Machine-Printed Character Recognition 10

Huaigu Cao and Prem Natarajan

Contents

H. Cao
Raytheon BBN Technologies, Cambridge, MA, USA
e-mail: hcao@bbn.com

P. Natarajan
Information Sciences Institute, University of Southern California, Marina Del Rey, CA, USA
e-mail: pnataraj@isi.edu

D. Doermann, K. Tombre (eds.), *Handbook of Document Image Processing and Recognition*, DOI 10.1007/978-0-85729-859-1_44,

Abstract

This chapter reviews salient advances in techniques for machine-printed character recognition. Section "Overview" provides a historical perspective (The description of the historical evolution of OCR is based upon the Wikipedia entry for this topic: http://en.wikipedia.org/wiki/Optical_character_recognition. The reader is referred to that page for a more detailed review) on how OCR techniques have evolved from the earliest stage (mechanical device) to special-purpose reading machines and to personal computer software. Section "Summary of the State-of-the-Art" summarizes the state of the art in machine-printed character recognition. Sections "Segmentation and Preprocessing", "Isolated Character Recognition", and "Word Recognition" describe core technologies including binarization, document image preprocessing, page segmentation, feature extraction, character classification, and language modeling that have been developed for modern character recognition systems. Section "Systems and Applications" introduces available machine-printed OCR systems and applications.

Keywords

Artificial intelligence • Binarization • Document analysis • Hidden Markov model • Image processing • Language model • Optical character recognition • Page segmentation • Pattern recognition • Reading machine

Introduction

Optical character recognition (OCR) is one of the earliest applications of artificial intelligence and pattern recognition. From the early fixed-font reading machines to today's omni-font OCR software, the research and development of machine-printed OCR have led to mature techniques and their successful commercialization.

Overview

History of Machine-Printed OCR

The history[1] of machine-printed OCR can be divided into distinct evolutionary phases: mechanical reading devices, special-purpose reading machines, and software applications for personal computers. The development of the computer industry played an important role in the evolution of the history of OCR, including the emergence of feature extraction and recognition techniques, diversification of applications, new market demands, and massive reduction in average retail prices.

Mechanical Reading Devices

Since the first reading machine for the blind was patented in 1809 [1], OCR has been applied to applications as diverse as converting characters to telegraph code and creating reading machines for the blind. In 1912, Emanuel Goldberg invented a reading machine for automatic telegraphic message transmission. He later applied OCR techniques to the task of data entry. Soon thereafter, in 1913, Fournier d'Albe of Birmingham University invented the optophone that scanned text in printed documents and produced time-varying tones to identify different letters. Visually impaired people could "read" the printed material after learning the correspondence between the tones and letters.

Developments in OCR continued throughout the 1930s. Two systems developed during that time are shown in Fig. 10.1 and were invented by Gustav Tauschek and Paul W. Handel. Despite the differences in the design of the templates (roll vs. plate), both systems used light sources to illuminate the print. The light passed through the print and the templates and activated photoelectrical selector circuits which were designed to "recognize" the right character according to the amount of received light. These early techniques used optical devices with rotating templates of hollow letters that were used to match letters in the print – a fact that likely led to the technology itself being called "optical character recognition."

Special-Purpose Reading Machines

OCR techniques became more important when computers were invented in the 1940s. In 1949, RCA developed the first computer-based prototype OCR system funded by the US Veterans Administration. An early text-to-speech technique was also developed to speak the recognized words and help blind people read the text. In 1950, David Shepard was awarded a patent for his OCR machine "Gismo." He later founded Intelligent Machines Research Corporation (IMR) and developed the first computer-based commercial OCR systems. IMR's systems were sold to clients such as Reader's Digest, the Standard Oil Company, and the Ohio Telephony Company. OCR technology truly came into prime time in the 1960s when the United States Postal Service (USPS) began to use OCR machines for mail sorting based on technology developed by Jacob Rainbow [2].

Software Applications for Personal Computers

Fast-forwarding to the 1990s, the market for OCR technology expanded rapidly due to two factors. The first was the emergence of OCR software developed for personal computers. For example, the first version of the OmniPage OCR software for Apple Macintosh and IBM PC released by Caere in 1988 featured full functionalities of page layout analysis for text and graphics and omni-font character recognition and debuted at a retail price of $800. While $800 might not seem so inexpensive in light of the price of desktop OCR software today, at the time it represented a 50-time reduction over the price of OCR machines which sold for over $40,000.

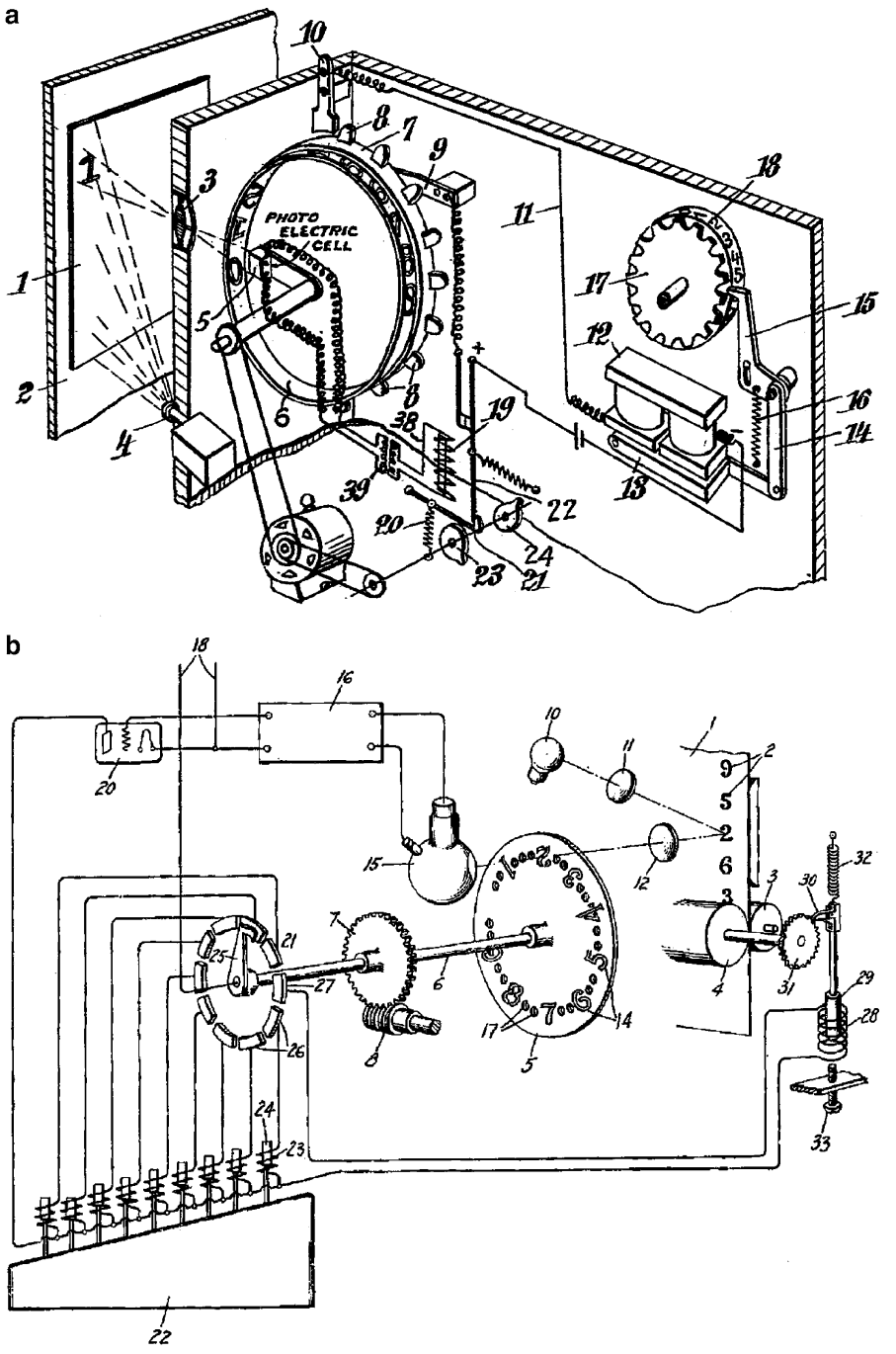

Fig. 10.1 Reading machines invented in the 1930s. (**a**) Reading machine by Gustav Tauschek (US Patent 2026329). (**b**) Reading machine by Paul W. Handel (US Patent 1915993)

The second factor was the development of a new generation of low-priced yet highly capable scanners that allowed users of personal computers to efficiently scan paper documents into digital images.

Today, OCR technologies are used for many tasks including scanner-based data entry, handheld price label scanners, recognition of documents with personal information such as bank checks and business cards, automatic mail sorting, and in business intelligence systems that support indexing and retrieval of large, heterogeneous document archives.

Technological Evolution

In early years, research in OCR techniques focused on hardware-only solutions that used mechanical implementations of template matching and selection circuits. One result of the mechanical approach was that the recognition techniques had limited to no generalization ability. The development of the computer revolutionized the field.

Since the invention of computers, sophisticated image processing and statistical pattern recognition techniques have been developed and applied to the tasks of document image analysis and recognition. Over time, independent research at several laboratories across the world has resulted in significant advances in OCR technology. These disparate efforts all share certain methodological similarities. For example, binarization is now considered a necessary step in OCR systems, and extensive research has been dedicated to the development of robust, adaptive techniques such as the Niblack [3] and the Sauvola and Pietikainen [4] algorithms. Page segmentation and layout analysis became hot topics in OCR research with the emergence of principled approaches to the analysis of complex page layout including text, graphics, and tables. Word recognition algorithms using character over-segmentation and dynamic programming have been well studied for cursive handwriting recognition and have also benefitted machine-printed OCR, especially for cursive machine-printed scripts such as Arabic. The Modified Quadratic Discriminant Function (MQDF) classification approach [5,6] has become the approach of choice for ideographic scripts such as Chinese, Japanese, and Korean that have large alphabets.

The hidden Markov model (HMM) approach was applied to character recognition [7] during the 1990s. The ability of HMM-based systems to automatically learn from unsegmented training data coupled with the script-independent modeling methodology originally developed by BBN [7] makes the HMM-based OCR technique an attractive choice for developing retargetable OCR systems that can be rapidly and economically configured for new scripts. In a manner reminiscent of its effectiveness in speech recognition, the HMM-based approach has shown remarkable flexibility and generalizability as evidenced by its highly successful application to the task of offline handwriting recognition [7].

Not surprisingly, the advent of smartphones and the concomitant ubiquitous availability of digital cameras have sparked widespread research interest in

developing algorithms and software capable of operating on the smartphone and with handheld camera-captured text images. Starting in 2005, a new workshop, camera-based document analysis recognition (CBDAR), has been held every other year to address the increasing amount of attention drawn to using camera phones and handheld cameras as the primary capturing device for OCR and present the advances in restoration, segmentation, and recognition of camera-captured documents. Images captured with handheld cameras manifest several new challenges for OCR; salient challenges include out-of-focus images with blurry text, illumination variations, and variable resolution. These challenges make binarization a particularly difficult task. As a result, there is new emphasis on using features that are directly computed from grayscale images to overcome challenges in binarization of camera-captured document images. Text detection has also become important as a research topic since locating text from video or camera-captured scene pictures where there are relatively few constraints on the layout of the text is more challenging than analysis of document images. In addition to the traditional areas of research in machine-printed character recognition, branches of the research, such as music note recognition and mathematical expression understanding, are now increasingly drawing the attention of people in the community.

From a system engineering perspective, major OCR vendors now provide distributed OCR services including web-based OCR and server-based OCR. Web-based OCR or online OCR services are typically targeted to the retail market by allowing Internet users to upload scanned document images and convert them into searchable text in prevailing formats such as HTML, PDF, and MS Word. Server-based OCR is typically deployed for distributed OCR services within the private network of an organization.

Summary of the State of the Art

The processing pipeline of a typical OCR system is shown in Fig. 10.2. First, the captured image goes through the page segmentation or text detection module and text regions are located. Often times, knowledge of the genre of the text image, e.g., whether it is a newspaper image, a magazine, a technical journal, or a business card, is used to improve the performance of the text detection module. Adaptive binarization and preprocessing steps such as noise filtering and deskewing are then applied to textual regions. Features are computed from binarized and preprocessed textual regions and sent to the word recognizer for recognition. Post-processing involves steps such as language model-based error correction and automatic highlighting of suspicious recognition errors for interactive human proofreading. A stochastic language model has become part of the recognition algorithm in many more modern systems, especially the HMM-based ones.

The core OCR performance on perfectly segmented machine-printed character images in most nonstylized fonts of major languages has reached an error rate

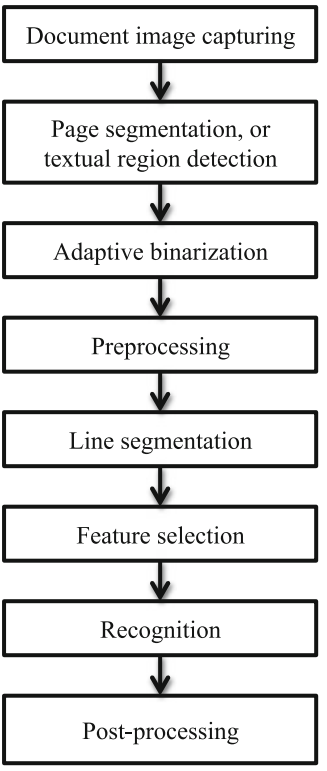

Fig. 10.2 Block diagram of a typical OCR system

of less than 1 % and, therefore, has very little room to improve. Segmentation often leads to more OCR errors than isolated character recognition. Most OCR systems have adopted recognition-driven character segmentation approaches to reduce segmentation errors. Page segmentation on complex-layout documents is now a mature technology, but text detection in camera-captured images and videos is still an active, challenging research subject.

Challenges also remain in recognition of noisy data (camera-captured/natural scene), infrequent words (named entities), offline handwritten documents, mixed printed/handwritten characters, and complex recognition-plus-parsing problems (music notes, mathematical expressions). Table 10.1 provides a summary view of the key areas of work.

As mentioned earlier, a variety of commercial OCR systems are available today for machine-printed character recognition applications including data entry, digital library, automatic mail sorting, form reader, aid for the blind, and vehicle plate reading.

In the following sections, each of the important steps in the pipeline of Fig. 10.2 is considered in detail.

Table 10.1 Maturity of areas in machine-printed character recognition

Area	Typical data	Status
Isolated character feature selection and extraction	Scanned document images	Mature techniques available
Isolated character classification	Scanned document images, camera-captured scene	Mature techniques available
Word and character segmentation	Scanned document images	Mature techniques available
Page segmentation on well-formed layout (magazine, newspaper)	Scanned document images	Mature techniques available
Character feature selection and extraction	Scanned document images	Mature techniques available Some room to improve remains
Text detection and segmentation	Camera-captured scene	Significant improvements possible
Text recognition	Scanned document images	Mature techniques available for office quality documents Significant room for improvements in "degraded" or "real-world" documents such as faxed hardcopy, forms, and mixed machine-print + handwritten content
Text recognition	Camera-captured images and videos	Active area of research; limited capabilities available

Segmentation and Preprocessing

Before the document image reaches the core OCR engine (character recognizer), it needs to go through a few preparatory steps which include binarization, page segmentation, automatic deskewing, line segmentation, and, sometimes, word/character segmentation.

Binarization

The goal of binarization is not only to compress the size of document images but also to separate the text from the background to enable the computation of useful features for character recognition. While many scanning devices often provide binarization with an adjustable constant threshold as an option when the user scans the document, the constant threshold can produce suboptimal binarization in the presence of noisy or textured backgrounds. The most commonly used automatic constant-threshold binarization algorithm is the Otsu algorithm [8]. The algorithm aims to minimize the combined spread (intra-class variance) of the intensity of the two classes separated by the threshold. Adaptive thresholding algorithms such as the Niblack

algorithm [3] and the Sauvola algorithm [4] are typically more effective at binarizing noisy, textured document images using locally selected thresholds. A survey of the literature shows that researchers have investigated a variety of other binarization approaches (▸Chap. 4 (Imaging Techniques in Document Analysis Process)).

Page Segmentation

Page segmentation refers to the process of segmenting the document image into homogeneous regions such as single-column textual regions, graphical regions, and tables. The X–Y cut is a top–down approach that divides the page recursively using the vertical and horizontal projection profiles of the page. Bottom–up approaches such as the Voronoi diagram-based segmentation [9] and the docstrum [10] have also been applied to page segmentation. Mao and Kanungo [11] (▸Chap. 5 (Page Segmentation Techniques in Document Analysis)) provides the reader with a comprehensive survey on different page segmentation approaches (Fig. 10.3).

Line, Word, and Character Segmentation

Line segmentation is the problem of dividing a textual region into images of lines of text. Word segmentation is the problem of dividing a line image into word images. Character segmentation is the problem of dividing a word or line image into character images. While HMM-based approaches jointly perform segmentation and recognition and do not require a separate pre-segmentation of the line into words or characters, most other approaches require such pre-segmentation.

Among segmentation problems for Latin-script machine-printed document images, line segmentation, and word segmentation are easier to perform than page segmentation and character segmentation. They can be performed using the horizontal and vertical projection profiles, respectively, and can be refined using connected component analysis. In general, character segmentation is a very challenging problem and requires feedback from recognition to improve the accuracy. Typically, approaches that require pre-segmentation will segment the script into lengths that are much shorter than the average length of a character and use recombination rules to stitch the atomic segments into character hypotheses. For non-HMM-based approaches, heuristics such as the average character width can be used as feedback to separate character images recursively. In the case of HMM-based approaches character duration models can be automatically trained and naturally integrated into the recognition process [7] (▸Chap. 8 (Text Segmentation for Document Recognition)).

Deskew

In most cases, the goal of deskewing is to make the text boundaries parallel to the image boundaries. Typically, it is a two-step process. The first step uses

Fig. 10.3 Page segmentation results on a newspaper image

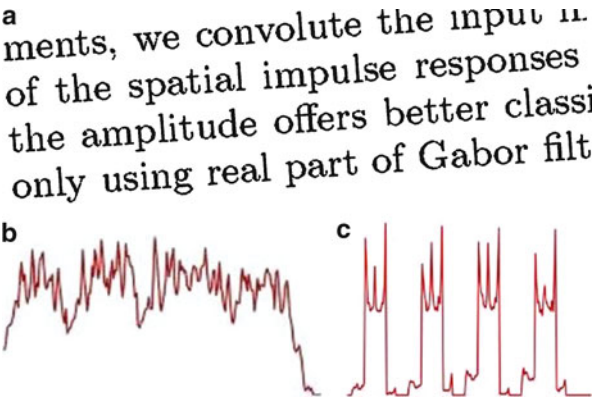

Fig. 10.4 Deskew using projection profile. (**a**) Skewed textual region. (**b**) Horizontal projection profile. (**c**) Projection profile at optimal angle

one of several projection-profile techniques to estimate the skew angle across the entire document. The second step is simply the application of an appropriate transformation to rotate the image by the inverse of the skew angle. There are several techniques described in the literature for deskewing scanned document images [12–15].

A more complex problem related to deskewing is the problem of straightening out the warping of a text line that happens when a thick book is scanned using a scanner. Again, researchers have developed effective techniques for dewarping the text lines in these cases [16–20] (Fig. 10.4) (▶Chap. 4 (Imaging Techniques in Document Analysis Process)).

Normalization

Differences in the placement of bounding boxes and variations of fonts and sizes can increase the within-class character shape variance. This affects zoning-based features in particular. Thus, character image normalization is an important step in most of zoning-based feature selection algorithms. Classic character image normalization methods aim to equalize stroke density with the character using estimated moment features of the character image [21]. Learning-based classification approaches can effectively combine features extracted from the raw image as well as the normalized image to improve performance over the use of either set of features by themselves.

Isolated Character Recognition

Isolated character recognition was an important research area in machine-printed character recognition. Although it is no longer an active area, explorations in this area have provided us indispensable technical progress, the results of which are embodied in today's segmentation-based OCR techniques.

There was an early recognition of the need for discriminative features and feature transformations that can help a recognition engine to distinguish character glyphs from one another effectively. Therefore, in addition to classifier design and training, design and implementation of feature extraction techniques was an important focus of the research in isolated character recognition.

In the following, some of the salient feature extraction and classification techniques that were developed for the task of isolated character recognition are considered. Of course, immense research activity in the area of machine learning has resulted in general pattern recognition approaches like the support vector machine (SVM) [22] that can also be applied to the problem and used within an OCR system.

Feature Extraction

Despite the many differences between existing languages in the world, at a fundamental level character and word glyphs in the vast majority of them are rendered as line drawings. Except for very few symbols, none of the scripts of modern languages use shading to express any meaning. This lack of creativity in scripts has allowed more rapid progress in the development of recognition approaches than would otherwise have been possible.

Histogram features are the most commonly used family of features for character recognition. Histogram features at the pixel level of the character image are obtained by decomposing the image signal around the pixel into several orientations using directed transformations (gradient operator, concavity, Gabor filter, etc.). The features of the whole image are usually obtained by sampling the character image both horizontally and vertically at certain intervals and computing histogram features at those selected locations. Character images may be normalized to distribute the mass evenly in the frame of the character before sampling. Histogram features are an effective tool to describe the structure of line drawings. These types of features are discussed in more detail later in this section.

GSC Features

The gradient, structural, and concavity (GSC) features [23] are an excellent example of histogram features for binarized character images. A character image is partitioned into four by four patches. From each patch, three types of features (gradient, structural, and concavity) are computed. The gradient features are computed by applying the Sobel gradient operators in Eq. (10.1):

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{10.1}$$

to the whole character image. The gradient at each pixel is quantized into 12 orientations: $30 \times i^\circ$ ($i = 0, 1, \ldots, 11$). A total of 12 features are defined as

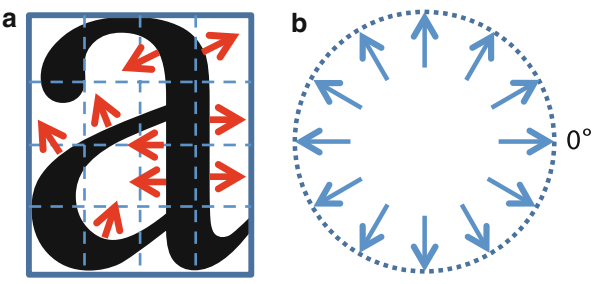

Fig. 10.5 Gradient histogram features. (**a**) Gradient orientations in a character image. (**b**) Twelve quantized orientations

$$f_{gi} = \frac{\text{\#black pixels in the patch with gradient orientation } i}{\text{\# pixels in the patch}} \tag{10.2}$$

The numerator in Eq. (10.2) does not include pixels of zero gradients as their orientations are undefined (Fig. 10.5).

Structural features are derived from gradient features. The $12^{3\times3}$ possible assignments of orientations of all 3 by 3 image blocks are clustered into 12 classes, representing the local structure of the central pixel. A total of 12 structural features are defined for each patch. Each structural feature is defined as

$$f_{si} = \frac{\text{\#black pixels of structural class } i \text{ in the patch}}{\text{\# pixels in the patch}} \tag{10.3}$$

Concavity features[2] decompose the image background signal (white pixels) into different orientations. These orientations describe the concavity of text strokes. The steps to compute concavity features are as follows. First, scan the character image from one side to the opposite side row by row if from left to right or from right to left and column by column if from top to bottom and from bottom to top. While scanning, keep track of all scanned white pixels until the first black pixel is encountered and stop. Figure 10.6 shows labeled white pixels while scanning from left to right and from top to bottom.

In the next step, histogram features are taken in the same way as in the gradient and structural features using the following computation:

$$f_{ci} = \frac{\text{\# white pixels in the patch that can only be reached when scanning at orientation } i}{\text{\# pixels in the patch}} \tag{10.4}$$

Similarly, the hold feature is defined as

$$f_{\text{hole}} = \frac{\text{\#white pixels in the patch that cannot be reached from any orientation}}{\text{\# pixels in the patch}} \tag{10.5}$$

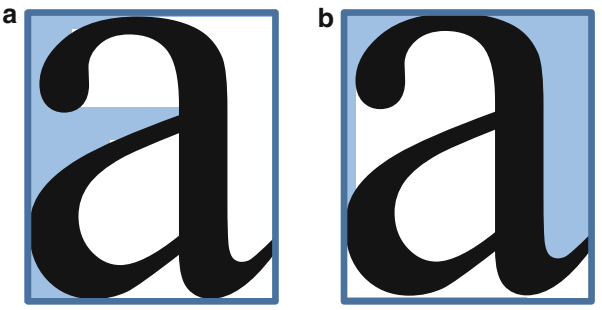

Fig. 10.6 Labeled *white pixels* while scanning from left to right and from top to bottom. Right-to-left and bottom-to-top scans are omitted. (**a**) left-to-right scan. (**b**) top-to-bottom scan

Directional Element Features

The directional element features [6] are another type of histogram features. First, the size of the character image is normalized to 64 by 64 pixels. The contour is then extracted from the character image. Next, all pairs of neighboring pixels on the contour are categorized into four orientations: 0°, 90°, 45° and 135°. Next, the character image is divided into 49 (7 by 7) overlapping patches, and each patch is further divided into four areas A, B, C, and D. A is a 4 by 4 area in the center of the patch. B is the 8 by 8 area in the center of the character image minus area A. C is a 12 by 12 area in the center of the character image minus areas A and B. D is the patch minus areas A, B, and C. The numbers of neighboring pixel pairs at each orientation i are computed for areas A, B, C, and D, denoted by $x_i^{(A)}, x_i^{(B)}, x_i^{(C)}$, and $x_i^{(D)}$, respectively. The directional element feature at orientation i is defined as

$$x_i = 4x_i^{(A)} + 3x_i^{(B)} + 2x_i^{(C)} + x_i^{(D)} \tag{10.6}$$

The resemblance between gradient features and directional element features is obvious. Gradient features decompose image signal using directions of normal lines on the contour, whereas directional element features decompose image signal using directions of tangent lines on the contour. On the one hand, normal line directions are more informative than tangent line directions since the latter can be derived from the former, but not vice versa. On the other hand, tangent line directions are more appropriate when it is known beforehand that the foreground is not always darker (or lighter) than the background in the text (Fig. 10.7).

Percentile Features

Percentile features [7] are an interesting variant of histogram features. These features are typically computed on narrow, overlapping slices of character glyphs by considering the distribution of pixel intensities over the vertical extent of the slice. Typically, first and second differences in the percentile features across adjacent

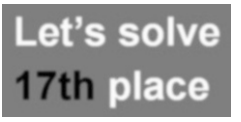

Fig. 10.7 Video text showing the foreground is not always darker than the background

slices are also used in addition to the raw percentiles themselves. The reader is referred to [7] for a detailed description of percentile features and their computation.

Gabor Features

The Gabor filter is also one way to decompose the image signal into arbitrary orientations. The Gabor filter is a 2-D linear transform defined as a directed sine wave modulated by a 2-D Gaussian low-pass filter. Gabor features are less sensitive to noise and suitable for degraded, hard-to-binarize document images:

$$h(x, y; \lambda, \phi, \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left\{-\frac{1}{2}\left(\frac{R_1^2}{\sigma_x^2} + \frac{R_2^2}{\sigma_y^2}\right)\right\} \exp\left\{i\frac{2\pi R_1}{\lambda}\right\} \quad (10.7)$$

where

$$\begin{aligned} R_1 &= x\cos\phi + y\sin\phi \\ R_2 &= y\cos\phi - x\sin\phi \end{aligned} \quad (10.8)$$

ϕ and $1/\lambda$ are the Gabor filter's orientation of interest and frequency of interest, respectively. Usually, 0°, 90°, 45°, and 135° are chosen for ϕ and twice the average stroke width in the image is chosen for λ (Fig. 10.8).

Gabor features for OCR in the literature [24–26] differ in whether the real part, imaginary part, or magnitude is computed, whether averages of positive responses and negative responses of the filter are accumulated separately for each patch, and whether the responses are binarized automatically.

Character Recognition

Perhaps the simplest character recognizer is one that uses a simple set of features in combination with a distance-based classifier. After dimension reduction using the Principal Component Analysis or Linear Discriminant Analysis, the distance from the projected feature vector of a character image to the trained centroid of each character class is computed. The label of the class that gives the shortest distance is assigned to the character image as the character recognition result.

The MQDF is a modified version of the Quadratic Discriminant Function (QDF) classifier. The Quadratic Discriminant Function (QDF) for each class is defined as

$$f_0^{(l)} = \left(x - \mu^{(l)}\right)^T \left\{\Sigma^{(l)}\right\}^{-1} \left(x - \mu^{(l)}\right) + \log\left|\Sigma^{(l)}\right| - 2\log\Pr\left(\omega^{(l)}\right) \quad (10.9)$$

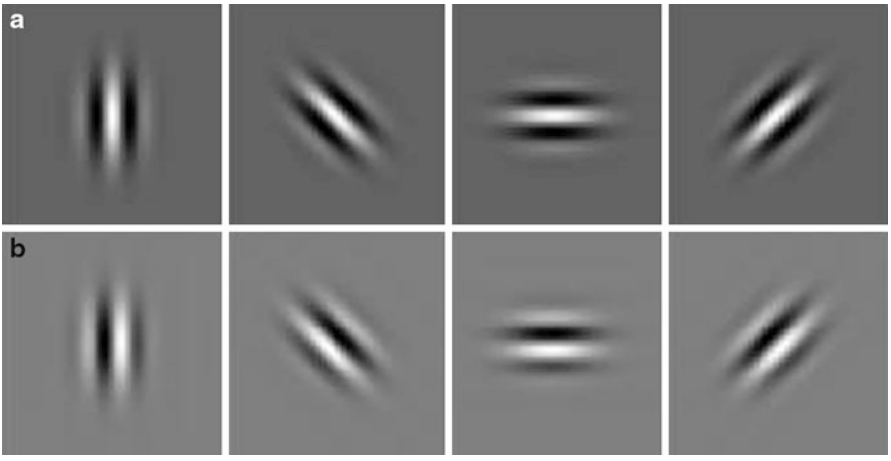

Fig. 10.8 Spatial impulse response of a Gabor filter. The filter is visualized such that *black* represents negative values, the *gray* background represents zero, and *white* represents positive values. (**a**) Real part. (**b**) Imaginary part

where x is an n-dimensional feature vector, l is the class label, $\mu^{(l)}$ and $\Sigma^{(l)}$ are the mean vector and covariance matrix of class l, and $\log \Pr\left(\omega^{(l)}\right)$ is the prior probability of class l. $\mu^{(l)}$and $\Sigma^{(l)}$ are obtained using the maximum likelihood estimation. The QDF can also be rewritten as

$$f_0^{(l)} = \sum_{i=1}^{n} \frac{1}{\lambda_i^{(l)}} \left\{ \varphi_i^{(l)T} \left(x - \mu^{(l)} \right) \right\}^2 + \log \prod_{i=1}^{n} \lambda_i^{(l)} - 2 \log \Pr\left(\omega^{(l)}\right) \tag{10.10}$$

where $\lambda_i^{(l)} (\lambda_i^{(l)} \geq \lambda_{i+1}^{(l)})$ and $\varphi_i^{(l)}$ are the ith eigenvalue and eigenvector of $\Sigma^{(l)}$. Several modifications to the QDF in Eq. (10.10) have been proposed in the literature. They differ in their definition depending on whether the objective is to reduce the estimation errors in eigenvectors corresponding to small eigenvalues or to generalize the normal distribution.

- MQDF for reducing the estimation errors in eigenvectors [5]

$$f^{(l)} = \sum_{i=1}^{n} \frac{1}{\lambda_i^{(l)} + h^2} \left\{ \varphi_i^{(l)T} \left(x - \mu^{(l)} \right) \right\}^2 + \log \prod_{i=1}^{n} \left(\lambda_i^{(l)} + h^2 \right) - 2 \log \Pr\left(\omega^{(l)}\right) \tag{10.11}$$

  where h is an empirically selected constant or

$$f^{(l)} = \sum_{i=1}^{n} \frac{1}{\lambda_i'^{(l)}} \left\{ \varphi_i^{(l)T} \left(x - \mu^{(l)} \right) \right\}^2 + \log \prod_{i=1}^{n} \left(\lambda_i'^{(l)} \right) - 2 \log \Pr\left(\omega^{(l)}\right) \tag{10.12}$$

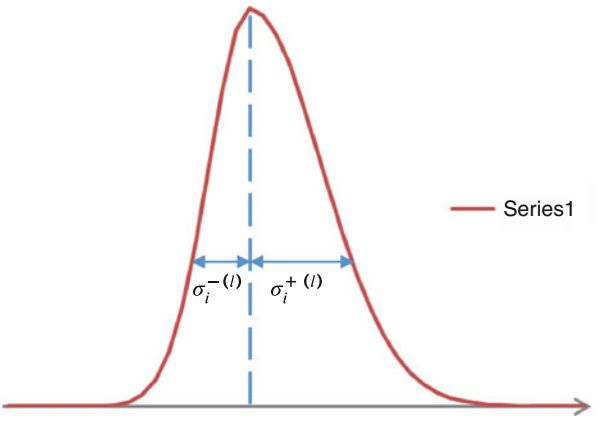

Fig. 10.9 Asymmetric distribution assumed by the MQDF defined in Eq. (10.14)

where

$$\lambda_i^{\prime(l)} = \begin{cases} \lambda_i^{(l)}, \text{if } i \le k \\ h^2, \text{if } i > k \end{cases} \tag{10.13}$$

and h and k are empirically selected parameters.

- MQDF that generalizes the normal distribution [6]

$$f^{(l)} = \sum_{i=1}^{n} \frac{1}{\hat{\lambda}_i^{(l)}} \left\{ \varphi_i^{(l)T} \left(x - \mu^{(l)} \right) \right\}^2 + \log \prod_{i=1}^{n} \hat{\lambda}_i^{(l)} - 2 \log \Pr\left(\omega^{(l)} \right) \tag{10.14}$$

where

$$\hat{\lambda}_i^{(l)} = \begin{cases} \left(\sigma_i^{+(l)} \right)^2, \text{if } \varphi_i^{(l)T} \left(x - \mu^{(l)} \right) \ge 0 \\ \left(\sigma_i^{-(l)} \right)^2, \text{if } \varphi_i^{(l)T} \left(x - \mu^{(l)} \right) < 0 \end{cases} \tag{10.15}$$

and $\left(\sigma_i^{+(l)} \right)^2$ and $\left(\sigma_i^{-(l)} \right)^2$c are quasi-variances estimated separately for two conditions in Eq. (10.14). Equation (10.14) assumes that the de-correlated signal $\varphi_i^{(l)T} \left(x - \mu^{(l)} \right)$ follows the asymmetric, quasi-normal distribution shown in Fig. 10.9. The asymmetric distribution can better model the distribution of the signal which is usually different from the normal distribution.

While the MQDF was reviewed in detail because of the attention it had received in isolated character recognition work, classifiers such as the Gaussian mixture model (GMM), SVM, and neural networks that have achieved advances in later research on handwriting recognition can and have been applied to machine-printed character recognition for finer modeling and better performance at the expense of longer training and decoding time. Indeed, if one were to build an isolated character recognition system today, Bayesian or kernel-based classification techniques would top the list of viable candidates for the classification task.

Word Recognition

Historically, with the exception of HMM-based approaches, character segmentation and recognition has provided a basis for word recognition. Even in the case of HMM-based techniques, character recognition forms the basis of word recognition. Character segmentation can be improved using the matching score produced during character recognition – of course, improving the matching score requires the design of better features, more effective feature transformations, and more accurate classification approaches. Furthermore, contextual information such as the word lexicon and language model can also be used to further improve word recognition. These techniques will be discussed in this section.

Character Segmentation

Features for Finding Segmentation Points

Character images can be identified using white space and connected component analysis when they are well separated. When touching characters exist, features such as local minima in the vertical projection profile [27] can be used to locate optimal locations of character boundaries (Fig. 10.10) (▸Chap. 8 (Text Segmentation for Document Recognition)).

Recognition-Based Segmentation

Sequential segmentation of character images without the feedback from recognition is vulnerable to touching characters and ambiguous character boundaries. It contributes a significant fraction to OCR errors when it is applied to document images of omni-font, variable aspect ratio characters.

Recognition-based segmentation creates multiple segmentation hypotheses for a word image and selects the optimal hypothesis using matching scores. Kovalevsky's algorithm [28] is based on a global search on the lattice of all possible segmentation hypotheses for the optimal path where the mean-squared matching error is minimized. Casey's algorithm [29] is based on similar idea. Starting from an initial guess of the boundary of the current character, it gradually moves the right boundary to the left until a meaningful character is found. And then, the same process is repeated for the next character until it either exhausts the set of cut points or every segmented character image matches a character class within a predefined threshold.

In [28] and [29], experimental results on machine-printed documents are reported. In these earliest recognition-based segmentation algorithms, sliding windows are used to implicitly provide sequences of tentative segmentation points. Nearly all later research works [30–32] address segmentation for handwriting recognition rather than machine-printed OCR owing to the fact that handwriting recognition is a more challenging problem and has remained an active area. In most of these works, complex dissection methods are developed to provide

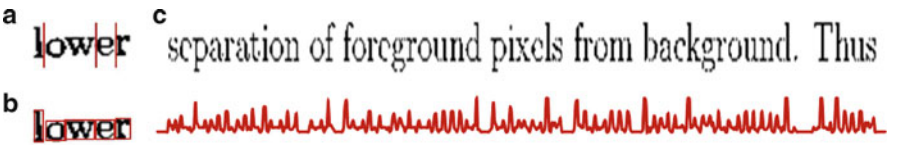

Fig. 10.10 Features for character segmentation. (**a**) White space between characters. (**b**) Connected components of a word image. (**c**) Vertical projection profile of a line image

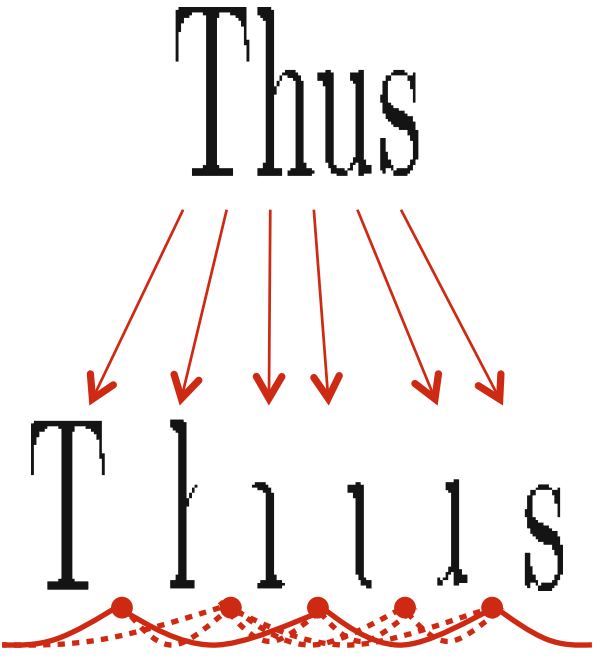

Fig. 10.11 Word segmentation using over-segmentation. The optimal segmentation is denoted by the path of *solid lines*

over-segmentation of word images, and word lexicons are used to limit the size of the search space. The basic steps are summarized as follows (Fig. 10.11):

1. Find sufficiently many segmentation points to include all correct segmentation points. In addition to white space, connected components, and projection profile analysis, concaves in ligatures between characters are also used to locate segmentation points.
2. Select an optimal subsequence out of the segmentation points that minimizes the character sequence matching error using dynamic programming.

Complex dissection methods are not always superior to the implicit, sliding window-based segmentation owing to the fact that the latter does not rely on heuristic rules and can be generalized to text in different languages. The HMM-based OCR can also be thought of as an approach to generate implicit segmentation. Using the HMM, one can automate the process of whole-sentence data labeling and increase the scalability of the OCR system.

Hidden Markov Model OCR

In the HMM OCR system [7], a text line image is modeled as a left-to-right signal and represented as a sequence of feature vectors sampled at a fixed frame rate (Fig. 10.12). Each frame is further divided horizontally into several patches and image features are computed from each patch. Remember a similar way was adopted to divide a character image into patches in section "GSC Features". Here, the number of horizontal cuts is still fixed, but the number of vertical cuts is variable.

The HMM-based speech recognition algorithms can be adapted to solve the OCR problem by replacing speech features with the multivariate sequential OCR features. The HMM of a character is a first-order stationary Markov chain of n emitting states with left-to-right state transition shown in Fig. 10.13. Arrows in Fig. 10.13 indicate nonzero state transition probabilities. The whole sequence of feature vectors is treated as a time series, conforming to the convention in speech recognition. Each vector $\boldsymbol{O}_t$ is called an observation of the time series at time t. All feature vectors of each state follow a continuous observational probability distribution modeled as a GMM.

One can build word HMMs in a manner such that the HMM for a word is the linearly connected model of HMMs for all characters in the word. Thus, Fig. 10.13 can also represent a word HMM except that the number of states in the word HMM is n times the number of characters in the word.

Isolated word recognition can be performed by evaluating word posterior probabilities using word HMMs. One can also use the complex HMM state transition map shown in Fig. 10.9 for continuous word recognition. Each row in Fig. 10.14 denotes a word HMM. The rectangular terminals are non-emitting states that denote the start and end of a character. No observational distribution is associated with non-emitting states.

For a description of how candidate segmentations from an HMM system can be used to combine scores from an HMM system with scores from complex 2-D classification approaches reminiscent of isolated character recognition in a tractable manner, the reader is referred to [33].

Parameter Learning

Parameters of HMM including Gaussian means, covariances, mixture weights, and transition probabilities are estimated from the training data using the expectation-maximization algorithm known as the Baum–Welch algorithm. In the E-step, the probability of transitioning from state i to j at time t $\Pr\left(S_t = i, S_{t+1=j} \,|O, \lambda\right)$ and the probability for selecting the kth mixture component in state j for the observation at time $t\Pr\left(S_t = j, M_{t=k} \,|O, \lambda\right)$ are estimated for each t. In the M-step, model parameters λ are reestimated using estimated $\Pr\left(S_t = i, S_{t+1=j} \,|O, \lambda\right)$ and $\Pr\left(S_t = j, M_{t=k} \,|O, \lambda\right)$ and the observational sequence $\boldsymbol{O}$.

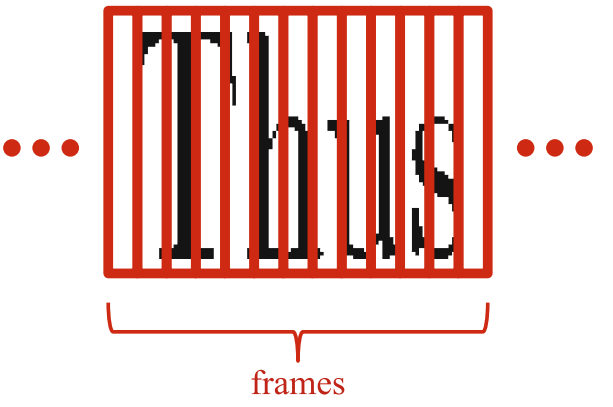

Fig. 10.12 Frames sampled at constant sample rate using a sliding window

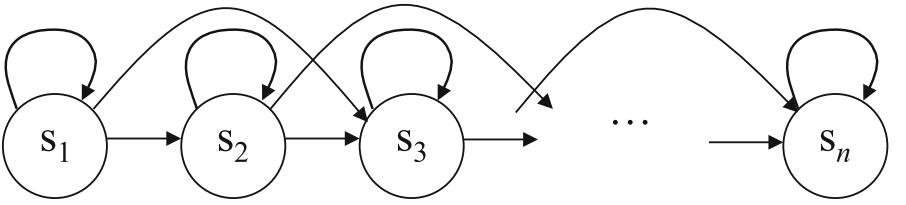

Fig. 10.13 Left-to-right state transitions in the character HMM

Recognition

Recognition is the task of finding the optimal hidden state sequence s^* that produces the observation with maximal posterior probability

$$s^* = \underset{s}{\operatorname{argmax}} \Pr\,(s|O,\lambda) \tag{10.16}$$

using the Viterbi algorithm. Beam search can be used to reduce the search space in the Viterbi algorithm and accelerate the decoding process.

n-Gram Language Model

The n-gram language model describes the joint probability that a series of words appear in the language in order $w_1 \rightarrow w_2 \rightarrow \cdots \rightarrow w_N$. Assuming the probability is a Markov chain, the n-gram language model can be written as

$$\begin{aligned}\Pr\,(w_1, w_2, \ldots, w_n) &= \Pr\,(w_1)\Pr\Pr\,(w_2|w_1)\ldots\Pr\,(w_T|w_{T-n+1}, \ldots, w_{T-1}) \\ &= \prod_{i=1}^{T} \Pr\,(w_i|w_{i-n+1}, \ldots, w_{i-1})\end{aligned} \tag{10.17}$$

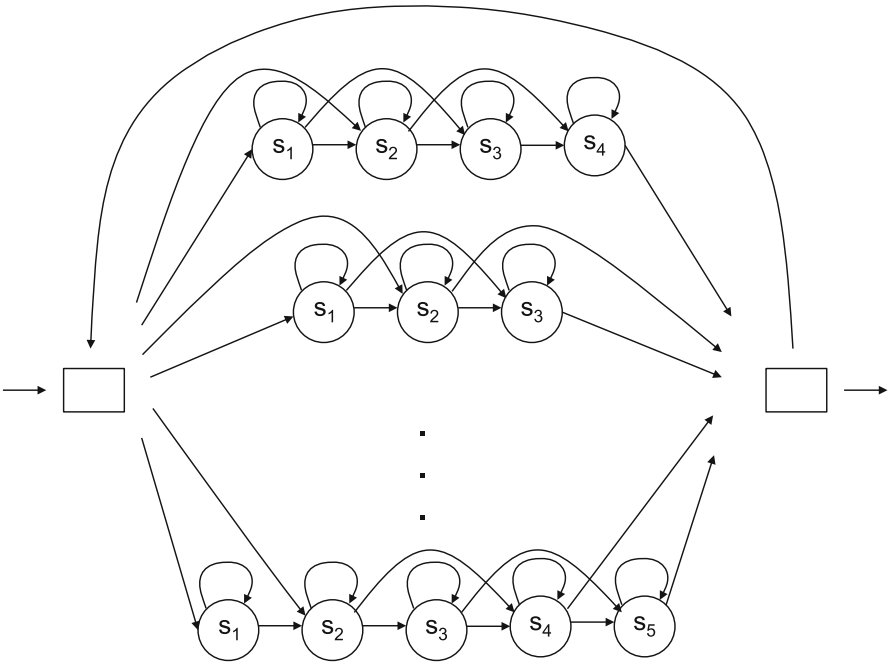

Fig. 10.14 State transitions of the HMM for continuous word recognition

When the context in the word sequence is considered, the objective of OCR can be formulized as finding the optimal word sequence that produces the maximum posterior probability on the observational sequence:

$$w* = \underset{w}{\text{argmax}} \Pr(w|O) = \underset{w}{\text{argmax}} \Pr(w) \Pr(O|w) \tag{10.18}$$

Pr (w) and Pr $(O|w)$ are provided by the word model and word recognizer, respectively. As the simplest case, a bigram language model can be integrated into the decoding algorithm using word networks [34] such that the bigram probabilities are treated as between-word transition probabilities shown in Fig. 10.15. Note that this is only applicable to bigram. Integration of trigram and higher-order language models requires complex algorithms that memorize the search history. An alternative is to generate multiple candidate word sequences using bigram in the first pass of decoding and reevaluate these word sequences using a high-order language model in the second pass of decoding.

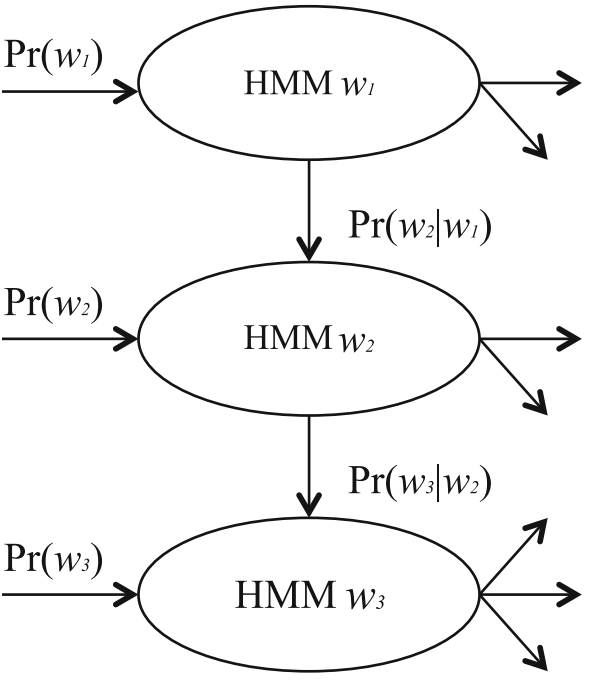

Fig. 10.15 Bigram modeled as between-word transition in an HMM word network

Systems and Applications

Applications

Desktop OCR

Desktop OCR features the use of scanners as the input device, automatic page segmentation, form analysis, multi-language omni-font character recognition, and support of common output file formats. The applications of desktop OCR include but are not limited to automated data entry, digital library, and Internet publishing.

Web Services

More and more OCR vendors such as Free Online OCR, OnlineOCR.net, and Google Docs now provide online OCR services as web applications. Users can upload their document images to the vendor's website and receive the OCR results.

Camera OCR

The applications of OCR using digital cameras as the input device can be divided into two categories. The first category of OCR applications is based on handheld

cameras or smartphones. These applications include business card, invoice, and bank check recognition. The second category is based on closed-circuit television and is mostly applied to video surveillance including license plate number and freight container number recognition.

Commercial Software

Commercially available software products are available both in the form of end-user applications and as software development kits (SDKs) that allow users to customize the capability to fit their needs. This section presents a brief survey of commercially available OCR capabilities.

Omni-Font OCR Software

The latest commercial OCR software applications are all typically equipped with the capability of reading multiple languages and fonts. For example, FineReader v11 by ABBYY now supports any combination of 189 languages, although dictionary support is only provided for 45 languages. OmniPage v17 by Nuance now supports over 120 languages.

To optimize the user experience, many productivity features are supported by major OCR systems to render the OCR result. These features include retention of original layout and fonts and conversion to prevalent formats including MS Office, WordPerfect, searchable PDF, HTML, and SharePoint.

Below is a list of some of the major OCR vendors in alphabetical order:

- ABBYY
- Nuance
- Presto! OCR
- Presto! PageManager
- Raytheon BBN Technologies Corp.
- Readiris
- Sakhr Inc.

SDK and Open-Source OCR

Many vendors provide SDKs to provide users the ability to customize the OCR software for their proprietary tasks and documents. Here is a list of OCR software applications for which the SDK is available [35], again sorted in alphabetical order:

- ABBYY FineReader
- Aquaforest OCR SDK
- CuneiForm/OpenOCR
- Digital Syphon's Sonic Imagen
- ExperVision TypeReader & RTK
- Indian Scripts OCR
- LEADTOOLS
- NSOCR

- Ocrad
- OmniPage
- PrimeOCR
- Puma.NET
- Raytheon BBN Technologies Corp.
- Readiris
- Sakhr Inc.
- Transym OCR

Tesseract is an open-source library for machine-printed recognition featuring line, word, and character segmentation and character recognition for Latin scripts. It has demonstrated competitive OCR accuracy according to the Annual Test of OCR Accuracy [36].

Well-Known Evaluations and Contests

The Annual Test of OCR Accuracy was held by the Information Science Research Institute (ISRI) in 1992–1996 to evaluate the performance of participants' OCR systems on text zones from English and Spanish magazines and newspapers. According to their results, the lowest character recognition error rates of all teams were from 0.5 to 5 %, depending on the image quality of the dataset.

Contests aiming at evaluating methods in subareas complementary to character recognition were initiated along with the International Conference on Document Analysis and Recognition (ICDAR). The contests of those subareas include Robust Reading and Text Locating (initiated in 2003) for text detection and recognition of camera-captured scenes, Page Segmentation Competition (initiated in 2003), and Document Image Binarization Contest (DIBCO) (initiated in 2009).

Conclusion

This chapter provides a brief review of some of the salient aspects of the history of machine-printed character recognition techniques, described basic feature extraction and classification techniques for machine-printed character recognition, and presented applications, systems, and evaluations of OCR.

Although there are mature techniques for features and classification of isolated characters, major sources of character recognition errors including text detection and segmentation are still important research topics in the community. OCR of documents that are not compiled in any natural language, e.g., music notes and mathematical equations, are particularly fertile areas for the intrepid researcher as are tasks such as inferring the logical reading order of multicolumn text zones and even data presented in a tabular manner.

Increasing popularity of multimedia and the concomitant rise of smartphones brings opportunity for recognition of text in videos and camera-captured images. It also offers new challenges in the design and extraction of features in

low-resolution, colored images, in the adaptation of models that reuse large volumes of binarized document image training data and in continuous online learning that leverages crowd-sourced information such as opportunistic or targeted annotations and corrections provided by users.

Cross-References

▸Document Analysis in Postal Applications and Check Processing
▸Page Segmentation Techniques in Document Analysis
▸Text Segmentation for Document Recognition

Notes

[1]The description of the historical evolution of OCR is based upon the Wikipedia entry for this topic: http://en.wikipedia.org/wiki/Optical_character_recognition. The reader is referred to that page for a more detailed review.

[2]Some of the features categorized as concavity features in [8] have nothing to do with stroke concavity. They are derived from the local intensity and whether or not pixels belong to horizontal and vertical strokes.

References

1. Schantz HF (1982) The history of OCR, optical character recognition. Recognition Technologies Users Association, Manchester Center
2. Shahi M, Ahlawat AK, Pandey BN (2012) Literature survey on offline recognition of handwritten Hindi curve script using ANN approach. Int J Sci Res Publ 2(5):362–367
3. Niblack W (1986) An introduction to digital image processing. Prentice Hall, Englewood Cliffs
4. Sauvola J, Pietikainen M (2000) Adaptive document image binarization. Pattern Recognit 33(2):225–236
5. Kimura F, Takashina K, Tsuruoka S, Miyake Y (1987) Modified quadratic discriminant functions and the application to Chinese character recognition. IEEE Trans Pattern Anal Mach Intell 9(1):149–153
6. Kato N, Suzuki M, Omachi S, Aso H, Nemoto Y (1999) A handwritten character recognition system using directional element feature and asymmetric Mahalanobis distance. IEEE Trans Pattern Anal Mach Intell 21(3):258–262
7. Natarajan P, Lu Z, Schwartz R, Bazzi I, Makhoul J (2001) Multilingual machine printed OCR. In: Bunke H, Caelli T (eds) Hidden Markov models – applications in computer vision. Series in machine perception and artificial intelligence, vol 45. World Scientific Publishing Company, River Edge, NJ, USA
8. Otsu N (1979) A threshold selection method from Gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
9. Kise K, Sato A, Iwata M (1998) Segmentation of page images using the area Voronoi diagram. Comput Vis Image Underst 70:370–382
10. O'Gorman L (1993) Document spectrum for page layout analysis. IEEE Trans Pattern Anal Mach Intell 15:1162–1173

11. Mao S, Kanungo T (2001) Empirical performance evaluation methodology and its application to page segmentation algorithms. IEEE Trans Pattern Anal Mach Intell 23(3): 242–256
12. Lu Y, Tan CL (2003) A nearest neighbor chain based approach to skew estimation in document images. Pattern Recognit Lett 24:2315–2323
13. Kapoor R, Bagai D, Kamal TS (2004) A new algorithm for skew detection and correction. Pattern Recognit Lett 25:1215–1229
14. Li S, Shen Q, Sun J (2007) Skew detection using wavelet decomposition and projection profile analysis. Pattern Recognit Lett 28:555–562
15. Singh C, Bhatia N, Kaur A (2008) Hough transform based fast skew detection and accurate skew correction methods. Pattern Recognit Lett 41:3528–3546
16. Zhang Z, Tan CL (2001) Recovery of distorted document images from bound volumes. In: Proceedings of the 6th international conference on document analysis and recognition, Seattle, pp 429–433
17. Cao H, Ding X, Liu C (2003) A cylindrical surface model to rectify the bound document image. In: Proceedings of the 9th IEEE international conference on computer vision, Nice, vol 1, pp 228–233
18. Brown MS, Tsoi Y-C (2006) Geometric and shading correction for images of printed materials using boundary. IEEE Trans Image Process 15(7):1544–1554
19. Liang J, DeMenthon D, Doermann DS (2008) Geometric rectification of camera-captured document images. IEEE Trans Pattern Anal Mach Intell 30(4):591–605
20. Zhang L, Yip AM, Brown MS, Tan CL (2009) A unified framework for document restoration using inpainting and shape-from-shading. Pattern Recognit 42(12):2961–2978
21. Miyoshi T, Nagasaki T, Shinjo H (2009) Character normalization methods using moments of gradient features and normalization cooperated feature extraction. In: Proceedings of the Chinese conference on pattern recognition, Nanjing
22. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other Kernel-based learning methods. Cambridge University Press, Cambridge/New York
23. Favata JT, Srikantan G, Srihari SN (1994) Handprinted character/digit recognition using a multiple feature/resolution philosophy. In: Proceedings of the fourth international workshop frontiers in handwriting recognition, Taipei, pp 57–66
24. Huo Q, Ge Y, Feng Z-D (2001) High performance Chinese OCR based on Gabor features, discriminative feature extraction and model training. Proc Int Conf Acoust Speech Signal Process 3:1517–1520
25. Wang X, Ding X, Liu C (2005) Gabor filters-based feature extraction for character recognition. Pattern Recognit 38(3):369–379
26. Chen J, Cao H, Prasad R, Bhardwaj A, Natarajan P (2010) Gabor features for offline Arabic handwriting recognition. In: Proceedings of the document analysis systems, Boston, pp 53–58
27. Lu Y (1993) On the segmentation of touching characters. In: Proceedings of the international conference on document analysis and recognition, Tsukuba, pp 440–443
28. Kovalevsky VA (1968) Character readers and pattern recognition. Spartan Books, Washington, DC
29. Casey RG, Nagy G (1982) Recursive segmentation and classification of composite patterns. In: Proceedings of the 6th international conference on pattern recognition, Munich
30. Fujisawa H, Nakano Y, Kurino K (1992) Segmentation methods for character recognition: from segmentation to document structure analysis. Proc IEEE 80(8):1079–1092
31. Favata JT, Srihari SN (1992) Recognition of general handwritten words using a hypothesis generation and reduction methodology. In: Proceedings of the fifth USPS advanced technology conference, Washington, DC
32. Sinha RMK, Prasada B, Houle G, Sabourin M (1993) Hybrid recognition with string matching. IEEE Trans Pattern Anal Mach Intell 15(10):915–925
33. Natarajan P, Subramanian K, Bhardwaj A, Prasad R (2009) Stochastic segment modeling for offline handwriting recognition. In: Proceedings of the international conference on document analysis and recognition, Barcelona, pp 971–975

34. Fink GA (2007) Markov models for pattern recognition: from theory to applications. Springer, Berlin/Germany, Heidelberg/Germany, New York/USA
35. Comparison of optical character recognition software, Wikipedia page. http://en.wikipedia.org/wiki/Comparison_of_optical_character_recognition_software
36. Rice SV, Jenkins FR, Nartker TA (1995) The fourth annual test of OCR accuracy. In: Proceedings of the annual symposium on document analysis and information retrieval, Las Vegas, Nevada, USA

Further Reading

Natarajan P, Lu Z, Schwartz R, Bazzi I, Makhoul J (2001) Multilingual machine printed OCR. Int J Pattern Recognit Artif Intell 15(1):43–63

Natarajan P, Subramanian K, Bhardwaj A, Prasad R (2009) Stochastic segment modeling for offline handwriting recognition. In: Proceedings of the international conference on document analysis and recognition, Barcelona, pp 971–975