Chapter 1
Introduction

Classical signal and image processing uses linear processing techniques. These are methods based in the familiar Fourier, Z, and Laplace transforms. These methods assume that signal and image data may be processed by mapping them onto lower-dimensional orthogonal spaces resulting in solutions designed by decomposing the input into sinusoidal components and processing them individually. While mathematically elegant, this imposition of linearity results in a very limited set of processing operations compared to the total set of solutions possible, i.e., both linear and nonlinear. For example, techniques based on rank ordering of values, logical and geometric processing approaches can give excellent results, particularly for image processing applications. This approach should not be viewed as an alternative to the classical methods, but as a superset of techniques containing many new novel techniques as well as the linear techniques listed above.

The model chosen to convey these concepts is that of digital logic. This is because it can quite literally capture any processing operation, linear or nonlinear, that may be required. Many engineers and computer scientists are comfortable with its notation and concepts. Minimization techniques and software tools are available to reduce complex solutions into their simplest form, and the solutions translate readily into electronic hardware or software implementations.

Every digital signal or image processing operation can be viewed at its most basic level as the manipulation of a series of finite-length binary strings. Whether the operation is implemented on a processor through software or in dedicated hardware, the data and the algorithms are invariably mapped through electronic logic components, which are inherently binary in nature.

Therefore, every digital signal and image processing task can be cast in terms of a logical representation. It does not matter if the data is binary, grayscale, color, or multiband, nor whether the operation is linear or nonlinear. If it can be programmed, then it can be placed in the context of a logical representation.

In nonlinear image and signal processing, the design of operators is carried out by seeking the optimum mapping from one set of binary strings to another. This contrasts with the linear approach which formulates a solution by optimizing coefficients within a generalized multiply-accumulate context. It should be noted, however, that even this linear method is then mapped into digital logic for computation.

In these terms, linear models may be perceived as restricted subsets within a logical framework. Hence, a nonlinear solution to the same problem will be a more general result that will be either better or the same as the linear solution, provided that other conditions are met. One of the most important conditions is that sufficient training data is available.

So why do linear solutions remain so common? There are a number of reasons. The first is familiarity. Engineers and signal processors are trained in linear techniques and are reluctant to depart from the security of these familiar solutions unless the subsequent improvements are great.

Also, the superposition properties of linear models makes parameter estimation straightforward. This means that a small number of examples of system behavior may be used to infer performance across a range of conditions. In theory, a linear system may be completely described by observing the same number of training examples as the rank of the system. In practice, even allowing for the system observations to be noisy, the model may be fully characterized with only a small amount of over-determination. Also, if the linear system model is extended by adding extra parameters, only a linear increase in the number of training examples is required.

The situation is much more complex for nonlinear systems. The task is to seek the optimal logical mapping from all possible mappings. No simple superposition properties exist, and in the most general unconstrained design case, every combination of input variables must be observed a sufficient number of times in order to estimate the conditional probabilities of the output. Extending the system model by adding more parameters leads to a rapid increase in the size of the required training set. This contrasts sharply with the linear problem where it is only required that one estimate the autocorrelation matrix, which is a much smaller set of values than the conditional probabilities.

For logical mappings containing a large number of variables, the required training set may be impossibly large. It may well be that even after observing a huge set of training examples, some combinations have not been observed or have been observed an insufficient number of times to make a statistically accurate estimate of their conditional probabilities.

In the face of these estimation difficulties, it is not surprising that linear methods remain popular. Also, in many problems such as circuit analysis and audio applications, linear solutions are quite satisfactory. These systems are inherently linear with their steady state and transient behavior being completely modeled as a product of sinusoids and decaying exponentials. Other systems make much use of Gaussian noise models and these sit naturally in a linear context. In these cases there is no need to look any further, this model is satisfactory.

However, these linear approaches that work so well for many problems are not necessarily as useful for image processing applications. The 2D nature of image processing problems combined with human visual perception often requires more involved decisions than is the case in 1D signal processing. For example, the tasks might include object and texture classification or size distribution estimation. In many cases, the 2D image is a single projection of the 3D world via unspecified models with unknown parameters. The additional problems of perspective, shadow, and occlusion lead to further ambiguities that can only be resolved with the application of experiential knowledge.

Visual perception is a complex task; it is not tolerant of the linear approximations that arise from frequency decomposition and the projection of signals into orthogonal subspaces. As a result of the perceptual importance of edges, the essential components of images tend to occupy a wide range of the frequency domain. The corrupting noise processes may well overlap the signal in such a way as to make linear separation impossible. It is also difficult to quantify image quality through simple measures such as mean-absolute error (MAE) and mean-square error (MSE).

For example, an image may be restored in such a way that it contains only a tiny variation in MAE from some ideal original, but if the higher frequency components are lost or there is significant phase distortion, it may look very poor to a human observer. On the other hand, large variations in brightness and contrast (leading to large error measures) may be tolerable provided that the edges are distinct.

Despite these points, linear image processing techniques have thrived because of their mathematical elegance and their ability to describe continuous signals. Also, the process of sampling such that continuous signals are represented only by their values at discrete points may be completely described by linear mathematics.

Despite this, there are strong arguments for seeking solutions to image processing problems in terms of logical mappings. Consider a linear "image-to-image" processing task, which might include restoration, noise reduction, enhancement, or shape recognition.

We begin with a signal that is sampled in three dimensions (two spatial and one intensity). Let us assume that the image is $256 \times 256 \times 8$ bits. Whatever processing is to be carried out, the result will eventually be mapped back into the same discrete signal space. The bits within the finite strings of the input image are interpreted as part of an unsigned binary number in order to be given an arithmetic meaning. In most linear operations, such as filtering, the unsigned integers will be converted to real or complex numbers containing a mantissa and an exponent. In order to compute the various linear multiply-accumulate transformations, these numbers are then mapped into electronic circuits and viewed as finite-length binary strings. The circuits operate at their most basic level by employing digital electronics to carry out Boolean algebra on the binary strings to produce different binary strings.

The resulting binary values are then mapped back to real or complex numbers that are eventually clipped and quantized into the $256 \times 256 \times 8$ bit signal space that forms the output image.

So even though we may have carried out a fundamentally linear operation such as a Fourier or wavelet transform, it has been implemented as a series of logical operations. We have mapped the signal in terms of binary strings through digital logic to a resulting set of binary strings. However, we have in effect imposed *linearity constraints* such that at every stage of processing the following two statements are true:

1. The binary strings being manipulated have a direct interpretation in terms of real or complex numbers.
2. The logical operations applied to the strings are restricted to those that carry out equivalent linear operations, such as multiplication and addition of real or complex numbers.

Nonlinear image processing is presented here as a generalization of the above operation by removing the linearity constraints. It seeks the optimum mapping implemented directly in logic. The linear solution should be viewed as a special case of the set of all logic-based solutions rather than as an alternative. Given this generalization, the optimum nonlinear solution will be either better or equivalent to the linear solution, but it should not be worse. This inequality holds regardless of the problem or the criteria, provided that the training data is sufficient.

The above argument has lead to various researchers in this field issuing the provocative claim that "all image processing is nonlinear."[1]

The principal reason for adopting this strategy is to see if the other solutions available through a logical approach are useful and offer advantages over linear solutions. Linear solutions can be easy to compute. It is not difficult to derive the optimum linear smoothing filter for an image with noise, but the result of applying this filter is an image which is invariably blurred, causing a loss of signal information. Here, a nonlinear solution such as the median filter gives much better results leading to noise removal and edge preservation without blurring, despite the fact that the median filter takes no account of the image or noise statistics.

In removing the linear constraint, the process of finding the optimum solution becomes much more difficult to compute. However, if the consequences of linear processing are unacceptable results, we must try to do this.

The work in this area has focused on the design of filters. Many applications are possible within this context such as noise reduction, shape, character and object recognition, enhancement, restoration, texture classification, spatial and intensity sampling, and rate conversion.

In practice, all filters are limited in some way. These limits are known as constraints. For example, the filter designed for a particular application may be constrained to lie in a particular class. The optimum filter is therefore the best filter within that class. In this work, we seek the optimum filter from the class of filters that have a logical implementation. This also includes morphological and rank-order filters (which may be cast in the above context and therefore may provide solutions that have an interpretation in terms of shape or numerical ordering).

Linear filters require little training data. In theory, only the same number of examples as the number of parameters is necessary to determine a solution. However, for nonlinear filters, the training process amounts to the estimation of the conditional output probabilities. In the most general case, each training example only provides information about one specific combination of input variables. It is not possible to infer anything about the behavior of the filter for other sets of inputs. For a stochastic system, a sufficient number of observations of every input combination would be required to arrive at a robust estimate. The number of input combinations grows rapidly as the number of input variables increases.

In order to be able to design the filter from a realistically sized training set, further constraints must be applied to the filter. The filter is an estimator; it uses the input values to estimate an unobserved quantity. By making simple assumptions about the image statistics, we can estimate the output value at a specific point by considering only a finite window of observations centered at that point. For binary values, the output becomes a logical function of the input variables. If the window contains *n* points, there are 2^n combinations of input variables for which the relevant output must be estimated. Therefore, there are 2^{2^n} possible functions (or filters) and it is the objective of the design process to determine which one of these functions corresponds to the optimum.

Among the 2^{2^n} functions that may be applied within an *n* point window, there will be many subclasses of functions. We may decide to restrict the choice to a filter that is idempotent or increasing. *Idempotence* implies that the filter has only a one-off effect on the image such that repeated application of the filter leads to no further modification of the image. *Increasing* implies that the filter preserves signal ordering. It can be shown that *increasing* filters map to logical functions that contain no complementation of the input variables. This drastically reduces the size of the training set required and therefore makes filter design easier. This can be explained in terms of logic (since a much smaller set of functions is under consideration) or in terms of statistical estimation (since now a single training example may be used to infer information about other combinations of input variables).

If we assume that the statistics of the image are wide-sense stationary, then we may assume that the same optimum function applies at every point in the image. The filter then becomes translation-invariant. This not only simplifies the processing, but in effect increases the available training data because we do not distinguish between data collected at different locations in the image.

Nonlinear filters can be effective in retaining structural information while removing background clutter in a way not possible with linear operations. They can often be application-specific.

Historically, nonlinear filters have developed along three independent strands: morphological, rank-order, and stack. However, all can be brought together and expressed in the context of logic.

Mathematical morphology has its roots in shape.[2,3] A signal is probed by a structuring element to determine if it "fits" inside the signal. Mathematically, it has been expressed in set theory as explained by Minkowski. Initially, the work grew from binary images, although it can equally well be applied to 1D signals and has since been extended to grayscale[4] and complete lattices.[5]

Morphology was developed in the context of set theory. It does, however, take little more than a change in notation to show that the basic operation of erosion corresponds directly to a logical AND of the input variables. For all practical purposes, what is called an *erosion* in morphology is called a Minkowski subtraction in set theory. It is also called an *intersection* in mathematics and in digital electronics it is called an AND function. Matheron made the observation that every increasing, translation-invariant set operator may be represented as a union of erosions. To an electronics engineer this means that all operators can be implemented as a sum of products (and they do not require complementation). The building blocks of mathematical morphology such as erosion, dilation, opening, closing, and their repetitions under unions and intersections all have straightforward implementations in digital logic.

The second historical line came from the field of rank-order-based filters. These are inherently grayscale in nature and have at their core the ordering of the variables within an input window into their rank order. Trivial examples are the maximum and minimum but the success story of these filters is the median. It possesses powerful noise-removal properties and requires no knowledge of signal and noise distributions. It can be shown to be the optimum estimator of samples in unbiased noise for an MAE criteria.

The final strand of nonlinear filtering is stack filters which are based on Boolean logic operations applied within a finite window. They process grayscale signals by thresholding them at a number of levels and filtering the resultant stack of binary signals with a logic function.

The three types of filters have the following relationship:

$$\text{Order-statistic filters} \subset \text{stack filters} \subset \text{morphological filters}$$

In other words, morphological filters are the most general of the three, stack filters are a subset within morphological filters, and order-statistic filters are a subset within stack filters.

The literature describing the above methods tends to be quite academic and mathematical. It is the purpose of this book to bring these methods together and explain them in terms of logical operations. The objective is to bring these techniques to a whole new community, namely electronic engineers and computer scientists. The text assumes a basic knowledge of logic minimization such as could be achieved through simple K maps. It also uses very basic statistics to identify the optimum filters in the examples given.

The remainder of the book is structured as follows:

Chapter 2 introduces the concept of logic-based image processing through a document restoration example. Chapter 3 considers methods of evaluating the errors in filtering and gives more examples of document processing including resolution changing, edge noise, and optical character recognition. Chapter 4 looks at filter training and the trade-off between the different types of errors. Chapter 5 develops the relationship between logic-based image processing and mathematical morphology and introduces increasing filters. Chapter 6 establishes the link between logic-based image processing and certain classes of order-statistic filters involving variations on the median. Chapter 7 extends these concepts to grayscale through the model of computational morphology. Chapter 8 describes how each of the classes of filters may be implemented in electronic hardware. Chapter 9 presents a case study on image processing of astronomical images. Lastly, Chapter 10 presents conclusions.

With this new perspective on image processing, let us consider a number of applications starting with document restoration.

References

1 E. R. Dougherty and J. Astola, *An Introduction to Nonlinear Image Processing*, SPIE Press, Bellingham, WA (1994).
2 G. Matheron, *Random Sets and Integral Geometry*, Wiley, New York (1975).
3 J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, New York (1982).
4 J. Serra, *Image Analysis and Mathematical Morphology*, vol. 2, Academic Press, New York (1988).
5 H. J. Heijmans, *Morphological Operators*, Academic Press, New York (1994).