# A Study on the Document Zone Content Classification Problem

Yalin Wang[1], Ihsin T. Phillips[2], and Robert M. Haralick[3]

[1] Dept. of Elect. Eng. Univ. of Washington
Seattle, WA 98195, US
ylwang@u.washington.edu

[2] Dept. of Comp. Science, Queens College, City Univ. of New York
Flushing, NY 11367, US
yun@image.cs.qc.edu

[3] The Graduate School, City Univ. Of New York
New York, NY 10016, US
haralick@gc.cuny.edu

**Abstract.** A document can be divided into zones on the basis of its content. For example, a zone can be either text or non-text. Given the segmented document zones, correctly determining the zone content type is very important for the subsequent processes within any document image understanding system. This paper describes an algorithm for the determination of zone type of a given zone within an input document image. In our zone classification algorithm, zones are represented as feature vectors. Each feature vector consists of a set of $25$ measurements of pre-defined properties. A probabilistic model, decision tree, is used to classify each zone on the basis of its feature vector. Two methods are used to optimize the decision tree classifier to eliminate the data over-fitting problem. To enrich our probabilistic model, we incorporate context constraints for certain zones within their neighboring zones. We also model zone class context constraints as a Hidden Markov Model and used Viterbi algorithm to obtain optimal classification results. The training, pruning and testing data set for the algorithm include $1,600$ images drawn from the UWCDROM-III document image database. With a total of $24,177$ zones within the data set, the cross-validation method was used in the performance evaluation of the classifier. The classifier is able to classify each given scientific and technical document zone into one of the nine classes, $2$ text classes (of font size $4-18$pt and font size $19-32$ pt), math, table, halftone, map/drawing, ruling, logo, and others. A zone content classification performance evaluation protocol is proposed. Using this protocol, our algorithm accuracy is $98.45\%$ with a mean false alarm rate of $0.50\%$.

## 1 Introduction

A document is varied in content. It can contain numerous zones that may have text, math, figure zones, etc. Each of these zones has its own characteristic features. For example, a math zone may contain symbols like $=$, $+$, $\sum$, $\int$, $\cdots$, which a text zone may not contain. On the other hand, figure zones may not contain any symbols or text. Captions and pure text vary in font size and style. This paper describes an algorithm for the determination of zone type of a given zone within a give document image. In the design of a zone classifier, a set of measurements of pre-defined properties of a given zone forms a feature vector. The features include mean of the run length mean and variance, spatial mean and variance, etc. A probabilistic model is used to classify each zone on the basis of its feature vector [1]. We employ a decision tree classifier in the classification process. Two methods are used to optimize the decision tree classifier to eliminate the data over-fitting problem. Furthermore, to enrich our probabilistic model, we incorporate context constraints within neighboring zones for some zones and we model zone class context constraints as a Hidden Markov Model and used Viterbi algorithm [2] to obtain optimal classification results.

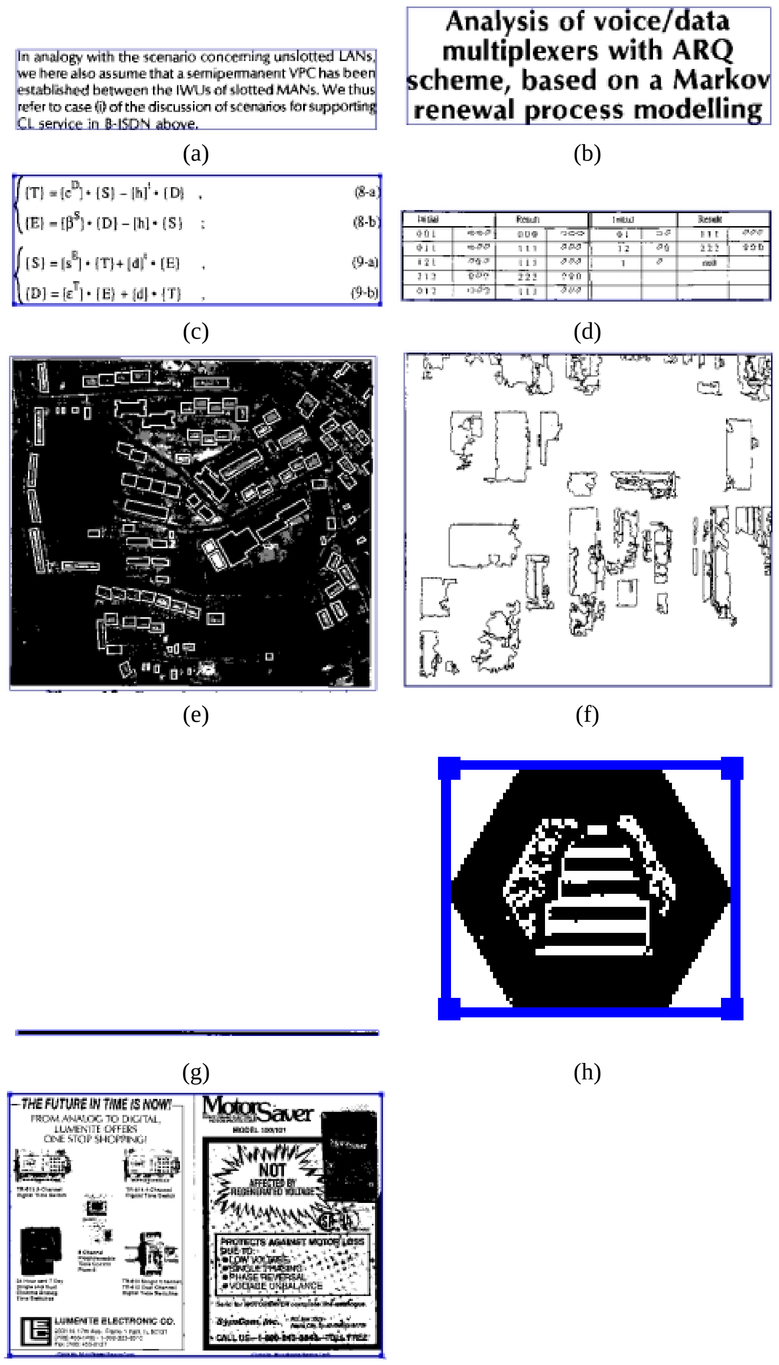
Our earlier work, also using feature vectors, is described in Wang et. al [3] and Wang et. al [4]. There we used $69$ features. Here we achieve better performance with $25$ features. Liang et. al [5] developed a feature based zone classifier using only the knowledge of the widths and the heights of the connected components within a given zone. Chetverikov *et al.* [6] studied zone content classification using their general tool for texture analysis and document blocks are labeled as text or non-text using texture features derived from a feature based interaction map (FBIM). Le et. al [7] proposed an automated labeling of zones from scanned images with labels such as titles, authors, affiliations and abstracts. Their labeling is based on features calculated from optical character recognition (OCR) output, neural network models, machine learning methods, and a set of rules that is derived from an analysis of the page layout for each journal and from generic typesetting knowledge for English text.

We propose a set of performance evaluation criteria to evaluate the performance of our algorithm and a set of experiments were conducted. In the experiment, each zone is specified by a unique zone identification number, a rectangular box which encloses the zone and is represented by the coordinates of the leftmost-top and rightmost-bottom points, and the zone type. The zones are the zone ground-truth entities from UWCDROM-III document image database [8]. The database includes $1,600$ scientific and technical document images with a total of $24,177$ zones. The zone classes we consider are text with font size $\leq 18$pt, text with font size $\geq 19$pt, math, table, halftone, map/drawing, ruling, logo, and others. The examples of each class are shown in Figure1. Our algorithm accuracy rate is $98.45\%$ and the mean false alarm rate is $0.50\%$.

The remaining of this paper is divided into $5$ parts. In Section 2, we present the detail description of the feature vector used in the algorithm. In Section 3, we give a brief description of the classification procedure. The performance evaluation protocol and experimental results are reported in Section 4. The feature reduction analysis is presented in Section 5. Our conclusion and statement of future work are discussed in Section 6.

## 2 Features for Zone Content Classification

Every zone in the document is a rectangular area. Black pixels are assumed to be foreground and the white pixels are background. For each zone, run length and spatial features are computed for each line along two different canonical directions: horizontal, diagonal. These two directions are shown in Figure 2. In the notations, we use subscript



(a)

In analogy with the scenario concerning unslotted LANs, we here also assume that a semipermanent VPC has been established between the IWUs of slotted MANs. We thus refer to case (i) of the discussion of scenarios for supporting CL service in B-ISDN above.

(b)

Analysis of voice/data multiplexers with ARQ scheme, based on a Markov renewal process modelling

(c)

$$\begin{cases} \{T\} = [c^D] \cdot \{S\} - [h]^t \cdot \{D\} , & (8\text{-a}) \\ \{E\} = [\beta^S] \cdot \{D\} - [h] \cdot \{S\} ; & (8\text{-b}) \end{cases}$$

$$\begin{cases} \{S\} = [s^E] \cdot \{T\} + [d]^t \cdot \{E\} , & (9\text{-a}) \\ \{D\} = [\varepsilon^T] \cdot \{E\} + [d] \cdot \{T\} , & (9\text{-b}) \end{cases}$$

(d)

| Initial | | Result | | Initial | | Result | |
|---|---|---|---|---|---|---|---|
| 0 0 1 | [illegible] | 0 0 0 | [illegible] | 0 1 | [illegible] | 1 1 1 | [illegible] |
| 0 1 1 | [illegible] | 1 1 1 | [illegible] | 1 2 | [illegible] | 2 2 2 | [illegible] |
| 1 2 1 | [illegible] | 1 1 3 | [illegible] | 1 | [illegible] | null | |
| 2 1 2 | [illegible] | 2 2 2 | [illegible] | | | | |
| 0 1 2 | [illegible] | 1 1 3 | [illegible] | | | | |

(e)

(f)

(g)

(h)

THE FUTURE IN TIME IS NOW!
FROM ANALOG TO DIGITAL, LUMENITE OFFERS ONE STOP SHOPPING!

LUMENITE ELECTRONIC CO.

MotorSaver
MODEL 100/101

NOT AFFECTED BY REGENERATED VOLTAGE

PROTECTS AGAINST MOTOR LOSS DUE TO:
- LOW VOLTAGE
- SINGLE PHASING
- PHASE REVERSAL
- VOLTAGE UNBALANCE

SymCom, Inc.

CALL US—1-800-843-8848—TOLL FREE

(i)

**Fig. 1.** Illustrates examples of nine zone content classes.(a) Text 1 class; (b)Text 2 class; (c) Math class; (d) Table class; (e) Halftone class; (f) Map/drawing class; (g) Ruling class; (h) Logo class; (i) Others class.

$h$ and $d$ to represent two directions. When to discriminate foreground and background features is necessary, we use superscript $0$ and $1$ to represent foreground and background features, respectively. For example, $rlmean_h^0$ represents background run length mean feature computed in horizontal direction. A total of 25 features are computed for each zone. In the following, we describe each feature in detail.

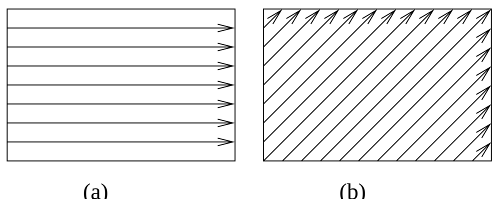


**Fig. 2.** Illustrates the two directions in which we compute run length and spatial features.(a) horizontal; (b) diagonal.

## 2.1 Run Length Features

A *run length* is a list of contiguous foreground or background pixels in a given direction. A total of 10 run length features are used, they include foreground/background run length mean and variance in each of the two directions.

Let $\mathcal{RL}_h^1$ and $\mathcal{RL}_d^1$ denote the foreground run length sets on the two directions. $|\mathcal{RL}_h^1|$, and $|\mathcal{RL}_d^1|$ constitute the first $2$ features.

The next four features include foreground and background run length mean features on two directions in a given zone. Denote them as $rlmean_h^0$, $rlmean_d^0$, $rlmean_h^1$ and $rlmean_d^1$.

$$rlmean_h^0 = \frac{1}{|\mathcal{RL}_h^0|} \sum_{rl \in \mathcal{RL}_h^0} rl, rlmean_d^0 = \frac{1}{|\mathcal{RL}_d^0|} \sum_{rl \in \mathcal{RL}_d^0} rl$$
$$rlmean_h^1 = \frac{1}{|\mathcal{RL}_h^0|} \sum_{rl \in \mathcal{RL}_h^0} rl, rlmean_d^1 = \frac{1}{|\mathcal{RL}_d^0|} \sum_{rl \in \mathcal{RL}_d^0} rl$$

The next four features are foreground and background run length variance features on the two directions in a given zone. Denote them as $rlvar_h^0$, $rlvar_d^0$, $rlvar_h^1$ and $rlvar_d^1$. They can be obtained by calculating the mean of the squares of all the run lengths in the zone and subtracting them by the square of the run length mean. Specifically, they can be computed by the equations below.

$$rlvar_h^0 = \frac{\sum_{rl \in \mathcal{RL}_h^0} rl^2}{|\mathcal{RL}_h^0|} - (rlmean_h^0)^2$$
$$rlvar_d^0 = \frac{\sum_{rl \in \mathcal{RL}_d^0} rl^2}{|\mathcal{RL}_d^0|} - (rlmean_d^0)^2$$
$$rlvar_h^1 = \frac{\sum_{rl \in \mathcal{RL}_h^1} rl^2}{|\mathcal{RL}_h^1|} - (rlmean_h^1)^2$$
$$rlvar_d^1 = \frac{\sum_{rl \in \mathcal{RL}_d^1} rl^2}{|\mathcal{RL}_d^1|} - (rlmean_d^1)^2$$

## 2.2 Spatial Features

Four spatial features are designed to capture the foreground pixel distribution information. We denote the foreground pixel set in a given zone as $\mathcal{F}$. Spatial mean, $\mu$, and spatial variance, $\delta$, can be defined as

$$\mu = \frac{1}{|\mathcal{F}|} \sum_{p \in \mathcal{F}} w_p \qquad \delta = \frac{1}{|\mathcal{F}|} \sum_{p \in \mathcal{F}} (w_p - \mu)^2$$

where $w_p$ is a weight assigned to each foreground pixel $p$. With two directions, we obtain four features.

As shown in Figure 2, we have two different directions to compute the run lengths. In each direction, we start computing from a point on a zone border and continue at a given direction until we hit another zone border again. We call such a computation route as a *run segment*. For every run segment the sum of foreground run lengths gives the *run segment projection*. Given a direction, each foreground pixel belongs and only belongs to one run segment. We associate each foreground pixel with a weight of run segment projection. We let the foreground pixels in the same run segment have the same weights so we have two different weight definitions according to each direction. We denote the starting and ending pixel coordinates of a horizontal run segment as $(x_{h,1}, y_{h,1})$, $(x_{h,2}, y_{h,2})$. We denote by $(x_{d,1}, y_{d,1})$, $(x_{d,1}, y_{d,1})$ the starting and ending pixel coordinates of a diagonal run segment.

The weights for horizontal and diagonal directions are denoted as $w_h$ and $w_d$, $w_h = y_{h,1}$ and $w_d = y_{d,2} - x_{d,2}$.

Denote the set of run segments in two directions as $\mathcal{L}_h$ and $\mathcal{L}_d$. For a run segment, say, $l_h$, we denote its horizontal run segment projection on it as $proj_{h,l}$. In our algorithm, we compute spatial means and spatial variances as follows.

$$spmean_h = \frac{1}{|\mathcal{F}|} \sum_{l \in \mathcal{L}_h} w_h \times proj_{h,l}$$

$$spmean_d = \frac{1}{|\mathcal{F}|} \sum_{l \in \mathcal{L}_d} w_d \times proj_{d,l}$$

$$spvar_h = \frac{1}{|\mathcal{F}|} \sum_{l \in \mathcal{L}_h} [proj_{h,l} \times (w_l - spmean_h)^2]$$

$$spvar_d = \frac{1}{|\mathcal{F}|} \sum_{l \in \mathcal{L}_d} [proj_{d,l} \times (w_d - spmean_d)^2]$$

### 2.3 Autocorrelation Features

For each run segment, we define four functions: run segment projection, number of foreground run lengths, run length mean and spatial mean. We get $8$ features by computing their autocorrelation functions using Fourier transform.

Denote the set of run length in a horizontal and a diagonal run segment as $\mathcal{RL}_{h,l}$ and $\mathcal{RL}_{d,l}$. Run segment projection function has been defined earlier, which are $proj_{h,l}$ and $proj_{d,l}$. The function of the number of foreground runs on each run segment are straightforward. The function of run length mean on each run segment can be defined as follows.

$$rlmean_{h,l} = \frac{proj_{h,l}}{|\mathcal{RL}_{h,l}|}, \qquad rlmean_{d,l} = \frac{proj_{d,l}}{|\mathcal{RL}_{d,l}|}.$$

Let $(x_{h,s}, y_{h,s})$, $(x_{h,e}, y_{h,e})$ be the two end points of a horizontal run length, and $(x_{d,s}, y_{d,s})$, $(x_{d,e}, y_{d,e})$ be the two end points of the diagonal run length. The definition of $pos$ and $leng$ functions are given as

$$\begin{aligned} pos_{h,rl} &= x_{h,s}, \qquad & leng_{h,rl} &= x_{h,e} - x_{h,s} \\ pos_{d,rl} &= x_{d,s}, \qquad & leng_{d,rl} &= x_{d,e} - x_{d,s} \end{aligned} \tag{1}$$

The spatial mean function for each line can be defined as follows.

$$\begin{aligned} spmean_{h,l} &= \frac{1}{proj_{h,rl}} \Big( \sum_{rl \in \mathcal{RL}_{h,l}} pos_{h,rl} \times leng_{h,rl} + \\ &\quad \frac{1}{2} \Big( \sum_{rl \in \mathcal{RL}_{h,l}} (leng_{h,rl})^2 - proj_{h,rl} \Big) \Big) \\ spmean_{r,l} &= \frac{1}{proj_{d,rl}} \Big( \sum_{rl \in \mathcal{RL}_{d,l}} pos_{d,rl} \times leng_{d,rl} + \\ &\quad \frac{1}{2} \Big( \sum_{rl \in \mathcal{RL}_{d,l}} (leng_{d,rl})^2 - proj_{d,rl} \Big) \Big) \end{aligned}$$

After we compute one function on each run segment, we can get a sequence of values, indexed by the run segment number. Using the Fast Fourier Transform [9], we can get the autocorrelation functions value for every function. Each feature is the slope of the tangent to the autocorrelation function values whose indexes are close to $0$. We used general linear least squares method [9] to compute the slope of the points near $0$.

### 2.4 Background Features

Although some background analysis techniques can be found in the literature([10],[11]), none of them, to our knowledge, has extensively studied the statistical characteristics of their background structure. Our signature-like background features are designed to give us more information on the distributions of the big foreground chunks in a given zone. We define a *large horizontal blank block* and a *large vertical blank block* as in [4]. The background feature is the total area of large horizontal and large vertical blank blocks, $A$.

### 2.5 Text Glyph Feature

A glyph is a connected component in a given zone. A text glyph is a character candidate glyph. Most of zones have some text glyphs. The information of how many text glyphs a given zone has is also an useful feature. The number of text glyphs in this zone, $W$, normalized by the zone area is the text glyph feature.

The so-called text glyphs are not from any OCR output. They are outputs of a statistical glyph filter. The inputs of this filter are the glyphs after finding connected component operation. The statistical glyph filter classifies each connected component into one of two classes: text glyph and non-text glyph. The filter uses a statistical method to classify glyphs and was extensively trained on UWCDROM-III document image database.

### 2.6 Column Width Ratio Feature

It is a common observation that math zones and figure zones have a smaller width compared to text zones. For any zone, the quotient of the zone width to the width of its column is calculated as $\frac{C}{Width_{column}}$, where $C$ is the zone width and $Width_{column}$ is the width of the text column in which the zone is.

## 3 Classification Process

A decision tree classifier makes the assignment through a hierarchical, tree-like decision procedure. For the construction of a decision tree [1], we need a training set of feature vectors with true class labels. At each node, the discriminant function splits the training subset into two subsets and generates child nodes. A discriminant threshold is chosen at each node such that it minimizes an impurity value of the distribution mode at that node. The process is repeated at each newly generated child node until a stopping condition is satisfied and the node is declared as a leaf node on a majority vote.

In building a decision tree classifier, there is a risk of memorizing the training data, in the sense that nodes near the bottom of the tree represent the noise in the sample, As mentioned in [12], some methods were employed to make better classification. We used two methods [4] to eliminate data over-fitting in decision tree classifier.

To further improve the zone classification result, we want to make use of context constraint in some zone set. We model context constraint as a Markov Chain and use the Viterbi algorithm( [2]) to find the most likely state sequence. To apply the Viterbi algorithm( [2]), we have to know the probability that each zone belongs to each class. This probability is readily estimated from the training data set by decision tree structure. The details can be found in [4].

## 4 Experiments and Results

A hold-out method is used for the error estimation in our experiment. We divided the data set into $9$ parts. We trained the decision tree on the first $4$ parts, pruned the tree using another $4$ parts, and then tested on the last $1$ part. To train the Markov model, we trained on the first $8$ parts and tested it on the last $1$ part. Continue this procedure, each time omitting one part from the training data and then testing on the omitted one part from the training data and testing on the omitted part. Then the combined $9$ part results are put together to estimate the total error rate [1].

**Table 1.** Possible true- and detected-state combination for two classes

| True Class | Assigned Class | |
|---|---|---|
| | a | b |
| a | $P_{aa}$ | $P_{ab}$ |
| b | $P_{ba}$ | $P_{bb}$ |

The output of the decision tree is compared with the zone labels from the ground truth in order to evaluate the performance of the algorithm. A contingency table is computed to indicate the number of zones of a particular class label that are identified as members of one of the nine classes. The rows of the contingency table represent the true classes and the columns represent the assigned classes. We compute four rates here: *Correct Recognition Rate (CR)*, *Mis-recognition Rate (MR)*, *False Alarm Rate (FR)*, *Accuracy Rate (AR)*. Suppose we only have two classes: a and b. The possible true- and detected-state combination is shown in Table 1. We compute the four rates for class a as follows:

$$CR = \frac{P_{aa}}{P_{aa} + P_{ab}}, MR = \frac{P_{ab}}{P_{aa} + P_{ab}}$$
$$FR = \frac{P_{ba}}{P_{ba} + P_{bb}}, AR = \frac{P_{aa} + P_{bb}}{P_{aa} + P_{ab} + P_{bb} + P_{ba}}$$

In our experiment, the training and testing data set was drawn from the scientific document pages in the University of Washington document image database III [8]. It has $1,600$ scientific and technical document pages with a total of $24,177$ zones. The class labels for each of the zones are obtained from the database. These zones belonged to nine different classes. For a total of $24,177$ zones, the accuracy rate was $98.43\%$ and mean false alarm rate was $0.50\%$, as shown in Table 2.

In Figure 3, we show some failed cases of our experiment. Figure 3(a) is a Table zone misclassified as Math zone due to the presence of many numerals and operators.

**Table 2.** Contingency table showing the number of zones of a particular class that are assigned as members of each possible zone class in UWCDROM-III. In the table, $T_1$, $T_2$, $M$, $T$, $H$, $MD$, $R$, $L$, $O$ represent text with font size $\leq 18$pt., text with font size $\geq 19$pt., math, table, halftone, map/drawing zone, ruling, logo, others, respectively. The rows represent the ground truth numbers while the columns represent the detection numbers.

| | T1 | T2 | M | T | H | M/D | R | L | O | CR | MR |
|---|---|---|---|---|---|---|---|---|---|---|---|
| T1 | 21426 | 23 | 40 | 7 | 1 | 7 | 1 | 3 | 3 | 99.60% | 0.40% |
| T2 | 19 | 104 | 1 | 0 | 1 | 2 | 0 | 0 | 1 | 81.25% | 18.75% |
| M | 47 | 1 | 686 | 2 | 0 | 18 | 1 | 1 | 2 | 90.50% | 9.50% |
| T | 6 | 0 | 4 | 162 | 0 | 35 | 0 | 1 | 2 | 77.14% | 22.86% |
| H | 1 | 0 | 1 | 1 | 345 | 27 | 0 | 0 | 0 | 92.00% | 8.00% |
| M/D | 2 | 3 | 20 | 20 | 28 | 648 | 1 | 1 | 5 | 89.01% | 10.99% |
| R | 3 | 0 | 2 | 0 | 0 | 2 | 424 | 0 | 1 | 98.15% | 1.85% |
| L | 7 | 3 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 15.38% | 84.62% |
| O | 4 | 0 | 2 | 0 | 2 | 7 | 1 | 0 | 6 | 27.27% | 72.73% |
| FR | 3.34% | 0.12% | 0.30% | 0.13% | 0.13% | 0.42% | 0.02% | 0.02% | 0.06% | | |

Figure 3(b) is a Map/Drawing zone misclassified as Table zone in that the content of the figure is just a table. Figure 3(c) shows a most frequent error of our current system. Our system classified a Math zone into Text 1 zone class. Sometimes our system still lacks a good ability to detect such a single line math equation zone which, even worse sometimes, includes some description words. Figure 3(d) shows an error example in which a Math zone was misclassified as a table zone because of its sparse nature.

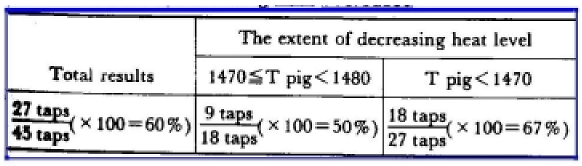
## 5 Feature Reduction Analysis

In our early work [3] [4], we used a feature vector consisting of $69$ features and got very good results. In our recent work, we tried to reduce the unnecessary features from the feature vector while keeping a good performance. By analysis and experiments, a total of $44$ features were eliminated.

As shown in Figure 4, there were four feature computation directions. Since the images in UWCDROM-III are all de-skewed already, there do not exist strong variations in different directions. We changed the four directions to the current two directions 2. It directly removed $32$ features from the feature vector.

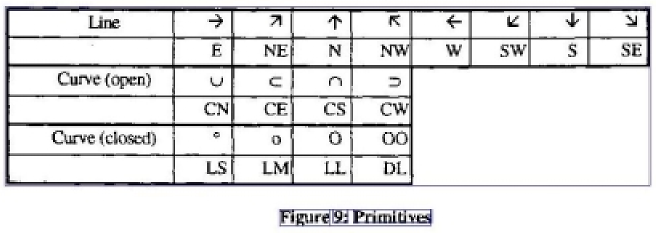
Some features are redundant. For example, there were four background features, background run length number in the two given directions, a fraction of black pixels to the total number of pixels and total area of large horizontal and large vertical blank blocks. Since the feature, total area of large horizontal and large vertical blank blocks, is computed using the other three feature information, we eliminated the other three features. There were two zone area related features, zone bounding box area and a fraction of the number of text glyphs to the zone bounding box area. There are dependent features so we eliminated the first of them.

There were $16$ features computed by autocorrelation function. We defined four functions which are computed in two different directions. The features are the slope of the



| Total results | The extent of decreasing heat level | |
|---|---|---|
| | 1470≦T pig<1480 | T pig<1470 |
| $\frac{27\ \text{taps}}{45\ \text{taps}}(\times 100=60\%)$ | $\frac{9\ \text{taps}}{18\ \text{taps}}(\times 100=50\%)$ | $\frac{18\ \text{taps}}{27\ \text{taps}}(\times 100=67\%)$ |

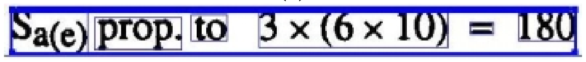
(a)



| Line | → | ↗ | ↑ | ↖ | ← | ↙ | ↓ | ↘ |
|---|---|---|---|---|---|---|---|---|
| | E | NE | N | NW | W | SW | S | SE |
| Curve (open) | ∪ | ⊂ | ∩ | ⊃ | | | | |
| | CN | CE | CS | CW | | | | |
| Curve (closed) | ° | o | O | OO | | | | |
| | LS | LM | LL | DL | | | | |

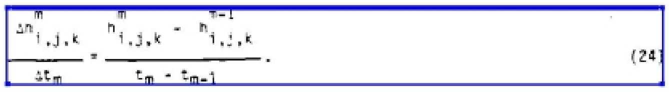
Figure 9: Primitives

(b)



$S_{a(e)}$ prop. to $3 \times (6 \times 10) = 180$

(c)



$$\frac{\Delta n^{m}_{i,j,k}}{\Delta t_m} = \frac{h^{m}_{i,j,k} - h^{m-1}_{i,j,k}}{t_m - t_{m-1}}. \qquad (24)$$

(d)

**Fig. 3.** Illustrates some failed examples. (a). Table zone misclassified as Math zone; (b). Map/drawing zone misclassified as Table zone; (c). Math zone misclassified as Text 1 zone; (d). Math zone misclassified as Table zone.

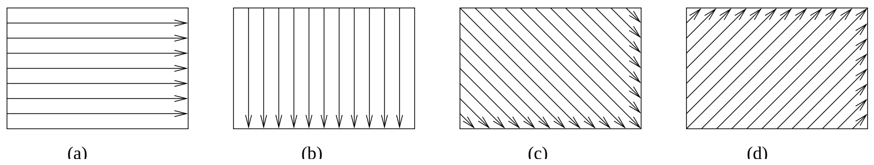


**Fig. 4.** Illustrates the four directions in which we computed run length and spatial features in our earlier work [3][4]. (a) horizontal; (b) vertical; (c) left-diagonal; (d) right-diagonal.

tangent to the autocorrelation function values whose indexes are close $0$ and, the index for which the autocorrelation function goes to $10\%$ of its maximum value. By experimenting, we eliminated $8$ of them. From the experimental results, we believe our feature reduction was successful.

## 6 Conclusion

Given the segmented document zones, correctly determining the zone content type is very important for the subsequent processes within any document image understanding system. This paper describes an algorithm for the determination of zone type of a given zone within an input document image. In our zone classification algorithm, zones are represented as feature vectors. Each feature vector consists of a set of $25$ measurements of pre-defined properties. A probabilistic model, decision tree, is used to classify each zone on the basis of its feature vector [1]. Two methods are used to optimize the decision tree classifier to eliminate the data over-fitting problem. To enrich our probabilistic model, we incorporate context constraints for certain zones within their neighboring zones. We also model zone class context constraints as a Hidden Markov Model and used Viterbi algorithm [2] to obtain optimal classification results.

To compare the performance of this algorithm with our two previous algorithms [3][4], in term of the accuracy rate and the false alarm rate, the identical data set was used in the experiment. The data set consists of $1,600$ UWCDROM-III images with a total of $24,177$ zones. The cross-validation method was used in the performance evaluation of the three algorithms. Table 3 shows the result. The accuracy rate and the false alarm rate are similar for the current and the last algorithms. However, since the features used in the current algorithm was reduced from $69$ features to $25$, the classification speed of the current algorithm was reduced proportionally.

**Table 3.** Illustrates the performance evaluation results of three papers.

| Paper | Accuracy Rate | False Alarm Rate |
|---|---|---|
| [3] | $97.53\%$ | $1.26\%$ |
| [4] | $98.52\%$ | $0.53\%$ |
| This Paper | $98.53\%$ | $0.50\%$ |

A few failed cases (Figure 3) are reported in this paper. As of our observation, many errors are due to the difficult discrimination between single line math and text 1 class. Our future work include the development of math zone identification technique, modeling zone content dependency feature in a more general zone set.

## References

1. R. Haralick and L. Shapiro. *Computer and Robot Vision*, volume 1. Addison Wesley, 1997.
2. L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–285, February 1989.
3. Y. Wang, R. Haralick, and I. T. Phillips. Improvement of zone content classification by using background analysis. In *Fourth IAPR International Workshop on Document Analysis Systems. (DAS2000)*, Rio de Janeiro, Brazil, December 2000.
4. Y. Wang, R. Haralick, and I. T. Phillips. Zone content classification and its performance evaluation. In *Sixth International Conference on Document Analysis and Recognition(ICDAR01)*, pages 540–544, Seattle, WA, September 2001.
5. J. Liang, R. Haralick, and I. T. Phillips. Document zone classification using sizes of connected components. *Document Recognition III, SPIE'96*, pages 150–157, 1996.
6. D. Chetverikov, J. Liang, J. Komuves, and R. Haralick. Zone classification using texture features. In *Proc. International Conference on Pattern Recognition*, pages 676–680, Vienna, 1996.

7. D. X. Le, J. Kim, G. Pearson, and G. R. Thom. Automated labeling of zones from scanned documents. *Proceedings SDIUT99*, pages 219–226, 1999.
8. I. Phillips. Users' reference manual. *CD-ROM, UW-III Document Image Database-III*, 1995.
9. W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.
10. A. Antonacopoulos. Page segmentation using the description of the background. *Computer Vision and Image Understanding*, pages 350–369, June 1998.
11. H. S. Baird. Background structure in document images. *Document Image Analysis*, pages 17–34, 1994.
12. W. Buntine. Learning classification trees. *Statistics and Computing journal*, pages 63–76, 1992.