Geometric Image Analysis Using Multiscale Statistical Features

James M. Coggins and Elizabeth Graves
Department of Computer Science
University of North Carolina at Chapel Hill

Many medical image analysis problems involve pixel classification tasks such as segmentation and landmark identification. This paper describes a methodology for defining an optimal feature space for performing such tasks based on a principled, generalized image decomposition. The methodology is demonstrated on two kinds of test images, and the nature of the optimization is illustrated.

The Artificial Visual System (AVS) (Figure 1) is a computational framework for computer vision. [1] In AVS convolves an image with a series of D spatial filters, yielding a series of D filtered images. While any set of filters can be applied in this way, families of filters can be designed to produce a compact, task-specific decomposition or a general, task-independent decomposition of the image. The filter set disperses information about image structure through the filtered images. The outputs of the filters at a particular location, \((x,y)\), define a pattern (in the sense of statistical pattern recognition), \(P_X\) (Figure 2). This pattern can be treated as a point in a D-dimensional feature space; the convolution of each filter with the original image computes one feature for every pixel. Thus, we say that the filters map each pixel to a point in a D-dimensional feature space, so we can refer to pixels \(I_X\), or to their corresponding points \(P_X\).

Mapping pixels into a feature space is a powerful idea. The tools of statistical pattern recognition\(^2,3,6\) (classification, clustering, multidimensional scaling, various transformations and warpings of space) can be applied to discover structure or to determine an appropriate labeling of each pixel. Decision methods are nonlinear functions that recombine the information dispersed by the filter set to yield the desired inferences. Recombination algorithms as simple as thresholding, absolute value, or location of relative extrema have proven useful in the past.

This approach to computer vision parallels the way electromagnetic spectroscopy is used to discover properties of distant stars and galaxies, or the way LANDSAT images yield information about the surface of the earth\(^7,11,12\). Light is dispersed to form a spectrum that is sampled to yield measurements. The measurements are analyzed using nonlinear decision procedures to infer properties of the star, galaxy, or terrestrial footprint. The AVS acts as a spatial prism, dispersing image energy (variations of intensity) in multiscale neighborhoods of each pixel into different bands of scale or orientation (depending on the filter design). Decisions can be

![Figure 1: An Artificial Visual System](image1)

![Figure 2: Creating the Pattern P_X](image2)
made about the objects portrayed in the image based on nonlinear processing of the patterns $P_x$. This approach reveals a unified, coherent vision for computer vision:

**computer vision is spatial spectroscopy.**

Our goal is to define a unified, scientific foundation for computer vision based on spatial spectroscopy and to develop methods for engineering AVS-like systems to solve practical (mostly biomedical) problems (see also [2]).

### Structural Descriptions of Images

An image is an uninterpreted recording of intensity. The interpretation of image content begins with a description of image structure. The description of image structure produced by an AVS is governed by the filter set. Filter sets for an AVS can be defined ad hoc based on an understanding of the task to be performed, but our experience suggests that intuition misses subtleties that a more principled approach can capture. A more general, principled mechanism for defining the filter set derives from a general, differential geometric analysis of image structure.

Koenderink\(^7,8,9\) has defined several sets of operators based on the Taylor Series expansion of the local structure of an image,

$$L((x+\alpha, y+\beta), \sigma) = \sum_{n=0}^{\infty} \frac{1}{n!} \left( \frac{\partial}{\partial x} + \frac{\partial}{\partial y} \right)^n L(x, \sigma) (1)$$

where $L(x, \sigma)$ is the intensity of the image at location $(x, y)$ after convolution with a Gaussian with standard deviation $\sigma$. For the description of an image to introduce no spurious structure as the scale of measurement is enlarged, the resolution kernel must be a solution to the diffusion equation,

$$\frac{df}{dt} - \frac{d^2 f}{dx^2} = 0.$$

The Gaussian is the Green's function (impulse response function) of the diffusion equation. These considerations lead to the use of derivatives of Gaussians as the set of filter kernels that produce general descriptions of image structure. The variance of the Gaussian selects the scale of measurement, and the derivatives select structural features of the intensity function. In practice, the Taylor series must be truncated. Truncating the series after including all derivatives of some integral order results in a representation that retains the desired invariances. This truncated series is called the N-jet of the image function. There are $(N+1)(N+2)/2$ kernels in the N-jet.

Expanding the binomial in (1) yields the Cartesian family of solutions. Koenderink also defines a polar family and a "wavelet" family. The $i$th-order derivatives in any one of these families can be linearly combined to form the $i$th-order derivatives of the other families.

To use these families of operators to describe image structure requires definition of filter kernels for each operator at each scale of interest. To measure through 3rd-order derivatives at a series of 6 scales requires 60 filter kernels. While Koenderink's receptive field families yield a complete geometric description of the intensity structure, it is expensive to perform so many convolutions, and with such a large number of correlated features, the metric to use for measuring distances in the feature space is unclear.

An approximation to the Gaussian derivative kernels yield tremendous computational savings. Convolving multiscale Gaussian filters $G(\sigma)$ with an image $I$ and recording in $P_x$ the filter outputs at $x$ and at certain nearby locations, $x+\delta_j$ we obtain a pattern that is related to Koenderink's polar operator family. The pattern is given formally by

$$P_x = \{I*G(\sigma)\}(x+\delta_j), \ 0 \leq i < 6, \ 0 \leq j < H$$

where $I$ is the image, $G(\sigma)$ is a Gaussian and $i$ indexes the scales used in a decomposition, $x$ is the pixel location, and $\delta_j$ is the $j$th displacement vector for the $i$th scale. The displacement vectors we are using are oriented in the eight compass directions, and their lengths are related to the standard deviation of the Gaussian. Since the feature vector incorporates responses to Gaussians at multiple scales and at multiple nearby locations, the feature vector at a particular location captures image structure across space and scale.

Approximations to Koenderink's generic neighborhood operators are linear combinations of these Gaussians and offset Gaussians. (For example, the Laplacian is approximated by a difference of centered Gaussians at different scales.) Thus, approximations to many of the derivative of Gaussian operators are present as lines in the Gaussian/offset Gaussian feature space.

Computationally, we can perform $k$ convolutions with Gaussians and extract up to 9 values (at $x$ plus at offsets in the eight compass directions at distance $k\sigma$) from each filtered image to obtain $9k$
measurements that capture the (k-2)-jet across k scales. In fact, we get both more and less than the (k-2)-jet. Some higher-order structure is accessible in the feature space since k scales are measured, but some of the low-order “propeller” patterns are not well-captured by these particular offset Gaussians. Thus, the feature set is not quite orientation-invariant.

Exploring the Feature Space

An experiment begins by labeling regions in the image to define each class. Many such labelings are possible on a given image. The labelings of most interest for this exploration involve class distinctions that are inherently geometric in nature. Point operations are insufficient to distinguish the classes. Part of the objective of this study is to begin characterizing classes that require higher-order geometric measures to distinguish the classes and thereby to begin answering the feature selection question: Of the many features available in the N-jet, which ones are most relevant for the defined task? See [1] for samples of simpler geometric discriminations.

All pixels are mapped to the feature space defined by 3 Gaussian blurs and 5 spatial offsets. The pixels for each class form clouds in the feature space. The cloud of points for class c can be characterized by its mean vector (centroid) \( \mu_c \) and covariance matrix \( \Sigma_c \). For an arbitrary pixel \( x \), the Mahalanobis distance of that pixel’s pattern \( P_x \) to the Gaussian density representing class c is

\[
d(x,c) = (P_x-\mu_c)^T \Sigma^{-1}_c (P_x-\mu_c).
\]

The covariance matrix, and thus the Mahalanobis stance, account for the different variances of the attributes and for their correlations. The value of \( d(x,c) \) is a probability-normalized, squared distance on the (hyperellipsoidal) multivariate Gaussian cloud, similar to the squared z-score, \( z^2=(x-\mu)^2/\sigma^2 \). Classification can be based on a threshold (all pixels for which \( d(x,c) \) is less than a threshold are labeled as belonging to class c) or on a maximum likelihood criterion (pixel x is assigned to the class with minimum Mahalanobis distance from x).

The class as labeled by the user corresponds to multiple clusters or non-hyperellipsoidal clusters of feature space (for example, the class contains both white-on-black and black-on-white objects) when the multivariate Gaussian is not an effective model for the distribution of points in feature space. A different probability density must be used, or the class must be split into components better matched to a multivariate Gaussian. For example, Gaussian mixture densities may provide improved fit at low computational cost.

Given the Gaussian fit for each class, we can compute the Hotelling matrix, \( H=S_B S_W^{-1} \), where \( S_B \) is the scatter matrix of the centroids (the Scatter Between) and \( S_W \) is the pooled scatter matrix of the classes (Scatter Within) [4]. (A scatter matrix S is \( n \Sigma \) for the set of n points.) The Hotelling Trace, \( J=\text{trace}(H) \), is a measure of the separation between multivariate Gaussian densities that is related to the probability of error that would be obtained using an optimal linear classifier [5]. It can be thought of as a multivariate signal-to-noise ratio. The value of J increases with larger separation between the classes and decreases with larger variances in the classes. J is highest, then, for compact, isolated clusters.

Computing J for many classes together gives an averaged measure of “clustering tendency” in the whole data set, but as averages often do, presents a less than complete picture of the class structure.

The J statistic does not indicate how to create a reduced feature space capturing the class separation. Such an optimized feature space can be obtained, however, by computing the eigenvectors of the H matrix. Since the trace is the sum of the eigenvalues, the eigenvectors corresponding to the largest eigenvalues of H are the vectors in feature space that contribute most to the separation of classes as measured by J. If there are C classes, there will be C-1 eigenvectors of H with nonzero eigenvalues. Thus, in a 5-class classification task, we can obtain an optimal 4-dimensional space that captures all of the separation of the classes that exists in the high-dimensional measured space. A reduced feature space with linearly independent axes is defined by selecting the L (0<L<C) eigenvectors corresponding to the largest eigenvalues of H.

The next step is to construct a visualization of how the reduced space is formed. The measured coordinates correspond to convolutions with multiscale Gaussians and shifted Gaussians. The eigenvectors are linear combinations of those axes. Performing this same linear combination (whose coefficients are given by the coordinates of the eigenvector) on the Gaussian and shifted Gaussian
kernels defines a single filter kernel corresponding
to each eigenvector. Thus, we can visualize the
eigenvectors as images (eigenkernels) and
determine by examining them what spatial
structures distinguish the classes, what the effect
is of adding additional classes to the task, and how
the eigenvectors relate to differential invariants.

One such study is shown in Figure 3. The original
image (Figure 3a) consists of bicolored disks on a
gray background with two orientations: white on
top (WT) and white on bottom (WB). The task is to
distinguish the pixels in WT from those in WB.
Figure 3b shows the (sole) eigenkernel for this
task. The eigenkernel is essentially a first-
derivative operator, as one might expect. The
necessity of a first-derivative operator for this task
was not input or externally influenced; the first
derivative arose out of the geometry of feature
space and out of the nature of the task defined by
the selection of training pixels.

The study in Figure 4 is based on the same image
(Figure 3a), but it shows the two eigenkernels for
the task of distinguishing the WT and WB classes
plus a background class. The first eigenkernel
(Figure 4a) is basically a zeroth-order Gaussian
that distinguishes background pixels from pixels in
the disks. Because of the strong separation in
feature space between background and disk pixels,
this eigenkernel provides 70.2% of the Hotelling
Trace in this task, even though it does little to
distinguish classes WT and WB. The rest of the
class separation is captured by the fascinating
ekernel in Figure 4b (this image was log scaled to
make fainter, peripheral structure easier to see).
This eigenkernel is almost a polar second cross-
derivative operator: one derivative in $\rho$ and one
derivative in $\theta$.

Conclusion

A methodology has been demonstrated for defining
a pixel classification task, mapping it into the form
of a statistical pattern recognition study, defining
a high-dimensional generalized feature space for
geometric classification tasks, and optimizing that
feature space. We have also shown how
eigenkernels are constructed and illustrated how
eigenkernels change with the classification task
and how they can be used to characterize the
salient geometric structure for solving a particular
classification task. Research is in progress to
determine the role of higher-order differential
operators in pixel classification and the situations
when nonlinear combinations of geometric features
might be required.

References

1. J. M. Coggins, "Statistical Investigations of
Multiscale Image Structure", Visualization in
Biomedical Computing 1992, Richard A. Robb,

2. J. M. Coggins, "A Statistical Approach to
Multiscale, Medial Vision", Proceedings of the
Conference on Mathematical Methods in Medical

3. P. A. Devijver and J. Kittler, Pattern
Recognition: A Statistical Approach, Prentice-Hall,

4. R. Duda and P. Hart, Pattern Classification and
Scene Analysis, Wiley-Interscience Publication,

Myers, "The Hotelling Trace Criterion and Its
Correlation with Human Observer Performance",

Clustering Data, Englewood Cliffs, NJ: Prentice

7. J. R. Jensen, Introductory Digital Image
Processing: A Remote Sensing Perspective,

8. J. J. Koenderink and A. J. van Doorn, "Generic
Neighborhood Operators", IEEE PAMI 14(6), June

9. J. Koenderink, "Operational Significance of
Receptive Field Assemblies," Biological

10. J. Koenderink & A. J. van Doorn,
"Representation of Local Geometry in the Visual

11. A.E. Roy and D. Clarke, Astronomy Principles

12. P. H. Swain and S. M. Davis, eds, Remote
Sensing: The Quantitative Approach, McGraw-Hill,
1978.
Figure 3: Original image and eigenkernel for 2-class study
(a) original image (b) eigenkernel resembling a first-derivative operator

Figure 4: Eigenkernels for 3-class study on image of Figure 3(a)
(a) Eigenkernel 1 (b) Eigenkernel 2