UNIVERSITY OF SOUTHAMPTON

The Image Ray Transform

by

Alastair H. Cummings

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Physical and Applied Sciences
School of Electronics and Computer Science

March 2012

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL AND APPLIED SCIENCES
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Alastair H. Cummings

Image feature extraction is a fundamental area of image processing and computer vision. There are many ways that techniques can be created that extract features and particularly novel techniques can be developed by taking influence from the physical world. This thesis presents the Image Ray Transform (IRT), a technique based upon an analogy to light, using the mechanisms that define how light travels through different media and analogy to optical fibres to extract structural features within an image. Through analogising the image as a transparent medium we can use refraction and reflection to cast many rays inside the image and guide them towards features, transforming the image in order to emphasise tubular and circular structures.

The power of the transform for structural feature detection is shown empirically in a number of applications, especially through its ability to highlight curvilinear structures. The IRT is used to enhance the accuracy of circle detection through use as a preprocessor, highlighting circles to a greater extent than conventional edge detection methods. The transform is also shown to be well suited to enrolment for ear biometrics, providing a high detection and recognition rate with PCA, comparable to manual enrolment. Vascular features such as those found in medical images are also shown to be emphasised by the transform, and the IRT is used for detection of the vasculature in retinal fundus images.

Extensions to the basic image ray transform allow higher level features to be detected. A method is shown for expressing rays in an invariant form to describe the structures of an object and hence the object itself with a bag-of-visual words model. These ray features provide a complementary description of objects to other patch-based descriptors and have been tested on a number of object categorisation databases. Finally a different analysis of rays is provided that can produce information on both bilateral (reflectional) and rotational symmetry within the image, allowing a deeper understanding of image structure. The IRT is a flexible technique, capable of detecting a range of high and low level image features, and open to further use and extension across a range of applications.

Contents

List of Figures

List of Tables

List of Pseudocode Functions

List of Symbols

ϕ	Random initial direction of a ray
τ	The target intensity
$\mathbf{A}$	The IRT accumulator
I'	The transformed image
$\mathbf{I}$	The image
$\mathbf{R}_L$	Vector direction of the reflected ray
$\mathbf{R}_R$	Vector direction of the refracted ray
θ_C	The critical angle
θ_I	The angle of incidence
θ_L	The angle of reflection
θ_R	The angle of refraction
D	The RMS difference
d	Maximum number of direction changes (depth) a ray can undergo before it is terminated
D_S	Lower limit (stopping condition) for RMS difference measure
k	Parameter controlling the exponential refractive index model for the IRT
l	Maximum distance (length) a ray can travel before it is terminated
N	The maximum number of rays to cast during the IRT
n	Refractive index of a medium
n_i	Refractive index at intensity i
$n_{\max}$	The maximum refractive index in the linear model for the IRT
$\mathbf{E}$	Edge magnitude

$\mathbf{N}$	Normal of the boundary between two media
$\mathbf{p}^{<t>}$	Position vector of a ray at step t
$\mathbf{V}$	Direction of a ray
γ_v	The fraction of the beam's mean that each ray is adjusted by
B	The width of the beam and the number of rays within it
λ'_b	The cumulative length of an equalised ray up to point b
λ_a	The cumulative length of a ray up to point a
ϕ_b	The angle of an equalised ray segment b
$\psi_{g,c}$	The angle of segment c of an equalised ray at scale g
d_r	The depth (number of segments) that a specific ray has been traced
g	The scale of a ray feature, between 1 and g_{max}
g_{max}	The largest scale of a ray feature
$h_{g,e}$	The ray feature (angle difference of ψ) at scale g and index e
l_a	The length of a single segment of a ray
l_q	The length of every segment in an equalised ray
m	The number of segments of equalised length in a ray feature
$o(x)$	Piecewise function that is 0 if $x < 0$ and x otherwise
$w_{a,b}$	The amount that segment a contributes to equalised segment b's direction
$\mathbf{V}'_b$	The direction of a segment of an equalised ray
$\mathbf{V}_a$	The direction of a single segment of a ray
$\Delta_{u,v}$	Difference between two ray feature descriptors
ρ	Threshold for similarity between two descriptors
σ_G	Scale parameter for the GST
σ_R	Scale parameter for the reflectional symmetry IRT
$C(i,j,\sigma)$	GST symmetry function between pixels
$D(i,j,\sigma)$	GST distance function between pixels
$E_L(i)$	Logarithmic edge magnitude at a pixel

$P(i, j)$	GST phase function between pixels
R	Set of pixels that a ray travels through
$\mathbf{S}$	Symmetry accumulator for the reflectional symmetry IRT

List of Acronyms

AD anisotropic diffusion

AUC area under curve of a ROC graph

DoG Difference of Gaussian

GST generalised symmetry transform

HT Hough transform

ICA independent component analysis

IRT Image Ray Transform

IRT-n IRT with linear refractive indices

IRT-k IRT with exponential refractive indices

IRT-nk IRT with aggregated linear and exponential results

MAA maximum average accuracy

NBC naïve Bayes classifier

PCA principal component analysis

RMS root mean squared

ROC reciever operating characteristic

SIFT scale invariant feature transform

Declaration of Authorship

I, Alastair H. Cummings, declare that this thesis titled, "The Image Ray Transform" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Acknowledgements

Firstly my thanks go to my supervisor Prof. Mark S. Nixon for his help throughout the course of my PhD. Whilst his advice and guidance in my research and writing were always invaluable, I am most grateful for the trust he showed in me to develop the ideas in this thesis even when the outcome was far from certain.

Thanks also go to my other supervisor Dr. John N. Carter for providing the initial seed for many of the core ideas of my work as well as his feedback on both the physics and my publications. I am also grateful for Dr. Tony Mansfield and the National Physical Laboratory for their support and interest in my work.

I'd like to thank my friends and colleagues from ISIS, ECS and the ORC for their assistance and friendship over the last few years.

Finally, my gratitude goes to my parents, for their love and encouragement throughout my PhD and everything beforehand and to Kate for being my ever-present support in everything I do.

Chapter 1

Introduction

The detection of structural features within images is important to a wide range of computer vision and image processing applications. From medical imaging to biometrics, shape detection to object categorisation, all can be enabled and enhanced through the improved detection of structures. Tubular (or curvilinear) structures are one example of a structural feature, being a ribbon of smooth intensity surrounded by a different intensity, and can be found in a wide array of images, such as blood vessels in medical images or the structure in any number of natural or synthetic objects. Much work on tubular features has concentrated on solving the problem for 3D medical images [48] and exhaustive analysis of cross-sections [50]. Many tubular structures in the physical world are used to direct the flow of fluids, such as water in pipes or blood in blood vessels, but another interesting example is the way in which tubular optical fibres are used to guide light along their length. If the image can be analogised in such a way that tubular structures in the image were made to act as if they were optical fibres in the physical world, guiding light along them, this could be detected and exploited to create a powerful structural feature detector. This thesis aims to discover whether it is possible to analogise an image in such a way, and if so what form does the resultant technique take, and what applications can it be used for?

1.1 An Optical Analogy for Structural Feature Detection

One method of creating a technique where tubular structures could be detected through analogy to optical fibres is to transform the image so that tubular structures act as if they were glass and other areas of the image act as the surrounding air, recreating the physical situation under which optical fibres exploit total internal reflection to guide light. If we were to define the optical properties of a transparent medium based upon the properties of the image, and were to shine a domestic torch through this medium, the beam created by the torch would reflect and refract according the the properties of

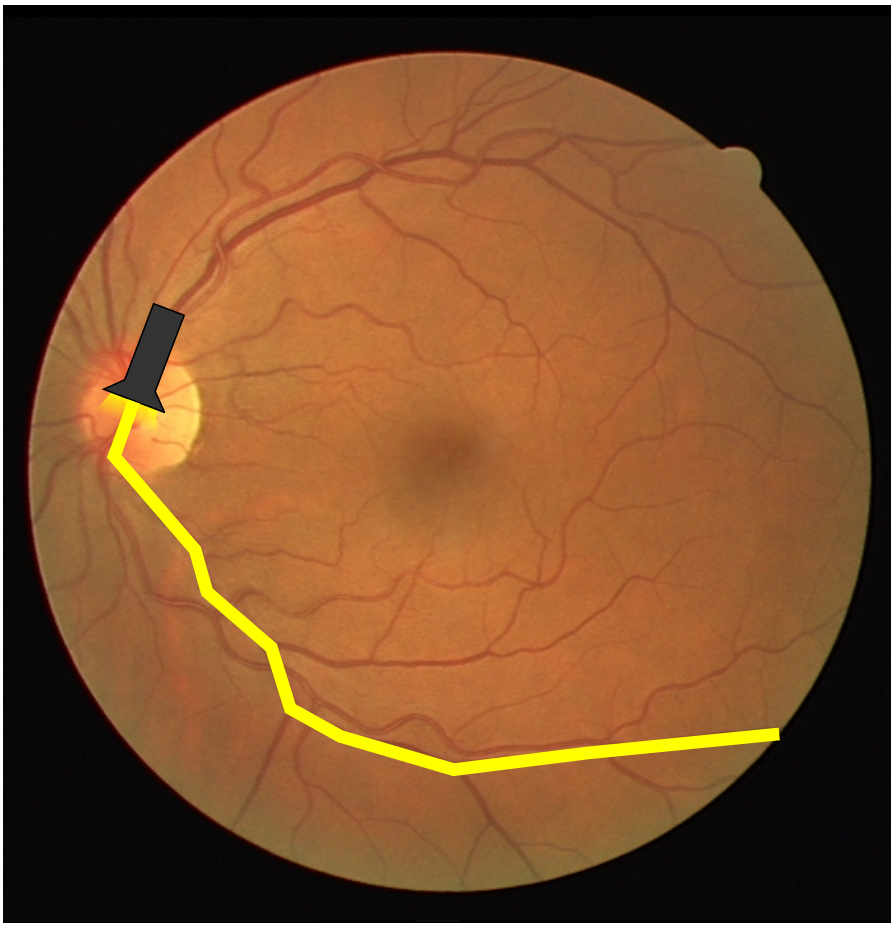


FIGURE 1.1: Illustration of a single ray from the IRT.

the image. When light entered a part of the transparent medium that resembled that of an optical fibre, it would travel along its length, guided by total internal reflection. In figure 1.1 we can see how the path that the light takes can be influenced by the image, in this case following the length of a blood vessel. Through accumulation of the paths of many torches shone from many different positions, a transform can be produced that highlights tubular structures in a novel fashion. The technique that has been developed is based on using light rays rather than torches and is called the Image Ray Transform (IRT).

In addition to tubes, other structures can be highlighted by the transform. Smoothly curving structures, such as those found in circles, can direct light in a similar fashion to optical fibres, needing only one bounding edge to repeatedly reflect the light along its length. Many other less easily defined structures can also be highlighted by the transform.

1.1.1 Structural Feature Detection

There are myriad possible applications of a structural feature detector such as the IRT; three domains where the structural feature detection capability of the transform was empirically assessed were circle detection, enrolment of ear biometrics and segmentation of the retinal vasculature. Circles are a basic image shape that are commonly detected with the Hough transform (HT). The quality of this detection can be improved through use of the IRT as a preprocessor to highlight circle boundaries. Ears are a novel biometric, with a number of advantages over faces that make them useful in certain situations[36], changing little with age, being more easily collected with less anxiety than other biometrics and being unchanged by expression. Through use of the IRT we can improve the enrolment procedure, detecting and normalising the ear, and subsequently increase the recognition rate across a database. Detection of blood vessels within retinal fundus images is an important application to aid in the diagnosis and monitoring of ocular diseases like diabetic retinopathy [62]. By applying two variations of the IRT as a preprocessor, the retinal vasculature can be highlighted and segmented using simple thresholding techniques.

1.1.2 High Level Features

An object within an image can be identified by a list of features that the object contains. Such features are often based upon patches or points, describing them with invariance to rotation, scale and translation to enable recognition of the object across different images. The many rays that are used to apply the IRT to an image can also be used as invariant feature descriptors, each one describing a different structure of an object. This approach contrasts and complements other approaches that focus upon local features rather than structure that may span the entire object. We show this feature descriptor improving object categorisation with a bag-of-words model.

Symmetry is an important high level feature that is often expensive or difficult to calculate. Through extension to the IRT, a fast, edge-focused reflectional symmetry operator can be created, providing a significant improvement over the complexity of the generalised symmetry transform (GST) [74]. A rotational symmetry operator can also be derived, exploiting the rotational invariance of the ray-based feature descriptor to find centres of rotational symmetry throughout images.

1.2 Thesis Overview

This thesis describes the creation of a technique based upon an analogy to light, the Image Ray Transform, and its application to structural feature detection, as a feature

descriptor, and as a symmetry operator. The chapters are organised as follows. Chapter 2 sets the contributions into context, describing other techniques based upon physical analogies and explaining the motivation for using light, as well as summarising work and relevant techniques of the applications on which we use the IRT. Chapter 3 details the IRT, the law of optics behind it, extensions to the basic transform, and an analysis of parameters. Chapter 4 describes the application of the IRT as a structural feature detector to circle detection, ear biometrics and medical images. Chapter 5 introduces the invariant feature descriptor based upon the IRT as well as its use in experiments on object categorisation and chapter 6 presents some preliminary work on symmetry and other variations of the transform. Finally chapter 7 draws overall conclusions, identifies possible future directions of research for the transform.

1.3 Related Publications

A number of publications have arisen from this work. The initial definition of the IRT, the different models of refractive index, target intensities, and the transform's application to circle detection (sections 3.2, 3.3.2, 3.3.3 and 4.1) were initially published in:

- A. H. Cummings, M. S. Nixon, and J. N. Carter. Circle detection using the image ray transform. In *Int'l Conf. Computer Vision Theory and Applications (VISAPP 2010)*, 2010.

The work on applying the IRT to enrolment for ear biometrics, as well as the addition of a stopping condition (sections 3.3.1 and 4.2) was published in:

- A. H. Cummings, M. S. Nixon, and J. N. Carter. A novel ray analogy for enrolment of ear biometrics. In *4th IEEE Int'l Conf. on Biometrics Theory, Applications Systems (BTAS 10)*, 2010.

The use of the IRT to segment blood vessels from retinal images described in section 4.3 was published as:

- A. H. Cummings and M. S. Nixon. Retinal vessel extraction with the image ray transform. In *6th Int'l Symp. on Visual Computing (ISVC10)*, 2010.

A complete definition of the IRT, beam IRT, and the work on ear recognition on IRT enrolled images and noisy images with beam IRT enrolment (chapter 3 and section 4.2.3) has been published as:

- A. H. Cummings, M. S. Nixon, and J. N. Carter. The image ray transform for structural feature detection. *Pattern Recognition Letters*, 32(15):2053–2060, 2011.

Finally the work extending the IRT to symmetry and feature descriptors (sections 6.1 and 6.2 and chapter 5) is currently submitted as:

- A. H. Cummings, M. S. Nixon, and J. N. Carter. Using features from the image ray transform for object categorisation and symmetry. *Computer Vision and Image Understanding*, In review, 2011.

Chapter 2

Background

This chapter discusses some other computer vision techniques that have employed physical analogies as their basis, covering heat, force, water and light. It also describes a number of applications where the use of structural feature detector like the Image Ray Transform (IRT) is prudent.

The chapter is arranged as follows. In section 2.1 we describe previous work on physical analogies, detailing previous techniques based upon heat, force, water and light. Section 2.2 discusses circle detection and common methods, section 2.3 reviews ear biometrics research and section 2.4 describes the problem of retinal vasculature detection. The background, tools and techniques of object categorisation are detailed in section 2.5 and section 2.6 explains the importance of symmetry and details some methods by which it can be found.

2.1 Physical Analogies for Computer Vision

There are a number of ways to approach the creation of techniques for computer vision. As we choose to base the IRT on a way of treating tubular structures in an image as optical fibres, and shining light through them, the IRT belongs to a class of vision techniques that are based upon analogies to physical phenomena, a paradigm that provides some unique advantages [64] over others. Through analogy to easily understood natural concepts, techniques can be developed that are less abstract and more readily comprehended than many other methods. An analogy based technique also provides advantages in parameterisation, as parameters often relate to physical measures and so can be manipulated more intuitively than may occur with parameters in other, more abstract techniques. Lastly, the connection to an analogy is itself flexible, the aim being to provide a structure for the technique but not an accurate simulation, and the method can be adapted from the definition to improve its results whilst still maintaining the

advantages that the analogical link provides. Heat flow, force fields, and water are all common analogies for techniques in computer vision but the basis of the IRT is an analogy to light.

2.1.1 Heat Flow

Analogies to heat flow have been used many times in the creation of feature extraction techniques, often due to its inherent smoothing ability. Anisotropic diffusion (AD) [67] is an image smoothing operator that is superior to techniques such as Gaussian smoothing due to its edge-aware nature. It models the diffusion of heat (as intensity) through an image and occurs iteratively, each iteration at a coarser scale. The heat flow is specified to occur to a greater extent in areas of low edge strength, where little detail will be lost and to a lesser extent near strong edges. As edges are one of the most important low level features available in an image, anisotropic diffusion enables noise to be eliminated whilst maintaining salient features.

Anisotropic diffusion and heat flow have many applications in other vision problems. With the addition of a temporal dimension to the AD equation, movement can be detected. Direkoğlu and Nixon [27] developed a method to find moving edges in images. They employed anisotropic diffusion to remove noise and emphasize high contrast edges and then used heat flow in the temporal dimension to find movement of edges. Makrogiannis and Bourbakis [55] took a similar spatio-temporal diffusion approach to detect areas of motion. The smoothing properties of heat flow are also useful for segmentation. Direkoğlu and Nixon [28] used heat flow to segment the image into non-contiguous regions, and then geometric heat flow (AD along edges) to smooth the boundaries. Manay and Yezzi [56] also used anti-geometric heatflow across edges rather than homogeneous regions for segmentation.

2.1.2 Force

Force fields, such as gravitational and electromagnetic fields, provide a method for vision techniques to exploit the inverse square law, allowing image information to be weighted by proximity. Hurley et al.'s force field transform [37] generates a force field from an image that is analogous to a gravitational or magnetic field. Each pixel is assumed to attract every other pixel with a force dependent on its intensity and the inverse square law. The force $\mathbf{F}$ at each pixel is found as follows, where $P(\mathbf{r}_i)$ is the intensity of the pixel at position $\mathbf{r}_i$:

$$\mathbf{F}(r_j) = \sum_{i\,|i\neq j} P(\mathbf{r}_i) \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3}. \tag{2.1}$$

The sum of these forces generates a vector field representing an image. This force field can help in feature extraction and has been used to create an ear biometric. It is

interesting to note that Equation 2.1 is very similar to the equation for the mean shift algorithm [33], both weighting data by its proximity to a point, but differing in the choice of points that this value is evaluated at (all points with the force field transform as opposed to a number of iteratively updated estimated means in mean shift).

Other approaches have combined electromagnetic field analogies and active contours. Jalba et al. [44] used their Charged Particle Model for shape recovery and segmentation by generating an electric field from the edge map function. This approach allowed charged particles to be guided to contours by this field and their interactions with the fields caused by other particles. In the model of Xie and Mirmehdi [80] the image border and evolving contour are assumed to have an electric current running through them, and it is the interaction of these currents that generates a force field. This field guides the development of the contour, changing with it to guide the contour to the image border.

2.1.3 Water

Two different approaches have been employed to enable use of a water analogy. The earlier and more mature analogy is the watershed method and is concerned with collection of water across a landscape. In geography, a watershed is the division of a continent where water flows into different drainage basins on either side. For example the Great Divide along the Rockies and Andes mountain ranges dividing America between draining into the the Pacific and Atlantic oceans on the west and east sides. This idea is easily applicable to the problem of image segmentation if the image is considered to be a landscape with height linked to intensity. Although the possible use of this analogy was first identified by Digabel and Lantuejoul [26], the first such algorithm was described by Beucher and Lantuejoul [8], and the method continues to be developed. A major weakness of the watershed method, however, is that in most cases it will over-segment images, and manually placing markers or combining with another technique is necessary to sensibly combine segments and improve the results.

An alternative approach is to base the analogy on the physical properties of water flow such as adhesion and velocity. Liu and Nixon [51] created the water flow algorithm for the purposes of image segmentation, with a particular focus on vascular features such as blood vessels or rivers. It is an iterative region growing algorithm that begins from a source point at which there is a flow of water into the image. The water front is repeatedly expanded by comparing forces such as the driving force, adhesion to edges, resistance from image related obstacles and statistical forces ensuring homogeneity. It compared favourably to other segmentation techniques, and was tested on a variety of medical imaging applications.

2.1.4 Light

There are a number of techniques that take influence from some aspects of light rays or bear some similarity to their properties. An example of such an approach applicable to vision is the Eikonal equation. The solution of the Eikonal equation describes the shortest amount of time that a ray starting at the boundary of a set Ω in $\mathbb{R}^n$ will take to reach a point $x \in \Omega$. It takes the form

$$|\nabla u(x)| = F(x), \tag{2.2}$$

where $u(x)$ is the solution and $F(x)$ is an input describing the time cost at x and u on the boundary of Ω is 0. Previous vision techniques have used the Eikonal equation as a distance metric to enable shape from shading and watershed segmentation [57] but these do not attempt to take full advantage of the analogy. A method with a slightly stronger analogical base is the digital engraving system of Pnueli and Bruckstein [70], which uses a potential field generated from the application of the Eikonal equation to an image to create a halftone version of that image. This halftone image is composed of a number of lines that reproduce the intensities of the original image by being more densely spaced in some areas than others.

Previous work has considered techniques that bear some resemblance to the concept of tracing rays through images, used in the IRT. This method does not approach feature extraction from a strongly analogical direction, as we do, nor use the concept in the same way, but is still of interest nonetheless. The JetStream contour extraction technique by Perez et al. [66] is capable of extracting silhouettes or roads by contour evolution. From a seed point, the contour is extended by short segments whose direction is determined probabilistically, based upon the properties of the image and constrains the shape of the curve.

2.2 Circle Detection

The Hough transform (HT) is a standard method of shape detection in images. The HT was originally designed for detecting lines [35], but can be applied to the detection of arbitrary shapes, including circles. It exploits the relationship between a shape in an image (with x and y values) and the corresponding Hough space, inferring information about the parameters of a shape from each edge point (with lines these are the y intercept and gradient). For all edge points in the image, we accumulate in Hough space for every shape which an edge point could comprise (in the line HT this involves drawing a line in Hough space for every image point). The peaks in the Hough space accumulator provide the parameters of the detected shapes. In extension to circles, we require a 3D Hough space accumulator, with position (x, y) and radius (r). Accumulation in Hough space

occurs in a cone, centred on the x, y position and of radius r at each layer in the radius dimension.

The naïve 3D HT has a high computational cost, especially when the range of possible radii is large. Yuen et al. [84] review a number of variations upon the circle HT that can reduce computational cost, and we detail a version[25] that decomposes the transform into two stages: finding the centre position and finding the radius. The position accumulation employs a 2D accumulator and uses the normal of the edge direction at each point to draw a line that passes through all possible centres for all possible radii. The peak of this accumulator is then taken as the centre of the circle. The radius accumulation uses a 1D accumulator, checking each circle of points for each radius around the detected centre and accumulating at edge points. After normalising for radius size the peak of this accumulator provides the radius of the circle.

2.3 Ear Biometrics

The use of ears for identification has a long history, dating back to the anthropometric system of Bertillon [7] in the late nineteenth century. His system included measurements of ear length as a method to aid identification of criminals. Iannarelli [39] was the first to perform a comprehensive study into the uniqueness of individual ears, creating an identification system based upon measurements of parts of the ear. Iannarelli's study provided the first evidence of the uniqueness of the ear, even between identical twins, and showed that the general shape of the ear is constant, merely growing in size with age. Figure 2.1 shows the ear and labels important parts of the pinna (outer ear). Between subjects the shape of the helix, internal ridges such as the antihelix, as well as features such as the tragus, intertragic notch and the lobe vary significantly.

Automated ear biometrics have been of interest more recently, and some research suggests that they have similar performance to face recognition [14]. Automated systems take a variety of approaches, some similar to the manual biometrics of the past, others exploiting the entire structure of the ear for accuracy. The ear provides advantages over other biometrics such as faces through its predictable surroundings and invariance to expression and ageing, although it is also prone to occlusion from hair. The ear biometric problem (as with all biometrics) is split between enrolment and recognition. Enrolment is the discovery, localisation and normalisation of the ear image, whilst recognition deals with identification of a subject by ears.

2.3.1 Ear Recognition

The creation of a biometric requires the discovery of unique features that can be measured and compared in order to correctly identify subjects. The first automated example

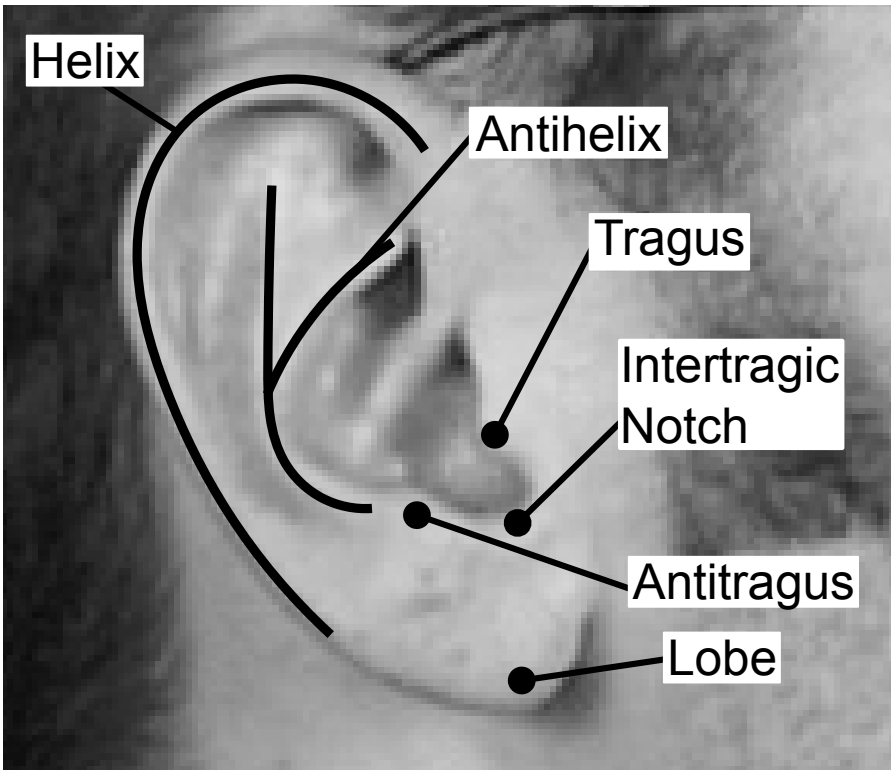


FIGURE 2.1: Parts of ear anatomy. From an image from the XM2VTS database.

of an ear biometric is that of Burge and Burger [10], using the significant edges of the ear to generate a Voronoi diagram and proving the viability of the ear as a biometric. A number of other methods take an approach focusing upon the geometry and structure of the ear. Moreno et al. [59] used feature points of the ear common in forensics as well as using cuts across the ear to produce a feature vector, while Chorás [18] measured intersections of the ear contours with circles radiating from the centre of the ear. Hurley et al. [37] used the forcefield transform (section 2.1.2) to create an ear biometric by following the direction of the force field to find unique potential energy wells. Later work [38] used a convergence operator (the inverse of the vector field divergence operator) to find similar features, and Dong and Mu [29] expanded upon this using linear discriminant analysis, although with limited success. A geometric approach was also taken by Mu et al. [60], using the shape of outer ear contour as well as the location of various intersections within the ear. There have also been a number of approaches using 3D representations of the ear with techniques such as morphable models [12], local shape features [16] and iterative closest point matching [42, 83].

The most famous technique used for face recognition [78], principal component analysis (PCA) , is also suitable for use in ear recognition. PCA [46] is an orthogonal transform of a dataset to new axes that causes the projection onto the first axis (principal component) to represent the greatest variance in the data, and subsequent orthogonal axes to represent decreasing amounts of variance. These principal components are the

FIGURE 2.2: The first 18 eigen-ears found through PCA on the XM2VTS database.

eigenvectors of the covariance matrix of the data, and the original data can be expressed as a weighted combination of a subset of these vectors. In biometrics the eigenvectors can be displayed as eigen-ear images (or originally, eigen-faces), showing the areas of greatest variance, and the original images can be approximated by a combination of the eigen-ears. The first 18 eigen-ears calculated from performing PCA on images from the XM2VTS biometric database [58] are shown in figure 2.2. The first eigen-ear is primarily concerned with illumination and hair, but the others highlight different areas of the pinna. Recognition can be performed by projecting images onto a subset of the eigen-ears and finding gallery images that project similarly to a probe image.

The first application of PCA to ear recognition was by Victor et al. [79], although they found it to be inferior to face recognition. Chang et al. [14] also used PCA and found that ears provided similar performance to faces, and that a fusion of the two biometrics produced the best results. A similar technique, independent component analysis (ICA), was also used by Zhang et al. [87] with performance superior to that of PCA.

2.3.2 Ear Enrolment

The majority of the recognition techniques described above require, and have been tested on, manually enrolled datasets. These datasets have been normalised for scale, position and rotation by hand and provide a good test data for the development of new recognition methods. However, any real world use of ear biometrics requires that this enrolment step

be done automatically and in this sense good enrolment techniques are as important as strong recognition techniques. This especially true as the best methods, such as PCA are very sensitive to poor enrolment.

Enrolment has not received the same level of attention as recognition, but recent research has begun to tackle the problem. One method for locating the ear is to exploit its elliptical shape and use common shape detection methods. Alvarez et al. [3] proposed a combination of active contours and an ovoid model, whilst Arbab-Zavar and Nixon [4] used a form of the elliptical HT on the edges of the ear to detect it. Colour was used in addition to shape by Prakash et al. [71] suggesting that colour can augment the enrolment process. The use of feature points within the ear in order to detect and transform the ear to a common model is also used widely. Ibrahim et al. [40] used the response of the ear to a bank of banana wavelets (gabor wavelets with curvature) to detect it, whilst Bustard and Nixon [11] used scale invariant feature transform (SIFT) feature points and a homography transform to fit ear images to a model. An intensive training approach was taken by Islam et al. [41] using weak classifiers based upon Haar wavelets combined using AdaBoost to create a very potent ear detection method.

2.4 Retinal Vascalature Detection

The shape of blood vessels in the retina of the eye can provide valuable information enabling the diagnosis and monitoring of a number of diseases, including diabetic retinopathy. Through segmentation of the vasculature this information can be provided automatically, and the progress of any disease measured accurately. Figure 2.3 shows some examples of retinal fundus images from the DRIVE database [77]. The vascular features vary in scale and intensity, and the background texture of the retina is also highly variable. In addition there are areas such as the optic disc (high intensity, source of the vessel tree) and the fovea (dark spot often found in centre of the images) that provide starkly different contexts for the detection of vessels.

The identification of this application and the first attempt at a solution were provided by Chaudhuri et al. [15], using matched filters to highlight vascular features. Recent work has expanded upon matched filters with the addition of a first-order derivative of Gaussian filter and adaptive thresholds [86], and through training of the filter parameters [1]. Adaptive local thresholding was used by Jiang and Mojon [45] in order to deal with the range of intensities found in fundus images. A morphological approach was taken by Zana and Klein [85], focusing upon the shape and tree structure of vessels to aid detection. The segmentation technique based upon an analogy to water flow described in section 2.1.3 was also used to segment retinal blood vessels [52]. Other unsupervised approaches taken to this task have split the the images into small and large vessels, treating each differently so as to improve the overall result [2, 73]. Niemeijer et al.

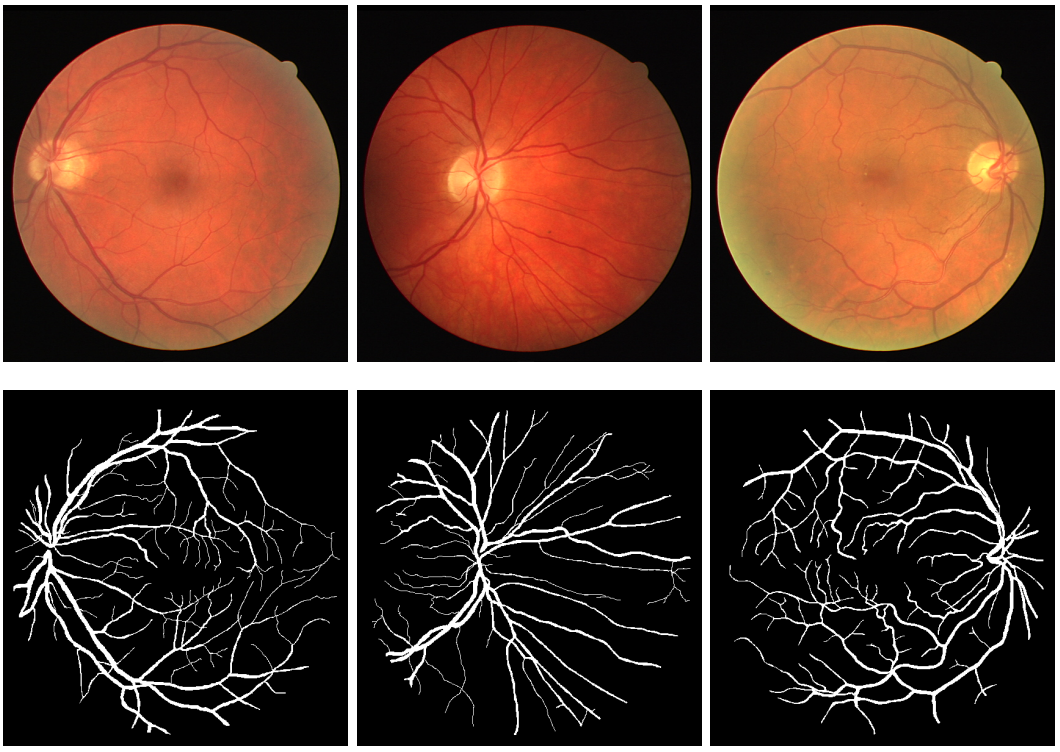


FIGURE 2.3: A selection of fundus images and their ground truth from the DRIVE database.

[62] applied many of these techniques to the DRIVE database. Staal et al. [77] used a ridge based classification system to segment the images, whilst Soares et al. [76] used supervised classification and Morlet wavelets to detect the vessels.

2.5 Object Categorisation

Object categorisation is a challenging application of computer vision. It involves the classification of objects within an image into a number of categories (vehicles, people, trees etc.) through the detection and description of salient features and learning which of those features characterise different categories of objects. The features can be found from a range of image properties including gradient, colour, intensity or structure, and a common method of classifying these features is through the bag-of-visual-words model.

2.5.1 Feature Descriptors

A wide range of feature descriptors have been used for object categorisation. Feature descriptors usually describe the local neighbourhood of a detected feature point in some invariant way that is conducive to comparisons with other descriptors. These points may be selected randomly, through a interest point detector such as the Difference of

Gaussian (DoG) [53] detector, or at regular intervals throughout the image. A simple example of a feature descriptor would be a local intensity or colour histogram, describing the patch around the point in a rotationally invariant manner, but sacrificing detail about the orientation and spatial structure.

An example of a simple descriptor that provides some degree of scale invariance whilst maintaining spatial information are the scaled intensity patches used by Fei-Fei and Perona [32]. This method extracts a square intensity patch of random size between 10 and 30 pixels around the selected point. This patch is then scaled to an 11x11 patch, providing an amount of scale invariance.

SIFT [53] is perhaps the most commonly used feature descriptor, providing a scale and rotationally invariant method to describe the neighbourhood of a feature point, usually detected in scale space by the DoG operator. The interest points found by the DoG detector are determined on a version of the image with a specific scale, and descriptor creation is carried out on this scaled image, providing scale invariance. Additionally, the dominant orientation around the point is found by examining a local histogram of gradient directions, and the calculation of the descriptor is normalised to account for this. The descriptor itself is generated from a number of local histograms of orientation. The 16×16 pixel area around the point is divided into sixteen windows, and an eight bin histogram is created for each window and filled by the orientations of the pixels inside, weighted by the distance to the original keypoint. The normalised and flattened version of these sixteen, eight bin orientation histograms provides the 128 value feature descriptor.

2.5.2 The Bag-of-Visual-Words Model

The bag-of-words model is a common method used for analysis of text corpora, providing a way to easily store and compare many documents. A text document can be characterised by counting the number of times each word appears, which can then be used in categorisation or the retrieval or similar documents. For example, a document about car maintenance may often mention words such as “oil”, “engine” or “tyres” whereas a document concerned with economics might have many occurrences of “oil”, “price” or “GDP” but few mentions of car-related words. Similar documents can be found by calculating the distance between these words counts, and categorisation can be performed by using them as a feature vector for a classifier.

Extension of the model to vision was initially performed by Zhu et al. [88] and requires a different approach, as there are no obvious words in the visual domain. Pixels alone mean very little; for instance, there is no difference between a black pixel on a car and a black pixel on a road, and some context is required in order to make meaningful visual words. This context can be provided by feature descriptors of the type described in

section 2.5.1, often based upon the intensity or orientation of patches. Whilst the use of feature descriptors provides the "alphabet" for our words, we must create a finite dictionary or codebook of valid words in order to have a finite vector of word counts. Each feature descriptor is then said to be a word of the type closest to it, a process known as vector quantisation.

The codebook of size k is created by clustering the features for a database of images into k clusters that provide an adequate representation of the words available across the database, whilst not being so large as to cause similar features to be assigned to different clusters. Clustering is a difficult problem with no perfect solution [43]; however, a common method of performing this step is to use k-means clustering, a simple but effective iterative clustering method. Each cluster is initialised, usually randomly, with some of the data points. This initialisation may take the form of randomly assigning each point a cluster, choosing k data points as cluster centroids and assigning close points to that cluster, or other more complex methods. The algorithm then iteratively calculates the centroid of all members of each cluster before assigning each data point to a cluster, different if a new cluster centroid is now nearer than the current one. This continues until no more changes occur, for a number of iterations or until the change in centroids or error is reduced sufficiently. Function 2.1 is an implementation of k-means clustering that initialises clusters from randomly selected data points and continues until the clusters are stable.

Function 2.1: kMeansClustering

```
numPoints, featureLength ← Size(features);
// Initialise centroids to random data points
for i ← 0 to k do
    centroids[i]← features[rand(0,numPoints)];
end
hasChanged ← true;
while hasChanged do
    hasChanged ← false;
    // Find the nearest cluster centroid to each point and assign it to
       that cluster
    for j ← 0 to numPoints do
        newCluster ←nearestCluster(features[j], centroids);
        if newCluster ≠ cluster[j] then
            cluster[j]← minCluster;
            hasChanged ← true;
        end
    end
    // Calculate the new cluster centroids
    for i ← 0 to k do
        centroids[i]← calculateClusterCentroid(cluster,k,features);
    end
end
```

As well as the value of k, the quality of the clusters produced by k-means clustering depends heavily upon how the clusters are initialised, often becoming trapped in local minima. One method is to repeat the algorithm with many different random initialisations and keep the result which produces the most compact clusters. An alternative is to use k-means++ [6] to provide a better initialisation by ensuring that the initial cluster centres are spread evenly throughout the feature space, and prevent the algorithm becoming stuck in a local minima. The algorithm does this by adjusting the probability distribution used to select the data points. The first is chosen from a uniform distribution; then the distance from each point to this cluster centre is calculated. The next cluster centre is chosen from a distribution weighted by the square of this distance, causing data points that are further away to have greater chance of being selected. This is repeated, with the distribution now defined by each point's distance to the nearest, rather than the original, centre until k centres have been selected, and then the rest of the k-means algorithm proceeds as normal. This method improves the computational cost of k-means by reducing the need for multiple runs to avoid local minima, but also as the dispersed initial clusters improve the time taken for convergence to a much greater extent than the more complex initialisation increases cost.

The computational cost of k-means clustering is high, nearly all time being spent calculating the distances between centroids and data points, something that must be done k times for each data point at each iteration. A number of solutions that reduce the computational complexity have been suggested. Approximate k-means [69] uses a forest of k-d trees to provide a good approximation for distance calculations, reducing the time spent finding the nearest cluster. Hierarchical k-means [63] uses a low value of k but applies it hierarchically, clustering each larger cluster into another k clusters, for l levels until k^l clusters have been produced.

Once the words have been found through clustering and each image has been quantised into a histogram of these words, a classifier is used to determine the correct object categories that may be present in each image. One method that allows this to be done easily with large vocabularies and many categories is the naïve Bayes classifier (NBC), first used for categorisation by Csurka et al. [19]. The probabilistic classifier uses naïve independence assumptions about features to simplify the calculations based upon Bayes' theorem. These independence assumptions are naïve as they are often wrong, e.g. the visual words for eye and nose are likely to be highly dependent, but this does not prevent the classifier from being accurate. From Bayes' theorem, for a category C and set of features v the probability of the image being in that category given the features is

$$P(C|v_1,\ldots,v_n) = \frac{P(C)P(v_1,\ldots,v_n|C)}{P(v_1,\ldots,v_n)}. \tag{2.3}$$

We can safely ignore $P(v_1,\ldots,v_n)$, as it is constant across all categories, leading to

$$P(C|v_1,\ldots,v_n) \propto P(C)P(v_1,\ldots,v_n)|C). \tag{2.4}$$

With the naïve independence assumptions and conditional probility this leads to

$$P(C|v_1, \ldots, v_n) \propto P(C) \prod_{j=1}^{n} P(v_j|C). \tag{2.5}$$

The values of $P(C)$ and $P(v_j|C)$ can be calculated empirically from a training set of m images. $P(C)$ is the frequency of the class across all the training images. $P(v_j|C)$ can be represented by a normal probability distribution with mean and variance defined by the values of v_j for each class.

2.6 Symmetry

Symmetry is a valuable high level image feature present both in natural and man-made objects that can provide important information about object structure, but can be difficult and computationally expensive to determine. Three types of symmetry of particular interest are reflectional (bilateral), rotational and translational. Reflectional symmetry is concerned with the identification of axes of symmetry, whilst to detect rotational symmetry the centres of rotation must be found. Translational symmetry is a considerably easier problem, finding only regions that have similar translated regions elsewhere in the image. In addition to these three basic types there are others found from combination, such as radial (reflection and rotation) or glide-reflection (translation and reflection).

One well established approach to the detection of reflectional symmetry is Reisfeld et al.'s generalised symmetry transform (GST) [74]. The GST compares every pair of points in the image, and accumulates at their midpoint, weighted by a symmetry measure. This measure is the combination of three functions relating to distance, phase and edge strength. If i and j are indices to two image points $\mathbf{P}_i$ and $\mathbf{P}_j$, then the distance function D describes the scale at which symmetry is found, and is controlled by σ_G:

$$D(i, j, \sigma_G) = \frac{1}{\sqrt{2\pi\sigma_G}} e^{-\frac{|\mathbf{P}_i - \mathbf{P}_j|}{2\sigma_G}}. \tag{2.6}$$

Small values of σ_G search locally, whilst larger values search the entire image. The phase function P controls the contribution from the alignment of edge direction:

$$P(i, j) = (1 - \cos(\theta_i + \theta_j - 2\alpha_{ij})) \times (1 - cos(\theta_i - \theta_j)). \tag{2.7}$$

θ is the edge direction at those points and α_{ij} is the direction of a line joining the two points $\mathbf{P}_i$ and $\mathbf{P}_j$. The phase function produces the maximum response when the edge direction at the two points is opposite. The edge magnitude is used in logarithmic form,

$$E_L(i) = \log(1 + E(i)). \tag{2.8}$$

The symmetry, $C(i, j, \sigma_G)$, of points i and j is then

$$C(i, j, \sigma_G) = D(i, j, \sigma_G) \times P(i, j) \times E_L(i) \times E_L(j). \tag{2.9}$$

The symmetry at a single point μ is the sum of the symmetry of the set $\Gamma(\mathbf{P}_\mu)$ of all pairs of points whose midpoint is at μ. The symmetry, $S_{\mathbf{P}_\mu}(\sigma)$, at point μ is therefore

$$S_{\mathbf{P}_\mu}(\sigma_G) = \sum_{i,j \in \Gamma(\mathbf{P}_\mu)} C(i, j, \sigma_G). \tag{2.10}$$

Function 2.2: generalisedSymmetryTransform

```
edgeImage, edgeDirection ←edgeDetection(image);
logEdges ← log(1 + edgeImage);
for x1 ← 0 to width do
    for y1 ← 0 to height do
        for x2 ← 0 to width do
            for y2 ← 0 to height do
                mx ← (x1 + x2)/2;
                my ← (y1 + y2)/2;
                symval ← distance(x1,y1,x2,y2,sigma) ∗
                phase(x1,y1,x2,y2, edgeDirection ) ∗ logEdges[x1, y1] ∗ logEdges[x2, y2];
                symmetry[mx, my] ← symmetry[mx, my] + symval;
            end
        end
    end
end
```

Function 2.2 presents the GST in pseudocode. It is clear from the number of loops that a major disadvantage of the generalised symmetry operator is the high computational cost, as every pixel must be compared to every other and their symmetry calculated. With an image of width and height both equal to n, the GST has a complexity of $\mathcal{O}(n^4)$.

Many recent approaches to symmetry detection have focused on the analysis of invariant features (such as SIFT). Loy and Eklundh [54] matched features in the original image with similar, mirrored versions and updated an accumulator in a weighted fashion using a symmetry function based upon that of the GST. A similar approach was also taken for rotational symmetry, as any two features that have the same descriptor values but with different rotations suggest a centre of rotation. This centre can be found through accumulation across all matching pairs of points. Additionally, the order of rotational symmetry can be found through analysis of a histogram generated from the angle of rotation about these centres. Another approach by Cho and Lee [17] found pairs of symmetrical invariant features and through merging of close features grew reflectionally symmetrical regions.

There are a number of alternative approaches to rotational symmetry. Lee et al. [49] exploited frieze-expansions to transform the problem to one of translational symmetry. This makes the use of Fourier analysis to detect the order and type of symmetry easier. An alternative, but conceptually similar approach to detecting rotational (as well as reflectional) symmetry is to use a polar Fourier transform to formulate the problem as one that can be solved with signal processing [47]. Prasad and Davis [72] used features from a gradient vector flow field to help define a graph, and used this graph to enable the detection of centres and orders of rotational symmetry.

2.7 Conclusions

In this chapter we have presented the context for the work on the IRT. We have shown the many different analogies to physical phenomena that have been used to develop techniques for computer vision, including heat, water and force. The limited development of analogies based upon light waves has been highlighted and identified the opening this presents for a new technique based upon an analogy to light. A number of applications for the IRT have been identified and discussed. The detection of circles and the retinal vasculature have been described, and we have signified the importance of good enrolment to enable ear biometrics. We also identify other areas where an analogy to light rays may be useful for finding higher level features. Object categorisation with the bag-of-visual-words model is a difficult problem and a variant of the IRT may be able to provide a new and complementary approach. Finally, symmetry requires analysis of images on a higher level than simple structures does, and can be computationally expensive. An analogy to light rays can form a basis for new computer vision techniques that are able to tackle these problems in novel ways.

Chapter 3

The Image Ray Transform

This chapter describes the Image Ray Transform (IRT), a novel feature detection technique based upon an analogy to light rays. The transform analogises an image as a matrix of glass blocks in order to cast rays of light through it. It exploits total internal reflection to focus rays into certain areas and update an accumulator, highlighting structural features such as tubes and circles. Later we examine some applications of this structural feature detection (chapter 4) and show how the IRT can be enhanced to extract higher level features. (chapter 5).

This chapter is organised as follows. Section 3.1 describes the relevant laws of optics that are employed in order calculate the IRT and section 3.2 describes the mechanics of the IRT. Some enhancements to the IRT that widen the scope of the transform are investigated and detailed in section 3.3. A thorough discussion of parameter selection for the transform occurs in section 3.4 and section 3.5 describes an extension using rays tethered together as a beam for a transform with increased robustness to noise. Finally conclusions are drawn.

3.1 Laws of Optics

The IRT is based upon a subset of the laws of optics, employing aspects of the analogy that improve results whilst omitting or adapting those that lead to excessive computation or inferior results. We do not attempt to simulate light, but rather build upon the principles of optics to construct the IRT. Rays are a method of modelling the propagation of waves, most often light, with specific regard for the direction of the wave as it interacts with its environment and ignoring wave-like interactions such as diffraction. The material through which light travels is referred to as a medium and has a number of physical properties, of which only refractive index is relevant to the IRT. A ray propagates in a straight line through a medium (with refractive index n_1) until it reaches a

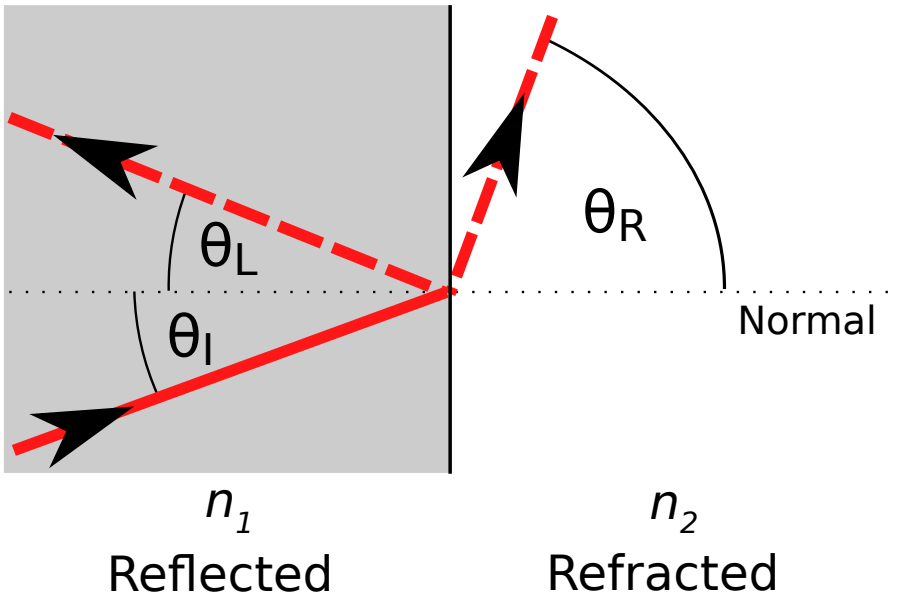


FIGURE 3.1: Refraction and reflection of light at a boundary of two media m_1 and m_2 with refractive indices n_1 and n_2 respectively.

boundary with another medium of differing refractive index (n_2), at which point it will change direction. Figure 3.1 shows an example of this, the solid line being the incoming ray incident on the boundary at an angle to the boundary normal of θ_I and the dashed lines being two possible outgoing rays, dependent on whether refraction or reflection occurs. The dashed ray on the left of the figure shows reflection, where the ray reflects from the media boundary at angle of θ_L, and $\theta_L = \theta_I$. On the right of the figure is a ray that has been refracted and transmitted into the second medium, its direction changing so that it travels at an angle of θ_R to the boundary normal. For simplicity, we do not consider cases where a light ray may split into a reflected and a refracted part. Refractive index is the ratio of the speed of light in a vacuum to the speed of light within a medium. The relationship between the refractive indices and the angles of the incident and refracted light is described by Snell's law:

$$\frac{\sin\theta_I}{\sin\theta_R} = \frac{n_2}{n_1}. \tag{3.1}$$

When $n_1 < n_2$ this implies that light will bend towards the boundary normal, i.e. $\theta_R < \theta_I$. When $n_1 > n_2$ it implies that it will bend away from the normal, i.e. $\theta_R > \theta_I$. Snell's law is unsuitable for use where it would result in $\theta_R > 90°$ as refraction into the original medium is impossible. The maximum value of θ_I before this occurs is the critical angle θ_C, and is given as

$$\theta_c = \sin^{-1}\left(\frac{n_2}{n_1}\right). \tag{3.2}$$

When $\theta_I > \theta_C$ the ray undergoes total internal reflection, and propagates in direction θ_L. These principles of refraction and total internal reflection form the basis of the Image Ray Transform.

3.2 Mechanics of the Image Ray Transform

The IRT operates by casting rays through an image, analogised as a transparent medium. The transformed image is generated by recording the course of each ray, resulting in enhanced structural features due to rays being focused within certain types of feature.

3.2.1 Analogising the Image

The first step in performing the IRT on an image is to consider it as a structure through which rays can travel. We analogise the image as a matrix of glass blocks, each block representing a pixel. The refractive index of these blocks is derived from properties of the pixel they represent. If we take a simple image of a white circle on a dark background (figure 3.2(a)) we want to analogise it as a glass cylinder surrounded by air (3.2(b)). If we consider the 10×10 version of this image, shown in 3.2(c), we convert that into a set of blocks as in 3.2(d), where blue pixels represent glass, and have an appropriate refractive index, whilst grey pixels represent air and have a lower index. We use a 3D example to illustrate the analogy more clearly, although the ray casting in the IRT occurs in a single 2D plane. A simple linear relationship between pixel intensity and refractive index can be used:

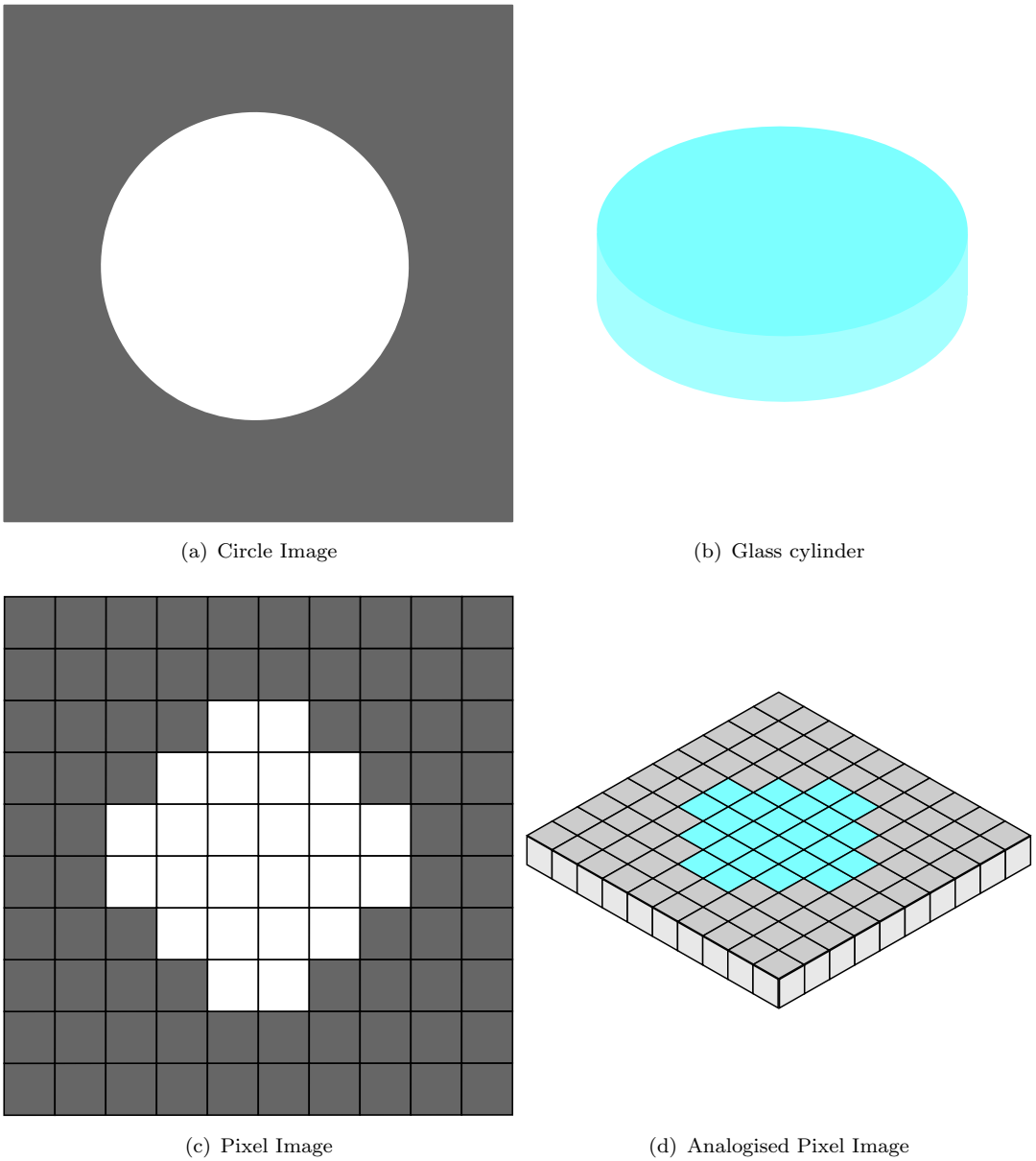
$$n_i = 1 + \left(\frac{i}{255}\right) \cdot (n_{\max} - 1). \tag{3.3}$$

In this case refractive index is scaled between 1 and a maximum refractive index parameter, $n_{\max}$, with pixel intensity between 0 and 255. Different methods of setting refractive indices are described in section 3.3.3. It is within these glass blocks that ray casting occurs. This is shown in `analogiseImage` (function 3.1), where every element

Function 3.1: analogiseImage

```
for x ← 0 to width do
    for y ← 0 to height do
        refractionMatrix[x, y] ← 1 + (image[x, y]/255.0) ∗ (nMax − 1);
    end
end
return refractionMatrix;
```

in the refraction index matrix is set according to equation 3.3 and the intensity of the corresponding image pixel.



(a) Circle Image

(b) Glass cylinder

(c) Pixel Image

(d) Analogised Pixel Image

FIGURE 3.2: The analogisation of an image to a set of blocks with differing refractive indices.

3.2.2 Ray Casting

Within the array of glass blocks that represent our image we cast a large number of randomly initialised rays. Figure 3.3 shows an example of the path of a ray as it travels through a simple image. Each row of pixels is the same intensity and therefore the same refractive index. The ray is initialised at a random position A with a random direction and first advances to position B. At B the refractive indices of the current and next pixel are compared (n_1 and n_2 respectively), and as $n_1 = n_2$ the ray continues with no change in direction. At C, $n_1 < n_2$, and so by Snell's law (equation 3.1) the ray will bend towards the normal. At D $n_1 > n_2$, and $\theta_I < \theta_C$ and so again by Snell's law the

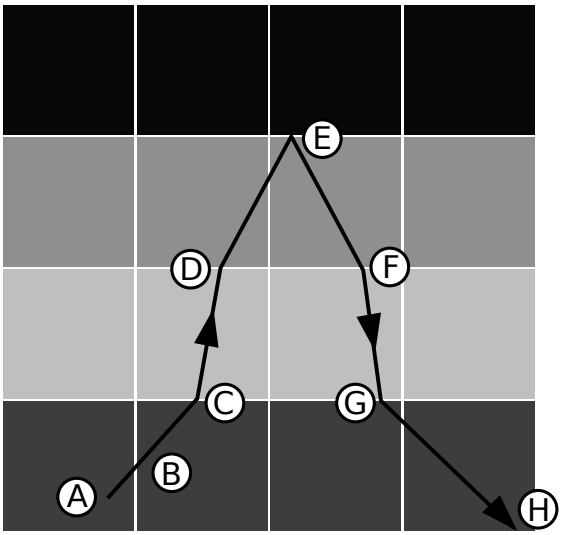


FIGURE 3.3: An example of the course a ray might take in a simple 4x4 image.

ray bends away from the normal. At E, because $n_1 > n_2$ and $\theta_i > \theta_c$, total internal reflection occurs, preventing the ray from entering the top row of pixels. At F and G the occurrences at C and D are repeated in reverse order whilst at H the ray exits the image and the casting of the ray terminates.

More formally, each ray is initialised at a random position (x, y) and with a random direction ϕ, drawn from a uniform distribution U. In an image of size $w \times h$ then

$$x \sim U[0, w),\ y \sim U[0, h),\ \phi \sim U[0, 2\pi). \tag{3.4}$$

We convert ϕ to a vector direction $\mathbf{V}$ for more convenient calculations, and initialise the ray at position vector $\mathbf{p}$

$$\mathbf{p}^{<0>} = (x, y)^T, \tag{3.5}$$

and generally, at time t:

$$\mathbf{p}^{<t>} = \mathbf{p}^{<t-1>} + \mathbf{V}. \tag{3.6}$$

When a ray enters a block for the first time, the accumulator matrix ($\mathbf{A}$) entry for that block is incremented by 1.

$$\mathbf{A}(\mathbf{p}^{<t>}) = \mathbf{A}(\mathbf{p}^{<t>}) + 1 \tag{3.7}$$

The accumulator is only incremented on the ray's first entry into a block in order to prevent small loops in the ray's path causing small clusters of pixels to be repeatedly incremented, which are more likely to be caused by noise than a desired structural feature.

When the ray crosses a block boundary where media on either side have different refractive indices, a vector formation (adapted from Hill [34]) of the laws described in

section 3.1 must be used to calculate a new direction. If $\mathbf{N}$ is the normal of the boundary and n_1 and n_2 are the refractive indices of the first and second media respectively then the angle of incidence θ_I is

$$\cos\theta_I = \mathbf{N} \cdot \mathbf{V}. \tag{3.8}$$

Calculation of the critical angle θ_C using equation 3.2 is necessary at this point if $n_1 > n_2$. If the ray refracts (that is, $n_1 < n_2$ or $n_1 > n_2$ and $\theta_I < \theta_C$) then for clarity we assign

$$n = \frac{n_1}{n_2}, \tag{3.9}$$

and the direction of the refracted ray $\mathbf{R}_R$ is

$$\mathbf{R}_R = n\mathbf{V} + (n(\mathbf{N} \cdot \mathbf{V}) - \cos\theta_R)\mathbf{N}, \tag{3.10}$$

where $\cos\theta_R$ is

$$\cos\theta_R = \sqrt{1 - n^2(1 - \mathbf{N} \cdot \mathbf{V})}. \tag{3.11}$$

In other cases the ray totally internally reflects. The direction of reflection $\mathbf{R}_L$ is then

$$\mathbf{R}_L = \mathbf{V} - 2(\mathbf{N} \cdot \mathbf{V})\mathbf{N}. \tag{3.12}$$

$\mathbf{R}_L$ or $\mathbf{R}_R$ are then assigned to the new ray direction$\mathbf{V}'$

$$\mathbf{V}' = \begin{cases} \mathbf{R}_L & \text{if } n_1 > n_2 \wedge \theta_I > \theta_C \\ \mathbf{R}_R & \text{otherwise .} \end{cases} \tag{3.13}$$

There are a number of termination criteria that are used to determine when we should cease to follow a ray. An obvious one is to stop following the ray when it exits an image, but there may be situations where rays continue to travel indefinitely without reaching the image border. In cases such as these we must impose an artificial limit, either on the number changes in direction or the total length of the ray. The depth limit terminates the ray after it has undergone d changes in direction (refractions or reflections), whilst the length limit, l, stops the ray after it has covered a specified distance. The process of casting a random ray is shown in function 3.2 as `castRandomRay`. The pseudo code shows initialisation and casting of the ray as well as accumulation on the first entry into pixels and the conditions under which casting ceases. This is repeated for N rays and the transformed image is found from the accumulator $\mathbf{A}$.

3.2.3 The Ray Analogy

The IRT is based on a light analogy, ignoring or simplifying many properties of light in order to improve the method. At media boundaries incident light will often split into

Function 3.2: castRandomRay

```
// Randomly initialise ray paramters
x ← rand(0, width), y ← rand(0, height), φ ← rand(0,2π);
// Convert angle direction to a vector
V_x ← cos φ, V_y ← sin φ;
depth, length ← 0;
// Initialise visited array to false for all pixels
visited[:]← false;
// Whilst we haven't exceed a limit or exited the image
while depth ≤ maxDepth ∧ length ≤ maxLength ∧ withinImage(x,y) do
    // Update accumulator, if we haven't already done so with this ray
    if ¬visited[x, y] then
        accumulator[x, y] ++;
        visited[x, y] ← true;
    end
    // Advance position to next block.
    isIndexChange, distanceMoved, x', y' ←advanceRay(x, y, V_x, V_y);
    if isIndexChange then
        n_1 ← refractionMatrix[x, y], n_2 ← refractionMatrix[x', y'];
        V_x, V_y ←calcNewDir(V_x, V_y, normals[x, y], n_1, n_2);
        depth ++;
    end
    x ← x', y ← y';
    length ← length + distanceMoved;
end
```

a refracted and a reflected part, dividing energy between them. This is not modelled for reasons of computational complexity: it would cause the exponential growth in the number of rays (up to a maximum of 2^d rays from a single initial ray) and require significant extra computation to model the properties of the ray and media needed to calculate the division of the ray, for which no improvement to result of the technique can be foreseen. Other wave-like behaviour, such as dispersion, is also ignored through the use of rays rather than waves to simulate light providing further savings to computational cost.

Whilst the method is strongly founded upon the analogy to light rays, there are changes made in order to improve the result of the transform whilst maintaining the advantages of the analogy. The normals ($\mathbf{N}$) that are used to calculate the new direction of the ray are not set to horizontal or vertical block boundaries ($\mathbf{B}_0$ and $\mathbf{B}_0$ in figure 3.4) as could be expected, but rather the normal of the tangential edge direction found by the Sobel operator ($\mathbf{E}$) at that point:

$$\mathbf{N} = \mathbf{E} + \frac{\pi}{2}. \tag{3.14}$$

These normals are more representative of the image, ensuring the result of the transform reflects the information within the original image to a greater extent than if the transform adhered strictly to the analogy.

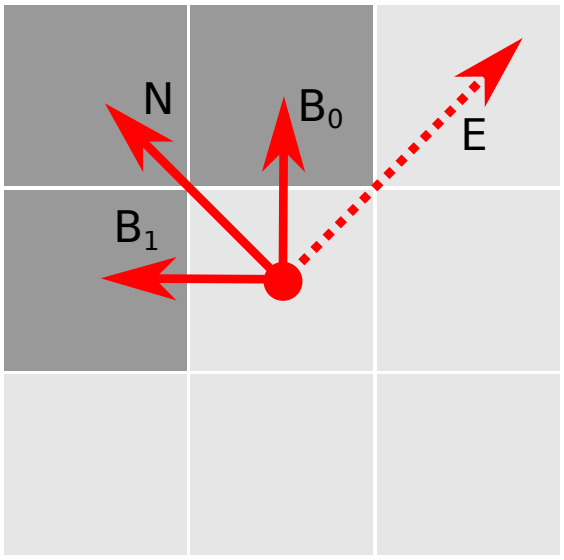


FIGURE 3.4: The normals that could be used to calculate reflections and refractions.

The detection of structural features by the transform can be improved by adapting some of the laws describing when refraction occurs. When a ray moves from a lower to higher refractive index it is allowed to continue on its original course rather than to bend towards the normal. Over the course of many refractions and reflections this leads to rays travelling along structural features rather than across them with greater ease and frequency, improving detection. This can be achieved by replacing equation 3.13 with

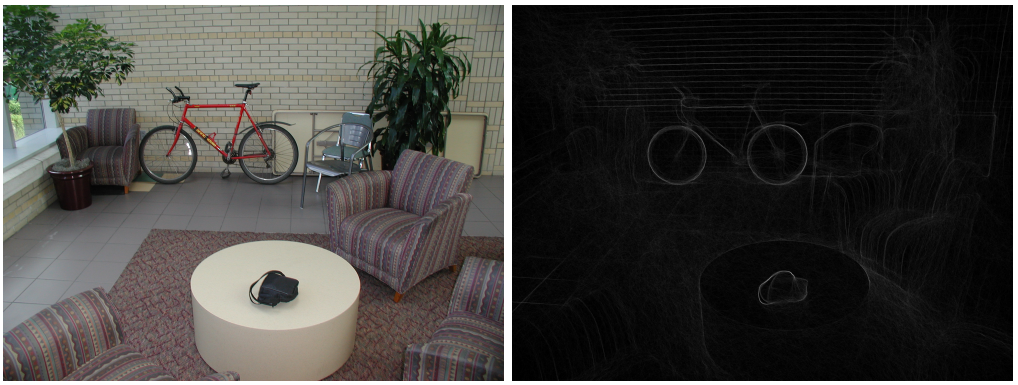
$$\mathbf{V}' = \begin{cases} \mathbf{R}_L & \text{if } n_1 > n_2 \wedge \theta_I > \theta_C \\ \mathbf{R}_R & \text{if } n_1 > n_2 \wedge \theta_I < \theta_C \\ \mathbf{V} & \text{otherwise.} \end{cases} \tag{3.15}$$

An example of the effect of varying when refraction and reflection occurs will be shown in section 3.2.4.

Function 3.3: imageRayTransform

```
normals ← sobelDirection(image) + π/2;
refractionMatrix ← analogiseImage(image, nMax);
accumulator[:] ← 0;
for t ← 0 to N do
    castRandomRay(refractionMatrix, accumulator, normals, width, height, maxDepth,
    maxLength);
end
transformedImage ← normalise(accumulator);
return transformedImage
```

The main structure of the IRT can be seen in function 3.3. After calculating the normals by equation 3.14 and the refractive index matrix with `analogiseImage`, N random rays are cast before the transformed image is found from the normalised accumulator.



(a) Original (b) IRT Result

FIGURE 3.5: An image (from WORD) processed by the IRT

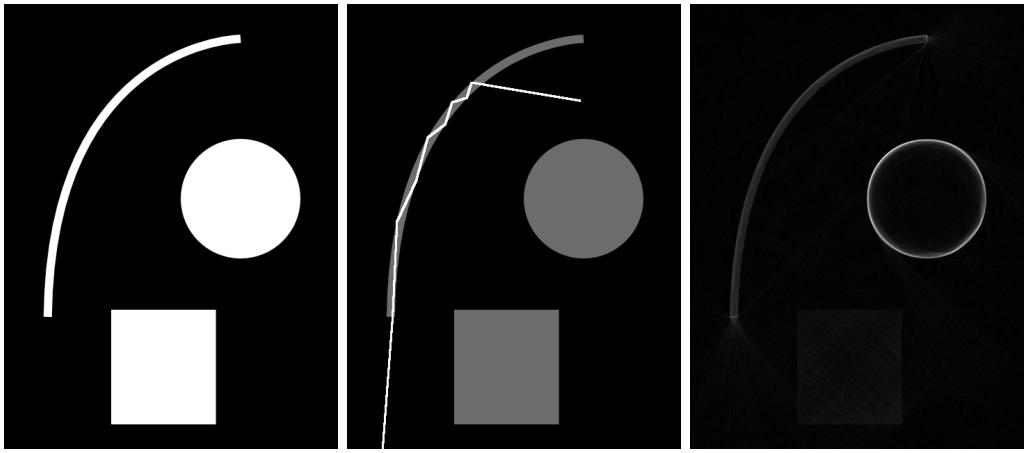
3.2.4 Transform Examples

Figure 3.5 shows an image from the Wiry Object Recognition Database (WORD) [13], and the result of the IRT on the inverted image (section 3.3.2 will discuss the reasons for inversion). The features that are strongly emphasised in this image are ones that the IRT is adept at highlighting. Tubular structures, long, thin areas of mostly constant width and intensity such as the horizontal mortar on the wall, the bicycle frame and the bag handle. Circular structures are also accentuated by the IRT, the bicycle wheels being a prominent example.

The transform's ability to detect tubular structures is due to the use of the light analogy and figure 3.6 illustrates how this occurs. Areas of larger refractive index (the white areas in figure 3.6(a)) can be analogised as glass, whilst smaller refractive indices (the dark background) are air. Rays that enter tubular structures tend to be directed along the the length of the structure, totally internally reflecting from the edges due to the difference in refractive indices. This is analogous to a waveguide, such as an optical fibre, directing light along its length. In figure 3.6(c) the tube is highlighted to a greater extent than the box, and on examination a spread of rays exiting at either end of the tube can be observed.

Figure 3.7 shows a number of images from the Caltech 101 dataset [31] in which the ray transform has been used to highlight tubular structures. Across a variety of images and both natural and synthetic objects the transform is able to strongly highlight tubular structures. Application and evaluation of the IRT with tubular structures is covered in more detail in sections 4.2 and 4.3.

In figure 3.6 the most prominent structure in the transformed image is the circle. The IRT is able to highlight circular structures through a similar mechanism to tubular features. Figure 3.8 displays a synthetic circle image through the course of the transform. The single ray in figure 3.8(b) has begun within the circle but due to the difference in



(a) Original (b) Single Ray (c) IRT Result

FIGURE 3.6: Artificial image demonstrating the ability of the IRT to extract structural features.

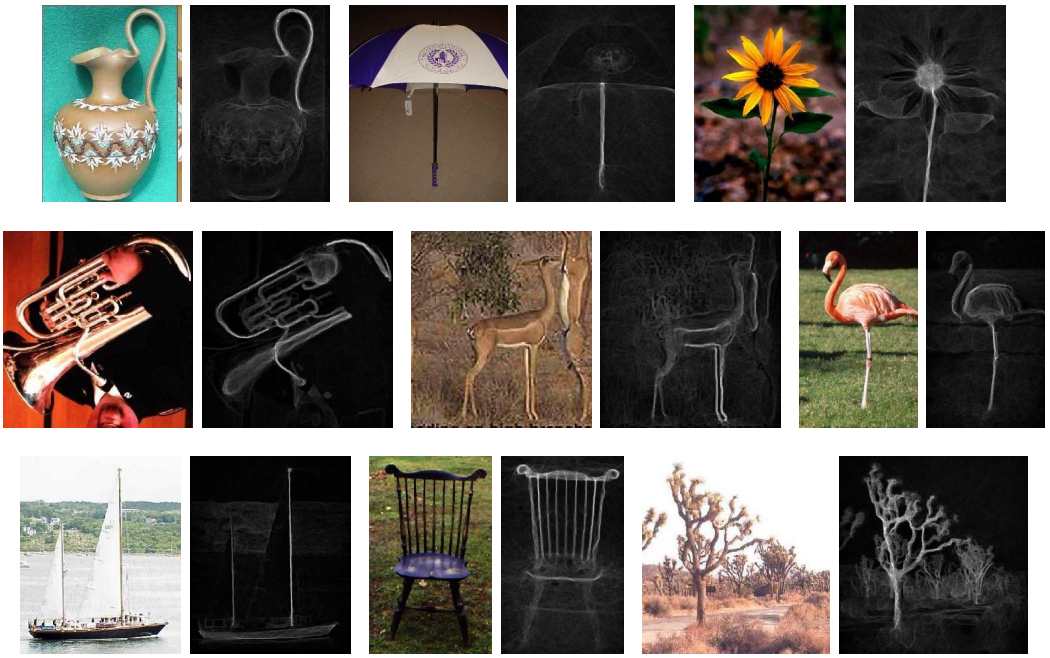
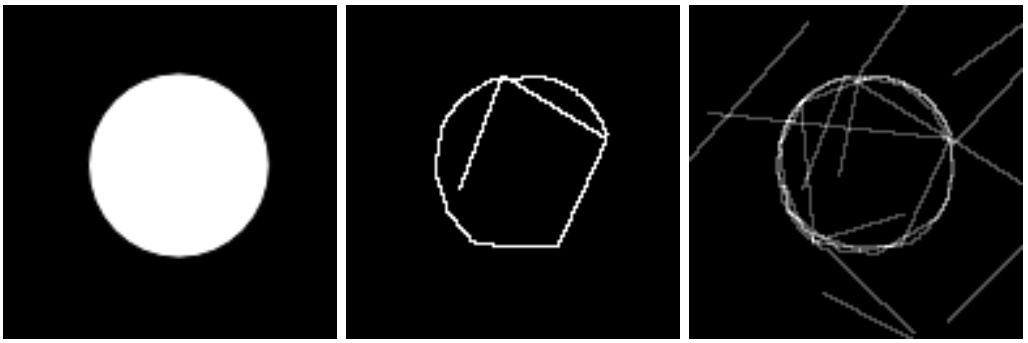
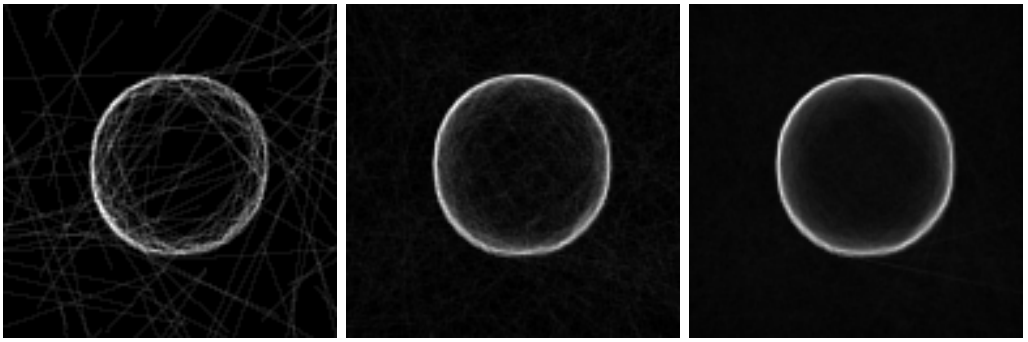


FIGURE 3.7: A range of images containing tubular features transformed with the IRT.



(a) Original (b) $N = 1$ (c) $N = 10$

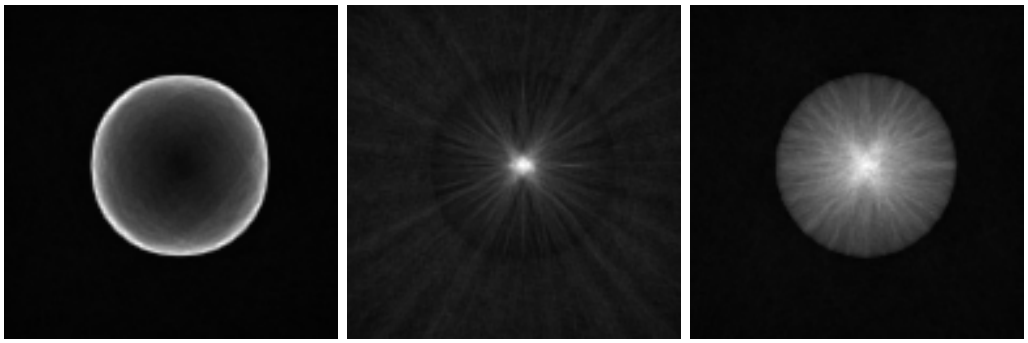


(d) $N = 100$ (e) $N = 1000$ (f) $N = 10000$

FIGURE 3.8: Accumulator throughout ray transform on simple circle.

refractive index between the circle and background, has totally internally reflected at the boundaries of structure. This occurs as it travels around the edge of the circle, never touching the border at less than the critical angle. Over thousands of rays this produces a transformed image with a strong border and reducing gradient toward the centre.

The different methods of calculating how changes in direction are calculated at media boundaries (equation 3.15) can be illustrated by the result on a circle, in figure 3.9. A simple circle (from figure 3.8(b) is transformed with the IRT three times, with rays that change direction when moving from larger to smaller indices, smaller to larger indices or at any change in index. Only changing direction when moving from larger to smaller indices (figure 3.9(a)), as used throughout this work, ensures that rays tend to travel along the direction of structures, in this case with the tangent of the circle. Alternatively, only changing when moving from smaller to larger indices (figure 3.9(b)) causes rays to tend to travel towards the normals, in this occurrence leading towards the centre of the circle. The result in this case appears to be very similar to the contents of a circle Hough transform (HT) accumulator, and suggests at first glance that it may be useful for such an application. However, this result is only produced when the circle is of uniform intensity (as occurs here), variation would change the direction of the rays away from being towards the centre in all but the most trivial of cases. If we always change direction at changes in media (figure 3.9(c)) then we get a combination of the two results, where rays are trapped within the circle but tend to point towards the centre.



(a) $n_1 < n_2$ (b) $n_1 > n_2$ (c) Always

FIGURE 3.9: Results on a circle image when varying the cases in which a rays direction is changed at a media boundary.

FIGURE 3.10: A range of images containing tubular features transformed with the IRT.

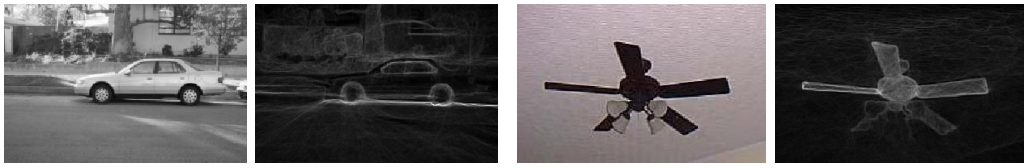


FIGURE 3.11: Example images showing some of the limitations of the IRT.

Two examples of the detection of circular features in natural images are shown in figure 3.10. Both the top of the barrel and centre of the sunflower are emphasised around their borders. Emprical testing of the IRT's effects on circle detection is detailed in section 4.1.

Some of the limitations of the IRT's application to structural feature detection are shown in figure 3.11. The difficulty of directing the IRT to emphasise only tubular or circular structures is shown in the image of a car. The IRT might be expected to highlight the car wheels most strongly as they are a strong circular feature, but the shadow of the car produces a stronger tubular structure, directing rays away from the wheels. The results of the transform can also be affected by other, undesired structures in the image. In the second image of the ceiling fans we aim to highlight the fan blades equally, but the occlusion of two of the blades reduces the response across the whole of those blades, rather than just the occluded area. The transform is adept at emphasising structural

features such as tubes and circles, but its efficacy can be reduced by complex occlusions and it is not easily directable.

3.3 Enhancements to the Image Ray Transform

The IRT as described in section 3.2 is a useful technique for processing images. However, through some simple enhancements the transform can be strengthened and the range of situations in which it is appropriate can be widened. The addition of a stopping condition reduces the number of parameters that must be set, and alternative methods of relating intensity and refractive index can make the transform suitable to more images as well as producing significantly different results.

3.3.1 Stopping Conditions

The optimal number of rays that should be cast in order to ensure a transform result is of sufficient quality is a challenge that can be met in a number of ways. Enough rays must be traced to sufficiently cover the image and reduce noise, but every ray has a computational cost that must be minimised. A heuristic used throughout the development of the transform was to set $10000 \leq N \leq 20000$, as the result tends to converge in most images between those values. An improved method for deciding when the transform should cease is to monitor the resultant image and stop when it no longer changes significantly between iterations. With a sufficiently large number of rays the magnitude of the values in the accumulator will be such that, following normalisation to an integer between 0 and 255, no change is observed when the path of a new ray is accumulated. For this to occur an extremely large number of rays are required and so we seek a method that provides a metric for this decreasing variation in order to measure when the transform should be stopped, whilst providing minimal extra computational cost to the technique. There are many ways to do this, two of which were investigated.

The Shannon entropy [75] of a variable is a measure of its uncertainty. By treating every pixel in the result image as a variable and summing the entropy of all pixels an upper bound on the entropy across the image can be found. In order to simplify the calculations, we assume each pixel i is a binary variable and its value at iteration t is calculated by

$$X^{<t>}(i) = \begin{cases} 0 & \text{if } \mathbf{I}'^{<t>}(i) = \mathbf{I}'^{<t-1>}(i) \\ 1 & \text{otherwise .} \end{cases} \tag{3.16}$$

that is, 1 if the intensity in the normalised accumulator output image ($\mathbf{I}'$) has changed since the last iteration and 0 otherwise. The Shannon entropy of a pixel is found using

the entropy function, summed across all possible values of the variable

$$H(X(i)) = -\sum_{a=1}^{n} P(x_a) \log_b P(x_a), \tag{3.17}$$

where $P(x_a)$ is the probability of variable X having value x_a and which in our binary case can be expressed as

$$H(X(i)) = H(p) = p \log p + (1-p) \log(1-p), \tag{3.18}$$

where p is the probability of $x = 1$, and hence $1 - p$ is the probability of $x = 0$. As we require an entropy measure that changes as the transform converges, these probabilities must be found by examining only the recent history of the variable, rather than its entire history:

$$p_i^{<t>} = \frac{1}{T} \sum_{z=(t-T)}^{t-1} x^{<z>}(i), \tag{3.19}$$

where T is the number of previous values of the variable to be looked at. Finally, by summing the entropies of every pixel we can find an upper bound for the entropy across the image at iteration t

$$H(\mathbf{I}'^{<t>}) = \sum_{i \in \mathbf{I}'} H\left(X^{<t>}(i)\right). \tag{3.20}$$

The value of the entropy measure throughout a single application of the image ray transform can be seen in figure 3.12 and it is clear that as the transform progresses the measure of entropy reduces, reflecting the reducing variation in the result between iterations.

Whilst these results are interesting, due to the simplification to binary variables they do not accurately represent the scale of the change between iterations. More importantly, calculation of the entropy causes a prohibitive increase in the computational and memory costs of the transform as the accumulator must be normalised following every ray and the history of each variable must be stored.

A simpler alternative to entropy is to measure the difference in the normalised accumulator image between iterations. We use the root mean squared (RMS) difference between the intensities, given by:

$$D^{<t>}(\mathbf{I}'^{<t>}, T) = \sqrt{\frac{1}{|\mathbf{I}'|} \sum_{z \in \mathbf{I}'} \left(\mathbf{I}'^{<t>}(z) - \mathbf{I}'^{<t-T>}(z)\right)^2}, \tag{3.21}$$

where T is the number of iterations between each comparison. The results produced by this method (figure 3.13) are more consistent with the observed resultant image, with little change occuring after 10000 iterations.

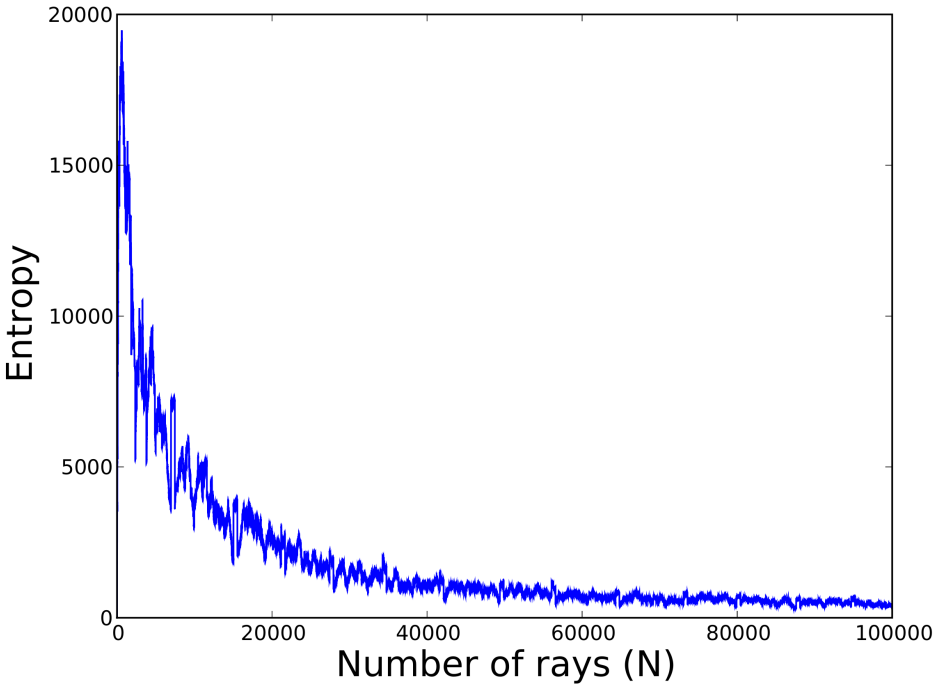


FIGURE 3.12: Entropy throughout an execution of the IRT. Entropy history length $T = 500$.

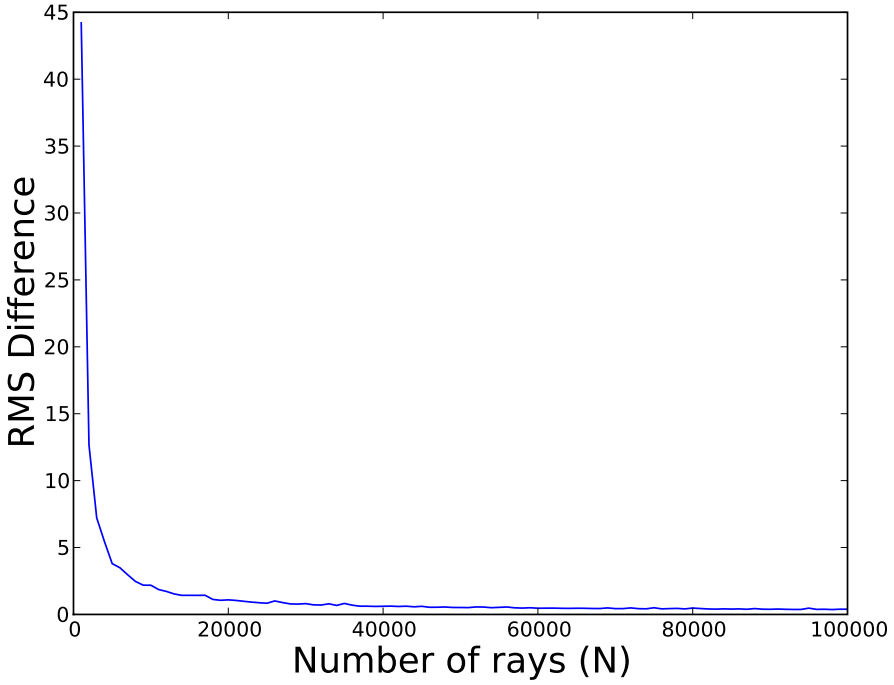


FIGURE 3.13: RMS difference measure throughout a ray transform. Iterations between comparisons $T = 1000$.

The RMS difference measure requires little calculation when T is set to a sufficiently large value, as expensive normalisations are only calculated infrequently. It also has the advantage of an intuitive meaning; in the case of figure 3.13, it is the change in intensity per pixel per 1000 rays. It can be used as a stopping condition by setting a lower limit (D_S) that, once exceeded, will cause the transform to stop. Function 3.4 shows

Function 3.4: checkStoppingCondition

```
if i % T = 0 then
    previousNormImage ← currentNormImage;
    currentNormImage ← normalise(accumulator);
    D ← 0;
    for x ← 0 to width do
        for y ← 0 to height do
            D ← D + (currentNormImage[x, y] − previousNormImage[x, y])^2
        end
    end
    D ←SquareRoot(D/(width ∗ height));
    return D < D_S
else
    return False
end
```

`checkStoppingCondition`, an example of the implementation of the RMS stopping condition. Such a function would be called after every ray had been cast, calculating the RMS difference every T iterations and returning true if it was found to be less than D_S. Discussion of optimal values for D_S occurs in section 3.4.

3.3.2 Target Intensities

The IRT, as described in section 3.2, will only emphasise features that are of a higher intensity than their surrounding area. This occurs because total internal reflection only works to prevent rays moving from higher to lower refractive indices (hence intensities) and not vice versa. Different measures can be taken to handle this, depending upon the desired result.

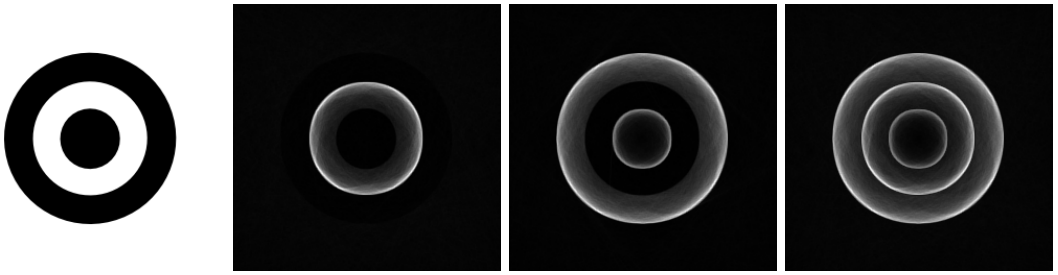
When the approximate intensity of the desired structures is known, a simple transformation can be applied to make that intensity (the target intensity) the largest value. We do this by finding the difference between the target intensity τ and the original intensity i_o for all pixels in the image:

$$i_\tau = 255 - |i_o - \tau|. \tag{3.22}$$

Further to this, if the intensity of the structures is not known, or varies, the IRT can be performed multiple times with different target intensities, and the results combined by selecting the maximal value at each pixel as shown in function 3.5 A specific and useful case is the selection of values such that $\tau = \{0, 255\}$ (or the original and inverted

Function 3.5: aggregateTransforms

```
for x ← 0 to width do
    for y ← 0 to height do
        transformedImage[x, y] ← max(transformedImage0[x, y], transformedImage1[x, y])
    end
end
```



(a) Original (b) $\tau = 255$ (c) $\tau = 0$ (d) $\tau = \{0, 255\}$

FIGURE 3.14: Transforms of a synthetic image demonstrating the advantages of different target intensities and aggregation.

image) allowing structures that are both light surrounded by dark, and vice versa, to be highlighted.

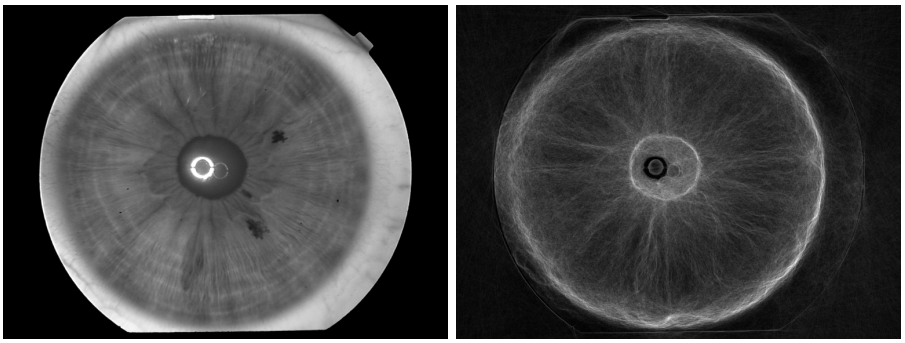
Figure 3.14 provides an example of how the addition of target intensities can improve the range of results possible with the ray transform. When $\tau = 255$ in figure (b), only the medium white circle is highlighted. Changing τ to 0 as in figure (c) allows the large and small black circles to be emphasised instead. If we wish to highlight all three circles in the transformed image, we can use aggregation of the two results as in figure (d). Combinations of transforms with different values of τ are used for enhancing circle detection in section 4.1.

3.3.3 Alternative Models for Refractive Indices

The linear model described in equation 3.3 is only one way of linking the properties of the analogised blocks and the original image. In some cases the differences in intensity between a feature and the surrounding area are not significant enough to extract them with the linear refractive indices calculated by equation 3.3. In such a case, an alternative version can be used (equation 3.23) that assigns refractive indices exponentially, to ensure greater difference, and more refraction and reflection:

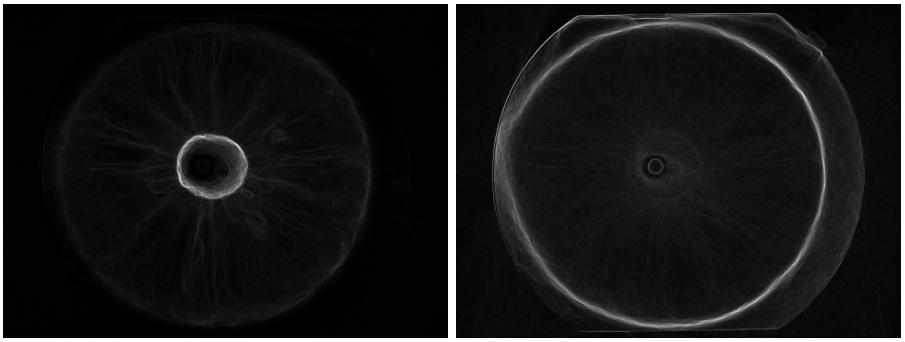
$$n_i = e^{\frac{i}{k}}. \tag{3.23}$$

In this case it is k rather than $n_{\max}$ that controls the scale of the refractive indices. The results of a number of transforms on an image of an iris are shown in figure 3.15. By varying the values of τ, $n_{\max}$ and k used, different circles present in the eye can



(a) Original

(b) $\tau = 40, n_{\max} = 40$



(c) $\tau = 0, k = 50$

(d) $\tau = 128, n_{\max} = 2$

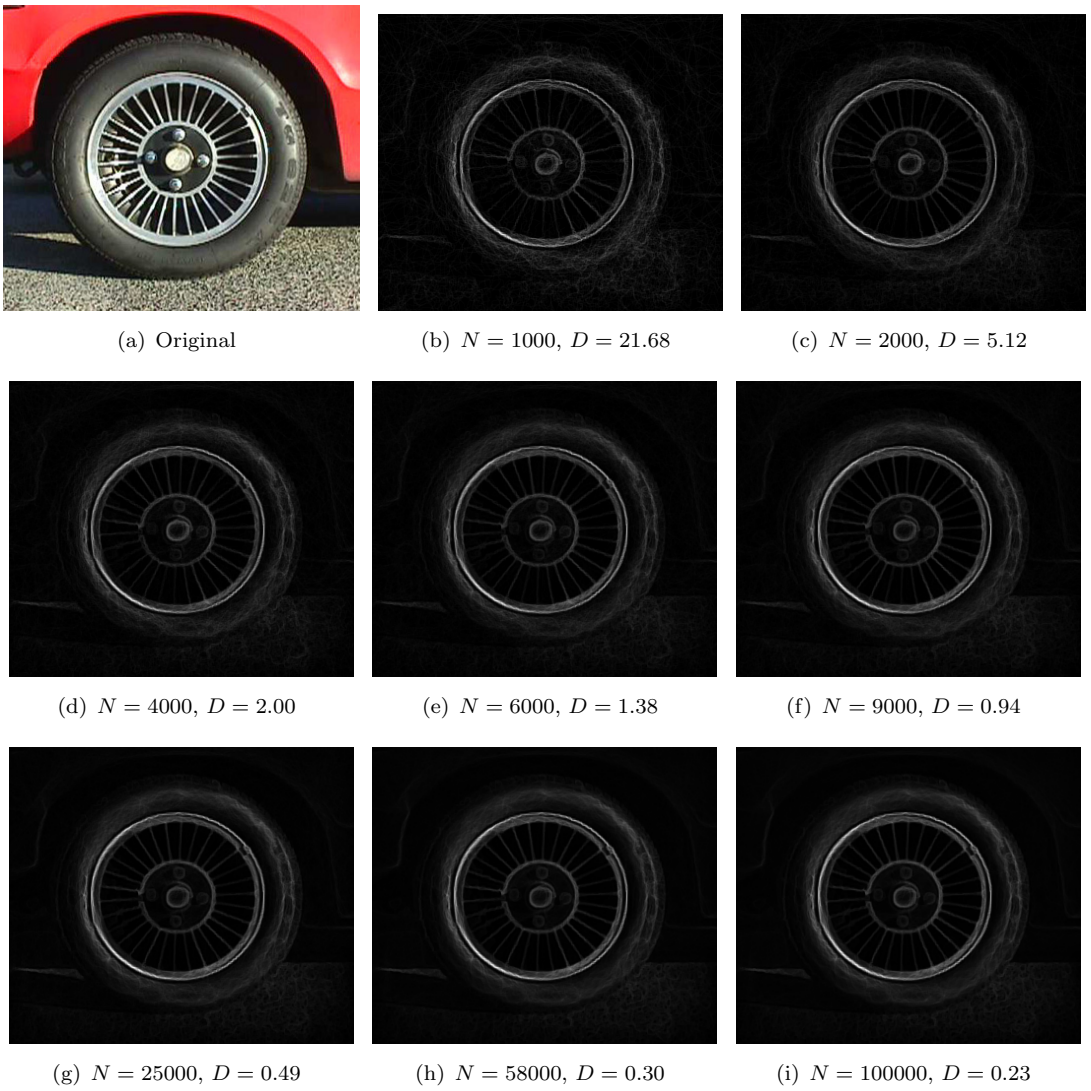
FIGURE 3.15: Ray transform of an iris image, with differing parameters extracting different areas.

be highlighted. In figure (b), the two strongest circles (around the pupil and the iris) are extracted, whilst figures (c) and (d) show each circle being extracted individually. In section 4.3 the complementary nature of the different models of refractive index are exploited for the extraction of retinal vasculature.

The refractive index does not necessarily need to be derived from the intensity of the pixel it represents. It can be advantageous to instead use the edge magnitude at each pixel to decide the value (found by the Sobel operator). This provides many structural features for the transform to follow, and is used in section 6.2 to calculate image symmetry.

3.4 Parameter Selection

As with most techniques, parameter selection is an important step in ensuring high quality results. Whilst the ray transform has a number of parameters, it has been designed so that many of them are derived from image information, or that static values found from experimentation are appropriate. Parameter selection for the IRT is primarily concerned with balancing noise and the speed of the transform. As the transform is non-deterministic due to the random initialisation of rays, a sufficient distribution of rays to cover pertinent features and limit noise is necessary. Conversely, every step of



(a) Original (b) $N = 1000$, $D = 21.68$ (c) $N = 2000$, $D = 5.12$

(d) $N = 4000$, $D = 2.00$ (e) $N = 6000$, $D = 1.38$ (f) $N = 9000$, $D = 0.94$

(g) $N = 25000$, $D = 0.49$ (h) $N = 58000$, $D = 0.30$ (i) $N = 100000$, $D = 0.23$

FIGURE 3.16: Progress of the ray transform through 100,000 iterations. $n_{\max} = 40, d = 256, \tau = 0$.

following a ray has a computational cost, which must be minimised at the same time. This section explores the parameters of the IRT and provides quantitative and qualitative analysis of their effects on the speed of the transform and the properties of the transformed image.

The parameter that has most effect on both the speed and quality of the transform is N, the number of rays. Strongly linked to this is the stopping condition parameter (D_S) described in section 3.3.1 that stops the transform when the accumulator is no longer changing significantly between iterations. Figure 3.16 shows an example of the result of the IRT on an image through the course of the transform, as N increases and D decreases. All uses of D_S also set $T = 1000$, as this prevents excessive numbers of computationally expensive accumulator normalisations. With a small value of N on an image of this size (350×301 pixels) and complexity the transform produces a noisy

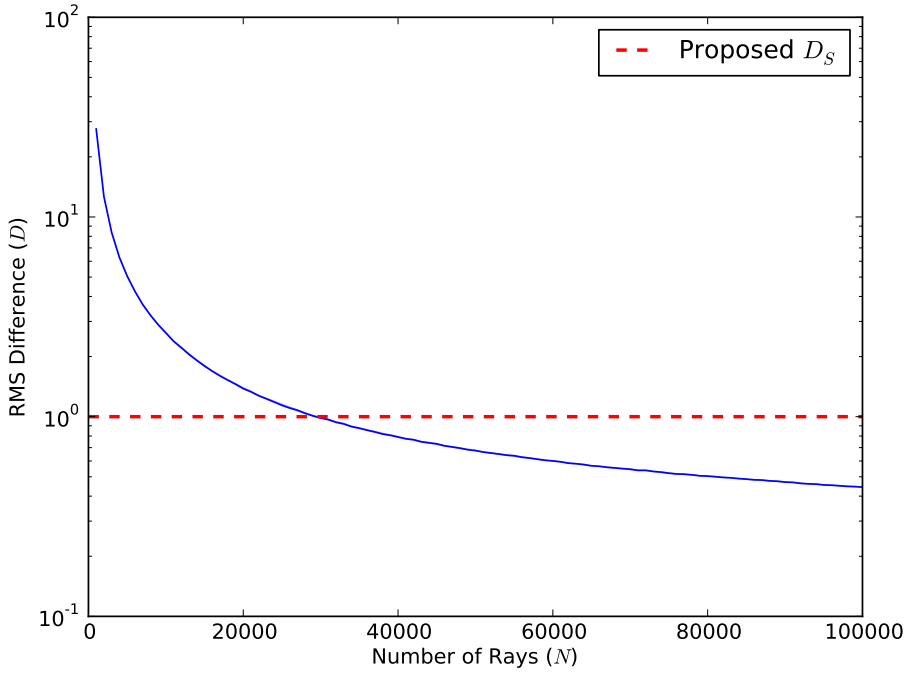


FIGURE 3.17: The variation of D as N increases, from 1000 images transformed with the IRT.

result, with individual rays still visible. The value of D is much greater than 1 as the contents of the normalised accumulator change significantly between calculations of the RMS difference. As N increases the result has significantly less noise and undergoes very little change even as N increases to 100000, this being reflected in the reduced value of D.

To determine an appropriate value of D_S we experimentally observed the behaviour of the RMS difference D as N increased. This was done across 1000 randomly selected images of differing scales from the VOC2008 dataset [30]. Figure 3.17 shows the mean values of D as N increased throughout these transforms, as well as a proposed value of D_S. Setting $D_S = 1$ is an appropriate value as it balances the reduction of noise with the reduction of unnecessary rays, the transform taking twice as long to reach $D = 0.5$ as $D = 1$. Image size is also a factor in the number of rays needed to sufficiently explore the image. The relationship between scale and number of rays was tested through experiments upon the 1000 images previously used, artificially scaling them to provide a wider range of sizes. Figure 3.18 shows how the number of rays needed to reach our stopping condition (i.e. when $D = 1$) increases as the number of pixels within the image increases. Whilst there is a large degree of variance in the number of rays necessary (due to the number of rays also being dependent upon the image features themselves) the linear regression shown suggests that there may be a relationship between the size of the image and the number of rays necessary to perform the IRT with sufficient accuracy.

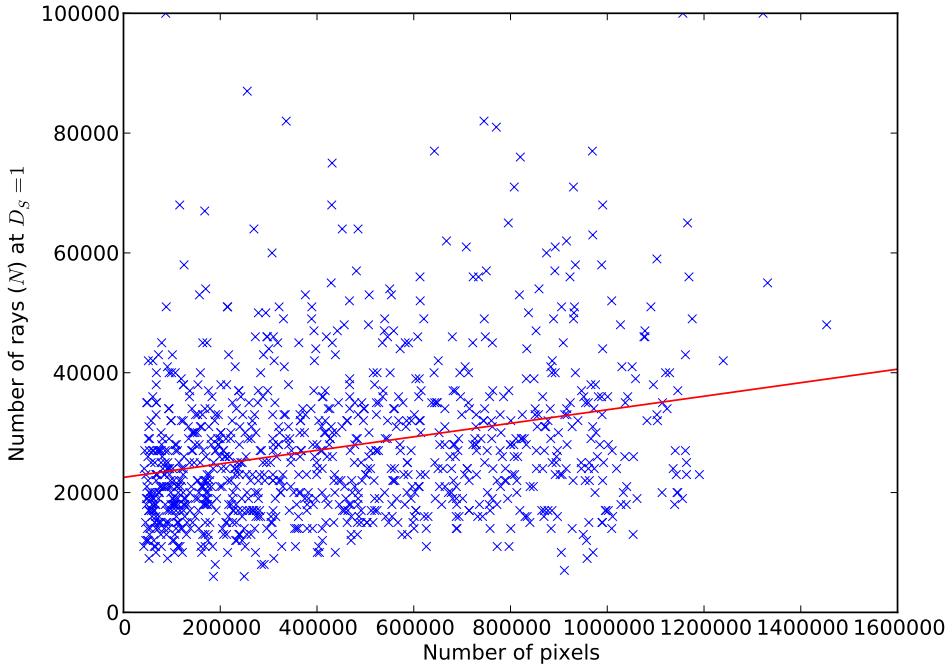
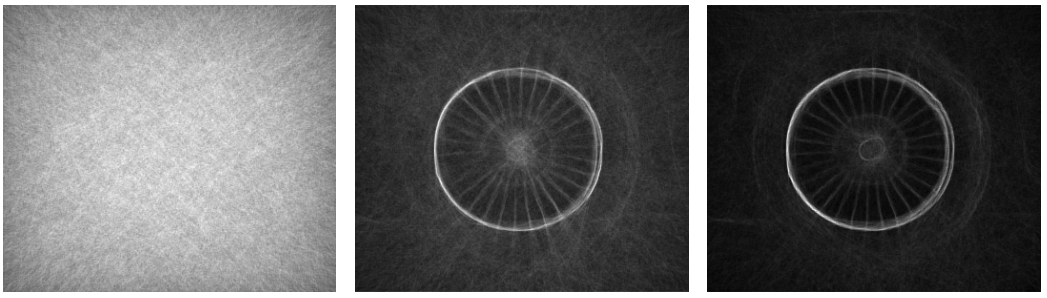
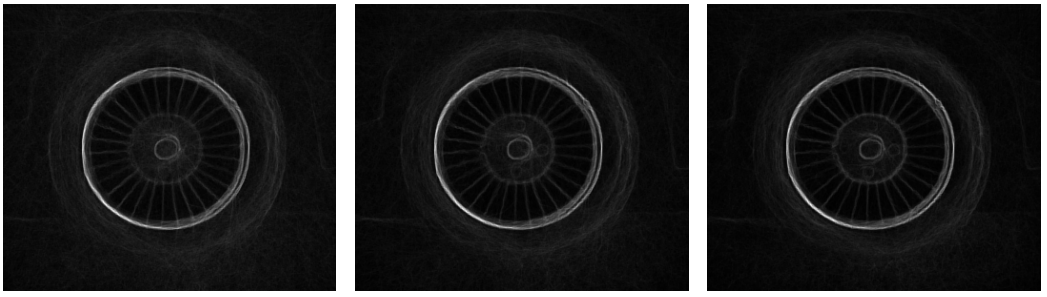


FIGURE 3.18: Number of rays need to reach stopping condition on images of different sizes.

The parameter that controls refractive index ($n_{\max}$ or k) has an effect on the the features that are highlighted most strongly by the transform. Figure 3.19 shows the image in figure 3.16(a) transformed with increasing values of the linear maximum refractive index, $n_{\max}$. When $n_{\max} = 1$ (figure 3.19(a)), all blocks have the same refractive index and there is no interaction between the image and the rays, leading to a random distribution of rays. This figure does show that the areas around the edge of the image receive fewer rays than the central area. When $n_{\max} = 1.1$ (figure 3.19(b))we find that only the circular metal rim and tubular spokes are well highlighted. Additionally it can be seen that with the weak difference in refractive index the rays travelling along the tubular spokes tend to continue past the rim into the wheel. As the value of $n_{\max}$ increases between 1.1 and 2 the previously highlighted features are emphasised more strongly and new features such as the hub's central circle and nuts emerge. Further increase in $n_{\max} > 5$ has minimal impact on the output of the transform. These properties of $n_{\max}$ are also demonstrated in figure 3.20, which shows both the mean time and number of rays for the IRT to compete across 100 VOC2008 images (the error areas shows the standard error of the mean). With low values of $n_{\max}$, the refractive index provides minimal guidance for the rays, causing many of them to not interact with image structures to a great degree and requiring more rays to be cast in order to stabilise the accumulator. On the other hand when $n_{\max} > 10$ there is effectively no change in the time taken to complete the transform, reflecting the minimal change in the qualitative examples shown in figure 3.19. The value of $n_{\max}$ that is appropriate depends partly upon the application; small values



(a) $n_{\max} = 1$ (b) $n_{\max} = 1.1$ (c) $n_{\max} = 1.25$



(d) $n_{\max} = 1.5$ (e) $n_{\max} = 1.75$ (f) $n_{\max} = 2$

(g) $n_{\max} = 5$ (h) $n_{\max} = 40$ (i) $n_{\max} = 80$

FIGURE 3.19: Results of ray transform for a range of values of $n_{\max}$. $d = 256$, $D_S = 1$.

pick out only the strongest features at an increased computational cost and increased noise, whilst larger values quickly find many features with minimal noise. Values above 10 provide performance that is identical both quantitatively and qualitatively, and for most experiments such a value (most often 40) was used.

The value of k, the exponential refractive index parameter, has a very different effect upon the transform when it is modified. Figure 3.21 presents some results of the transform for increasing values of k. The exponential assignment of refractive indices to intensity produces very large differences in refractive index and can lead to unusual results. Figure 3.21 uses an image of a face from the XM2VTS database to demonstrate this more clearly than the image of the wheel. For values of $k < 1$, rays are confined to odd paths, often representing noise to a greater extent than image features. When $1 < k < 20$ the IRT highlights some distinct areas that entrap rays such as the forehead, cheeks and ears. Further increase reduces the effect of the exponential operator until

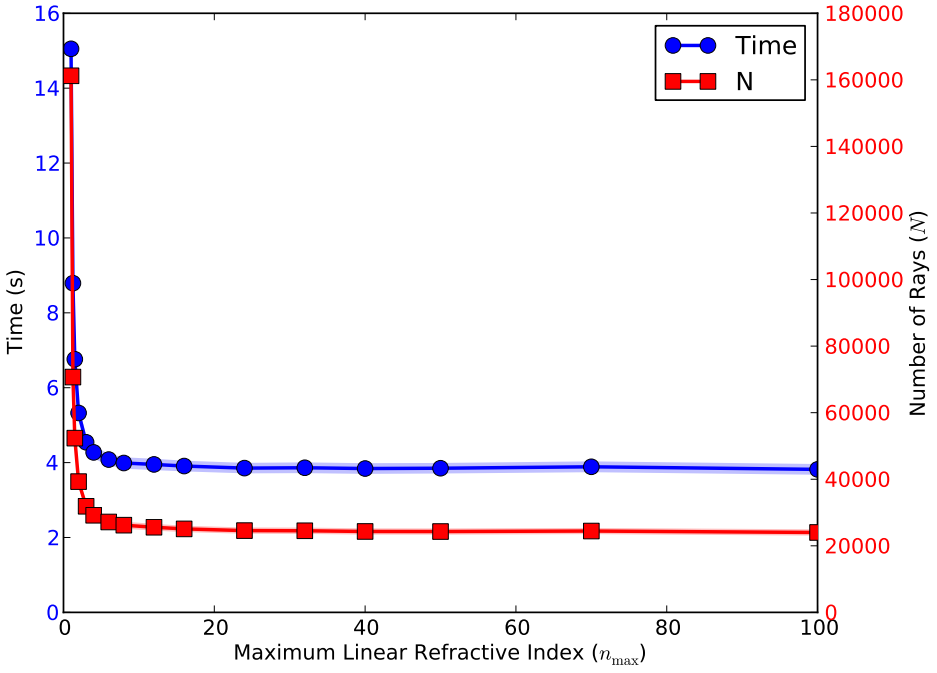


FIGURE 3.20: Mean number of rays need to reach stopping condition and time taken across 100 images with varying values of $n_{\max}$. Shaded areas shows the standard error of the mean.

the result is almost identical to results with large values of $n_{\max}$. The effect of k on the time of the transform is shown in figure 3.22, again on 100 VOC2008 images with the standard error shown. The behaviour shown here is best explained in reference to the behaviour of $n_{\max}$, as when the qualitative result of k approaches that of $n_{\max}$, when both values are large, the transform takes a similar length of time to complete. Smaller values of k cause rays to be constrained to features to a greater degree, and so lead to the accumulator converging with fewer rays. With very small values of k, where the transform does not extract any features of use, it takes more rays to converge as the result is affected to a great extent by noise.

The maximum number of changes in direction through refraction or reflection before termination of the ray (the maximum depth, d) is the primary parameter for limiting the time spent casting a ray. Figure 3.23 shows how different values of d change the result of the transform. Small values of d tend to terminate rays before they find and begin to follow a feature, or reduce the amount of time that a feature can be followed, leading to few areas being highlighted strongly and excessive noise. As d increases, rays can follow features for a greater number of direction changes, and more fully explore the image. There is no noticeable difference between the results of the transform with values of $d > 256$. Different values of d also emphasise different scales of features; when $d = 16$ the wheel's spokes are strongly highlighted, but when d is large the circular metal rim becomes the focus of the emphasis of the transform (as rays that enter the spokes now

(a) Original (b) $k = 0.1$ (c) $k = 0.5$

(d) $k = 0.7$ (e) $k = 1$ (f) $k = 10$

(g) $k = 20$ (h) $k = 50$ (i) $k = 100$

FIGURE 3.21: Results of ray transform for range of values of k. $d = 256$, $D_S = 1$.

have enough time to travel into and around the rim rather than being terminated). The value of d has a marked effect on the number of rays needed to stabilise the accumulator as seen in figure 3.24. Small values of d produce short rays, and require many rays to cover the entire image, whilst larger values need fewer. Additionally, as small values prevent the rays collecting into features easily, the transform requires more rays to reduce the inherent noise, in a similar way to when $n_{\max}$ is very small. As d increases many fewer rays are needed and the time taken for the transform reaches its minimum at $d \approx 100$, after which time increases as some rays are traced for longer than necessary to highlight features. A value of $d = 256$ provides the best balance between computational cost and the quality of the result of the transform.

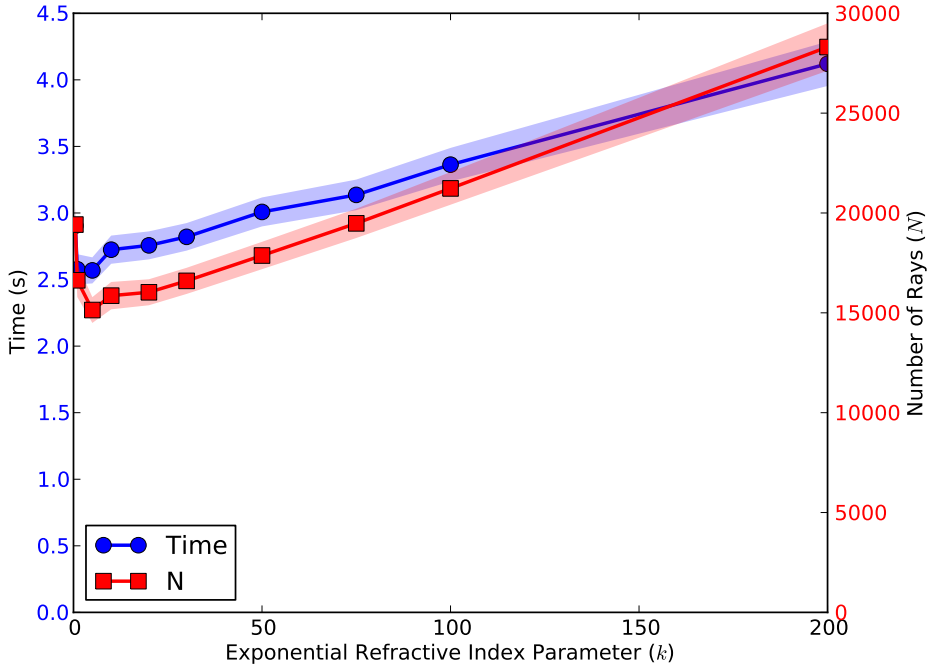
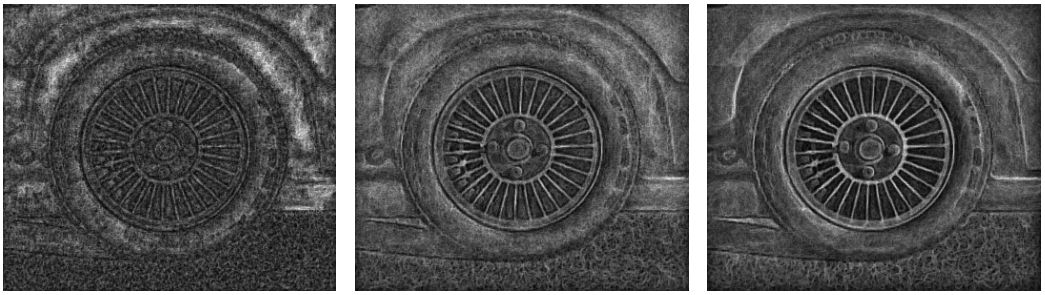


FIGURE 3.22: Mean number of rays need to reach stopping condition and time taken across 100 images with varying values of k. Shaded areas shows the standard error of the mean.

In order to ensure that any ray terminates within a reasonable amount of time, we include a maximum length parameter l, which is always set to twice the length of the image diagonal ($l = 2\sqrt{w^2 + h^2}$). After a ray has travelled a distance of l it is terminated. This value is set to be very large so that it is very rarely reached, as in all natural images the maximum depth will be reached beforehand however, it prevents a ray from continuing for a long time in some synthetic images.



(a) $d = 2$ (b) $d = 8$ (c) $d = 16$

(d) $d = 32$ (e) $d = 64$ (f) $d = 128$

(g) $d = 256$ (h) $d = 512$ (i) $d = 1024$

FIGURE 3.23: Results of ray transform for a range of values of d. $n_{\max} = 40$, $D_S = 1$.

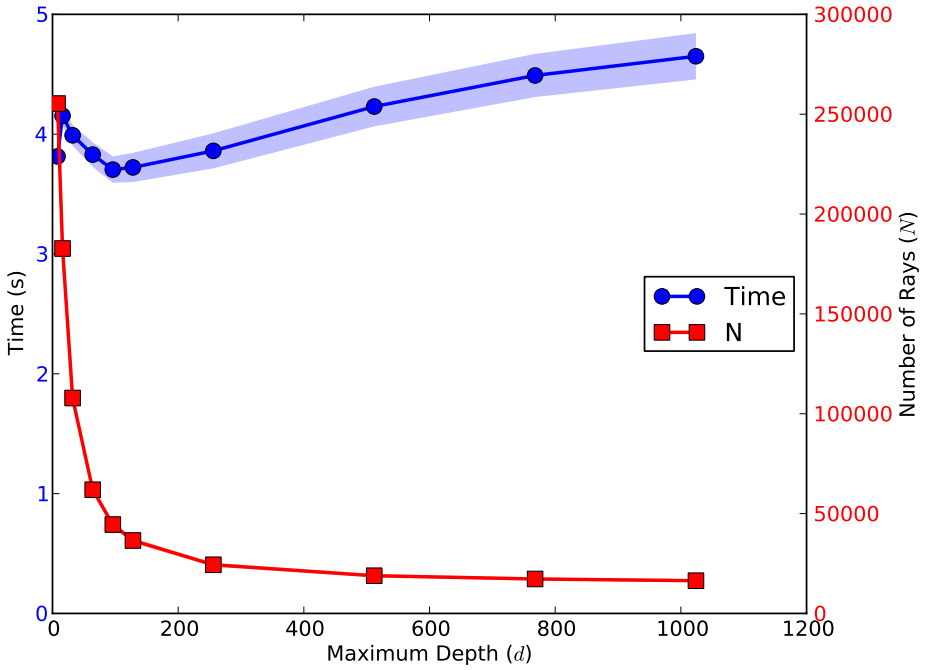


FIGURE 3.24: Mean number of rays need to reach stopping condition and time taken across 100 images with varying values of d. Shaded areas shows the standard error of the mean.

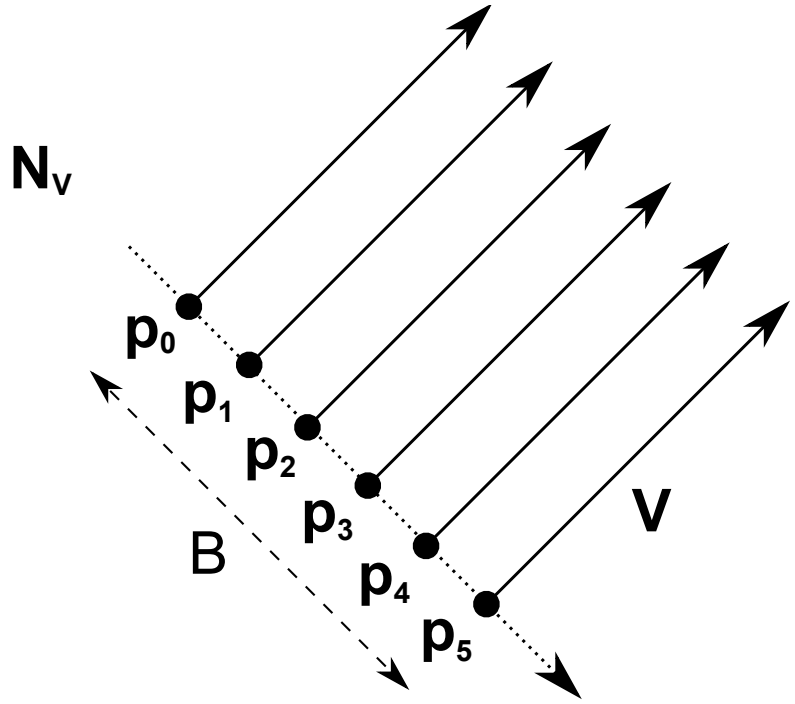


FIGURE 3.25: Initialisation positions ($\mathbf{p}$) of a number of rays in a beam, where $\mathbf{B} = 6$.

3.5 Beams

The IRT can be extended to use beams, multiple rays arranged in a line orthogonal to their direction. The addition of this mechanism produces a technique that is able to overcome noise successfully and extract larger features from images. Assuming $\mathbf{p}_0$ and $\mathbf{V}$ are the starting position and direction of a ray, assigned randomly as in the standard IRT, and $\mathbf{N_V}$ is the normal to $\mathbf{V}$ then the starting positions of the B rays $\mathbf{p}_b$ where $0 \geq b < B$ that constitute the beam, also of width B, are

$$\mathbf{p}_b = \mathbf{p}_0 + b \cdot \mathbf{N_V}. \tag{3.24}$$

This is illustrated by figure 3.25, showing the initialisation positions of a beam comprised of six rays aligned orthogonally to their velocity.

This alone does not add anything to the IRT. The rays must be followed in parallel and they must be forced to maintain a coherent single beam. This can be done by introducing a tether, a step where the directions of all rays that compose the beam are adjusted towards the beam's mean

$$\mathbf{V}'_b = (1 - \gamma_v)\,\mathbf{V}_b + \gamma_v \left(\frac{1}{B} \sum_{q=0}^{B-1} \mathbf{V}_q \right). \tag{3.25}$$

Figure 3.26 shows how a single ray q has its velocity changed so that it is a weighted sum of its original velocity ($\mathbf{V}_q$) and the mean velocity across the whole beam ($\overline{\mathbf{V}}$).

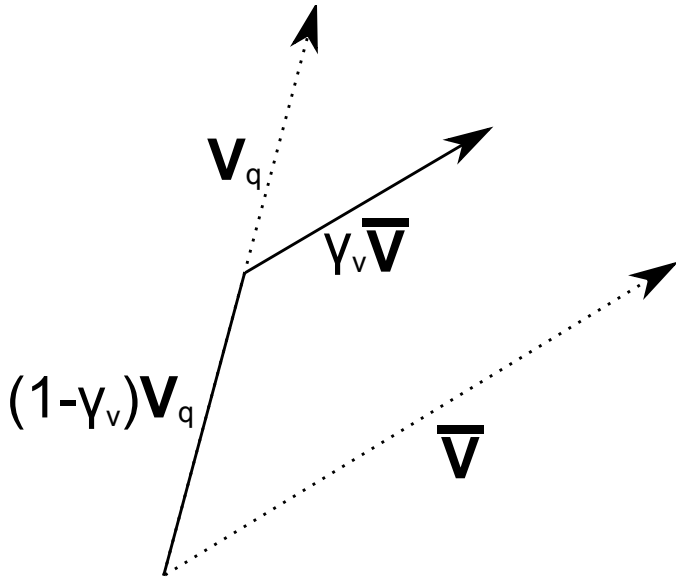


FIGURE 3.26: The tethering step, adjusting the velocity of individual rays towards the beam's mean velocity.

The parameter γ_v controls the rate at which the direction moves towards the mean, and so controls the extent to which individual rays can diverge from the beam. When $\gamma_v = 0$ no directional adjustment occurs whilst when $\gamma_v = 1$ all directions are set to the mean direction after every iteration. With an intermediate value such as $\gamma_v = 0.5$ the beam is able to change direction if the majority of the rays comprising the beam are changing in the same direction. Function 3.6 shows the implementation of the beam IRT in `castBeam`. Much of the pseudocode is similar to `castRandomRay`, but includes initialisation of rays orthogonally to their initial direction and the weighted sum addition of the each ray's direction with the beam mean after each ray has been moved.

The different results produced by the beam IRT compared to the IRT are shown in figure 3.27. In this image of a face, the IRT (figure 3.27(b)) provides a small amount of emphasis to the the forehead and hairline. With the strong tethering of rays in the beam IRT causing them to move as one, a strong highlight is provided around the border of the whole face (figure 3.27(c)). The single beam shown in figure 3.27(d) illustrates the way in which the beam starts to the left of the mouth and then moves together up the face, following the shape of the cheek, and round the forehead. An example of the use of beams to improve the transformation of noisy images is shown in section 4.2.3.

3.6 Implementation

The code presented in functions 3.1 to 3.6 provides an example of a possible implementation of the IRT. The actual implementation is similar, although optimised to a

Function 3.6: castBeam

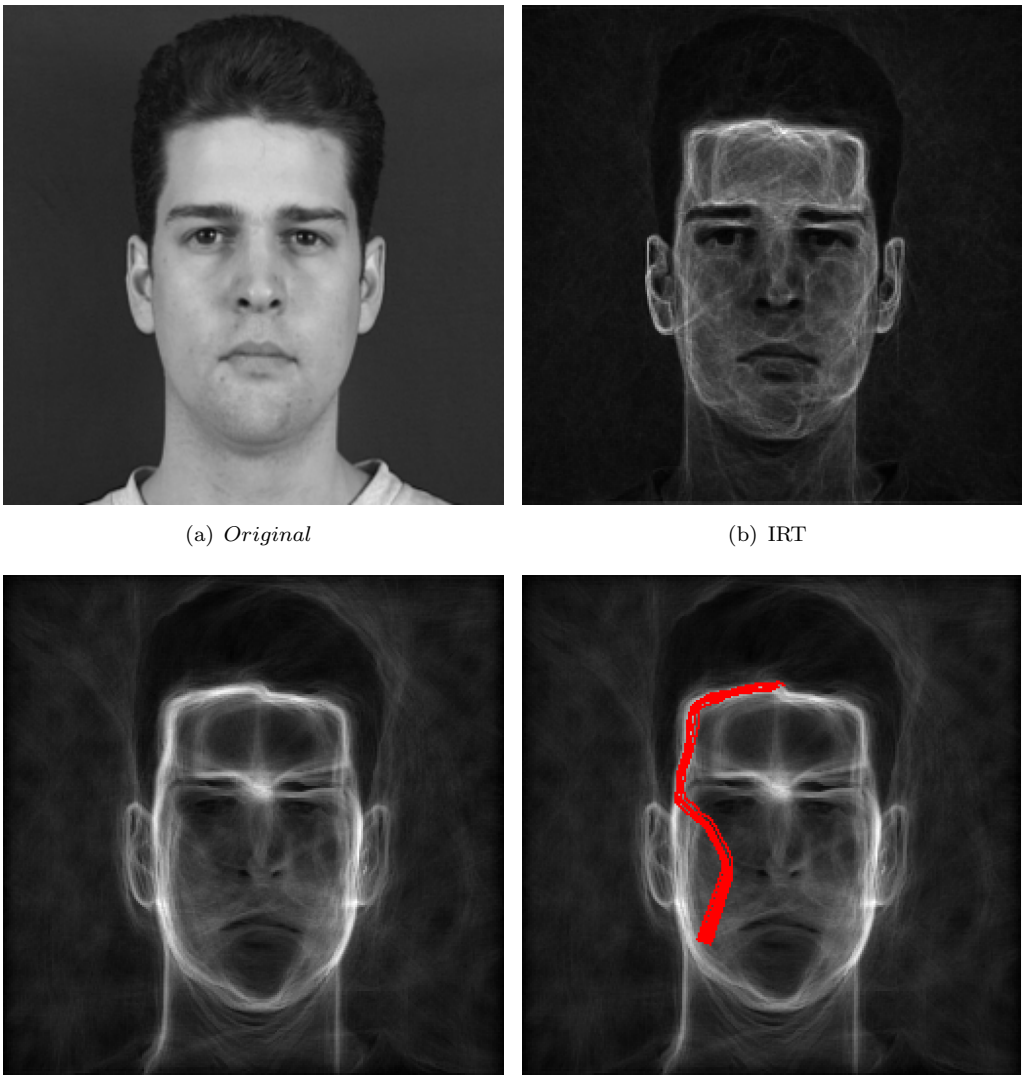
```
x, y, V ← randomRayInitialisation(width, height );
vNormal ← normal(V);
// p and v are arrays of size (B, 2) storing each ray's position and
   direction.
p[0] ← x, y;
v[0] ← V;
for b ← 1 to B do
    p[b] ← p[0] + b ∗ vNormal;
    v[b] ← initVx, initVy;
end
// depth and length are of size B, storing each ray's depth and length.
depth[:] ← 0, length[:] ← 0;
visited[:]← false;
// Stop casting the beam when any ray breaks a condition
while max(depth) ≤ maxDepth ∧ max(length) ≤ maxLength ∧ withinImage(p) do
    for b ← 0 to B do
        // Move ray a set amount, dealing with refractions/reflections and
           updating variables and the accumulator.
        moveBeamRay(p[b], v[b], depth[b], length[b], refractionMatrix, normals,
        accumulator, visited)
    end
    meanV ← mean(v);
    for b ← 0 to B do
        v[b] ← gammaV ∗ meanV + (1 − gammaV) ∗ v[b]
    end
end
```

greater degree. The transform was written in Python with critical sections written as a C extension with F2PY [68]. A Java demonstration applet was also produced [1]

Computation time depends partly upon selected parameters as shown in section 3.4. Those tests were performed on a 1.6 GHz processor, but on a more recent 2.5 Ghz processor an image of size 512×512, with standard parameters, computation was recorded to take 1.93 s. Whilst this is longer than techniques such as Sobel, which took 0.022 s on the same machine, it is still far from being computationally expensive. It should also be noted that the additional methods used for segmentation and detection in chapter 4, such as the HT, template matching and hysteresis thresholding, all have a computational cost significantly greater than the IRT. Whilst it has not been implemented, the IRT is highly parallisable. Each ray can be cast independently, only needing to update the accumulator on termination of the ray.

With regards to memory requirements, the transform is not excessive. Whilst the Python version was not optimised for memory usage, peak usage was approximately 40MB on

[1] `http://users.ecs.soton.ac.uk/ahc08r/rt/`



(a) *Original* (b) IRT

(c) Beam IRT (d) Highlighted beam

FIGURE 3.27: Example of difference in results between the beam IRT and the IRT.

a 512×512 image. With some sensible optimisations the memory foot print could be kept below 10MB for images around that size.

3.7 Conclusions

In this chapter we have shown how the principles of optics can be exploited through analogy to create a novel structural feature detector in the Image Ray Transform. A transformed image can be produced by treating the image as an array of glass blocks and casting randomly initialised rays through it whilst recording their paths. The behaviour of the transform with several parameterisations has been discussed, showing how differing results can be produced through variation of one or more parameters. Throughout

we have provided sample implementations and discussed the issues of the computational cost of the transform as well as how different parameter values affect it. Additionally we have discussed enhancements that aid parameter selection through stopping conditions, as well as methods to significantly alter the results and improve the transform through alternative models of refractive indices and the use of rays aligned as a beam.

Chapter 4

Structural Feature Detection

In chapter 3 we presented the Image Ray Transform (IRT) and described it as a structural feature detector, particularly for circular and tubular structures. In this chapter we show experiments in circular detection, biometrics and medical imaging that demonstrate the IRT's strength in this role. There are of course numerous other areas for which the IRT would be appropriate for use as a structural feature detector, these applications being selected to demonstrate its wider utility.

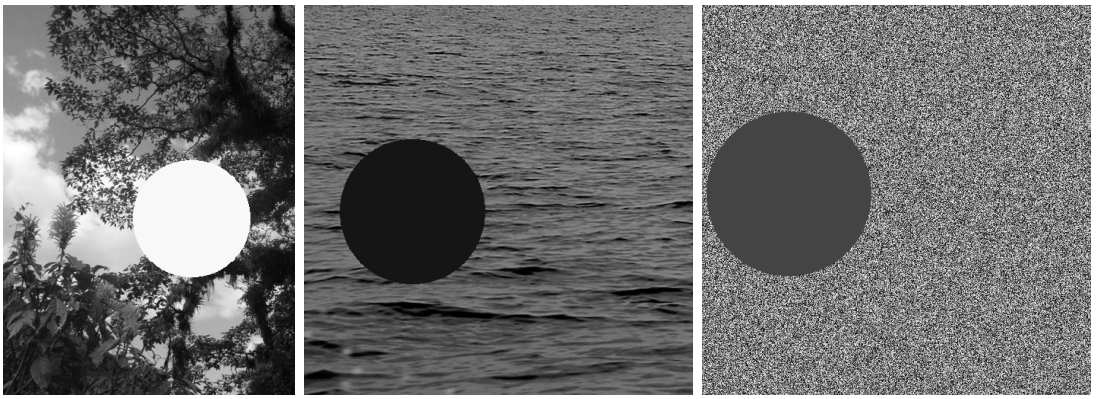
This chapter proceeds in the following manner. Section 4.1 documents the use of the Image Ray Transform to enhance circle detection with the Hough transform (HT). Section 4.2 describes use of the transform to detect tubular structures for the enrolment of ear biometrics, comparing enrolment and recognition performance with other automatic and manual methods. Tubular structures in retinal fundus images are detected and evaluated in section 4.3 before we draw conclusions.

4.1 Circle Detection

Section 3.2.4 showed some examples of the image ray's transform ability to highlight circles. This occurs because rays tend to bounce around the edge of the circle as they enter, creating a strong gradient. Due to the lack of a suitable dataset for testing circle detection techniques, one was produced and tested with the HT for circles.

4.1.1 Synthetic Circles on Natural Images

To test the suitability of the IRT for circle detection, a number of empirical tests were performed. Images were created by adding a number of circles onto a series of background images with complex features, displayed in figure 4.1. Images of the type shown in figure 4.1(c) had backgrounds randomly generated wherein the intensity of each pixel



(a) Tree (b) Sea (c) Noise

FIGURE 4.1: Examples of images generated on different backgrounds.

was drawn individually from a uniform distribution with interval $[0, 255]$, whilst the other images (figures 4.1(a) and (b)) had the same background for each test image. Each test image had a different circle placed upon it with random position and radius (x_0, y_0 and r_0). These 1000 generated images were then processed by the IRT with the parameters set as $N = 10000$, $n_{\max} = 40$, $d = 256$. Two transforms were performed with $\tau = \{0, 255\}$ and aggregated as in function 3.5 so as to detect both light and dark circles on backgrounds of both higher and lower intensity.

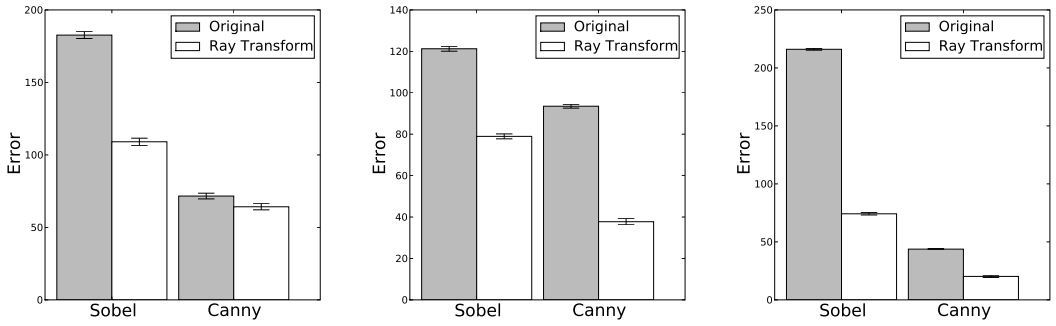
We apply an edge detector to the transformed and original synthetic images (we tested with both a thresholded 3×3 Sobel and the Canny operators). Edge detection was necessary on the images transformed with the IRT as the gradient around the circle does not produce a single strong edge suitable for the HT. These edge detected images were then tested with the decomposed HT for circles (described in section 2.2), providing a detected position (x, y) and radius (r). The distance to the correct position and radius was then used to as the error to evaluate the performance of the methods:

$$\text{Error} = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (r - r_0)^2}. \tag{4.1}$$

4.1.2 Circle Detection Results

Figure 4.2 shows the mean detection error across all images for each category of synthetic image. The error bars represent the standard error of the mean detection error. The application of the IRT to the images prior to edge detection provides a marked improvement in performance.

Both the Canny and Sobel operators rely upon a change in intensity to produce edges, and in the thresholding process weak edges will often be removed and will not contribute to the voting process in the HT. The IRT highlights structure, and so small

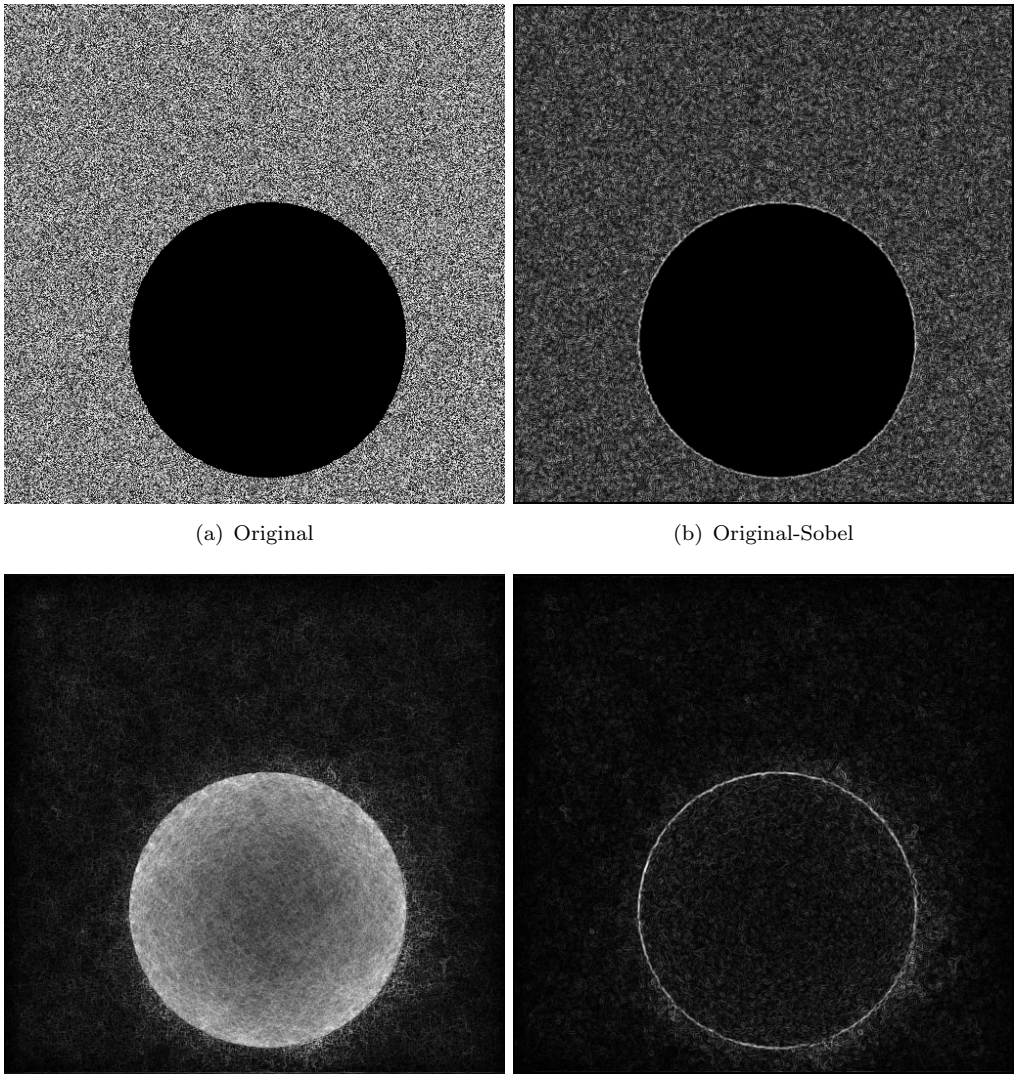


(a) Sea (b) Tree (c) Noisy

FIGURE 4.2: Mean error for circle detection on different background images.

intensity differences can be exaggerated to the point where they become more significant. Figure 4.3 shows an original and ray transformed image having undergone the Sobel operator. Whilst the application of the ray transform introduces some noise within the circle, the edge around the circle is far stronger than any other edge in the image. This contrasts with the Sobel operator alone where the edges created by the background are of comparable strength to the edge surrounding the circle.

In images set on a background of random noise, use of the ray transform improves detection at all intensities of circle, but has a more significant effect in cases where there is very little difference in intensity between circle and average background intensity (figure 4.4). With the Canny operator and a noisy background, both the ray transformed image and the original image perform well on most circle intensities. The ray transform is generally more accurate because it does not need such a high level of smoothing to reduce the strength of the background noise, and so the detected centres are not displaced by as great an amount. The increase in error in detection in both cases coincides with the intensity of the circle being close to the mean intensity of the background. The ray transformed images achieve significantly better results than the original images in these cases however, as the results for the original images suggest that no circle was found in any case, whilst in the IRT image the correct circle was found in most cases. This extra accuracy shows that the extra computational effort required to preprocess the images with the IRT is worthwhile.



(a) Original

(b) Original-Sobel

(c) Ray Transform

(d) Ray Transform-Sobel

FIGURE 4.3: The results of a single image after the application of a range of techniques.

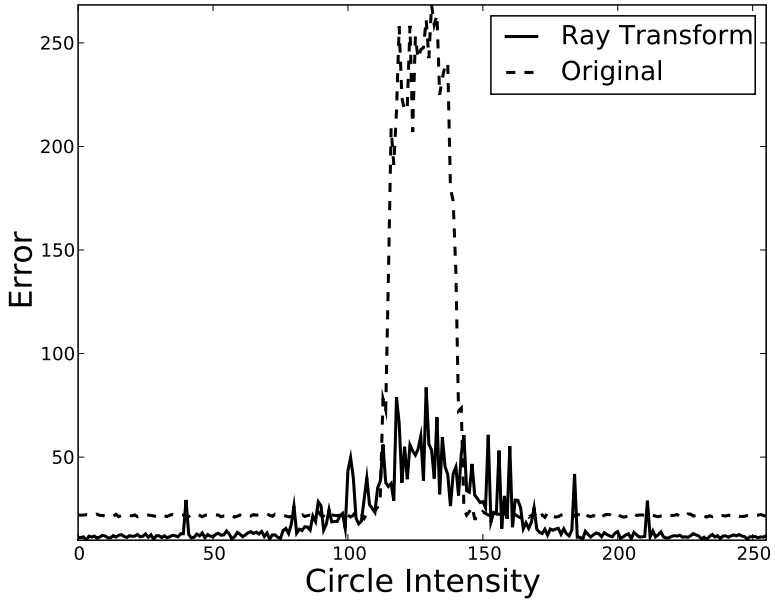


FIGURE 4.4: Error across circle intensities for Canny HT tests with a noisy background

(a) Ear image (b) IRT result

FIGURE 4.5: An ear image from XM2VTSDB and the structures which the IRT emphasises.

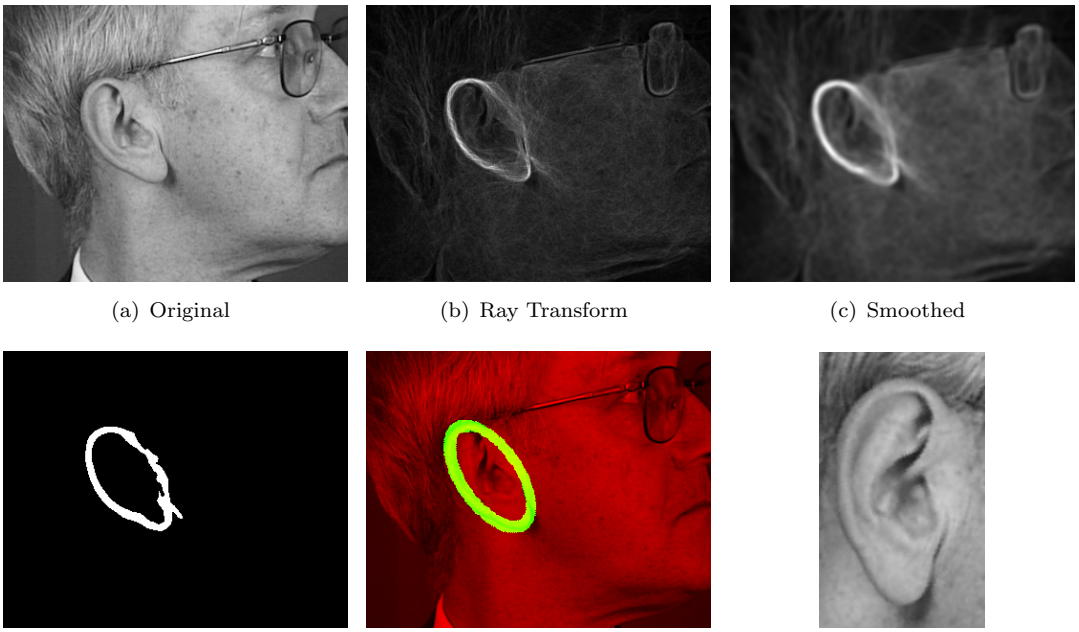
4.2 Enrolment for Ear Biometrics

The IRT's ability to highlight tubular features is its most powerful and widely applicable feature. One area in which it can be demonstrated is the realm of ear biometrics. Around the edge of the ear is the helix (see figure 2.1) which acts a tubular structure. In figure 4.5(b) it can be seen that the transform has highlighted the helix (and part of the lobe) prominently, and this can be exploited to create an enrolment technique. This technique uses the IRT to create an image in which enrolment is considerably easier than in any edge or intensity image. Enrolment then occurs using a simple template matching technique, but many other enrolment techniques could be enhanced through use of the IRT as a preprocessing step on the original ear image.

4.2.1 Enrolment with the Image Ray Transform

The technique used here exploits the IRT's strength at extracting tubular features to highlight the helix of the ear, and then uses a series of simple steps to extract and normalise the ear. It should be noted that the main contribution here is the application of the IRT: without doubt more complex techniques (including some of the alternative enrolment methods described in section 2.3.2) could be used to increase the success of the enrolment process.

The initial step is to apply the IRT to the ear image with the parameters $D_S = 1, n_{\max} = 40, d = 256$. These parameters are standard values that produce an image that has enough rays of sufficient length cast to reduce the noise to an acceptable level, and the high value of $n_{\max}$ makes rays conform strongly to structural features within the image. This produces an image in which the helix of the ear is highlighted (figure 4.6(b))



(a) Original (b) Ray Transform (c) Smoothed

(d) Threshold (e) Detection (f) Extraction

FIGURE 4.6: Example of the steps taken to achieve successful ear enrolment.

(a) (b) (c) (d) (e) (f) (g)

FIGURE 4.7: A selection of the templates tested for enrolment.

(sometimes in addition to other tubular features such as spectacles). Due to the non-deterministic nature of the transform, it is advantageous to apply Gaussian smoothing to the result (figure 4.6(c)) to reduce noise. The image is then thresholded (figure 4.6(d)) to produce an image with a strong helix as its focus. Histogram equalisation is used to allow a consistent threshold (of 249) across all images to be used. Template matching is then used with an elliptical template across a range of rotations and scales (figure 4.6(e)). A number of templates that were used are shown in figure 4.7; however, the strongest result was found with the modified elliptical template in figure 4.7(g). This template is wider on the left side, in order to match the tendency of the IRT to have a stronger response around the helix than the tragus. The templates were rotated between 0° and 45°, and scaled around the average size of the ears, and all were convolved with the IRT result. The matched section is then normalised and extracted (figure 4.6(f))

Technique	DB size	Detection Rate
IRT	252	99.6%
Elliptical HT [4, 5]	252	100%
Homography [11]	252	100%
Cascaded AdaBoost [41]	203	100%
Colour and Shape [71]	150	94%
Banana Wavelets [40]	252	100%

TABLE 4.1: IRT enrolment and other previous ear enrolment results.

using the template parameters.

4.2.2 Enrolment Results

Enrolment was performed on 252 images (4 per subject) from the XM2VTS database [58] using the technique above. This is a section of the database without occluded ears that has been used previously [38], and provides basis for comparison. The mean computation time for each ear for the entire technique was 5.45 s, whilst the ray transform alone took only 1.47 s and required approximately 19000 rays to be traced. In figure 4.8, a range of images after the thresholding stage are shown. The propensity of the technique to extract the helix strongly in all images is shown, as well as cases where spectacle frames (figures 4.8(a), (b) and (c)) or other features are highlighted (light hair in (a) and part of the forehead in (d)). The extent to which features are highlighted depends strongly upon their intensity (as discussed in section 3.3.2), primarily highlighting ear helices due to their high intensity skin surrounded by low intensity shadows. Other structures such as some spectacle frames, and more rarely hair, are highlighted through this mechanism as well. In general it was found that these extra features retained through smoothing and thresholding do not affect the enrolment results in most cases, because they are rarely of an elliptical shape of the type that would be found in the template matching stage.

The results of the extraction were encouraging and figure 4.9 shows a selection of the extracted ears. Out of 252 images, 99.6% of extracted images contained the subject ear. 98.4% of these images had the ear correctly normalised, with the incorrectly normalised images having small scale or rotation errors. Table 4.1 presents our results in comparison previous work into ear detection and enrolment. Where possible we have presented results for the XM2VTS database as it is the most common database. Whilst detection results in the literature appear impressive, the connection between them and a high recognition rate (a much better test of a biometric) is weak, and good recognition depends upon more accurate localisation and normalisation than these techniques tend to provide.

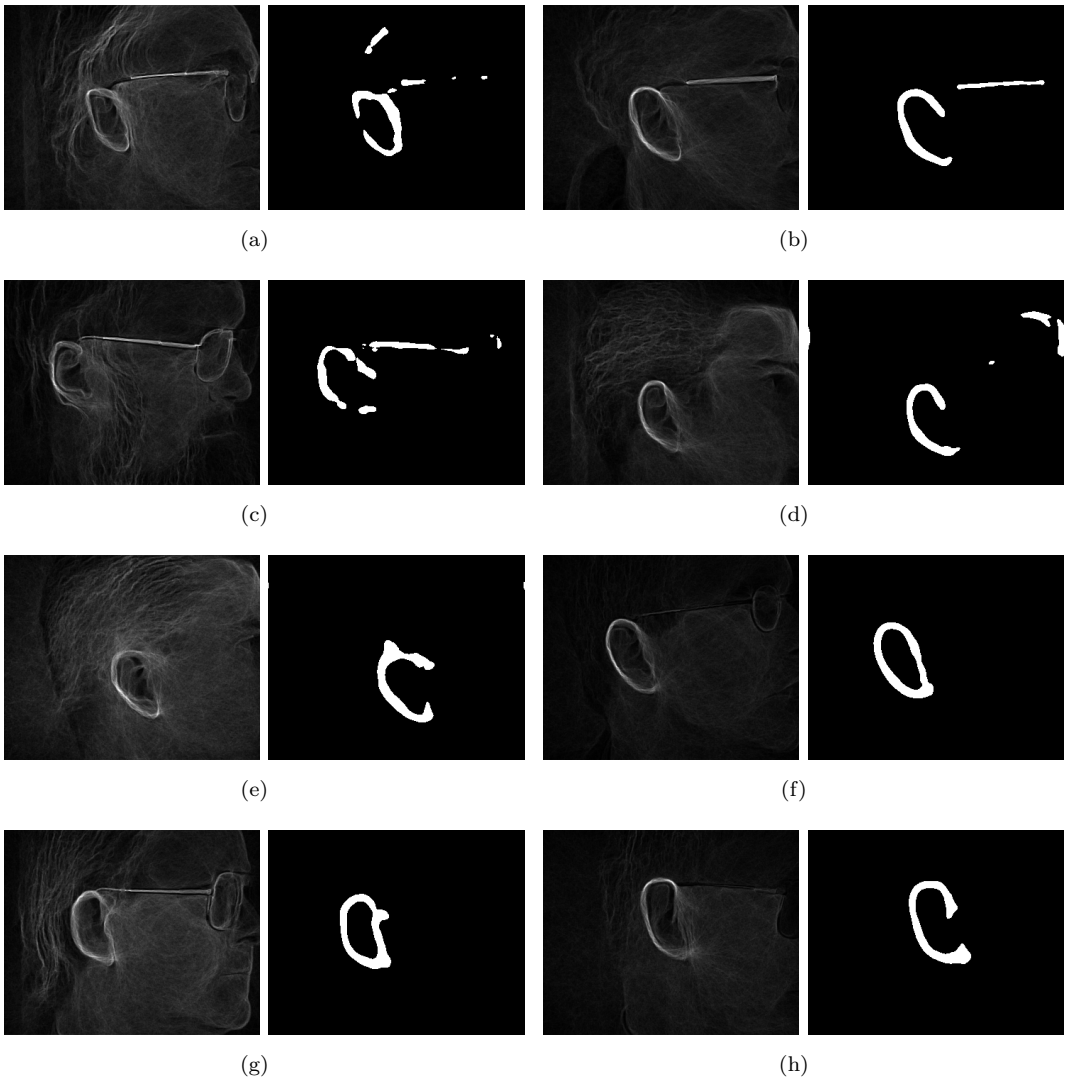


FIGURE 4.8: Selection of transformed images before and after smoothing and thresholding.

The extension of the transform to beams described in section 3.5 can be used to improve the robustness of the ear enrolment technique to noise. Gaussian noise ($\sigma = 20$) was added to the 252 ear images and the tests repeated as previously described, using the beam IRT with parameters $B = 8$ and $\gamma_v = 0.5$ (shown in figure 4.10). The use of the beam transform on noisy images increased the rate of detection of ears from 93.3% to 96.8%, and rate of correct normalisation of detected ears from 64.7% to 93.9%.

Figure 4.11(a) shows the only example of this detection failure on the original ear images. Whilst the IRT emphasises the ear strongly, it has similarly done so with the forehead, to the extent that the latter matches more strongly with the template. With the addition of noise both the IRT and its beam variant failed in detection more frequently. Figure 4.11(b) shows an example where a number of structures in combination have created one larger structure that responds strongly to both the IRT and the template matching. The

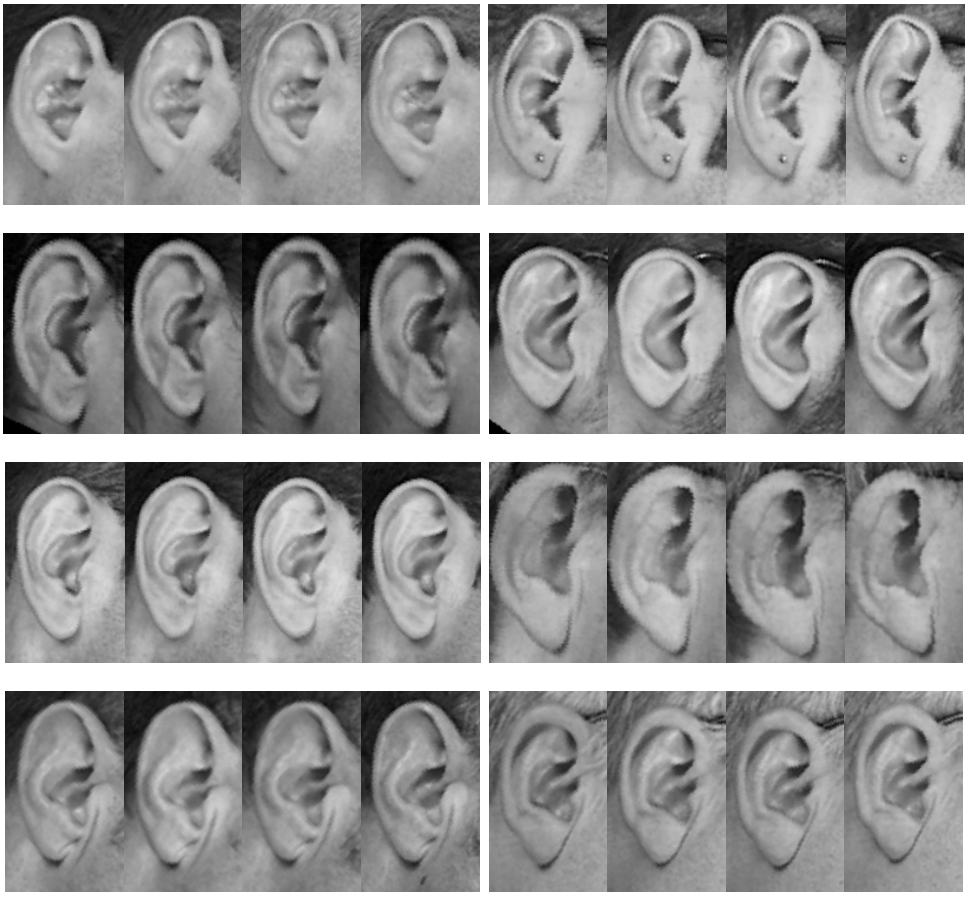
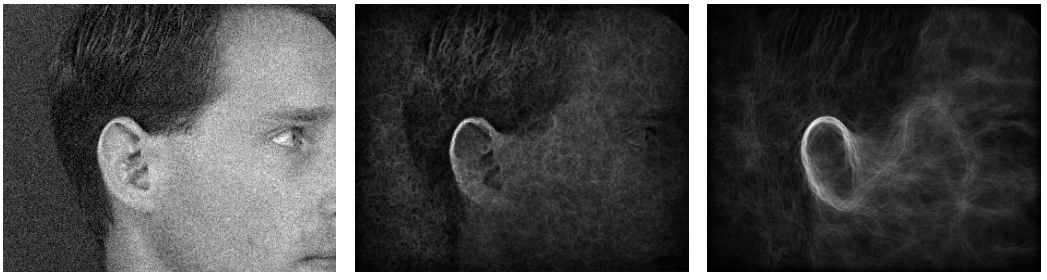
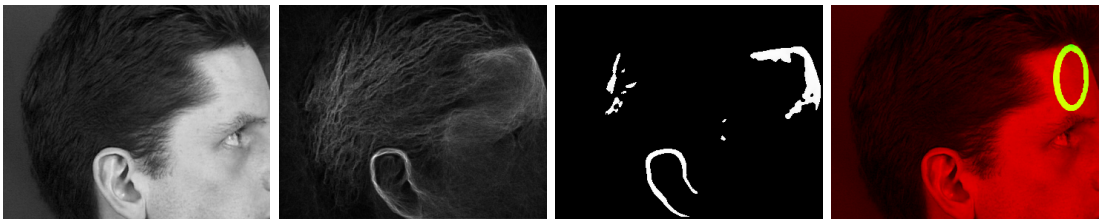


FIGURE 4.9: Extracted and normalised ears for a selection of subjects

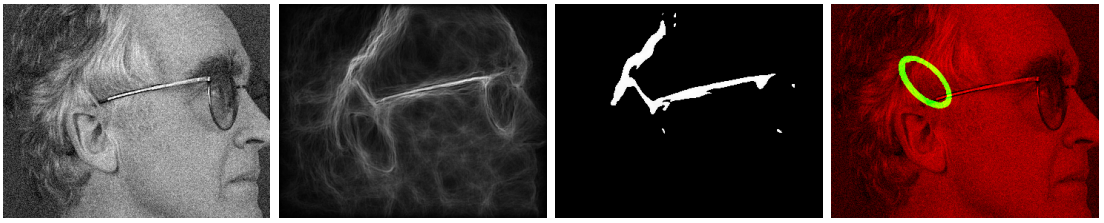


(a) Noisy Image (b) IRT (c) Beam-IRT

FIGURE 4.10: Noisy ear image transformed with the versions of the IRT.



(a) IRT Failure



(b) Beam IRT failure

FIGURE 4.11: Examples of failure of the ear enrolment technique.

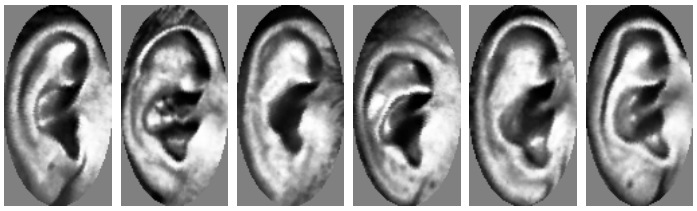
spectacle frames and white areas of hair are both highlighted strongly by the transform and the template matching step, to a much greater degree than the ear. There are a number of improvements to the enrolment technique that could be made to deal with these issues, including removal of straight lines prior to thresholding and the use of a superior matching technique, such as the Hough transform.

4.2.3 Recognition Results

We quantitatively test the ear enrolment technique by using principal component analysis (PCA) (see section 2.3.1) for recognition on enrolment by the IRT. PCA recognition is a powerful technique but is reliant on accurate ear registration, making it well suited to test the IRT-based enrolment technique. We use a standard PCA implementation [9] as used previously for ear recognition [14, 82]. Images are normalised for position, scaled to be the same as that of the template (80×150) and are rotated to be vertical using the parameters found during template matchings. Recognition was attempted on both the original image (figure 4.12(a)) as well after application of histogram equalisation and masking of the images (figure 4.12(b)). The mask was based upon the template used to match the ear, setting all pixels outside the ellipse to a medium intensity. In all cases results were significantly better using the original images, rather than those with equalised histograms, masks or any combination of the two. During PCA we discard the first eigenvector (as it usually represents illumination variance only) and retain 60% of the remaining eigenvectors, as done by Chang et al. [14].

We compare our recognition results on the 63 subjects of the XM2VTS database with those of the automated elliptical Hough transform-based enrolment system of Arbab-Zavar and Nixon [4], as well as the results on manually enrolled images. Table 4.2 shows

(a) Original Ears



(b) After masking and histogram equalisation

FIGURE 4.12: The ear image that failed to be extracted correctly

Enrolment Technique	Recognition Rate
Manual [5]	98.4%
IRT (Yambor Angle)	95.2%
IRT	93.7%
Elliptical HT [5]	75.1%
Homography [11]	96%

TABLE 4.2: Rank one recognition rates of different enrolment techniques with PCA.

the rank one recognition rate on PCA with the Manhattan distance (except where stated otherwise), using one of the subjects images as a probe and the rest as the gallery. Our method is clearly superior to that of the Hough transform technique, provides similar performance to the distance based recognition of the Homography method (whilst at a significantly lower computational cost) and is approaching the quality of manually enroled images. Further improvement can be made through use of the Yambor Angle metric [81], similar to the Mahalanobis distance, where the distance between two vectors u and v are normalised by the size of each eigenvalue, λ:

$$Y(u,v) = -\frac{1}{|u||v|}\sum_{i-1}^{k}\frac{1}{\sqrt{\lambda_i}}u_i v_i. \tag{4.2}$$

Recognition was also tested on noisy images using both the IRT and the beam variant. The averaging effect of the beams caused the rank one recognition rate to be increased from 44.4% to 74.6%.

We can see that the IRT-based ear enrolment technique is capable of performing quick detection and normalisation of ears, with sufficient accuracy to provide recognition results approaching those of manually enroled images. The addition of the beam IRT

variant allows the enrolment method to overcome the presence of Gaussian noise within the image. Most importantly, the strength of the technique is derived from the preprocessing provided by the IRT, and the later steps could easily be replaced with other good enrolment for a further improved result.

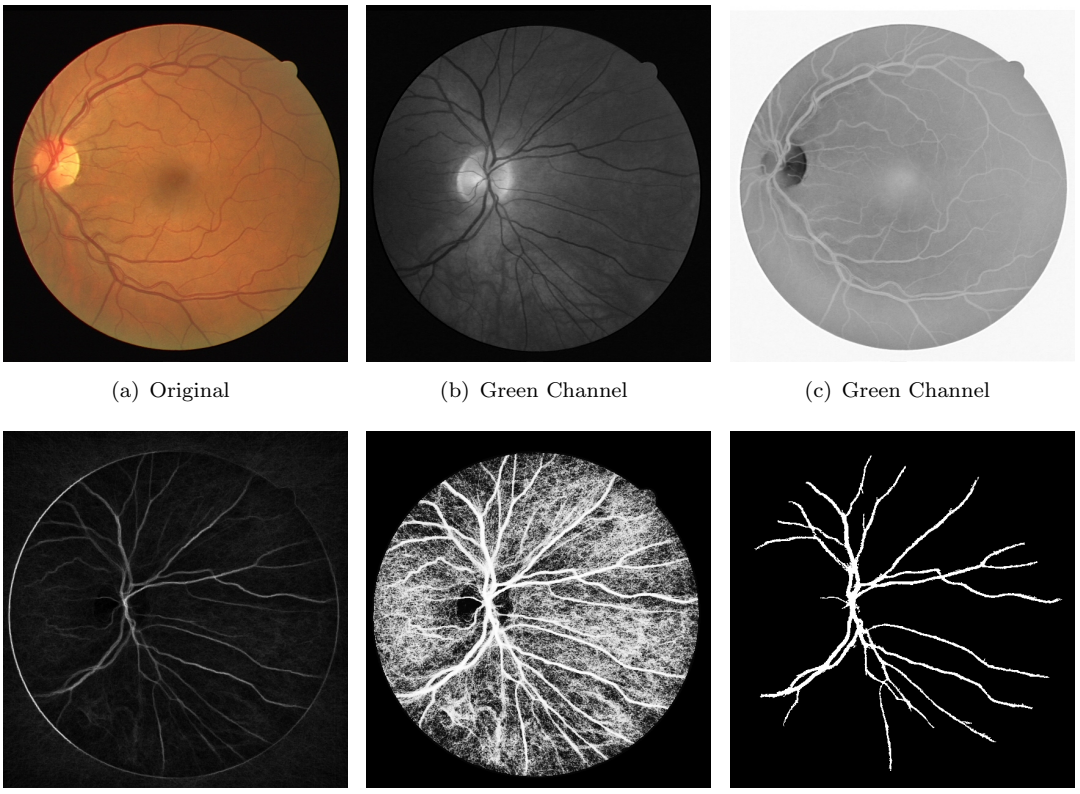
4.3 Segmentation of Blood Vessels in Retinal Images

The detection of blood vessels within eye fundus images is an another appropriate application to further demonstrate the IRT's ability to highlight tubular structures. Zana and Klein [85] describe vessels to be "Bright features defined by... linearity, connectivity, width and by a specific Gaussian-like profile.". Such a description aligns very well to the tubular, fibrous structures that the IRT is capable of highlighting. As we do not use any advanced techniques for classifying the images that result from the transform, we do not expect accuracy superior to other techniques (although we try to establish that this is possible with superior classification techniques). We primarily intend to show that the IRT is a suitable preprocessor for many retinal vessel extraction techniques, more appropriate than the original or edge detected images. Current techniques do not achieve accuracy of much more than 95%, and through use of an appropriate preprocessing technique such as the image ray transform we suggest that these results may be improved further.

4.3.1 Extraction Technique

We use the DRIVE database [77] to test our technique and compare to others as this is a standard database with ground truth available. The database consists of 20 test and 20 training fundus images, of which we only use the test images. The green channel of the retinal image is extracted for use as it provides the greatest contrast for blood vessels (figure 4.13(b)). The image is effectively inverted by the use of the target parameter (figure 4.13(c)), so that the lighter vascular structures are highlighted most strongly by the transform. Use of the transform also introduces additional problems. The edge of the fundus image acts as a circle and is highlighted by the transform; we expand the masks included in the database by three pixels in order to remove these unwanted features. The exponential transform also highlights the fovea strongly, which is undesired, and so we used an automatic method (template matching with a Gaussian template) to find and remove them.

The image ray transform was applied to the green channel images (figure 4.13(d)) using both linear (referred to as IRT-n, equation 3.3) and exponential refractive indices (IRT-k, equation 3.23), as both gave results that differed significantly. Parameters used were $d = 256$, $\tau = 0$ and $D_s = 1$. For the linear refractive indices $n_{\max} = 40$ and for the



(a) Original (b) Green Channel (c) Green Channel

(d) IRT (e) Histogram Equalisation (f) Hysteresis Thresholding

FIGURE 4.13: Steps taken to extract retinal blood vessels.

exponential $k = 7$. We also tested on aggregated version of linear and exponential indices (IRT-nk), so as to exploit the advantages of both techniques.

The results of the transform were then histogram equalised (figure 4.13(e)) as this allowed thresholds to be selected that are appropriate across all transformed images. Finally hysteresis thresholding (figure 4.13(f)) was performed to segment the image into vessel and background pixels. The upper threshold for hysteresis thresholding was set to 253 for all ray transform techniques, whilst the lower threshold was selected to give the highest accuracy across the database (IRT-n: 234, IRT-k: 235 and IRT-nk: 230).

We compare individual pixels in the thresholded images with the ground truth, segmenting them as vessel or non vessel depending upon their thresholded value and calculate the performance of the technique. Two common metrics were calculated that are often used to test the strength of retinal blood vessel extraction: maximum average accuracy (MAA) and the area under curve of a ROC graph (AUC).

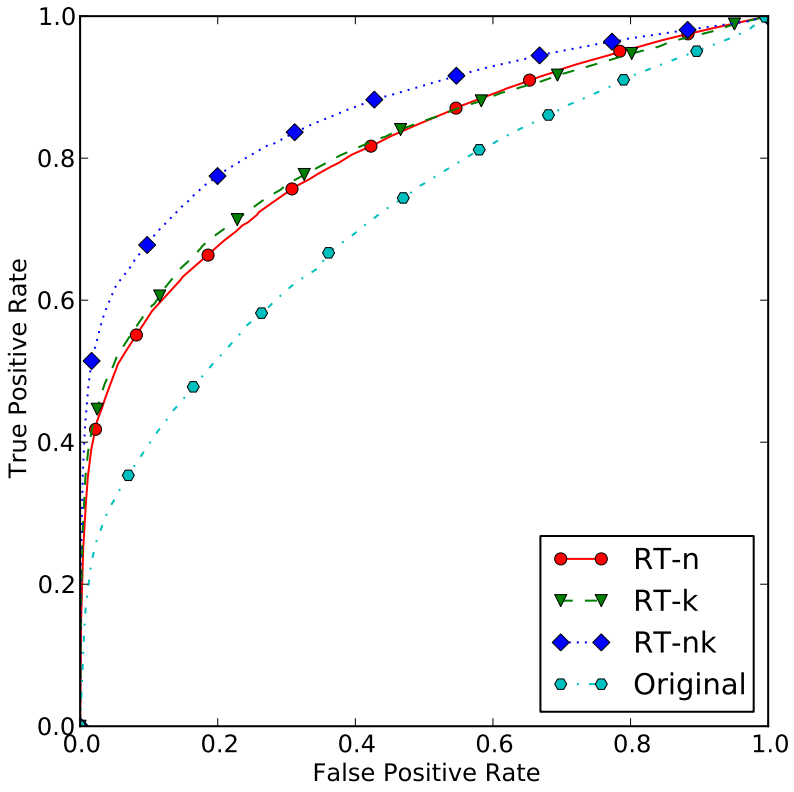
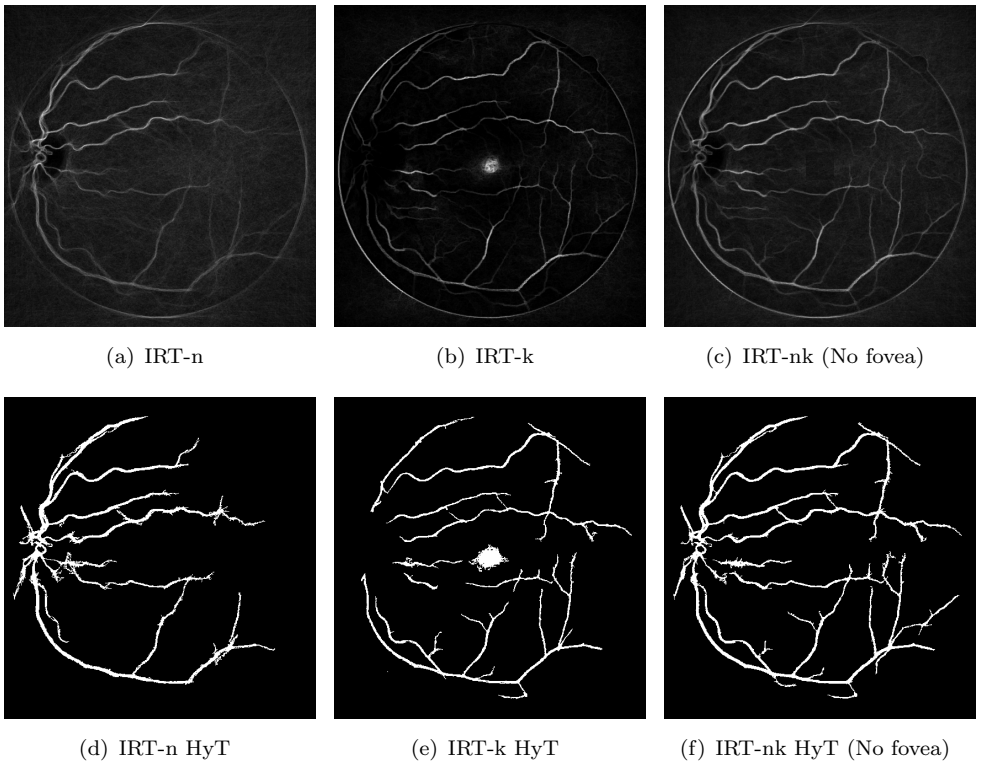


FIGURE 4.14: ROC curve of the discriminatory ability of variants of our technique with the image ray transform.

4.3.2 Extraction Results

This technique is capable of segmenting pixels with accuracy comparable to a selection of contemporaneous unsupervised techniques. Figure 4.14 is a ROC graph showing the ability of each variant of our technique to segment vascular pixels correctly. The graph was produced by varying the value of the lower hysteresis threshold to segment more or less pixels as vessel or non-vessel. Whilst all variants are significantly better than the untransformed images, the aggregation of the two different types produces a marked increase in discriminatory ability, suggesting that different vascular structures are emphasised by each. The transforms themselves do not have prohibitive computational cost: the linear transform took 4.65s on average whilst the exponential transform took only 1.91s on a 2.53GHz processor. The transform with exponential refractive indices was significantly faster as rays adhered to the vessels more strongly and so the resultant image had less noise and converged more quickly. These times are, however, significantly shorter than the time that the rest of the classification process took.

In figure 4.15 a number of transformed images and their thresholded versions are displayed. The ray transform with linear refractive indices (figures 4.15(a) and (d)) is adept



(a) IRT-n (b) IRT-k (c) IRT-nk (No fovea)

(d) IRT-n HyT (e) IRT-k HyT (f) IRT-nk HyT (No fovea)

FIGURE 4.15: Selecting retinal blood vessels by variants of the ray transform and hysteresis thresholding (HyT).

at highlighting larger features and those near the optic disc but introduces some noise across the image. In contrast, using exponential refractive indices (figures 4.15(b) and (e)) forces rays to adhere to small vascular features more strongly, introducing less noise across the whole image. However it highlights the fovea in the centre of the image and fails around the optic disc. The highlighting of the fovea was removed through matching with a Gaussian template before segmentation. These complementary results allow the aggregated images (figures 4.15(c) and (f)) to have the strengths of both and negates some of the weaknesses. Most vessels highlighted by either version are present in the thresholded combined version, and it has been improved by automated removal of the highlighted fovea. Results for different lower thresholds retain more vascular pixels, but also segmented more noisy background pixels as vessels: a segmentation technique less vulnerable to noise would improve results considerably.

Table 4.3 and figures 4.16 and 4.17 show the results for our technique in comparison with others of varying complexity, also on the DRIVE database. Using our simple segmentation technique, the ray transformed images achieve superior MAA and AUC values than the original intensity images, implying that the ray transform has emphasised the vascular features to a greater extent than they had been before. Our method is comparable to other unsupervised techniques, despite the simplicity of our classifier. As expected, supervised pixel classification techniques produce superior results, but through

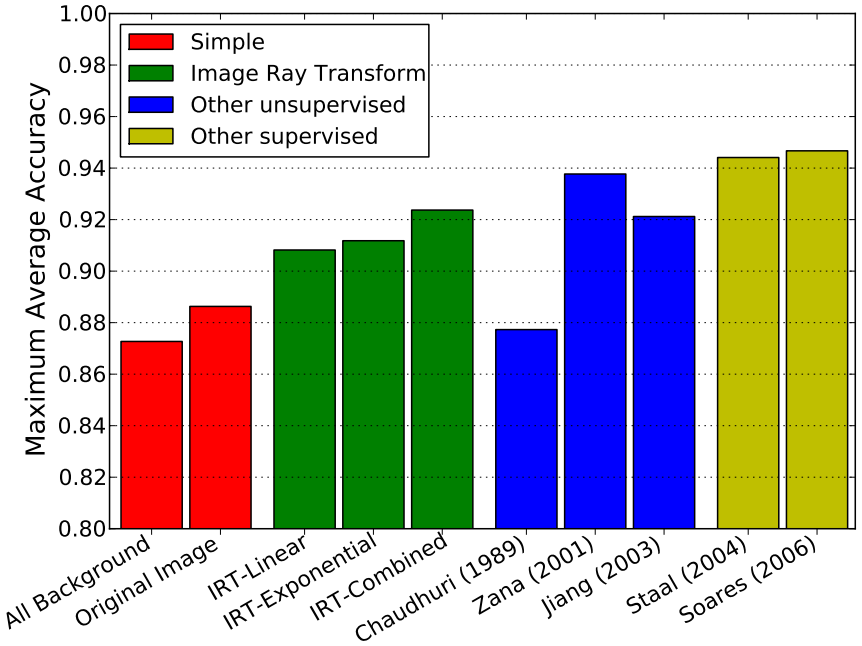


FIGURE 4.16: Maximum average accuracy of the IRT technique and other techniques.

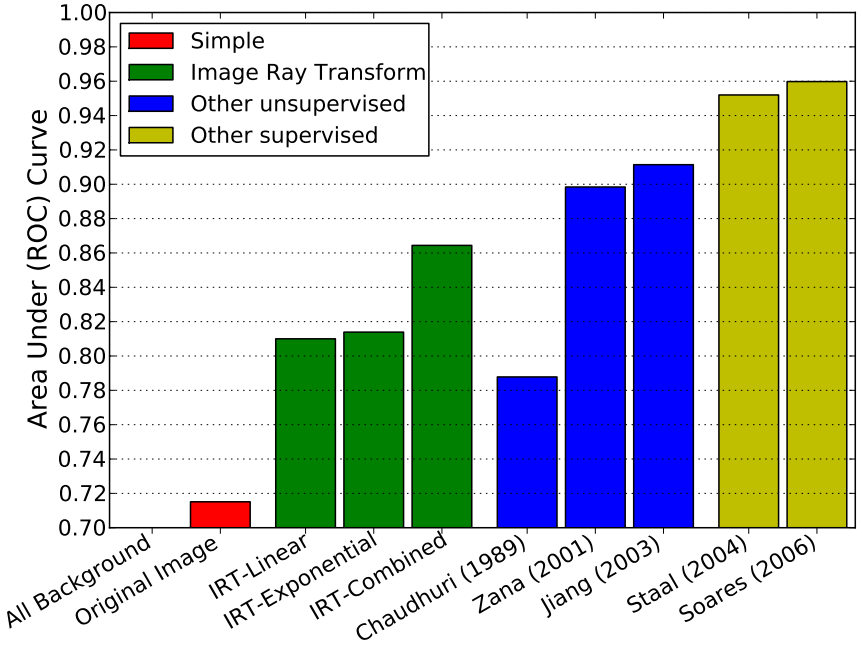


FIGURE 4.17: Area under the ROC curve of the IRT technique and other techniques.

Technique	MAA	AUC	Notes
IRT-n	0.9082	0.8100	Linear Refractive Indices
IRT-k	0.9118	0.8139	Exponential Refractive Indices
IRT-nk	0.9237	0.8644	Aggregated IRT-k and IRT-n
Original Image	0.8863	0.7151	Original, our classification technique
All Background	0.8727	-	Most likely class
Chauhuri et al. [15, 62]	0.8773	0.7878	Unsupervised
Zana et al.[85, 62]	0.9377	0.8984	Unsupervised
Jiang et al.[45, 62]	0.9212	0.9114	Unsupervised
Staal et al. [77]	0.9441	0.9520	Supervised
Soares et al. [76]	0.9467	0.9598	Supervised

TABLE 4.3: Maximum average accuracy (MAA) and area under the ROC curve (AUC) for our technique and others

use of the image ray transform as a preprocessor to a supervised learning method it should be possible to improve results further.

4.4 Conclusions

This chapter has shown how the IRT can be used to highlight structural features in images, and how this can be employed for specific applications. Further, the computational complexity is such that it appears to be an interesting contender for a choice as a preprocessing stage for many applications. We have shown that it can act as a preprocessor for circle detection and significantly improve results with different techniques, even on difficult cases where there is little intensity difference between the circle and background.

The transform's aptitude for highlighting tubular features has been employed in order to create an effective technique for ear enrolment and retinal blood vessel segmentation. The transform has an inherent ability to strongly highlight the helix of the ear in all cases, and only rarely highlights other facial features in similar ways, making extraction simple and reliable. The quality of this enrolment has been shown in successful tests on recognition. Clearly, the IRT is a low-level feature extraction technique which can be specifically tuned to ear extraction. Additionally we have shown that through the use of beams as local averaging the IRT can be improved so that it is more robust to noise.

The IRT's ability to highlight retinal blood vessels has been used with a very simple segmentation technique to produce performance comparable with other techniques. Results show that the transformed images highlighted vascular features to a greater extent than they are highlighted in the original image, suggesting that use of the transform as a preprocessor of better pixel classification or segmentation methods will increase their performance.

Chapter 5

Ray Image Descriptors

In chapter 4 we demonstrated how the Image Ray Transform (IRT) can be used to highlight structural features within an image through the casting of a large number of random rays through an image, and analogising structures to optical fibres. Whilst this technique is useful in cases where we aim to highlight either all structures or those of a specific shape, we also can exploit the detection of structural features to describe images at a higher level. Any object can be said to be composed of a number of structural features: a bicycle has two wheels and a frame, whilst a table has a number of legs and a top. Figure 5.1 shows how rays can can travel along these structures; if they can be

FIGURE 5.1: Illustration of the detection of structural features within a bicycle by rays.

represented with sufficient invariance they can be used to categorise objects using the bag-of-visual-words model.

5.1 Descriptor Based Upon the Image Ray Transform

The rays that are used to highlight structural features during the IRT can also be used to describe some of the higher level objects within that image. We propose an invariant feature descriptor that can be used for object categorisation with the bag of visual words model.

The image features are described by the global paths of the rays, so the path of each ray is used as the basis of the feature descriptor. To achieve a representation with invariant descriptive capability, each ray is transformed so as to be equally sampled in distance and the curvature described at different sampling scales, illustrated in figure 5.2. The initial ray shown in (a) is transformed into a ray where the length of each segment between changes in direction is constant in (b). This transform occurs at several scales giving a coarse to fine description of the ray's path. The change in angle between each segment that produces the feature h is shown in (c) at the smallest scale, whilst (d) and (e) show the coarser scales with segments that are the mean of the finer scales. These features are rotationally invariant, being based upon the angle differences between segments. Scale invariance can be achieved by varying the size of rays but normalising all to the same sized feature vector.

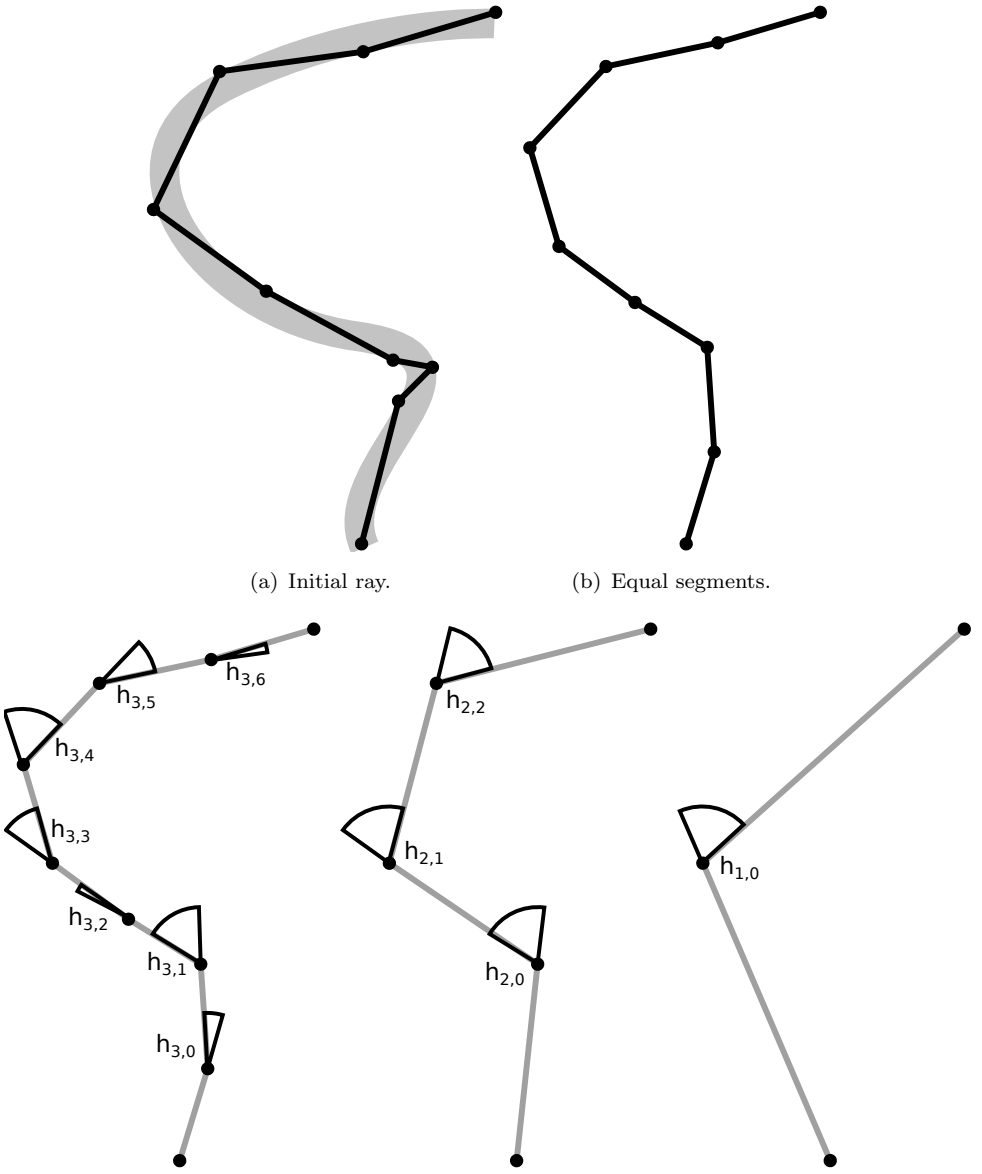
Function 5.1: generateRayFeatures

```
// Cast ray and find mean length of segment
directions, lengths ← castRandomRay(image );
meanLength ← mean(lengths );
// Equalise ray so each segment is of the same length
eqDirections ←equaliseSegments(directions, lengths, meanLength);
// Convert to angle and record this as the finest scale
angles ← atan2(eqDirections );
features ← createFeatures(angles );
```

The pseudocode presented in function 5.1 describes the procedure for generating ray features from a ray, the main steps being the casting of the ray, the equalisation of the distance between direction changes (segments), the conversion to angles and then to invariant features.

The direction of the ray and the length of each segment are recorded as the rays progress through the image. Use of the direction rather than position gives our feature translation invariance. The direction of a segment is then

$$\mathbf{V}_a = (v_x, v_y), \tag{5.1}$$



(a) Initial ray. (b) Equal segments.

(c) h at scale $g = 3$. (d) h at scale $g = 2$. (e) h at scale $g = 1$.

FIGURE 5.2: The calculation of the ray features h (angle changes) from a ray of 8 segments ($d = 8, m = 8$).

where a is the index of the segment and $0 \leq a < d_r$, d_r being the number of segments in that ray (up to a maximum of d). Each segment also has a different length l_a. These segments must be transformed into m segments of length equal to the average, l_q

$$l_q = \frac{1}{m} \sum_{a=0}^{a<d_r} l_a. \tag{5.2}$$

We produce a new direction vector $\mathbf{V}'_b$, where $0 \leq b < m$ and b is the index of the new segment, each segment being a weighted sum of the original ray segments. We define the cumulative lengths at each direction change along the original (λ) and equalised (λ')

rays as

$$\lambda_0, \lambda'_0, = 0, \tag{5.3}$$

$$\lambda_a = \lambda_{a-1} + l_a \qquad 1 \leq a < d_r, \tag{5.4}$$

$$\lambda'_b = \lambda'_{b-1} + l_q \qquad 1 \leq b < m, \tag{5.5}$$

so that λ_0 is the start of the first segment, λ_1 is the end of the first segment and start of the second, and so on. We also define the piecewise function o as

$$o(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \tag{5.6}$$

then we can express the weight $w_{a,b}$ that a segment a of the original ray has on the equalised segment b as

$$w_{a,b} = \frac{1}{l_q} o(l_a - o(\lambda'_b - \lambda_a) - o(\lambda_{a+1} - \lambda'_{b+1})). \tag{5.7}$$

This first calculates the length of the original ray segment that is within the equalised ray segment before calculating what proportion of the equalised ray segment is occupied by the original ray segment. Then the new direction $\mathbf{V}'_b$ of any equalised ray segment is

$$\mathbf{V}'_b = \sum_{a=0}^{a<d_r} w_{a,b} \mathbf{V}_a. \tag{5.8}$$

Function 5.2: equaliseSegments

```
cumulLength[0] ← 0;
// Calculate cumulative lengths along each ray
for a ← 1 to rayDepth do cumulLength[a] ← lengths[a − 1] + cumulLength[a − 1];
for b ← 0 to m do cumulEqLength[a] ← meanLength ∗ b;
// Find the amount each segment contributes to each equalised segment
for b ← 0 to m − 1 do
    eqDirections[b, :] ← 0;
    for a ← 0 to rayDepth − 1 do
        // Length of segment that matches equalised segment
        inSegmentStart ← zeroNegative(cumulEqLength[b]-cumulLength[a]);
        inSegmentEnd ← zeroNegative(cumulEqLength[b + 1]-cumulLength[a + 1]);
        inEqSegment ← zeroNegative(lengths- inSegmentStart - inSegmentEnd);
        // Fraction of this segment in this equalised segment
        segmentWeight[a] ← inEqSegment/meanLength;
        eqDirections[b, :] ← eqDirections[b, :] + segmentWeight ∗ directions[a, :];
    end
end
```

Function 5.2 describes the process of converting a series of segments with a direction

and length into segments with equal lengths and directions, found from a weighted combination of the original ray.

We now convert our vector direction to the angle ϕ using atan2($\mathbf{p}$) to preserve quadrant information,

$$\phi_b = \text{atan2}(\mathbf{V}'_b). \tag{5.9}$$

Our final feature vector is the difference between the angles of adjacent segments, providing rotation invariance. Before that is done the vector can be enhanced by calculating the mean angles at a range of scales. This ensures that rays that may have large differences at fine scales can still match with those that are similar at coarser scales. To ensure that we can calculate up to the coarsest scale we say that m must be equal to a power of 2,

$$\exists g_{\max} \in \mathbb{N} : m = 2^{g_{\max}}. \tag{5.10}$$

We then calculate the mean angles at a range of scales g, where $0 < g \leq g_{\max}$, and the number of angles at that scale is $c_{\max,g} = 2^g$. $\phi_{g,c}$ is the angle at scale g, and c its index where $0 \leq c < c_{\max,g}$. We calculate the largest scale from ϕ:

$$\psi_{g_{\max},b} = \phi_b, \quad 0 \leq b < m. \tag{5.11}$$

The angles at finer scales are calculated from the mean of two values at the previous, coarser, scale:

$$\psi_{g,c} = \frac{1}{2}\left(\psi_{g+1,2c} + \psi_{g+1,2c+1}\right). \tag{5.12}$$

We produce our feature vector from the difference between adjacent angles at the same scale, $0 < e < e_{\max,g}$ where $e_{\max,g} = c_{\max,g} - 1$,

$$h_{g,e} = \psi_{g,e+1} - \psi_{g,e}. \tag{5.13}$$

One further issue presents itself to us, as a consequence of using rays. A ray that travels along a structure in one direction will not have a similar feature vector to one traversing the same feature in the opposite direction. To tackle this problem we ensure that the change in angle at the coarsest scale is positive (or clockwise), transforming the features if this is not true through reversion and negation. So if $h_{1,0} < 0$, we flip and negate h:

$$h_{g,e} = -h_{g,(e_{\max,g}-e)} \quad 0 < g \leq g_{\max} \wedge 0 \leq e < e_{\max,g}. \tag{5.14}$$

The complete process of converting the angles of the equalised segment directions into our rotationally invariant features is shown in function 5.3. This includes the calculation of the angles at a coarser scales by finding the mean of the corresponding angles at the

Function 5.3: createFeatures

```
scaledAngles[gMax, :] ← angles[:];
// Propagate angles to coarser scales
for g ← (gMax − 1) to 1 do
    for c ← 0 to cMax do
        scaledAngles[g, c] ← 0.5 ∗ (scaledAngles[g + 1, 2 ∗ c] + scaledAngles[g + 1, 2 ∗ c + 1]);
    end
    for e ← 0 to eMax do
        features[g, e] ← scaledAngles[g, e + 1] − scaledAngles[g, e];
    end
end
// Invert features if going in wrong direction
if features[1, 0] < 0 then
    for g ← 0 to gMax do
        for e ← 0 to eMax do
            features[g, e] ← −features[g, eMax − e];
        end
    end
end
```

finer scale, the angle difference between adjacent angles, and the inversion of the features if they are oriented in the wrong direction.

The one-dimensional concatenated version of h is our final descriptor representing this ray, and an image is represented by many thousands of such ray descriptors. The size of our feature descriptor, remembering that $2^{g_{\max}} = m$ is then

$$|h| = \sum_{g=1}^{\log_2 m} 2^g - 1 = 2m - (2 + log_2 m). \tag{5.15}$$

Our experiments use a 120 length feature vector generated with $m = 64$.

5.2 Object Categorisation with Ray Descriptors

We now seek to evaluate the the ray descriptors by using them for object categorisation with a bag-of-visual-words model (section 2.5.2) and comparing them with the descriptors described in section 2.5.1.

5.2.1 Categorisation Method

The bag of words model requires that the descriptors be encoded through a vocabulary, in this case generated using the k-means clustering algorithm (section 2.5.2). In order to reduce the computational cost of this calculation we sample the descriptors rather

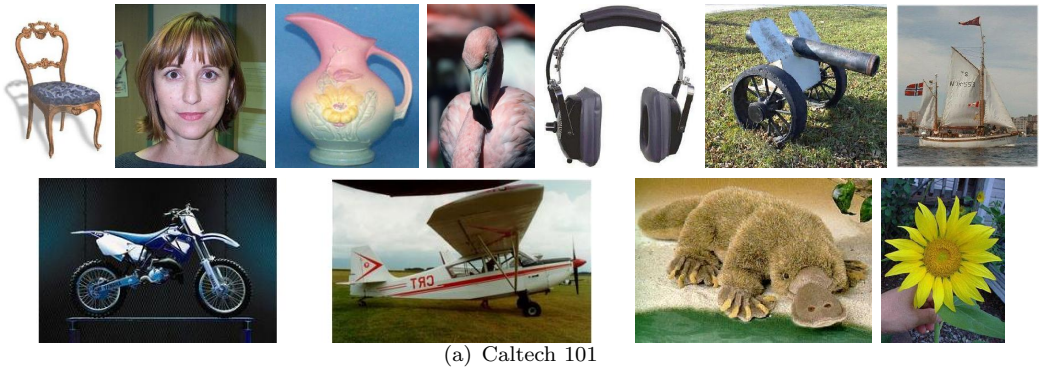
than using all of them, and employ the k-means++ technique [6] to improve the quality of cluster centres. Specifics of the value of k and the data used to cluster depend upon the dataset being used, and are described in sections 5.2.1.1 and 5.2.1.2. In addition to the ray feature descriptors, we compare our results with experiments using alternative feature descriptors: SIFT and the grid of normalised intensity patches, both described in section 2.5.1.

We use a multiclass naïve Bayes classifier [19] to classify each image. This method is simple but effective, as we strive only to show the effectiveness of the ray features in conjunction with other features. Additionally, multiclass classification occurs in a similar manner to single class classification and the model that is built in training can be more easily inspected than with more complex kernel based classifiers. Following the extraction of feature descriptors and the creation of a vocabulary through k-means clustering, the descriptors found in each image are assigned to the closest cluster (via Euclidean distance) and the frequency of each cluster (or word) in each image is calculated. The word counts of each of the training images are then entered into the naïve Bayes classifier (NBC), from which it builds a model of each class. In the NBC used, each word count is modelled as a Gaussian variable with mean and variance calculated from the observed training images' word counts. This model provides a probability or confidence that an image contains an specific class of object and allow us to either pick the most likely class in the single class case with Caltech 101, or apply a threshold in the VOC multiclass case to select the objects that are likely to be present.

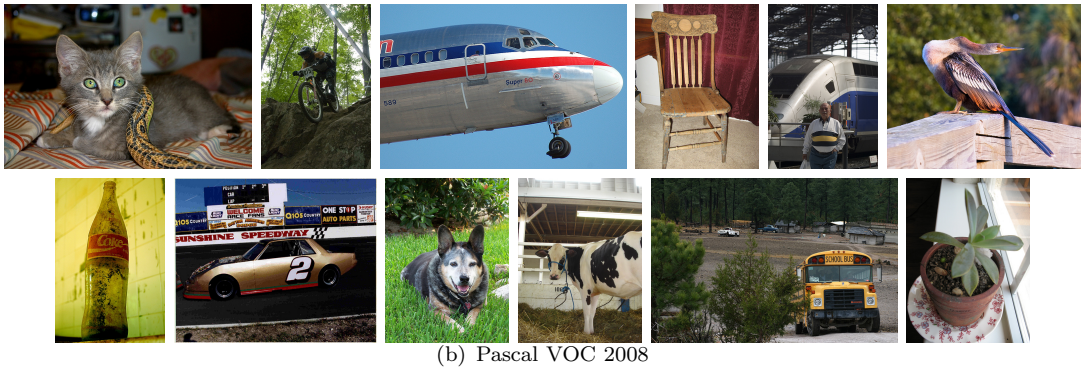
5.2.1.1 Caltech 101

The Caltech 101 database [31] provides 101 categories with between around 30 and 100 images for each (see figure 5.3(a) for examples). The images are generally devoid of clutter, and of a similar scale and rotation. Each image is labelled with a single class, providing a simpler problem than that described in section 5.2.1.2. We perform the image ray transform twice, with $N = 10000$, on the original and inverted image. As the dataset includes little scale variation, we use a constant value of $d = 64$ (and $m = 64$) rather than varying it across a range. The value of $m = 64$ gives a feature vector of similar size (120) to SIFT and the normalised grid intensity descriptors (128 and 121 respectively). Following descriptor extraction, the vocabulary is created by sampling across the entire dataset. A number of values of k were tested from 100 upto 5000, leading to selection of the best values of k for k-means clustering of 2500 for the intensity grid, and 3000 for the ray features and SIFT.

Due to the small number of images in some categories, we trained a naïve Bayes classifier on a varying number (between 1 and 30) of training images and normalise the results for class size. The training images were randomly selected, and we repeated this a number of times (20) to ensure results were consistent.



(a) Caltech 101



(b) Pascal VOC 2008

FIGURE 5.3: Example images from the two datasets used to test object categorisation.

5.2.1.2 Pascal VOC

We also test our descriptor on the Pascal VOC 2008 detection challenge dataset [30]. This dataset provides 20 classes with a smaller variance in category size than present in the Caltech 101 set. The images contain clutter, significant scale variance and many images have more than one category present (examples are shown in figure 5.3(b)). Images are already divided into training, validation and test sets and are used as such. Transform parameters are the same as those used for the Caltech 101 dataset, but due to the large amount of scale variation in the dataset we vary d randomly between 16 and 256 in order to capture features across a range of object scales, although we continue to set $m = 64$. For the k-means clustering stage we tested a range of vocabulary sizes from 100-20000, and found that the best results used $k = 500$ for SIFT, $k = 5000$ for intensity grid and $k = 1000$ for the ray features.

5.2.2 Categorisation Results

Results on the Caltech101 database showed that the IRT could increase descriptive capability. In figure 5.4 we show the successful classification rate, normalised for the size of each class and its test set against the size of the training set. The shaded areas around each line represent the standard error across the 20 random selections

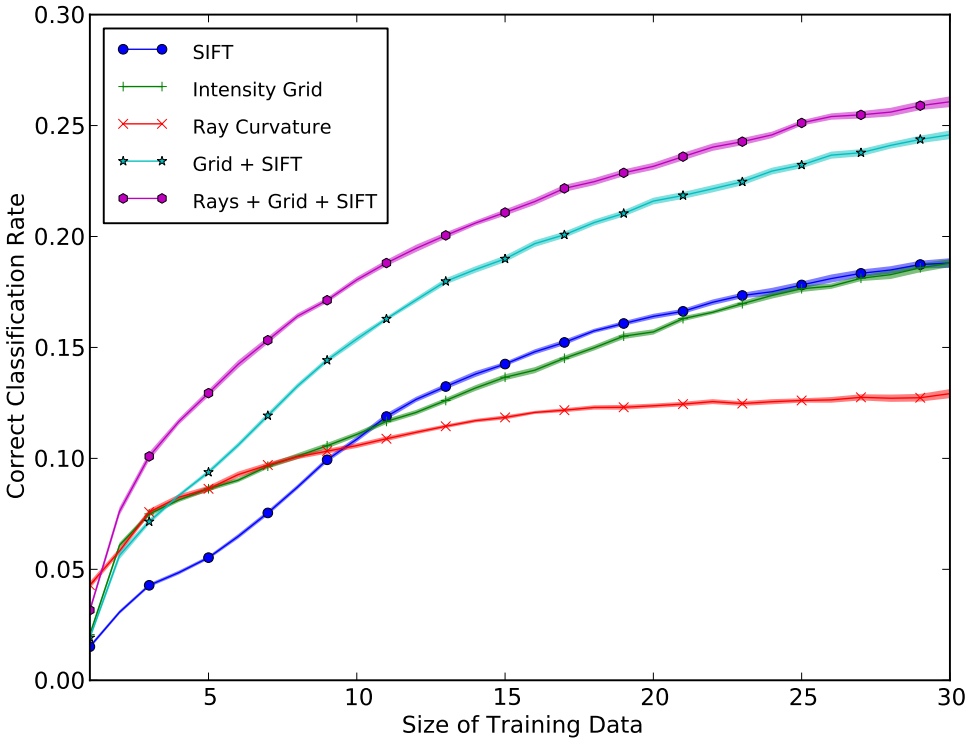


FIGURE 5.4: Correct classification rate against training size on the Caltech 101 dataset.

of training and test sets. It shows that the ray descriptor classifies objects with few training samples with greater success than either SIFT or the intensity patches. With more training examples this advantage is lost, but when used in conjunction with the other features, always leads to an increase in classification rate. Figure 5.5 shows the similarly normalised rank classification with a training size of 5. With this training size the ray features are strongest, and results are improved with the combination of the alternative features.

On the VOC2008 dataset, results suggested that the ray features were superior to the other features that were tested. The VOC challenge recommends the use of (and the submission server provides) average precision to gauge performance, and we use that in addition to precision-recall curves to evaluate our results. The naïve Bayes classifier provides a confidence value that each image contains a certain class, and through variation of that we can analyse the precision and recall of the classifier. The precision of the classifier on a specific class is the number of images that are correctly identified as containing that object class (true positives, TP), over the total number of images identified as containing it (that is, TP plus false positives, FP):

$$precision = \frac{TP}{(TP + FP)}. \tag{5.16}$$

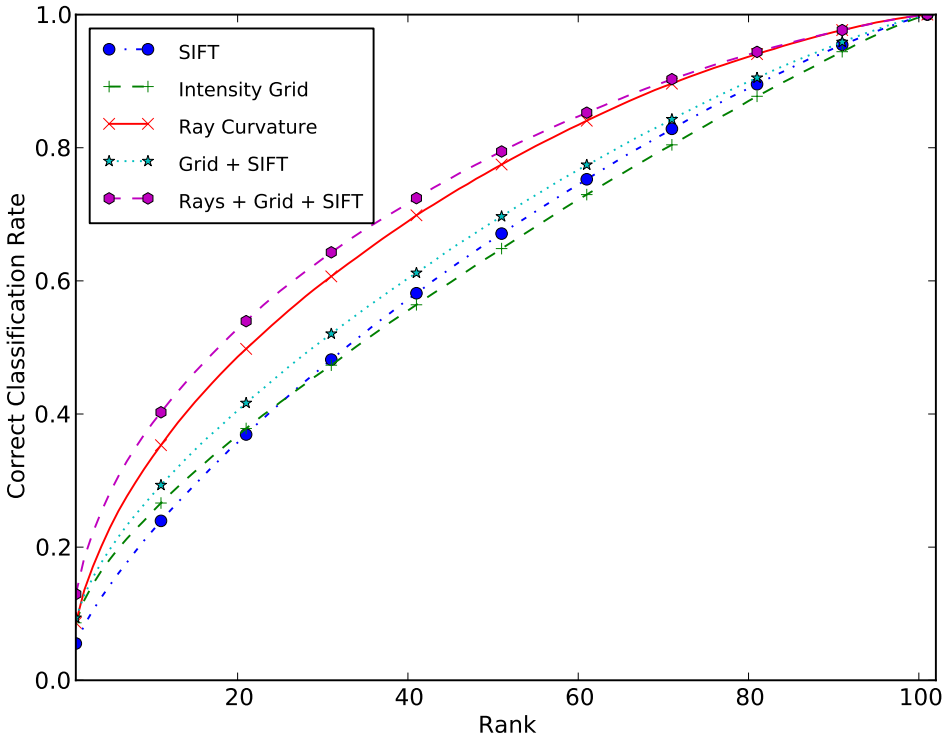


FIGURE 5.5: Rank classification rate on the Caltech 101 dataset with a training size of 5.

Recall measures the proportion of images containing the object that are correctly identified (where FN is the count of false negatives), that is:

$$recall = \frac{TP}{(TP + FN)}. \tag{5.17}$$

By varying the confidence threshold at which we declare an image to contain an object, we can vary the value of precision and recall. Average precision (AP) of a class is a measure that can characterise the relationship between precision and recall. The VOC challenge uses a specific interpolated version of average precision to do this, where the precision achieved at a number of defined levels (with VOC, 11 values between 0 and 1) of recall is calculated and the mean value is found across all recall levels:

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1.0\}} p(r) \tag{5.18}$$

Where $p(r)$ is the precision when recall is r and specific levels of recall are found by adjusting the confidence threshold as required. A final measure used is the Mean Average

Features	**Mean Average Precision**
Ray	0.142
SIFT	0.129
Grid	0.117
Grid + SIFT	0.120
Ray + Grid + SIFT	0.128

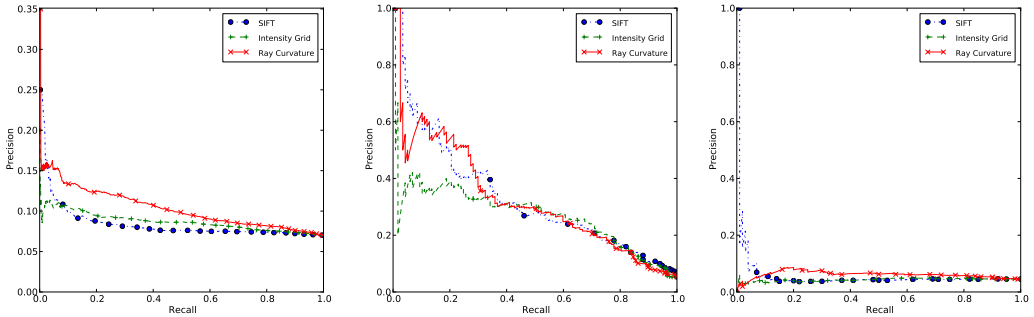
TABLE 5.1: Mean Average Precision for VOC2008 data. Naïve Bayes classifier was trained on training set and tested on validation set.

Precision (MAP), found by averaging AP across all classes (C):

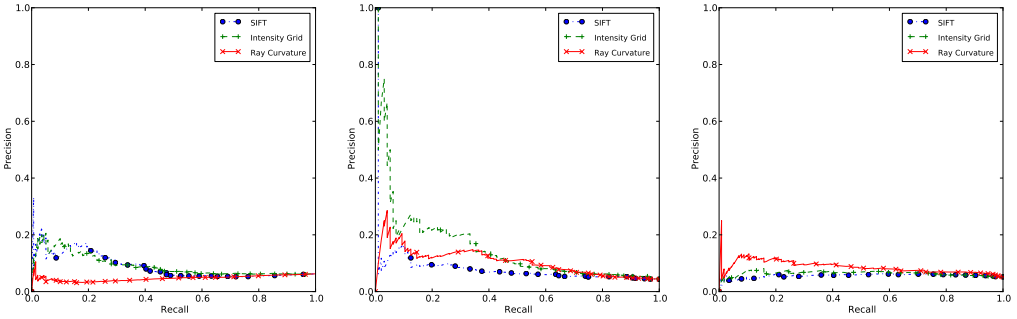
$$MAP = \frac{\sum_{c \in C} AP(c)}{|C|}. \tag{5.19}$$

This provides a single metric by which we can evaluate each technique.

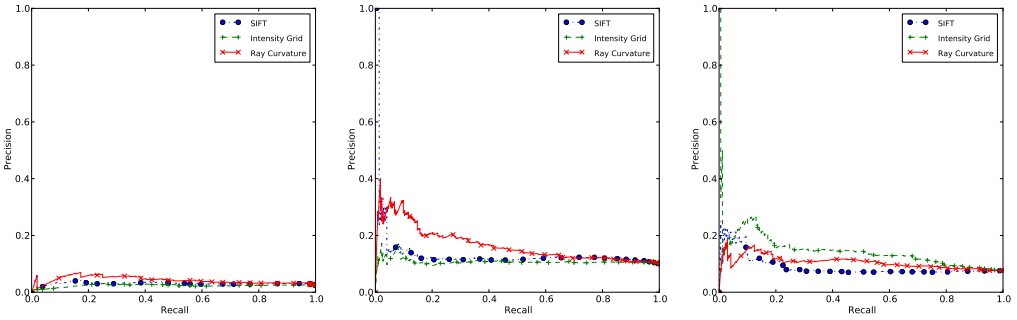
The VOC challenge is a considerably more difficult problem than that provided by Caltech 101 due to the variation in scale, pose and the large amount of clutter present in most of the images. Additionally, many images contain multiple objects, which provide confusing data to the classifier. We provide results for classifiers trained on the training data alone and tested on the validation data, as well as ones trained on both the training and validation data and tested on the test data. Table 5.1 shows the mean average precision (across the 20 classes) when tested on the validation set. From this we see that the ray features provide a 10% improvement in average precision over SIFT, and a 21% improvement over the grid features. Combining features does not have a positive effect, as occurs with the Caltech dataset, primarily due to the overall poorer performance of the alternative features with the VOC data. Figures 5.6 and 5.7 show the precision-recall curves for these results. In figure 5.6(a) we show the mean precision and recall across all twenty classes (not weighted by class frequency). The ray curvature technique has the highest initial precision that is maintained as recall increases. The remainder of figures 5.6 and 5.7 show individual class precision-recall curves and there is great variation between the success of each class. Around half the object categories appear to show little to no success, the classifier having failed to learn useful information about the features most often present in the classes. Whilst the intention of the classifier was to learn the features that related to specific objects, training on uncropped images with backgrounds and other objects may have led to the context also present in the training images to be the focus of the classifier's learning. In cases where the rest of the image is either devoid of features (the aeroplane category is most often surrounded by sky or runway), predictable (the car or motorbike are often pictured on roads), or the object tends to take up a large fraction of the image (the person category often includes close up face images) the classifier is able to concentrate on relevant object features more easily or to use reliable context features to improve the model. The ray descriptors allowed detection of both the motorbike (figure 5.7(c)) and person (figure 5.7(d)) categories with greater accuracy than with either of the other feature descriptors. One explanation of



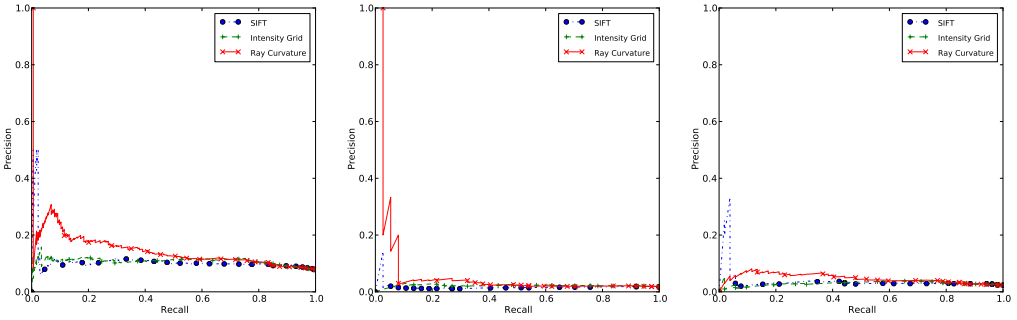
(a) Mean across all classes (b) Aeroplane (c) Bicycle



(d) Bird (e) Boat (f) Bottle

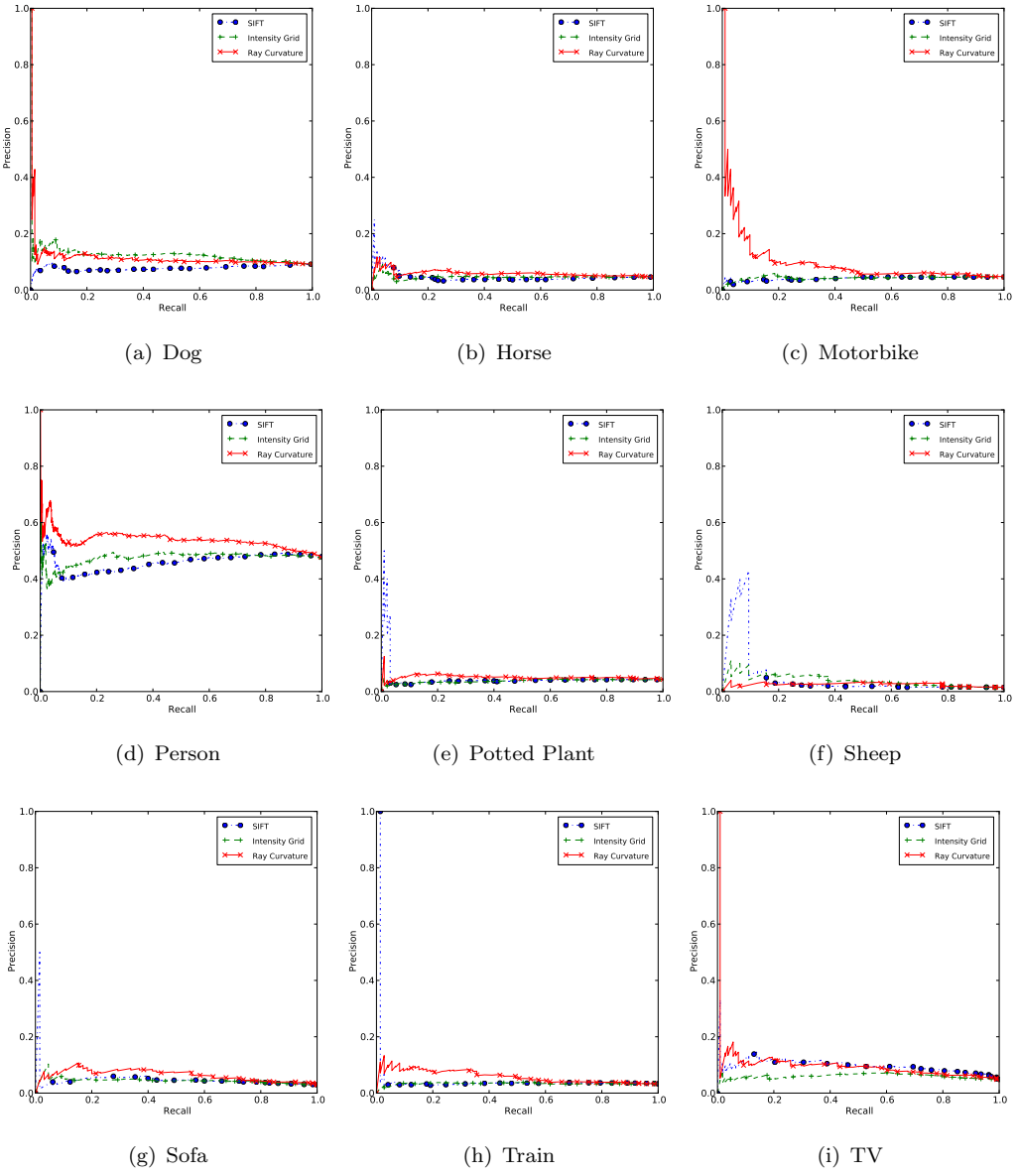


(g) Bus (h) Car (i) Cat



(j) Chair (k) Cow (l) Dining Table

FIGURE 5.6: Precision-recall curves for ranked classification on the validation dataset.



(a) Dog (b) Horse (c) Motorbike

(d) Person (e) Potted Plant (f) Sheep

(g) Sofa (h) Train (i) TV

FIGURE 5.7: More precision-recall curves for ranked classification on the validation dataset.

this would be the prevalence of tubular structures within these object classes (as the framework of the motorbike, and limbs of humans). Another is that the denser sampling performed in the extraction of ray descriptors provided a greater emphasis on reliable context (the prevalence of roads and buildings in motorbike scenes) or images where the object is the focus of the scene (hence larger and not surrounded by other objects, as often occurs with images of people).

Table 5.2 shows results when the classifier was trained on the training and validation set and tested on the test set. Results are not as complete, due to limits caused by using the challenge server for analysis and not having access to the test set labels. It shows

Features	Mean Average Precision
Ray	0.136
SIFT	0.130
Grid + SIFT	0.123
Ray + Grid + SIFT	0.133

TABLE 5.2: Mean Average Precision for VOC2008 data. Naïve Bayes classifier trained on training and validation set, tested on test set

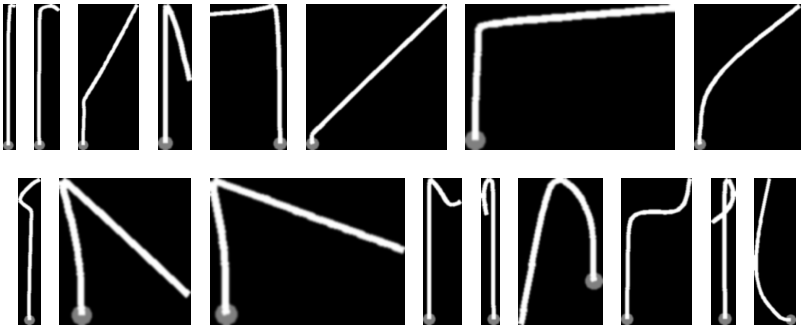


FIGURE 5.8: A visualisation of a selection of ray features selected as the vocabulary for the VOC2008 experiments.

an increase of 8% in the mean average precision with the addition of the ray features to the other features.

Figure 5.8 shows a visual representation of a selection of words from the vocabulary generated for the VOC experiments. The representation is generated by examining only the finest scale information and drawing a straight line segment (beginning at the gray circle), adjusting the direction by the first value at the finest scale and drawing another straight line segment. This continues until all direction changes at the finest scale have been drawn. Examination of the ray words shows most exhibiting one or two sharp changes in direction or a more gradual curve. There are also a large number of ray words that consist of either straight rays or straight rays with a small direction change near their termination. These are a consequence of the mechanics of the IRT, as random rays often propagate for a significant distance without many direction changes until they interact with a structure. Interactions with a structure lead to an increased number of direction changes, but reduce the distance covered between each direction change. Due to the way the ray feature descriptor is constructed the long sections of straight travel are afforded greater weight and any fine detail discovered by the ray after such a journey is not retained.

The ray features provided the greatest boost to performance over other feature descriptors on the motorbike category, and this warrants further discussion. There are number of possible reasons that this category performed so well compared to the other categories. Figure 5.9 shows the twenty most strongly weighted ray words in the motorbike model.

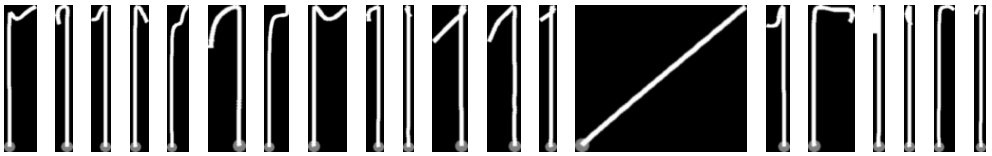


FIGURE 5.9: A visualisation of the twenty most strongly weighted features in the NBC model for the motorbike category.

Examination of them (and other strongly weighted words) shows that rays that are primarily straight are the most useful in successfully categorising motorbikes. Returning to the explanations of the precision-recall results earlier, this suggests that the dense sampling is ensuring that the reliably structureless context around motorbikes (lots of sky and road) is being used for classification in addition to any information from the actual structure of the motorbike.

These results suggest that ray curvature provides valuable features, capable of complementing other features as well as describing objects effectively alone. Analysis also suggests that the ray descriptors would benefit from further refinement. The IRT could be used to select a subset of the rays to be used as descriptors, those that are in areas deemed to be structural features by the transform. There are also a large number of ray descriptors that are primarily straight, with interesting direction change near their end where a structure has been found. A method to remove the straight section of the ray, and concentrate the descriptor on the interesting fraction of the ray would likely provide ray descriptors that reflect the structure in the object to a greater degree.

5.3 Conclusions

The IRT has previously been shown to aid in detection of structural features. We have shown that through analysis of the rays that are used for this detection, invariant features can be produced that can successfully characterise objects within images and be used to enrich object categorisation. The ray descriptor is invariant to translation, rotation and scale, and when used in a bag-of-words model can categorise objects with success. The ray descriptors are particularly noteworthy for complementing other common patch-based features that are often used with the bag-of-words model, working across a much wider scale and on higher level structural features. Across multiple datasets, the ray descriptors work well both alone and in conjunction with other descriptors, enhancing results with few training samples. However, results suggest that much of the ray descriptor's discriminative power may come from the context surrounding objects, and many rays descriptor represent straight rays in structureless image areas and provide little information about the structure of objects.

Chapter 6

Further extensions to the Image Ray Transform

A number of areas of future work have been the subject of preliminary investigation but require further work to bring them beyond theory and towards mature application. Some initial work on adapting the IRT for symmetry has been carried out, showing that rotational symmetry can be exposed through analysis of ray descriptors, and that a variation of the image ray transform can provide a fast, edge focused, reflectional symmetry operator. An enhancement that may be useful in some situations would be targeted initialisations for rays, using application specific prior information to provide the locations for rays to be created. Information about the orientation of structural features could be inferred from the direction of rays, and such a method could be useful for circle or ear detection. A completely different version of the transform could also be created that uses the principles of radiosity to produce an ideal transform, where we examine a distribution of light, rather than individual rays, to eliminate problems of noise at the cost of complexity.

6.1 Rotational Symmetry from Ray Descriptors

The rotational symmetry of an image can be expressed through finding centres of rotation, that is, points around which features of the image have been rotated. Our ray descriptors (as described in chapter 5) provide a method of doing so, with some similarity to that of Loy and Eklundh [54].

As our ray descriptors are rotationally invariant, we can say that any two similar features of differing orientation provide evidence for the existence of a centre of rotational symmetry at a point. Firstly we must decide whether two features are similar. Equation 6.1 shows a measure of difference ($\Delta_{u,v}$) between two rays, u and v, normalised to lie

between 0 and 1,

$$\Delta_{u,v} = \frac{1}{m} \sum_{b=0}^{b<m} \left(\frac{|h_{u_b} - h_{v_b}|}{2\pi} \right), \tag{6.1}$$

where h is the feature vector for the finest scale. We calculate the value of $\Delta_{u,v}$ for each pairwise combination of rays and compare it to a threshold value, ρ, which is set to ensure to select only rays that are sufficiently similar for futher analysis. With the identification of pairs of similar rays, the calculation of their projected centre of rotation can occur using standard equations for rotation. We calculate this value only for the centre points of the two rays, as their similarity makes calculation along all points in the ray redundant. Using the standard 2D rotation matrix where θ is the difference in rotation between the two rays:

$$r = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \tag{6.2}$$

and two variables to simplify calculation,

$$z_x = x_0 - x_1 r_{00} - y_1 r_{01} \tag{6.3}$$

$$z_y = y_0 - x_1 r_{10} - y_1 r_{11}. \tag{6.4}$$

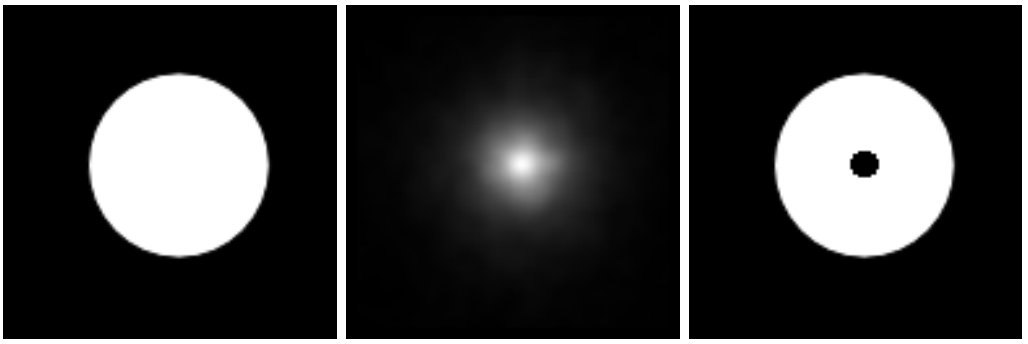
The coordinates of the centre of rotation can then be found as

$$x = \frac{z_x(1 - r_{00}) + z_y r_{01}}{(1 - r_{00})^2 + r_{01}^2} \tag{6.5}$$

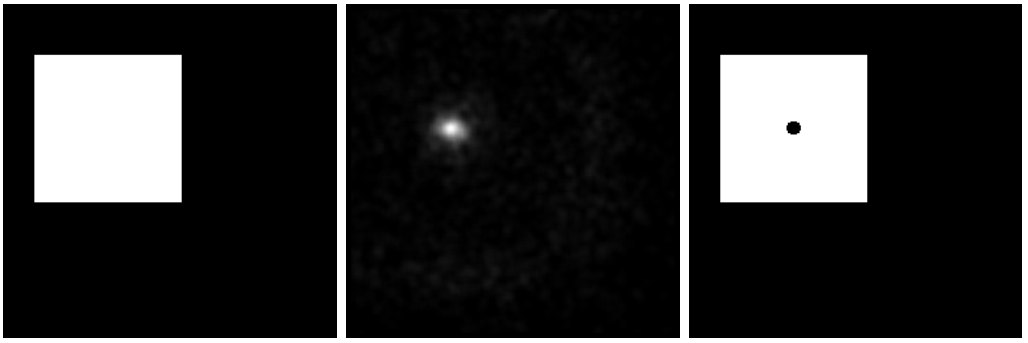
$$y = \frac{x(1 - r_{00}) - z_x}{r_{01}}. \tag{6.6}$$

If this point lies outside the image it is discarded, else the proposed centre is marked in an accumulator. This check for similarity and centre calculation is carried out for all combinations of rays. After all rays have been compared we smooth the accumulator with a Gaussian filter to improve accuracy and take the maximum value as the centre. Figure 6.1 shows three synthetic images with their centre of rotation calculated and their accumulators. Figure 6.1(a) and (b) show results on very simple shapes, whilst (c) is from the dataset of Park et al. [65] and adds a range of intensities to make symmetry detection more challenging.

Results on some natural images from the Park et al. [65] dataset are shown in figure 6.2. Figures 6.2(a), (b) and (c) show the successful detection of the centre of symmetry in both natural and man-made objects. Figure 6.2(d) shows the failure of the technique in an image with a large proportion of blank space. The failure of the symmetry detection in this image is due to the large number of unguided, approximately straight rays covering the blank space. As these rays are all of a similar shape, but rotated, they are treated as prime candidates for use in detection of a centre of rotation, despite not representing



(a)



(b)

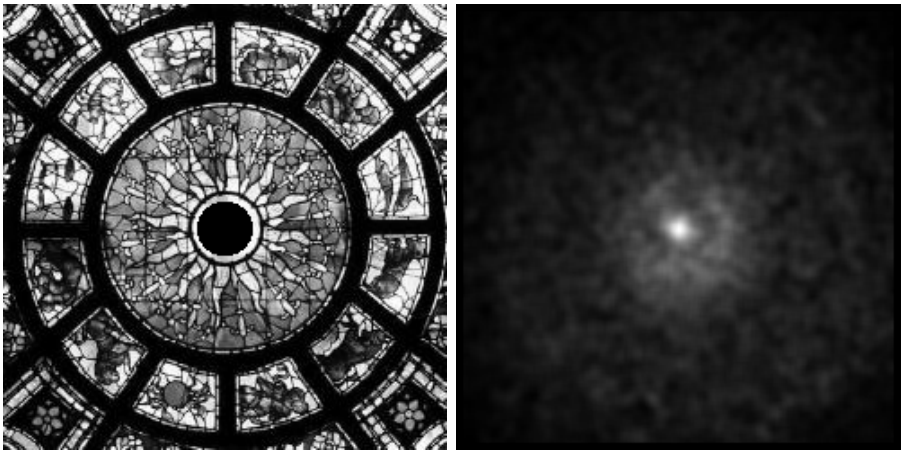
(c)

FIGURE 6.1: Rotational symmetry centres on a selection of synthetic images.

any useful feature. Future work should improve on this case by only comparing rays in areas that have received strong responses from the Image Ray Transform (IRT), areas where it is known that useful structures reside.

It is also feasible to determine the order of symmetry of the detected centres. If we use a histogram to record the most popular size of the angles of rotation between similar features, we can infer the order by certain patterns in the histogram. For example, symmetry of order 3 has peaks at 120° and 240°. It was found that whilst this was possible in some cases, in many cases there were not enough similar rays, especially at higher orders, to reliably fill the histogram and hence determine the peaks and order of symmetry.

(a)



(b)

(c)

(d)

FIGURE 6.2: Rotational symmetry centres and accumulator on a selection of natural images.

6.2 Reflectional Symmetry from the Image Ray Transform

With modification, the IRT can also provide information of the reflectional symmetry within an image. This fast symmetry operator is based upon the generalised symmetry transform (GST) described in section 2.6.

Function 6.1: symmetryImageRayTransform

```
normals ← sobelDirection(edgeImage) + π/2;
refractionMatrix ← analogiseImage(edgeImage, nMax);
symAccumulator[:] ← 0;
for t ← 0 to N do
    ray1, ray2 ← castParallelRays(refractionMatrix, normals, maxDepth,
    maxLength);
    for x1, y1 ∈ ray1 do
        for x2, y2 ∈ ray2 do
            mx ← (x1 + x2)/2;
            my ← (y1 + y2)/2;
            symval ← symmetryFunction(x1,y1,x2,y2,sigma);
            symAccumulator[mx, my] ← symAccumulator[mx, my] + symval;
        end
    end
end
return symAccumulator
```

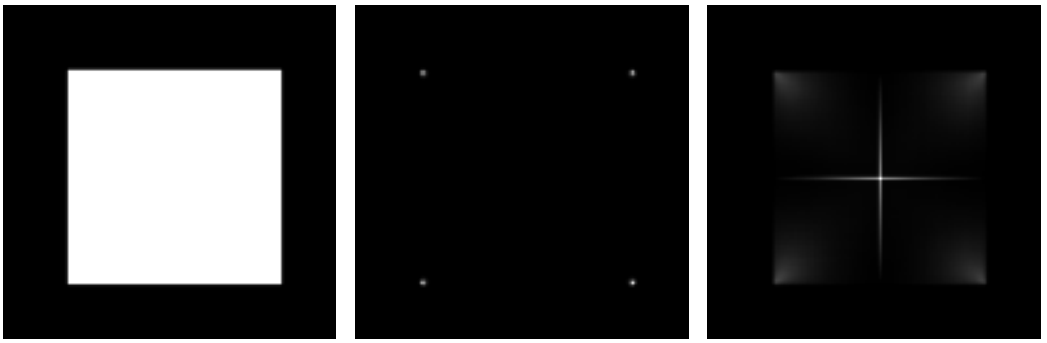
The outline of the technique is shown in function 6.1 and requires the casting of two parallel rays. These rays are focused into edges by performing the transform on the Sobel edge detected image, providing strong tubular structures at edges. The pixels that these rays pass through are then compared for symmetry using the GST symmetry function in equation 2.9, and the result is accumulated. Through this guided sampling of pixels, symmetry can be found with far less computational cost than with the GST, providing a symmetry analysis with a focus upon edges.

A pair of initially parallel rays (u and v) are cast, randomly spaced along a line perpendicular to their direction. As the rays advance, the paths followed by the two rays will be coherent for symmetrical structures and random for those without symmetry. While these rays are being followed, the points through which they pass are added to the sets R_u and R_v,

$$\left\{x \in R | x = \left\lfloor \mathbf{p}^{<t>} \right\rfloor, t \in \mathbb{Z}, 0 \leq t \leq l\right\}, \tag{6.7}$$

and the length limit l is used to put an upper bound on the number of pixels through which the ray can pass, and hence the size of these sets. We then calculate the symmetry by taking every combination of points i and j from R_u and R_v.

$$\left\{(i,j) | R_u \times R_v\right\}. \tag{6.8}$$



(a) Original (b) $\sigma_R = 0.5$ (c) $\sigma_R = 50$

FIGURE 6.3: The application of symmetry ray transform to a 128x128 image of a square with varying values of σ. $N = 10000$, $d = 128$ and $n_{\max} = 40$.

The symmetry between these two points is calculated as in equation 2.9 and is added to the symmetry accumulator $\mathbf{S}$ at position μ, the midpoint of i and j

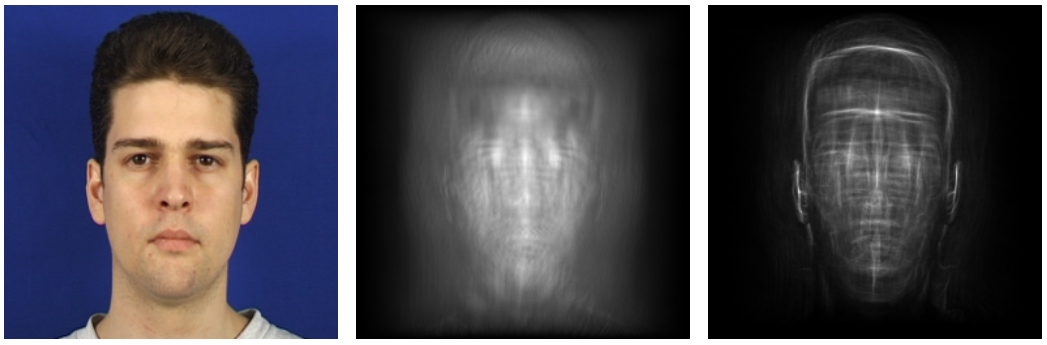
$$\mathbf{S}(\mu_{ij}) = \mathbf{S}(\mu_{ij}) + C(i, j, \sigma_R), \tag{6.9}$$

where σ_R is the scale parameter (as opposed to σ_G in the standard GST). This is repeated for N pairs of rays. The result of this symmetry transform is similar to that of the generalised symmetry transform, as the same features are extracted most strongly in both. Figure 6.3 shows the result of the transform on an image of a square. Figures 6.3(b) and 6.3(c) show the effect of varying the scale parameter σ, and demonstrates how local features such as corners, or global features like axes can be extracted as with the generalised symmetry transform. The ray symmetry transform will concentrate on pixels of high edge strength and edge length more than others, and this can give rise to results that differ from those of the generalised operator. Additionally, the effect that σ_R has on the result differs: values that provide one scale with the generalised operator require larger values to produce a similar scale with the ray implementation. This is due to the use of an accumulator; pairs of rays that are initialised close to each other may follow the same structure, comparing and accumulating from nearby pixels many more times than more distant pixels. The value of σ_R must be increased to a much greater degree in order to counteract this behaviour, and focus on symmetry of a coarser scale.

Figure 6.4 shows the symmetry within a circle according to the ray operator. The edges of the circle are found with small values of σ, whilst with a large value the center is correctly identified. Application to an image of a face provides a better display of the differences between the ray and the generalised version. We tested on an image from the XM2VTS [58] biometric database. The result of the ray symmetry transform differs to that of the generalised symmetry transform. The ray version (figure 6.5(c)) primarily compares strong edges, and so the resultant transform also has strong edges. The generalised symmetry transform operates by comparing every pixel to every other

(a) Original (b) $\sigma_R = 0.5$ (c) $\sigma_R = 50$

FIGURE 6.4: The application of symmetry ray transform to a 128x128 image of a circle with varying values of σ_R. $N = 10000$, $d = 128$ and $n_{\max} = 40$.



(a) Original (b) General. $\sigma_G = 30$ (c) Ray. $\sigma_R = 100$

FIGURE 6.5: Symmetry on a face from the XM2VTS database with different techniques. For 6.5(c) parameters are $N = 50000$, $d = 256$ and $n_{\max} = 40$.

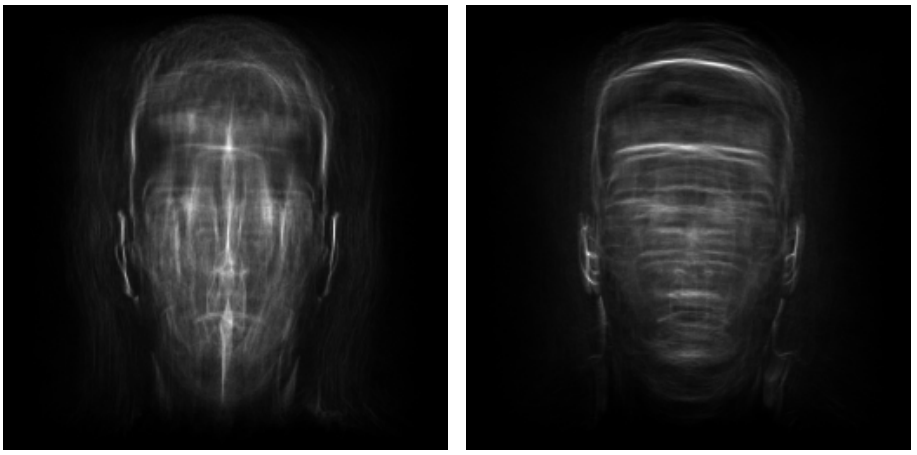
pixel, and so the resultant image 6.5(b) is smoother and the significant lines of symmetry are less clear. The comparison also shows the variation in the scale parameter: the generalised transform produces a result most similar to the ray symmetry operator with $\sigma_R = 100$ when $\sigma_G = 30$.

If we add another term to the phase equation $P(i, j)$ described in equation 2.7, the symmetry in a particular direction, P' can be determined,

$$P'(i, j, c) = P(i, j) \times (1 - \cos(2(\alpha + c))), \tag{6.10}$$

where c is the direction of the plane through which symmetry is to be found; for horizontal symmetry c is 0, and for vertical c is $\frac{\pi}{2}$. The horizontal and vertical symmetry of the face image are shown in figure 6.6.

A large advantage that this transform has over other symmetry transforms is its low computational cost. For a square image, of size $n \times n$ the computational cost of the generalised symmetry transform is $\mathcal{O}(n^4)$. If we assume that l, the maximum number



(a) Horizontal (b) Vertical

FIGURE 6.6: Horizontal and Vertical symmetry within the face image using the ray symmetry transform.$N = 50000$, $d = 256$ and $n_{\max} = 40$ $\sigma_R = 100$.

of pixels that a ray can be recorded going through, is of similar size to n (as usually occurs) then the computational cost of the ray symmetry transform is $\mathcal{O}(Nn^2)$. With a 2.5GHz processor on a 256x256 image for example, the generalised transform took 1700s whilst the ray transform with $l = 256$ took 10-50s for a range of N between 10000 and 50000. This is 0.5% to 3% of the time of the generalised transform. This result also implies that at this size of image N is the dominant parameter in computation time.

6.3 Probabilistic Ray Initialisation

In many cases where little or no prior knowledge about an image is available it is appropriate to initialise rays with uniform probability across the image. However, there may be some cases where more is known about an image and the structure we hope to emphasise, perhaps the general region of a desired object or some interest points. In these cases it may improve the speed and quality of the transform result if we focus rays into these areas. One way of doing this is altering the ray initialisation, basing it upon a defined probability distribution rather than a uniform distribution. This replaces equation 3.4 with:

$$p \sim Z[0, wh),\ \phi \sim U[0, 2\pi), \tag{6.11}$$

where Z is our discrete probability distribution and p is the selected pixel. If, for example, the value of probability distribution at the pixel (0,0) in Z was 0.5, half of all rays would be initialised within that pixel. Equation 6.11 will select only a pixel; to provide a continuous ray initialisation point we should choose a subpixel position in a

similar manner to the fully randomised initialisation:

$$x_s \sim U[0,1),\ y_s \sim U[0,1). \tag{6.12}$$

This leads to final pixel position of:

$$x = x_s + x_p,\ y = y_s + y_p. \tag{6.13}$$

The distribution, Z, could be calculated in a number of ways. Using edge strength or curvature information about the image may be appropriate in some cases whilst the summation of a series of normal distributions in the neighbourhood of detected interest points is another alternative. The selection of Z is strongly application dependent, and provides a way to tailor the IRT, improving results and the speed of the transform.

6.4 Direction From Rays

The result that the IRT produces is a scalar count of the number of rays that pass through each pixel, but information about the orientation of structures could also be derived from the transform. Within structural features, the orientation of rays at any one pixel will tend to be be clustered around two opposing directions, as rays travel one of two ways down the structure. We extend our accumulation function to sum the direction of the ray within a 180° range, inverting the direction if it lies outside

$$\mathbf{A}'_{dir}(\mathbf{p}) = \mathbf{A}_{dir}(\mathbf{p}) + \begin{cases} \mathbf{V} & \text{if } \mathbf{V}_x \geq 0 \\ -\mathbf{V} & \text{if } \mathbf{V}_x < 0. \end{cases} \tag{6.14}$$

When the transform is complete we can then divide this sum by the value of the normal accumulator $\mathbf{A}_{irt}$ at each point to leave the average direction

$$\mathbf{A}'_{dir} = \frac{\mathbf{A}_{dir}}{\mathbf{A}_{irt}}. \tag{6.15}$$

This direction provides information about the orientation of structures, at all points within them, rather than just at the edges as would be provided by the edge orientation. Outside of structures the values are often meaningless, due to the low number of rays and lack of consistent direction in the rays that do pass through such pixels.

6.5 Radiosity Based Image Ray Transform

Radiosity is way of representing the amount of radiation leaving a surface through reflection or emission, and is used in computer graphics as a rendering algorithm. The rendering algorithm works to discover the radiosity at all points in a scene either through

solving equations or iteratively. A version of the IRT could be developed that is inspired by this principle to provide a theoretically more accurate and less noisy result by not using individual rays. In place of individuals rays we consider the overall flow of radiation from one pixel to its surrounding pixels.

A "perfect" IRT implementation could be described as one in which an infinite number of rays initialised at all points and in all directions were followed until their completion. Obviously this is impossible, but with a implementation based upon radiosity an approximation can be made that is closer to this ideal than that provided to us by the standard IRT. This implementation works upon a distribution over all directions of light at each pixel. For this explanation we assume it to be continuous, but any implementation would have to discretise it into a number of bins.

To begin, let us say that at every pixel (p_i) is an initial amount of light, uniformly distributed across all directions: $R(p_i)$ is this distribution. We can then step forward in time and calculate how the distribution of light at each pixel would change. We assume that all the light in a pixel is positioned at the centre, and at each timestep will travel one pixel's width distance, bringing it either to a different pixel (perhaps with a different direction if refracted) or, if reflected, staying in the same pixel with an altered direction. If we define $D(p_i, p_j)$ to be the distribution of rays transferred between pixels p_i and p_j and $N(p_i)$ to be the set of the four neighbours of p_i then the new distribution of rays is

$$R'(p_i) = D(p_i, p_i) + \sum_{p_n \in N(p_i)} D(p_n, p_i). \tag{6.16}$$

The value of $D(p_i, p_i)$ is the the distribution of light that begins to move to a neighbouring pixel but is reflected back into p_i with an altered direction. This reflected direction distribution can be found using a version of equation 3.12 when the angle is above the critical angle of the media boundary. $D(p_n, p_i)$ is the light that successfully moves from p_n to neighbouring pixel p_i, subject to being refracted according to equation 3.10.

The result of the transform in this method could be found in a similar way to the standard IRT, using an accumulator to record the movement of light at each iteration. Alternatively, the system could be iterated over until it is stable; although whether this would occur or not is unknown. The method described above simplifies the problem significantly by not also considering the positional distribution of light across a pixel, as this would add additional, and perhaps unnecessary computational cost. Preliminary work on this version of the IRT suggested that the computational and memory requirements of the method would be extensive, and more simplifications may be required to create a feasible technique. Despite these problems it would provide an interesting insight into the IRT that may not be accessible in any other way.

Chapter 7

Conclusions and Future Work

This thesis has aimed to discover whether we can create a technique to extract tubular structures (as well as other structures) in images by making them analagous to optical fibre through use of an analogy to light. This has led to development of the IRT. In demonstrating the abilities of the IRT we have used it in a range of applications and extended it to give it new capabilities.

7.1 Conclusions

Through the use of an analogy to light, the IRT can be used to detect structural features within an image. Considering an image to be a glass medium and following light rays as they move through it allows us to detect structural features by exploiting their similarity to waveguides such as optical fibres. This property can be used to emphasise such structures or be analysed in order to find higher level features of the image.

To simplify the use of the transform and to widen the range of appropriate applications, a number of enhancements to the basic transform have been created. To reduce the number of parameters and difficulty of parameter selection, an automated stopping condition was created that detects when the transform has converged. This allows images of different scales or structure to have a sufficient number of rays to reduce noise, without increasing computation time unnecessarily. As we do not seek to accurately simulate light, through variation of the method linking intensity and refractive index the power of the transform can be increased. Target intensities allow the transform to detect features of any intensity, and the aggregation of multiple transforms with different target intensities allows all intensities to be detected simultaneously. Through use of an exponential model between intensity and refractive index, the IRT can detect features with minimal local contrast, producing very different results than the linear model. The beam IRT provides a way of improving the ability of the transform to deal with noise

by tethering a number of rays together into a beam and averaging their movement to reduce errant direction changes caused by noise.

The IRT is a technique which has demonstrated an inherent ability to emphasise structural features within images. The properties of structures such as circles and tubes are used by the transform to focus rays into these areas, highlighting them in the transform output. We have shown that by using the IRT as a preprocessor for other techniques we can achieve results that are superior to those without the IRT. In conjunction with edge detection and the Hough transform (HT) for circles, the IRT can improve the accuracy of circle detection with an easily justifiable small cost to computational complexity. A strong technique for the enrolment of ear biometrics can be created by joining the IRT with template matching to highlight, detect, extract and normalise ear images. This automated technique also produces a higher recognition rate with PCA than other automated techniques. The transform has also been used as a preprocessor to segment blood vessels in retinal fundus images, producing good results despite the use of very simple segmentation techniques.

Higher level features can also be found by the IRT through analysis of the paths that rays take through an image. An object can be described by the structures that it contains and structures can be described by rays, if we turn them into invariant features. These invariant ray feature descriptors can successfully be used in a bag-of-visual-words model to categorise images that contain objects. Additionally, these ray features describe objects in a complementary way to other descriptors, often improving results when used in tandem. Symmetry can also be found by analysis of rays. Rotational symmetry can be calculated by using the orientation of similar rays to calculate the centre of rotation, whilst reflectional symmetry can be efficiently calculated through comparing the symmetry of points along two parallel rays.

The IRT is a powerful technique for the detection of structural features as well as higher level features. Enhancements to the transform extend its applicability to new areas and improve performance in different situations, but there are further areas and improvements that can be made. We have shown that the IRT has applications in shape detection, ear biometrics, medical imaging, object categorisation and symmetry, but there are many more possible applications of the transform yet to be discovered.

7.2 Future Work

We have shown the IRT to have a wide range of applications, but various problem domains are ripe for further investigation including applications already described. The IRT itself is well suited to further extension, and some of these extensions will further increase the number of applications for which the transform can be used.

The strength of the IRT in the detection of structural features was shown strongly by its use for enrolment for ear biometrics; however, with additional work these results could be further improved. The thresholding and template matching techniques are purposefully simple to show the strength of the IRT but the enrolment method could be improved by enhancing either of these parts. A local thresholding technique would prevent situations where other structures that respond more strongly than ears (e.g. spectacles) cause the removal of the true ear at this early stage. An elliptical HT would provide a superior detection method to template matching, and would be expected to improve the quality of detection and recognition. Alternatively, the ray descriptors could also be used to detect areas that contain many rays that are commonly found within ears. Our investigation into ear recognition used principal component analysis (PCA), a standard technique, but the IRT could also be used as a basis for a new recognition method, perhaps using fixed ray initialisation points or ray descriptors. Another area in which the IRT may have use would be gait biometrics, detecting the tubular structure of the human body over a spatio-temporal image sequence.

The detection of blood vessels in retinal fundus images via the IRT provided promising results. The results using hysteresis thresholding were good, but a superior method of segmentation is likely to provide a significant increase in detection. An adaptive local thresholding technique would cope with the variance in transformed vessel intensity to a greater degree than our current, mostly global, approach. The aggregation of the two versions of the transformed image using exponential and linear indices provides good results, but a better fusion of the two methods could be found that eliminates noise from both images whilst maintaining salient details. Whilst the aim of this work was to detect tubular blood vessels, it is clear that the IRT also has potential to detect the fovea and the optic disc. This was a challenge in extracting vascular features, but detection of the fovea and optic disc are problems for which computer vision has been applied [61] and in the future the IRT could also be evaluated for its ability to detect these features.

There are many ways that the IRT itself could be extended, providing a greater number of situations in which it can be applied. An obvious extension is to 3D, enabling the use of the transform on 3D images, as well as video. Many medical imaging technologies produce 3D images, and the IRT would be well suited to emphasising vascular features such as blood vessels, as shown with retinal fundus images. 3D also allows the transform to be used upon videos when considered as a 3D spatio-temporal image. This may have uses in object tracking; a ball moving would appear as a 3D tubular structure, prone to being highlighted by the transform. A closer look at termination criteria may also be warranted, stopping rays when they no longer emphasise salient areas rather than when a set limit has been reached. The extensions and variations of the IRT detailed in sections 6.3 to 6.5 would enhance the understanding and utility of the tehcnique if further developed and explored. Through the use of rays the IRT has a design suitable for large speed gains through parallel computing, as each ray can be computed independently,

only needing to be combined in the accumulator. Future implementations should take advantage of this, as well as investigating the use of GPUs in order to use hardware specifically designed for the ray casting around which the transform is constructed.

The method by which we analogise images into glass blocks could be changed to reduce computation times or provide a scale-space interpretation of the transform. If images were segmented prior to the transform being performed, the glass blocks could be polygons covering many pixels, simplifying calculations for rays, and perhaps providing a result with a reduced amount of noise. The principles of scale-space could also be applied to the transform, allowing glass blocks to be size $m \times m$ pixels rather than 1×1, and have properties drawn from the neighbourhood of pixels. Performing the transform with a range of values of m and comparing them would provide an interesting scale-space analysis of structural features within the image.

The work on ray descriptors provided encouraging initial results and further work should examine their use in conjunction with a wider array of feature descriptors and across more, larger datasets. The descriptors themselves may be improved by an alternative representation, one possibility being invariant moments. The current technique also produces a lot of "empty" descriptors, describing areas containing no structural features due to the random initialisation. A future method must find a way of preventing or removing these rays in order to significantly reduce the number of descriptors necessary for categorisation. With extension to 3D and video, these descriptors could also be used in human action or gesture recognition as well as object categorisation. The work on symmetry could also be extended, either using features to find reflectional symmetry as done by Loy and Eklundh [54] or an adaptation of the transform itself to produce rotational symmetry.

References

[1] M. Al-Rawi, M. Qutaishat, and M. Arrar. An improved matched filter for blood vessel detection of digital retinal images. *Computers in Biology and Medicine*, 37 (2):262 – 267, 2007.

[2] K. Allen, N. Joshi, and J.A. Noble. Tramline and np windows estimation for enhanced unsupervised retinal vessel segmentation. In *8th Int'l Symp. on Biomedical Imaging (ISBI'11)*, 2011.

[3] L. Alvarez, E. Gonzalez, and L. Mazorra. Fitting ear contour using an ovoid model. In *39th Int'l Carnahan Conf. on Security Technology (CCST05)*, 2005.

[4] B. Arbab-Zavar and M. Nixon. On shape-mediated enrolment in ear biometrics. In *3rd Int'l Symp. on Visual Computing (ISVC07)*, November 2007.

[5] B. Arbab-Zavar, M. Nixon, and D. Hurley. On model-based analysis of ear biometrics. In *IEEE Int'l Conf. on Biometrics Theory, Applications Systems (BTAS 07)*, September 2007.

[6] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *18th ACM-SIAM Symp. on Discrete Algorithms (SODA'07)*, 2007.

[7] A. Bertillon. *La photographie judiciaire, avec un appendice sur la classification et l'identification anthropometriques*. Gauthier-Villars, 1890.

[8] S. Beucher and C. Lantuejoul. Use of watersheds in contour detection. In *Proc. of Int'l Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation*, 1979.

[9] J. R. Beveridge, D. Bolme, B. A. Draper, and M. Teixeira. The CSU face identification evaluation system. *Machine Vision and Applications*, 16(2):128–138, 2005.

[10] M. Burge and W. Burger. Ear biometrics in computer vision. In *Proc. of 15th Int'l Conf. on Pattern Recognition (ICPR)*, 2000.

[11] J. D. Bustard and M. Nixon. Robust 2D ear registration and recognition based on sift point matching. In *2nd IEEE Int'l Conf. on Biometrics Theory, Applications Systems (BTAS 08)*, September 2008.

[12] J.D. Bustard and M.S. Nixon. 3D morphable model construction for robust ear and face recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2010)*. IEEE, 2010.

[13] O. Carmichael and M. Hebert. Word: Wiry object recognition database. rope.ucdavis.edu/~owenc/word.htm, January 2004. Carnegie Mellon University.

[14] K. Chang, K.W. Bowyer, S. Sarkar, and B. Victor. Comparison and combination of ear and face images in appearance-based biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1160–1165, Sept. 2003.

[15] S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, and M. Goldbaum. Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Transactions on Medical Imaging*, 8(3):263–269, 1989.

[16] H. Chen and B. Bhanu. Human ear recognition in 3D. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):718–737, 2007.

[17] M. Cho and K. M. Lee. Bilateral symmetry detection via symmetry-growing. In *British Machine Vision Conference (BMVC 2009)*, 2009.

[18] M. Chorás. Image feature extraction methods for ear biometrics: A survey. *6th Int'l Conf. on Computer Information Systems and Industrial Management Applications. CISIM '07.*, June 2007.

[19] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual Categorization with Bags of Keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV04*, 2004.

[20] A. H. Cummings and M. S. Nixon. Retinal vessel extraction with the image ray transform. In *6th Int'l Symp. on Visual Computing (ISVC10)*, 2010.

[21] A. H. Cummings, M. S. Nixon, and J. N. Carter. Circle detection using the image ray transform. In *Int'l Conf. Computer Vision Theory and Applications (VISAPP 2010)*, 2010.

[22] A. H. Cummings, M. S. Nixon, and J. N. Carter. A novel ray analogy for enrolment of ear biometrics. In *4th IEEE Int'l Conf. on Biometrics Theory, Applications Systems (BTAS 10)*, 2010.

[23] A. H. Cummings, M. S. Nixon, and J. N. Carter. The image ray transform for structural feature detection. *Pattern Recognition Letters*, 32(15):2053–2060, 2011.

[24] A. H. Cummings, M. S. Nixon, and J. N. Carter. Using features from the image ray transform for object categorisation and symmetry. *Computer Vision and Image Understanding*, In review, 2011.

[25] E. R. Davies. A modified Hough scheme for general circle location. *Pattern Recognition Letters*, 7(1):37–43, 1984.

[26] H. Digabel and C. Lantuejoul. Iterative algorithms. In *Proc. of 2nd European Symp.Quantitative Analysis of Microstructures in Material Science, Biology and Medicine*, pages 85–99, 1977.

[27] C. Direkoğlu and M. S. Nixon. Moving-edge detection via heat flow analogy. *Pattern Recognition Letters*, 32(2):270 – 279, 2011.

[28] C. Direkoğlu and M. S. Nixon. On using an analogy to heat flow for shape extraction. *Pattern Analysis & Applications*, 2011.

[29] J. Dong and Z. Mu. Multi-pose ear recognition based on force field transformation. In *2nd Int'l Symp. on Intelligent Information Technology Application (IITA'08).*, 2008.

[30] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html, 2008.

[31] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106:59–70, 2007.

[32] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2005)*, 2005.

[33] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, 1975.

[34] F. J. Hill. *Computer graphics using OpenGL*, chapter 12, page 678. Prentice Hall, 3rd edition, 2000.

[35] P. Hough. Method and means for recognizing complex patterns. U.S. Patent 3.069.654, 1962.

[36] D. J. Hurley, B. Arbab-Zavar, and M. S. Nixon. The ear as a biometric. In *Handbook of biometrics*. Springer, 2007.

[37] D. J. Hurley, M. S. Nixon, and J. N. Carter. Force field energy functionals for image feature extraction. *Image and Vision Computing*, 20:311–317, 2002.

[38] D. J. Hurley, M. S. Nixon, and J. N. Carter. Force field feature extraction for ear biometrics. *Computer Vision and Image Understanding*, 98:491–512, 2005.

[39] A.V. Iannarelli. *Ear Identification.* Paramont Pub. Co., 1964.

[40] M. Ibrahim, M. Nixon, and S. Mahmoodi. Shaped wavelets for curvilinear structures for ear biometrics. In *6th Int'l Symp. on Visual Computing (ISVC10)*, 2010.

[41] S. Islam, M. Bennamoun, and R. Davies. Fast and fully automatic ear detection using cascaded AdaBoost. In *Proc. of IEEE Workshop on Application of Computer Vision (WACV)*, 2008.

[42] S. Islam, M. Bennamoun, A. Mian, and R. Davies. A fully automatic approach for human recognition from profile images using 2D and 3D ear data. In *Proc. of the 4th Int'l Symp. on 3D Data Processing, Visualization and Transmission (3DPVT 2008)*, 2008.

[43] A.K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[44] A. C. Jalba, M. H. F. Wilkinson, and J.B.T.M. Roerdink. CPM: A deformable model for shape recovery and segmentation based on charged particles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1320–1335, 2004.

[45] X. Jiang and D. Mojon. Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):131 –137, 2003.

[46] I. T. Jolliffe. *Principal Components Analysis.* Springer, New York, 2nd edition, 2002.

[47] Y. Keller and Y. Shkolnisky. A signal processing approach to symmetry detection. *IEEE Transactions on Image Processing*, 15(8):2198 –2207, 2006.

[48] K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Trousset. Model-based detection of tubular structures in 3d images. *Computer Vision and Image Understanding*, 80(2):130–171, 2000.

[49] S. Lee, R.T. Collins, and Y. Liu. Rotation symmetry group detection via frequency analysis of frieze-expansions. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2008)*, 2008.

[50] C. Lemaître, J. Miteran, and J. Matas. Definition of a model-based detector of curvilinear regions. In *12th Int'l Conf. on Computer Analysis of Images and Patterns (CAIP 2007)*, 2007.

[51] X. U. Liu and M. S. Nixon. Water flow based complex feature extraction. *Lecture Notes in Computer Science*, 4179:833–845, 2006.

[52] X. U. Liu and M. S. Nixon. Water flow based vessel detection in retinal images. In *IET Int'l Conf. on Visual Information Engineering*, pages 345–350, 2006.

[53] D.G. Lowe. Object recognition from local scale-invariant features. In *7th IEEE Int'l Conf. on Computer Vision (ICCV'99)*, 1999.

[54] G. Loy and J. Eklundh. Detecting symmetry and symmetric constellations of features. *Lecture Notes in Computer Science*, 3952:508–521, 2006.

[55] S. K. Makrogiannis and N. G. Bourbakis. Motion analysis with application to assistive vision technology. In *16th IEEE Int'l Conf. on Tools with Artificial Intelligence (ICTAI 2004)*, 2004.

[56] S. Manay and A. Yezzi. Anti-geometric diffusion for adaptive thresholding and fast segmentation. *IEEE Transactions on Image Processing*, 12(11):1310 – 1323, 2003.

[57] P. Maragos. PDEs for morphological scale-spaces and eikonal applications. In A. C. Bovik, editor, *The Image and Video Processing Handbook*, chapter 4.16, pages 587–612. Elsevier Academic Press, 2nd edition, 2005.

[58] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTSDB: The extended M2VTS database. In *2nd Int'l Conf. on Audio and Video-based Biometric Person Authentication (AVBPA 1999)*, 1999.

[59] B. Moreno, A. Sanchez, and J.F. Velez. On the use of outer ear images for personal identification in security applications. *Proc. of IEEE 33rd Annual 1999 Int'l Carnahan Conf. on Security Technology.*, 1999.

[60] Z. Mu, L. Yuan, Z. Xu, D. Xi, and S. Qi. Shape and structural feature based ear recognition. *Advances in Biometric Person Authentication*, 3338/2005:663–670, 2005.

[61] M. Niemeijer, M.D. Abrāmoff, and B. van Ginneken. Fast detection of the optic disc and fovea in color fundus photographs. *Medical Image Analysis*, 13(6):859–870, 2009.

[62] M. Niemeijer, J. Staal, B. van Ginneken, M. Loog, and M.D. Abramoff. Comparative study of retinal vessel segmentation methods on a new publicly available database. In *Proc. of SPIE*, volume 5370, page 648, 2004.

[63] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR 06)*, 2006.

[64] M. S. Nixon, X. U. Liu, C. Direkoglu, and D. J. Hurley. On using physical analogies for feature and shape extraction in computer vision. *The Computer Journal*, 54(1): 11–25, 2011.

[65] M. Park, S. Leey, P.C. Cheny, S. Kashyap, A.A. Butty, and Y. Liuy. Performance evaluation of state-of-the-art discrete symmetry detection algorithms. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2008)*, 2008.

[66] P. Perez, A. Blake, and M. Gangnet. Jetstream: Probabilistic contour extraction with particles. In *Proc. Int'l Conf. on Computer Vision (ICCV 2001)*, 2001.

[67] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[68] P. Peterson. F2PY: a tool for connecting Fortran and Python programs. *International Journal of Computational Science and Engineering*, 4(4):296–305, 2009.

[69] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[70] Y. Pnueli and A.M. Bruckstein. Digidürer - a digital engraving system. *The Visual Computer*, 10(5):277–292, 1994.

[71] S. Prakash, U. Jayaraman, and P. Gupta. A skin-color and template based technique for automatic ear detection. In *7th Int'l Conf. on Advances in Pattern Recognition (ICAPR'09)*, 2009.

[72] V. Prasad and L.S. Davis. Detecting rotational symmetries. In *Proc. of the 10th IEEE Int'l Conf. on Computer Vision*, 2005.

[73] G. S. Ramlugun, V. K. Nagarajan, and C. Chakraborty. Small retinal vessels extraction towards proliferative diabetic retinopathy screening. *Expert Systems with Applications*, 2011.

[74] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context-free attentional operators: the generalized symmetry transform. *International Journal of Computer Vision*, 14(2): 119–130, 1995.

[75] C.E. Shannon and W. Weaver. A mathematical theory of communication. *Bell System Technical Journal*, 27(623):52, 1948.

[76] J.V.B. Soares, J.J.G. Leandro, R.M. Cesar Jr, H.F. Jelinek, and M.J. Cree. Retinal vessel segmentation using the 2-D morlet wavelet and supervised classification. *IEEE Transactions on Medical Imaging*, 25(9):1214–1222, 2006.

[77] J. J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken. Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*, 23(4):501–509, 2004.

[78] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[79] B. Victor, K. Bowyer, and S. Sarkar. An evaluation of face and ear biometrics. *International Conference on Pattern Recognition (ICPR)*, 2002.

[80] X. Xie and M. Mirmehdi. MAC: Magnetostatic active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):632–647, 2008.

[81] W. S. Yambor, B. A. Draper, and J. R. Beveridge. Analysing PCA-based face recognition algorithms: Eigenvector selection and distance measures. In *Empirical Evaluation Methods in Computer Vision*. World Scientific, 2002.

[82] P. Yan and K. W. Bowyer. Empirical evaluation of advanced ear biometrics. *Computer Vision and Pattern Recognition Workshop*, 2005.

[83] P. Yan and K.W. Bowyer. Biometric recognition using 3D ear shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1297–1308, 2007.

[84] H. K. Yuen, J. Princen, J. Illingworth, and J. Kittler. Comparative study of Hough transform methods for circle finding. *Image and Vision Computing*, 8(1):71–77, 1990.

[85] F. Zana and J.C. Klein. Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. *IEEE Transactions on Image Processing*, 10 (7):1010–1019, 2001.

[86] B. Zhang, L. Zhang, L. Zhang, and F. Karray. Retinal vessel extraction by matched filter with first-order derivative of Gaussian. *Computers in Biology and Medicine*, 40:438–445, 2010.

[87] H. J. Zhang, Z. C. Mu, W. Qu, L. M. Liu, and C. Y. Zhang. A novel approach for ear recognition based on ica and rbf network. In *Proc. of Int'l Conf. on Machine Learning and Cybernetics*, volume 7, 2005.

[88] L. Zhu, A.B. Rao, and A. Zhang. Theory of keyblock-based image retrieval. *ACM Transactions on Information Systems (TOIS)*, 20(2):224–257, 2002.