Tina Memo No. 2008-010
Internal.

Useful Image Processing Methods.

Neil A Thacker.

Last updated
12 / 11 / 2015

WWW.TINA-VISION.NET

Imaging Science and Biomedical Engineering Division,
Medical School, University of Manchester,
Stopford Building, Oxford Road,
Manchester, M13 9PT.

Useful Image Processing Methods.

NAT(12/10/08)

Introduction

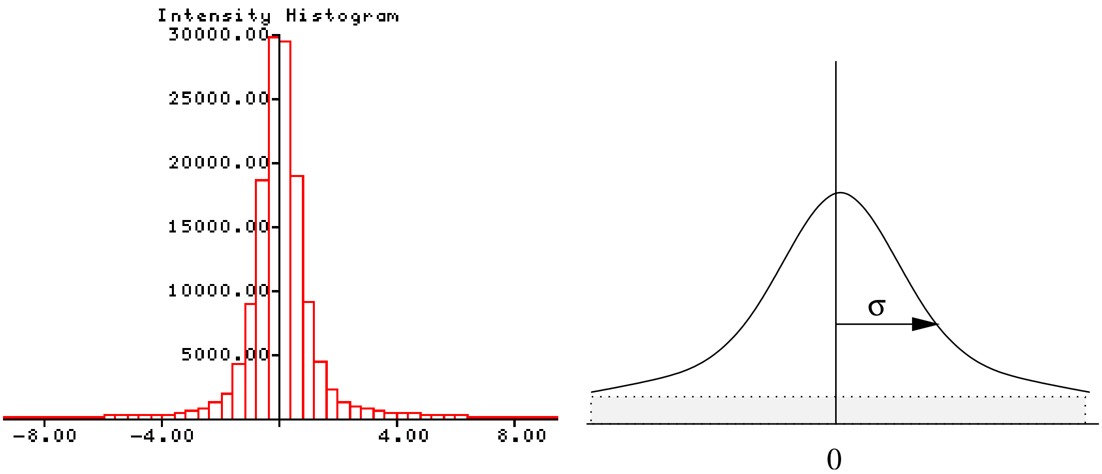
This document has been written to provide a description of some algorithms that are the frequently used in TINA [6], but would be considered too simple to be published. The common characteristic of these techniques is that they have relatively well defined statistical properties, allowing them to be used as **useful** intermediate stages in an image processing system.

Noise Estimation

Motivation

Many computer vision algorithms can be seen to have a large number of control parameters which are key to their successful application. These parameters tend to proliferate when constructing systems rendering the them unreliable for general use. In many cases these parameters are present in order to mitigate the effects of poor data and can be traced back to the effects of input image noise. Dealing with such issues in a satisfactory manner is made much easier if the input images and results of any processing stages have spatially uniform noise (σ_I). Processed images can often be pre-processed by a non-linear transformation in order to obtain this property. Free parameters can then be eliminated, if some way can be found to relate them to a noise estimate. In some cases (such as thresholds for feature detectors such as Canny) this may be a simple proportionality. Pragmatic application of this strategy therefore requires a method for automatically estimating noise in an arbitrary image.

Method



(a) (b)

Figure 1: *The noise is estimated from the variance of the distribution of second derivatives, following subtraction of uniform background.*

The method we prefer is based upon the observation that high order derivatives in images are dominated by the effects of image noise. Following this line of reasoning a histogram of the second order derivative from an image will illustrate two main features; a long tail of values associated with genuine image structure, and a peak at zero associated entirely with the noise process. The distribution of noise on a derivative of an image has a variance which is in a fixed proportion to the variance in the original image [3]. Measuring the width of the peak at zero thus provides a method for estimating the original image noise. This simple idea needs some care to make it work reliably in arbitrary images.

Two histograms are formed for the second derivatives in x and y. Valid derivatives are identified by eliminating any zero values adjacent to another zero value. This is done to remove blank areas (common in medical data) and saturated regions (common in digital images) which would otherwise contribute a large spike to the center of the distribution. The central peak of the distribution is treated by subtracting the background from genuine image structure (assumed uniform around zero) (figure 1). The variance of the central peak is then estimated from the histogram. The process is iterated in order to get a sufficient number of bins (≈ 20) within the noise peak. The variances are then scaled to get an estimate of the original image noise, and the vertical and horizontal estimates compared to identify image formation peculiarities before taking an average. For images larger than 32x32 pixels, and not resulting from earlier processes which might introduce spatial correlation (smoothing), the method is observed to estimate the true image noise to within a few %. The estimate of noise obtained using this method is more than good enough for the estimation of subsequent control parameters. The algorithm is available as the button **noise** in the **Imcalc Tool**, and has been used frequently as a key stage in data analysis during the writing of papers. The method should not be run on processed data unless it is known to have the property of uniform independant random noise.

Tangential Smoothing

Motivation

Though the best way to deal with image noise when computing derived quantities is to apply a valid statistical estimation to the problem, it is often far quicker to use a generic noise removal technique and then apply a naive estimator. We therefore often require methods to reduce image noise before computing noise sensitive local image properties (such as orientation). The most common method found in the literature is to use a Gaussian convolution (at some spatial scale). Though simple and efficient, this strategy has the disadvantage of also changing the nature of the signal in the image, destroying fine structure. In order to overcome this failing researchers investigated the use of anisotropic filtering [1]. The idea here is to apply convolutions in a way which minimises the effect on the underlying image structure. Published algorithms are often iterative or involve large scale filters. However, for most applications this will result in over-smoothing and an output which is of no practical value for subsequent processing. There is a case to be made for the smallest possible anisotropic filter, which has the least impact on the information content of the image.

Method

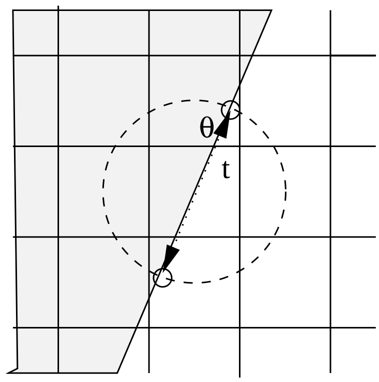


Figure 2: *The location of image samples used in tangential smoothing.*

The smallest possible symmetrical anisotropic filter is one which takes a simple average of two adjacent values. In tangential smoothing these two values are selected in the direction which aligns with the local image structure. The tangential vector t is computed using the conventional first derivatives (though other methods could be used), and two grey-level values are linearly interpolated at a distance of one pixel away from the central value in this direction (figure 2). These two values are then averaged.

$$I'(x) \;=\; \frac{I(x+t) \;+\; I(x-t)}{2}$$

This new image has some useful statistical properties. As the output is simply a form of simple averaging, for an input image with uniform independent random noise, the output image has a noise level which is also uniform and approximately $\sqrt{2}$ smaller (the level of rotational uniformity depending upon the interpolation scheme). The process can also be applied (with caution) to images with slowly varying noise characteristics. As use of the central pixel value and extreme values are avoided in this calculation the processed image will have no drop-out or outliers. Tangential smoothing therefore has similar response characteristic to the median filter, with the main difference being that quantized (integer) values become real valued [1]. Taken together, these properties allow the filter to be applied to parametric images, such as those generated by analysis of temporal data sets. In addition, the result has noise which is approximately statistically independent of the original (see Bland-Altman plots below). Therefore, if there are no out-liers to have to worry about, it is legitimate to average the output image with the input to reduce the noise further ($\approx \sqrt{3/8}$). This level of noise reduction approaches what we might expect to be close to the limit of any method which does not assume a specific structural model for the image content.

With regard to the ability of tangential smoothing to remove noise, its performance is only bettered by techniques which loose the spatial uniformity (such as Non-Local Means [5]). The output can therefore be used as a reliable stage in data processing, while the more aggressive noise filters can not. The algorithm is available in the **Imcalc Tool** and is executed using the **tsmooth** function. It has been used frequently in data preprocessing for the production of data for published papers. We will always used this simple technique in preference to Gaussian smoothing, and have even developed 3D versions (which average along a plane) for some projects. Though the process can be iterated in order to generate a larger scale anisotropic filter, this is probably not advisable for quantitative tasks.

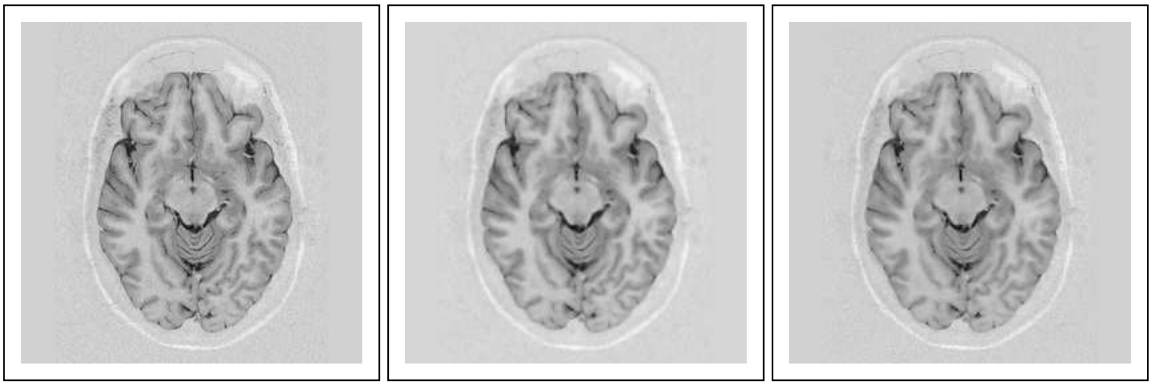


Figure 3: An MR image of a normal brain(a) I, (b) tangential smoothing T(I) and (c) outlier pixels replaced with their original values.

Bland-Altman Plots

Motivation

Even when we are unaware of their specific origins, all data processing algorithms are designed on the basis of a model of data noise. More often than not this model is one of uniform independant Gaussian noise (as estimated above), but more complicated models are also used. Data needs to conform to these models if we wish an algorithm to work appropriately.

Noise models in images can be determined by constructing and fitting Bland Altman plots from differences between repeat measurements. Unfortunately, often we do not have the second image needed and must use other means to estimate local image noise. We explain below how a method designed for assessing the degree of correlation between images can be extended to approximate construction of a Bland-Altman plot from a single image.

Method

Let us assume that we wish to establish that two images; an input image I and a filtered image $T = f(I)$ (figure 3), have independant noise processes. We can do this by looking at appropriate scatterplots, using the **scat** button in the **Imcalc Tool**. Whilst the correlations between signal might produce quite arbitrary structures within the plot, any correlation in noise between the data will show up as a consistent (oriented) point-spread process. In

[1] A very important difference if you wish to investigate the distribution of image values.

particular, if we plot I vs T then point-spread structure will align with the axes of the plot if data from I and T are uncorrelated.

This point spread process may be difficult to identify, but some rotations of the data, such as $I - T$ vs $I + T$ may illustrate the behaviour more clearly. This scattergram can also have other uses, if T is a noise filtered version of I (for example Tangential Smoothing (figure 3 b)) then $I - T$ is an estimate of image noise, and $I + T$ is an estimate of signal, so that our scattergram is a form of Bland-Altman plot. Rotations of data original data will remain uncorrelated provided that the linear composition process (here (1,1) and (1-1)) is orthogonal, i.e. the dot product of the corresponding weight vectors ((1,1).(1,-1)) is zero. Unfortunately, $I - T$ and $I + T$ are correlated (figure 4 b).

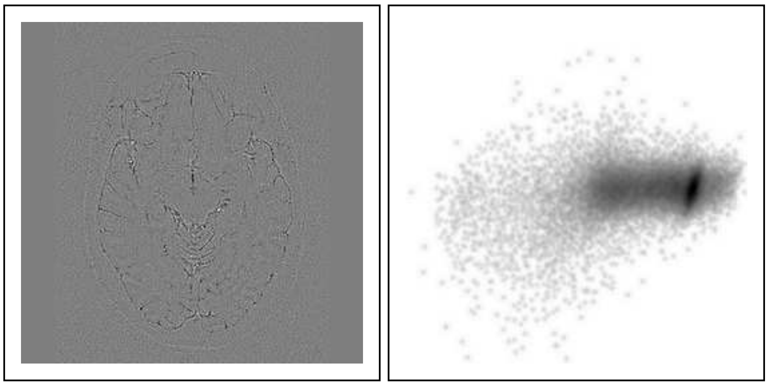


Figure 4: The difference image (I-T) and a scatter plot of I-T, I+T, illustrating correlated noise structure (note orientation of p.s.f.). The large spread of values on the left hand side is due to outliers (extreme values) generated by the interpolation assumptions used in tangential smoothing.

We aim now to use our understanding of the behaviour of scatterplots to obtain a quantitative assessment of the correlation between the noise in two images, I and T. We assume that the noise on T can be approximated[2] as a linear combination of independant noise processes s and i, seen on each pixel.

$$t \; = \; (1-\alpha)s \; + \; \alpha i \qquad\qquad - (1)$$

Under this assumption it must be possible to construct a linear combination $R(\beta) \;=\; \beta T + I$ such that R is statistically independant of $I - T$.

We can now go further and quantify the value of α, i.e. the amount of linear correlation of T with I.

From (1) the noise on R is given by

$$r \; = \; \beta t \; + \; i \; = \; \beta(1-\alpha)s \; + \; (\beta\alpha + 1)i \qquad - (2)$$

whilst the noise on $I - T$ will be

$$i \; - \; t \; = \; (\alpha - 1)s + (1-\alpha)i \qquad\qquad - (3)$$

For the scatterplot to show independance there must then be zero correlation beteen r and $i - t$. Mathematically, the dot product of the factors describing the two independant noise processes (s and i) must be zero, i.e.

$$\beta(1-\alpha)(\alpha-1) \; + \; (\beta\alpha+1)(1-\alpha) \; = \; 0$$

Solving for α, we find that the scatterplot for $R(\beta)$ vs $I - T$ shows no correlation, then the average proportion of noise from I in T is

$$\alpha \; = \; \frac{\beta - 1}{2\beta} \qquad\qquad - (4)$$

In the limit $\beta \to \infty$, $\alpha \to 1/2$, so the maximum correlation factor we can estimate with this method is 0.5.

Provided that the data are indeed uncorrelated the distribution of $I - T$ and $I + T$ can be fitted to determine any functional dependancy of the noise on the signal ($I \pm g(I)$). Unfortunately, if the data are correlated (figure 6 b) then the standard assumptions needed to fit the plot (including the noise process being symmetrical about the

[2]This approximation will be exact for a linear filter.

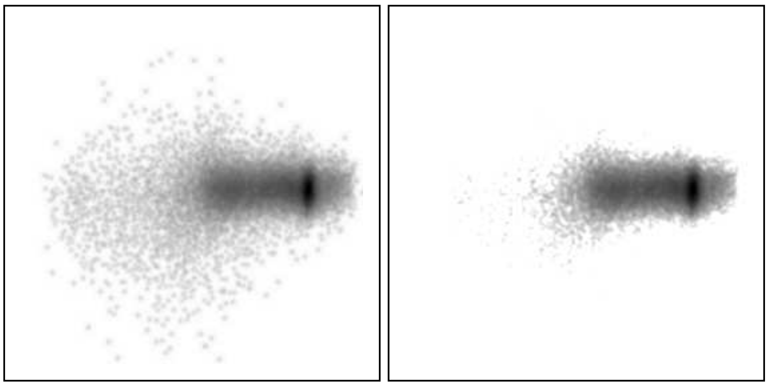


Figure 5: The scattergram of I-T, I+5T, illustrating uncorrelated noise structure, and the same scattergran now thresholded to remove outliers.

signal axis) are invalidated. However, as R is a scaled approximation to I, we can now fit the BA plot of $I - T$ vs R instead.

For the data shown in figure 4, a value of $\beta = 3.5$ is found to approximately decorrelate the noise in R and $I - T$ (figure 5 a), ie. $\alpha = 0.36$. As a final step this data needs to be thresholded in order to remove the outliers from $(I - T)$. Only then is it suitable for estimation of a noise model $g(I)$ (figure 5 b).

Also, we can now improve our Tangential Smoothing algorithm, by replacing any pixels which are not consistent with simply removing noise (the thresholded values) by their original grey-level values (figure 2 c). The **iscat** button in the **Imcalc Tool** supports the steps necessary to do this. This is a legitimate way of avoiding inappropriate smoothing which can also be used with other noise filters.

Variance Based Edge Strength

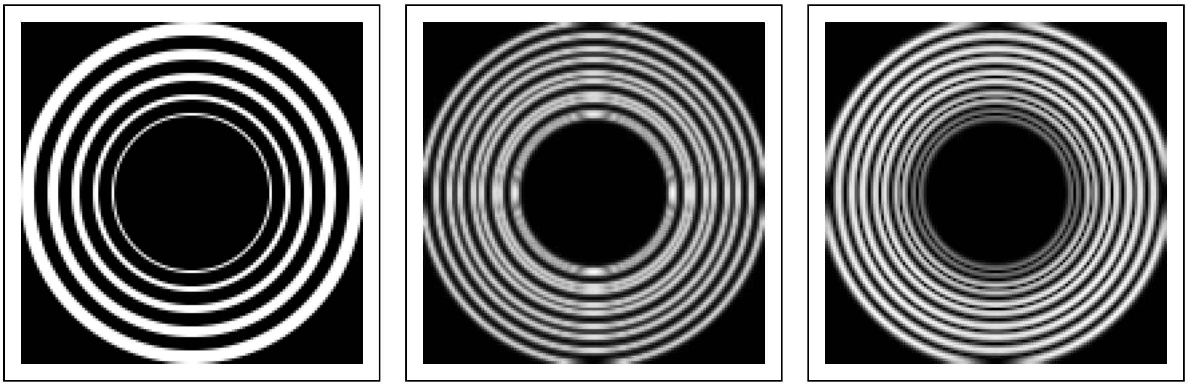
Motivation

Following the large numbers of edge detection papers published in the literature in the early years of computer vision, the decision was made by the community that papers would no longer be solicited for conferences on this topic, and that any new ideas would need to be very special to warrant conference or journal space. Interestingly, despite this community-wide decision the topic didn't achieve closure. Although it has long been accepted that the Canny algorithm has some merit, it is also well known that edge detection based upon step boundaries was not a good solution for much, if not most, image data. Mathematically motivated approaches, such as those based upon wavelets, have attempted to revitalise the area but have not been entirely successful. In fact, what is needed is not a grand mathematical theory but a basic appreciation of practical vision tasks.

The main practical problem faced by attempting to use Canny edges in a vision system appears to be the loss of fine line structures. Edges are not detected well for lines with a thickness ≈ 1 pixel, as the derivatives at those locations are zero. If the region is magnified or smoothed (broadening the structure) then edges, when they are finally detected, are detected twice (on either side of the line). As a consequence, attempts to simply summarise structural geometry of objects, for example, will be missing many elements which a human observer would identify as key if they were drawing a sketch. One common approach to deal with this is to use alternative detectors, the most familiar of which is probably the Difference of Gaussian (DOG). The computational form of a DOG can be related to image derivatives, but its use for feature detection is best understood as a matched filter. The DOG therefore responds best to structures which are similar to the combined convolution kernal, this excludes step edges. Ideally we would like a method which responds to both step edges and thin lines.

Method

The approach we prefer is to relate feature detection to well defined statistical measures, such as Fisher Information (see Tina memo 2007-001). From this limited set of approaches, the example which has the ability to detect fine line structure is local image variance (ie: the difference between the central pixel and its neighbours), defined over



(a) (b) (c)

Figure 6: *The response of a variance based feature detector ($\sigma = 0.7$ pixels) to lines of increasing thickness (1-5 pixels), illustrating the shift between step edge detection and line detection (b). The sum squared derivatives (Canny) is shown for comparison (c). Whilst the main point here is the reduction in computed response for the central ring in (c), it is perhaps interesting to note that the aliasing artefacts computed in (b) can be "perceived" in (a) but are not generated at all in (c).*

a region of N pixels (figure 6).

$$V \;=\; \frac{1}{N}\sum_i^N (I_i \;-\; \hat{I})^2 \quad \textit{where} \quad \hat{I} \;=\; \frac{1}{N}\sum_i^N I_i$$

Where I_i is the image grey-level. We can apply this over a weighted region (eg: Gaussian) in order to improve the noise characteristics and achieve rotation invariance.

$$W \;=\; \sum G(r_i)(I_i \;-\; \hat{I})^2 \quad \textit{where} \quad \hat{I} \;=\; \sum G(r_i) I_i$$

where $G(r_i)$ is a radial weighting. This can be rewritten in convolution form as

$$W \;=\; G \otimes (I^2) \;-\; (G \otimes I)^2$$

Points with high values on the basis of this measure are those which have locally large grey-level variation. This will respond equally well to step edges and thin line structures. Features will of course be shifted relative to a step edge method, but this can be accommodated in later processing, in a way which failing to detect an edge in the first place can not.

Assuming uniform independent noise on the input image, it can be shown that the expected variance on W is

$$var(W) \;=\; \sum_i^N G(r_i)^2 (I_i \;-\; \hat{I})^2$$

ie: the same process with the width of the Gaussian scaled by $\sqrt{2}$. As a consequence the noise on W is approximately proportional to W. In theory a proportional error can be transformed to homogenous errors using a logarithm, but in this case $U \;=\; \sqrt{W}$ is found to have more uniform noise (figure 7 b). This property of noise uniformity is found to be useful, and is key to the use of such a feature detector in subsequent stages of a system. One example of which is in the estimation of feature orientation (see below). Variance based edge detection is used in several places in the TINA software, and can be executed via the appropriate set of button presses in the **Imcalc Tool**. The difference between variance based edges and step edges is generally subtle. The variance based detector has been provided as an option for the feature validation process (**model proj** in the **Imcalc Tool**) and seems to offer a better spatial response, particularly around specular highlights (figure 8 b).

Feature Map Orientation

Motivation

It can be observed that the first derivatives of an image, as approximated by a finite difference convolution ($\frac{1}{2}(-1, 0, 1)$) often do not summarise the orientation of local image structure in the way we might like. The first

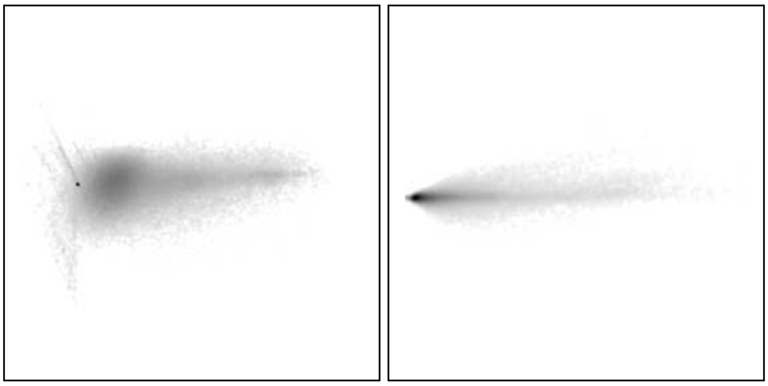
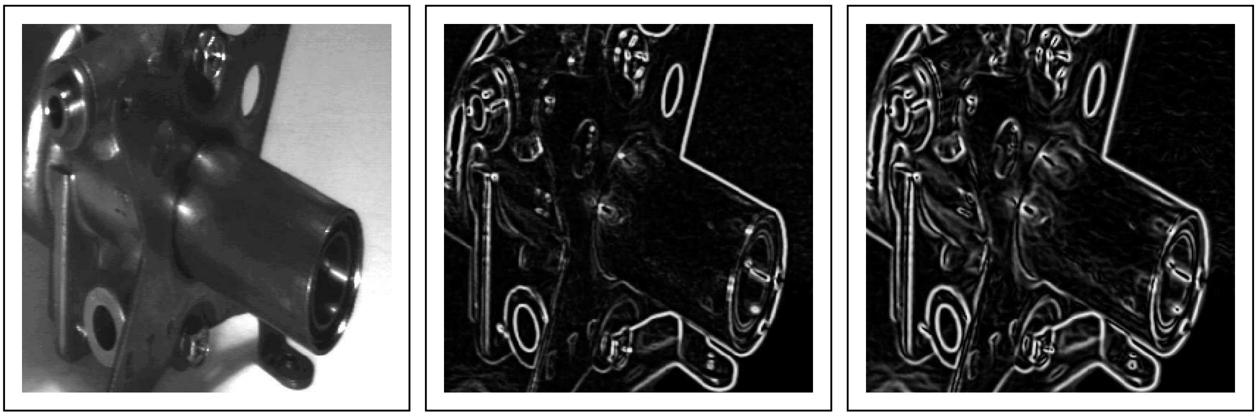


Figure 7: Bland-Altman plots (constructed as defined previously) for $log(W)$ and $U = \sqrt{W}$ using the data shown in figure 8.



(a) (b) (c)

Figure 8: *The response of a variance based (b) feature detector ($\sigma = 0.7$ pixels), and step edge detector (c), when multipled by the probability of the enhanced feature being above threshold.*

derivative is appropriate for step edge features, but has difficulties over thin structures. The problem is analogous to the problem of edge detection, defined above. For alternative feature detectors we often need a new way of measuring and representing local image orientation which is equivalent to the first derivative used for step edges.

Method

For a feature strength image U with uniform independent noise we can calculate the second and cross spatial derivatives for a quadratic approximation to the local structure

$$D = \begin{bmatrix} \partial^2 U/\partial x^2 & \partial^2 U/\partial x\partial y \\ \partial^2 U/\partial x\partial y & \partial^2 U/\partial y^2 \end{bmatrix} = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

using two evaluations of the horizontal and vertical finite difference convolution operator.

The effective diagonal separation of the difference terms scales the estimated cross derivative by a factor of $\sqrt{(}2)$ so that.

$$\sqrt{2}C = \frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

Bearing this in mind, the orientation of the local image patch can be computed using the following relationships

$$D = RWR^T = \begin{bmatrix} w_1 cos^2\theta + w_2 sin^2\theta & (w_2 - w_1)(sin\theta cos\theta) \\ (w_2 - w_1)(sin\theta cos\theta) & w_2 cos^2\theta + w_1 sin^2\theta \end{bmatrix}$$

where R is a rotation matrix for local structure and W is a diagonal matrix of singular values (squares of the eigen-values) so that

$$tan2\theta \ = \ 2C/(A-B) \qquad \rightarrow \qquad \theta \ = \ \frac{tan^{-1}(2C/(A-B))}{2}$$

Assuming uniform independent errors on the input image U, $var(2C) \ = \ 1/2$ and $var(A-B) \ = \ 3/4$. If these values were equal then the propagated variance on θ would be

$$var(\theta) \ \propto \ [(2C)^2 \ + \ (A-B)^2]^{-1} \ = \ (w_1 \ - \ w_2)^{-2}$$

ie: the difference in the square of the eigen-values for the local image. This would be a rotation invariant quantitation of structural alignment. The slight difference in $var(2C)$ and $var(A-B)$ implies slight differences in accuracy for measurement with worst precision ($\approx +\ 10\ \%$) associated with structure which is diagonal to the image axes. For most purposes such differences are well within the tolerance of algorithm stability and can be neglected.

A suitable substitute for the first derivatives of an image, are to use

$$t \ = \ (\sqrt{var(\theta)}cos\theta, \sqrt{var(\theta)}sin\theta)$$

for the tangential vector or

$$p \ = \ (\sqrt{var(\theta)}sin\theta, -\sqrt{var(\theta)}cos\theta)$$

for the perpendicular. Images of the x and y components of p and t are expected to have approximately uniform random noise. Both of these representations allow reconstruction of $var(\theta)$ as

$$var(\theta) \ \propto \ \sigma_I^2/|t|^2 \ = \ \sigma_I^2/|p|^2$$

which is analogous to the error characteristics obtained with the use of first derivatives when measuring orientations of step edges ie:

$$e \ = \ (\partial I/\partial x, \partial I/\partial y) \quad and \quad var(\theta) \ \propto \ \sigma_I^2/|e|^2$$

In particular, p can be used as a direct substitute to first derivatives in algorithms (such as edge linking) which require a definition of orientation which is based upon features other than step edges (such as difference of Gaussian or local variance) but need to be statistically equivalent to first derivatives.

The above technique is used as part of the processing for the variance based edge detection in TINA, but can also be executed by use of an appropriate series of button presses in the **Imcalc Tool**.

Rank Order Filtering

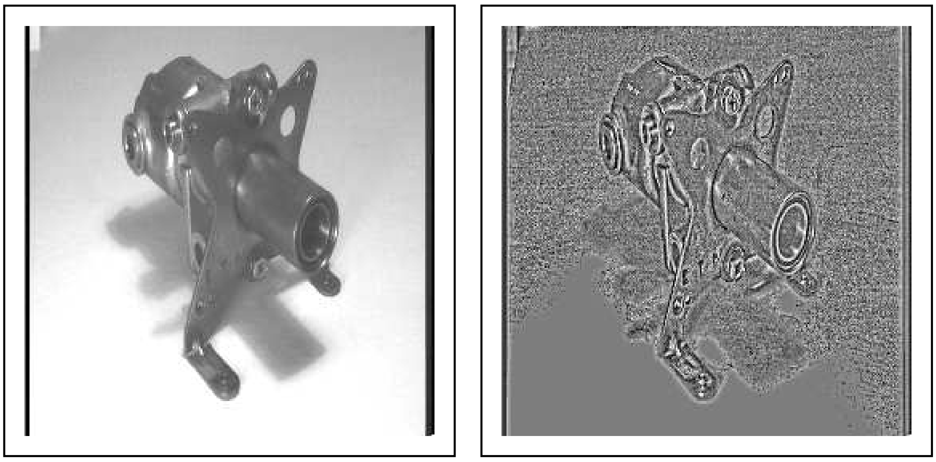
Motivation

Many images are formed in such a way that the absolute grey-level value at each pixel has no quantitative meaning. Sometimes the grey-levels may be calibrated and rescaled (such as the use of Hounsfield units in CT images), but quite often the image formation process prevents us from identifying a simple relationship between the observed image and the original scene. This problem is made more obvious when trying to compare two images of a scene. Even images of the same scene, rotated by only a few degrees, might demonstrate significant changes due to the effects of illumination. In the absence if complete knowledge of scene structure and illumination, the best that might be hoped is that the data is monotonically related between the two images for equivalent locations. In statistical terms we would say that the data is non-ordinal, and many methods are available for analysis of such data, which are based upon the idea of first ordering the data to produce a ranked list. It is possible to apply the same idea to images, and the result is a rank order filter. Some might say that processes in the primate retina seem to involve some aspects of competitive response, similar to this sort of ranking process. The statistical justification for why such an image normalisation scheme might be present in biological structures seems straightforward.

Method

The idea is to define a local region around a central pixel, and then to compute a rank score which quantifies how many of the surrounding values are smaller than the one in the center. The process can be thought of as a small scale histogram equalisation, except the region used moves with each pixel.

Simple ranking will give very noisy output in regions which are uniform except for noise. This output instability can be removed if a soft ranking is performed, which takes account of the intrinsic noise level and estimates the



(a) (b)

Figure 9: *An image and the rank filtered output.*

probability that the central value g is greater than one of the randomly selected neighbours n. For spatially uniform Gaussian image noise this calculation should use the error (*erf()*) when comparing the difference between the central pixel and any neighbour [4].

$$P(g > n \in N) \ = \ \frac{1/2 \ + \ \sum_i^N erf(\frac{g-n_i}{\sqrt{2}\sigma_I})}{N+1}$$

It is, however sufficient to approximate the error function as a linear ramp, so that the rank R $(0 < R < N)$ is calculated as a sum of truncated linear sections. This is the form of the algorithm which is available in the **Imcalc Tool** and executed upon running the **rank** process. The slope of the linear approximation can be set according to the estimated image noise σ_I (a good approximation is to use a linear ramp which has a length $10\sigma_I$).

In principle the rank output from two images can be compared directly using a conventional similarity score, such as least squares, even when the original images can not. Other authors have used this process as part of a region matching strategy for stereo algorithms [2]. Strictly this should be done in a way which takes account of the propagated effects of image noise on the soft-rank value. Soft-rank images are also useful for display purposes, making the local effects of structure much more visible than in the original image.

The soft-rank of a pixel value has several useful properties. As for histogram equalisation, the histogram of the resulting output image is uniform, as each rank value is itself an instance of a random sample which is drawn from a uniform distribution [3]. We can also show that, under a wide variety of feature detection models, the estimate of rank probability is likely to be monotonically related to the probability of detecting connected features. For example, one effective method for detecting edges from a gradient image is to demand that the feature strength (eg: U) is greater than some number of its neighbours (the **Imcal Tool** applies this simple strategy when casting a feature significance map to an edge map). If the probability that the central value would be larger than its neighbours is P then the probability that it is greater than exactly six from eight randomly selected neighbours $P_{(6/8)}$ is obtained from the binomial distribution

$$P_{(6/8)} \ = \ 28P^6(1-P)^2$$

The probability that it is greater than 6 or more neighbours (P_e) is then

$$P_e \ = \ P_{(6/8)} \ + \ P_{(7/8)} \ + \ P_{(8/8)} \ = \ 28P^6 \ - \ 48P^7 \ + \ 21P^8$$

and the corresponding hypothesis test H_e is given by the normalised integral

$$H_e \ = \ 12P^7 \ - \ 18P^8 \ + \ 7P^9$$

On a 3x3 region this strategy identifies connected (one pixel wide) edge ridges well. However, there are clearly many different ways that we can define a detection process. We may choose to use a larger region than 8, or to say

[3]This is strictly only true if we believe that the central value is not special and there may be exceptions to this rule in some circumstances. In particular, softrank values cluster around a value of $N/2$ in structureless regions. It makes sense to suppress these regions by multiplying by the probability of the feature strength being above threshold, as in figure 9.

that a gradient greater than all neighbours is a corner, not an edge. Also several approximations must be made in estimating a given value of P using the local rank (R). There is consequently some degree of flexibility involved with how we can choose to define and compute P_e.

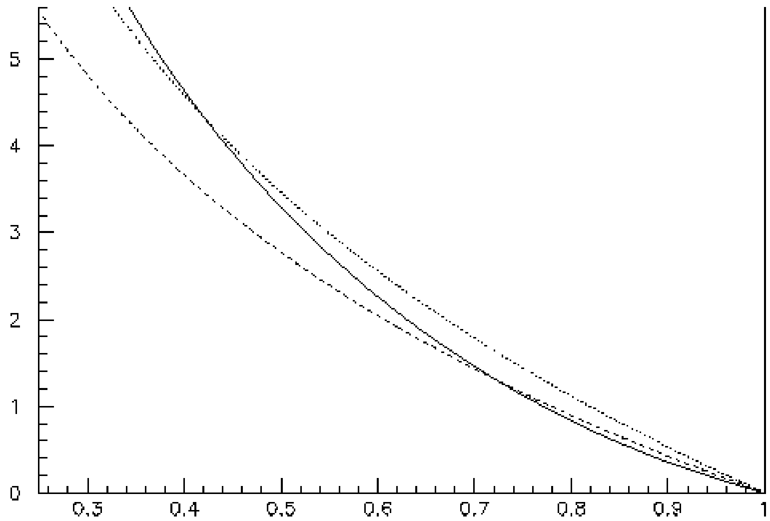


Figure 10: *The similarity of $-log(4P)$ (dotted curve) and $-log(5P)$ (dashed curve) to $-log(H_e)$ (solid curve). The minimum difference of $|log(H_e) - Mlog(P)|$ is less than 0.2 for $H_e > 0.01$ with $M = 4.6$. This value would therefore be expected to be the best matching curve (on the basis of a Kolmogorov-Smirnov test) in real data, if H_e is based upon a valid definition for an edge.*

It is therefore legitimate to select simple functions which approximate P_e and H_e well with a small number of parameters, such as

$$P_e \approx P^M \approx (R/N)^M$$

with $M \approx 4.5$ for H_e (Figure 6), with the best value for free parameters then determined on data. Such a polynomial relationship tells us that the softrank value is a good approximation to the likelihood of detecting a connected edge feature. This has numerous uses for object localisation and feature hypothesis verification.

References

[1] G. Gerig, O. Kubler, R. Kikinis, and F. A Jolesz. Nonlinear Anisotropic Filtering of MRI Data. IEEE Trans. Med Imag., vol. 11, pp. 221-232, 1992.

[2] M O'Neill and M Denos, Automated System for Coarse to Fine Pyramidal Area Correlation Stereo Matching. Image and Vision Computing, 14:225–236, 1996.

[3] K. Rank, M. Lendl and R.Uberhauen, Estimation of Image Noise Variance. IEE, Proc Vis. Image Signal Process, 146, 2, 1999.

[4] N.A.Thacker, Avoiding Zero and Infinity in Sample Based Algorithms, Tina memo 2009-008, 2009.

[5] J.V. Manjon, M. Robles, N.A. Thacker, Multispectral MRI De-noising Using Non-local Means. Proc. MIUA 2007, 41-46, Aberystwyth, 2007.

[6] www.tina-vision.net