Word and Sentence Extraction Using Irregular Pyramid

Poh Kok Loo[1] and Chew Lim Tan[2]

[1]School of the Built Environment & Design, Singapore Polytechnic,
Singapore 139651
[2]School of Computing, National University of Singapore,
Singapore 117543

Abstract. This paper presents the result of our continued work on a further enhancement to our previous proposed algorithm. Moving beyond the extraction of word groups and based on the same irregular pyramid structure the new proposed algorithm groups the extracted words into sentences. The uniqueness of the algorithm is in its ability to process text of a wide variation in terms of size, font, orientation and layout on the same document image. No assumption is made on any specified document type. The algorithm is based on the irregular pyramid structure with the application of four fundamental concepts. The first is the inclusion of background information. The second is the concept of closeness where text information within a group is close to each other, in terms of spatial distance, as compared to other text areas. The third is the "majority win" strategy that is more suitable under the greatly varying environment than a constant threshold value. The final concept is the uniformity and continuity among words belonging to the same sentence.

1 Introduction

In today's environment, document that one has to read and process is growing at an exponential rate. The requirement to be able to extract and summarise text information on a document with speed and accuracy is increasingly important. Although most of the information has already been digitized, the transformation of the imaged document's text content into a symbolic form for reuse still remains as a very active research area [1]. Before the text content can be recognized by an OCR system and converted into a symbolic form, it must be extracted from the imaged document. There have been many studies about text extraction. Some focuses on the extraction of characters [2], a few works directly on the word level [3], [4] and others direct their attention to layout analysis [5]. Our algorithm will perform the extraction task starting from the word level and progress into the layout analysis to extract sentences. Some techniques make use of a model to guide the extraction where the specific type of document under process is known in advance. Others will base on just the raw input data, which is what our algorithm will do. Due to the complexity of the problem and the vast variety of methods, it is not easy to categorize all methods. Nevertheless, in our context we will categorize them into non-pyramid and pyramid techniques. The majority of the proposed methods fall under the non-pyramid techniques.

D. Lopresti, J. Hu, and R. Kashi (Eds.): DAS 2002, LNCS 2423, pp. 307–318, 2002.

Most non-pyramid techniques perform detailed spatial analysis of the text area. Assumption about the physical spatial property of the image is required. In most cases, text images must be aligned or grouped in a homogeneous direction (i.e. horizontally or vertically). Most of them require the application of further skew correction technique [3], [5] before the content can be extracted properly. The statistically based approach proposed by Wang, Phillips and Haralick [4] require the inter-glyph and inter-word distances in its probability computation that are horizontally aligned. The splitting and merging technique as proposed by Wong, Casy and Wahl [6] requires text information to be separable in the horizontal or vertical direction. Others rely on the detailed analysis of the inter-component spacing. The labelling algorithm [7] also requires the alteration in the image horizontal scale to facilitate the extraction.

In the pyramid category, the majority of the proposed methods make use of the regular pyramid structure. Most of these studies require connected component analysis. A strong assumption of disjoint components is needed to ensure correct extraction of text images [8]. In our algorithm, no connected component analysis is required. The aggregation of pixels into characters and character into words is done through the natural grouping of pixels and regions. This proposed algorithm has no assumption in terms of the size, font, orientation and layout of text images. Although the regular pyramid shares the same benefit as in irregular pyramid with the ability to carry out image abstraction in achieving reduced computation cost and permitting local analysis of image features, it suffers from its rigid contraction scheme and there exists the shift dependent problem [9]. Due to the effect of the fixed decimation function (eg. summation of 4 pixels into 1) in a regular pyramid the coarse representation of the image will be distorted if the original image is shifted. For irregular pyramids, in particular, the stochastic pyramid as proposed by Kropatsch [10], only local information is required in its decimation. Large decimation ratio is obtained and thus results in a faster pyramid construction time. The shift dependence problem no longer exists, since its structure is flexible enough to match with the input content. The content of the image will control the aggregation process. To date there are only a handful of studies on irregular pyramids. Some address the issues of increasing the efficiency in building the pyramid through the reduction of pyramid levels [11]. Others use the pyramid structure to perform region segmentation [12], [13], edge detection [11], or connected component analysis [7]. No direct attempt is made to use the irregular pyramid to extract logical text group from a text image in existing works. This paper will propose an algorithm based on the irregular pyramid to perform text extraction. This is the result of our continued work on a further enhancement to our previous proposed algorithm [14].

2 Fundamental Concepts

There are a few basic concepts that our algorithm relies on. They are the inclusion of background information, concept of “closeness”, density of a word region, majority “win” strategy and the directional uniformity and continuity among words in a sentence. The following subsections describe these concepts in detail.

2.1 Inclusion of Background Information

This is one of the unique features of our algorithm. Unlike most of the proposed methods, we include the background information in the processing. Background information are those white areas (ie. for a binary image) surrounding the actual black foreground where the actual text images are. Most researchers focus their attention only on the actual text image and discard the background information. In our algorithm, the main focus is still the foreground data, but the background information is also processed with a lower priority. Unlike the other methods where the strong assumption of horizontal alignment allows strict geometric analysis among text region, it is too complex or even not possible for our algorithm to base on this analysis to achieve our objective. In order to process text images of any size and orientation, we need the background region to guide the algorithm to perform the extraction. Clues about how various fragments of the text image are held together are in the background area.

2.2 Concept of 'Closeness"

We have discussed how English texts are formed and arranged in terms of the proximity among the characters and words but exactly what kind of distance is considered "close". If we are processing text with a common font, size and orientation, then perhaps it is possible to find a value in defining "closeness". Some papers [12], [15] have explored the possibility to compute such a value for their extraction activity with certain degree of success. Nevertheless, the method is quite restrictive in terms of the kind of document it can process. Once there is a considerable mixture of fonts or sizes and the orientation gets very irregular, than the likelihood of getting such a value become impossible. In our algorithm, instead of attempting to compute this value, we define a general concept of "closeness".

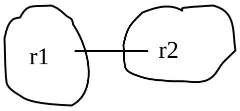

Fig. 1. 'Closeness" between two regions

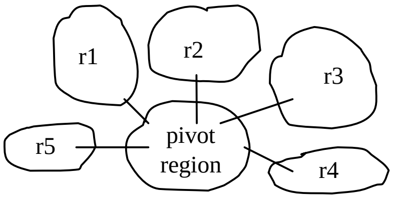

Fig. 2. "Closeness" among multiple regions

Two regions are considered "close" if they are next to each other. Multiple regions are considered "close" if they are "close" to a common pivot region. The pivot region is like a pulling force to attract all neighboring regions together. There is no computation of any distance. Figures 1 and 2 are the diagrams showing these two cases. No attempt is made to define what is "close" beyond the immediate surrounding. Regions just need to be present in the neighborhood to be "close".

2.3 Density of a Word Region

In our algorithm, a word region is defined as a collection of pixel points. It includes both foreground and background pixels. It is a regional area enclosing a complete word comprising of multiple characters. The mass of the region is defined as the total number of black pixels. The area of the region is defined as the total number of pixel points including both black and white pixels. The density of this region is computed as the mass over the total area of the region. This value indirectly reflects how much background information is used to enclose a complete word. A larger density value shows that the characters in the word are placed closer together. In contrast a smaller density indicates a loosely positioned character within the word. This density value is independent of the size, font and orientation of the text. To capitalize on this property, our algorithm has made use of the density value in two different ways. The first is used as a value to determine whether a word region has been formed or it is still a region holding word fragment. A complete word region is formed by many smaller regions. Each smaller region will hold different fragments of the word. As compared to a complete word region, the density of a region containing word fragments varies greatly among its neighboring regions. Such variation in density becomes a suitable condition to determine word formation. The second is used as a criterion to determine a correct word formation among a group of neighboring words. This leads us to our next concept.

2.4 Majority “Win” Strategy

Since our algorithm has no restriction to the kind of document it can process, the variation over the text image feature will vary greatly. The possibility to make global decision by using some constant factor becomes very low. There is simply no way to enforce a common condition that all can follow. Under such a scenario, the next best strategy is to get the majority agreement. If the majority of the members among the community under a process agree, then the members in question should agree also. This concept is implemented throughout our algorithm.

2.5 Directional Uniformity and Continuity among Words in a Sentence

Most English words exhibit the shape of an elongated region. The region will have a longer axis and a shorter axis in the direction perpendicular to the longer axis. Directional path of a word region is defined as the path along the longer axis. As we examine such a directional path of all words in a sentence, usually we can find uniformity in terms of the direction. All words in a sentence will follow the same direction. This is a common scenario observed in most text documents.

But there can be situations where there is no directional uniformity among words within a sentence. This frequently occurs in advertisements or posters where words are aligned in different orientations to have the artistic effect. Words in the same sentence can be positioned in different directions. Although uniformity has lost in this instance, there is still some form of continuity among the words belonging to the same sentence. Regardless of how artistic is the words’ arrangement, continuity among words will still exist to allow human reader to relate such group of words. Figure 3 shows two examples. In our algorithm, two words are considered continuous if the projections of their directional paths intersect. Regardless of where the intersection points are, as long as they lie within the image boundary they are considered valid. By basing on the property of uniformity and continuity, words can be grouped to form sentence.

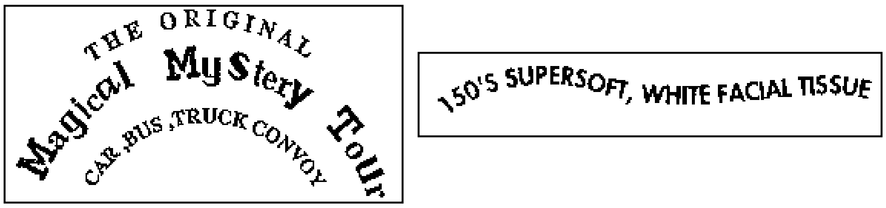

Fig. 3. Words uniformity and continuity in a sentence

3 Irregular Pyramid Structure

One key feature in our algorithm is the use of an irregular pyramid structure to represent the image content. In addition to just image representation, the algorithm uses this structure for its entire extraction process. Without the pyramid structure, some of our concepts cannot be realized. The present research makes use of an irregular variant, which is adapted from Kropatsch's irregular pyramid [11], [16].

The main use of a pyramid structure is to hold an image content. The structure is formed by a set of resolution levels. Starting from the base where the original image resides, each successively higher pyramid level will hold a transformed image content of a lower resolution. The image content of the lower pyramid level is condensed and represented by a smaller set of data point at the higher pyramid level. The resulting pyramid will be a structure holding successively coarser version of the image from the lower pyramid level till the pyramid apex. The construction of any pyramid structure involves three main components. They are the input image, a transformation function and an output image. An input image from a lower pyramid level is fed into a transformation function where the resulting output image is produced and placed on a higher pyramid level. The objective of the transformation function is to analyze and summarize the content of the input image and produce a smaller and yet representative set of image data point to form the output image. Unlike regular pyramid, the transformation function for an irregular pyramid structure is more complex. Instead of predetermining the number of data points to be transformed onto the next level, a decimation process is used to decimate un-qualified data points. Only data points that satisfy the criteria are promoted to the next level. Besides the decimation process, irregular pyramid also has two other processes that are not required in a regular pyramid construction. These are the explicit selection of children and neighbors. Due to the constancy nature in a regular pyramid transformation process that is known in advance, the children and neighbor can easily be determined. In an irregular pyramid, this must be maintained explicitly with the survivor. The number of children and neighbors that a survivor must maintain will vary.

In our algorithm, the objective is to extract a word group of any size, font or orientation. A region that encloses such a word group can be of any irregular size and shape. Our strategy is first to identify the potential center of a word group (ie. local maximal among the neighboring regions). With this central region as the survivor we assign the neighboring non-surviving regions (ie. fragments of the word group) to become its children. Another way to view this is to allow the center of the word group to become a pivot region to pull in all neighboring regions that are the fragments of the word group. In order to achieve this, the decimation and the claiming criteria in our algorithm are set as follows. The decimation criterion is to allow region with the largest surviving value (ie. local maximal) to survive and decimate all other regions. Each region is assigned with a surviving value equal to its own mass plus all its neighboring masses. The motivation is to allow a heavier region or a region with many mass neighbors to become a survivor. Such a region has a higher likelihood of being the center of a word group and thus become a better pivot region. As for the claiming rule, the criterion is to permit a region that has an unstable density with respect to its neighboring region to claim more children than a stable region.

4 The Algorithm

Our proposed algorithm is divided into two main sections. The first is for the extraction of word groups. The other is the concatenation of words into a sentence. Both processes are based on the same pyramid structure. In the algorithm, each data point on a pyramid level represents a region. Each individual region will maintain a list of attributes. The attributes are the area of the region, mass of the region, the children list, the neighbor list, a surviving value and the growing directional value. Figure 4 contains the pusedo code of the main algorithm for word extraction. As compared to our previous proposed method in [14], we have made revision to some of the old procedures and added a few new procedures. Below will only highlight the details of the revision and addition.

4.1 The Revision and Addition in the Word Extraction Process

In our previous algorithm there are two main stages in the child selection process. On pyramid level 0 and 1, a survivor will claim all neighboring non-survivors (ie. mass or non-mass region) if there is no other survivor claiming for the same non-survivor. If conflict arises, preference is given to the survivor with a larger surviving value. We call this ‘general’ claiming of neighboring regions. Starting from level 2 and onwards, a more elaborate child selection process is used and we call this the ‘special’ claiming. A survivor will first claim all neighboring regions with mass. A non-mass region will only be considered if it helps in the stability of the region’s density. The assumption is that on level 0 and 1 there is less possibility to locate a complete word group and thus should encourage more growing. The growing process will slow down from level 2 and onwards as the likelihood to locate a word group is higher.

After we have experimented with more test images we discover that this leads to the problem of “over growing” and “under growing” of the word region. “Over growing”

```
Create pyramid base level with (original image)
Pick survivors
Select children for each survivor
For (each pyramid level where
    -   the total number of pixel > 1 AND
    -   more word groups continue to form in the last pyramid level)
 {  Create pyramid higher level with (previously formed level)
    Update the survivor neighborhood list
    Assign pulling status (ie. general/special) to each region
    Adjust the pulling status (ie. “smoothing”)
    Assign surviving value to each region
    Pick survivors for the next higher pyramid level
    Select children for each survivor   }
```

Fig. 4. The word extraction algorithm

occurs when more than one correct word groups are merged. The region has over grown to include more than one word. This will usually happen when the word size is too small. On the other hand when the word size is too big, it will result in “under growing”. A region is “under grown” if it fails to enclose the entire word group. Fragments of the word are extracted as isolated regions. The key reason to the above problems is the timing in switching from “general” to “special” claiming. The purpose for “general” claiming is to bridge the spacing gap in between characters. As a survivor claims or pulls in non-mass blank region, it is using the blank region to grow outwards to bring more word’s fragment into the neighborhood. As a result when the size of the word is small, word region may have already been formed at a very early stage of the pyramid level. If we continue to allow “general” claiming, then the word region will continue to grow outwards and chances of growing into other word group will occur. This is the result of “over growing”. The scenario of “under growing” is the exact opposite. If the switch from “general” to “special” claiming or pulling is done too early (ie. before the word’s fragments are in the neighborhood), then fragments that belong to the same word may remain as an isolated region. Figure 5 shows an overgrown word group where all individual words are erroneously grouped together as one “word”. Figure 5 also shows an under grown word where part of the letter “i” and the letter “B” are detached from the word groups.

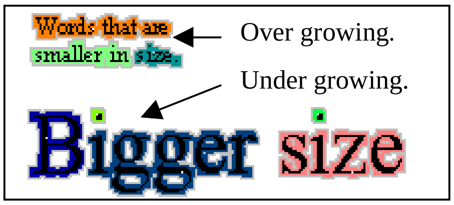

Fig. 5. Problem with over/under growing

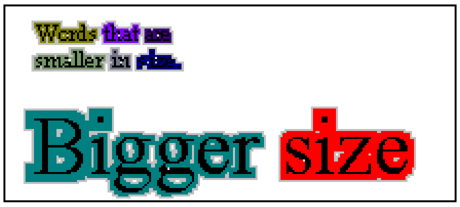

Fig. 6. Result with the amended algorithm

In order to solve this problem, the algorithm is amended to include a “pulling” status flag for each region. The flag will indicate whether a survivor should use "general" or "special" pulling (ie. pulling of neighboring mass or non-mass non-survivors) to claim its neighboring children. This will allow individual survivors to grow independently from each other depending on their own local situation. Instead of every one following a rigid global decision, the survivors will make their own local decision. In order to make the decision, the survivor will analyze its immediate neighbors. Density is again used as the factor for the decision. If the density of the region formed by merging the survivors and all its surrounding neighbors stay within an acceptable range of the average density of all its neighboring regions, then the region is considered stable and no further growing is required. In contrast, the survivor will continue to grow until the density reaches a stable level. This is achieved in the newly added function on line 7 of figure 4. Although with the above setting we have solved the "over/under growing" problem, a side effect occurs. As the algorithm allows individual survivors to grow at their own rate, the locality of growing becomes random. In order to ensure local growing consistency among neighboring regions, the algorithm is modified to include another processing step (ie. line 8). The purpose is to perform some "smoothing" over the pulling status of the neighboring survivors. A region will maintain its original pulling status if the majority of its neighboring regions also have the same pulling status. This will enforce nearby regions, usually belonging to the same word group, to have the same pulling status. Figure 6 is the result of the new algorithm.

4.2 Sentence Extraction Process

In order to assist the extraction of sentences, an additional attribute called the growing directional weight is added. This attribute is used to retain and reflect the growing path of a word region. It is an array of 8 entries containing the total mass in a specified growing direction. The growing direction is categorized into 8 segments. Just like the 8-connectivity direction, it comprises of top, top-right, right, bottom-right, bottom, bottom-left, left and top-left.

As a survivor is promoted to a higher pyramid level, it retains its original directional weight from the lower level. Using this set of weight as the base, it analyzes all the child regions. By taking the center of mass of the overall region (ie. all regions covered by the survivor) as the pivot point, the direction of where the child is located is computed. Each child is grouped under one of the directions mentioned above. The algorithm will now compare the directional mass inherited from the lower level by the survivor in a specified direction with the total mass of all children in that direction. It will retain the maximal value. As the algorithm progresses up the pyramid level, the directional weight attribute held by the survivor on the highest pyramid level will reflect the largest growing mass in the respective direction. A higher mass value reflects more growing. More word fragments are being pulled in from that direction.

Once the word extraction process stops when there is no possibility to find more word groups (ie. the same number of word groups on two consecutive levels), the extraction of sentence will begin. Figure 7 shows the main algorithm for the extraction of sentences. The new objective is to continue to grow the word region in order for words that belongs to the same sentence to merge as one bigger region. In another words the algorithm must allow words to grow into the correct neighboring regions for words belonging to the same sentence to merge. The algorithm will continue to grow a word region (ie. pull in more blank region), but only in 2 directions. They are the directions with the highest mass value in the directional weight attribute. It reflects that the formed word group is oriented along the 2 directions. There exist cases where no clear directional path can be found. This usually occurs in words that are very short in length (eg. in, of, is, etc). In this situation, the algorithm will examine the surrounding of such a word. The growing direction of the word is determined by the growing direction of the set of closest neighboring word region. The majority win concept is used. The word is assigned with the most frequently occurring growing direction among its neighboring word regions. If no maximum mode exists, the growing direction of the closest word region is used. Unlike word extraction where the closeness among regions is the immediate neighborhood (ie. two regions are next to each other), in sentence extraction all word regions are isolated with a distance apart. As a result the "closeness" definition is redefined as the shortest Euclidean distance between the boundaries of two regions.

```
For (each pyramid level where
    -   the total number of pixel > 1 AND
    -   (the first merging of word group has not occurred OR
    -   more word groups continue to form in the last pyramid level))
  {  Create pyramid higher level with (previously formed level)
     Update the survivor neighborhood list
     Assign pulling status (sentence) to each region
     Assign surviving value to each region
     Pick survivors for the next higher pyramid level
     Select children (sentence) for each survivor   }
```

Fig. 7. The sentence extraction algorithm

The task of growing a word region is to pull in more blank regions along the detected directions. Although the original 8 directional segments are used, further refinement is required. Problem will occur if we have the long word group as shown in figure 8. If the growing direction for the word group is on the left and right, by following the original directional segment all blank regions that are located in A and B will grow together. This is not desirable. Chances for this word group to grow into the wrong region (ie. up and down) are high. As a result, refinement is made in the algorithm to allow a more pointed and targeted growing direction. This will permit the growing of region B only.

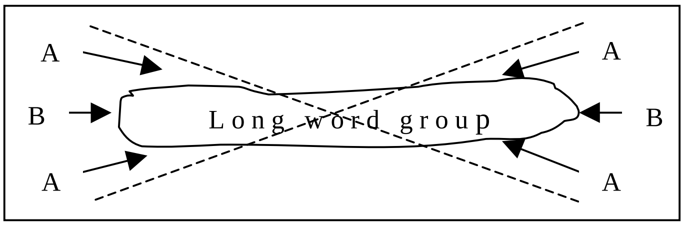

Fig. 8. Targeted growing direction

As we can see in the sentence extraction main algorithm in figure 7, it is almost the same as the word extraction process. The only changes are in the assignment of pulling status, selection of child and the stopping criteria for the entire process. The basic operations are the same as in word extraction, but the attribute in focus and the criteria used are different. Instead of using regional density, the algorithm will make use of the directional weight in its analysis. The criterion used to select children is amended to select only children in the growing directional path. This criterion also allows the growing to be more targeted and pointed towards the growing direction. For the stopping criteria, it is a 2-stage process. In the first stage the growing will begin and continue until it encounters the first merging of words. In the second stage, the growing will proceed until it has detected no further merging of word regions in two consecutive levels.

5 Experimental Results

We now report some of our test cases. In order for our system to focus only on text extraction, large graphically objects are removed through a pre-processing stage [8]. All images are converted to binary images by a thresholding algorithm. The first test case is an advertisement poster with text of varying sizes in the same sentence and aligned in a non-traditional orientation (ie. non-horizontal). This has demonstrated the capability of the algorithm to extract word of different sizes on the same document and even with varying orientation. Figure 9 shows the result of the word extraction. Figure 10 illustrates the merging of words to form their respective sentences. All word groups are correctly merged to the correct sentence. The second test case is a newspaper advertisement for toys. Figures 11 show the output results. All sentences are correctly identified including the three sloping texts, represented by a bounding box for each sentence.

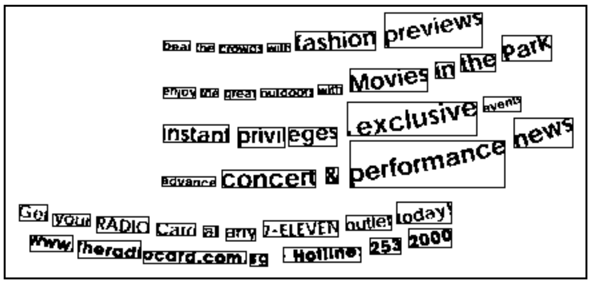
fashion previews
park
Movies in the
exclusive
events
instant privileges
news
advance concert & performance
Got your RADIO Card at any 7-ELEVEN outlet today!
www theradiocard.com.sg Hotline: 253 2000

Fig. 9. Advertisement poster after word extraction

6 Conclusions and Future Work

This paper has proposed a new method to perform word and sentence extraction from imaged documents with large variation in the text size, font, orientation and layout

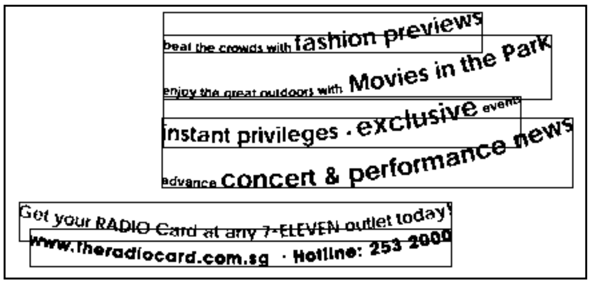
beat the crowds with fashion previews
enjoy the great outdoors with Movies in the Park
instant privileges · exclusive events
advance concert & performance news
Get your RADIO Card at any 7-ELEVEN outlet today!
www.theradiocard.com.sg · Hotline: 253 2000

Fig. 10. Advertisement poster after sentence extraction

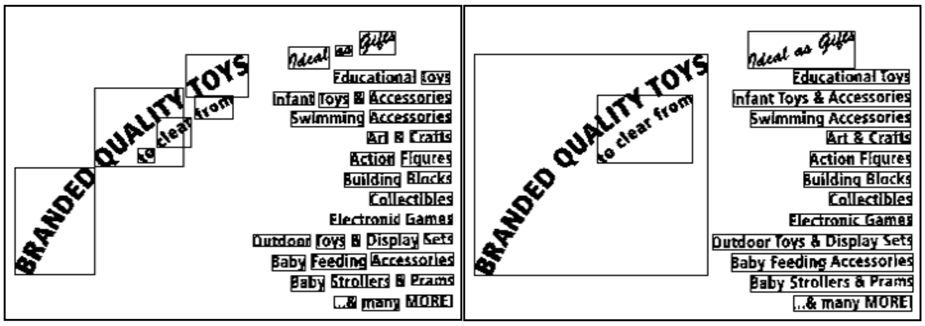

Fig. 11. Toys advertisement after word (ie, left) and sentence (ie. right) extraction

within the same document. The entire algorithm is based on the irregular pyramid structure with the application of four fundamental concepts. Through the process of building the irregular pyramid structure, the algorithm achieves the task of merging characters into words, and words into sentences. It also illustrates the ability to process words of varying orientations and layout where many existing techniques have avoided. Our next task of research is to build a complete system. Starting from the pre-processing to binarize the input image and the elimination of large graphical objects, to the extraction of words and sentences, and finally the correction of text alignment in a form that is acceptable by an OCR system.

References

1. G. Nagy, “Twenty years of document image analysis in PAMI”, IEEE Trans. PAMI, Vol. 22, No. 1, 38–62, (Jan 2000).
2. Richard G. Casey and Eric Lecolinet, “A survey of methods and strategies in character segmentation”, IEEE Trans. PAMI, Vol. 18, No. 7, (July 1996).
3. U. Pal and B.B. Chaudhuri, “Automatic separation of words in multi-lingual muti-script Indian documents”, In Proc. 4th Int. Conf. On Document Analysis and Recogn. (ICDAR ‘97).
4. Yalin Wang, Ihsin T. Phillips and Robert Haralick, “Statistical-based approach to word segmentation”, Proceedings of the ICPR 2000.

5. Dae-Seok Ryu, Sun-Mee Kang and Seong-Whan Lee, "Parameter-independent geometric document layout analysis", IEEE, (2000).
6. K.Y.Wong, R.G.Casy and F.M.Wahl, "Document analysis system", IBM J. Res. Development, Vol 26, 642–656 (1982).
7. G.Nagy and S.Seth, "Hierarchical representation of optically scanned documents", In Proc. 7th Int. Conf. Patt. Recogn. (ICPR), 347–349 (1984).
8. C.L.Tan and P.O.Ng, "Text extraction using pyramid", Pattern Recognition, Vol. 31, No. 1, 63–72 (1998).
9. W.G.Kropatsch and A.Montanvert, "Irregular versus regular pyramid structures", In U. Eckhardt, A. Hubler, W.Nagel, and G.Werner, editors, Geometrical Problems of Image Processing, 11–22 (1991).
10. W.G.Kropatsch, "Irregular pyramids", Proceedings of the 15th OAGM meeting in Klagenfurt, 39–50 (1991).
11. Horace H.S. Ip and Stephen W.C.Lam, "Alternative strategies for irregular pyramid construction", Image and Vision Computing, 14, 297–304 (1996).
12. A. Montanvert, P.Meer and A.Rosenfeld, "Hierarchical image analysis using irregular tessellations", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol 13, No. 4, 307–316 (1991).
13. G. Bongiovanni, L. Cinque, S. Levialdi and A. Rosenfeld, "Image segmentation by a multiresolution approach", Pattern Recognition, Vol. 26, 1845–1854, (1993).
14. P.K. Loo and C.L. Tan, "Detection of word groups based on irregular pyramid", Proc. 6th Int. Conf. on Document Analysis and Recogn (ICDAR), 2001.
15. Boulos Waked, Ching Y. Suen and Sabine Bergler, "Segmenting document images using white runs and vertical edges", Proc. 6th Int. Conf. on Document Analysis and Recogn (ICDAR), 2001.
16. Hideyuki Negishi, Jien Kato, Hiroyuki Hase and Toyohide Watanable, "Character Extraction from Noisy Background for an Automatic Reference System", In Proc. 5th Int. Conf. On Document Analysis and Recogn. (ICDAR), 143–146 (1999).