Chapter 7
Reading-Life Log as a New Paradigm of Utilizing Character and Document Media

Koichi Kise, Shinichiro Omachi, Seiichi Uchida, Masakazu Iwamura, Masahiko Inami and Kai Kunze

Abstract "You are what you read." As this sentence implies, reading is important for building our minds. We are investing a huge amount of time for reading to input information. However the activity of "reading" is done only by each individual in an analog way and nothing is digitally recorded and reused. In order to solve this problem, we record reading activities as digital data and analyze them for various goals. We call this research "reading-life log." In this chapter, we describe our achievements of the reading-life log. A target of the reading-life log is to analyze reading activities quantitatively and qualitatively: when, how much, what you read, and how you read in terms of your interests and understanding. Body-worn sensors including intelligent eyewear are employed for this purpose. Another target is to analyze the contents of documents based on the users' reading activities: for example, which are the parts most people feel difficult/interesting. Materials to be read are not limited to books and documents. Scene texts are also important materials which guide human activities.

K. Kise (✉) · M. Iwamura
Osaka Prefecture University, 1-1 Gakuencho, Naka, Sakai, Osaka 599-8531, Japan
e-mail: kise@cs.osakafu-u.ac.jp

M. Iwamura
e-mail: masa@cs.osakafu-u.ac.jp

S. Omachi
Tohoku University, 6-6-05, Aramaki Aza Aoba Aoba-ku, Sendai, Miyagi 980-8579, Japan
e-mail: machi@ecei.tohoku.ac.jp

S. Uchida
Kyushu University, 744, Motooka, Nishi, Fukuoka 819-0395, Japan
e-mail: uchida@ait.kyushu-u.ac.jp

M. Inami
The University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo 113-8654, Japan
e-mail: inami@inami.info

K. Kunze
Keio University, 4-1-1 Hiyoshi, Kohoku, Yokohama, Kanagawa 223-8526, Japan
e-mail: kai.kunze@gmail.com

T. Nishida (ed.), *Human-Harmonized Information Technology, Volume 2*,
DOI 10.1007/978-4-431-56535-2_7

Keywords Reading-life log · Document image retrieval · Scene character recognition · Scene character dataset · Font generation · Eye-tracking · Wordometer · Smart eyewear · Document annotation · AffectiveWear · Augmented narrative

7.1 Introduction

"Reading[1] is to the mind what food is to the body." This sentence emphasizes the importance of reading for building our minds. By knowing what you have been read, we are able to know more about you. Reading-Life Log (RLL) is a project that focuses on a human's reading activity and to know and enhance human ability.

For the majority of people, reading is a primary means of acquiring information. Few can spend a whole day without reading anything in their modern life. In other words, people's life is to input information by reading and to process it. However, the activity of "reading" is done only by each individual in an analog way. Although people spend a great deal of time reading, the activity of reading itself cannot be used later, because of its analog nature. In order to solve this problem, we record reading activities as digital data and analyze them for various goals. We call this research "reading-life log."

There is a wide variety of research items in the "reading-life log": what, when, and how much you read, and how you read in terms of your interests and understanding. In the research of the reading-life log, we obtain the above information by observing both readers and objects to be read. To observe readers, we employ various sensors most of which are body-worn: for example, an eye-tracker for the analysis of eye gaze, and an EEG device for the analysis of brain activity. To observe objects to be read, we employ a camera mounted on the reader. It enables us to extract character information read by people.

By acquiring the above information from a single person, we are able to estimate the quality and the quantity of knowledge he/she has acquired through reading. This information can be used for many purposes. For example, if the reader wishes to improve his/her ability in a language, we can help him/her by visualizing the amount of learning, as well as by showing the weak points. This process can be viewed as knowing people through materials to be read. On the other hand, by accumulating reading activities and reactions regarding a material, we are able to grasp information about who reads it, who likes it, which parts interest whom, and so on. This type of information is valuable for revising the contents of the material. A step forward would be to use it to establish the relationship among its readers, as well as the relationship among materials through their readers.

[1] Table 7.1 and Figs. 7.1, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 7.10, 7.11, 7.12, 7.13, 7.17, 7.18, 7.19, 7.20 are originally published in [1] and copyrighted by IEICE. They are granted to use in this article with the permission number 16KB0074. The research described in this chapter has been approved by the research ethics committee in Osaka Prefecture University.

Materials to be read are not limited to books and documents, but include posters, sheets of paper on bulletin boards, and signboards. When these materials are considered, information processing by readers is not limited to acquiring knowledge but to guiding their activities by the information. An easy example is a sign: we can guide ourselves by reading directions to the goal. From the opposite viewpoint, reading text on such materials allows us to estimate the reader's goal.

In order to realize the abovementioned information processing, what are the necessary functions we need to implement? It is at least necessary to read the characters in documents and other materials. Detection of reading activities and analysis of them are also important functions. In this chapter, we describe our latest results towards realizing the reading-life log.

Section 7.2 overviews the whole research of the reading-life log. Section 7.3 first describes fundamental technologies and tools for implementing the reading-life log. Section 7.4 is the main part of this chapter and describes several different reading-life logs and their technologies. Section 7.5 concludes this chapter with some future work to be undertaken.

7.2 Overview of the Research Field

7.2.1 Functions

An overview of all our research topics is shown in Fig. 7.1. The purpose of the research is to establish mutual analysis of materials (characters/documents) and their reading activities to establish a human-harmonized information environment for reading. The left-hand part of the applications indicates the analysis of reading activities based on characters and documents. On the other hand, an analysis of characters and documents based on reading activities is shown on the right.

The goal of the applications in the left-hand part is to build a reading-life log about what, when, and how much the reader reads. Depending on the materials to be read, this application is subdivided into two areas: reading-life log for documents and reading-life log for scene text.

The reading-life log for documents requires the computer to read documents simultaneously like the reader does. This function can be supported by two fundamental technologies: real-time character recognition and real-time document image retrieval. By using either of them, we can access the contents that the reader reads.

For the reading-life log for scene text, we employ real-time character recognition and omnidirectional character recognition as its fundamental technologies. The former recognizes characters pointed out by a camera. The latter on the other hand requires no pointing action; all characters around the reader are recognized to characterize the scene.

Both real-time and omnidirectional character recognition are based upon the technologies of basic character detectors and recognizers as well as a large-scale char-

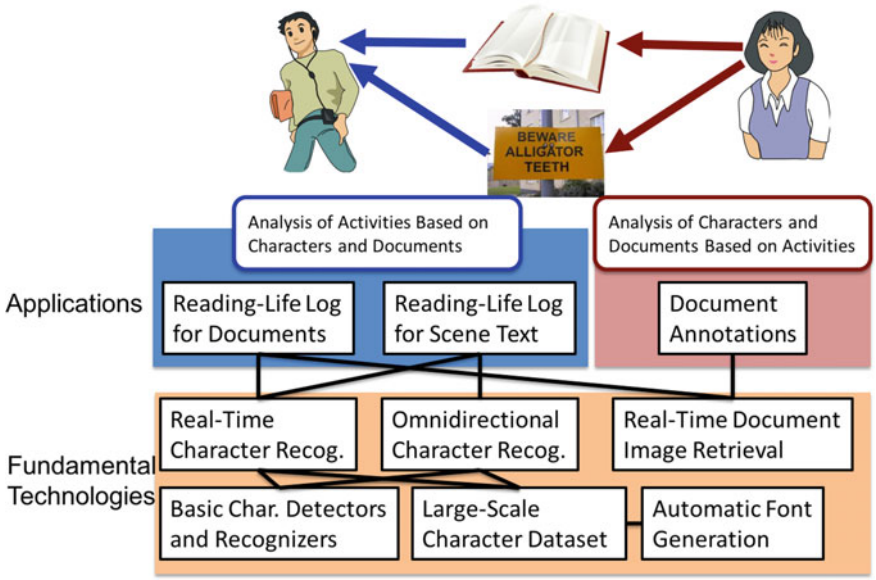


Fig. 7.1 Overview of research on the reading-life log

acter dataset, which is used as learning samples of recognizers. Although most of the samples in the dataset are labeled manually, this can be automatically done by using other technologies. For example, a technology called automatic font generation allows us to produce any font automatically, which can be used as a learning sample. Another way is to employ real-time document image retrieval for automatic labeling of camera-captured characters.

The right-hand part of the figure represents the processing of the opposite direction. Documents and characters are analyzed based on their reading activities. For example, difficult words for a reader are automatically recognized by analyzing eye gaze while reading. Although this is an example of automatic annotation of documents in terms of difficult words, we also provide the functionality of annotating documents manually. Annotated documents can be used in many different ways. In the case of difficult words, a direct application is to give the list of difficult words at the end of the day to encourage the reader to review them. Another, more sophisticated, way is to create an entertainment by playing annotations of a document.

7.2.2 Devices

For the case of the reading-life log for scene text, the main and only device we use is a camera. In the case of omnidirectional recognition, an omnidirectional camera that

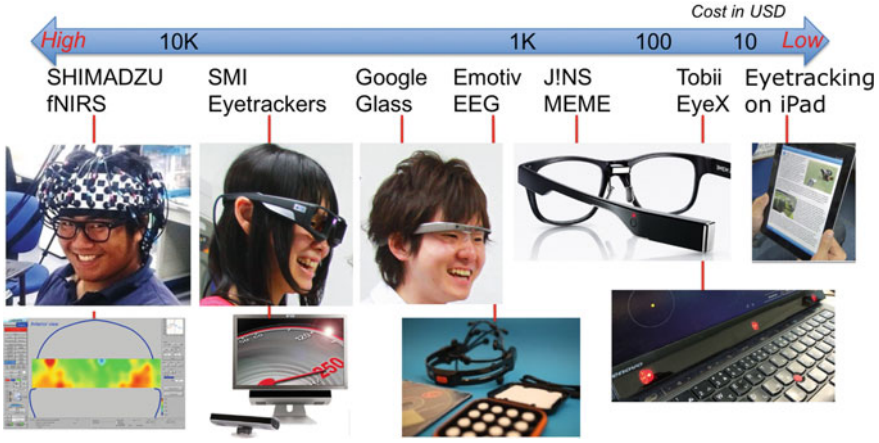


Fig. 7.2 Various devices for the reading-life log

enables us to capture 360-degree images is used. For other cases, a normal camera is employed.

To implement the reading-life log for documents, on the other hand, we employ various devices as shown in Fig. 7.2. The most expensive device we use is functional near-infrared spectroscopy (fNIRS), which measures brain activity from localized blood flow. We also employ EEG for a similar purpose. Eye-trackers ranging from expensive ones (SMI) to free ones (software implementation on iPad) are used to know where the reader is looking. Google Glass allows us to detect blinks, which can be used for recognizing activities. In addition to the same function, J!NS MEME enables us to sense eye movement based on electrooculography (EOG). The details of how to use and what kind of information we can obtain are explained at each application.

7.3 Fundamental Technologies

Before explaining the application technologies of the reading-life log, let us show some basic technologies and building blocks of the reading-life log.

7.3.1 Basic Character Detectors and Recognizers

Recent character recognition methods for scanned business documents can provide satisfactory recognition performance, by huge research efforts in a long history from

the Tauschek patent (1829). In fact, commercial OCR software is now very common and bundled with scanning machines and document viewers, such as Adobe Acrobat.

In contrast, character recognition for text captured in photographs is still a difficult task. This task is so-called scene text recognition and many researchers are still tackling it. Possible reasons that make scene text recognition difficult are as follows: various font designs, especially decorated fonts, complex backgrounds, various illumination conditions, and non-frontal camera angles. In addition to these difficulties in "recognizing individual characters," another, more serious, difficulty lies in "detecting scene texts." Although scene text detection is no difficulty for human beings, it is still very difficult for computers; even state-of-the-art techniques cannot achieve an f-ratio of more than 90 [2, 3].

We have tried to develop breakthrough techniques for these tasks, i.e., scene text detection and scene text recognition. In later subsections, the techniques will be detailed. It is worth noting that development of techniques and their results are valuable for detection and recognition tasks for more general visual objects. Characters should be the easiest subject of detection and recognition tasks because they have been designed artificially and revised for error-less communication for thousands of years. Accordingly, characters are one of the best subjects for observing the fundamental performance of individual detection and recognition methodologies.

7.3.1.1 Trials of Scene Text Detection

As noted above, scene text detection is still an open problem and an unavoidable problem for scene text recognition systems. Various detection methods have been proposed so far [4, 5]. The most typical approach to scene text detection is to discover certain image features that can discriminate text parts from non-text parts. For example, we examined two features, i.e., color uniformity and edgeness [6], because it is possible to assume that the same color is used for printing a letter (or even a word) and each letter is separable from its background by a sharp edge contour. In [6], it was found that edgeness is better than color uniformity in detection accuracy.

Another approach is the so-called multiple hypothesis, where multiple features and detectors are used and the multiple detection results are then finally combined into a single detection result. This approach is reasonable because there is neither an "almighty" feature nor a detector that can deal with huge variations of scene texts. Even if a text is not detected by color uniformity, it will be rescued by other features, such as edgeness, and vice versa [7, 8]. Since this approach is very simple, it is possible to extend it in various directions. In particular, it is possible to use various methodologies for combining multiple results. In [8], the combination using global optimization is proposed and achieves top-level detection performance.

An important but overlooked concept of scene text detection is the "context" of scene text. Simply speaking, context is the surroundings of scene text and it gives a prior probability that text is inside of it. For example, since no text is in the sky, the probability of scene text is almost zero around a sky region. In this example, "sky" is the context giving a lower prior probability. "Tree" is also a context

for a lower probability whereas "signboard" is a context for a higher probability. Consequently, image-based scene understanding, or semantic segmentation, is very important for scene text detection, although research on semantic segmentation seems rather independent of scene text detection. In [9, 10], the usefulness of context in scene text detection is experimentally proved.

Visual saliency is also a good clue for detecting scene text. It is natural to assume that scene text is salient because the role of scene text is to give some textual message to the reader and this role is not fulfilled unless it catches the reader's eyes by the appearance of scene text. This assumption is positively supported by a large-scale experiment using various types of visual saliency [11, 12]. This fact allows us to use the value of visual saliency as prior to scene text like the context. The saliency assumed in [11, 12] is Itti's saliency, which is a computational model of general human visual psychology, and thus the result proves that scene context is also salient not only for computers but also for human beings. Note that several methods of evaluating visual saliency specialized for scene text have been proposed. The saliency according to these methods is very different from the original idea of saliency for visual psychology research.

7.3.1.2 Part-Based Methods for Scene Text Detection Recognition

Part-based methods are widely used for visual object recognition. The concept is to describe an entire image by a set of local regions. Information on each local region is encoded in a certain way and the encoded results of all local regions are aggregated into a single representation. Bag-of-Features (BoF) is the most famous part-based method. From an entire image, keypoints are first detected and the small local region around individual keypoint is then represented as a feature vector, such as SIFT and SURF. Roughly speaking, a keypoint is often detected around a corner or a region with complex texture and the feature vector captures some direction of the texture in the local region. Each feature vector is quantized and then voted into a histogram. This histogram-based aggregation of votes from all local regions is BoF. An advantage of BoF is its robustness against global deformation. This is because a visual object is represented as a set of its local regions in BoF and the global structure of the object is thus no longer preserved.

Part-based methods have rarely been utilized for character recognition. One possible reason is that any character is comprised of a single or multiple lines and its local structure cannot represent character class information. In other words, global structure is far more important than local structure for character recognition and it is thus anticipated that no part-based method can achieve reasonable recognition accuracy.

In spite of this negative anticipation, we were able to prove that handwritten digits can be recognized with more than 95% accuracy by a part-based method [13]. Our method is based on majority voting. Specifically, it recognizes individual parts by referring to a part dictionary with the nearest-neighbor approach. If we have 50 parts from a digit image, we will have 50 (local) recognition results. Then, the class that

becomes the most frequent recognition class among the 50 results is determined as the final recognition result.

The high recognition accuracy is achieved by the nature of majority voting. Imagine the recognition accuracy of individual parts is very low, say 30%. For a digit image of class "6," 30% of the votes go to "6" and the remaining 70% of the votes (by misrecognition) go to other classes. If misrecognition occurs randomly, each class (except for "6") will get $70/9 = 7.8\%$ of the entire votes and this is far less than 30%. This means that the correct class "6" will be selected as the class with the largest votes regardless of the low accuracy of 30%. This example suggests that the image of "6" will be misrecognized only in the case that "0" (or another class) could have more than 30% of the votes, and it is not so easy to incorporate so many votes into one wrong class.

Since it is proved that part-based methods are applicable to character recognition, we can now exploit their advantages. For example, we can recognize characters even if their global structure is severely destroyed by partial occlusion and decoration [13]. In addition, part-based methods will make the text detection process easier. Even if a character is detected incompletely, it still has a chance of correct recognition.

7.3.1.3 Character Recognition Under Low Image Resolution

Scene text is often very small in a camera-captured image. In fact, if a text is captured by a distant camera, its size in the photo tends to be small. Such small-sized text images are difficult to recognize without special treatment. Super-resolution is a possible remedy for this problem. As another remedy, we proved that the mutual subspace method can improve recognition performance dramatically [14]. This paper also shows that enhancement of the difference between resembling classes can improve recognition performance.

7.3.1.4 Character Recognition with a Larger Dataset

In visual object recognition research, the power of using a large dataset has been proved so far. A larger dataset requires larger computation resources and often human efforts in attaching the ground truth. Recently, the former point is relaxed by commercial GPGPUs and the latter point, by crowd sourcing. The larger a dataset becomes, the more precisely it can grasp the real distribution of patterns. Consequently, even the simple 1-nearest-neighbor method can achieve very high recognition accuracy with a large dataset.

A large dataset is, of course, very beneficial for character recognition. We have prepared about 1 million patterns of handwritten digits and machine-printed digits and then analyzed how they are distributed in a feature space and how the dataset size affects recognition performance. The results show that we need 10 times more reference patterns to halve misrecognitions of handwritten digits. Moreover, we performed a network analysis of the large dataset and found that machine-printed digits

have a dense-clustered distribution and that handwritten digits have a confusing (i.e., non-unimodal) distribution [15]. It is also possible to detect wrongly labeled patterns by using anomaly detection methods, if we have a sufficient number of data [16].

7.3.2 Construction of a Large-Scale Scene Character Dataset

In the history of pattern recognition research, datasets have played important roles. For example, in the research on Japanese offline handwritten character recognition, the ETL Character Database (http://etlcdb.db.aist.go.jp/) consisting of ETL-1 to ETL-9 [17] played important roles. However, due to the large cost of constructing a dataset, available public datasets were of small scale. Hence, as summarized in Table 7.1, we constructed five new datasets. So as to encourage character recognition research, we plan to make them publicly available unless this causes any copyright or privacy issues. The rest of the section is dedicated to introducing an overview of the datasets expected (1) in Table 7.1.

(1) Document image
This dataset was constructed by downloading PDF files available on the Internet so as to test the scalability of a camera-based document image retrieval method called locally likely arrangement hashing (LLAH) [18]. The downloaded PDF files were converted to produce their document images without any image distortion and stored in a database. LLAH was tested by using queries obtained by capturing printed version of the downloaded PDF files.

(2) Text in a camera-captured document
As shown in Fig. 7.3, document images captured with a camera suffer from perspective distortion, illumination change, blur, and so on. Since avoiding such degradation is difficult, recognition accuracy of characters and words in camera-captured document images has been far from satisfactory. A feasible solution is to collect many distorted real character and word images and use them to train a classifier. Though the solution requires ground truth of the collected images, manual groundtruthing requires laborious work and high cost.

Table 7.1 Constructed datasets and their scale

Contents	Scale
(1) Document image	100,000,000 pages
(2) Text in camera-captured document	1,000,000 words
(3) Scene text in still image	4,000 images, 25,000 words
(4) Scene text in video	55 videos, 500,000 words
(5) Scene text in video in Japan captured with omnidirectional camera	780,000 images, 790,000 words, 2,760,000 characters

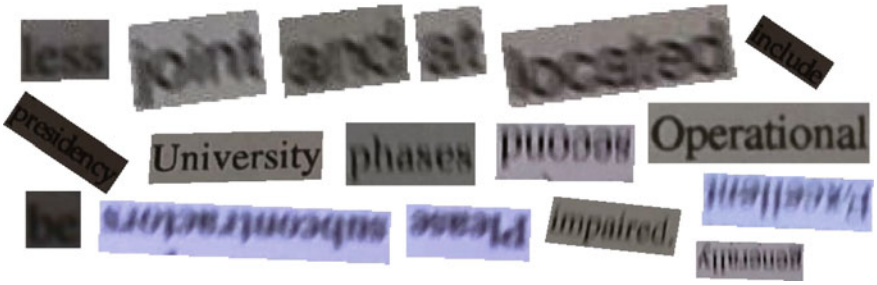


Fig. 7.3 Degraded labeled word images obtained from camera-captured document images

Hence, we propose an automatic groudtruthing method that enables us to construct a dataset by just flipping the pages of documents [19]. It utilizes LLAH, which makes it possible to find the page region corresponding to the captured document image from a large-scale document image dataset. This functionality can match a word in the captured document image with one in a PDF file. Thus, words in camera-captured document images can be groundtruthed based on text information contained in the corresponding PDF files. We have confirmed that the proposed method automatically groundtruthed 1 million word images with an accuracy of 99.98%.

(3) Scene text in a still image
In scene text recognition, datasets provided by the series of ICDAR Robust Reading Competitions [2, 3, 20–22] are used as the de facto standard. The Street View Text Dataset [23, 24] that collected text regions from Google StreetView is also often used. In such datasets, it is common that text regions of groundtruth are represented by bounding boxes of texts. This means that the bounding boxes contain not only text regions but also backgrounds, and they are not suitable for evaluating pixel-level character segmentation methods.
Thus, we created pixel-level groundtruth for the dataset of the ICDAR2003 Robust Reading Competition [20] and Street View Text Dataset [23, 24]. In addition, an original dataset consisting of 3,018 text images downloaded from Flickr was constructed with pixel-level groundtruth. They can be used not only for evaluating text detection and recognition methods but also for estimating the statistics of character pixels and background pixels [25]. Among these datasets, the ICDAR2003 Robust Reading Competition dataset was used in the ICDAR2013 Robust Reading Competition [2].

(4) Scene text in video
Conventionally, scene text recognition research treated scene texts in still images. Toward realization of human-harmonized information environment, however, we cannot ignore scene texts in videos recorded with wearable cameras and mobile devices. Thus, we constructed the first dataset of scene texts in video in collaboration with researchers at the Computer Vision Center (CVC) in Spain. The dataset consisted of 55 videos containing about 500,000 words regions

in English, French, and Spanish. The constructed dataset was used in a new challenge dedicated to scene text detection and recognition in videos in the ICDAR2013 and 2015 Robust Reading Competition [2, 3].

(5) Scene text in video in Japan captured with an omnidirectional camera
Most publicly available large-scale datasets only contain numerals and Latin characters. On the other hand, there was no dataset of Japanese characters including Chinese characters (kanji). Hence, in order to encourage development of detection and recognition methods for Japanese text, we constructed a large-scale Japanese scene text dataset [26]. In constructing the dataset, Point Grey Research Ladybug3, which is an omnidirectional camera equipped with six cameras, was used to capture movies of a scene of downtown Osaka. Among 780,000 images extracted from the captured videos, 31,000 images were manually groundtruthed. As a result, 910,000 text regions containing 790,000 words and 2,760,000 characters were obtained. The numbers of words and characters were almost four times those of the Street View House Numbers Dataset, known as the largest public dataset, consisting of 630,000 characters of 10 digits [27], while the unique numbers of words and characters were much fewer because they were extracted from consecutive images of videos. Since the images were extracted from videos, the constructed dataset can also be regarded as a dataset about "scene text in video" like the dataset explained in (4) above.

In addition to the constructed datasets mentioned above, we introduce two attempts.

- Automatic groundtruthing
  As available data are expected to increase and larger datasets are demanded, it is important to develop labor-saving ways of groundtruthing. Hence, we attempted automatic groundtruthing of scene texts [28]. First, a classifier was trained with a limited number of labeled data. Using the classifier, data without labels were groundtruthed. Then, the groundtruthed data were used to further train the classifier. Repeating the process, more labeled data and a better classifier were expected to be obtained. We confirmed that it worked at least for a small dataset.
- Data synthesis by font generation
  Since shapes of scene texts are diverse, training a classifier using various fonts is effective in improving character recognition performance. However, collecting many characters in various fonts is not easy. Thus, we propose an automatic font generation method that estimates the character shapes of unknown fonts. This is described in the next section.

7.3.3 Automatic Generation of Fonts

There are two main reasons for generating various kinds of fonts. One is to utilize impression of or additional information on the font. We see characters written in various fonts in our daily life. The impression we receive or the amount of information

differs according to the font. An appropriate font should be chosen considering situation, purpose, circumstance, etc. A typical example is the usage of "universal fonts." Universal fonts are designed to be recognized by various persons and in any environment. Many fonts used in our daily life have been replaced by universal fonts. However, designing a universal font is steady and hard work that requires enormous time and cost.

The other is the contribution to the improvement of character recognition accuracy. Any method for character recognition is fundamentally based on pattern-matching technology, and having various font data directly leads to improvement of character recognition accuracy. However, collecting various fonts requires enormous time and effort. If automatic generation of fonts is available, it will become possible to generate a large number of various character patterns.

For these reasons, if a character font can be generated automatically, it is expected to contribute to both improving communication of man and machine and the performance of machines. However, as far as we know, there are few researches on automatic generation of fonts, and there is no system or software for designing fonts automatically.

In this section, we introduce an attempt to generate fonts automatically using the technique of rearranging patches [29]. Given a small number of sample fonts, all the character images are generated. This method applies the patch transform [30]. The patch transform breaks an image into small patches and generates a modified image by rearranging them under a certain constraint. The target of this method is a natural image. The arrangement is defined as an optimization problem considering that there is no unnaturalness or inconsistency as an image.

Based on this idea, the proposed method generates a character image by breaking the given font patterns into small patches and rearranging them. Since character images are binary, there is a problem whereby the continuity of an image is lost near the boundary of patches and it easily becomes unnatural, an issue that should be resolved.

Figure 7.4 gives an outline of the proposed method. First of all, some character images of a specific font are given as sample patterns. Then, skeleton data of characters that should be generated are given. Character patterns are generated by rearranging the patches obtained from the sample patterns along the skeleton data. In the example of Fig. 7.4, in order to make the character image of "E," the images of "T" and "F" are chosen, and the image of "E" is generated with two patches of "T" and 33 patches of "F." Along with the skeleton of "E," appropriate patches are placed in order to cover the skeleton with the black pixels of patches. Considering the global structure of a character, the patches with similar shape context [31] features are arranged. All these conditions and the similarities between the adjacent patches are represented by a cost function, and the optimal solution is obtained by the belief propagation.

Figure 7.5 displays examples of the fonts generated automatically by the proposed method. Figure 7.5a shows the original font, and Fig. 7.5b shows the generated character images. We used 26 capital alphabetical letters. To generate a character image, five characters except for the target character are randomly selected from the original

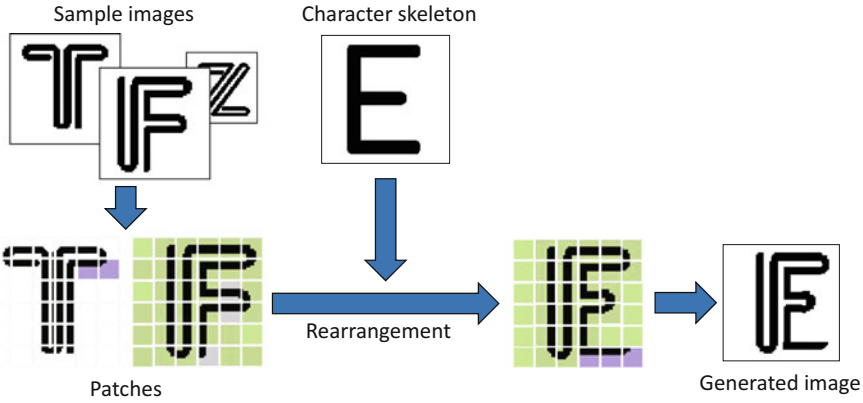
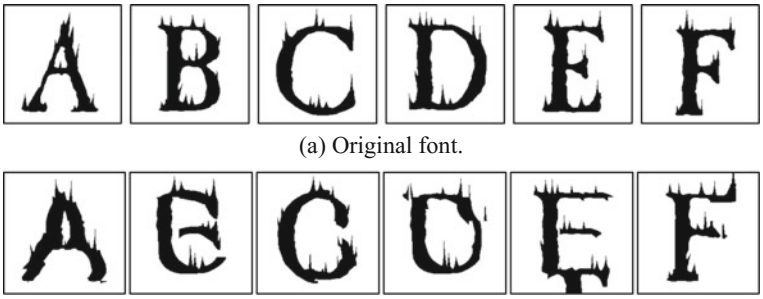


Fig. 7.4 Generation of fonts by rearranging patches



(a) Original font.

(b) Generated font.

Fig. 7.5 Example of generated fonts

font images. This figure shows that it is possible to generate character patterns that have the characteristics of the original character font only by breaking the images into patches and rearranging them. However, there are many unnatural portions and there is a room for improvement.

7.3.4 Real-Time Character Recognition

For a human-harmonized information environment, a machine is required to understand human intention and provide necessary information in a timely fashion. Hence, a machine should be able to recognize things as quickly as humans do. However, in the history of character recognition research, while people were interested in the processing time required for recognition with regard to convenience for practical use, it has never been explicitly aimed to realize real-time processing of recognition.

One reason is that recognition accuracy is considered to be more important than processing time. In general, processing time and recognition accuracy are in a trade-off relationship. Hence, reducing processing time means nothing other than reducing recognition accuracy. However, if we can reduce a long processing time with minimal reduction in recognition accuracy, such a character recognition technique will be very useful.

We propose two real-time character recognition methods. One is for recognizing alphabetical characters and numerals, and the other, for recognizing Japanese characters. Both were realized in the lazy learning framework; features extracted from training data were stored in the database in advance, and then recognition was performed by a fast similarity search that obtained features from the database similar to those extracted from the query. The fast search was realized by an approximate nearest-neighbor search (ANNS) technique, where ANNS does not guarantee that search errors do not happen so as to greatly reduce the computational time. Performance of ANNS methods is evaluated by search accuracy, processing time, and memory consumption. We propose a practically fastest ANNS method to realize certain accuracy, at least as of the time the paper was submitted [32].

The recognition method for alphabetical letters and numerals assumes that character regions are segmented by an established way such as binarization by a threshold, and segmented characters are recognized quickly in a way robust against perspective distortion [33]. As a recognition method robust against geometric transformation such as perspective transformation, geometric hashing [34] is known well. Letting N be the number of features in an image, the method requires computational cost of $O(N^4)$ so as to make the method robust against affine distortion. On the other hand, the proposed method greatly reduces the computational cost down to $O(N^2)$ by using geometric invariants in a different way from usual. As a result, we succeeded in running a camera-based character recognition method on a laptop computer in real time (more than 10 frames per second). The advantage of the method is that accuracy and recognition speed are not affected by change of layout of characters, character size, or camera angle; it works on images taken even from an elevation angle of 45°. So as to improve the method, a spell checker is integrated [35]. Use of the spell checker is effective in recognizing word images taken from an elevation angle of 20°; recognition accuracy of some words has been increased from 40 to 98%.

The recognition method for Japanese characters can recognize characters freely laid out on a complex background as shown in Fig. 7.6 [36]. The method uses local features such as SIFT (scale-invariant feature transform) [37] extracted from character images, as is often used in object recognition, to detect and recognize characters. The method is good at recognizing complex characters including Chinese characters; an experimental result shows that recall of 97% and precision of 98% are achieved. It runs on a laptop computer at about one frame per second. Since a large part of computational time is occupied by extraction of SIFT, to avoid this burden for speeding it up, we have changed it to an anytime algorithm [38]. The anytime algorithm is an algorithm that can output a calculation result at any time and a better result can be obtained as more time is spent. Introducing the feature of the anytime algo-

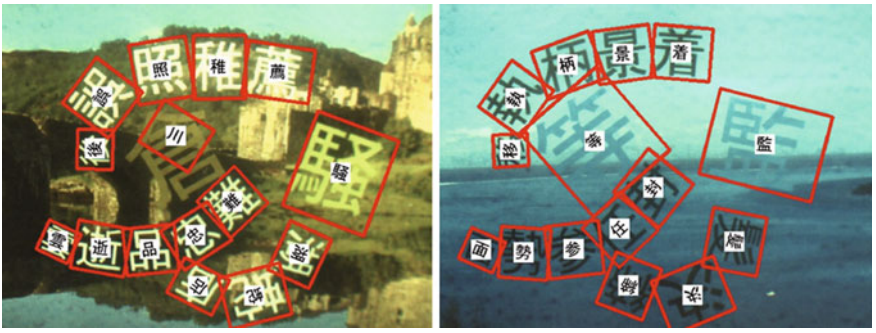


Fig. 7.6 Recognition result of the character recognition method for Japanese characters. The *red rectangles* represent the detected regions of characters and character images put in the rectangles recognition results

rithm makes it possible to realize flexible recognition where the recognition results of easier characters to recognize are obtained earlier and those of difficult characters, later. Figure 7.7 shows a comparison of the conventional method (not an anytime algorithm) and the proposed method (an anytime algorithm) with regard to recognition accuracy and processing time. Though the conventional method outputs the recognition result only all at once, the proposed method outputs it four times and more characters are recognized at each output. As a result, the proposed method can recognize 11 out of 14 characters earlier than the conventional method, though the processing time required for obtaining all the results increases. The proposed method has also been improved to cope with the problem of inaccurate estimation of the pose of the character of interest in the case that fewer local features are extracted from

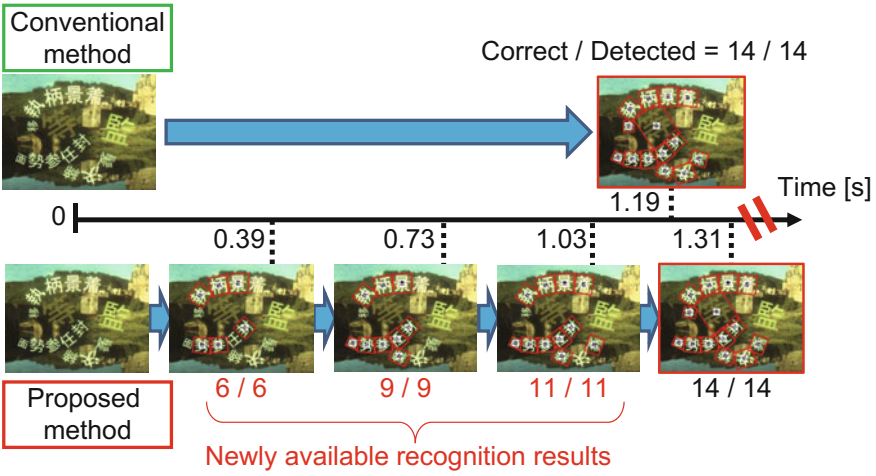


Fig. 7.7 Anytime algorithm of the Japanese scene text recognition method

a character region. The problem mainly arises when recognizing simpler characters such as hiragana and katakana. To avoid this problem, we propose a novel method that allows a robust estimation of the affine transformation matrix [39].

7.3.5 Omnidirectional Character Recognition

Omnidirectional character recognition is a process of recognizing all the characters in a 360-degree scene image. Unlike most of the existing methods assuming that the text areas are roughly detected or texts are included in the image, the purpose of omnidirectional recognition is to realize recognition without pointing. This technique enables us to support the discovery and offering of information that the user needs or the user has overlooked, supporting visually impaired persons, etc.

An omnidirectional image is obtained by an omnidirectional camera. Omnidirectional cameras are roughly classified into two types: using a spherical mirror and using multiple cameras. We selected the latter type considering the resolution of the acquired images. We used the camera called Ladybug3 of Point Grey Research. This camera includes five cameras arranged horizontally and one camera for the upper direction. All of the cameras are progressive-scan CCDs that can acquire 1600 by 1200 pixels at 15 frames per second.

An image acquired by an omnidirectional camera has a large number of pixels. On the other hand, the size of each character tends to be small. Therefore, in general, text detection requires time and the recognition accuracy is low. Moreover, since it operates outdoors, not only real-time processing but also robustness is required. In order to develop a system that offers the information necessary for a user, it must operate in real time and recognition accuracy must be high. To fulfill these conditions, methods based on template matching and edge detection are examined.

7.3.5.1 Template-Matching-Based Method

This method is based on the case-based method using the template-matching technique. In omnidirectional recognition, since the capturing environment is uncontrollable, it is important to cope with deteriorated character images. Therefore, two methods for low-resolution character recognition are considered. One is a template-matching method for recognizing low-resolution character images using high-resolution template images. The other is a technique of creating high-resolution images from many low-resolution images. In addition, we attempt to develop local features that are effective for low-resolution images. As a result, it was verified that character recognition can be achieved if the font is known and there is no geometrical distortion.

7.3.5.2 Edge Detection-Based Method

For detecting texts from scene images, it is known that edge information and binarization using color and intensity play complementary roles [40]. However, in the omnidirectional recognition task, rapidity of processing is important and the binarization process takes much time. We analyzed the processing time for each process and developed a fast detection method. First, edges are extracted from the acquired image. Then, candidate text areas are detected using the knowledge that the edges in text areas are strong, dense, and have various directions [40]. Then, binarization and labeling in the candidate area are performed to detect texts. Figure 7.8 shows an example of text areas detected by an omnidirectional camera. We constructed a system that deals with several frames per second without using special processors such as GPUs.

The detected characters are usually too small to be recognized by ordinary character recognition methods. Therefore, we developed a method based on the subspace method using the whole image of each character [14]. Exploiting the images obtained from multiple frames, a subspace that represents a character is constructed. The character is recognized by the similarities between the subspace and the subspaces constructed from training data. Figure 7.9 displays an example of character recognition.

7.3.6 Real-Time Document Image Retrieval

Document image retrieval is the task of retrieving the corresponding document image from the database in response to a query given as a document image. The query is often produced by using a camera, and it undergoes geometric distortion, blur, and

Fig. 7.8 Text detection by an omnidirectional camera

Fig. 7.9 Result of character recognition

variation of illumination. As a result, it is very different from the images stored in the database. Thus, the retrieval method should be capable of handling such distortions.

One may say that it is less meaningful to retrieve document images because documents are at hand when queries are captured. However, we have many applications such as provision of services that are associated with a specific part of a document image. For example, augmented reality is an easy-to-understand application of this technology.

Figure 7.10 illustrates an overview of this technology. On the left, a camera-captured query is shown. On the right of the figure, the retrieved document image is shown. The retrieval is based on the word-to-word matching indicated by the straight lines between two images. The technology is called locally likely arrangement hashing (LLAH) [19], which is still known as a state-of-the-art method in terms of robustness, accuracy, speed, scalability, and applicability. This technology is applicable to any script because it is independent of language and script. It just takes into account the distribution of centroids of connected components. It enables

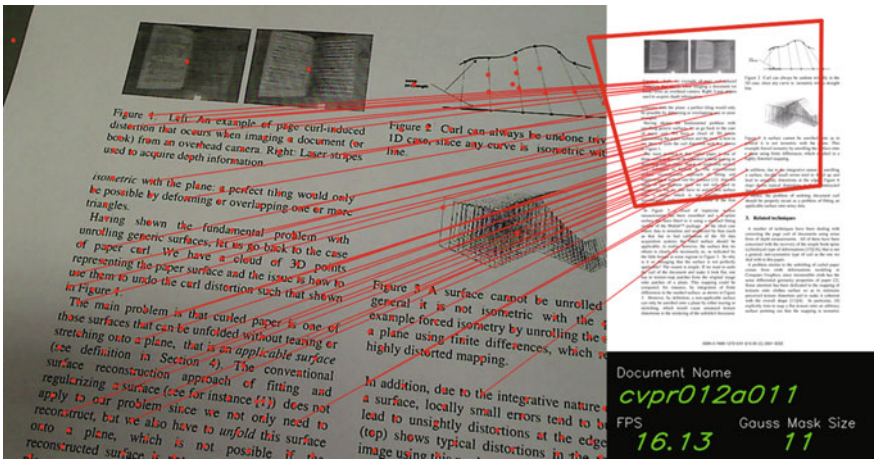


Fig. 7.10 Document image retrieval by LLAH

us to search a database of 100 million pages with an accuracy of 98.7% in 26.8 ms/query.

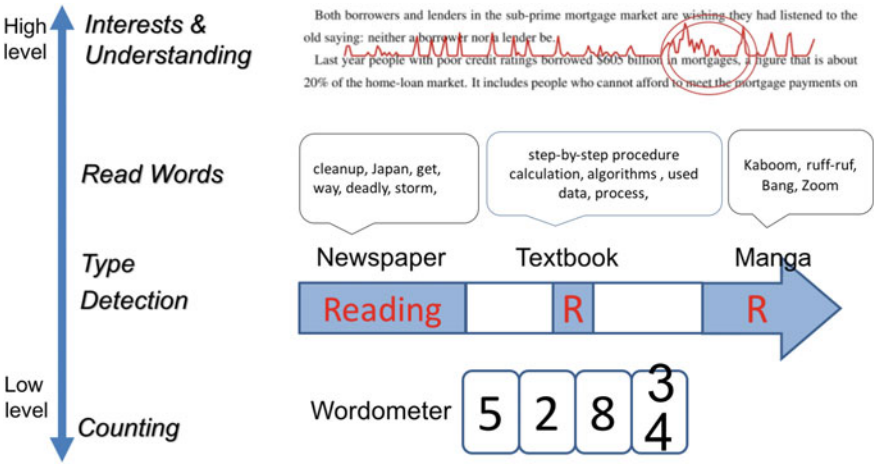


Fig. 7.11 Reading-life log for documents

7.4 Reading-Life Logs

In this section, we introduce various reading-life logs implemented by using the fundamental technologies.

7.4.1 Reading-Life Log for Documents

We have various functions that fall into this category of reading-life log. Figure 7.11 shows its overview that is characterized by what is to be measured: quantity is shown in the lower part of the figure and quality, in the higher part of the figure.

The simplest case is to measure purely the quantity of reading in terms of the number of read words. It is called a "wordometer" named after the pedometer. As compared to the pedometer, which measures the physical activity of the user, the wordometer quantifies the cognitive activity of the user. Another simple case is to measure the quantity of reading in terms of time. The function called reading detection detects reading activities among daily activities of the user. To be precise, it can output from when to when the user has read. A more content-oriented implementation of the reading-life log is "document type recognition," which classifies what type of document the user is reading. We also have the functionality of logging all read words, as well as logging the level of understanding and proficiency in the language as the highest (quality) level of the reading-life log. To implement the above functions, we employed the various devices shown in Fig. 7.2. In the following, details of each function will be explained.

7.4.1.1 Wordometer [41]

The wordometer has been implemented in many different ways. The simplest is just to measure the time taken to read a text. By multiplying his/her average speed of reading, the number of read words can be estimated. A better estimation can be obtained by combining document image retrieval LLAH and a wearable eye-tracker. The scene image captured by the eye-tracker is used as a query for LLAH to find which document the user is reading. This allows him/her to know the average number of words per text line. In addition, by analyzing the eye-tracking data as shown in Fig. 7.12, a long regression can be detected as a line break that can be used to estimate the number of read lines. By multiplying the average number of words per line and the number of read lines, we can estimate the number of read words. An even better estimation can be obtained by using a more sophisticated estimation. In our method, support vector regression is used to estimate the number of words, from eye-tracking data and the average number of words per line.

Another version of the wordometer using J!NS MEME EOG glasses is implemented. This will be described in the next Sect. 7.4.2 including more details about other implementations of the wordometer.

7.4.1.2 Reading Detection

Reading detection is the task of distinguishing reading activities from other activities. Head motion and blink frequency can be used for this purpose [42], because when reading, we blink much less with a specific head motion. Image features and eye-tracking data [43], as well as EEG signals [44] can also be used for reading detection. These are also based on the specificity of reading in terms of eye-tracking data and EEG signals.

7.4.1.3 Document Type Recognition

Document type recognition is to classify a document read into one of the predetermined classes of documents such as textbook, newspaper, novel, fashion magazine, or manga. By using this functionality, we are able to summarize reading activities

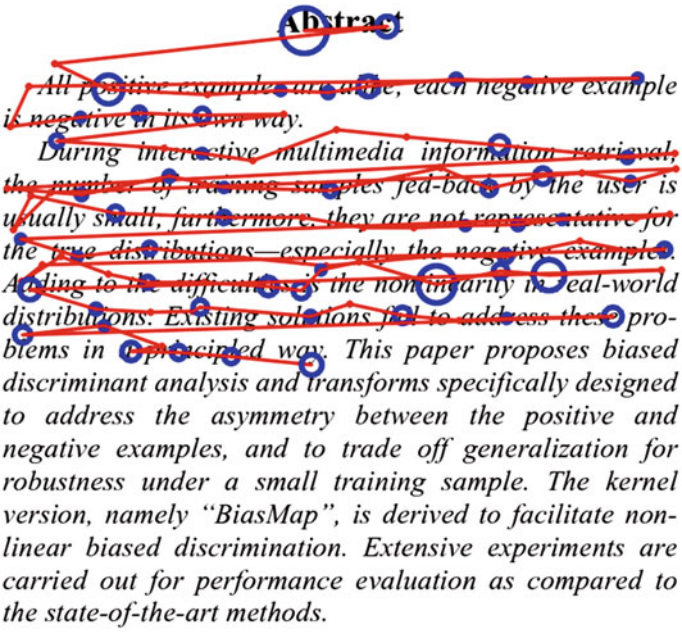


Fig. 7.12 Wordometer with an eye-tracker

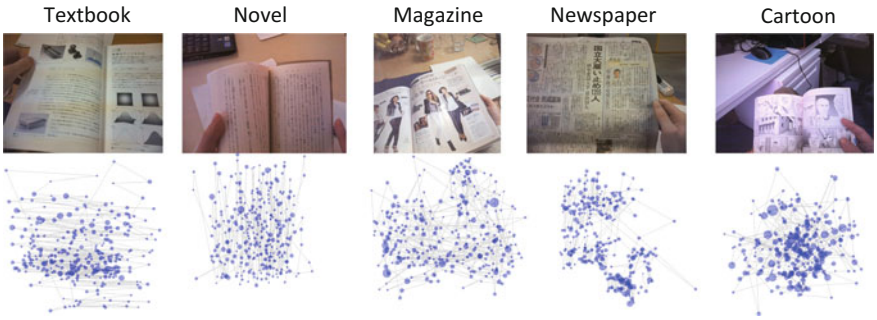


Fig. 7.13 Document type recognition based on the distribution of fixations and saccades

for each class of documents, for example, how many pages of a textbook you have read in a day.

Similar to other functions such as reading detection, we are able to recognize document types in many different ways. For example, recognizing document types is insignificant if document image retrieval is applicable. In the case that it is unavailable, we are still able to tackle this problem.

One way is to apply an object recognition technology. The Bag-of-Words representation of local features allows us to capture visual differences of document categories and thus to recognize the type [45]. Another way is to use eye gaze data [46]: the distribution of fixations and saccades. They definitely reflect the document layout as shown in Fig. 7.13 and can thus be employed for recognition. Surprisingly, an accuracy of 99% can be achieved by user-dependent training. An interesting counterexample of this approach is that if a document is not read in a typical way, it can be misclassified. For example, we asked subjects to read a fashion magazine and, for some male students, their gaze data are confused to be "textbook" because they really "read" the fashion magazine like reading a textbook. For other subjects with successful classification, their typical way of reading it was to browse the contents by skimming the text and looking at the pictures. We also attempted to use EEG signals for the recognition [44]. It is also possible to achieve similar accuracy as the case of eye gaze data. It is surprising that, by only looking at the EEG signals, we are able to recognize which type of document the user is reading.

7.4.1.4 Recording of Read Words

In addition to recognizing document types, we are able to log reading activities in more detail if we can record what the user has read. For this function of "recording of read words," it is necessary to associate eye gaze data with the contents of the document. Broadly speaking, there are two possible ways to realize this: retrieval-based and recognition-based methods.

The retrieval-based method associates eye gaze data with electronic contents of a document using the coordinates of fixations. When using stationary eye-trackers, the screen coordinates of fixations can be associated with the displayed contents. For the case of mobile eye-trackers, on the other hand, it is necessary to employ document image retrieval to achieve the association.

If there is no error in the eye gaze data, it is possible to know which word the user is looking at. However, in many cases, we are not able to avoid error. Even after careful calibration, a vertical error of about a few lines and a horizontal error of a few characters are unavoidable. Thus, what we can do is to estimate possible read words assuming that the error distribution is, for example, Gaussian. Based on this way of estimating possible read words, we are able to build a log in the form of Bag-of-Words (BoW) for a certain period of time [47]. Figure 7.14 represents a tag cloud representation of such BoWs. Another representation would be to build read paragraphs or read pages, which are with much fewer errors because of their larger area.

The recognition-based method is to obtain word information by applying character recognition to the images obtained by a scene camera. Although this approach is more error-prone compared to the retrieval-based approach, it allows us to log all word information not only on documents but also other scene text. This will be explained later in Sect. 7.4.3.

7.4.1.5 Estimating the Level of Understanding and Language Proficiency

Estimating the level of understanding and language proficiency is at the highest (quality) level of reading-life log for documents. Generally speaking, estimating how much a user understands is not an easy task because we do not have any means to evaluate the level of understanding. In order to make the task tractable, we limit the application field to an English standardized test called TOEIC (Test of English for International Communication). TOEIC is widely used in Asian countries including Japan as well as some Western countries. The test consists of two sections: listening and reading. We focus here on the reading section. In particular, we employ questions for Part 7 (reading comprehension) of the reading section. Based on the TOEIC test, we define the level of understanding as the number of correct answers to the questions in Part 7. In general, four questions are associated with a single long text. Thus, the number of correct answers ranges from zero to four. On the other hand, the language proficiency is estimated as the TOEIC score, which ranges from 10 to 990.

To estimate the number of correct answers, we employed fNIRS as the sensor. We measured the distribution of oxyhemoglobin on the forehead of subjects and analyzed its change while reading text. We attempt to solve this problem as a three-class problem, in which the following three classes are defined: (1) all four questions are correctly answered; (2) three questions are correctly answered; or (3) two or fewer questions are correctly answered. After training in a user-independent way, we have achieved an accuracy of 80%.

Fig. 7.14 Tag clouds generated from readers' reading behavior. As the number of read paragraphs increases, tag clouds become richer. It is also interesting to know the difference in tag clouds from different readers

To estimate language proficiency, we used SMI eye-trackers RED250 (250 Hz) and ETG (eye-tracking glasses) to analyze the behavior of reading text [48]. The first trial estimated English proficiency by analyzing eye gaze data when reading text [49]. We defined the task as three-class problem about TOEIC scores: (1) 400 or less; (2) more than 400 and less than 600; or (3) more than 600. By using a classifier trained in a user-independent way, we have achieved an accuracy of 91%. The second trial estimated the TOEIC score directly from eye gaze. We used ETG to obtain eye gaze data not only for the reading text part, but also for questions, and eye movement between text and question parts. As a result of applying trained regression, we were able to estimate the TOEIC score with an average error of 36 points after obtaining eye gaze data for 10 documents [50].

7.4.2 Smart Eyewear

To implement the reading-life log, we explored many different possibilities and used different systems from stationary setups towards wearable devices. Yet, regarding any life log, body-worn devices are a natural fit, as the user carries them. They can sense and recognize the environment from a first-person view of the user. This holds in particular for head-worn appliances, as humans perceive most of the world by sensing situated on the head. Therefore, the head seems to be a natural position for the reading-life log as well as cognitive assistance systems in general.

There are many interesting brain-sensing technologies like functional near-infrared spectroscopy (fNIRS) that can help to understand cognitive activities beneficial for the reading-life log, from cognitive load to indication of concentration and comprehension.

However, early head-mounted sensing devices are rather bulky, expensive, and socially stigmatizing. Recent commercial smart eyewear devices, such as Google Glass and PUPIL eye-trackers, lay the foundations for "eyewear computing." Fig. 7.15 shows the development from brain sensing and optical eye tracking to head-mounted computers (e.g., Google Glass) and smart glasses (e.g., J!NS MEME).

Regarding the social impact of reading-life log technologies, we wondered how to make them more attractive to the general public. For the first prototypes, we started collaborating with J!NS on J!NS MEME. MEME is lightweight and looks like ordinary glasses. Our main issue in implementing reading-life log technologies is that we are lacking cameras (egocentric as well as eye-tracking) as their setup is still too bulky. They require heavier processing and battery power. It is still not possible to equip easy-to-wear smart eyewear with such cameras. MEME focuses on electrooculography for eye movement tracking. As the eye is a dipole, we can use electrodes to recognize eye movements. Using this technique, we can recognize reading behavior in everyday scenarios [51]. As MEME shows (see Fig. 7.16), the electrodes can be unobtrusively embedded in glasses. For these reasons, we focus on reading detection and other reading habit-related research using these unobtrusive smart glasses.

Fig. 7.15 From brain sensing, through eye-tracking glasses, to unobtrusive smart eyewear

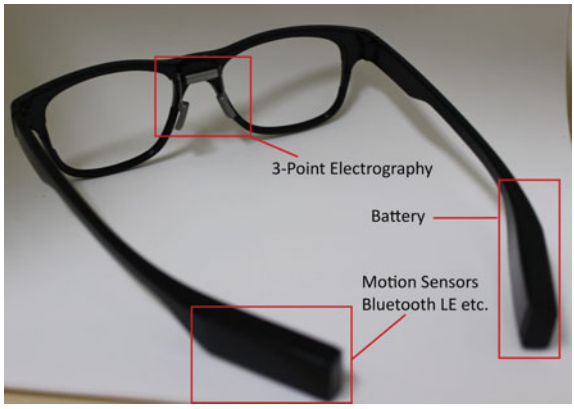


Fig. 7.16 NS MEME, smart eyewear using electrooculography and motion sensors

7.4.2.1 The Wordometer—Counting How Many Words You Read

Although reading is very well explored in the cognitive sciences and psychology, we still know very little about healthy reading habits. There are only few researchers tackling reading in real-life circumstances. Increased reading volume is associated with greater vocabulary skills and higher general knowledge as well as improved critical thinking. Smart eye-glasses are perfect for detecting reading activity and also quantify how many words a person reads.

As a first goal, we set out to just detect reading or not reading utilizing eye movement analysis.

Using optical eye tracking (SMI mobile glasses 2.0) we can recognize reading with very high accuracy (over 95%) in semi-controlled setups with around 30 users (Japanese students, age 20–27) [41, 51]. Porting a similar reading algorithm to J!NS MEME, we remain at around 85% accuracy for semi-controlled setups [52].

Not only can we detect when a user is reading but also the approximate amount. The idea behind the recognition is simple. Reading is detectable because of a relatively steady head position and repetitive forward-backward (or up-down) eye movements. These eye movements also include backwards saccades due to line breaks. We detect these backward saccades and get over their approximate length and their frequency the number of words a user read. The wordometer algorithm on the optical tracker now works with an error rate of 9% (std. 3%). On J!NS MEME, we are currently at 20% (std. 5%) [51]. There is room for improvement, yet comparing it to a step counter, we are in a similar accuracy range as the wrist-worn step counters available. It is still difficult to define what good reading habits are, yet the first tools for quantifying the amount of reading can help with this.

Also, for other reading-life log technologies, the wordometer provides useful information (e.g., distinguishing whether a user has actually read a sign or piece of text as opposed to just glancing over it). Yet, more important are the implications of the wordometer regarding the reading habits explored in the next section.

7.4.2.2 Quantifying Reading Habits with Smart Eyewear

Although we are increasingly aware of how important reading is for learning, it is difficult to get people to read more, especially as other types of more easily digestible content increase (e.g., videos). As tracking physical activities (e.g., step counts) can motivate users to be more active, we believe this also applies to reading. However, it is still difficult to define what healthy reading habits are, as we are lacking methods to quantify reading.

So tracking the words people read can not only motivate them to read more and improve their vocabulary and critical thinking skills, but also give initial insights into healthy reading habits. Having a measurement for speed, timing, and reading volume is a first step to exploring the cognitive life of users with the ultimate goal of improving learning.

7.4.3 Reading-Life Log for Scene Text

In our daily life, we read not only texts on paper documents but also scene texts, such as texts on signboards, price tags in supermarkets, labels on bottles and boxes, menus in restaurants, traffic signs, texts on displays, etc. Similar to the reading-life log for documents, it is possible to realize another reading-life log system for scene texts. If we can make a log of texts that we read in a scene, it will provide the reading-life log with another value. As suggested by the above examples, scene texts are often related to some object or location or activity, whereas document texts are not. Consequently, the reading-life log for scene texts will be a record of our activity and interaction with various objects, rather than a knowledge log.

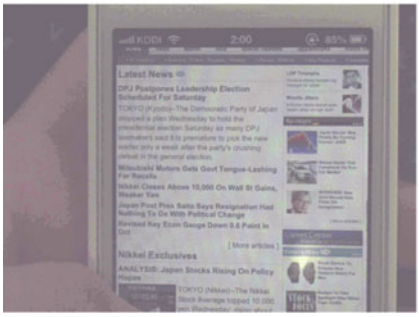
To realize the reading-life log for scene texts, we can no longer fully rely on document retrieval techniques. This is simply because it is not practical to register all scene texts into the system. In particular, scene texts are often not static, that is, they are changeable dynamically. Therefore, we need to take the most straightforward solution: the development of an accurate scene text detector and recognizer.

We also encounter the practical problem of choosing a video camera for the system. (Note that it should not be a still camera but a video camera because the purpose of the reading-life log is unconscious and continuous capture of textual information, like typical life-logs.) There are many requirements of the video camera of the reading-life log system for scene texts. The camera should be compact and light, since we need to carry it attached to the body. In addition, it needs to have high resolution to capture each character with a reasonable size (say, more than 5050 pixels) without approaching the target text with the camera, high shutter speed to avoid motion blur, and an auto-focus function or deeper focus to deal with texts at various distances.

Since the target text is captured into multiple frames by a video camera, it is necessary to unify the same text in multiple frames. There are two approaches to this unification. The first approach is so-called video mosaicing, or video stitching, that combines video frames into a large single image while dealing with overlapping parts among the frames. A scene text detector and recognizer is then applied to the large image. The second and more practical approach is text-level integration. In this approach, a scene text detector and recognizer is applied to individual frames and text recognition results are then integrated while unifying the same text in different frames [53]. Figure 7.17 shows the result of this second approach. Text in the captured video is moved by scrolling but the system can achieve a unified result with good success.

7.4.4 Document Annotations

As a way of analyzing contents based on reading activities, we focus here on document annotations. Documents are annotated manually or automatically based on the



(a) A frame capturing texts on a smartphone display.

Latest News

DPJ Postpones Leadership Election
Scheduled For Saturday
TOKYO (Kyodo)--Tlie Democratic Party of Japan
drojjpGd a p!an Wednesday to hold the
prcsfdenHai election Saturday as many DPJ
i iawrnafers s'^cTIi is premature to pick Ihe new
I leader only a week after the party's crushing
I defeat in the general election.
¥ Mitsubishi Motors Gets Govt Tongue-Lashing
jFor Recalls
I Nikkei Closes Above 10,000 On Walt St Gains,
Weaker Yen
Japan Post Pres Saito Says Resignation Had
j Nothing To Do V/ilh Political Change
j Rovised Key Econ Gauge Down 0.6 Point In
Oct
[More articles]
Nikkei Exclusives
I ANALYSIS: Japan Stocks Rising On Policy
SHopcs
TOKYO {Nikkei)-The Nikkei
Stock Average topped 10.000
yen Wednesday, rising about
1,400 points in the last month

(b) Integration result of text recognition results on multiple frames.

Fig. 7.17 Reading-life log for scene texts

user's reading activities and possible services to be provided. We would like to show one example of each type of annotation. We also show an example of a simple way of showing annotation by using a mobile eye-tracker and a head-mounted display. An example of manual annotation is a system based on document image retrieval. As devices for annotation, we employ mobile phones and Google Glasses as shown in Fig. 7.18. Taking as input a picture of a document, the corresponding document is searched in the database. Various annotations such as text, image, highlight, voice, and video are supported to be put on a document. When using a mobile phone, a part

Fig. 7.18 Putting and displaying annotations by using Google Glass

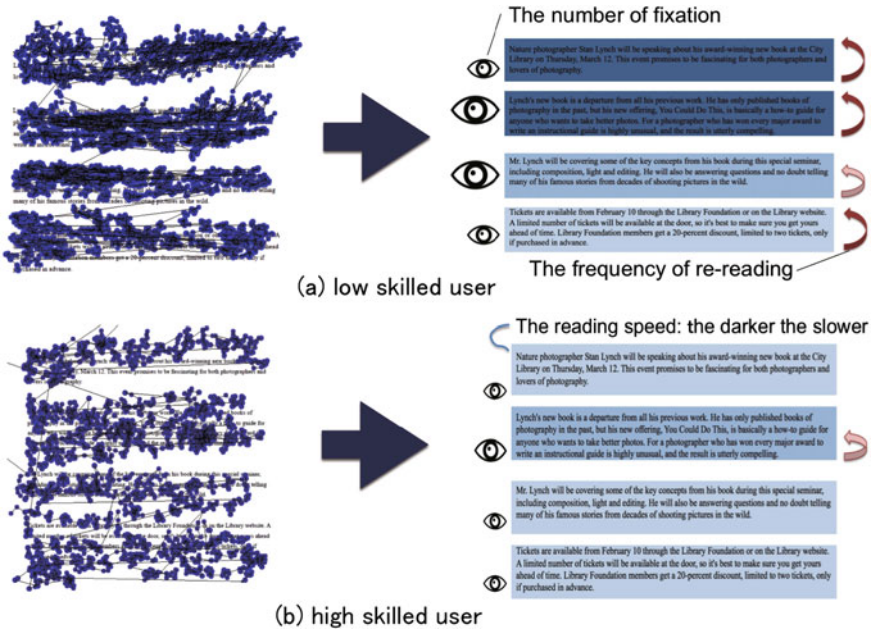


Fig. 7.19 Automated annotation by using eye gaze information

of a document at which the annotation is put can be specified by dragging that part. Retrieval of annotation is straightforward; as soon as the device captures a document, it can be used as a query and display retrieved annotations if they exist. If there is no corresponding entry of document in the database, the camera-captured document can be stored as a new entry at which the user can put annotations.

As an example of automatic annotation, we introduce here a visualization of the user's behavior while reading a document. For learners of foreign languages, it is useful to know which parts of a textbook are difficult to understand. For teachers, it is fruitful to know who is having trouble at what location in a textbook, and which parts in general learners have trouble with. In order to obtain clues for the above information, we focus here on the reading speed, the number of re-readings, and the number of fixations [54]. The parts that are read slowly, with many re-readings and fixations, indicate parts that the reader has difficulty understanding. An example of visualization is shown in Fig. 7.19. Figure 7.19a represents the behaviors of a novice learner, while Fig. 7.19b shows those of a skillful learner. In the figure, the slower the reading speed is, the darker a paragraph is. Similarly, the higher the number of re-readings is, the darker the arrow is. The number of fixations is displayed as the size of the eye icon. From the figure, it is easy to see that there is a clear difference between novice and skillful learners. For novice learners, not only by reviewing their performance visualized in this way, but also by comparing their behavior with other, more skillful learners allows them to motivate themselves to keep learning.

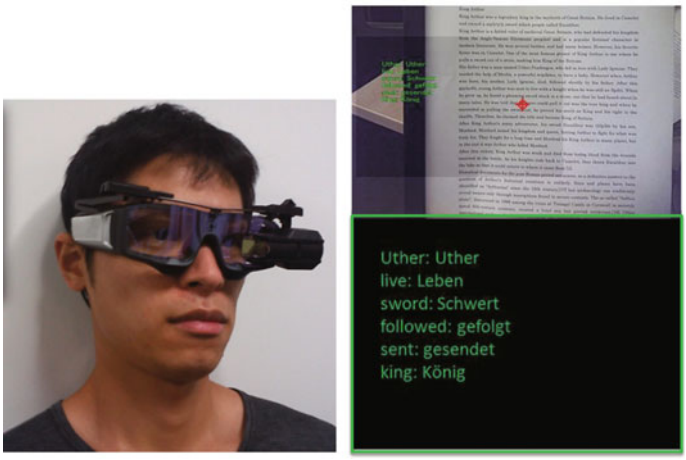


Fig. 7.20 Display of annotation by using a mobile eye-tracker and a head-mounted display

Let us move on to an example of displaying annotations. A representative method for displaying annotations is the research called Text 2.0 [55]. This method is to replay manually annotated contents based upon the reader's reading activities. A concrete use case is as follows. The reader is reading a document on a display with an eye-tracker. When the reader is reading a specific part with which a service is associated, it is provided. Examples of such service are sound effects and dictionary lookup. A more advanced method we developed along this line will be described in Sect. 7.4.6. One disadvantage of Text 2.0 is that the service is provided only on the displayed document. This problem can be solved by using a mobile eye-tracker with a head-mounted display [56]. A camera-captured document with the reader's eye gaze is given to the document image retrieval LLAH, in order to obtain information about the part of the document the reader is reading. After that, a mechanism similar to Text 2.0 is employed. Figure 7.20 shows the overall system and an example of dictionary lookup based upon the reader's eye gaze.

7.4.5 AffectiveWear

In addition to straightforward annotations related to eye movements (saccade speed, fixation count), which can give some indication of reader proficiency and interest [54], we can also record facial expressions with the text [57]. For this purpose, we implemented AffectiveWear, smart glasses that detect the distance of the skin from the glass frame using proximity sensors. With this technology, we can detect up to eight different facial expressions. Figure 7.21 gives a brief overview of the system and the types of facial expressions it can detect.

This system and similar approaches can give indications about the emotional state of the reader. Authors can get feedback regarding whether a specific text evokes the intended effect. One can get classifications about especially funny, sad etc. paragraphs and so on.

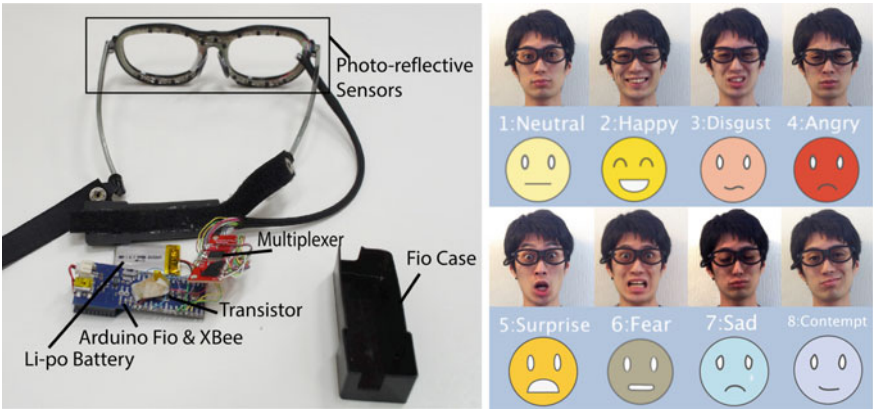


Fig. 7.21 AffectiveWear, detecting facial expressions using distance sensors between the skin and glasses frame

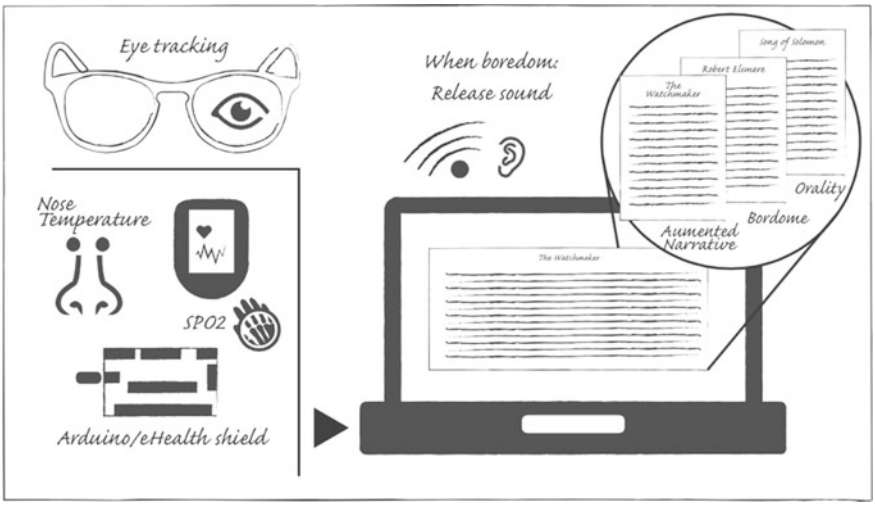


Fig. 7.22 Augmented narrative: using physiological signals as feedback mechanism for books triggering audio and haptic interactions

7.4.6 Augmented Narrative

Augmented narrative uses bio-feedback (nose temperature, eye blink, eye movement, heart rate) in a text-body interaction for a more immersive experience [58]. Figure 7.22 gives an overview about the augmented narrative concept. In augmented narrative, the input of the data detected by the sensors is understood to be that of the mental workload of the reader. The system is then set to distinguish whether the reader is bored, frustrated, or misunderstanding the story or whether the reader is in a state of flow and therefore engaged with the storyline. For example, when reading the story of Little Red Riding Hood, the system will know that the reader should be immersed in the story's climax when Little Red Riding Hood arrives at her grandmother's house, not realizing that, in her grandmother's bed, lies not her grandmother but the wolf. If the system finds no input of engagement when reading through this climatic part of the story, meaning there is no stimulation, then it infers that the reader needs extra-textual content and will release a sound, for example, the wolf's claws scratching the bed. Overall, augmented narrative intends to reconcile orality and literacy bringing them together in order to present the transmission of culture with the best of both worlds, by giving the reader a multimodal perception of events, as in oral cultures, that help the mental simulation of a story, in a narration whose meaning is conveyed in writing.

With a first prototype of the augmented narrative, we showed that we are able to detect engagement in a story using nose temperature and eye movements and that, in a further step, we can increase engagement by giving haptic and audio stimuli when the user loses interest [58].

7.4.7 Future Directions

In the future we would like to extend our research toward the following directions.

A promising application of reading-life log for scene text is to help disabled people such as visually impaired people and people with dyslexia. The technologies of real-time scene text recognition allows us to read text in the environment to give those people the ability of "reading." The most difficult part for implementing this service is its interface: how to display the results of recognition. If the machine reads all text in the scene and display them as sounds, the information overflows and annoying. A goal directed and/or spatially separated display of recognized results must be incorporated.

Another important application of reading-life log for scene text is memory aid. Not only for dementia people but also for healthy people, it is not always easy to remember everything needed for their life. Our technology can amplify the memory by recording all of the read text with indexes of time, place and context. People can search what they have read to remember things. Combination of this technology with

an intelligent interface of finding what to search enables us to augment the human memory. In other words, the user is connected to his/her reading-life.

The reading-life log for documents also has its future directions. One way is just to extend the current direction toward learning help. For the current target English as a foreign language, finding real problems of users and give them recommendations about what to do next are important task. We can also think of extending the application area from English to other subjects. We hope in the near future that the machine can help students to find their weak and strong points at each subject in order to motivate them to learn more.

An ultimate application of reading-life log for documents is to estimate the knowledge-level of users. One of the authors, Koichi Kise still remembers the comment made by the late Professor Naomi Miyake, who had been an advisor of our CREST. She told us that if we are able to record all what have been read, it is not necessary to have entrance examinations of universities. She thinks that the level of knowledge can be described by looking at his/her record of reading. Although such information is quite personal and may be problematic to record, it has a big impact of knowing people's abilities and interests. With a careful treatment of such information in terms of ELSI (Ethical, Legal and Social Issues), we believe that our life can be enriched, and our abilities can be extended by our technologies.

7.5 Conclusion

"You are what you have read." Based on the notion of this phrase discussed in the Introduction, we have implemented a variety of functions of the reading-life log using various sensors. Based on the implemented functions, readers are described in terms of the quantity and the quality of what they read. At the same time, documents and signboards are characterized by how they have been read and by whom. We do hope that this research will open a new field of research on readers and materials for reading.

We still have several things to do to make our technologies available to the public. One important aspect is to verify their effectiveness by conducting a larger user study. Another important point is how to use the reading-life log. A possible future direction would be to build an "actuator" to change a reader's behavior based on the facts found by the reading-life log.

References

1. K. Kise, S. Omachi, S. Uchida, M. Iwamura, A trial for development of fundamental technologies for new usage of character and document media. J. Inst. Electron. Inf. Commun. Eng. **98**(4), 311–327 (2015)

2. D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. Gomez, S.R. Mestre, J. Mas, D.F. Mota, J.A. Almazan, L.P. de las Heras, ICDAR 2013 robust reading competition, in *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR 2013)* (2013), pp. 1484–1493
3. D. Karatzas, L. Gomez, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V.R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, E. Valveny, ICDAR 2015 robust reading competition, in *Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR 2015)* (2015), pp. 1156–1160
4. S. Uchida, *Text Localization and Recognition in Images and Video, Handbook of Document Image Processing and Recognition* (Springer, 2014)
5. Q. Ye, D. Doermann, Text detection and recognition in imagery: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **37**(7), 1480–1500 (2015)
6. Y. Matsuda, S. Omachi, H. Aso, String detection from scene images by binarization and edge detection. IEICE Trans. Inf. Syst. (Japanese edition), **J93-D** (3), 336–344 (2010) (in Japanese)
7. R. Huang, P. Shivakumara, Y. Feng, S. Uchida, Scene character detection and recognition with cooperative multiple-hypothesis framework. IEICE Trans. Inf. Syst. **E96-D** (10), 2235–2244 (2013)
8. H. Takebe, S. Uchida, Scene character extraction by an optimal two-dimensional segmentation. IEICE Trans. Inf. Syst. (Japanese edition) (D), **J97-D** (3), 667–675 (2014) (in Japanese)
9. Y. Kunishige, Y. Feng, S. Uchida, Scenery character detection with environmental context, in *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR 2011)* (2011), pp. 1049–1053
10. A. Zhu, R. Gao, S. Uchida, Could scene context be beneficial for scene text detection?Pattern Recognit. **58**C, 204–215 (2016)
11. A. Shahab, F. Shafait, A. Dengel, S. Uchida, How salient is scene text? in *Proceedings of the 10th IAPR International Workshop on Document Analysis Systems (DAS2012)* (2012), pp. 317–321
12. R. Gao, S. Uchida, A. Shahab, F. Shafait, V. Frinken, Visual Saliency Models for Text Detection in Real World. PLoS one **9**(12), e114539 (2014)
13. S. Wang, S. Uchida, M. Liwicki, Y. Feng, Part-based methods for handwritten digit recognition. Front. Comput. Sci. **7**(4), 514–525 (2013)
14. S. Toba, H. Kudo, T. Miyazaki, Y. Sugaya, S. Omachi, Ultra-low resolution character recognition system with pruning mutual subspace method, in *Proceedings of the 2015 International Conference on Consumer Electronics—Taiwan (ICCE-TW)* (2015), pp. 284–285
15. M. Goto, R. Ishida, Y. Feng, S. Uchida, Analyzing the distribution of a large-scale character pattern set using relative neighborhood graph, in *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR 2013)* (2013), pp. 3–7
16. M. Goldstein, S. Uchida, A comparative study on outlier removal from a large-scale dataset using unsupervised anomaly detection, in *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods (ICPRAM2016)* (2016), pp. 263–269
17. T. Saito, H. Yamada, K. Yamamoto, On the data base ETL9 of handprinted characters in JIS Chinese characters and its analysis. IEICE Trans. Inf. Syst. (Japanese edition) (D), **J68-D**(4), 757–764 (1985) (in Japanese)
18. T. Nakai, K. Kise, M. Iwamura, Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval, in *Document Analysis Systems VII*, vol. 3872, Lecture Notes in Computer Science, (2006), pp. 541–552
19. S. Ahmed, K. Kise, M. Iwamura, M. Liwicki, A. Dengel, Automatic ground truth generation of camera captured documents using document image retrieval, in *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR 2013)* (2013), pp. 528–532
20. S.M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto, H. Miyao, J. Zhu, W. Ou, C. Wolf, J.-M. Jolion, L. Todoran, M. Worring, X. Lin, ICDAR 2003 robust reading competitions: entries, results and future directions. Int. J. Doc. Anal. Recognit. (IJDAR) **7**(2–3), 105–122 (2005)

21. S. Lucas, ICDAR 2005 text locating competition results, in *Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR2005)*, **1**, (2005), pp. 80–84
22. A. Shahab, F. Shafait, A. Dengel, ICDAR 2011 robust reading competition challenge 2: reading text in scene images, in *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR 2011)* (2011), pp. 1491–1496
23. K. Wang, S. Belongie, Word spotting in the wild, in *Proceedings of the 11th European Conference on Computer Vision (ECCV2010), Part I* (2010), pp. 591–604
24. K. Wang, B. Babenko, S. Belongie, End-to-end scene text recognition, in *Proceedings of the 13th International Conference on Computer Vision (ICCV2011)* (2011), pp. 1457–1464
25. R. Gao, F. Shafait, S. Uchida, Y. Feng, A hierarchical visual saliency model for character detection in natural scenes. Camera-Based Document Analysis and Recognition, LNCS **8357**, 18–29 (2014)
26. M. Iwamura, T. Matsuda, N. Morimoto, H. Sato, Y. Ikeda, K. Kise, Downtown osaka scene text dataset, in *Proceedings of the 2nd International Workshop on Robust Reading (IWRR2016)* (2016) (in printing)
27. Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A.Y. Ng, Reading digits in natural images with unsupervised feature learning, in *Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning* (2011), p. 9
28. M. Iwamura, M. Tsukada, K. Kise, Automatic labeling for scene text database, in *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR 2013)* (2013), pp. 1397–1401
29. H. Saito, Y. Sugaya, S. Omachi, S. Uchida, M. Iwamura, K. Kise, Generation of character patterns from sample character images, in IEICE Technical Report, PRMU2010-287 (2011) (in Japanese)
30. T.S. Cho, S. Avidan, W.T. Freeman, The patch transform. IEEE Trans. Pattern Anal. Mach. Intell. **32**(8), 1489–1501 (2010)
31. S. Belongie, J. Malik, and J. Puzicha, Shape context: a new descriptor for shape matching and object recognition, in *Advances in Neural Information Processing Systems* (2000), pp. 831–837
32. M. Iwamura, T. Sato, K. Kise, What is the most efficient way to select nearest neighbor candidates for fast approximate nearest neighbor search? in *Proceedings of the 14th International Conference on Computer Vision (ICCV 2013)* (2013), pp. 3535–3542
33. M. Iwamura, T. Tsuji, K. Kise, Memory-based recognition of camera-captured characters, in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems (DAS2010)* (2010), pp. 89–96
34. Y. Lamdan, H.J. Wolfson, Geometric hashing: a general and efficient model-based recognition scheme, in *Proceedings of the 2nd International Conference on Computer Vision (ICCV1988)* (1988), pp. 238–249
35. N. Asada, M. Iwamura, K. Kise, Improvement of word recognition accuracy with spellchecker based on tendency of recognition error of characters, IEICE Technical Report, **110**(467), PRMU2010-268, pp. 183–188 (2011) (in Japanese)
36. M. Iwamura, T. Kobayashi, K. Kise, Recognition of multiple characters in a scene image using arrangement of local features, in *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR 2011)* (2011), pp. 1409–1413
37. D.G. Lowe, Object recognition from local scale-invariant features, in *Proceedings of the International Conference on Computer Vision* (1999), pp. 1150–1157
38. T. Kobayashi, M. Iwamura, T. Matsuda, K. Kise, An anytime algorithm for camera-based character recognition, in *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR 2013)* (2013), pp. 1172–1176
39. T. Matsuda, M. Iwamura, K. Kise, Performance improvement in local feature based camera-captured character recognition, in *Proceedings of the 11th IAPR International Workshop on Document Analysis Systems (DAS2014)* (2014), pp. 196–201
40. X. Liu, J. Samarabandu, An edge-based text region extraction algorithm for indoor mobile robot navigation, in *Proceedings of the 2005 IEEE International Conference on Mechatronics and Automation*, **2**, (2005), pp. 701–706

41. K. Kunze, H. Kawaichi, K. Kise, K. Yoshimura, The wordometer—estimating the number of words read using document image retrieval and mobile eye tracking, in *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR 2013)* (2013), pp. 25–29
42. S. Ishimaru, J. Weppner, K. Kunze, A. Bulling, K. Kise, A. Dengel, P. Lukowicz, In the blink of an eye—combining head motion and eye blink frequency for activity recognition with Google Glass, in *Proceedings of the 5th Augmented Human International Conference* (2014), pp. 150–153
43. Y. Shiga, T. Toyama, Y. Utsumi, A. Dengel, K. Kise, Daily activity recognition combining gaze motion and visual features, in *PETMEI 2014: 4th International Workshop on Pervasive Eye Tracking and Mobile Eye-based Interaction, Proceedings of the 16th International Conference on Ubiquitous Computing* (2014), pp. 1103–1111
44. K. Kunze, Y. Shiga, S. Ishimaru, K. Kise, Reading activity recognition using an off-the-shelf EEG—detecting reading activities and distinguishing genres of documents, in *Proceedings of the12th International Conference on Document Analysis and Recognition (ICDAR2013)* (2013), pp. 96–100
45. Y. Utsumi, Y. Shiga, M. Iwamura, K. Kunze, K. Kise, Document type classification toward understanding reading habits, in *Proceedings of the 20th Korea-Japan Joint Workshop on Frontiers of Computer Vision*, **3**, (2014), pp. 11–17
46. K. Kunze, Y. Utsumi, Y. Shiga, K. Kise, A. Bulling, I know what you are reading: recognition of document types using mobile eye tracking, in *Proceedings of the 17th Annual International Symposium on Wearable Computers* (2013), pp. 113–116
47. O. Augereau, K. Kise, K. Hoshika, A proposal of a document image reading-life log based on document image retrieval and eyetracking, in *Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR2015)* (2015), pp. 246–250
48. K. Kunze, H. Kawaichi, K. Yoshimura, K. Kise, Towards inferring language expertise using eye tracking, in *CHI'13 Extended Abstracts on Human Factors in Computing Systems* (2013), p. 6
49. K. Yoshimura, K. Kunze, K. Kise, The eye as the window of the language ability: estimation of english skills by analyzing eye movement while reading documents, in *Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR2015)* (2015), pp. 251–255
50. H. Fujiyoshi, K. Yoshimura, K. Kunze, K. Kise, A method of estimating English skills using eye gaze information of answering questions of English exercises, IEICE Technical Report, **115**(24), PRMU2015-10, pp. 49–54 (2015) (in Japanese)
51. K. Kunze, K. Masai, M. Inami, Ö. Sacakli, M. Liwicki, A. Dengel, S. Ishimaru, K. Kise, Quantifying reading habits: counting how many words you read, in *Presented at the UbiComp'15: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (2015), pp. 87–96
52. S. Ishimaru, K. Kunze, K. Tanaka, Y. Uema, K. Kise, M. Inami, Smart eyewear for interaction and activity recognition, in *Presented at the CHI EA'15: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (2015), pp. 307–310
53. T. Kimura, R. Huang, S. Uchida, M. Iwamura, S. Omachi, K. Kise, The reading-life log—technologies to recognize texts that we read, in *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR 2013)* (2013), pp. 91–95
54. A. Okoso, K. Kunze, K. Kise, Implicit gaze based annotations to support second language learning, in *Proceedings of the 2014 ACM Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp2014)* (2014), pp. 143–146
55. R. Biedert, G. Buscher, S. Schwarz, J. Hees, A. Dengel, Text 2.0, in *Proceedings of the 28th ACM Conference on Human Factors in Computing Systems (CHI2011)* (2011)
56. T. Toyama, W. Suzuki, A. Dengel, K. Kise, User attention oriented augmented reality on documents with document dependent dynamic overlay, in *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR2013)* (2013), pp. 299–300

57. K. Masai, Y. Sugiura, K. Suzuki, S. Shimamura, K. Kunze, M. Ogata, M. Inami, M. Sugimoto, AffectiveWear: towards recognizing affect in real life, in *Presented at the UbiComp/ISWC'15 Adjunct: Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers* (2015), pp. 357–360
58. S. Sanchez, T. Dingler, H. Gu, K. Kunze, Embodied reading: a multisensory experience, in *Presented at the CHI EA'16: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (2016), pp. 1459–1466