Special cases where benefits arise from using the logarithm transform for illumination invariant feature extraction

R. B. Fisher
Department of Artificial Intelligence, Edinburgh University
5 Forrest Hill, Edinburgh EH1 2QL
rbf@dai.ed.ac.uk

Abstract

One of the goals of image feature extraction is to extract features from an image that are dependent on the scene, rather than the image, which also includes intensity information. In theory, a logarithmic transformation allows the extraction of many different types of image features, with the magnitude of the extracted feature being more representative of the scene property. Unfortunately, the magnitude of the effect is usually dominated by quantisation and image noise. This paper outlines the theory and demonstrates the special cases where there is an advantage to using the log transform.

1 Introduction

One of the goals of image feature extraction is to extract features from an image that are dependent on the scene, rather than the image, which also includes intensity information. A standard image model describes a digitally recorded image as:

$$P_{ij} = \alpha(\rho_{xy}\psi_{xy}I_{xy})^{\gamma} \tag{1}$$

where:

P_{ij}	measured intensity value at pixel (i, j)
α	a scale factor
ρ_{xy}	surface albedo at point (x, y) corresponding to pixel (i, j)
ψ_{xy}	surface orientation dependent reflectance scaling at point (x, y)
I_{xy}	incident illumination magnitude at point (x, y)
γ	gamma factor of the camera

Here, we assume that there is a linear relationship between the quantised image value and the analog input value (so that we can amalgamate all of the linear scale factors into α). Also, by reflectance, we mean both albedo (intrinsic light reflecting ability) and surface orientation dependent effects.

What is apparent from Equation (1) is that the measured pixel intensity value is a non-linear function of several factors. However, what is most commonly desired from image feature extraction are estimates of properties related to the surface reflectance ρ_{xy}, independent of the current illumination. Applying the standard feature detectors, such as edge detectors, bar detectors, blob detectors, corner detectors, etc. to an observed intensity image results in feature strengths that depend on the illumination as well as the underlying surface reflectance. This is almost axiomatic in the computer vision community and is taught to most students in their first course (e.g. [1], Sec 2.2.3 and Sec 3.2.4).

The usual recognition of this issue leads to the topic of color constancy and lightness ([3], Ch 9), whereby one assumes that the illumination is slowly varying, to allow separation of illumination and reflectance. The presumption behind the exploitation of lightness theory is that one uses the theory to reconstruct an image with the illumination removed. (This theory also requires many restrictive assumptions (planar world, no mutual illumination, uniform color patches, etc.).

However, rather than working on the reconstructed image, we can instead extract illumination independent features from a transformed image: by use of the logarithmic transformation exploited in the lightness computation, one can extract many different types of image features (or correspondingly, scene descriptions), with the magnitude of the extracted feature being a function of the underlying scene property. This paper describes this method and demonstrates its performance (or lack thereof) on a variety of scenes illuminated with different illuminants.

2 Theory

The feature extraction process begins with the same local logarithmic transformation as the lightness computation:

$$\begin{aligned}\log(P_{ij}) &= \log(\alpha(\rho_{xy}\psi_{xy}I_{xy})^{\gamma}) \\ &= \log(\alpha) + \gamma\log(I_{xy}) + \gamma\log(\psi_{xy}) + \gamma\log(\rho_{xy}) \qquad (2)\end{aligned}$$

This has the effect of changing the multiplicative effect of the illumination into an additive effect.

If the local feature extraction/transformation operator $\mathcal{T}_l$ satisfies the property:

$$\mathcal{T}_l L = 0$$

when L is an image whose values fit a linear model over the same scale as the operator $\mathcal{T}_l$, then $\mathcal{T}_l$ is able to remove a linear scaling of another image (*e.g.*, as occurs when a scene is illuminated). Then, the operator applied to an image of a scene V with scaling S gives:

$$\begin{aligned}\mathcal{T}_l log(VS) &= \mathcal{T}_l(log(V) + log(S)) \\ &= \mathcal{T}_l log(V) + \mathcal{T}_l log(S) \\ &= \mathcal{T}_l log(V) \qquad (3)\end{aligned}$$

if $log(V)$ is locally linear.

Appendix A demonstrates that the logarithm of the $\frac{1}{R^2}$ function is locally linear. Thus, $\mathcal{T}_l$ is able to remove the normal "$\frac{1}{R^2}$" illumination component from an image (which may also be a constant illumination, as in the case of a distant point light source). $\mathcal{T}_l$ is also able to remove the linear intensity variation arising on locally curved Lambertian surfaces. Appendix B demonstrates that the logarithm of the brightness on a curved patch is locally linear.

Examples of the $\mathcal{T}_l$ class of operator are the Laplacian, its discrete approximations, and the difference of Gaussian operators.

A second class of interesting operators is $\mathcal{T}_c$, have the property:

$$\mathcal{T}_c C = 0$$

when C is a locally constant image. Then, $\mathcal{T}_c$ is able to remove a constant illumination component from an image, as the $log(L)$ component is also constant in Eqn (3). Examples of this class of operator are the Roberts' Cross, Sobel and Canny operators, as well as the operators in $\mathcal{T}_l$.

Most local neighborhood arithmetic operators have approximately linear form. For example, doubling the values of the image pixels typically results in the doubling of output values. A typical example is the basic vertical gradient operator $\mid p_{i+1} - p_i \mid$, where p_{i+1} and p_i are the intensity of adjacent pixels. Thus, strength of output is proportional to the contrast between the pixels, and doubling the illumination doubles the pixel values, and hence doubles the contrast and gradient magnitudes.

When working in the log domain, a different interpretation arises, wherein the output value has a character more like the contrast ratio (as compared to the absolute contrast). In the case of the vertical gradient operator, the function is now:

$$\mid log(p_{i+1}) - log(p_i) \mid = \mid log(\frac{p_{i+1}}{p_i}) \mid = log(\frac{max(p_{i+1}, p_i)}{min(p_{i+1}, p_i)})$$

Doubling the pixel values now has no effect on the output.

While this theory looks promising, for most real images, the potential effects are more limited. For example, assume that there is an illumination contrast across a 256^2 image with a strong gradient creating a brightness of 255 at one edge and 128 at the other edge. The expected variation in average illumination across a 3×3 operator is $2\frac{255-128}{256} \doteq 1$, which is likely to be dominated by pixel noise. Thus, in practice, with current 8 bit cameras, there is unlikely to be any significant general benefit to using the logarithm transform. In effect, the local operators are generally too small to be affected by what is essentially a global phenomenon. On the other hand, as cameras move to 12 and 16 bits, the use of the logarithm transformation is likely to be of greater use, as then the expected variation might be more like $2(\frac{65536-32768}{256}) \doteq 128$, assuming that the noise level is on the order of a few low order bits.

However, in the course of evaluating the use of the log transform, we did observe several cases where some slight benefits might be achieved, and some where it is not advised to use the log transform.

3 Experiments

This section will look at several applications of this approach:

1. a small mask vertical gradient operator (a first-order operator), with the goal of assessing if the use of the log function does reduce the illumination gradient effects.

2. the Sobel operator (a first-order operator), with the goal of assessing the elimination of shading effects on curved surfaces.

3. a small mask horizontal dark bar operator (a second-order operator), where we investigate the effect of different strong illumination gradients on the same scene, with the goal of assessing the elimination of the illumination gradient using a second-order operator.

4. a small mask horizontal dark bar operator (a second-order operator), where we investigate the effect of different illumination levels on the same scene, with the goal of assessing the elimination of the illumination differences using a second-order operator.

In the experiments below, all input images were normal intensity images obtained by a standard CCD camera, which then had the conservative smoothing operator [2] applied to reduce point noise.

In all of the comparison experiments there is a problem of establishing the basis for comparison of an operator on two different types of images. The output values, when an algorithm is applied to the log image are much lower than when applied to an intensity image (because of the compression of input values given by the log function). The log image output also has a compression relative to the intensity values, and this requires nonlinear scaling to relate the two output images, as most operators are linear on their input data.

In addition, the log image output has a much higher noise distribution arising from quantisation effects in dark areas. Appendix C discusses this effect. The source of the problem can be seen by comparing the difference of adding 1 quanta noise to an underlying signal of 10 compared to adding it to an underlying signal of 100. In the intensity image, the difference is 1 in both cases; in the log image, the difference is $\log(11/10) = 0.09$ versus $\log(101/100) = 0.009$, or about 10 times worse at the low signal levels. In the future, if the data were from a 16 bit scanner, then there would be a larger dynamic range and so a few quanta of noise would have a smaller effect.

So, in general, to relate the two images for display and comparison, we have scaled the log image output so that the features detected in a small test image are comparable to the features detected in the corresponding intensity image (*i.e.* that the same features were extracted when the same threshold was chosen). The experiments reported here then used this rescale factor when the operators were applied to the test images.

3.1 Vertical Edge Detection

Figure 1 (right) shows an intensity image of a test pattern (left) with bars of intensity 1, 63 and 255 illuminated from the above. Figure 2 (left) shows the test image operated on by the $\mid p_{i+1} - p_i \mid$ vertical gradient operator (scaled by 4 and

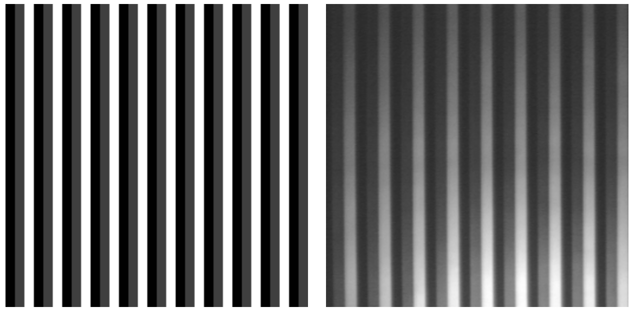

Figure 1: Synthetic and illuminated real image.

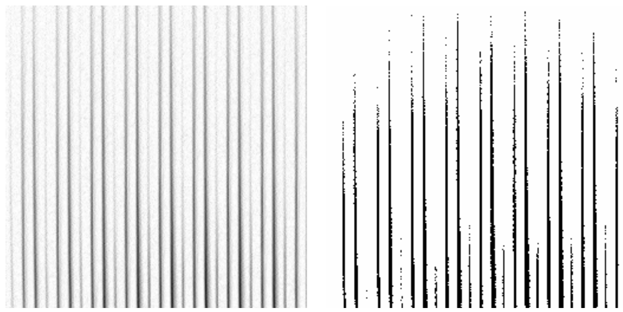

Figure 2: Vertical edge detector (inverted image) on intensity image and thresholded at 75.

clipped at 0 and 255). The right image shows the gradient image thresholded at 75.

Figure 3 shows the effect of the same operator on the log of the intensity image (scaled by 400). The right image shows the gradient image also thresholded at 75. Obviously, the choice of threshold is relatively arbitrary; however, here it was chosen to be the same as the intensity image threshold, and the operator scaling was adjusted to produce approximately the same set of edges for a thin horizontal test window across the real image. Comparison of the thresholded images shows slightly more edge detected in the log images, which arises because the effect of the illumination gradient has been reduced.

Comparison of the two unthresholded edge gradient images also verifies that the log image exposes the weaker edges at the top of the image more clearly, but at the cost of also increasing the noise.

Figure 4 (left) shows the histogram of a subset (near the edge at column 113 – out of 256) of the gradient values from the vertical gradient operator applied to the intensity image and (right) shows the corresponding histogram from the log image. The histograms verify one of the effects of the use of the log operator: the gradient magnitude values along the edge are more tightly clustered in the log image. In theory they should be constant, but in both cases, the histogram

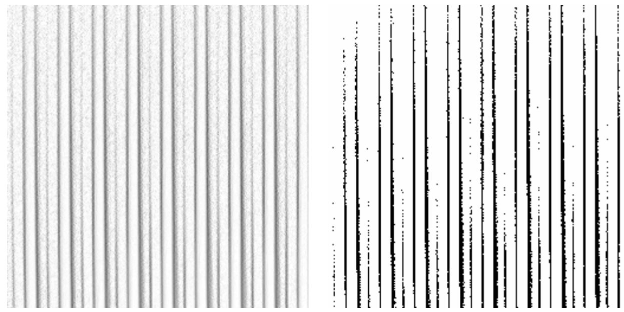

Figure 3: Vertical edge detector (inverted image) on the log of intensity image and thresholded at 75.

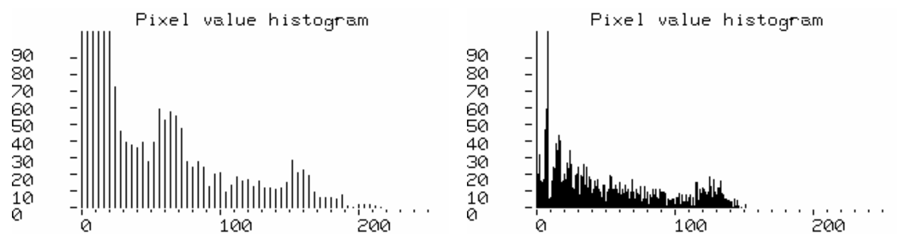

Figure 4: Histogram of intensity image vertical edge at column 109 (left) and histogram of corresponding log image edge (right).

peaks are spread out because of the aliasing between the edge position and image bluring. In the intensity image case, the gradient values are more spread out (with larger gradient values at the bottom of the image and smaller gradient values at the top of the image), resulting from the varying illumination leading to different contrasts across the edges.

3.2 Shading on Curved Surfaces

Figure 5 (left) shows a matte cylindrical surface illuminated to have shading across its surface. The middle image shows the inverted Sobel operator output on the intensity image and the right image shows the same for the Sobel operator on the log image. The log image output was scaled by 60.0 to produce a nearly identical set of edges when thresholded at the same level on a small test pattern placed near the cylinder.

In both the middle and right images, the point to note is that the smoothly shaded region on the cylinder has much lower gradient estimates in the log image. There is also a region on the right of the cylindrical surface where the measured intensities are low and so noise is increased. However, this effect is partly related to the operator scaling.

Applying a Laplacian of Gaussian operator to both the intensity and log im-

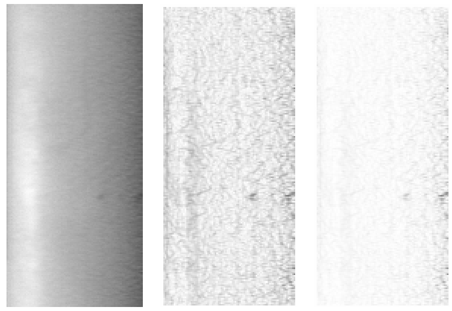

Figure 5: Original image (left), invert of Sobel on intensity image (middle), invert of Sobel on log image (right).

ages produced virtually identical (approximately) zero mean gaussian output value distributions, as predicted by Appendix B.

3.3 Horizontal bar detection with varying illumination direction and strong contrast

This experiment looked at applying a 5×5 bar detection operator:

2	2	2	2	2
-1	-1	-1	-1	-1
-2	-2	-2	-2	-2
-1	-1	-1	-1	-1
2	2	2	2	2

to the image in Figure 6 (left), with results from the intensity image in (centre) and the log image at (right). Both bar operator images are inverted for clarity. The same operator was applied to another image where the strong illumination contrast was now from the upper right (results not shown here). As seen in the two figures, there is virtually no difference in the bar detector output images.

3.4 Horizontal bar detection with varying constant illumination magnitude

This experiment applied the same 5×5 bar detection operator as in the previous experiment, except to two images with no illumination gradient, but with a large difference in average illumination levels. As seen in Figures 7 and 8, there is virtually no difference in the bar detector outputs, except that there is more noise in the dark image outputs (Figure 8), which is amplified in the log image outputs (Figure 8 (right)).

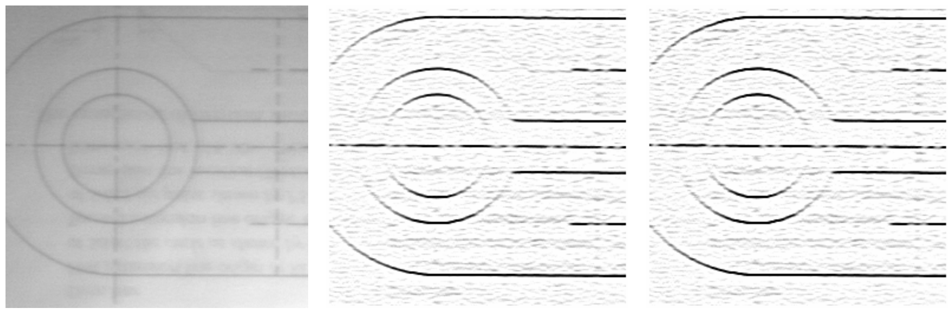

Figure 6: Bright image with strong illumination from the lower right (left), invert of bar detector on intensity image (middle), invert of bar detector on log image (right).

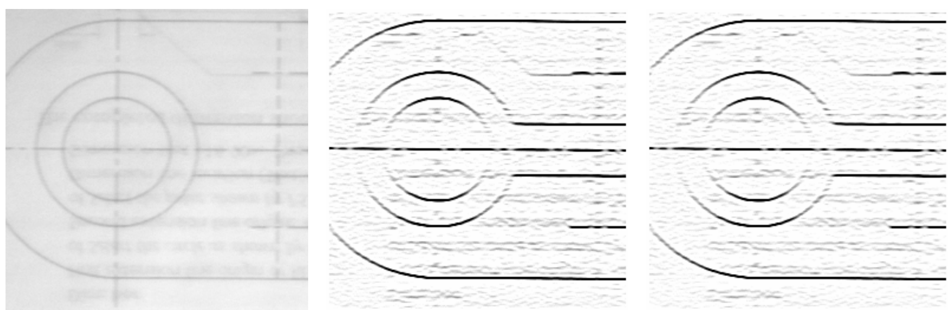

Figure 7: Bright image with constant illumination (left), invert of bar detector on intensity image (middle), invert of bar detector on log image (right).

4 Discussion and Conclusions

To summarise the results of the experiments presented above, these are the areas where operator output differs most significantly between the two approaches:

- **Strong illumination gradients**: If there is a strong illumination gradient across the image, then the output of a gradient operator on the log image is slightly less dependent upon the illumination than when applied to the intensity image. This can improve the consistency of operator results across an image, or improve the consistency of operator results between images of the same scene under changing illumination (when varying the illumination gradient direction).

- **Strong surface shading effects**: If there is a strong surface shading variation arising from oblique lighting on a curved surface, then the first-order operator outputs on the log image are more representative of the underlying scene.

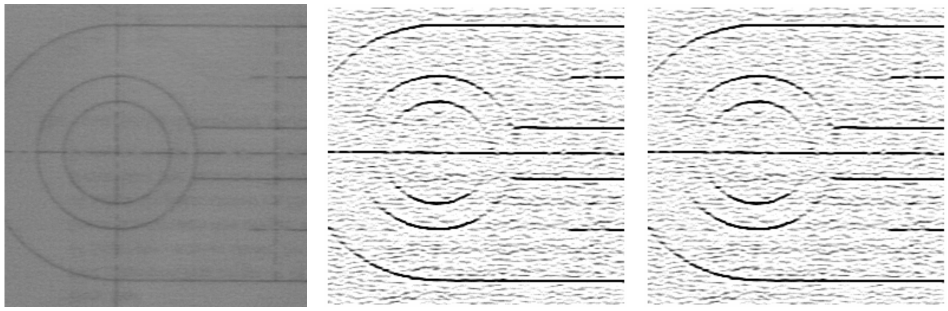

Figure 8: Dark image with constant illumination (left), invert of bar detector on intensity image (middle), invert of bar detector on log image (right).

- **Low intensity values**: If the image intensity values are low then, because of the finite quantisation of intensity values, image noise becomes much more significant in the log image.

The mathematical and empirical demonstrations in the paper assumed that the operators were being applied to smooth surface regions, where the illumination function was continuous across all pixels in an operator's input neighbourhood. This assumption is invalid across depth and orientation discontinuities and across illumination discontinuities (e.g. shadow boundaries). However, what we observed in these situations is that operator outputs did vary between the log and intensity images, but not in a significant manner. It is unclear of the consequences in image regions where surfaces are mutually illuminating.

In summary, the analysis and experiments presented above demonstrate that there are situations to consider the use of the first and second order operators on the logarithm of the intensity image, rather than on the intensity image directly. The cases where the differences are of significance are limited to special situations of strong illumination contrast, and strong shading and require bright images. The benefits might be even greater if 12 and 16 bit images become common. The paper also showed that, in more typical situations, the output images of the two approaches did not differ in obviously significant ways.

Acknowledgements

Thanks to C. Robertson for comments and suggestions.

References

[1] DH Ballard and CM Brown. Computer Vision, Prentice-Hall, New Jersey, 1982.

[2] RB Fisher, S Perkins, A Walker, E Wolfart. HIPR: Hypermedia Image Processing Reference. HTML document published on CDrom by John Wiley and Sons, Chichester, 1996.

[3] Horn, BKP, Robot Vision, MIT Press, 1986

A Logarithm of point source illumination is locally linear

Simplify the problem to lie in two dimensions, with the light source and image both lying in a plane. Suppose that we have a point light source at (0,0) and observe the illumination at points $(x-\delta, y), (x, y), (x+\delta, y)$. Then, the illumination at the three points is proportional to:

$$\frac{1}{y^2+(x-\delta)^2}, \frac{1}{y^2+x^2}, \frac{1}{y^2+(x+\delta)^2}$$

Taking the log gives:

$$-log(y^2+x^2-2x\delta), -log(y^2+x^2), -log(y^2+x^2+2x\delta)$$

The first order Taylor expansion of $log(1+x) = x$ gives:

$$-log(y^2+x^2)+\frac{2x\delta}{x^2+y^2}, -log(y^2+x^2), -log(y^2+x^2)-\frac{2x\delta}{x^2+y^2},$$

which is clearly linear for small offsets δ.

B Shading on cylindrical surface is locally linear

Simplify the problem to lie in two dimensions, with the light source direction $\vec{l} = (1, 0)$. A locally circular surface fragment with local radius R has center of curvature at the origin (0,0). The image plane is at $(0, d)$ and orthographic projection occurs perpendicular to the image plane.

Suppose that we observe the surface at points $(x-\delta, y_1), (x, y_2), (x+\delta, y_3)$. (where ${x_i}^2 + {y_i}^2 = R^2$). The surface normals $\vec{n}_i$ at these points are:

$$\frac{1}{R}(x-\delta, y_1), \frac{1}{R}(x, y_2), \frac{1}{R}(x+\delta, y_3)$$

The Lambertian shading law says that the observed brightness is proportional to $\vec{n}\cdot\vec{l}$. So, the expected brightness of the pixels is:

$$\frac{1}{R}(x-\delta), \frac{1}{R}x, \frac{1}{R}(x+\delta)$$

Thus, the brightness is locally linear. If we now look at the log image, and again apply the Taylor expansion $log(1+x) = x$, we get:

$$log(\frac{x}{R})-\frac{\delta}{x}, log(\frac{x}{R}), log(\frac{x}{R})+\frac{\delta}{x},$$

which is also locally linear for small offsets δ, but has greater variation when $x \doteq 0$ (*i.e.* when the light direction is nearly tangential at the surface boundary).

C Noise in low intensity regions is more consequential in the log image

We look at how the vertical gradient operator $\mid p_{i+1} - p_i \mid$ is affected by image noise. Assume that pixels p_{i+1} and p_i are sampled from a constant image region with mean value μ and Gaussian noise ϵ_i with variance σ^2.

Then, the expected (mean) value of $\mid p_{i+1} - p_i \mid$ is 0, and the expected variance is:

$$\sigma^2_{|p_{i+1}-p_i|} = 2\sigma^2$$

To calculate the variance of $\mid log(p_{i+1}) - log(p_i) \mid$, we first approximate this function by:

$$\begin{aligned}
log(p_{i+1}) - log(p_i) &= log(\frac{p_{i+1}}{p_i}) \\
&= log(\frac{\mu+\epsilon_1}{\mu+\epsilon_2}) \\
&= log((1+\frac{\epsilon_1}{\mu})(1+\frac{\epsilon_2}{\mu})^{-1}) \\
&\doteq log((1+\frac{\epsilon_1}{\mu})(1-\frac{\epsilon_2}{\mu})) \\
&\doteq log(1+\frac{\epsilon_1}{\mu}) + log(1-\frac{\epsilon_2}{\mu}) \\
&\doteq \frac{\epsilon_1}{\mu} - \frac{\epsilon_2}{\mu} \\
&= \frac{1}{\mu}(\epsilon_1 - \epsilon_2)
\end{aligned} \tag{4}$$

From this, we can see:

$$\sigma^2_{|log(p_{i+1})-log(p_i)|} = \frac{2\sigma^2}{\mu^2}$$

Thus, the noise in the vertical gradient of the log image is a function of the underlying intensity level, with the noise variance increasing as the brightness decreases.

The practical implication of this analysis is that, for the operator scaling that produces comparable edges selected in the intensity and log images, the background noise level in the log image is much higher than that of the intensity image in dark regions of the image, and is lower in bright regions.