of Wollongong, Department of Computer Science as a lecturer in 1984. Since joining Wollongong University, he has been an active researcher in the robotics field. Dr. Zelinsky spent nearly three years (1992-1995) working in Japan as a research scientist with Professor Shinichi Yuta at Tsukuba University, and Dr. Yasuo Kuniyoshi at the Electrotechnical Laboratory. In March 1995, he returned to the University of Wollongong, Department of Computer Science as a senior lecturer. In October 1996, Dr. Zelinsky joined the Australian National University, Research School of Information Science and Engineering as Head of the Robotic Systems Laboratory, where he is continuing his research into robotics, smart cars, and human-robot interaction. In January 2000, Dr. Zelinsky was promoted to professor and head of systems engineering at the Australian National University. Dr. Zelinsky is a member of the IEEE Robotics and Automation Society, a senior member of the IEEE, the IEEE Computer Society, and was President of the Australian Robotics & Automation Association (1998-2000).