Presented at ICDAR, Saint-Malo, France, Oct, 1991

Multiresolution Morphological Approach to Document Image Analysis

Dan S. Bloomberg

Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304, USA

Tel: 415 494-4128
Fax: 415 494-4241
e-mail: bloomberg@parc.xerox.com

Abstract

An image-based approach to document image analysis is presented. The methods are motivated by a merged view of shape and textural image properties at multiple scales. The principal binary image operations are morphological and multiresolution. The generalized opening is introduced for extraction of both shape and texture from an image. Threshold reduction operations are introduced for performing efficient and controllable shape and texture transformations between resolution levels. Some problems, such as halftone or dark area segmentation, can be in large part solved by a sequence of threshold reductions. Aspects of the approach are illustrated by the problem of identifying italic and bold words in text, using word-level extraction at lowered resolution. The computational costs of the basic operations are given, so that algorithm efficiency can be estimated, and the importance of operating at the lowest feasable resolution is demonstrated. For example, word segmentation and halftone extraction proceed in excess of 1.5x10^7 image pixels/second on a Sun Sparcstation2^{TM}.

Keywords: image processing, image analysis, binary morphology, multiresolution, threshold reduction, segmentation, document, shape, texture

Conference topics: Document image processing, Document understanding, Applications, Text recognition

1 Introduction

When constructing systems that analyze document images, it is difficult to ignore the fact that many questions about the image can be answered by a person with a mere glance. For example, we can determine subcognitively, in about 0.1 second, whether a document page has halftones, graphics, multiple columns, rules, freehand annotations, and other characteristics. Our ability to identify various regions of an image with no apparent effort is the inspiration for this work. For example, italic font styles, designed for emphasis, virtually leap off the page. Surely a digital computer can be programmed to identify, quickly and reliably, the salient characteristics of *italic-ness* within an image!

1.1 Background

Human performance on medium and large scale image structure is at great variance with present machine performance. Traditional methods in document image interpretation proceed from the bottom up, starting with the computation of a connected-component representation, and proceeding to derive information from these data structures. This approach is appropriate for character identification on text document images with a simple logical structure, such as a single column of text. However, for complex documents that may contain stipples and halftone images, rules and line graphics, multiple columns and hand annotations, such an approach suffers from several problems: (1) nearly unbounded time and memory requirements in the connected component computation; (2) unreliable character identification on image regions that are not machine-printed text; and (3) architectural complexity and consequent unreliability in the determination of large-scale structures. A striking characteristic of these methods is that they involve virtually no image processing, with all analysis proceeding from relations on and between data structures derived from the original image.

In an effort to alleviate some of these problems, Wong et al.[11, 9] preceeded their image analysis with an image processing step that transformed the image into a relatively small number of larger connected regions, such as textlines. This was accomplished using the intersection of horizontal and vertical closings of very large extent. After the image processing, they used heuristics, based on both the size and shape of these regions and the statistics of corresponding parts of the original image, to segment the image into text and non-text parts. A similar analysis for newspaper image segmentation, emphasizing texture measures computed on the image, has been reported recently[10].

Without special hardware, use of morphological image processing at high resolution has a large computational cost, particularly in time, and especially when large-scale features are to be identified. The amount of computation varies approximately inversely as the third power of the reduction factor. Two powers are due to the relative number of pixels, and the third power comes from the size of the structuring elements or the number of iterations required to cover a given feature. Taking advantage of this tremendous gain in computational efficiency when processing the image at lower resolutions, Bones[3] has recently described an approach for segmenting document images based on morphological image processing and image analysis at various levels of reduction.

1.2 Shape and texture

The *image-based* approach to image analysis described in this paper makes use of transformations based on the fundamental image properties of *shape* and *texture*. Shape and texture have traditionally been viewed as unrelated entities, where shape is a property of the arrangement of ON pixels in connected components and texture is a set of statistical properties of relations between ON and OFF pixels within a region whose size is much larger than the measures used for gathering the statistics. However, image-based image analysis forces these disparate measures to merge in interesting ways.

The first step is to broaden the notion of shape to include specific relations *between ON and OFF pixels*. Once shape is liberated from the constraining bonds of connected components, texture can be viewed as the statistical distribution of shapes in the image. Seen this way, the most important measures of texture are the *densities* of a set of salient shapes. Our intuition can then be used to suggest appropriate shapes for a particular analysis. For example, to distinguish italicized text, one can choose a shape consisiting of slanted edges (adjacent slanted sets of ON and OFF pixels, not slanted sets of ON pixels).

Second, when aggregating shape over larger regions, the concepts of shape and texture actually merge. To distinguish between regions of kanji and roman text, one might note the high density of thin, vertically adjacent parallel lines in some kanji characters. One might use horizontal edges as a shape feature and look for distributions of these features in close vertical proximity. Or one might use alternating ON and OFF pixels, closely adjacent vertically, as the shape feature, and find the regions of high density (texture) where these features are found in close horizontal proximity. Or one can describe the shape and texture together with a single shape feature. The merging is even more apparent when we consider that the vertically alternating ON and OFF pixel shape feature would more traditionally be viewed as a texture measure, because it extends over several cycles of ON and OFF pixels, and thus establishes a regular repeating pattern over a scale large compared to the periodicity. Thus, a salient characteristic of kanji-ness can be viewed locally as shape, or as various combinations of shape and texture, or even as a combination of two textures!

Third, shape and texture information can be structured hierarchically at different scales. The variability of images and image quality leads to uncertainty in our ability to specify or extract features. In the kanji text example, where the goal is to identify some fraction of the kanji characters, false positive “responses” from other regions must also be expected. A texture measure of positive kanji responses can be used at a larger scale to differentiate between the higher density *signal* from kanji regions and the low density *noise* of isolated errors.

Fourth, and of great importance, the shape and textural properties of an image can be altered in specific ways when transforming to lower resolution. The goal is to separate regions differing in shape and textural properties by choosing a set of reduction operations that differentially transform pixels within the regions. The tools for extraction and transformation of shape and texture are described in the next section.

1.3 The plan

How can an efficient and effective image analysis system be constructed? Let us adopt the following set of design criteria:

1. Image features are extracted at the lowest feasable resolution.
2. The primitive operations are implemented uniformly on the image and with the greatest parallelism permitted by the hardware.
3. Boolean operations on binary images are used whenever possible.
4. (Nearly) all operations are carried out on the image, and on appropriately reduced versions of the image.
5. Both shape (broadly defined) and textural features are to be used, as appropriate.
6. Separation of regions with different texture is accomplished by operations that either differentially transform texture with scale change, or differentially project texture components at constant scale.

Implementation of this programme is described in the remainder of the paper. In the next section, the fundamental tools of multiresolution morphology are presented. New image operations, such as the generalized opening and threshold reduction are introduced, with an attempt both to explain the need for these operations and to provide an intuitive understanding of their actions. The use of these tools on problems such as halftone and word segmentation, and the identification of words in italic and bold type styles, is outlined.

All algorithms have been implemented in C. Performance measures given here are CPU time on a Sun Sparcstation2^{TM}.

2 Tools of multiresolution morphology

The morphological tools consist of the standard transformations, augmented by a new operation, the generalized opening, that is effective for extraction of both shape and texture at a given scale. The multiresolution (reduction) operations consist of a generalization of morphological operations (called rank-order filters), followed by subsampling. All operations are image-to-image.

2.1 Morphological operations

The fundamental morphological operations, *erosion* and *dilation*[8, 5], are most efficiently implemented by translating the image and either ANDing or ORing it with itself. Specifically, letting X represent the binary image and the (usually) small set A represent the *structuring element* (SE), the *erosion* $\ominus$ and *dilation* $\oplus$ of X by A are defined as

$$X \ominus A = \bigcap_{z \in A} X_{-z} \tag{1}$$

$$X \oplus A = \bigcup_{z \in A} X_z \tag{2}$$

where X_z is the *translation* of X along the pixel vector z. and the set intersection and union operations represent bitwise AND and OR, respectively. These operations can be implemented as raster operations to take advantage of the word-parallel representation of the pixels within a computer.

From these two operations, the *opening* ○ and *closing* ● operations of X by A are defined, respectively, as

$$X \circ A = (X \ominus A) \oplus A \tag{3}$$
$$X \bullet A = (X \oplus A) \ominus A \tag{4}$$

Unlike the erosion and dilation alone, which continue to transform the image if repeated, the opening and closing are idempotent. They are also independent of the location of the reference position of the SE A. Also, unlike erosion and dilation, opening and closing are anti-extensive and extensive, respectively, which means that for any SE A,

$$X \circ A \subseteq X \tag{5}$$
$$X \bullet A \supseteq X \tag{6}$$

To handle patterns consisting of both ON and OFF pixels, Serra[8] generalized the erosion by defining a *hit-miss transform*, HMT, of an image X by a disjoint pair (A, B) of SEs as the set transformation

$$X \otimes (A, B) = (X \ominus A) \cap (X^c \ominus B) \tag{7}$$

where A and B are referred to as the "hit" and "miss" SEs, respectively. The HMT is useful for matching to general patterns. To extend such general matching to operations that have the special properties of the opening and closing, we have defined[1] a *generalized opening* of X by (A, B) as the set transformation

$$\Psi(X; A, B) = [X \otimes (A, B)] \oplus A. \tag{8}$$

The generalized opening is an HMT followed by a dilation with the hit SE A. It has a very simple geometrical interpretation. Just as the opening is a set consisting of the union of SEs for all matches of the SE to the image, the generalized opening is the union of hits A for all matches of the SE (A, B) to the image. It can be shown that the generalized opening, like the opening, is anti-extensive, center-independent and idempotent[1]. Thus, it can be used to filter general patterns in the same way that the opening can be used to project out patterns of ON pixels.

An erosion or HMT requires an exact match. An imperfect match, called a *rank order filter* or, equivalently, a *threshold convolution*, is a generalization of the erosion and dilation operations of morphology. The dilation is a threshold convolution where the threshold is 1; an erosion represents a threshold equal to the cardinality of the SE. The rank order filter can be used to compensate for variability in the image, but often at a significant cost in computational complexity. Nevertheless, as described in the next section, small rank order filters have been found to be extremely useful for multiresolution operations.

Using the word-parallelism inherent in the internal representation of a binary image, the basic morphological operations (1) and (2) can be implemented efficiently in software on a general purpose computer. The speed of such raster operations is conveniently measured in millions of boolean operation per second (Mbops). A Sun Sparcstation2^{TM} performs about 50 Mbops on large images.

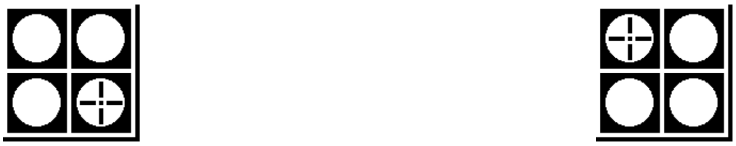
2.2 Multiresolution operations

Given the necessity of representing images at multiple scales for efficient analysis of content, how are we to effect the reductions? Regular subsampling of the image is useful for reducing the number of pixels. The density of the original image is typically (although not always) preserved. Haralick[6, 7] has emphasized the importance of using a low-pass filter prior to subsampling, to prevent aliasing of high frequency components, with a view toward minimizing differences between the subsampled and higher resolution image. Burt[4] has shown that multiresolution methods with small filters are capable of accurately characterizing texture at multiple scales. Preservation of image qualities may be useful for measurement, compression, reconstruction, and rendering, where fidelity to the original is paramount. However, it is not appropriate for aspects of image analysis where the intention is to alter different regions of the image preferentially, depending on the local textural and shape qualities.

Which filtering operations should be chosen? Consider the problem of segmenting an image into text and halftone image regions. A brute-force morphological approach might be to close the image with sufficiently large SEs to solidify the halftone parts, and then open the image with even larger SEs to remove the (somewhat blocked up but smaller) text parts. The opening would not affect the solid halftone regions, and the result would be a separation mask covering only the halftone areas. The closing removes OFF pixels that are “near” ON pixels, and the opening removes ON pixels that are “near” OFF pixels, with the scale of “near” given by the size of the respective SEs. Both operations can be viewed as alteration of short-range image texture.

This suggests that filtering operations before subsampling should be chosen to change the image texture so as to mimic operations at full scale. To solidify pixels within halftone regions, use a closing or dilation operation before subsampling; then at reduced scale, use an opening or erosion before further subsampling. Because it is expensive to use large SEs at high resolution, we can effect an arbitrary and efficient 2^n reduction by a cascade of n 2-fold reductions, pre-filtering with 2x2 SEs at each step.

Thus we tile the image into 2x2 squares and subsample the upper-left pixel of each tile. Consider the 2x2 SE whose reference position is located in the lower-left corner (Figure 1a). Dilation with this SE prior to subsampling is equivalent to setting a threshold of 1 ON pixel in the 2x2 pixel tile: if at least 1 pixel is ON, after dilation the pixel to be subsampled will surely be ON. Likewise, use of an erosion by the SE shown in Figure 1b prior to subsampling is equivalent to setting a threshold of 4 ON pixels in the tile: all four pixels in the tile must be ON if the subsampled pixel is to be ON. Clearly, flexibility is gained by generalizing to allow filters that threshold on 2 and 3 ON pixels within the tile. The requisite filtering operations are threshold convolution (or, equivalently, rank order filters) mentioned in the previous section.



(a) (b)

Figure 1. (a) Dilation filter for threshold 1; (b) erosion filter for threshold 4

We call the combination of a threshold convolution followed by subsampling a *threshold reduction*.

The halftone segmentation problem is efficiently addressed by a sequence of 2x reductions using a small threshold, say 1, to consolidate the halftone textured region, followed by further 2x reductions with a large threshold, say 4, to remove the text regions. Larger atomic reductions with more threshold levels for pre-filtering can also be used, but in practice we have found that the four different 2x threshold reductions provide sufficient flexibility.

It is not necessary to carry out the full threshold convolution before subsampling, because only one out of every four pixels is actually subsampled. Threshold reduction is efficiently implemented by forming a half-height, full-width image using a logical operation (OR or AND) between each odd row and the even row below it, and following this with a reduction to a half-height, half-width image using a lookup table that emulates similar column-wise logical operations. If both row and column operations are OR, each ON pixel in the reduced image is ON if any of the four pixels in the corresponding tile of the original image were ON. This is a threshold 1 reduction. Likewise, if both operations are AND, we get a threshold 4 reduction. It takes approximately twice as much work to reduce images with threshold values of 2 and 3. To do this, form two intermediate half-height, half-width images, using OR-AND and AND-OR for the row and column operations. The threshold 2 and threshold 3 reduced images are then found by taking the *union* and *intersection* of these intermediate reduced images, respectively. This implementation of threshold reduction is relatively fast. A 2x reduction using thresholds 1 or 4 proceeds at 25 Mpixel/sec; thresholds 2 and 3 proceed at 12 Mpixel/sec.

The effect on texture from a sequence of threshold reductions is fairly predictable. For example, a set of four sequential threshold 1 reductions is approximately equal to a dilation with a 16x16 brick SE, followed by subsampling. Pairs of ON pixels separated by less than about 16 pixels will typically be joined. Likewise, four sequential threshold 4 reductions are roughly equivalent to an erosion with a 16x16 brick SE, followed by subsampling. Regions of ON pixels smaller than such a brick will typically vanish in the reduced image. Just as dilation and erosion expand and shrink regions of ON pixels, threshold reduction using thresholds of 1 and 4 tend to expand and shrink solid regions, respectively, and some compensation may be required. The subsampling operation is not translationally invariant; consequently, some variation is to be expected due to the positioning of the 2x2 tiles on the image. A more detailed description of these transforms and their properties is given in [2].

3 Illustration: Font style identification

Omni-font OCR systems do not attempt to distinguish between different type styles, because on a single character basis the problem is relatively difficult. Yet, as mentioned at the beginning of this paper, italic words are instantly apparent visually. The usefulness of identifying different type styles goes beyond providing a nice feature for an OCR system. Italicized words typically have special significance in a document. If identified, they can be used as keywords for automatic indexing into a database of scanned document images. In this section, we outline how multiresolution morphology provides a simple and fast method for identifying italic words. The image is segmented with a granularity size of the *word*, rather than the individual *character*, because segmentation at the word level can be done reliably and independently of OCR functions. A related approach for bold words is also described.

The general method consists of the following steps:

1. Identify a feature or set of features (or even the lack of a feature!) that distinguishes the region

of interest from the rest of the image.

2. Reduce the image if possible and project the feature(s) out at the lowest reliable resolution.

3. Remove the noise (false positives) in the texture pattern resulting from the previous step. The resulting pixels constitute a *seed*.

4. Construct a *mask* at the same resolution, in which each connected component covers pixel sets in the original image at the selected level of granularity (in this case, the word).

5. Fill from the seed into the mask, resulting in a *selection mask* covering only those pixel sets with the selected property.

6. Use the selection mask to extract corresponding regions in the full resolution image.

The resolution of images displayed in this section is given in units that are independent of the resolution of the physical rendering device. All images are labelled with both the *sampling resolution*, in pixels/inch, and the *rendering resolution*, also in pixels/inch. The sampling resolution gives the size of the sampled (or subsampled) pixels in the image, whereas the rendering resolution gives the size of these pixels as rendered on the page. The *magnification* can be found as the ratio of sampling to rendering resolution.

For italics, the distinguishing feature is *edges* inclined at about 12 degrees from the vertical. (Edges are used, rather than slanted lines, to avoid responses from large solid blocks of pixels). The left edge is better than the right edge, because slanted right edges are more pronounced in words like “wavy”. We choose the SE shown in Figure 2.

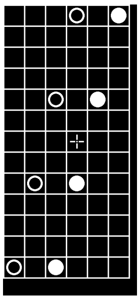


Figure 2. Structuring element for weak match to left edges of italic lines.

Four properties of this SE should be noted:

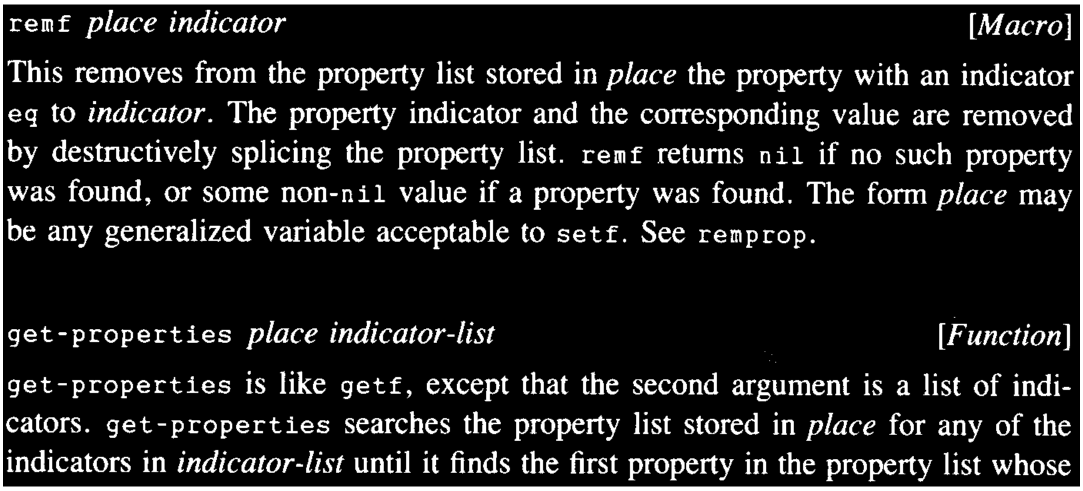
1. Both hits and misses are used. This is necessary to find edges.

2. The hits and misses are oriented along a slant, with the hits on the right to match a left edge.

3. A “don’t-care” is placed horizontally between each hit and miss.

4. There are three don’t-care lines vertically between each hit-miss pair.

This is a “weak” filter, designed to match left edges with both edge noise and variation in slant angle. A generalized opening produces an image textured with pixels representing a local measure of italic-ness.



`remf` *place indicator* [*Macro*]

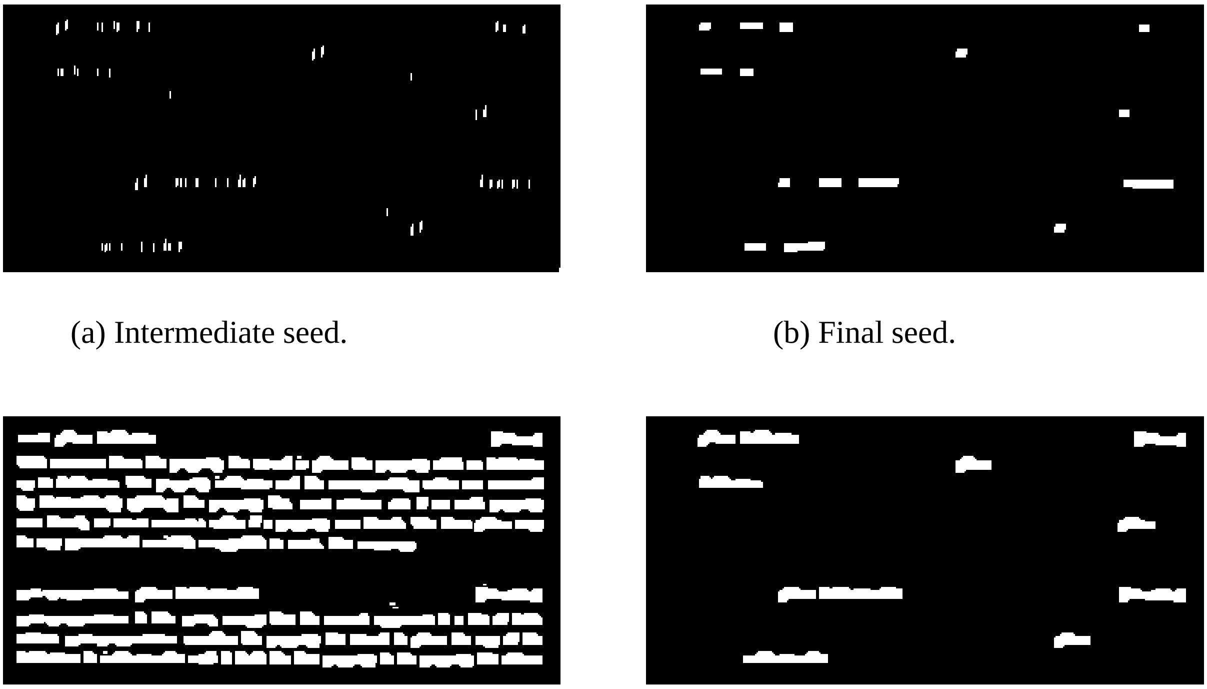
This removes from the property list stored in *place* the property with an indicator `eq` to *indicator*. The property indicator and the corresponding value are removed by destructively splicing the property list. `remf` returns `nil` if no such property was found, or some non-`nil` value if a property was found. The form *place* may be any generalized variable acceptable to `setf`. See `remprop`.

`get-properties` *place indicator-list* [*Function*]

`get-properties` is like `getf`, except that the second argument is a list of indicators. `get-properties` searches the property list stored in *place* for any of the indicators in *indicator-list* until it finds the first property in the property list whose

Figure 3. Example of text with italic font style.
Sampling resolution: 300/inch. Rendering resolution: 250/inch.

Consider the text image shown in Figure 3. The seed to be generated must have at least one ON pixel in each italic region and no ON pixels elsewhere. If the SE has been well-chosen, all italic regions will have some ON pixels and the texture of filtered pixels within italic regions should be differentiable from those without. For processing efficiency, two sequential 2x threshold 1 reductions are used. Noise (isolated pixels) is removed by a close/open sequence; the close joins and protects seed pixels, and the open removes un-coalesced noise. The close/open is preceeded by a small vertical dilation to guarantee horizontal alignment of seed pixels. Intermediate and final seed images are shown in Figures 4a and 4b.



(a) Intermediate seed.

(b) Final seed.

(c) Word mask.

(d) Final selection mask.

Figure 4. Steps in making seed, mask, and selection mask.
All images are sampled at resolution 75/inch; rendering at 120/inch.

The word mask (Figure 4c) is generated at 4x reduction, using a cascade of two 2x threshold 1 reductions on the original image, followed by a small horizontal dilation or close.

The separation mask (Figure 4d) is then generated by a filling operation, starting with the seed and clipping to the word mask. This is then used to extract the italic words at full resolution. Two methods are possible. The separation mask can be expanded 4x and ANDed with the full resolution image to produce an image with only italic words. Alternatively, the bounding box coordinates of the separation mask word components can be determined and used, when appropriately scaled, to extract the italic words individually from the full resolution image. One can attempt to extract italic words with only a seed. However, use of a word mask both improves signal/noise discrimination and permits reliable extraction of words as atomic units.

These methods are quite robust on a variety of fonts and typical font sizes. Improvements are possible by adjusting the SE sizes to the font size, which can be measured in several different ways. The operations proceed at about 1.3 Mpixel/sec.

Extraction of words in bold type style is a more difficult problem than italic. Boldness is most easily characterized by character stem widths. However, stem widths in a scanned image may differ greatly from that intended in the original printing, because the page could have been copied several times (with highly variable results) and because the scanner threshold may not have been set to reproduce this measure accurately. Thus, boldness must be determined as a *relative* measurement.

The procedure for bold type style extraction is similar to that for italic, with the differences occurring in the feature extraction step. The following method meets with some success. Progressively thin the image in the horizontal direction, alternately removing pixels from left and right ends of each horizontal run of ON pixels. At each thinning iteration, open the result with a vertical SE sufficiently large to respond primarily to the vertical stems, and, using a lookup table, count the number of ON pixels (or, better, non-zero bytes) in the image. Construct the ratio of the counts on successive iterations. If there is a small amount of bold text intermixed with normal widths, at some point the majority of the text stems will disappear, and the count ratio will suddenly decrease below a threshold (typically about 2.5:1). When this happens, stop the thinning process and use the most recent image, after opening for vertical stems, as the feature image out of which the seed is constructed. Bold word extraction is slower than italic, proceeding at about 0.5 Mpixel/second.

4 Discussion

It may initially seem surprising that we have built an image analysis system based on image processing, rather than on a set of rules. (A small set of heuristics about document images are necessary, and are implicitly embodied within structuring elements). After all, image analysis is a process of making decisions about the image, and explicit rules are traditionally used to represent cognitive understanding of the decision process. However, the *subcognitive* basis of human visual perception suggests using an image-based approach. Because the decision-making process is inherently nonlinear, it is necessary to use nonlinear image processing techniques. Such methods, as described in this paper, are capable of transforming the image in such a way that the final steps in the analysis, *if any,* are primarily counting and labelling activities. These binary image transformations are highly nonlinear, in that threshold decisions are made on each pixel at each step. (The operations are also generally irreversible and noncommutative). At the end of an analysis, any given pixel may have had many such threshold decisions, based on the values of other pixels in its vicinity. Furthermore, in a reduced image, each pixel is influenced by the value of a larger number of pixels in the full resolution image. As a result, systems can be built that both have sufficient nonlinearity to defy exact analysis, and yet operate in accordance with our geometrical intuition to make definite statements about an image.

It is often necessary obtain a statement *about* the image, such as the number of pixels or the number or location of components. This is typically done as the last stage in the analysis, when an image mask has been created at low resolution. Thus, a useful system must include operations for labelling connected components. As a general guideline, most of the image analysis (possibly exclusive of that directly related to OCR) should be within the image domain. If computation related to connected components or other complex data structures occupies a sizable fraction of the whole, then the algorithm probably does not take sufficient advantage of multiresolution morphology.

The operations described here are particularly well-suited for extraction of large image features. As described in Section 2.2, the combination of threshold reduction and supporting morphological operations can separate image and halftone regions from text and line graphics simply by carrying out a sequence of four successive threshold reductions. Most of the computation is in the first two reductions, which can be done with threshold 1. The segmentation time for an 8 Mpixel image is about 0.5 second; not human performance, but respectable. Likewise, the generation of a word mask, used as an intermediate step in type style segmentation, is implemented as a cascade of two 2x threshold 1 reductions, and consequently proceeds at a rate of nearly 20 Mpixel/second.

Other large scale image features that are apparent to the eye, even (or, especially) when viewed at considerable distance, such as rules and text columns, are also easily extracted using these methods. When blocking up text columns, it is advantageous to remove vertical rules first, because this increases the gutter size. Such rules are easily removed by opening the image with a small horizontal SE.

The examples in this short paper demonstrate a few applications of multiresolution morphology in the domain of document images. The strength of the approach—its simplicity, efficiency, and appeal to intuition—derive from the use of the image as the fundamental structure on which computation is performed. The versatility of these methods is perhaps best appreciated by the following game. Ask yourself a question about the image—its composition, layout, or whatever—that can be answered definitively by "yes", "no", or a small integer, or by the construction of an image mask. Then imagine a small set of image operations that will answer the question.

References

[1] D. S. Bloomberg and P. Maragos, "Generalized Hit-Miss Operations," in *SPIE Conf. on Image Algebra and Morphological Image Processing, Vol. 1350*, pp. 116-128, July 1990.

[2] D. S. Bloomberg, "Image Analysis using Threshold Reduction," in *SPIE Conf. on Image Algebra and Morphological Image Processing II, Vol. 1568*, San Diego, CA, July 1991.

[3] P. J. Bones, T. C. Griffin, and C. M. Carey-Smith, "Segmentation of document images," *SPIE Symp. on Electronic Imaging Science and Technology, Vol. 1258*, Feb. 1990.

[4] P. J. Burt, "The Pyramid as a Structure For Efficient Computation", *Multiresolution Image Processing and Analysis*, pp. 6-35, Berlin: Springer, 1984

[5] R. M. Haralick, S. R. Sternberg and X. Zhuang, "Image Algebra Using Mathematical Morphology," *IEEE Trans. PAMI-9*, July 1987.

[6] R. M. Haralick, C. Lin, J. Lee, X. Zhuang, "Multi-resolution morphology," *Int. Conf. on Computer Vision, London* pp. 516-520, June 1987.

[7] R. M. Haralick, X. Zhuang, C. Lin and J. Lee, "Binary Morphology: Working in the Sampled Domain," *CVPR '88, Ann Arbor, MI* pp. 780-791, June 1988.

[8] J. Serra, *Image Analysis and Mathematical Morphology*, Acad. Press, 1982.

[9] F. M. Wahl, K. Y. Wong, and R. G. Casey, "Block Segmentation and Text Extraction in Mixed Text/Image Documents", *CGIP, Vol 20*, pp. 375-390, 1982.

[10] D. Wang and S. N. Srihari, "Classification of Newspaper Image Blocks Using Texture Analysis", *CVGIP, Vol 47*, pp. 327-352, 1989.

[11] K. Y. Wong, R. G. Casey, and F. M. Wahl, "Document Analysis System", *IBM J. Res. Dev., Vol 26*, pp. 647-656, 1982.