Significance in Scale Space

F. Godtliebsen
Department of Mathematics and Statistics
University of Tromsø
N-9037 Tromsø, Norway

J. S. Marron
Department of Statistics
University of North Carolina
Chapel Hill, NC 27599-3260
USA

P. Chaudhuri
Indian Statistical Institute
Calcutta 700035
India

May 4, 1999

Abstract

This paper develops a methodology for finding which features in a noisy image are strong enough to be distinguished from background noise. It is based on scale space, i.e. a family of smooths of the image. Pixel locations having statistically significant gradient and/or curvature are highlighted by colored symbols. The gradient version is enhanced by displaying regions of significance with streamlines. Ideas are illustrated with simulated and real images.

1 Introduction

Smoothing methods provide one type of noise reduction in image analysis. When smoothing an image, an important question is which features that appear in a given smooth are important underlying structure, and which are spurious noise artifacts. The SiZer approach of Chaudhuri and Marron (1999) addresses this problem for one dimensional data. This is based on a scale space approach (see Lindeberg (1994) and Chaudhuri and Marron (1997) for more discussion of scale space), i.e. simultaneously viewing a family of smooths , together with a visualization for indicating regions where the smooths are significantly increasing and significantly decreasing. Here this idea is extended to two dimensional image data, and the resulting methodology is called "S^3", for "Significance in Scale Space".

An example of a noisy image is given in Figure 1. This is a Gamma Camera image of a phantom designed to reflect structure expected from cancerous bones. See Green (1990) for good discussion of the statistics of gamma camera imaging.

The cancer indicators are the bright spots on the "ribs". To get better contrast, we will investigate these in the context of the sub-image, shown in the yellow box.

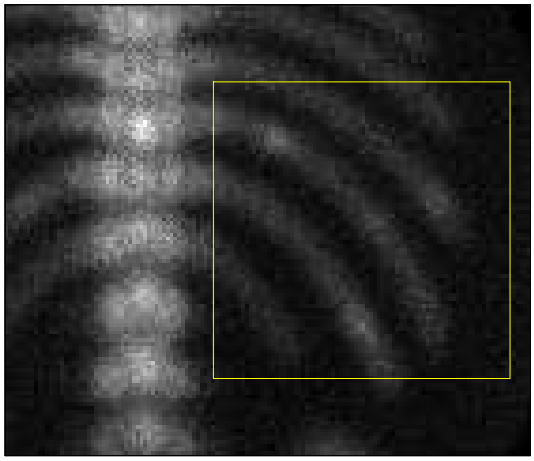

FIGURE 1: *Raw Gamma Camera Image, 120 × 140, with 80x80 subimage analyzed below highlighted.*

Smoothing of the subimage reduces the noise, and thus also better highlights the bright cancer indicators, as shown in Figure 2. These are convolutions of the yellow subimage, with spherical Gaussian window functions, having standard deviations $h = 1, 2, 4, 8$. These four images can be viewed as slices of the scale space for the image of Figure 1. For a better impression of the scale space, view the MPEG movie (these images, together with a denser grid of h values made into frames of a movie) in the file `sss1fig2.mpg` available at the web address: http://www.unc.edu/depts/statistics/postscript/papers/marron/SSS_paper/.

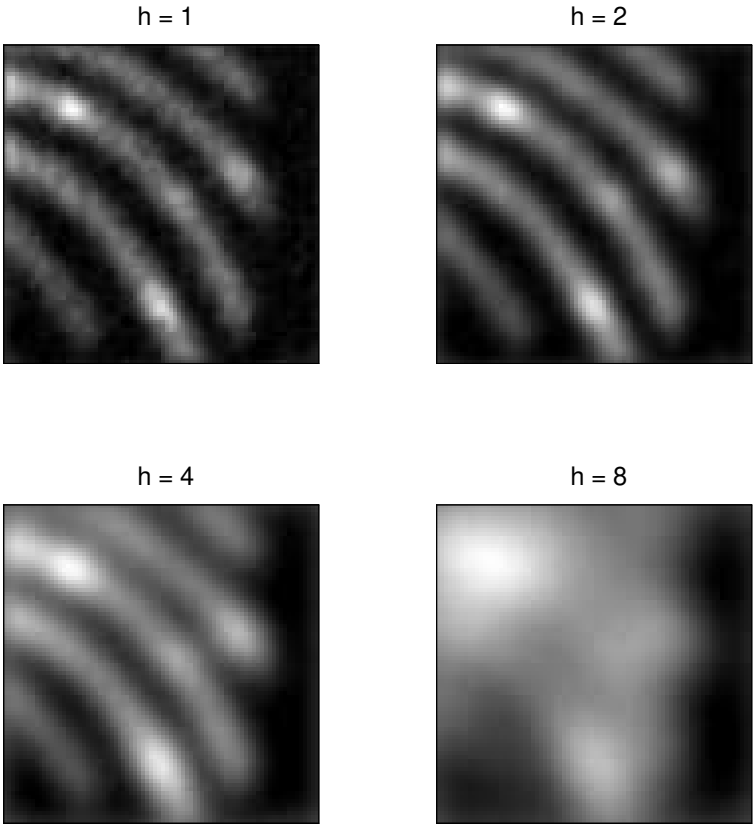

FIGURE 2: *Gaussian window smooths of data from subimage of Figure 1. These highlight bright spots on ribs.*

Note that at $h = 2$ and 4 bright spots are visible in the upper left, the middle right and below the center. There is a questionable bright spot quite close to center of the image. At $h = 1$, the bright spots are less visible, because there is substantial noise which degrades the image. At $h = 8$, the image is oversmoothed, so that even the ribbed structures are not visible. These features are typical in scale space. At the finer scales, there is usually a large amount of noise, at very coarse scales interesting structures are smoothed away.

In some image situations, noise can be reduced by improved imaging technology. However, in many others, noise is endemic and must be dealt with statistically. For the gamma camera example, there is a trade-off between image noise and the amount of radioactive isotope being introduced to the body (more isotope gives a sharper image, but is more harmful). In MRI, a similar trade-off occurs in terms of slice thickness: thick slices have less noise, but the resulting image in unduly influenced by surrounding tissue. In functional MRI, where movies are made, there is an additional dilemma: frames made in a shorter time give better time resolution, but are more noisy.

A great deal of effort has been devoted to the problem of image denoising. For access to the literature see for example Jain (1989) and Winkler (1995). In this paper we take a different viewpoint. Instead of trying to find an "optimal denoising", we address the perhaps more important question of "which features visible in the image represent important underlying structure, and which cannot be distinguished from background noise?". Our approach is via graphical devices which illustrate statistical significance of image features.

One method of highlighting such features by S^3 comes from studying the gradient (i.e. the "local slope") at each pixel. For example a "statistically significant peak" is surrounded by at least a ring of significant gradients pointing towards the peak. An implementation of this is shown in Figure 3, using the gamma camera data from Figures 1 and 2. Because different features can appear as significant at different scales, i.e. at different levels of resolution of image denoising, it is very important to look simultaneously at several scales, i.e. levels of smoothing of the image. So again four levels of smoothing are shown in Figure 3, and a denser family of smooths can be seen in the MPEG movie version, in the file `sss1fig3.mpg` at the above web address.

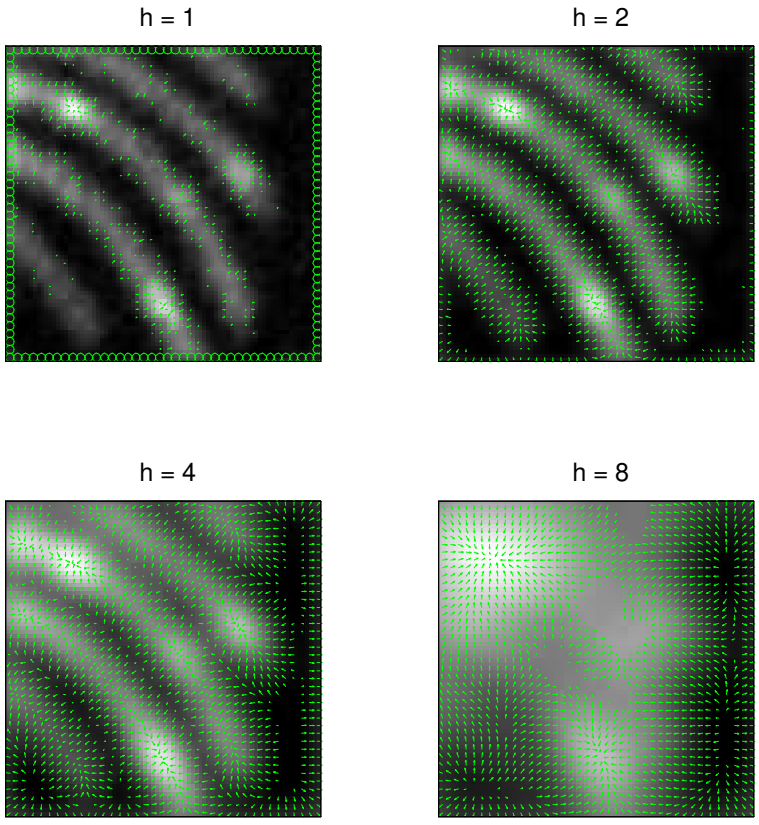

FIGURE 3: *Gradient based S^3 overlayed on family of smooths in Figure 2. Arrows show gradient at locations where it is statistically significant.*

The $h = 1$ smooth in Figure 2 still has substantial noise. The individual pixel gradients of that image are very unstable, because of this noise. Due to this instability, only a very few of them contain statistically significant information. This is reflected by rather few green arrows appearing in the $h = 1$ part of Figure 3. There are a few locations at rib boundaries, where the gradient is steep enough to be seen through the noise, shown as green arrows pointing from the dark regions towards the light regions. The three clear bright spots, already appear as significant, with nearly complete rings of arrows pointing towards them. The unclear bright spot, just to the right of the center is highlighted by more arrows than other rib locations, but its statistical significance is clearly not as strong. At the $h = 2$ level of resolution, all the ribs are now very clearly defined by the arrows. The bright spots have arrows on the ridges of the ribs pointing towards them. Again the less bright spot just to the right of the center is highlighted in the same way, but is less marked. At the smoothing level $h = 4$, the valleys between ribs are less visually apparent, and the ridges are now strong enough for arrows to run parallel to their crest. The bright spots are now at peaks of these ridge arrows. The $h = 8$ level of resolution shows lots of significant structure, but this is not particularly useful, since the features of interest are no longer visible at this level.

Details and variations on the gradient version of S^3 shown in Figure 3 are given in Section 2.4 Current thought on visualization of vector fields is that matrices of arrows as shown in Figure 3, are less effective than other approaches, such as "streamlines". A streamline version of this S^3 concept of the "significant gradient" is developed in Section 3.2.

Another S^3 method of highlighting statistically significant features in images is based on local curvature. Figure 4 shows an example of this, for the same data as above. Curvature can be statistically significant in several ways, which are indicated by different colored dots overlaid on the image. While the entire family of different scales should again be considered, only one is shown here to save space. An MPEG version of the whole family of smooths is available at the above location in the file `sss1fig4.mpg`.

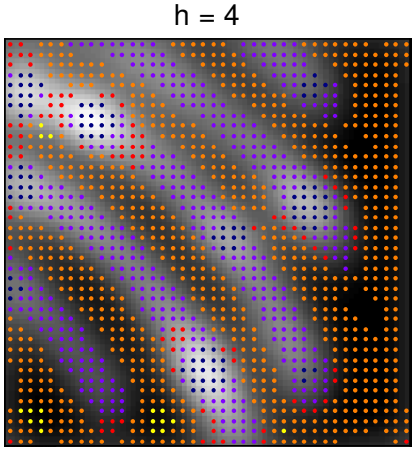

FIGURE 4: *Curvature based S^3 overlayed on a single smooth of the same subimage of the Gamma Camera Data. Dot colors indicate type of statistically significant curvature.*

The different types of significance of curvature are best understood through the eigenvalues of the Hessian matrix. At locations where one eigenvalue is significantly negative, and the other is not significantly different from 0, a purple dot is used, many of which are on the crests of the ridges formed by the ribs in Figure 4. When both eigenvalues are significantly negative, a dark blue dot is used, which appears only near the bright spots at the upper left. When one eigenvalue is significantly positive, an orange dot is used, as seen in the long valleys between the ribs. At "saddle points", one eigenvalue is significantly positive, and the other is significantly negative, and a red dot is used. Most ridge points appear as purple, but the bright parts of interest are highlighted as dark blue. Even the doubtful bright spot just to the right of the image center shows up well in this way. Details and variations on the curvature version of S^3 are given in Section 2.5.

Both the gradient and curvature versions of S^3 are useful. One reason is that some data analysts may feel they have better intuition about one notion of "feature", than about the other one. Another reason is that sometimes one method shows particular features better than the other. The two methods are combined into one visualization scheme in Section 3.1.

Development of the main mathematical and statistical structure of S^3 is done in Section 2. A number of variations, such as combining gradient and curvature versions of S^3, streamlines, and movie and slider versions are discussed in Section 3. Various statistical aspects of S^3 are analyzed via some simulated examples in Section 4. Derivations of technical formulas are done in the Appendix, Section 6.

More real data examples are given in Section 5. In addition to the gamma camera image treated in this section, additional real images analyzed in this

paper include optical images of electrical activity, perfusion MRI, and confocal microscopy. We believe these methods will prove to be widely applicable to many other types of images.

2 Mathematical Development

The statistical model underlying S^3 is

$$Y_{i,j} = s(i,j) + \epsilon_{i,j},$$

where $i = 1, ..., n$ and $j = 1, ..., m$ index pixel locations, where s represents the underlying signal (thought of as a smooth underlying function evaluated at a rectangular grid), and where the $\epsilon_{i,j}$ are the noise, assumed to be independent random variables. The gray level smooths shown above are simply Gaussian smooths, i.e. discrete 2-dimensional convolutions of a spherically symmetric Gaussian density, with the data, denoted

$$\widehat{s}_h(i,j) = \sum_{i'=1}^{n} \sum_{j'=1}^{m} Y_{i',j'} K_h\left(i - i', j - j'\right), \tag{1}$$

or in obvious matrix notation

$$\underline{\widehat{s}}_h = \underline{K}_h * \underline{Y}$$

where

$$K_h(i,j) = K_h\left(i\right) K_h\left(j\right),$$

for $i = (1-n), ..., (n-1)$ and $j = (1-m), ..., (m-1)$, where

$$K_h(i) = \frac{\exp\left(-(i/h)^2/2\right)}{\sum_{i'=1-m}^{m-1} \exp\left(-(i'/h)^2/2\right)}. \tag{2}$$

2.1 Boundary Effects

A drawback to this type of smooth is that it can suffer severe boundary effects. A way to understand this is to think of the data as being "infinitely continued by zero padding". I.e. extended to an infinite matrix, with the additional entries all having the value 0. When the $Y_{i,j}$ values near a boundary are far from 0, the smoothing process will average them with 0, which can seriously distort the image.

This problem is demonstrated in Figure 5, which shows an optical image that reflects electrical activity in an animal's brain. The upper left plot is the raw data, and it is apparent that the noise level is quite high. The upper right shows the $h = 4$ smooth of these data, $\widehat{s}_h$, using the formula (1). Note that at the edges, the color becomes quite dark. The reason is that zeros outside the

image are averaged in. The problem gets worse for larger bandwidths, which can be seen in the movie version of this, in the file `sss1fig5a.mpg` at the above web address. There are many ways to address this boundary problem, but for simplicity (especially in the later development involving derivatives) in our examples, we subtract the mean of the $Y_{i,j}$ before smoothing, i.e. our estimate becomes

$$\widehat{\underline{s}}_h = A\left(\underline{Y}\right) + \underline{K}_h * \left(\underline{Y} - A\left(\underline{Y}\right)\right),$$

where A is the matrix operator which returns the constant matrix whose common entries are the average of the entries of its matrix argument, i.e. each

$$A\left(\underline{Y}\right)_{i,j} = \frac{1}{nm} \sum_{i'=1}^{n} \sum_{j'=1}^{m} Y_{i',j'}. \tag{3}$$

The effect of this is shown in the lower right panel of Figure 5. The clearly important features, such as the dark areas in the upper left and lower right corners, show up more clearly with the boundary adjustment. The movie version of this family of smooths can be seen in the file `sss1fig5b.mpg`. Unlike the unadjusted movie, it is seen that the important bright and dark regions remain in the image even for bandwidths up to $h = 8$.

There are many more sophisticated boundary adjustment methods. These include the boundary reflection approach, where "mirror images" of the data are created at each boundary, the Nadaraya-Watson approach, where one divides by the sum of the kernel weights, and also various local polynomial approaches. These are not pursued here, because they become quite complex when estimating derivatives, as needed for S^3.

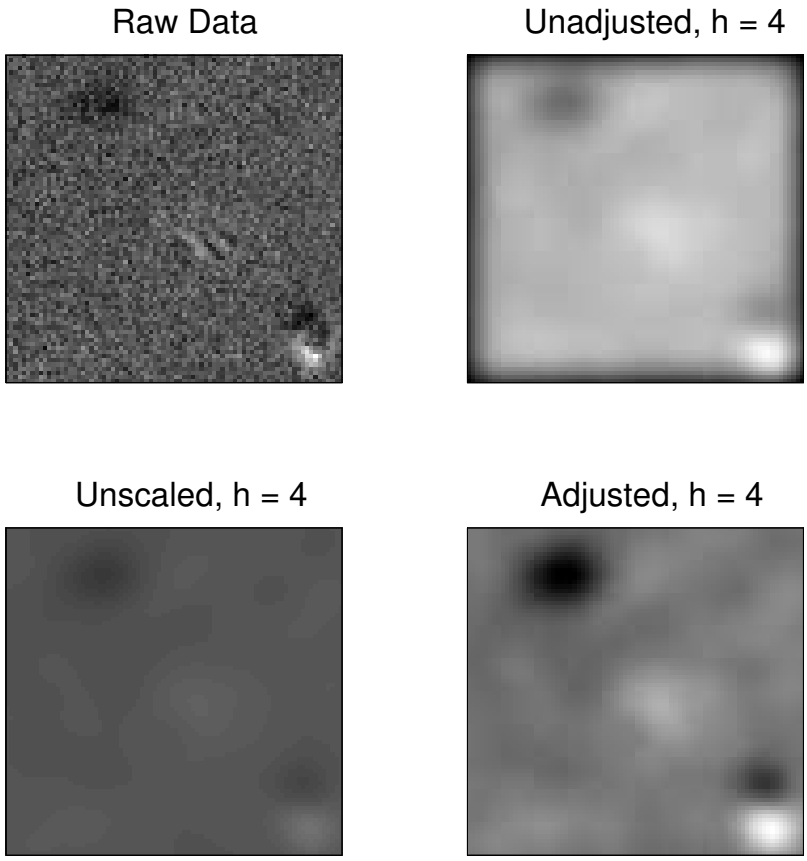

FIGURE 5: *Image showing elecrical activity. This illustrates boundary effects, and scaling issues.*

Another important graphical device used in all examples in this paper is to "stretch each image to use the full gray scale". The lower left part of Figure 5 shows why this is important. It is the same $h = 4$ boundary adjusted smooth as in the lower right, but the color scale is the same as for the raw data in the upper left. Note the contrast is much poorer, because much of the gray scale is unused.

These adjustments were used in most the examples in this paper.

2.2 Effective Sample Size

An important component of S^3 is the number of points inside each kernel window, called the "Effective Sample Size",

$$\underline{ESS} = (\underline{K}_h * \underline{1}) / (K_h(0,0)),$$

where $\underline{1}$ is the n by m matrix having a one in each entry, and where the denominator is the rescaling that assigns value one to the pixel in the center,

and appropriately downweighted values to the other pixels. When h is large, $ESS(i,j)$ is large, and vice versa for small h. In boundary regions, the above boundary effect works to give an appropriately small value of $ESS(i,j)$.

One use of the Effective Sample Size is to highlight regions where the data are too sparse for S^3. Using the standard Binomial rule of thumb "the normal approximation works when $np \geq 5$", regions where there less than 5 points in each kernel window, i.e. where $ESS(i,j) < 5$, are highlighted with a green circle. Such points only occur at the boundary, for bandwidths h as small as 1. This is shown in Figure 11b below.

Following the development in Section 3 of Chaudhuri and Marron (1999), ESS can also be used to make the inferences of S^3 simultaneous across location. The basis is the average Effective Sample Size

$$\overline{ESS} = \left(\sum_{i=1}^{n} \sum_{j=1}^{m} ESS(i,j) \right) /(nm).$$

Since there are nm independent data points, the smoothing process can be viewed as "averaging in groups of size $\overline{ESS}$. Thus the number of independent averages is approximately

$$\ell = \frac{nm}{\left(\overline{ESS}\right)}$$

As noted in Section 3 of Chaudhuri and Marron (1999), ESS has a strong relationship to the "effective degrees of freedom" of Hastie and Tibshirani (1990).

2.3 Variance Estimation

Another integral part of S^3 is an estimate of the noise level, i.e. of the variance of the $\epsilon_{i,j}$. How this should be done, depends on the particular application. In some situations, such as MRI, this noise level is quite well known as discussed in Godtliebsen (1991), so that value should be used. In other cases, the variance is unknown, but it is reasonable to assume that it is a constant value, σ^2. In still other cases, the variance may be quite different at different locations, i.e. one should consider $\sigma^2_{i,j}$, but it is assumed that the local variance varies smoothly with location.

In the heteroscedastic case, $\sigma^2_{i,j}$ can be estimated by smoothing the squared residuals. Again boundary issues are important, so it is recommended to first subtract the mean of the squared residuals. To make the estimated variance unbiased, we also make the classical "$\frac{1}{n}$" to "$\frac{1}{n-1}$" type of adjustment. This results in

$$\underline{\widehat{\sigma}}^2_h = \underline{ESSQ} \cdot \left\{ A\left(C_S\left(\underline{Y} - \underline{\widehat{s}}_h\right)\right) + \underline{K}_h * \left[C_S\left(\underline{Y} - \underline{\widehat{s}}_h\right) - A\left(C_S\left(\underline{Y} - \underline{\widehat{s}}_h\right)\right)\right] \right\}, \tag{4}$$

where $\underline{ESSQ}$ is the matrix with

$$\frac{ESS(i,j)}{ESS(i,j)-1}$$

in entry (i,j), where $\cdot$ denotes element by element matrix multiplication, where C_S is the matrix operator which squares all entries of its matrix argument, and where, as in (3), A is the matrix operator which returns the constant matrix whose common entries are the average of the entries of its matrix argument.

In the homoscedastic case, these estimates are pooled to estimate the common σ^2. Since interior points have a more stable $\widehat{\sigma}_h^2(i,j)$, an ESS weighted average is used,

$$\widehat{\sigma}_h^2 = \left(\sum_{i=1}^{n} \sum_{j=1}^{m} ESS(i,j) \widehat{\sigma}_h^2(i,j) \right) / \left(\sum_{i=1}^{n} \sum_{j}^{m} ESS(i,j) \right). \tag{5}$$

This pooled variance estimate is used for the examples shown in this paper.

2.4 Significant Gradient

At a given pixel location, indexed by (i,j) (which will be suppressed for simplicity of notation), the gradient of the underlying signal s is

$$G(s) = \left[(s_1)^2 + (s_2)^2 \right]^{1/2},$$

where s_1 is the partial derivative in the vertical direction (indexed by i) and s_2 is the partial derivative in the horizontal direction (indexed by j). The corresponding estimate of the gradient is

$$\widehat{G}_h(s) = \left[(\widehat{s}_{h,1})^2 + (\widehat{s}_{h,2})^2 \right]^{1/2},$$

where the partial derivatives are estimated by

$$\begin{aligned} \underline{\widehat{s}}_{h,1} &= \underline{K}_{h,1} * \underline{Y}, \\ \underline{\widehat{s}}_{h,2} &= \underline{K}_{h,2} * \underline{Y}, \end{aligned}$$

where

$$\begin{aligned} K_{h,1}(i,j) &= K_h'(i) K_h(j), \\ K_{h,2}(i,j) &= K_h(i) K_h'(j), \end{aligned}$$

using the notation (2).

The gradient version of S^3 flags pixels as significant when $\widehat{G}_h(s)$ is "higher than the noise level", in the sense that it rejects a hypothesis of the form

$$H_0 : G(s) = 0. \tag{6}$$

The null distribution of this test is based on the bivariate Gaussian distribution

$$\begin{pmatrix} \widehat{s}_{h,1} \\ \widehat{s}_{h,2} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \sigma_{12}^2 \\ \sigma_{12}^2 & \sigma_2^2 \end{pmatrix} \right), \tag{7}$$

which is exact if the noise terms $\epsilon_{i,j}$ have a Gaussian distribution, or follows by an appropriate Central Limit Theorem otherwise. Approximate values for σ_1^2, σ_{12}^2 and σ_2^2 are derived in Section 6.1, where it is also seen that $\sigma_{12}^2 \approx 0$. Thus

$$\frac{\widehat{s}_{h,1}^2}{\sigma_1^2} + \frac{\widehat{s}_{h,2}^2}{\sigma_2^2} \sim \chi_2^2,$$

so the null hypothesis (6) is rejected for those pixels where

$$\frac{\widehat{s}_{h,1}^2}{\widehat{\sigma}_1^2} + \frac{\widehat{s}_{h,2}^2}{\widehat{\sigma}_2^2} > q_{\chi_2^2}(\alpha'),$$

the appropriate quantile of the χ_2^2 distribution, using the estimates $\widehat{\sigma}_1^2$ and $\widehat{\sigma}_1^2$ as defined in (14) in the Appendix.

The probability α' is chosen to make the inference simultaneous across pixel values. This could be done with the Bonferroni inequality, but a better approximation is based on the "number of independent blocks" , ℓ, as developed in Section 2.2. To achieve the nominal level of α ($\alpha = 0.05$ is used in most examples shown in this paper) simultaneously over ℓ independent hypothesis tests, we need

$$\begin{aligned} \alpha &= P\left[\text{not } R_k, \text{ for some } k = 1, ..., \ell\right] = \\ &= 1 - P\left[R_k, \text{ for all } k = 1, ..., \ell\right] = \\ &= 1 - P\left[R_1\right]^\ell = 1 - (1 - \alpha')^\ell, \end{aligned}$$

where R_k is the event that the k-th hypothesis is accepted. Thus $\alpha' = 1 - (1 - \alpha)^{1/\ell}$. Since the χ_2^2 distribution is the exp(1/2) distribution, with c.d.f. $F(x) = 1 - \exp(-x/2)$, it follows that

$$q_{\chi_2^2}(\alpha') = -2 \log\left(1 - (1 - \alpha)^{1/\ell}\right).$$

For pixels where the hypothesis (6) is rejected, an arrow is drawn in the gradient direction, i.e. using the direction vector based on $\begin{pmatrix} \widehat{s}_{h,1} \\ \widehat{s}_{h,2} \end{pmatrix}$. Some experimentation, in the context of 64×64 pixel images printed four on a page, with the length of the vector, suggested that this should be 1.2 times the pixel size. For images around the size of 64×64, the arrows are a little short to provide a good indication of direction. One solution is to go to larger images, putting one on each printed page. Another is to "combine gradient information on 2×2 blocks". The image is partitioned into such blocks, and all 4 hypothesis tests are run. The direction vector representing the block is based on the component-wise average of the 4 partial derivative vectors. The length of the arrow is scaled according to how many of the four hypotheses are rejected. This summarization was used in the 80×80 S^3 analysis shown in Figure 3. Using both the single page approach, and 2×2 pixel blocks, we were able to consider images up to size 256×256, which was quite slow. For larger images, we

recommend applying S^3 to sub images (to smaller regions of interest, as done in Figure 1), although it is possible to extend the combination idea to larger blocks.

2.5 Significant Curvature

At a given pixel location, indexed by (i, j) (which will again be suppressed for simplicity of notation), the local curvature properties of the underlying signal s are described by the Hessian matrix,

$$\underline{H}(s) = \begin{pmatrix} s_{11} & s_{12} \\ s_{12} & s_{22} \end{pmatrix},$$

where s_{11} is the second partial derivative in the vertical direction (indexed by i), s_{22} is the second partial derivative in the horizontal direction (indexed by j), and s_{12} is the mixed partial derivative. These second order partial derivatives are estimated by

$$\begin{aligned} \widehat{\underline{s}}_{h,11} &= \underline{K}_{h,11} * \underline{Y}, \\ \widehat{\underline{s}}_{h,12} &= \underline{K}_{h,12} * \underline{Y}, \\ \widehat{\underline{s}}_{h,22} &= \underline{K}_{h,22} * \underline{Y}, \end{aligned} \tag{8}$$

where

$$\begin{aligned} K_{h,11}(i,j) &= K_h''(i) K_h(j), \\ K_{h,12}(i,j) &= K_h'(i) K_h'(j), \\ K_{h,22}(i,j) &= K_h(i) K_h''(j), \end{aligned}$$

using the notation (2). The eigenvalue representation of the Hessian matrix gives a "coordinate free" representation of the local curvature, and in particular the eigenvalues are a simple summary of the types and magnitudes of local curvature. In two dimensions, the eigenvalue problem has the explicit solution

$$\lambda_{\pm} = \left[(s_{11} + s_{22}) \pm \sqrt{(s_{11} - s_{22})^2 + 4 s_{12}^2} \right] / 2. \tag{9}$$

The curvature version of S^3 flags pixels as significant when there is "some significant curvature", in the sense that at least one of λ_+ and λ_- "emerges from the noise level". This can be quantified through the parameter $T = \max\{|\lambda_+|, |\lambda_-|\}$, and thus is formulated as the null hypothesis

$$H_0 : T = 0. \tag{10}$$

An alternate choice for the parameter T that we considered was the Euclidean distance to the origin, $\sqrt{\lambda_+^2 + \lambda_-^2}$, but this was less suitable because the shape of the null region did not allow easy separation into types of curvature, as given

in (13). The natural statistic for the test of (10) is $\widehat{T} = \max\left\{\left|\widehat{\lambda}_+\right|, \left|\widehat{\lambda}_-\right|\right\}$ where the eigenvalues of the estimated Hessian matrix come from replacing the second derivatives in (9) with the estimated versions given in (8).

The null distribution of this test is based on the trivariate Gaussian distribution

$$\begin{pmatrix} \widehat{s}_{h,11} \\ \widehat{s}_{h,12} \\ \widehat{s}_{h,22} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \sigma_{13}^2 \\ \sigma_{12}^2 & \sigma_{22}^2 & \sigma_{23}^2 \\ \sigma_{13}^2 & \sigma_{23}^2 & \sigma_{33}^2 \end{pmatrix}\right), \tag{11}$$

which is exact if the noise terms $\epsilon_{i,j}$ have a Gaussian distribution, or follows by an appropriate Central Limit Theorem otherwise. Approximate values for σ_{11}^2, σ_{12}^2, σ_{13}^2, σ_{22}^2, σ_{23}^2 and σ_{33}^2 are derived in Section 6.2, where it is seen that $\sigma_{12}^2 \approx \sigma_{23}^2 \approx 0$ and $\sigma_{11}^2 \approx \sigma_{33}^2 \approx 3\sigma_{13}^2 \approx 3\sigma_{22}^2 \approx \frac{3\sigma^2}{16\pi h^6}$. Thus an approximate distribution for $\widehat{T}$ is based on the trivariate normal distribution

$$\begin{pmatrix} \widehat{s}_{h,11}/\widehat{\sigma}_C \\ \widehat{s}_{h,12}/\widehat{\sigma}_C \\ \widehat{s}_{h,22}/\widehat{\sigma}_C \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 3 \end{pmatrix}\right), \tag{12}$$

where the combined estimate $\widehat{\sigma}_C$ is also developed in Section 6.2. The distribution of $\widehat{T}$ does not seem to have a simple closed form, but it can be developed by simulation. Details are given in the Appendix, Section 6.3. Given a nominal level α', the null hypothesis (10) is rejected for a given pixel when $\widehat{T} > \widehat{q}_{\widehat{T}}$, the simulated α' -th upper quantile of this distribution.

As in Section 2.4, the probability α' is chosen to make the inference simultaneous across pixel values. The same development shows that for a simultaneous level of α, use $\alpha' = 1 - (1-\alpha)^{1/\ell}$.

For pixels where the hypothesis (10) is rejected, a colored dot is used, where color codes the type of significant curvature as:

color	**feature**	**characterization**
yellow	hole	$\widehat{\lambda}_+, \widehat{\lambda}_- > \widehat{q}_{\widehat{T}}$
orange	long valley	$\widehat{\lambda}_+ > \widehat{q}_{\widehat{T}}, \left\|\widehat{\lambda}_-\right\| < \widehat{q}_{\widehat{T}}$
red	saddle point	$\widehat{\lambda}_+ > \widehat{q}_{\widehat{T}}, \widehat{\lambda}_- < -\widehat{q}_{\widehat{T}}$
purple	long ridge	$\left\|\widehat{\lambda}_+\right\| < \widehat{q}_{\widehat{T}}, \widehat{\lambda}_- < -\widehat{q}_{\widehat{T}}$
dark blue	peak	$\widehat{\lambda}_+, \widehat{\lambda}_- < -\widehat{q}_{\widehat{T}}$

(13)

These regions are easily understood by an x - y plot, where the axes are λ_+ and λ_-. The boundaries of the regions are the vertical and horizontal lines at $\pm\widehat{q}_{\widehat{T}}$.

The same problem with small pixel size that appeared in Section 2.4 happens here also for larger images. The problem again can be addressed by "combining curvature information on 2×2 blocks", although the combination is more complicated, because of the different colors involved. Details are given in the Appendix Section 6.4.

3 Variations

3.1 Combining Gradient and Curvature

The above visualizations for significant gradient and curvature can be combined into a single image as follows. At each pixel location, when the gradient is significant, draw an arrow, as in Section 2.4. If the curvature is not significant, color the arrow green, otherwise color it according to the type of significant curvature as developed in Section 2.5. If the gradient is not significant, but the curvature is, then use a colored dot.

Figure 6a shows the results of this for the Gamma Camera Data, from Figure 2. Other S^3 versions were shown in Figures 3 and 4, and this is essentially a combination of those. Because of the strong features in these images, most locations have a colored symbol. But the bright spots are somewhat more accentuated, because of the combination of the arrow directions, and the curvature colors. Note that the yellow dots at the bottom, just to the left of center, clearly highlight a local minimum there. Also the long valley running up the right side is quite visible as a series of orange dots.

A drawback to this visualization in S^3 is that there is a great deal of information present, and it takes some practice and experience to absorb it all. We suggest that first time users start with just gradient arrows, or just curvature dots.

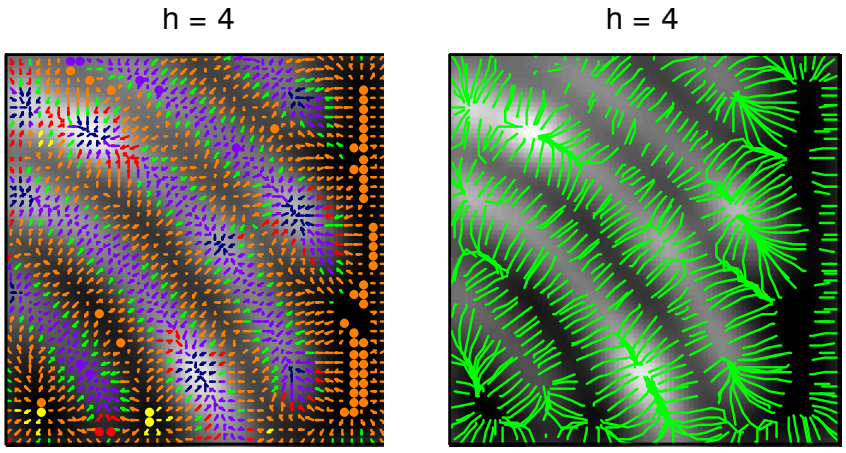

FIGURE 6: *For the Gamma Camera Data, (a) the combined gradient and curvature version of S^3 (b) the streamline version of S^3.*

A movie version of Figure 6a is available at the above web address. Figure 6b is discussed in the next section.

3.2 Streamlines

As noted for example in Helman and Hesselink (1989), matrices of arrows (as shown in Figure 3) are not particularly effective ways of visually presenting vector fields. In addition to being generally difficult to comprehend, there is also a "raster effect", i.e. vertical and horizontal arrows give a different impression and texture than arrows at other angles. A standard approach to this problem is "streamlines", which are essentially curves indicating the gradient direction. These indicate the structure of a surface, via their physical interpretation of the path that a drop of water would take in flowing downhill. In Figure 6b, this idea is adapted to the S^3 context, by drawing the lines only over pixels, where the gradient is significant, as defined in Section 2.4.

Note that Figure 6b conveys the significant slope information much more cleanly and easily than does the $h = 4$ part of Figure 3. In particular, significant ridges show up as streamlines running together. The important bright spots have such ridges on either side. The less clear bright spot, just to the right of the center has just a small suggestion of this "running together".

The main idea behind the construction of these streamlines is to start at a random point where the gradient is significant, and then take small steps in both (uphill and downhill) gradient directions, ending when either a region of nonsignificance or a peak/valley or a boundary is reached. The greatest challenge in the implementation is the "random" selection of the starting point. A uniform distribution is clearly inappropriate, because the random "clumping" that occurs gives an unpleasant and distracting visual effect. This problem is an ongoing research area in computer graphics.

On the basis of some experimentation, we recommend the following for use in S^3. The algorithm is a series of steps, where each step consists of the drawing of one streamline. The starting point for each streamline is a pixel with significant gradient chosen to make the next streamline "far" from the others. This choice is based on how many times each pixel with significant gradient has been touched by a streamline. If there is a significant gradient pixel that has had the minimum number of touches, that is chosen. When there is tie for the minimum number of touches, then all significant pixels are considered, and a nonuniform random choice is made. A random choice that worked well was for each significant pixel, to use a probability proportional to 100^{-nt} where nt is the number of "neighborhood streamline touches". The neighborhood is the 3×3 block centered at the given pixel, so nt is the sum over the 9 nearest (inclusive) neighbors of the streamline touches for the given pixel. Note that this distribution puts much heavier weight on pixels in regions with relatively few streamline touches, which thus gives good "spread" of streamlines.

In growing each streamline in the two gradient directions, an important tuning parameter is the step size. After some experimentation, we found that a factor of 0.5 times the pixel width worked well. For ending each streamline, we use the first to occur among three possible conditions. Ending Condition 1 is that the next step in the streamline construction lands in a region where the gradient is no longer significant. In that case, the streamline is extended in

that direction, but only to the edge of the pixel boundary. Ending Condition 2 is that the next step in the streamline construction extends outside the image. Here again the streamline is continued in that direction up to the edge. Ending Condition 3 is that the streamline passes a peak or valley. It is important to end the streamline when this happens, or else, the construction will continue to oscillate around the extremum, in a visually uninsightful way. A simple check for a crossing of an extremum is the angle between the present step and the previous one. If this angle is greater than $90^{\diamond}$, then streamline construction is stopped, and that step of the streamline is not shown. This results in streamlines not quite coming together at extrema, which we view as preferable to slight overlaps in such regions.

For ending the process, we chose to study the average, over significant gradient pixels, of the number of streamline touches. More experimentation resulted in a good visual impression when this average was 2, for 16×16 images. For larger images, this number should be proportional to the image size, so we recommend stopping when this average becomes $32/\min(n, m)$.

This version of streamlines should be viewed as only a workable first attempt. There are many ways in which this construction can be fine-tuned, and probably major improvements are also available. We suggest this as an interesting area for future research.

3.3 Movies and Sliders

A weakness of the figures shown in this paper, is that they only show S^3 at one or a few scales, i.e. a few levels of smoothing. There is much more information available about the data from studying other scales, which is easily obtainable by watching movies, where time is indexing the scale. The use of time as an analog for scale is also the basis of the heat diffusion model for smoothing, as discussed in Figure 3 of Chaudhuri and Marron (1998). As noted above, MPEG movies of many of these figures are web available, and we view these as an important data analytic device.

A problem with movies is that it is difficult to make a precise connection between time and the scale, i.e. bandwidth of the smooth. To address this problem, we have also developed a "slider" version of S^3 where the user can carefully control the scale, and thus fine-tune that parameter. Our current implementations of the movie and slider versions of the streamline variation, are only a clumsy first attempt, because we have not yet invested the effort required to avoid the streamlines randomly changing from frame to frame. This is another area for future work.

4 Simulated Examples

We have tried the various versions and variations of S^3, for a number of different simulated examples. In this section, the more important lessons and conclusions are summarized.

The relative performance of the variance estimates from Section 2.3 was considered by simulation. In particular, we tried S^3 using the true known variance σ^2, using the local variance estimate $\widehat{\sigma}_h^2(i,j)$ from (4), and using the pooled variance estimate $\widehat{\sigma}_h^2$ from (5). Examples are not shown here because they all look quite similar, and thus didn't seem worth the space. The very slight differences were in predictable directions. The pooled estimate $\widehat{\sigma}_h^2$ showed a very few more significant pixels than the true variance σ^2, since there is slight underestimation at the boundaries. The local estimate $\widehat{\sigma}_h^2(i,j)$ was on average comparable to the pooled estimate $\widehat{\sigma}_h^2$, but there was some systematic variation, with the local version finding slightly fewer features in regions of higher curvature in the underlying signal s (since bias inflates the estimate in such regions), but slightly more features in regions of less curvature. The pooled estimate was used in all examples shown in this paper.

A parameter of interest is the significance level α. This has less effect on the resulting picture than one might guess at first, as shown in Figure 7. The underlying signal s is a linear combination of Gaussian densities, with a large circular peak in the upper left, two elongated peaks in the lower right, and three smaller valleys, called "Peaks and Valleys". An image plot of this underlying target, together with several other images which were omitted here to save space, can be found in the file `sss1talk.ps` at the above web directory. The image size is 64×64, the height is normalized so the range of s is $[0,1]$, and Gaussian noise with standard deviation $\sigma = 0.16$ is added. Figure 7a clearly shows that all of the peaks and valleys are "really there". In particular, each of the valleys has some yellow regions, and each peak has some dark blue regions. Because the significance level α is much larger, it is not surprising that Figure 7b suggests that more features are significant. But it is perhaps surprising that so few additional regions are seen to be significant. This occurred for a number of examples, for a broad range of bandwidths, and for α over the rather broad range of $[0.001, 0.5]$. This shows that S^3 is rather insensitive to choice of significance level a. The level $\alpha = 0.05$ was used in all other examples shown in this paper. MPEG versions of Figures 7a and 7b, showing the full range of scale, are available at the above web address, in the files `sss1fig7a.mpg` and `sss1fig7b.mpg`.

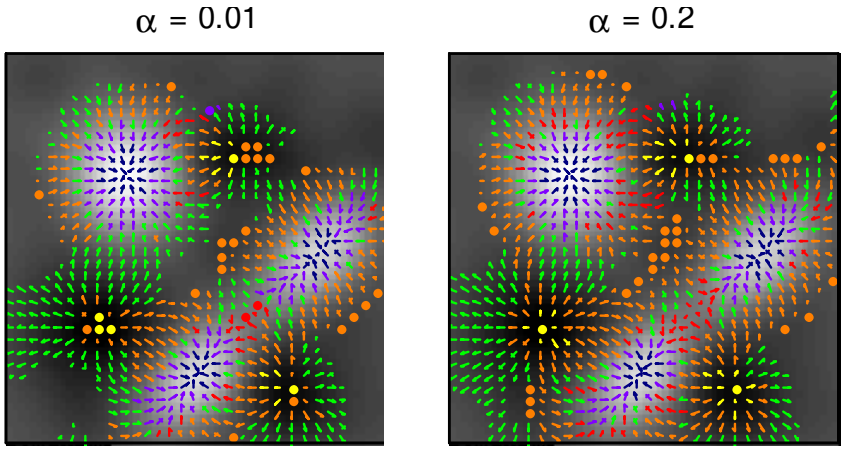

FIGURE 7: *Gradient and Curvature version of S^3, for simulated "Peaks and Valleys" example, using (a) $\alpha = 0.01$ (b) $\alpha = 0.2$.*

The Peaks and Valleys signal s from Figure 7 also shows the good characteristics of S^3 in high noise situations, as seen in Figure 8. Here the image size is again 64×64, but now Gaussian noise with $\sigma = 0.4$ has been added, which makes it quite hard to discern all of the features, see Figure 8a. The peaks can be seen as lighter regions, although it is not obvious that they are "really there" in the presence of such high noise. The three valleys are even harder to find. In particular, they are not easily distinguished from random dark regions, where there is no valley. Some exploration with the slider version of S^3 showed that the scale $h = 6$ found all of the features that are present, as shown in Figure 8b. In particular, all three peaks have some dark blue regions, with more dark blue for the rounder peak. Only the deeper valley has some yellow regions, and the other two only show up as orange (i.e. there was significance only in one curvature direction, not two. For the valley at the upper right, this is not surprising, since its shape is long and thin. The valley below the big peak shows no yellow, since it is not deep enough. The large number of orange dots near the center are artifacts of smoothing the peaks on either side by the relatively large bandwidth of $h = 6$. It is also interesting to look at the full range of scales, as shown in the MPEG version `sss1fig8b.mpg` at the above web address.

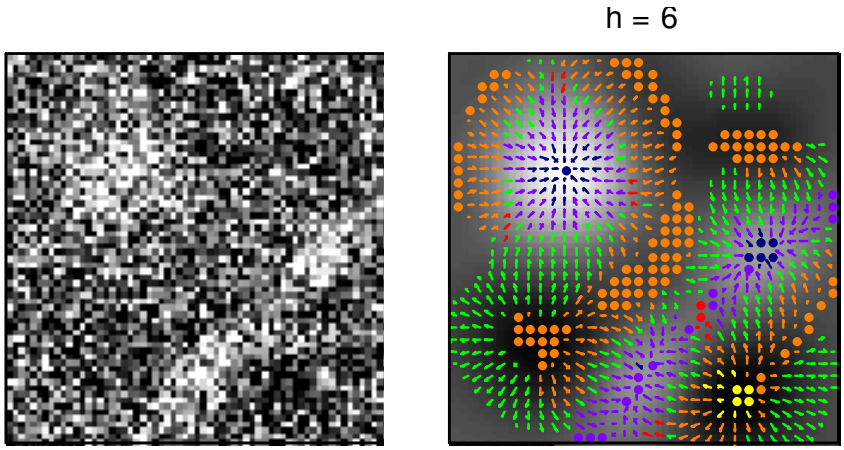

FIGURE 8: *For simulated Peaks and Valleys example, with high noise $\sigma = 0.4$, (a) Raw data, (b) Gradient and Curvature version of S^3.*

A simulation example showing how the streamline approach efficiently conveys information is shown in Figure 9. The underlying signal s here is a "Gaussian doughnut", i.e. the volume of revolution of the Gaussian density with nonzero mean, that has an off-centered cylinder removed. Thus the ridge on the left is smooth, and higher than the sharp ridge on the right. An image plot of this signal is also available in `sss1talk.ps`. Gaussian noise with $\sigma = 0.064$ is added to a 64×64 pixel image. Figure 9a shows the $h = 4$ smooth, together with the gradient and curvature version of S^3. Here and in Figure 9b, the boundary adjustment discussed in Section 2.1 was not used, since it introduced distracting valleys in the corners of the image. In the dark regions near the corners, there are orange arrows pointing upwards because the surface is convex in those regions. In the center of the image, the arrows are colored yellow, since the surface is convex in both directions here. The color purple is most prevalent in between, because the light ring is generally a ridge. The directions of the purple arrows are interesting, since at the top and bottom of the ridge, they show that it goes uphill towards the left, but not on the right and the left part of the ring. A movie version of Figure 9a, showing the full range of scales is available at the web address above, in the file `sss1fig9a.mpg`.

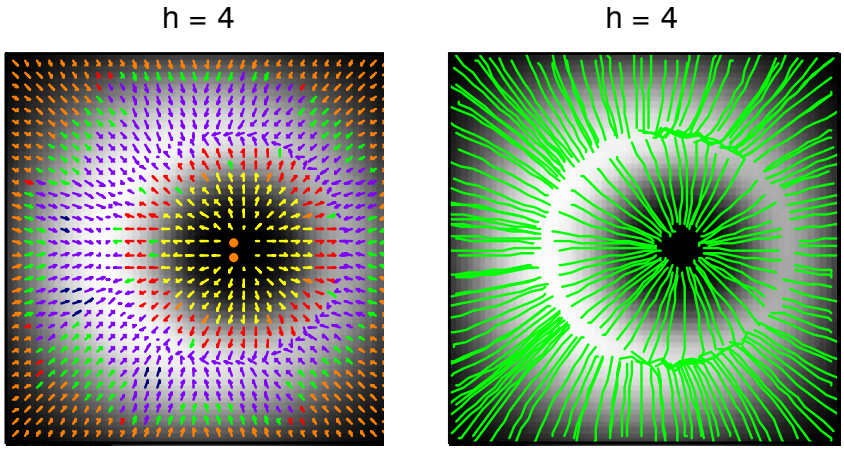

FIGURE 9: *For simulated Volcano example, (a) Gradient and Curvature version of S^3, (b) Streamline version of S^3.*

While these lessons are visible in Figure 9a, they are more accessible in Figure 9b. The circular part of the structure is immediately visible from the way that the streamlines run mostly in a radial direction. The slope in the ridge is apparent from the way that the streamlines join at the top and the bottom, but not at the left and right. Note also that the "raster effect" problem with matrices of arrows, as discussed in Section 3.2, is more visible in Figure 9a than it was in Figure 6a, while it disappears for the streamline version.

5 Real Data Examples

In this section, we study S^3 in action with some real data images.

First we revisit the brain activity data, from Figure 5. Figure 10a shows a streamline analysis of that data, using the scale $h = 5$. This shows that the light peak in the bottom right corner, and the dark valleys (just above and in the upper left corner) are significant features. The less prominent peak near the center is less significant, but most of it is present. Figure 10b shows an $h = 5$ streamline analysis of another subimage of the same data set. Here the feature found by S^3 is the light colored ridge, which is known to be a blood vessel.

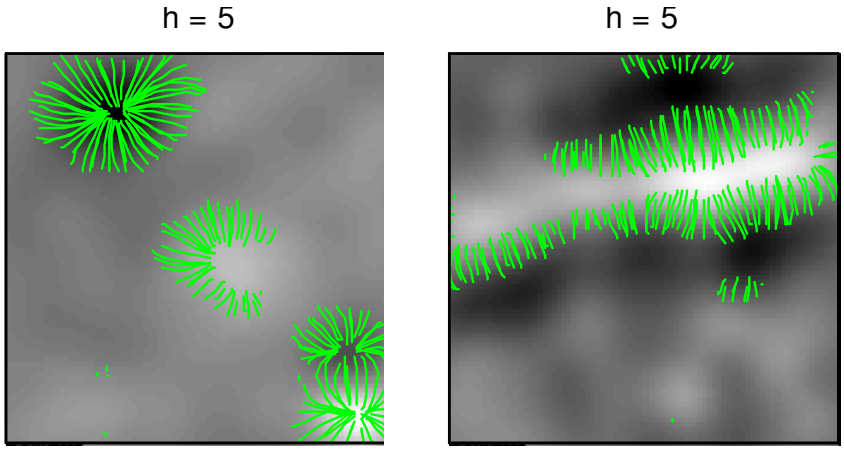

FIGURE 10: *Streamline version of S^3, (a) for "hot spot" part of Brain Activity data, (b) for "blood vessel" part of Brain Activity data.*

A very different real data example is shown in Figure 11. Here the raw image is derived from a time series of perfusion MRI images, as described in Section 4 of the rejoinder of Chu, et. al. (1998), and shown in Figure 11a. The interesting structure is the light regions to the left of center. The gradient and curvature S^3 analysis shown in Figure 11b now uses the quite small scale of $h = 1$, because larger scales eliminate the features of interest. Note that there are actually several different significant peaks, so the structures observed at this small scale are "really there". Versions of Figure 11b, for the full range of scales, can be seen as a movie in `sss1fig11b.mpg`, at the above web address.

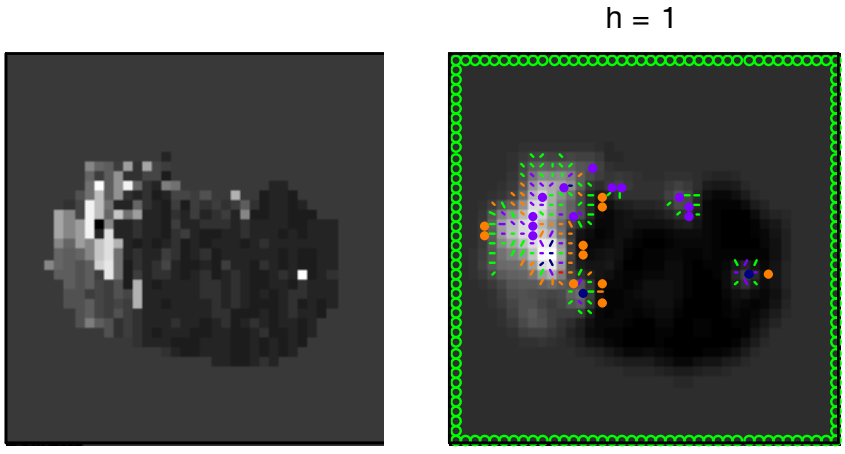

FIGURE 11: *For the MRI perfusion data, (a) Raw data, (b) Gradient and curvature version of S^3.*

The single bright pixel on the lower right side of Figure 11a, is known to be a (non-Gaussian) sampling artifact. This is flagged as significant by S^3 since this assumes Gaussian noise. The fact that a single pixel can generate significance in S^3 is an important difference between this method, and its one dimensional brother, SiZer (as developed in Chaudhuri and Marron (1999)), as was shown by Kim and Marron (1999). The small circles at the edges of Figure 11b indicate that there are less than 5 points in each kernel window there, as discussed in Section 2.2.

Another real data example which pushes S^3 to its limits is shown in Figure 12. The raw data are from confocal microscopy, courtesy of Håvard Rue. While the noise level is much lower than many of the above examples, the question of which image features can be distinguished from background noise still arises. For example, near the bottom on the left side of Figure 12a, there is perhaps a dim elliptical object, which is not easy to distinguish from nearby gray areas, that may be just noise. The streamline S^3 analysis in Figure 12b shows that in fact this is a statistically significant feature. The scale $h = 3$ is quite important here (and was chosen by some trial and error). This feature can also be seen via the gradient and curvature versions of S^3. This is not shown here to save space, but the movie version is available in the file `sss1fig12c.mpg`, at the above web address. This movie shows that streamlines are not uniformly better than the gradient and curvature version of S^3. In particular, streamlines do a poor job of separating the largest lightest elliptical feature near the top from its dim neighbor on the left, since at scales where the dim neighbor is significant, the lines just run over the top. However, when curvature coloring is applied, the neighbor is more clearly visible as a "lower step".

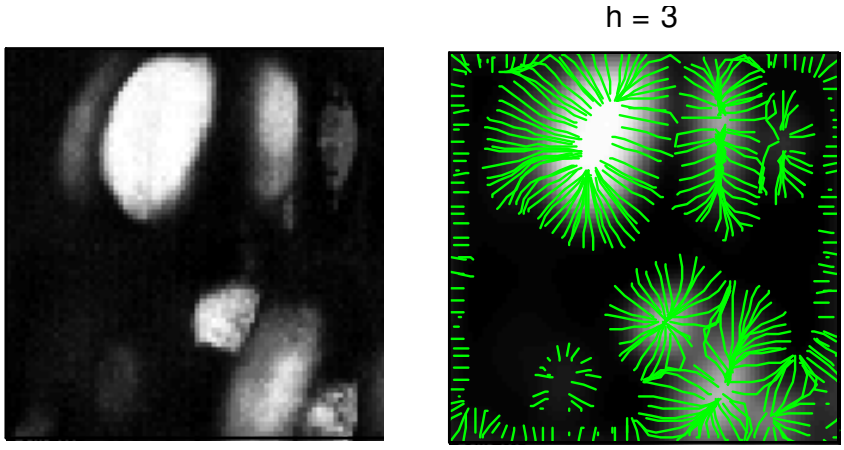

FIGURE 12: *For the confocal data, (a) raw image, can structure on lower left be distinguished from back ground noise? (b) Streamline S^3, shows structure on lower left is statistically significant.*

We have also applied S^3 to satellite images. These are not shown here to save space, and because they do not illustrate the essential ideas as well as those shown here.

6 Appendix: Details of formulas

6.1 Gradient

In this section estimates are derived for the variance parameters of the gradient normal distribution (7). From (1), using the derivative notation from Section 2.4, and the independence of the $Y_{i,j}$,

$$\begin{aligned} & var\left(\widehat{s}_{h,1}(i,j)\right) \\ = \; & cov\left(\sum_{i'=1}^{n}\sum_{j'=1}^{m} Y_{i',j'} K_{h,1}\left(i-i',j-j'\right), \sum_{i''=1}^{n}\sum_{j''=1}^{m} Y_{i'',j''} K_{h,1}\left(i-i'',j-j''\right)\right) \\ = \; & \sum_{i'=1}^{n}\sum_{j'=1}^{m}\sum_{i''=1}^{n}\sum_{j''=1}^{m} K_{h,1}\left(i-i',j-j'\right) K_{h,1}\left(i-i'',j-j''\right) cov\left(Y_{i',j'}, Y_{i'',j''}\right) \\ = \; & \sum_{i'=1}^{n}\sum_{j'=1}^{m} K_{h,1}\left(i-i',j-j'\right)^2 \sigma_{i,j}^2 \end{aligned}$$

where in the homoscedastic case $\sigma_{i,j}^2 = \sigma^2$. This is the basis for the proposed estimates

$$\begin{aligned} \underline{\widehat{\sigma}}_1^2 &= \left(\underline{K}_{h,1} \cdot \underline{K}_{h,1}\right) * \underline{\widehat{\sigma}}^2, \\ \underline{\widehat{\sigma}}_2^2 &= \left(\underline{K}_{h,2} \cdot \underline{K}_{h,2}\right) * \underline{\widehat{\sigma}}^2, \end{aligned} \tag{14}$$

where $\cdot$ denotes element by element matrix multiplication as in (4), and where $\underline{\widehat{\sigma}}^2$ is the matrix whose entries are either the local variance estimates from (4), or is the constant matrix, each of whose entries is either the given known variance, or the pooled estimated variance from (5).

To check that the covariance σ_{12}^2 is negligible, first by a similar calculation to the above

$$cov\left(\widehat{s}_{h,1}(i,j), \widehat{s}_{h,2}(i,j)\right) = \sum_{i'=1}^{n}\sum_{j'=1}^{m} K_{h,1}\left(i-i',j-j'\right) K_{h,2}\left(i-i',j-j'\right) \sigma_{i,j}^2.$$

For notational simplicity, explicit details are given only for the homoscedastic case $\sigma_{i,j}^2 = \sigma^2$. By Riemann summation (with step length 1), approximating

$$\begin{aligned}
\sigma_1^2 &\approx \sigma^2 \iint K_{h,1}(x,y)^2 \, dxdy = \sigma^2 \int K_h'(x)^2 \, dx \int K_h(y)^2 \, dy, \\
\sigma_{12}^2 &\approx \sigma^2 \iint K_{h,1}(x,y) \, K_{h,2}(x,y) \, dxdy \\
&= \sigma^2 \int K_h'(x) \, K_h(x) \, dx \int K_h'(y) \, K_h(y) \, dy = 0,
\end{aligned}$$

where the integral symbols are understood to mean definite integration over the range $(-\infty, \infty)$. But using Corollary 4.5 of Aldershof, et. al. (1995), for the Gaussian kernel,

$$\begin{aligned}
\int K_h(x)^2 \, dx &= \frac{1}{2\pi^{1/2} h}, \\
\int K_h'(x) \, K_h(x) \, dx &= 0, \\
\int K_h'(x)^2 \, dx &= \frac{1}{4\pi^{1/2} h^3}.
\end{aligned} \tag{15}$$

from which it follows that

$$\sigma_1^2 \approx \sigma^2 \frac{1}{8\pi h^4},$$

and that $\sigma_{12}^2 \approx 0$.

6.2 Curvature

In this section approximations are derived for the variance parameters of the curvature normal distribution (11), and the form (12) is justified. These calculations are similar to those of Section6.1, so fewer steps are included here. First,

using independence,

$$\begin{aligned}
var\left(\widehat{s}_{h,11}(i,j)\right) &= \sum_{i'=1}^{n}\sum_{j'=1}^{m} K_{h,11}\left(i-i',j-j'\right)^2\sigma_{i,j}^2,\\
var\left(\widehat{s}_{h,12}(i,j)\right) &= \sum_{i'=1}^{n}\sum_{j'=1}^{m} K_{h,12}\left(i-i',j-j'\right)^2\sigma_{i,j}^2,\\
var\left(\widehat{s}_{h,22}(i,j)\right) &= \sum_{i'=1}^{n}\sum_{j'=1}^{m} K_{h,22}\left(i-i',j-j'\right)^2\sigma_{i,j}^2,\\
cov\left(\widehat{s}_{h,11}(i,j),\widehat{s}_{h,12}(i,j)\right) &= \sum_{i'=1}^{n}\sum_{j'=1}^{m} K_{h,11}\left(i-i',j-j'\right)K_{h,12}\left(i-i',j-j'\right)\sigma_{i,j}^2,\\
cov\left(\widehat{s}_{h,11}(i,j),\widehat{s}_{h,22}(i,j)\right) &= \sum_{i'=1}^{n}\sum_{j'=1}^{m} K_{h,11}\left(i-i',j-j'\right)K_{h,22}\left(i-i',j-j'\right)\sigma_{i,j}^2,\\
cov\left(\widehat{s}_{h,12}(i,j),\widehat{s}_{h,22}(i,j)\right) &= \sum_{i'=1}^{n}\sum_{j'=1}^{m} K_{h,12}\left(i-i',j-j'\right)K_{h,22}\left(i-i',j-j'\right)\sigma_{i,j}^2.
\end{aligned}$$

This is the basis for the proposed estimates

$$\begin{aligned}
\widehat{\underline{\sigma}}_{11}^2 &= \left(\underline{K}_{h,11}\cdot\underline{K}_{h,11}\right)*\widehat{\underline{\sigma}}^2,\\
\widehat{\underline{\sigma}}_{22}^2 &= \left(\underline{K}_{h,12}\cdot\underline{K}_{h,12}\right)*\widehat{\underline{\sigma}}^2,\\
\widehat{\underline{\sigma}}_{33}^2 &= \left(\underline{K}_{h,22}\cdot\underline{K}_{h,22}\right)*\widehat{\underline{\sigma}}^2,\\
\widehat{\underline{\sigma}}_{13}^2 &= \left(\underline{K}_{h,11}\cdot\underline{K}_{h,22}\right)*\widehat{\underline{\sigma}}^2,
\end{aligned}$$

where $\cdot$ denotes element by element matrix multiplication.

Similar methods to the above are used to check that the covariances σ_{12}^2 and σ_{23}^2 are negligible, and to understand the variance structure in (12) . In

particular,

$$\begin{aligned}
\sigma_{11}^2 &\approx \sigma^2 \iint K_{h,11}(x,y)^2\,dxdy = \sigma^2 \int K_h''(x)^2\,dx \int K_h(y)^2\,dy, \\
\sigma_{22}^2 &\approx \sigma^2 \iint K_{h,12}(x,y)^2\,dxdy = \sigma^2 \int K_h'(x)^2\,dx \int K_h'(y)^2\,dy, \\
\sigma_{33}^2 &\approx \sigma^2 \iint K_{h,22}(x,y)^2\,dxdy = \sigma^2 \int K_h(x)^2\,dx \int K_h''(y)^2\,dy, \\
\sigma_{12}^2 &\approx \sigma^2 \iint K_{h,11}(x,y)\,K_{h,12}(x,y)\,dxdy \\
&= \sigma^2 \int K_h''(x)\,K_h'(x)\,dx \int K_h(y)\,K_h'(y)\,dy, \\
\sigma_{13}^2 &\approx \sigma^2 \iint K_{h,11}(x,y)\,K_{h,22}(x,y)\,dxdy \\
&= \sigma^2 \int K_h''(x)\,K_h(x)\,dx \int K_h(y)\,K_h''(y)\,dy, \\
\sigma_{23}^2 &\approx \sigma^2 \iint K_{h,12}(x,y)\,K_{h,22}(x,y)\,dxdy \\
&= \sigma^2 \int K_h(x)\,K_h'(x)\,dx \int K_h''(y)\,K_h'(y)\,dy.
\end{aligned}$$

Some of these integrals are evaluated above, and the others are

$$\begin{aligned}
\int K_h''(x)^2\,dx &= \frac{3}{8\pi^{1/2}h^5}, \\
\int K_h''(x)\,K_h(x)\,dx &= \frac{-1}{4\pi^{1/2}h^3},
\end{aligned}$$

again using Corollary 4.5 of Aldershof, et. al. (1995). This gives

$$\begin{aligned}
\sigma_{11}^2 &\approx \frac{3\sigma^2}{16\pi h^6}, \\
\sigma_{22}^2 &\approx \frac{\sigma^2}{16\pi h^6}, \\
\sigma_{33}^2 &\approx \frac{3\sigma^2}{16\pi h^6}, \\
\sigma_{12}^2 &\approx 0, \\
\sigma_{13}^2 &\approx \frac{\sigma^2}{16\pi h^6}, \\
\sigma_{23}^2 &\approx 0.
\end{aligned}$$

The ratios in the covariance matrix of (12) show how to combine the estimates $\widehat{\sigma}_{11}^2$, $\widehat{\sigma}_{22}^2$, $\widehat{\sigma}_{33}^2$ and $\widehat{\sigma}_{13}^2$ into the single variance estimate $\widehat{\sigma}_C$ (this approximation makes the simulated distribution in the next section case independent),

$$\widehat{\sigma}_C^2 = \frac{\widehat{\sigma}_{11}^2/3 + \widehat{\sigma}_{22}^2 + \widehat{\sigma}_{33}^2/3 + \widehat{\sigma}_{13}^2}{4}.$$

6.3 Curvature Null Distribution

Simulation values of the distribution of the eigenvalues of

$$\begin{pmatrix} \widehat{s}_{h,11}/\widehat{\sigma}_C & \widehat{s}_{h,12}/\widehat{\sigma}_C \\ \widehat{s}_{h,12}/\widehat{\sigma}_C & \widehat{s}_{h,22}/\widehat{\sigma}_C \end{pmatrix}$$

are based on Z_1, Z_2 and Z_3 independent $N(0,1)$, which are transformed by the square root of the covariance matrix in (12), and plugged into the eigenvalue formula (9) to give

$$\lambda^*_{\pm} = (Z_1 + Z_3) \pm \frac{1}{2}\sqrt{2(Z_1 - Z_3)^2 + 4Z_2^2}.$$

Simulation investigation of the $T^* = \max\left\{\left|\lambda^*_+\right|, \left|\lambda^*_-\right|\right\}$ distribution, showed that it was nearly linear on the scale of q vs. $-log_{10}(1-\alpha)$. Approximate quantiles of the T^* can be found by linear interpolation of the simulated quantiles given in the ASCII file `sigcurv2.asc` at the above web site. These values were based on 10^6 pseudo realizations, and cover a range that is needed for most S^3 applications (depending on the desired value of α).

6.4 Curvature Combination

This section discusses how to extend the color dot visualization developed in Section 2.5 to the case of 2×2 blocks of pixels as mentioned at the end of that section. The main challenge is that a number of different color could appear among the four pixels in a given block. Our resolution of this is based on the significances of the eight eigenvalues in the four pixels, summarized as

$$\begin{aligned} n_+ &= \#\left\{\widehat{\lambda} : \widehat{\lambda} > \widehat{q}_{\widehat{T}}\right\}, \\ n_- &= \#\left\{\widehat{\lambda} : \widehat{\lambda} < -\widehat{q}_{\widehat{T}}\right\}. \end{aligned}$$

Then we color the dot for a given block using essentially the rules:

color	**feature**	**characterization**
yellow	hole	$6 \le n_+ - n_-$
orange	long valley	$3 \le n_+ - n_- \le 5$
red	saddle point	$-2 \le n_+ - n_- \le 2, 3 \le \min\{n_+, n_-\}$
purple	long ridge	$-5 \le n_+ - n_- \le -3$
dark blue	peak	$n_+ - n_- \le -6$

No dot is used when n_+ and n_- do not satisfy any of these. It is possible that a block may have three individual pixels that would have the same color under the scheme (13), but that would be different from the color assigned here. In such cases, we change the block color to be that of the three pixels, i.e. the rule developed in this section is over-ridden by majority vote.

References

[1] Aldershof, B., Park, B. U., Marron, J. S. and Wand, M. P. (1995) Facts about the Gaussian probability density function, *Applicable Analysis*, 59, 289-306.

[2] Chaudhuri, P. and Marron, J. S. (1997) Scale space view of curve estimation, Institute of Statistics, Mimeo Series # 2357.

[3] Chaudhuri, P. and Marron, J. S. (1999) SiZer for exploration of structure in curves, (to appear in *Journal of the American Statistical Association*).

[4] Chu, C. K., Glad, I. K., Godtliebsen, F. and Marron, J. S. (1998) Edge preserving smoothers for image processing, *Journal of the American Statistical Association*, 93, 526-556.

[5] Godtliebsen, F. (1991), Noise Reduction Using Markov Random Fields, *Journal of Magnetic Resonance*, 92, 102-114.

[6] Green, P. J. (1990) Bayesian reconstructions from emission tomography data using a modified EM algorithm, *IEEE Transactions on Medical Imaging*, 9, 84-93.

[7] Hastie, T. J. and Tibshirani, R. J. (1990) *Generalized Additive Models*, Chapman and Hall, London.

[8] Helman, J. and Hesselink, L. (1989) Representation and display of vector field topology in fluid flow data sets, *IEEE Computer*, 22, 27-36.

[9] Jain, A. K. (1989) *Fundamentals of digital image processing*, Prentice Hall, Englewood Cliffs, NJ.

[10] Kim, C. S. and Marron, J. S. (1999) SiZer for jump detection, unpublished manuscript.

[11] Lindeberg, T. (1994) *Scale-Space Theory in Computer Vision*, Kluwer, Dordrecht.

[12] Winkler, G. (1995) *Image analysis, random fields, and dynamic Monte Carlo methods : a mathematical introduction*,Springer-Verlag, Berlin.