Context-based Multiscale Classification of Document Images Using Wavelet Coefficient Distributions

Jia Li, Robert M. Gray

Abstract

In this paper, an algorithm is developed for segmenting document images into four classes: background, photograph, text, and graph. Features used for classification are based on the distribution patterns of wavelet coefficients in high frequency bands. Two important attributes of the algorithm are its multiscale nature—it classifies an image at different resolutions adaptively, enabling accurate classification at class boundaries as well as fast classification overall— and its use of accumulated context information for improving classification accuracy.

Keywords

document image segmentation, text and photograph segmentation, multiscale classification, context-dependent classification, wavelet transform, goodness of match.

I. Introduction

The image classification problem considered in this paper is the segmentation of an image into four classes: background, photograph, text, and graph. *Photograph* refers to continuous-tone images, such as scanned pictures. *Text* is interpreted in the narrow sense of ordinary bi-level text. *Graph* includes artificial graphs and overlays that are produced by computer drawing tools. *Background* refers to smooth blank regions. This type

Jia Li is currently with Xerox Palo Alto Research Center, Palo Alto, CA 94304. Robert M. Gray is with the Information Systems Laboratory, Department of Electrical engineering, Stanford University, CA 94305, U.S.A. Email: jiali@isl.stanford.edu, rmgray@stanford.edu. This work was supported by the National Science Foundation under NSF Grant No. MIP-931190 and by a gift from Hewlett-Packard, Inc.

of classification is useful in a printing process for separately rendering different local image types. It is also a tool for efficient extraction of data from image databases [24].

Previous work on gray-scale document image segmentation includes Chaddha et al. [2], Williams and Alder [27], and Perlmutter et al. [14], [13]. Thresholding is used to distinguish text and non-text for video images in [2]. In [27], a modified quadratic neural network [11] is applied to segment newspaper images into photograph and non-photograph. In [14], [13], the Bayes VQ algorithm is applied. Another type of document image classification studied more often is the segmentation of half-tone (binary) document images [18], [6], [7], [23], [25], [17], [5], [12]. For binary images, run-length statistics, e.g., the average horizontal run-length of black dots, are important for distinguishing text and photograph [18], [28], [5].

A critical part of a classifier is the extraction of good features for classification. In our algorithm, features are formed by wavelet transforms [16], [10], which have played an important role in categorizing textures [21], [9] and detecting abnormalities in medical images [4], [26]. The principal approach of wavelet-based classification is to form features by statistics of wavelet coefficients, among which moments are the most commonly used [21], [9], [4], [26]. In this paper, however, direct attention is paid to the sample distribution or histogram pattern of wavelet coefficients. Features are defined according to the shape of a histogram.

Another important issue in classification is the extent to which a classifier should be localized to decide the class of an area. In general, if an image region is of a pure class, the larger the region is, the easier it is to identify the class. It is usually difficult, even for human beings, to classify a small region without context. Indeed, it may be impossible to judge correctly a small region's class without context because the same region may appear in areas with different classes. On the other hand, classification performed on large blocks is crude since a large block often contains multiple classes.

To overcome the conflict of over-localization and crude classification, a multiscale architecture is proposed. The multiscale classifier begins with large blocks and no context and, if necessary, then moves to smaller blocks using context extracted from larger blocks. Initially, only blocks with extreme features are classified. The failure to provide distinguished features by an unclassified block is likely to result from mixed classes in the block. The classifier, therefore, increases its scale or resolution, i.e., subdivides unclassified blocks for further

classification. Context information obtained from the classification at the first scale is used to compensate for the smaller block size at the higher scale.

To sum up, the classifier proceeds from low scales to high scales with context information being accumulated. Since more context information is available when the classifier examines smaller blocks, the conflict of over-localization and crude classification is avoided to a certain extent. The idea of classifying across several scales has been applied in previous work [8], [20], but our viewpoint is quite different. A typical way to benefit from multiscale analysis is to extract features across several scales so that the properties of an image block at multiple resolutions can be used for classification. In our algorithm, the multiscale structure is applied to efficiently build context information, which incorporates both the properties across multiple resolutions and the properties of surrounding areas into classification. The number of scales used is naturally adaptive so as to avoid unnecessary computation due to multiscale analysis.

In Section II, classification features are defined. The algorithm is described in Section III. In Section IV, results are presented. Conclusions are drawn in Section V.

II. Classification Features

A. Distribution of Wavelet Coefficients in High Frequency Bands

It has been observed based on many document images that for photographs, wavelet coefficients in high frequency bands, i.e., LH, HL, and HH bands [22] as shown in Figure 1, tend to follow Laplacian distributions. Although this approximation is controversial in some applications, it will be seen to work quite well as a means of distinguishing continuous-tone images from graph and text by means of the goodness of fit. To visualize the different distribution patterns of the three image types, the histograms of Haar wavelet coefficients in the LH band for one photograph image, one graph image, and one text image are plotted in Figure 2.

In addition to the goodness of fit to Laplacian distributions, another important difference to note in Figure 2 is the continuity of the observed distributions. The histograms of the graph image and the text image suggest that the coefficients are highly concentrated on a few discrete values, but the histogram of the photograph image shows much better continuity of distribution. In the special case shown in Figure 2, the histogram of

the text image contains five bins far apart from each other because the text image has bi-level intensities. For the graph image, a very high percentage of data locate around zero. Although the concentration of data around zero is not absolute, the amount of nonzero data is negligible. For the photograph image, there does not exist any value that similarly dominates. Instead, the histogram peaks at zero and attenuates roughly exponentially. It is worth pointing out that in practice histograms of the three image types are usually not as extreme as the examples shown here. Ambiguity occurs more with graph because it is intermediate between photograph and text.

Based on the observations, two features are extracted according to the histograms of wavelet coefficients in high frequency bands. The first one is the goodness of match between the observed distribution and the Laplacian distribution. The second one is a measure of the likelihood of wavelet coefficients being composed by highly concentrated values.

B. Goodness of Fit to the Laplacian Distribution

In order to measure the goodness of match between an observed distribution and a Laplacian distribution, the χ^2 test [19] normalized by the sample size N, denoted by $\bar{\chi}^2$, is used. The χ^2 test is a widely applicable measure of how well a set of observed data matches a specified theoretical distribution.

Given a random variable X, assume that the theoretical probability density function of X is $p(t)$, $t \in \Re$. The samples of X are $\{x_1, x_2, \ldots, x_N\}$. To test whether the samples are drawn from the distribution $p(t)$, the sample set is divided into k categories. The number of x_j's in category i is denoted by m_i. Usually, the categories are consecutive intervals in the domain of the data. Specifically, if we denote category i by C_i, then

$$C_i = \{x : \alpha_i < x \leq \alpha_{i+1}\} \ , \ i = 1, \ldots, k, \ \alpha_1 < \alpha_2 < \ldots < \alpha_{k+1} \ .$$

The relative frequency f_i for the ith category C_i is

$$f_i = m_i / N \ .$$

According to the theoretical distribution, the expected frequency F_i for C_i is

$$F_i = \int_{\alpha_i}^{\alpha_{i+1}} p(t)dt \, .$$

Thus, the expected count for x_i being in C_i is

$$M_i = N \cdot F_i \, .$$

The χ^2 test criterion is defined as

$$\chi^2 = \sum_{i=1}^{k} (m_i - M_i)^2 / M_i \, .$$

Pearson [19] showed that if the observed data are randomly drawn from the theoretical distribution, then the test criterion follows the theoretical χ^2 distribution in large samples. If the observations come from some other distribution, the observed f_i tends to deviate from the expected F_i and the computed χ^2 becomes large.

If we test the null hypothesis [19] H_0: the observations are randomly drawn from a specified theoretical distribution, a computed χ^2 greater than $\chi^2_{0.050}$ causes rejection of H_0 at the 5% significance level [19]. The value of $\chi^2_{0.050}$ is calculated according to the theoretical χ^2 distribution. A key parameter for the χ^2 distribution is the number of degrees of freedom. In this case, the number of degrees of freedom is $k - 1$, where k is the number of categories to which x_i's belong. If the variance of the theoretical distribution is estimated by the sample variance [19] of the observed data, the number of degrees of freedom is then $k - 2$.

There are many other methods to test the null hypothesis H_0, for example, the Kolmogorov test [1]. We have studied χ^2 in a greater depth because our goal is to obtain good features for distinguishing image types rather than to test the literal truthfulness of the hypothesis. The χ^2 statistic is chosen because deviation from an assumed distribution in every data bin C_i is taken into consideration and this statistic is easy to compute.

To measure the goodness of match between an observed distribution and a Laplacian distribution, we need

to determine the parameters of the Laplacian distribution. Recall that the pdf of a Laplacian distribution is

$$p_X(x) = \frac{\lambda}{2} e^{-\lambda|x|} \qquad -\infty < x < \infty \ , \ \lambda > 0 \ .$$

Since $VAR[X] = 2/\lambda^2$, the parameter λ is estimated by moment matching, i.e.,

$$\hat{\lambda} = \sqrt{\frac{2}{\hat{\sigma}^2}} \, ,$$

where $\hat{\sigma}^2$ is the sample variance of X.

One classification feature is defined as χ^2 normalized by the sample size N, denoted by $\bar{\chi}^2$, which is defined by

$$\bar{\chi}^2 = \chi^2/N = \sum_{i=1}^{k} (f_i - F_i)^2/F_i \ .$$

The feature is χ^2 normalized by N because we are interested in how close f_i and F_i are instead of whether the null hypothesis H_0 should be accepted. When the sample size is large, the observed relative frequency converges to its true distribution, which cannot really be a Laplacian distribution since x_i is bounded. Therefore, χ^2 increases approximately linearly with the sample size.

C. Likelihood of Being a Highly Discrete Distribution

A criterion, denoted by L, is defined to measure the likelihood of wavelet coefficients being composed by highly concentrated values. For example, Figure 2(c) shows that data only lie in the five far apart bins, indicating a high L. The efficiency of L estimating the likelihood of wavelet coefficients obeying a highly discrete distribution depends on how completely concentrated peak values can be detected, and how robust the identification of these values is to local fluctuations.

Since for highly concentrated data, their absolute values have the same property, histograms of absolute values are used to calculate L to save computation. An example histogram shown in Figure 3 is analyzed

first for gaining intuition regarding to a proper definition of L. It seems reasonable to assume that V_0 and V_1 are two concentration values. First of all, they are both peak values (maximum values) in a data range $[0, Z_1]$ and $[Z_1, Z_2]$. In addition, most of the data in either of the two data ranges are distributed in a narrow neighborhood of V_0 or V_1; and the number of data points vanishes towards the end points. On the other hand, although V_2 is a local maximum in $[Z_1, Z_2]$, it is not regarded as a concentration value since it appears to result from a fluctuation around V_1; V_3 and V_4 are not considered as concentration values because neither of them forms a tight cluster by most data in the range $[Z_2, Z_3]$. To summarize from the example, a concentration value is expected to possess the following properties:

1. A concentration value is the maximum in a certain data range, say $[Z_1, Z_2]$. Such a data range is defined as a 'zone.'
2. Most of the data in the range $[Z_1, Z_2]$ are distributed in a narrow neighborhood of the peak value.
3. The amount of data is vanishingly small at the ends of $[Z_1, Z_2]$.

In order to locate concentration values, an essential step is to divide data into zones based on the histogram. Intuitively, we imagine a zone to be a range, isolated from the remainder of the data axis, in which data clump. The histogram in a zone should have a peak value and vanish towards the ends of the zone. A more rigorous definition is provided to capture these ideas.

Suppose the complete data range is $[0, M]$. The histogram in $[0, M]$, normalized by the total number of samples, is represented by a nonnegative unit integral function $h(t), t \in [0, M]$. The interval $[0, M]$ is partitioned into connected zones $[t_0, t_1], [t_1, t_2], \ldots,$ and $[t_{r-1}, t_r]$, where $t_0 = 0$ and $t_r = M$. An interval $[t_i, t_{i+1}]$ is a zone if it satisfies the following conditions:

1. If $t_i \neq 0$ and $t_{i+1} \neq M$,
   (a) There exists a $t^* \in (t_i, t_{i+1})$, s.t. $h(t^*)$ is the maximum value for $t \in [t_i, t_{i+1}]$.
   (b) $h(t_i) \leq h(t)$, for $t \in (t_i, t^*)$
   $h(t_{i+1}) < h(t)$, for $t \in (t^*, t_{i+1})$
   (c) $h(t_i)/h(t^*) < \delta$, and $h(t_{i+1})/h(t^*) < \delta$, where δ is a threshold, which is set to be 0.05 in our implementation.

(d) There does not exist $[\tau, \tau'] \subset [t_i, t_{i+1}]$ and $[\tau, \tau'] \neq [t_i, t_{i+1}]$, so that $[\tau, \tau']$ satisfies the above three conditions.

2. If $t_i = 0$ and $t_{i+1} \neq M$, the four conditions are almost the same as those of the previous case, except for two relaxed conditions listed below:

(a) The maximum point t^* is allowed to appear at the end point $t_i = 0$.

(b) It is not required that $h(t_i)/h(t^*) < \delta$ and $h(t_i) \leq h(t), \text{for} \quad t \in (t_i, t^*)$.

3. If $t_{i+1} = M$, the only condition for $[t_i, t_{i+1}]$ to be a zone is:

(a) There does not exist $[\tau, \tau'] \subset [t_i, t_{i+1}]$ and $[\tau, \tau'] \neq [t_i, t_{i+1}]$, so that $[\tau, \tau']$ satisfies the previous definition for a zone.

In the above list of conditions, the first case applies to a normal zone, while the second and the third cases relaxed conditions for zones at the ends of the data range $[0, M]$. For a normal zone, the first condition requires that the zone contains an internal maximum point. This maximum point is the candidate concentrated value in the zone. The second condition ensures that each end point of the zone is a local minimum in the entire data range and a global minimum in the interval between the end point and the maximum point in the zone. The third condition requires that the empirical probability density at each end point of the zone is significantly smaller than the maximum in the zone. This requirement guards against local fluctuations of the distribution. In addition, the combination of the second and the third conditions forces the histogram to vanish towards the ends of the zone. The fourth condition guarantees the partition of $[0, M]$ being the finest possible. The second and the third cases specify the conditions for a zone located at one end of the data range. Some requirements in the first case are discarded so that a zone partition always exists for a data range with continuous probability density function. Every zone in a partition may have a different level of concentration. In the special case when the distribution is sufficiently smooth, the entire data range is identified to be one zone, and the concentration level of the zone is nearly zero.

The algorithm for finding the partition of $[0, M]$ is described in the Appendix. To define L, suppose that the partition has been obtained. Let us start with defining a concentration level for every zone. Then, L is calculated as a weighted sum of these concentration levels.

For a zone $[t_i, t_{i+1}], i = 0, 1, \ldots, r-1$, denote the concentration level by β_i. Let the maximum point in the zone be t^*. The width of the neighborhood around t^*, in which data are considered to be sufficiently close to t^*, is set to w. Subject to the constrained data range, the two end points of the neighborhood around t^*, denoted by τ_1 and τ_2, are thus

$$\tau_1 = \begin{cases} t^* - w & t_i < t^* - w \\ t_i & \text{otherwise .} \end{cases}$$

$$\tau_2 = \begin{cases} t^* + w & t^* + w < t_{i+1} \\ t_{i+1} & \text{otherwise .} \end{cases}$$

The probability of being in $[\tau_1, \tau_2]$ is $p_i = \int_{\tau_1}^{\tau_2} h(t)dt$, and the probability of being in $[t_i, t_{i+1}]$ is $p_i' = \int_{t_i}^{t_{i+1}} h(t)dt$. The concentration level β_i is then defined as a thresholded version of the ratio of p_i and p_i', which is

$$\beta_i = \begin{cases} p_i/p_i' & p_i/p_i' > \lambda \\ 0 & \text{otherwise ,} \end{cases} \qquad \text{where } \lambda \text{ is a threshold .}$$

The quantity β_i is set to zero when it is below a threshold in order to increase robustness to noise. Simulations do not show significant difference, however, if the thresholding is not applied. Once the β_i's are obtained, L is evaluated as

$$L = \sum_{i=0}^{r-1} p_i \cdot \beta_i \ .$$

Note that $0 \leq L \leq 1$. The parameter w decreases with the decrease of the sample size of the histogram. The reason is that, for a block of pixels, when the block is small, the pixels are more likely to have similar intensities. In order to be sure that the block has highly concentrated values, the width for closeness should be smaller.

D. Features at Multiple Scales

To perform multiscale classification, features $\bar{\chi}^2$ and L are computed at several scales. At every scale, an image is divided into blocks with the corresponding size. For each block, a histogram is computed based on the high frequency wavelet coefficients of the block. Histograms at different scales are computed using the same bin size.

For a smaller block, the statistics $\bar{\chi}^2$ and L of its histogram are less effective for distinguishing image classes due to the inadequate amount of sample data. This coincides with intuition. When an image block is very small, regardless of its class, it tends to be smooth. The high frequency wavelet coefficients of a very smooth block cluster tightly around zero. Therefore, feature L is close to 1. When the distribution of the wavelet coefficients is close to a Laplacian distribution with a very small variance, the statistic $\bar{\chi}^2$ is nearly zero. The clear separation of large $\bar{\chi}^2$ and L for non-photograph blocks and small $\bar{\chi}^2$ and L for photograph blocks is thus not retained at high scales.

To show the variation of features across scales, two examples are provided in Figure 4. The first example is a photograph image. As was mentioned before, a photograph image tends to have low L's; while a graph or text image tends to have high L's. However, Figure 4(c) shows the trend of L shifting to higher values with the decrease of the block size. As a result, distinguishing image types solely by L at higher scales yields lower classification accuracy. A dual example is provided for a graph image shown in Figure 4 (b). With the decrease of the block size, more $\bar{\chi}^2$'s move towards zero. The motivation of the multiscale classification algorithm is to reduce reliance on the more ambiguous features at higher scales by context information.

E. The Selection of Wavelet Transforms

When we choose a wavelet transform for computing features, a few criteria are considered. First, and most importantly, a good transform should yield distinct features for different image types. Second, the transform is required to have a good localization property. After an image is decomposed by the wavelet transform, coefficients at the same spatial location as a particular block should not be heavily influenced by surrounding blocks. Otherwise, it will be difficult to classify at boundaries between image types. Provided that the two

criteria are satisfied, a wavelet transform with less computation is preferred.

In our system, the one-level Haar wavelet transform is used. As is shown by tests on many document images, the Haar transform provides distinct features for various image types. Comparison with wavelet transforms with longer filters, such as the Daubechies 8 and Daubechies 4 transforms [3], shows that the difference between those transforms in terms of providing good features is negligible. On the other hand, the Haar transform has the best localization property since its wavelet filter is the shortest. For the same reason, this transform costs the least amount of computation.

III. The Algorithm

A. Global Structure

The overall structure of the classification algorithm is presented by a flow chart in Figure 5. The details of each step are explained later. To understand the flow chart, let us clarify notations. A scale is also referred to as a resolution, denoted by r. The maximum resolution allowed in classification, denoted by R, is set by users. The object $CI(r)$ represents the context information already achieved at resolution r. It is essentially a matrix with every element storing the characteristic statistics of a classified image block. The dimension of $CI(r)$ increases with r since an image is divided into more blocks at higher resolutions.

As is shown in the flow chart, the classifier consists of two major parts. The upper block in Figure 5 is the first pass classification, in which an image is classified at the crudest scale and the initial context information $CI(0)$ is generated. The lower block is a closed-loop recursive process. Every iteration corresponds to the classification at one resolution. The classifier starts with resolution zero and increases it in increments of one until the entire image is classified or the resolution exceeds R. With the resolution increased by one, both the width and the height of an image block are reduced by half. At every resolution, if the features of an image block strongly suggest that it is purely text, graph, or photograph, then the block is classified to the corresponding class. Otherwise, the block is labeled as undetermined. The background class is not considered here because background blocks are identified before the multiscale classification, as is shown in Figure 5. It is proper to classify background without the help of the multiscale structure because background blocks

are simply the ones with a unique intensity. Classification at a higher resolution is applied if undetermined blocks exist. At the beginning of classification at a particular resolution r, the context information $CI(r)$ is inherited from the context information obtained at the previous resolution, i.e., $CI(r-1)$. However, instead of being fixed through the classification at resolution r, $CI(r)$ is updated for every newly classified block. The characteristic statistics of every newly classified block is added to $CI(r)$ to serve as context information for later classification.

B. First Pass Classification

In the first pass, the classifier starts with identifying background. As background contains absolutely smooth regions, it is straightforward for the classifier to distinguish them. Computation is saved by marking out background first. The block size used for background classification is usually the same as the block size at the finest resolution. If any subblock of a block in the crude resolutions has been classified as background, its wavelet coefficients are not used for computing features. If the rest of the block is classified as photograph, the subblock keeps its background class. If the rest of the block is classified as text or graph, the class of the subblock is switched to text or graph accordingly, since text or graph blocks often contain absolutely smooth subblocks. This rule results in more globally meaningful classification.

After the background is marked out, classification is performed at resolution zero. Suppose the starting block size is S. For an $S \times S$ block, it is classified as photograph, graph, or text if its features are sufficiently distinguished. Otherwise, the block is labeled as undetermined. For every block classified, its characteristic statistics are stored in $CI(0)$. Recall that the classification features are the goodness of fit to the Laplacian distribution, denoted by $\bar{\chi}^2$, and the likelihood of having a highly discrete distribution, denoted by L. The conditions for a block to be labeled as a particular class are:

1. Photograph: $\bar{\chi}^2 < C_\chi$, where $C_\chi = 0.9$ in our system.
2. Text: $L = 1$ and the pixel intensities in the block concentrate at two values, that is, the image block is nearly bi-level.
3. Graph: $L = 1$ and the pixel intensities in the block concentrate at more than two values; or, $C_l < L < 1$,

where $C_l = 0.9$ in our system.

The thresholds are determined by observing the feature distributions of one image. As the block size in the first pass classification is large, a block with a pure class intends to provide significantly distinguished features. In particular, for a photograph block, $\bar{\chi}^2$ is close to zero; for a graph or text block, $\bar{\chi}^2$ is large and L is close to one. Since the feature distributions of different classes are well separated, the particular values of thresholds are not critical. Sensitivity to the threshold variation is further reduced by the boundary refinement step applied to correct mistakes in the first pass classification.

The characteristic statistics stored in $CI(0)$ for the three image classes are as follows:

1. Photograph: $\bar{\chi}^2$, L, the mean value and standard deviation of pixel intensities in the block.
2. Graph: $\bar{\chi}^2$, L, and the mean value of pixel intensities in the block.
3. Text: The bi-level values of pixel intensities in the block.

C. Global Decisions on Background and Text

In order to distinguish graph from text and background, information based on an entire image is often needed. Although a graph region generated by computer tools is highly likely to contain large smooth areas or bi-level intensity areas, its intensity values usually differ from those of the main background and text in the image. To decide the background intensity and the text bi-level values, an over-all analysis on the image is performed according to the classification result obtained from the first pass. We refer to the background intensity as the dominant background mode, or background mode if no confusion arises. The text bi-level values are referred to as dominant text mode (or text mode).

Observe that the definitions of background mode and text mode are usually more subjective than objective. Our algorithm is designed to align with most subjective judgments. An intensity is chosen as the background intensity for the entire image if it occurs most frequently in blocks classified as background. A similar philosophy applies to determining the text mode. As this part of the algorithm is decoupled from the rest, the criteria can be modified easily by users without side effects. Furthermore, this step is optional in that locally meaningful results can still be achieved without it.

D. Boundary Refinement

As was mentioned before, in order to obtain distinct features, classification is performed on large blocks in the first pass. A large block size is especially important for distinguishing graph from text or photograph because graph is an intermediate between the other two classes. On the other hand, misclassification tends to occur with a large block that contains mixed classes because one class may be so dominant that the classifier marks the entire block as this single class. A boundary refinement mechanism is therefore applied at every resolution to offset this problem. For every block that is newly classified at a certain resolution, if it is adjacent to blocks with different classes, adjustment is performed to refine the boundaries between classes. This step is optional. If it is not applied, differences are most likely to occur at boundaries between graph and text. The graph class tends to intrude the text class because it appears normal to the classifier that a graph region contains local text.

To refine boundaries between two classes, the classifier examines every pair of adjacent blocks labeled as different classes. After extracting a slice of pixels that lies in one block and is adjacent to the other block, the classifier compares statistics of the slice with those of the neighboring block. If this slice is shown to be more similar to the neighboring block, it is switched to the class of the neighboring block. This procedure is repeated until a slice is shown to be more similar to the block in which it lies.

To demonstrate the effect of boundary refinement, the classification result of an image after the first pass and the improved result after the boundary refinement are presented in Figure 6. The photograph part is shown in original, but the text part is marked by a unique intensity. Before the boundary refinement, a part of the text is classified as photograph because this part is grouped into blocks with the photograph. In this case, as the statistics of the photograph dominate those of the text, the classifier fails to detect those blocks containing mixed classes. As a result of the boundary refinement, the division of text and photograph is located more accurately.

E. Update Context Information

At the beginning of classification at resolution r, the context information matrix $CI(r)$ is inherited from $CI(r-1)$. Since a block at previous resolution $r-1$ is divided into four subblocks at resolution r, the statistics of the four subblocks stored in $CI(r)$ are duplicated from the statistics of the parent block provided that the parent block has been classified. There are two reasons for duplicating the statistics of the parent block: first, duplicating saves computation; second, more importantly, information is inherited from the crude resolution to avoid over-localization. Because of boundary refinement, the class of a subblock may have been changed from the class of the parent block; and a subblock may contain mixed classes. If either of the two cases occurs, the statistics stored in $CI(r)$ are recalculated for the subblock.

F. Context-based Classification

Suppose the current resolution is r and the context information is $CI(r)$ accordingly. The classifier scans the image and intends to classify all the undetermined blocks that are neighbors of blocks already classified. These blocks are classified based on both their features and the context information provided by their classified neighbors. If information is adequate to decide the class of a block, its statistics are added to $CI(r)$. Otherwise, the block remains marked as undetermined. After one scan, since $CI(r)$ is updated due to extra blocks classified, the classifier repeats the previous process to examine the remaining undetermined blocks. The iteration stops when the entire image is classified or no more blocks can be classified based on the current context $CI(r)$, which is equivalent to no new information having been added to $CI(r)$ in the scan. Then, the classifier proceeds to a higher resolution.

Before describing the context dependent classification algorithm, let us introduce two notations. First is the micro classifier $\mathbb{C}$, which classifies a block by incorporating features of the block and properties of its adjacent blocks. Adjacent blocks, in particular, refer to the four blocks above, below, to the right, and to the left of a block. Details of $\mathbb{C}$ are explained after the main algorithm. The second notation is the context list $\mathfrak{L}$. This list records the classes of adjacent blocks for all the unclassified blocks. Specifically, for every unclassified block, there is a corresponding element in $\mathfrak{L}$ that stores the classes of its four neighboring blocks. At each resolution

r, N is the total number of image blocks at this resolution. The algorithm of context dependent classification is presented below:

1. Let $0 \rightarrow m$, $1 \rightarrow j$.
2. If $j \leq N$, perform the following steps; otherwise, go to step 3.
   (a) If B_j is unclassified, perform the following steps; otherwise, go to step 2b.
      i. If B_j is adjacent to a classified block, perform the following steps; otherwise, go to step 2(a)ii.
         A. Use micro classifier $\mathbb{C}$ to classify B_j.
         B. If $\mathbb{C}$ can decide the class of B_j, store the characteristic statistics of B_j to the context information matrix $CI(r)$ and let $m + 1 \rightarrow m$.
         C. If $\mathbb{C}$ cannot decide the class of B_j, add the block B_j to the list $\mathfrak{L}$.
         D. Go to step 2b.
      ii. Add block B_j to the list $\mathfrak{L}$.
   (b) Let $j + 1 \rightarrow j$, go back to step 2.
3. If $\mathfrak{L}$ is not empty and $m > 0$, perform the following steps; otherwise, go to step 4.
   (a) Let $0 \rightarrow m$.
   (b) Update $\mathfrak{L}$:

   For every B_i in $\mathfrak{L}$, if B_i has newly classified adjacent blocks, i.e., blocks classified after the addition of B_i to $\mathfrak{L}$ or the latest updating of B_i in $\mathfrak{L}$, update B_i's record about the classes of its adjacent blocks.
   (c) If $\mathfrak{L}$ is modified after the update, i.e., there exists at least one block B_i in $\mathfrak{L}$ that has a new record about the classes of its adjacent blocks,
      i. set pointer $\mathbf{P}$ to the first element in $\mathfrak{L}$.
      ii. If $\mathbf{P}$ is NOT at the end of $\mathfrak{L}$, perform the following steps; otherwise, go back to step 3.
         A. Suppose the element pointed by $\mathbf{P}$ is the record of block B_i.
         B. If B_i has newly classified adjacent blocks,

         first, use micro classifier $\mathbb{C}$ to classify B_i;

         second, if the class of B_i is decided by $\mathbb{C}$, store the characteristic statistics of B_i to the context

information matrix $CI(r)$, delete B_i from list $\mathfrak{L}$, and let $m+1 \rightarrow m$.

C. Move $\mathbf{P}$ to the next element in $\mathfrak{L}$, go back to step 3(c)ii.

(d) Go back to step 3.

4. Stop.

Next we explain the details of the micro classifier $\mathbb{C}$, which classifies blocks using context information stored in $CI(r)$. The particular decision rule for each class is listed below:

1. If a block B is adjacent to a text block:

   if B is a bi-level intensity block and its two values are the same as those of the text block, it is classified as text; otherwise, no decision is made.

2. If a block B is adjacent to a graph block:

   if the feature L of B is close to that of the graph block and the mean value of B is also close to that of the graph block, B is classified as graph; otherwise, no decision is made.

3. If a block B is adjacent to a photograph block:

   if the mean value of B is close to that of the photograph block and its L is not very close to 1 (extreme value for text and graph), it is classified as photograph; otherwise, no decision is made.

One may wonder how to make a decision if a block is adjacent to several classes and is classified as different classes according to its neighbors. In particular, a block may be labeled as both graph and photograph, or both text and graph. Priorities are set for choosing a class if inconsistent labeling occurs. In our algorithm, both text and photograph have higher priority than graph. The reason for assigning higher priority to text is the strict requirement for a block being text. If the requirement is met, the probability of making a mistake is low. Photograph is assigned with higher priority than graph due to practical considerations. If a graph region is scanned or printed by a photograph standard, its quality is unlikely to be degraded, but not vice versa.

IV. Results

We applied the algorithm to segment 9 images with size 1650×1275 pixels. The classification error rates with respect to human labeling are listed in Table I. The average classification error rate is 4.1%. The

parameters used in the algorithm were chosen based on other document images downloaded from the World Wide Web. Simulations show that classification results of the tested images are not sensitive to the parameters. At the crudest resolution, the features of different classes are usually distributed so far apart that choosing parameters is not critical. In addition, the sensitivity to parameters is reduced by boundary refinement. At a higher resolution, context information compensates for vague features so that the sensitivity to parameters is retained low.

One classification example is shown in Figure 7. In this example, the starting block size of classification, i.e., the block size at resolution 0 is 64. The number of resolutions is 3. The classification error rate for this image is 1.19%. The photograph region is perfectly classified. In one misclassified region, text is labeled as graph. All the other misclassified regions are blank regions lying between text and graph. In practice, classifying a blank region as a part of graph or text is acceptable.

Figure 8 shows the step-by-step classification of a part of the image through the multiscale process. The classified image in the middle is the result at the first resolution. Part of the boundary between photograph and text is marked as undetermined class. At the higher resolution, after the undetermined region is divided into smaller blocks, the classifier labels the blocks correctly using context information.

It is evident that the classified images are very "clean" although no post-processing is applied. This results primarily from the multiscale structure of the algorithm. Since the classifier starts with considerably large block size, blocks classified at the crudest resolution form sufficiently large regions naturally. At higher resolutions, although blocks are smaller, the incorporation of context information tends to produce "clean" classification. Another advantage of the multiscale structure is the reduction of computation. A majority part of an image is usually classified at the crudest resolution. Although extra computation is needed to classify with context information at higher resolutions, the overall computational cost is low since very few blocks are left to be classified at the higher resolutions. The multiscale structure can thus be considered as an efficient mechanism to allocate computing resources. More computation is performed for more ambiguous regions. The CPU time to classify an image with size 1650×1275 pixels on a Pentium-300MHz PC with the Linux operating system is roughly 2 seconds.

We compare our results with Perlmutter's [13], [14]. We have not made comparisons with the gray-scale document image segmentation algorithms of [2], [27] because those algorithms target specific applications. In [2], [27], the graph class is not considered, and the quality of images is worse.

Perlmutter discussed two classification problems with document images. The first is the classification of photograph and non-photograph. The second is the classification of photograph, text, and others. The definition of photograph is the same as ours. The text class, however, differs from the text class that we have defined. In our case, as the intention is to classify document images in a globally meaningful way, local text is not always labeled as text. For example, if text occurs as a part in an artificial graph, table, or overlay, it is labeled as graph to form an integral part with the rest of the graph region. In Perlmutter's case, however, all the local text is labeled as text. Consequently, it is not proper to compare the numerical results directly. In [13], the classification result of the image shown in Figure 7 has considerable amount of misclassified photograph blocks. Although the misclassified photograph blocks can be cleaned up by a good post processing algorithm, the post processing takes extra time.

If the graph, text, and background classes are grouped as non-photograph, we can compare our results with those in [13]. The image shown in Figure 7 is perfectly classified by our algorithm in the sense of the two classes. In [13], many different sets of features are tested. When color information is used, the best set of features provides an error rate of 2.53% without post processing for the example image shown in Figure 7. The non-photograph class is classified with nearly zero error rate (under 0.5%) and the photograph class is classified with 13.46% error rate. If only the intensity information is used, as in our simulation, the result in [13] achieves a classification error rate of 13.25%. No improvement is obtained by post processing. Results for two additional images are also provided in [13]. Comparisons for these two images are omitted because overlays, which are regarded as graph here, are hand-labeled as photograph in [13].

V. Conclusions

We have defined two features that are capable of distinguishing local image types in document images. Instead of using moments of wavelet coefficients as features for classification, as is commonly done, we define

features according to the distribution patterns of wavelet coefficients. The first feature is defined as the goodness of match between the empirical distribution of wavelet coefficients in high frequency bands and the Laplacian distribution. The second feature is defined as a measure of how likely the wavelet coefficients in high frequency bands concentrate at a few discrete values. An algorithm was developed to calculate this feature efficiently.

A multiscale context dependent classification algorithm has been developed. The multiscale structure accumulates context information from low resolutions to high resolutions. At the starting resolution, classification is performed on large blocks to avoid over-localization. However, only blocks with extreme features are classified to ensure that blocks of mixed classes are left to be classified at higher resolutions. The unclassified blocks are divided into smaller blocks at the higher resolution. The smaller blocks are classified based on the context information achieved at the lower resolution. Simulations show that by incorporating context information, classification accuracy is significantly improved. The multiscale structure also provides a mechanism to save computation by applying more sophisticated rules to classify more ambiguous regions. The comparison with related work shows that our algorithm provides both lower classification error rates and better visual results.

Acknowledgments

The authors acknowledge the helpful suggestions of the reviewers.

Appendix

We describe the algorithm for finding the zone partition for a data range $[0, M]$ given an empirical distribution. This completes the definition of the classification feature L, the likelihood of being a highly discrete distribution. According to the second constraint of the definition for a zone, the two end points for a zone have to be local minima except for the two zones $[0, t_1]$ and $[t_{r-1}, M]$, that is, $t_1, \ldots, t_{r-1}$ are local minimum points. So the problem is reduced to finding all of the local minima of $h(t)$. On the other hand, not every local minimum point is an end point of a zone due to the third constraint. The value at an end point of a zone is required to be sufficiently small compared with the maximum value in the zone. Consequently, we have to extract the local maxima in the process of seeking local minima and guarantee that the global maximum

(one of the local maxima) in a zone is significantly greater than the values at the two end points. Thus, a partition is characterized by a sequence of interleaved local maximum and local minimum points, t_0^*, t_1, t_1^*, t_2, ..., t_{r-1}, t_{r-1}^*. Every local maximum point t_i^* is also a global maximum point in interval $[t_i, t_{i+1}]$.

In the following algorithm, we assume that $h(t)$ is a continuous function on $[0, M]$, so that there exists a local maximum between two local minima and there exists a local minimum between two local maxima. The continuity constraint is always acceptable in practice because we can only obtain estimated samples of $h(t)$, and the function $h(t)$ can be estimated by a continuous interpolation of the samples.

1. Find the first local maximum point starting from 0 and set t_0^* to it.
2. Set True $\rightarrow$ pre_is_maximum, and True $\rightarrow$ seek_minimum.
3. Set $t_0^* \rightarrow T$, $0 \rightarrow j$.
4. If $T < M$,

(a) If seek_minimum = True

i. Find $T < t < M$, so that $h(t)$ is a local minimum.

If such $h(t)$ does not exist, $M \rightarrow t_{j+1}$, $M \rightarrow T$, stop.

ii. If pre_is_maximum = True

{ if $h(t)/h(t_j^*) < \delta$, then set $t \rightarrow t_{j+1}$, $j+1 \rightarrow j$, False $\rightarrow$ pre_is_maximum. } ;

Else (i.e., pre_is_maximum = False)

{ if $h(t) < h(t_j)$, $t \rightarrow t_j$}

iii. Set False $\rightarrow$ seek_minimum, $t \rightarrow T$.

(b) If seek_minimum = False

i. Find $T < t < M$, so that $h(t)$ is a local maximum.

If such $h(t)$ does not exist, $M \rightarrow t_{j+1}$, $M \rightarrow T$, stop.

ii. If pre_is_maximum = True

{ if $h(t) > h(t_j^*)$, $t \rightarrow t_j^*$ } ;

Else (i.e., pre_is_maximum = False)

{ if $h(t_j)/h(t) < \delta$, then set $t \rightarrow t_j^*$, True $\rightarrow$ pre_is_maximum. }

iii. Set True $\rightarrow$ seek_minimum, $t \rightarrow T$.

5. If $T < M$, go back to 4;

else, stop.

The flag seek_minimum in the algorithm is alternated so that the sequence of all the interleaved local maxima and local minima, denoted by $\tilde{t^*}_0, \tilde{t}_1, \tilde{t^*}_1, \tilde{t}_2, \ldots, \tilde{t}_{r'-1}, \tilde{t^*}_{r'-1}$ can be found. Since this sequence may not satisfy all the constrains for a partition, we need to choose a subsequence that forms a partition. This is accomplished by effective control of the flag pre_is_maximum. The algorithm determines all the t_i and t_i^* in the order of minimum to maximum. The process, however, is not completely sequential in the sense that t_i, t_i^*, and t_{i+1}, which correspond to the end points, and maximum point of a zone, are adjusted simultaneously, instead of being determined one by one. Only when a zone $[t_i, t_{i+1}]$, with maximum point t_i^*, satisfying all the conditions is found, the left end point t_i and the maximum point t_i^* are fixed. The right end point t_{i+1} may still be changed when seeking for the next zone $[t_{i+1}, t_{i+2}]$. Nevertheless, the end point t_{i+1} is ensured to be replaced by a point with a lower probability density value if there is ever a replacement. This strategy guarantees the division between two zones to take place at a point with distribution density as low as possible.

To gain better understanding, consider a complete cycle of obtaining a zone $[t_i, t_{i+1}]$ given that the zone $[t_{i-1}, t_i]$ has been found. At the beginning, the flag pre_is_maximum is set as false and the algorithm is seeking a maximum point. According to the two flags seek_minimum and pre_is_maximum, the algorithm may be in four states. In Table II, we list the points that might be initially set or changed at the current state, and the next possible state to enter. The algorithm transits between states according to Table II. Except for determining the first zone $[0, t_1]$, which starts with both seek_minimum and pre_is_maximum being true, the cycles for the other zones always start with both seek_minimum and pre_is_maximum being false. The special case for $[0, t_1]$ happens because the left end point is tied at 0.

References

[1] P. J. Bickel and K. A. Doksum, *Mathematical Statistics: Basic Ideas and Selected Topics*, Prentice Hall, Englewood Cliffs, NJ, 1977.

[2] N. Chaddha, R. Sharma, A. Agrawal, and A. Gupta, "Text Segmentation in Mixed-mode Images," *Proceedings of Asilomar Conference on Signals, Systems and Computers.*, pp. 1356-1361, vol. 2, Nov. 1994.

[3] I. Daubechies, *Ten Lectures on Wavelets*, Capital City Press, 1992.

[4] A. P. Dhawan, Y. Chitre, C. Kaiser-Bonasso, and M. Moskowitz, "Analysis of Mammographic Microcalcifications Using Gray-level Image Structure Features," *IEEE Transactions on Medical Imaging,* vol. 15, no. 3, pp. 246-259, June 1996.

[5] A. M. El Sherbini, "A new feature vector for classification of binary images," *Proc. 11th IASTED Int. Conf. Applied Informatics*, pp. 330-333, 1995.

[6] J. L. Fisher, S. C. Hinds, and D. P. D'Amato, "A rule-based system for document image segmentation," *Proc. IEEE 10th Int. Conf. Pattern Recognition*, pp. 567-572, Atlantic City, NJ, June 1990.

[7] L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 910-918, Nov. 1988.

[8] M. H. Gross, R. Koch, L. Lippert, and A. Dreger, "Multiscale Image Texture Analysis in Wavelet Spaces," *Proceedings of 1st International Conference on Image Processing*, vol. 3, Nov. 1994.

[9] G. Loum, P. Provent, J. Lemoine, and E. Petit, "A New Method for Texture Classification Based on Wavelet Transform," *Proceedings of Third International Symposium on Time-Frequency and Time-Scale Analysis,* pp. 29-32, June 1996.

[10] S. G. Mallet, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 11, no. 7, July 1989.

[11] N. J. Nilsson, *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*, McGraw-Hill, NY, 1965.

[12] S. Ohuchi, K. Imao, and W. Yamada, "Segmentation method for documents containing text/picture (screened halftone, continuous tone)," *Transactions of the Institute of Electronics, Information and Communication Engineers D-II*, vol. J75D-II, no. 1, pp. 39-47, Jan. 1992.

[13] K. O. Perlmutter, "Compression and Classification of Images Using Vector Quantization and Decision Trees," *Ph.D thesis*, Stanford University, 1995.

[14] K. O. Perlmutter, N. Chaddha, J. B. Buckheit, R. M. Gray, and R. A. Olshen, "Text Segmentation in Mixed-mode Images Using Classification Trees and Transform Tree-structured Vector Quantization," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2231-2234, vol. 4, Atlanta, GA, May 1996.

[15] K. O. Perlmutter, S. M. Perlmutter, R. M. Gray, R. A. Olshen, and K. L. Oehler, "Bayes risk weighted vector quantization with posterior estimation for image compression and classification," *IEEE Trans. Image Processing*, vol. 5, no. 2, pp. 347-360, Feb. 1996.

[16] O. Rioul and M. Vetterli, "Wavelets and Signal Processing," *IEEE Signal Processing Magazine,* vol. 8, no. 4, pp. 14-38, Oct 1991.

[17] J. Sauvola and M. Pietikainen, "Page segmentation and classification using fast feature extraction and connectivity analysis," *Proc. 3rd Int. Conf. Document Analysis and Recognition*, Montreal, Que., Canada, Aug. 1995.

[18] F. Y. Shih and S. Chen, "Adaptive document block segmentation and classification," *IEEE Trans. Systems, Man and Cybernetics*, vol. 26, no. 5, pp. 797-802, Oct. 1996.

[19] G. W. Snedecor and W. G. Cochran, *Statistical Methods,* Iowa State University Press, Ames, Iowa, 1989.

[20] M. Tabb and N. Ahuja, "Multiscale Image Segmentation by Integrated Edge and Region Detection," *IEEE Transactions on Image Processing*, vol. 6, no. 5, pp. 642-655, May 1997.

[21] M. Unser, "Texture Classification and Segmentation Using Wavelet Frames," *IEEE Transactions on Image Processing,* vol. 4, no. 11, pp. 1549-1560, Nov. 1995.

[22] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, chapter 7, Prentice-Hall Inc., 1995.

[23] F. M. Wahl, K. Y. Wong, and R. G. Casey, "Block segmentation and text extraction in mixed text/image documents," *Computer Vision, Graphics, Image Processing*, vol. 20, pp. 375-390, 1982.

[24] J. Z. Wang, J. Li, G. Wiederhold, O. Firschein, "System for Screening Objectionable Images," *Computer Communications Journal*, to appear, Elsevier Science, 1998.

[25] D. Wang and S. N. Srihari, "Classification of newspaper image blocks using texture analysis," *Computer Vision, Graphics, Image Processing*, vol. 47, pp. 327-352, 1989.

[26] J. B. Weaver, D. M. Healy, H. Nagy, S. P. Poplack, J. Lu, T. Sauerland, and D. Langdon, "Classification of Masses in Digitized Mammograms with Features in the Wavelet Transform Domain," *Proceedings of the SPIE - Wavelet Applications,* vol. 2242, pp. 704-710, April 1994.

[27] P. S. Williams and M. D. Alder, "Generic texture analysis applied to newspaper segmentation," *Proc. Int. Conf. Neural Networks*, vol. 3, pp. 1664-1669, Washington, DC, June 1996.

[28] K. Y. Wong, R. G. Casey, and F. M. Wahl, "Document analysis system," *IBM J. Res. Develp.*, vol. 6, pp. 642-656, Nov. 1982.

Image ID	1	2	3	4	5	6	7	8	9
P_e	5.65%	0.000%	0.108%	3.64%	11.0%	1.19%	0.841%	14.5%	0.000%

TABLE I

RATIOS OF CLASSIFICATION ERRORS WITH RESPECT TO HAND-LABELING FOR 9 SAMPLE IMAGES

Current State (S, P)	Points to be adjusted	Possible Next State (S, P)
(F, F)	t_i^*	(T, F), (T, T)
(T, F)	t_i	(F, F)
(F, T)	t_i^*	(T, T)
(T, T)	t_{i+1}	(F, T), (F, F)

TABLE II

TRANSITIONS IN THE 'ZONE' DIVISION ALGORITHM AND THE POINTS ADJUSTED AT EVERY STATE. S: SEEK_MINIMUM, P: PRE_IS_MAXIMUM, T: TRUE, F: FALSE.

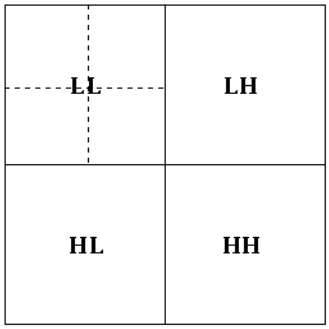
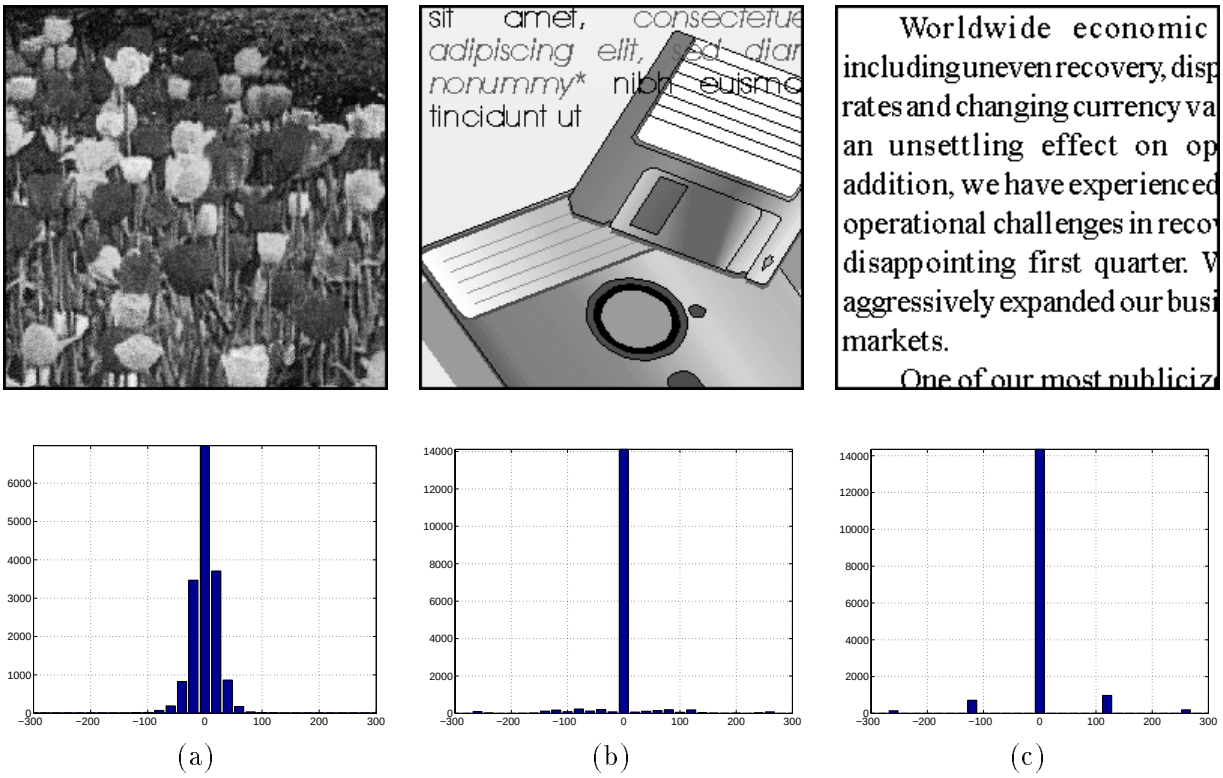


Fig. 1. The decomposition of an image into four frequency bands by wavelet transforms



(a) (b) (c)

Fig. 2. The histograms of Haar wavelet coefficients in the LH band: (a) photograph image, (b) graph image, (c) text image. First row: original images. Second row: histograms.

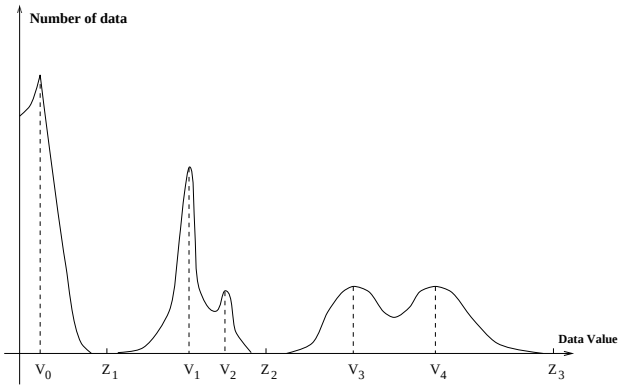
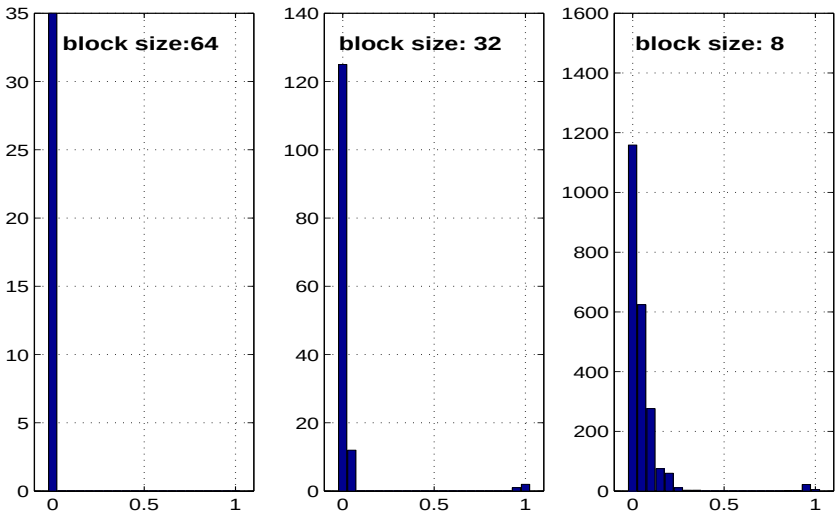
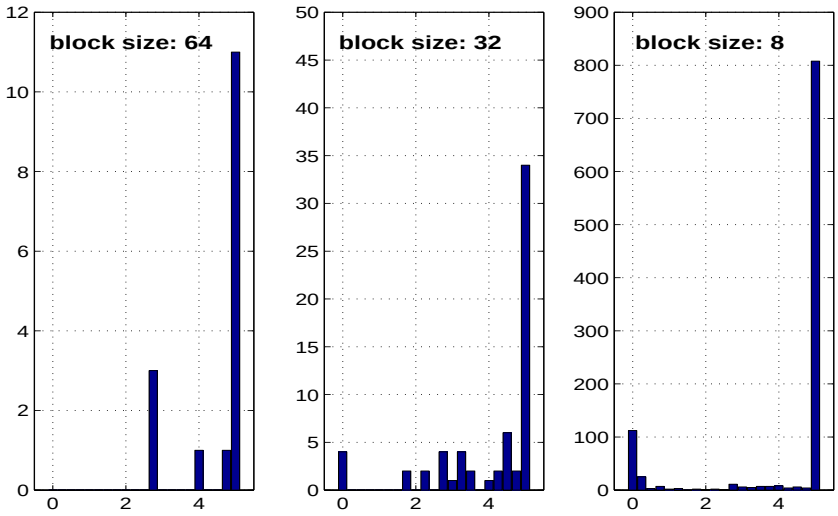


Fig. 3. The concentration values of a histogram

(a) (b)



(c)



(d)

Fig. 4. Features at multiple resolutions: (a) a photograph image, (b) a graph image, (c) histograms of L at multiple resolutions for the photograph image, (d) histograms of $\bar{\chi}^2$ at multiple resolutions for the graph image.

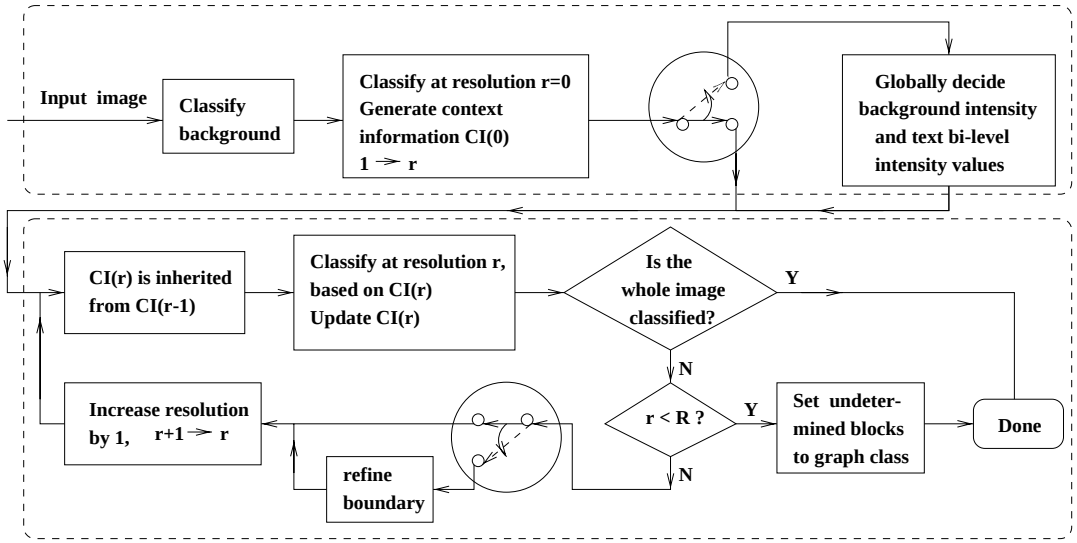
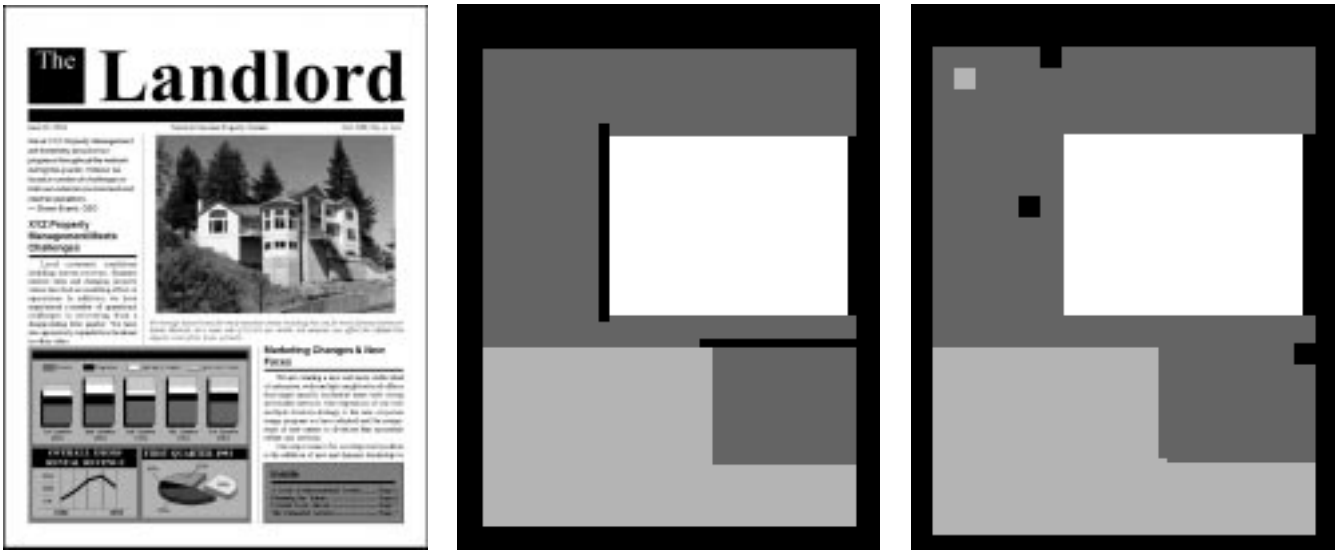


Fig. 5. The flow chart of the classification algorithm

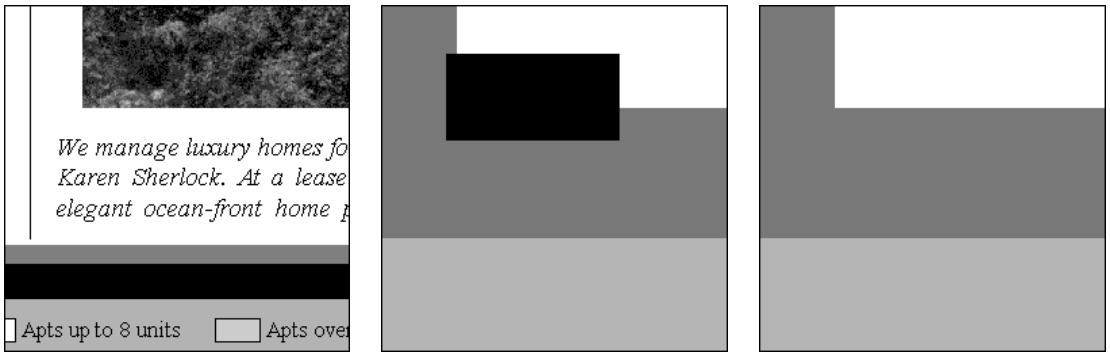
(a) (b) (c)

Fig. 6. An example of boundary refinement: (a) original image, (b) result after the first pass classification, (c) result after the boundary refinement. The text region is marked by a unique intensity; and the photograph region is shown in original.



(a) (b) (c)

Fig. 7. One sample image and its classification results: (a) original image, (b) hand-labeled classification, (c) classified image. White: photograph, light gray: graph, dark gray: text, black: background.



(a) (b) (c)

Fig. 8. One part of the sample image and its classification results at two resolutions: (a) original image, (b) the result at the first resolution, (c) the result at the second resolution. White: photograph, light gray: graph, dark gray: text, black: undetermined.