Tina Memo No. 2006-002
Internal Report

A Statistical Framework for Detection of Connected Features

Sasirekha Palaniswamy, Neil A Thacker and Chris P Klingenberg

Last updated
15/12/2005

This document forms part of the **Features and Measurement** Series available from www.tina-vision.net.

2002-005	Tutorial: An Empirical Design Methodology for the Construction of Machine Vision Systems.
1996-001	Algorithmic Modelling for Performance Evaluation.
2006-002	A Statistical Framework for Detection of Connected Features.
2006-007	Quantitative Verification of Projected Views Using a Power Law Model of Feature Detection.
1997-003	Tutorial: Supervised Neural Networks in Machine Vision.
1995-003	Invariance Network Architecture.
1992-001	Combining the Opinions of Several Early Vision Modules using a Multi-Layer Perceptron.
2005-006	Curve Fitting and Image Potentials: A Unification within the Likelihood Framework.
1996-002	Tutorial: The Likelihood Interpretation of the Kalman Filter.
1994-003	Using a Switchable Model Kalman Filter.
2004-012	Tutorial: Computing 2D and 3D Optical Flow.
2005-011	Comparing the Performance of Least-Squares Estimators: when is GTLS Better than LS?
1994-001	Tutorial: Overview of Stereo Matching Research.
1992-002	Online Stereo Camera Calibration.
1995-002	Calibrating a 4 DOF Stereo Head.
2000-009	An Evaluation of the Performance of RANSAC Algorithms for Stereo Camera Calibration.
2001-011	The Evolution of the TINA Stereo Vision Sub-System.
2007-011	A Methodology for Constructing View-Dependent Wireframe Models.

WWW.TINA-VISION.NET

Imaging Science and Biomedical Engineering,
School of Cancer and Imaging Sciences,
University of Manchester, Stopford Building,
Oxford Road, Manchester M13 9PT, U.K.

A Statistical Framework for Detection of Connected Features

Sasirekha Palaniswamy, Neil A Thacker and Chris P Klingenberg
Imaging Science and Biomedical Engineering,
School of Cancer and Imaging Sciences,
University of Manchester, Stopford Building,
Oxford Road, Manchester M13 9PT, U.K.
neil.thacker@manchester.ac.uk

Abstract

This document provides a general idea of what edge-detection is and how it works e.g. for computer vision etc., edge detectors are often operated with arbitrary parameters such as thresholds. Determining the significant values for these parameters on a trial and error basis may be a problem. It is therefore beneficial to try to understand edge-detection in terms of established quantitative methods. Here we show how the idea of an Hypothesis test can be used for significance testing and that provided there is the same null hypothesis distribution everywhere in an image, applying Hypothesis testing is the same as thresholding. We show how the method of error propagation can be used to find out if we have uniform noise on a feature enhancement. We apply this analysis to the Canny algorithm for detection of step edges.

We explain that for other than step edges this algorithm needs modification and how this can be done while staying within the overall framework for the detection of connected features via non-maximal suppression and hysteresis thresholding. The DoG is a linear filter which has the required properties for algorithmic stability, and can be used for the detection of ridge structures. The orientation of the ridge is defined for this process as the direction of maximum second derivative. This ridge detector is then evaluated for the task of fly wing analysis, by looking at the specific characteristics of noise and scale stability.

Background

Edges are image attributes that are useful in image analysis and classification. The edge is widely defined to be an abrupt change of grey level values and its location is identified as the midpoint of the edge slope [1]. The sharp changes in intensity are important as they correspond to illumination changes such as shadows and illumination gradients or changes in orientation, distance from the viewer or surface reflectance, material, or to object boundaries [2]. Edges can also be perceived in regions with different optical characteristics such as contrast, colour and texture. The subjective approach of edge definition and characterisation and its wide range of applications have given rise to the development of large number of edge detectors that vary in performance [1].

Edge detection plays a significant role in object recognition and shape analysis. It simplifies image processing & analysis by drastically reducing the volume of the data to be processed whilst preserving useful and important structural information in the image. Essential information in the image is preserved in the edge map of the image and edge structures have an apparent relevance in the biological systems according to Marr [2]. Vernon [3] states that an image comprised of boundaries alone is a higher-level representation of the scene than the original grey level image. Moreover, edge information in an image tends to be robust to a certain extent under varying illumination or related camera parameters. Edge structures are considered primitive features and used widely in computational vision in areas such as scene segmentation, motion detection etc., There is an assumption that the edges detected have some physical significance [4]. The theoretical basis for the concept of an edge is best described by the mathematical concept of "diffeomorphic equivalence". The basic idea is that, as the process of image formation can be described by a continuous function (e.g. an optical model) of the underlying scene structure, differential discontinuities in the scene are preserved in the image. The extracted boundaries are useful in defining the location and shape of the features in the image. Object recognition becomes highly feasible if the boundary of an object can be traced successfully [5].

One major consideration of the behaviour of any feature detection algorithm is how well it performs in the presence of noise. In particular, is it capable of extracting the required image features for a wide range of images. Edge

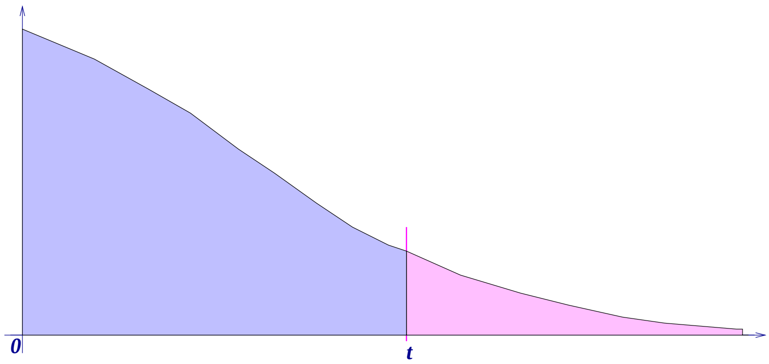


Figure 1: Gaussian distribution and thresholding

detection is conventionally carried out after filtering the image using a filter optimised for particular characteristics. These often aid in filtering the noise, however, this will also blur the edges since edges mostly comprise high spatial frequencies [5].

Edge Detection Algorithms

Most edge detection algorithms involve three steps:

- the image is convolved with a feature enhancement mask (for step edge detection this may be a gradient calculation);
- the pixel positions with enhanced response greater than certain threshold level are labelled as feature tokens, eliminating features that are most likely caused by noise;
- for connected feature structures, linking is performed to ensure edge continuity by bridging the gaps of the isolated fragmented edges.

Hypothesis Testing

The first two stages in a feature detection process, thresholding of a response from a feature (or edge) enhancement, can be related to conventional methods for hypothesis testing. We could attempt to construct a decision system based upon Bayes theorem, but this could never be properly characterised for arbitrary scenes[1]. However, as we generally know something regarding the sensitivity of our measurement system (image noise), we can compute the equivalent of a null hypothesis for the detected feature being entirely consistent with the noise processes. This has a fully quantitative, and therefore testable, interpretation based upon noise rejection.

We define the image content consistent with the null hypothesis as any theoretical 2D structure which would generate a zero response for noise free data, that is any feature response being detected is just noise. In order to prove otherwise i.e. detect the feature of interest, there should be sufficient evidence to reject the null hypothesis.

To test the null hypothesis we calculate the probability of the noise distribution for a feature detection response greater than the particular value of z i.e., say t, then it can be denoted by

$$P(z > t) = \int_t^{\infty} p(z)dz \tag{1}$$

Testing the probability of the null hypothesis at a particular value is therefore equivalent to thresholding the feature response (or any monotonic function thereof) at that particular value. If "t" represents the threshold value, by thresholding the distribution by "t", the presence or absence of significant features can be determined.

The threshold value, "t" will apply a consistent hypothesis test to the entire image provided that the effects of noise in the input image produce a spatially consistent distribution in the enhancement image. This can be tested by error propagation (described below) to ensure that noise is uniform throughout the feature enhancement image.

[1] We cannot construct a meaningful quantitative test based directly upon the presence of the feature, as we do not know a-priori the characteristics of any features present.

Further, the results of this analysis would tell us how the threshold should vary across the image, should the enhancement image not have uniform noise characteristics so that the applied threshold always results in the same level of noise rejection.

This framework potentially allows the construction of a feature detector for any form of local image structure and therefore provides very little constraint on the definition of what we would like to detect. In fact if we take the view that the feature enhancement stage should be considered as a matched filter, then we know that the best detector for any image structure must be based upon the structure we are looking for. Therefore, we cannot expect to have one single optimal edge or feature detector which is best for the detection of all edges under all situations. However, the hypothesis test interpretation does tell us that well behaved algorithms will have enhancement stages which respond with spatial consistency in the presence of noise. Such a behaviour will naturally result in a "robust" response to image data.

Error Propagation

Using the concept of error propagation the uniformity of noise in the image and its effect in the output image can be estimated. The first order analysis provides an understanding of the effects of noise, and provides a necessary but not sufficient condition for algorithm stability.

For any function, y= f(x), a small change δ_x in x, can cause a change δ_y in the function f(x). This change in the function, δ_y can be estimated by taking the partial derivatives of the function, f(x) with respect to x and considering the small change δ_x

$$\delta_y = \frac{dy}{dx}\delta_x \tag{2}$$

Thus, any change in the function, that depends on a variable x, can be calculated from the amount of change on the variable x itself. For the specific case of convolution of an image, the output image can be represented as

$$f_{(x,y)} = \sum_m \sum_n G_{mn} I_{(x+m,y+n)} \tag{3}$$

Any change σ in the input image I, will cause a change δf in the output image f. Therefore, the change in the output image can be calculated from estimates of the variance in the input image. For uniform independent random noise this can be represented as

$$\delta f^2_{(xy)} = \sum_m \sum_n G^2_{mn} \sigma^2_{(x+m,y+n)} \tag{4}$$

In the above equation the effects of the parameters of the convolution kernel can be summarised by a constant, K

$$\sum_m \sum_n G^2_{mn} = K \tag{5}$$

Therefore we can re-write the changes in the output image as below:

$$\delta f^2_{(xy)} = K.\sigma^2_{(x+m,y+n)} \tag{6}$$

This shows that any change in the output image is directly proportional to the changes in the input image and spatially non-varying irrespective of the specific convolution. Therefore, performing Gaussian convolution or a linear operator such as the Difference of Gaussian filter (described below), does not introduce any arbitrary spatial dependency on the noise process in the output image.

Canny Edge Detector

Edges are points where the magnitude of the first derivative of the images 'surface' is a local maximum ie., an edge is present in the smoothed intensity function, f(x) if $\left|\frac{df}{dx}\right|$ is a maximum [4]. Since grey images are two dimensional, edges have a direction as well as magnitude. Therefore, the edge detector should be capable of identifying edges occurring at different orientations in the image. In two dimensional images, edges are considered as linear features at which the magnitude of the first derivative along the gradient direction is maximum [4]. This is equivalent to taking the zero-crossings of the second derivative but numerically more reliable.

The Canny algorithm involved the following important steps:

- The image is blurred with a two-dimensional smoothing kernel to reduce the effects of noise.
- The intensity gradient, magnitude & orientation of the gradient direction are detected in the smoothed image $f(x, y)$. They can be obtained as below:

  The gradient,

$$f = \left[\frac{df(x,y)}{dx}, \frac{df(x,y)}{dy}\right] \tag{7}$$

  In the Canny edge detector, features are located over local maxima in the magnitude of the gradient vector,

$$mag(f) = \sqrt{\left(\frac{df(x,y)}{dx}\right)^2 + \left(\frac{df(x,y)}{dy}\right)^2} \tag{8}$$

  It can be shown, again by the use of error propagation, that this calculation will give rise to a spatially uniform noise distribution and is therefore a suitable basis for a statistically stable feature enhancement.

  The gradient direction is the direction in the image for which the gradient changes more rapidly. It is denoted by θ, and can be calculated as follows in case of a step edge:

$$\theta = tan^{-1}\left[\frac{\left(\frac{df(x,y)}{dy}\right)}{\left(\frac{df(x,y)}{dx}\right)}\right] \tag{9}$$

- Non-maximal suppression is carried out in the direction perpendicular to the estimated orientation so as to retain only the edges with maximum gradient values.
- Adaptive thresholding with hysteresis is performed to eliminate noise & streaking of edge contours. This makes provisions for retaining true edges with fluctuating intensity values.

The Canny edge detector is based on optimising three important criteria that are essential in the process of edge detection:

- good detection i.e. the probability should be low for a) not detecting real edges b) detecting false edges;
- good localisation i.e. the detected edges should be near to the centre of the true edges;
- good response i.e. there should be only one true response to a single edge.

The edge detector proposed by Canny had a similar shape to that of a first derivative of a Gaussian and seems ideal for detecting step edges. The edge points in the images were detected by finding the maxima in the first derivative [6].

Interestingly, although Canny used the criteria above to adjust some aspects of the step-edge detection process, such as the convolution kernel, the noise characteristics of the algorithm were already guaranteed to be stable once the basic framework had been specified. In the context of our interpretation of feature detectors as hypothesis tests, the optimisation process did not affect the noise sensitivity of the algorithm, only the specific definition of the shapes of features corresponding to a null hypothesis[2]. A large part of the success of the Canny edge detector should be attributed to the design of the algorithm and not the criteria selected for optimisation. We should therefore be able to re-use the basic framework for the detection of connected features provided we make suitable replacements for the feature enhancement and orientation calculation.

Methods

Ridges The fly wing has a series of wing veins which appear in an images as ridge features and the optimal detection filter would be one that is similar in profile to these ridges but gives a zero response for non feature regions (such as uniform or gradually varying background). This would be consistent with application of a threshold as a hypothesis test while giving maximal response of the edge detection algorithm to wing structure. The shape of the wing profile inside the wing is shown in Fig. 2(b) and it would be ideal to use a filter similar in shape to that of the wing profile, such as a Gaussian. However, the different wing profiles, at the wing edges, can only be accommodated with a single detector if we reduce the significance of image data away from the centre of the

[2] In this respect, a simple gradient based enhancement stage does not give exactly zero response in smoothly graded regions.

ridge. The shape of wing profiles combined with the need for some kind of radial weighting in the detection process can be interpreted as a Likelihood estimate for the scale of local image structure (Appendix A). This interpretation appears to suggest that there may be some problem with specificity of detection for the process of enhancement using a convolution matched kernel, perhaps leading to poor false detection rates (FDR) when analysing complicated scenes. However, images of fly wings do not have much in terms of complicated images structure beyond the wing ridges and uniform regions. Radial weighting also allows us to define a zero response for uniform background regions by providing a finite extent to any negative portions of the template. Therefore, the Gaussian filter is slightly modified/transformed in shape as shown in Fig. 2(e). This shape of the filter can be achieved by taking the difference of two Gaussian filters. There are several edge detection algorithms that use the DoG due to the correlation between its response and the measured receptive fields of both retinal ganglion and lateral geniculate cells [7]. These filters also have the advantage that they are radially symmetrical and will therefore enhance features such as ridges at all orientations equally. This prevents the need to have to apply feature detection at multiple orientations, in the same way that a sum of squared derivatives enhances step edges at all orientations in the Canny edge detector. We can therefore detect enhanced ridges first and calculate orientations, needed to determine connectivity, later.

Ridge Canny Algorithm The Ridge Canny algorithm consists of an enhancement of ridge features using a Difference of Gaussian operation followed by the non-maximal suppression and hysteresis thresholding steps described above.

In detail, Gaussian filtering is defined for a two dimensional image ($f(x, y)$) as

$$h(x, y) = \sum_k \sum_l f(x - k, y - l)G(k, l) \tag{10}$$

The Gaussian function is represented as

$$G_\sigma(k, l) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{k^2 + l^2}{2\sigma^2}\right) \tag{11}$$

We consider $\sigma > 1$ pixel because Gaussian filtering with $\sigma < 1$ has few effects on suppressing noise [8].

Two Gaussian smoothed images with varying σ_a & σ_b are obtained and they are subtracted from one another to obtain the difference of Gaussian image (DoG),

$$DoG = h_{\sigma_{ab}}(x, y) = h_{\sigma_a}(x, y) - h_{\sigma_b}(x, y) \tag{12}$$

where, σ_a & σ_b are the respective standard deviations of the Gaussian convolutions which in this algorithm are always taken to be in the ratio 1:2. Following the analysis presented in Appendix A, a DoG filter with convolution widths differing by a factor of two ($G_\sigma - G_{2\sigma}$), can also be written as

$$\frac{exp(-r^2/8\sigma^2)}{\sqrt{2\pi}2\sigma}(2exp(-3r^2/8\sigma^2) - 1)$$

Corresponding to a radial Gaussian weighting of width 2σ and an offset Gaussian template t' of width $\sqrt{4/3}\sigma$. This enhancement stage thus contains the features required to produce a locally maximal response when centred over fly wing features as described in the previous section.

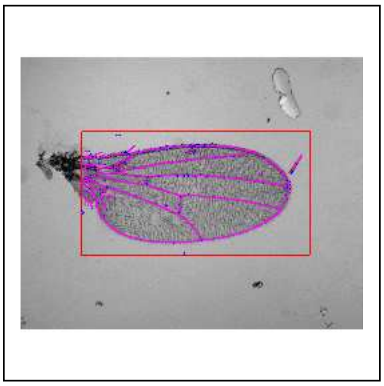
Gaussian smoothing suppresses only high frequency spatial information and subtracting the smoothed images from one another preserves the spatial information that lies between the range of frequencies existing in the smoothed images. Therefore, most of the high spatial frequencies are discarded which also includes the random noise.

Since the edges present in the fly wing images are ridges, the orientation of the gradient direction cannot be calculated using the first derivative of the filtered image. Instead, it is calculated using the Hessian matrix of the difference of Gaussian image. The Hessian matrix is constructed from the partial derivatives of this image [9].

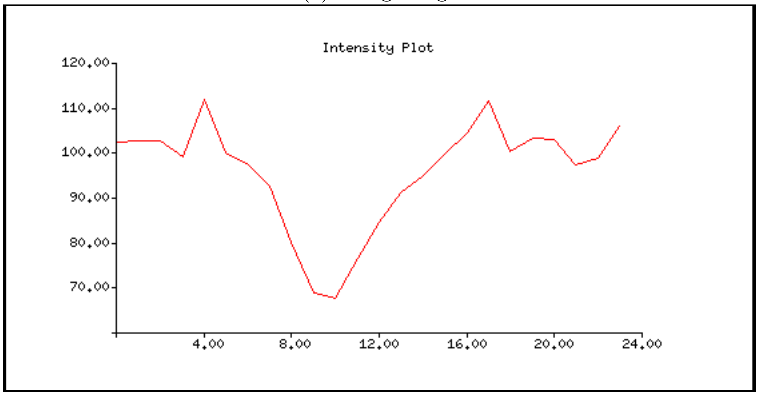
$$H_\sigma(x, y) = \begin{bmatrix} \left(\frac{d^2 h_{\sigma_{ab}}}{dx^2}\right) & \left(\frac{d^2 h_{\sigma_{ab}}}{dxdy}\right) \\ \left(\frac{d^2 h_{\sigma_{ab}}}{dydx}\right) & \left(\frac{d^2 h_{\sigma_{ab}}}{dy^2}\right) \end{bmatrix} \tag{13}$$

The θ value corresponding to the maximum second derivative can be computed as follows:

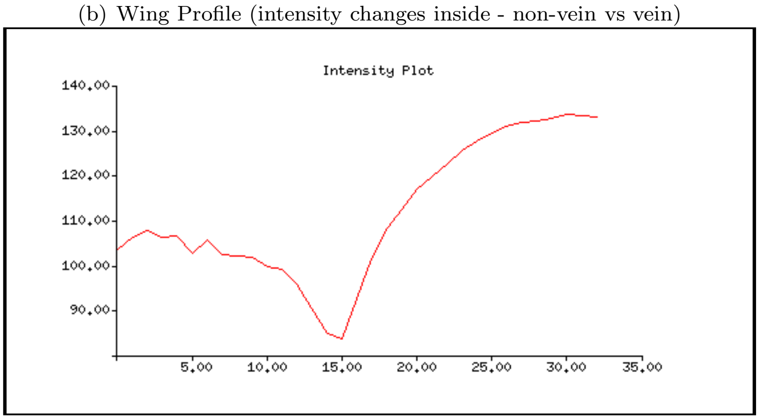
$$\theta = atan2 \left[\frac{\frac{d^2 h_{\sigma_{ab}}}{dxdy}}{\frac{d^2 h_{\sigma_{ab}}}{dx^2} - \frac{d^2 h_{\sigma_{ab}}}{dy^2}} \right] \tag{14}$$



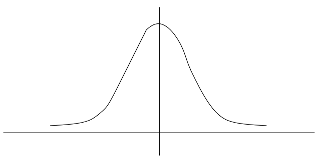
(a) Wing image



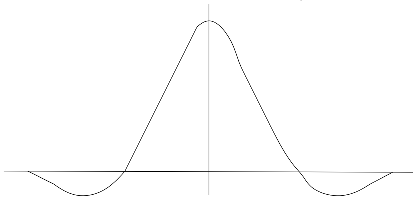
(b) Wing Profile (intensity changes inside - non-vein vs vein)



(c) Wing Profile (intensity changes - inside vs outside)



(d) Gaussian plot



(e) Difference of Gaussian plot

Figure 2: Fly wing profile and shape of the filter

The difference of Gaussian feature enhancement and corresponding orientation estimate can now be substituted into the standard Canny edge detection framework, which incorporates non-maximal suppression, for detection of an enhanced ridge, and hysteresis thresholding, for stable detection of connected edge structure close to the noise floor.

Sigma Value	Detected Ridges of width (in pixels)	Comments
1	3,4,5	Detected ridges of width 3,4 & 5 pixels; 1 pixel - not detected; 2 pixels - partially detected 6 pixels - double edge detected
2	2,3,4,5,6	Detected ridges of width 2,3,4,5 & 6 1 pixel - not detected
3	2,3,4,5,6	Detected ridges of width 2,3,4,5 & 6 1 pixel - not detected
4	2,3,4,5,6	Detected ridges of width 2,3,4,5 & 6 1 pixel - not detected
5	4,5,6	Detected ridges of width 4,5 & 6 1 & 3 pixels - not detected 2 pixels - wrongly detected (detected much outer than actual ridges)
6	5,6	Detected ridges of width 5 & 6 1 & 3 pixels - not detected 2 - detected just outside the actual ridge 4 - detected just inside the actual ridge
7	6	Detected ridges of width 6 pixels 1,3 & 4 - not detected 2 pixels - wrongly detected (detected much outer than actual ridge) 5 pixels - detected just inside the actual ridge

Table 1: Scale sensitivity table for the original image.

Results

There are two aspects of the detection process which need to be investigated for our application. The first is the effect of scale of the detection filter on image features of a particular size. The other is the effect of noise. Only by ensuring that the data under analysis falls within the expected range of detectable structure can we be confident that the method is suitable for use. We can investigate the first of these issues with synthetic images of known structure scale and the second by investigation of noise sensitivity.

Optimisation of parameters-Scale The scale parameter σ_g and the threshold values(T_1 & T_2) has an impact on the detection and localisation of edges.

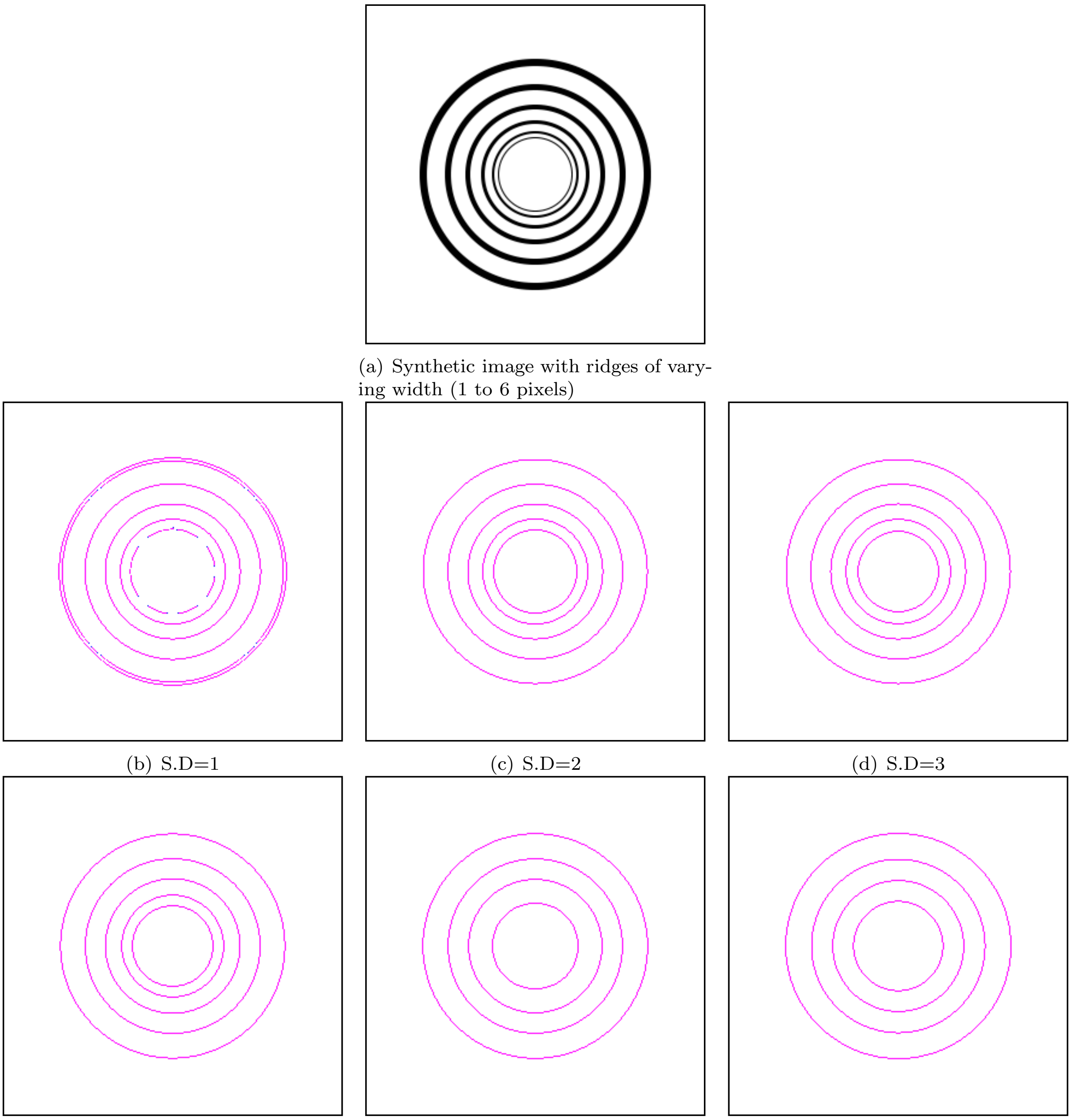
Generally, the fly wing image has ridges that are less than 10-12 pixels wide. The performance (sensitivity) of the algorithm at different scales (ridge-width) has to be tested so as to determine the optimal parameters(sigma values) that can be used for the scales (ridge-width) that we are interested in.

The scale sensitivity of the algorithm can be tested on a synthetic image (e.g. Fig. 3(a)). The synthetic image has ridges of varying width (1 to 6 pixels) and each ridge is away from the next one by a standard deviation of 3 pixels. The efficiency and scale sensitivity of the algorithm on the dataset(fly wing images) can be pre-determined by estimating the performance on the synthetic image.

As σ_g increases Gaussian filtering is more effective in reducing the noise frequency. However, after a certain value of σ_g, the edge detected becomes wider and therefore in some cases double edges are detected. Also, we can notice (Fig. 3) that ridges that are only of one pixel width are harder to detect with these specified sigma values. This is because the Gaussian filters used cannot be accurately represented at this scale.

An effective algorithm should be capable of detecting ridges irrespective of their width. A simple computational method would be to up-interpolate[3] the image for various sigma values is represented in the image by a factor of two and then apply the Ridge Canny algorithm. The scale sensitivity of the algorithm on the up-interpolated Fig. 4 and Table 2. Figure 5 shows the comparison of performance of Ridge Canny algorithm on the original and the up-interpolated image for different scales.

[3]This is the process of synthesising an image with more pixels than the original via a functional interpolation of the data.



(a) Synthetic image with ridges of varying width (1 to 6 pixels)

(b) S.D=1

(c) S.D=2

(d) S.D=3

(e) S.D=4

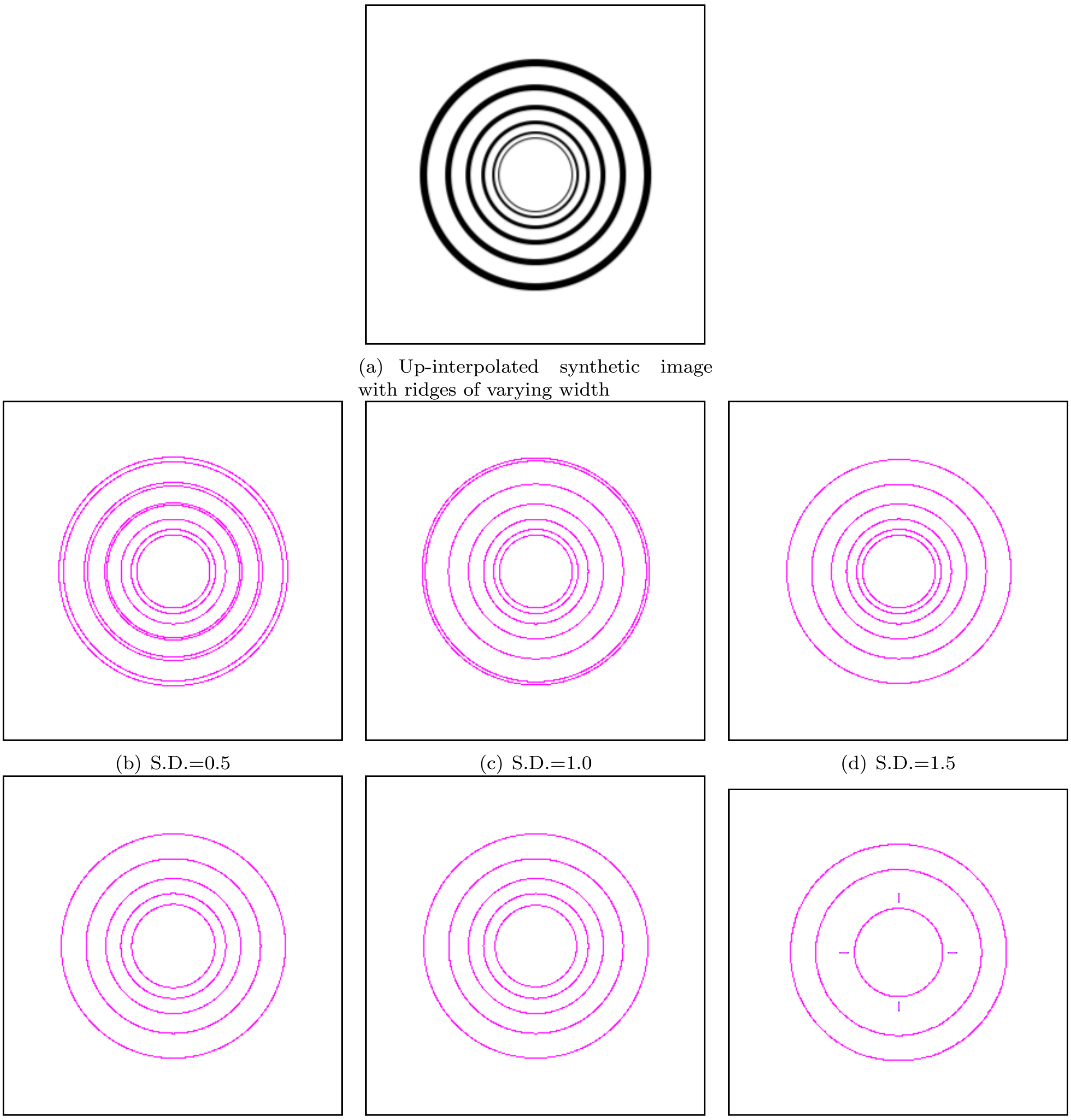
(f) S.D=5

(g) S.D=6

Figure 3: Scale sensitivity of Ridge Canny with varying sigma values (1-6)

Thus, the arbitrary parameters such as the sigma value for Gaussian smoothing and the threshold values for hysteresis thresholding are optimised to obtain best results. The optimal threshold values for Ridge Canny algorithm were also determined by testing the fly wing images with varying upper and lower threshold values (T_1 & T_2). (As the threshold value increased the noise decreased but when it crossed certain threshold value, some of the edge pixels were not detected).

Estimation of Noise Sensitivity The estimation of the sensitivity of the Ridge Canny algorithm to random noise reveals that the algorithm is capable of detecting the features even when there was a large additive noise (Fig. 6). The algorithm was tested for up to a noise factor of $\approx$10 times higher (Fig. 6(b)) than the estimated noise present in the original image (Fig. 6(a)). There was no considerable effect on the performance of the algorithm even with an increased noise (of upto $\approx$10 times) on the image (Fig. 6(h)). The algorithm seems to extract the features (landmarks) that we are interested in, ideally all the landmarks despite the noise present. The position of the landmarks (features) are plotted on the edge map to demonstrate this 7.This proves stability of the algorithm even in images with increased noise.



(a) Up-interpolated synthetic image with ridges of varying width

(b) S.D.=0.5

(c) S.D.=1.0

(d) S.D.=1.5

(e) S.D.=2.0

(f) S.D.=2.5

(g) S.D.=7.0

Figure 4: Scale sensitivity on the up-interpolated image for varying sigma values (0.5 - 7.0)

Discussion and Conclusions

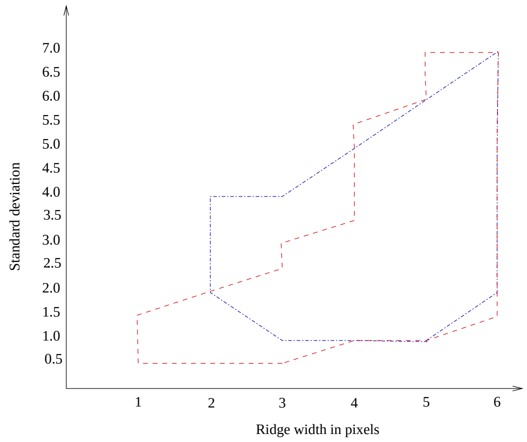
This report has outlined the extension of a framework consistent with the Canny step edge detector, for use in more general feature detection tasks. The main argument is that the main framework for the detection of connected features is inherently stable, and can be interpreted as consistent with the use of statistical hypothesis testing. The features selected can be adjusted for the purpose of locating alternative structures, provided that we substitute the feature enhancement stage (local gradient estimation in Canny) with another which has stable noise characteristics. We show how any convolution process is stable and explain how feature detection via convolution can be related to a likelihood estimate of the contribution of the feature shape to the image structure. Some readers may find the derivation of feature detection from conventional statistics interesting if they have not considered the task in this way previously. Certainly, modern text book in this area do not make use of this interpretation.

We have then used our understanding of the feature detection process to motivate a difference-of-Gaussian convolution as the basis for ridge detection in the specific task of fly wing analysis. The suitability of this feature detector has been tested in terms of ridge scale sensitivity and the effects of noise. In both cases we have shown that the use of a fixed difference-of-Gaussian ridge detector is suitable for the available data. Other aspects of

Sigma Value	Detected Ridges of width (in pixels)	Comments
0.5	1,2,3	Detected ridges of width 1,2,& 3 pixels; 4,5 & 6 pixels - double edge detected
1.0	1,2,3,4,5	Detected ridges of width 1,2,3,4 & 5 6 pixels - double edge detected
1.5	1,2,3,4,5,6	Detected ridges of width 1,2,3,4,5 & 6 All ridges detected precisely
2.0	2,3,4,5,6	Detected ridges of width 2,3,4,5 & 6 1 pixel - not detected
2.5	3,4,5,6	Detected ridges of width 3,4,5 & 6 1 pixel - not detected 2 pixels - detected just inside the ridge
3.0	3,4,5,6	Detected ridges of width 3,4,5 & 6 1 pixel - not detected 2 pixels - detected just inside the actual ridge
3.5	4,5,6	Detected ridges of width 4,5 & 6 pixels 1 pixel - not detected 2 pixels - wrongly detected (detected much inner than actual ridge) 3 pixels - detected just inside the actual ridge
4.0	4,5,6	Detected ridges of width 4,5 & 6 pixels 1 pixel - not detected 2 pixels - detected just inside the actual ridge 3 pixels - detected just inside the actual ridge
4.5	4,5,6	Detected ridges of width 4,5 & 6 pixels 1 pixel - not detected 2 pixels - detected just inside the actual ridge 3 pixels - detected much inner than the actual ridge
5.0	4,5,6	Detected ridges of width 4,5 & 6 pixels 1 & 3 pixels - not detected 2 pixels - detected just outside the actual ridge
5.5	4,5,6	Detected ridges of width 4,5 & 6 pixels 1 & 3 pixels - not detected 2 pixels - detected just outside the actual ridge
6.0	5,6	Detected ridges of width 5 & 6 pixels 1 & 3 pixels - not detected 2 pixels - detected outside the actual ridge 4 pixels - detected just inside the actual ridge
6.5	5,6	Detected ridges of width 5 & 6 pixels 1,3 & 4 pixels - not detected 2 pixels - wrongly detected (detected outside the actual ridge)
7.0	5,6	Detected ridges of width 5 & 6 pixels 1,3 & 4 pixels - not detected 2 pixels - wrongly detected (detected outside the actual ridge)

Table 2: Scale sensitivity table for the up-interpolated image.

feature detection performance, such as false detection rate (FDR), repeatability and orientation accuracy, have not been assessed at this time. Though FDR has been identified here as a particular problem with use of a linear filter for feature enhancement, the simple nature of the fly wing images and the intended use of robust shape recognition strategies means that this issue would be best addressed in the context of its effects on subsequent shape analysis.



1

Figure 5: Optimising scale on various ridge widths. The Dotted (blue) line represents the scale performance of Ridge Canny algorithm on the original image and the dashed (red) line represents the scale performance on the up-interpolated image.

Appendix A

Many image processing text books will say that features in an image I can be enhanced, ready for threshold detection, using a convolution with a kernel shaped similar to the the feature t

$$f(x,y) = \sum_{ij} t(i,j) I(x+i, y+j)$$

As with many image processing techniques, this approach can be related to a likelihood-based analysis of the image. For an image with independent random Gaussian noise of width σ we can write the likelihood for a template t' describing an image region as

$$L = \sum_{ij} \frac{(\alpha(x,y)t'(i,j) - I(x+i, y+j))^2}{\sigma^2 + var(t')}$$

where $var(t')$ is the variance (ie: the accuracy) of the assumed template, and α is a scale factor defining the strength of the linear contribution of t' to the image. Assuming that $var('t)$ is independent of α, the likelihood estimate of α is

$$\alpha(x,y) = \sum_{ij} \frac{t'(i,j) I(x+i, y+j)}{\sigma^2 + var(t')}$$

This justifies using the template

$$t = \frac{t'(i,j)}{\sigma^2 + var(t')}$$

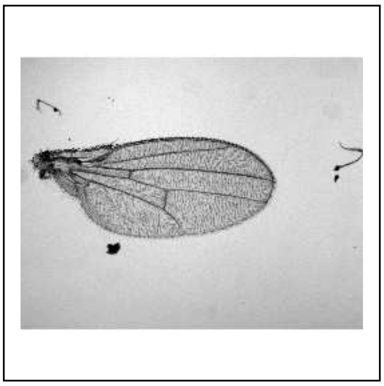
for detection purposes ($\alpha = f$).

Exact knowledge of the template within a defined region, and ignorance elsewhere leads directly to $t = t'$. However, it is also sensible to assume that our knowledge of the expected template reduces away from the centre (ie: a radial weighting) such as

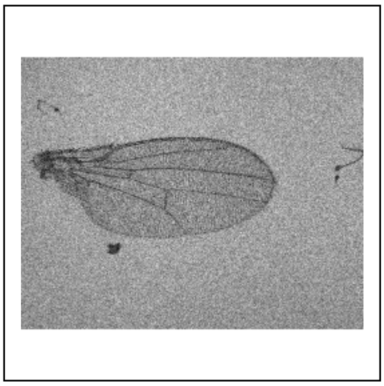
$$t = \frac{t'(r,\theta)}{\sigma^2 + w(r)}$$

for example $w(r)$ may be set to give an overall Gaussian weighting

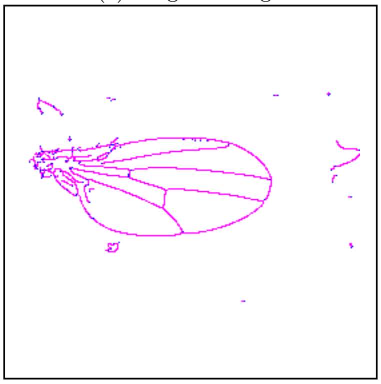
$$t = t'G(r) \qquad with \qquad w(r) = 1/G(r) - \sigma^2$$



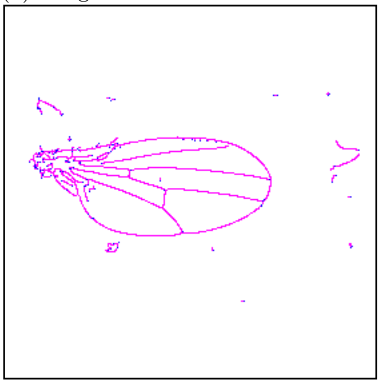
(a) Original Image



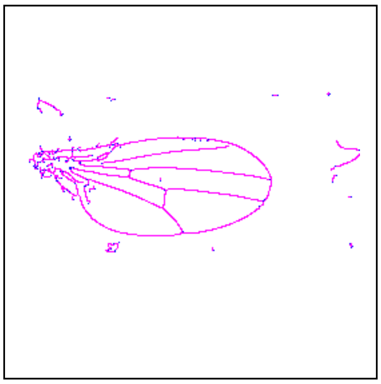
(b) Image with ≈10 times added noise



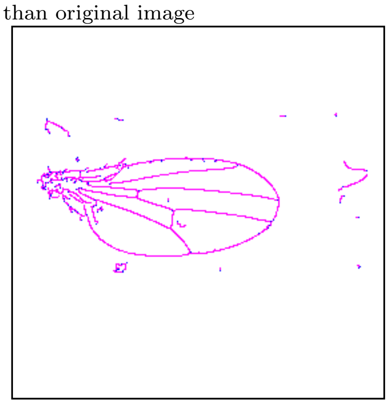
(c) Result on original image



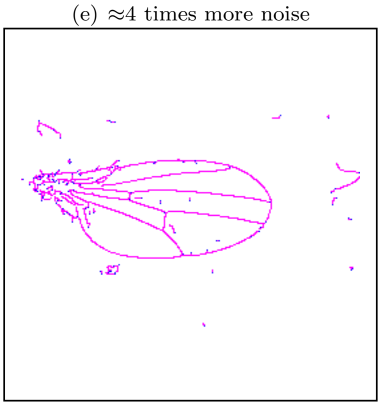
(d) Noise present is ≈2 times higher than original image



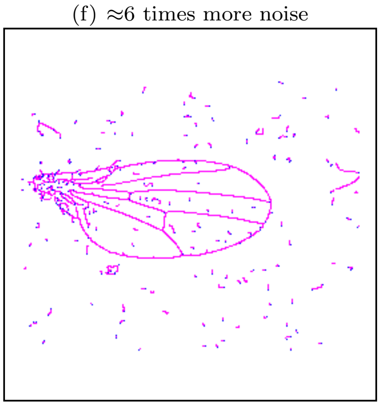
(e) ≈4 times more noise



(f) ≈6 times more noise



(g) ≈8 times more noise



(h) ≈10 times more noise

Figure 6: Noise Sensitivity estimation for Ridge Canny algorithm a) Original image b) Image with ≈10 times noise c) Result of Ridge Canny on original image. Figures d,e,f,g & h are the results of Ridge Canny algorithm on images with added noise (≈ 2,4,6,8 & 10 times the noise present in the original image).

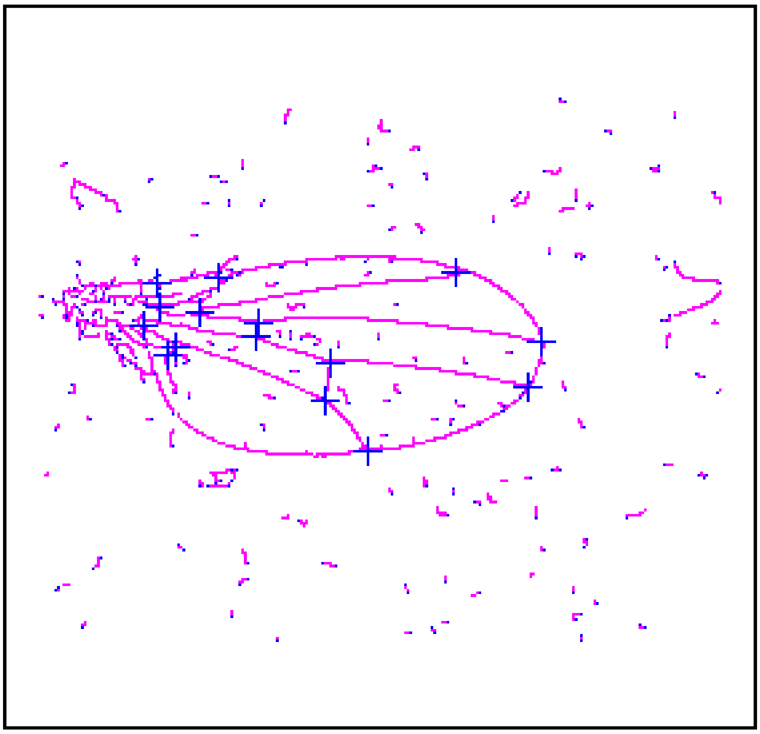


Figure 7: Landmarks (features) plotted on the edge map of the image with ≈10 times more noise

Notice, that from a matching perspective this does not give a unique interpretation for any particular t', as there are an infinite number of possible weightings we can use in the construction of t.

In addition, for a given level of image 'signal' the best response will be given to an image looking like t and not t'. In some applications more specificity to shape may be achieved by locating features on the likelihood L, rather than using α.

References

[1] T Peli and D Malah. A study of edge detection algorithms. *Computer Graphics and Image Processing*, 20:1–21, 1982.

[2] D Marr and E Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London, Series B*, 207:187–217, 1980.

[3] D Vernon. *Machine Vision: Automated Visual Inspection and Robot Vision.* Prentice-Hall, 1991.

[4] J J Clark. Authenticating edges produced by zero-crossing algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):43–57, 1989.

[5] J S Huang and D H Tseng. Statistical theory of edge detection. *Computer Vision, Graphics and Image Processing*, 43:337–346, 1988.

[6] J F Canny. A computational approach to edge detection. *IEEE Transactions on Pattern analysis and Machine Intelligence*, 8(6):679–698, 1986.

[7] D Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* W. H. Freeman and Co., 1982.

[8] Y Midoh, K Miura, K Nakamae, and H Fujioka. Statistical optimization of Canny edge detector for measurement of fine line patterns in SEM image. *Measurement Science and Technology*, 16:477–487, 2005.

[9] C Steger. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):113–125, 1998.