

**DOCUMENT IMAGE PROCESSING
USING IRREGULAR PYRAMID STRUCTURE**

LOO POH KOK

NATIONAL UNIVERSITY OF SINGAPORE

2004

**DOCUMENT IMAGE PROCESSING
USING IRREGULAR PYRAMID STRUCTURE**

LOO POH KOK

(B.Sc.(Magna Cum Laude), M.Sc)

**A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE**

2004

Acknowledgements

I would like to thank my supervisor, Associate Professor, Tan Chew Lim, for his continuous patience in guiding me, having discussions, providing me materials and spending numerous hours correcting my papers.

I would like to thank Mr. Yuan Bo, for providing me the regular pyramid algorithm to serve as a starting point for my research.

I would like to thank the School of Design and the Environment, Singapore Polytechnic by allowing me to pursue this research study. In particular sincere thank to my Deputy Director Mrs. Winnie Wong who is also my ex-project supervisor while I was studying in the Singapore Polytechnic. Without her encouragement and guidance in finishing my very first programming project, I would not be in this stage. I would also like to thank my section head Mrs. Sia Bee Gee for her understanding during the course of my study.

Finally I would like to thank my parents, family members for their support and encouragement. I would like to thank my wife Oh Yeen Tan. I will never forget your sacrifices and understanding for supporting me all these years.

Table of Contents

1. Introduction.....	1
1.1 Motivation in Document Image Processing.....	4
1.2 Motivation in Pyramid Structure	8
1.3 Our Contributions	9
1.3.1 Binary Input Document Images	9
1.3.2 Gray Scale Input Document Images	10
1.3.3 Color Input Document Images.....	11
1.3.4 Pyramid Structure	12
1.4 Thesis Outline	13
2. Pyramid Structure	14
2.1 Basic Concept of Pyramid Structure.....	14
2.2 Application of Pyramid Structure	17
2.3 The Pyramid Model	20
2.4 Types of Pyramid Structure	24
2.4.1 Traditional Regular Pyramid.....	25
2.4.2 Overlapped or Linked Regular Pyramid	29
3. Irregular Pyramid	35
3.1 Types of Irregular Pyramid	35
3.2 Irregular Pyramid Construction Process	41
3.2.1 Creating a New Pyramid Level.....	42
3.2.2 Selecting Neighbors	43
3.2.3 Selecting Survivors	46

3.2.4	Selecting Children.....	54
3.2.5	Stopping Criteria.....	58
3.2.6	Handling of Root Nodes	59
3.3	Irregular Pyramid in Textual Segmentation.....	60
4.	Word Segmentation in Binary Imaged Documents	61
4.1	Related Works.....	62
4.2	Fundamental Concepts.....	67
4.2.1	Inclusion of Background Information.....	67
4.2.2	Concept of “closeness”	68
4.2.3	Density of a Word Region	69
4.3	Pyramid Model.....	70
4.4	Pyramid Formation	72
4.4.1	Selection of Survivors.....	73
4.4.2	Selection of Children	74
4.4.3	Stopping Criteria.....	76
4.5	Experimental Results	77
4.6	Summary and Discussion.....	83
5.	Identification of Textual Layout	84
5.1	Fundamental Concepts.....	84
5.1.1	Density of a Word Region	85
5.1.2	Majority “win” Strategy.....	86
5.1.3	Directional Uniformity and Continuity.....	86
5.2	Pyramid Model.....	88
5.3	The Algorithm.....	90

5.3.1	Word Extraction Process.....	90
5.3.2	Sentence Extraction Process	95
5.4	Experimental Results	98
5.5	Summary and Discussion.....	103
6.	Adaptive Thresholding in Gray Scale Images	104
6.1	Related Works.....	104
6.2	The Algorithm.....	107
6.3	Pyramid Model.....	109
6.4	Segmentation.....	111
6.4.1	Base Pyramid Level Formation	112
6.4.2	Higher Pyramid Level Formation	116
6.5	Binarization and Filtration	116
6.6	Experimental Results	118
6.7	Summary and Discussion.....	123
7.	Textual Segmentation from Color Document Images	124
7.1	Related Works.....	125
7.2	Color Space and Distance Measurement	130
7.3	Proposed Method	133
7.3.1	Pre-processing Stage.....	133
7.3.2	Pyramid Model.....	134
7.3.3	Detailed Segmentation Stage	137
7.4	Threshold Derivation	140
7.5	Experimental Results	141
7.6	Summary and Discussion.....	150

8. The Storage Requirement and the Processing Speed Analysis.....	151
8.1 Storage Requirement Analysis.....	151
8.1.1 Regular Pyramid Model.....	151
8.1.2 Adaptive Irregular Pyramid Model.....	152
8.1.3 Our Irregular Pyramid Model	155
8.2 A Rough Estimation of Complexity	157
8.3 Processing Speed Analysis	158
9. Conclusions and Future Directions.....	160

Summary

This thesis will present the research in the use of the irregular pyramid structure in document image processing. The focus is in the segmentation and the extraction of textual components from binary, gray scale and color document images with mixed texts and graphics. The thesis presents our solution to address the common problem in handling documents with texts in varying sizes and orientations during the segmentation while most methods have assumed a Manhattan or a dominant skew document layout. The solution extends beyond the isolation of word groups to the identification of logical text groups (e.g. sentences) containing word groups with non-uniform orientations. It also presents an adaptive thresholding solution which does not require the pre-determination of a fixed local window size for the binarization of the gray scale textual objects. Finally the thesis discusses our solution in the segmentation of the textual regions from color document images where others have problem in the isolation of the textual component as a compact region. All the proposed solutions are based on the classical irregular pyramid framework with novel construction algorithms to adapt to the specific requirements in our document image analysis tasks. The key differences are in the design of the survivor and the child selection processes where alternative in the derivation of the surviving values and the utilization of the different selection criteria in varying applications are implemented. Our model also differs from the traditional pyramid formation process in the alteration of the processing objective on different pyramid levels where a same objective is applied to all levels in the traditional process. The thesis highlights many past methods, discusses their pros and cons and supports our proposed methods with various experimental results.

Chapter 1

Introduction

Document image processing is a sub-field under the general image processing research arena. It focuses on the processing of document images where the existence of textual content is assumed. Although there may be graphical objects present, the emphasis is on the processing of the textual components.

A document image can be defined as a static representation of a specific recorded instance of a transaction. It can be either in a hardcopy or a softcopy format. The former requires some form of scanning process to convert it into an electronic format. Unlike the majority of the ASCII documents, the contents are represented by a collection of pixels. Despite having some textual information within the document, the contents are merely groups of pixels. Just like its graphical counterpart in the document, it cannot be used in any indexing or searching tasks. In order to make use of such textual contents, the subject areas must be isolated and through some recognition processes converted into a searchable and editable format. The focus of our research is to explore the use of irregular pyramid model to isolate or extract such textual content. The task in the segmentation and the extraction of text from mixed text and graphic document images remains a very essential and important processing step. Many applications require and demand an efficient and accurate text segmentation and extraction technique in their processing. The applications can be classified as front-end processing or back-end processing.

In the front-end processing category, the extracted textual content is put into immediate use by the application. The traditional applications like the extraction of postal code from an

envelop address block will be used immediately to direct the mail sorting machine to place the envelope into the correct bin. Such applications will require accurate and fast extraction and recognition of the textual content. The vehicle license plate recognition system used in car park payment management and the monitoring of container truck moving in and out of the sea port are some other applications in this category. The accurate identification of license plate numbers and the tracking of time of entering and leaving of the respective vehicles will allow correct processing of vehicle parking charges. The automatic tracking and recording of container track vehicle numbers will avoid tedious manual monitoring and traffic congestion at the gate. Reference [72] described such a number plate reading system. Some other similar applications are in road signs identification for unmanned vehicle navigation system and parts identification in factory automation. These applications share a common requirement to detect text in a real scene as described in [73, 74, 75, 76, 77]. Web page processing is another type of application under this category. Although the majority of the web contents can be extracted and searched through the analysis of the HTML code, text embedded in some of the graphical components are not within the reach of a normal search engine. Despite the availability to use the tag feature, most web designers never use it. As a result, important and key information placed within the image is non searchable by most search engines. In order to solve this problem, the embedded textual content must be identified, extracted and converted into a searchable format as mentioned in [78, 79, 80, 81, 82, 83]. One common concern in this category of applications is the speed of segmentation and extraction.

The second category pertains to those applications that require the extracted textual content for back-end processing. The process is usually done in batches and the content is captured and stored for later usage. Although speed is not as crucial as the previous category, the accuracy and the automation of the process is vital. The extracted content is

mainly for archiving, indexing and categorizing of large amount of document images for later processing, retrieval and searching purposes. There is a large group of applications under this category. The indexing in the digital image library, multimedia components database, geographical information system and video database require the prior extraction of textual content. As reported in some papers, image indexing based on text extraction is more effective than using object shape extraction which is more complex and computationally costly. As mentioned in Osamn Hori's paper [86], the extraction of video text which contains meaningful information about the video contents can act as a keyword in video indexing for searching and categorization of video. Many other papers [84, 85, 86, 87, 88, 89, 90] also proposed their own methods in this area of applications. Besides the indexing applications, other applications such as the automatic engineering drawing scan-input system, form processing and the digitized manuscripts of old literatures also require efficient text segmentation and extraction method. The conversion of old engineering drawings into appropriate CAD format requires the separation of the textual and the graphical components. Several papers have proposed different methods for this task [91, 92, 93, 94, 95, 96]. Form processing, as in [97, 98, 99], is another type of application under this category. It involves the scanning of filled-in forms, isolating the filled-in areas and finally extracting and recognizing the filled-in contents for processing. Wong et al [100] described such a system making use of the color content to aid the extraction of filled data from a standard form layout. The digitizing of old literatures [102, 103, 104, 105] where the target document images are frequently degraded also requires careful isolation of the textual component from the interference of noise regions. In [101] the author reported a system to convert rare and precious old literature manuscripts into a digitized format. The system converts the manuscript into both page image format and also in full text format to enable the viewing of literature in its original form and also the searching of literature based on the

full text format. Finally applications like the newspaper document analysis [106, 107, 108] and map interpretation [200] also require some form of textual segmentation activities.

1.1 Motivation in Document Image Processing

On the one hand, the analysis of the document images is a more restrictive form of general image processing, bounded within the document images domain. On the other hand it also requires a higher precision in terms of the processing due to the existence of the smaller target components and the closer proximity of the objects. A traditional document image processing system will involve many processes. Some are the pre-processing steps which include the filtering of noise, the correction of document skew, the binarization of gray scale input images or the quantization of color document images. The process will then be followed by the actual segmentation, the extraction and finally the categorization of image contents. The post processing steps will involve the preparation of the extracted content which is followed by the recognition process. Despite decades of studies by many proposed methods in handling these processes, they are still some existing problems which allow rooms for improvement. Some of the problems have been reported in numerous published surveys on document image processing [117, 132, 135, 142, 143, 146, 148, 155, 170]. In this thesis we will focus only on those processes that we have suggested alternative solution to the problems.

Most of the document image processing algorithm requires some form of skew correction before the actual segmentation. Although there are numerous proposed methods in performing skew correction, problems remain in terms of the accuracy and the strong assumptions requiring a dominant skew angle for the entire document or a common skew direction within the same text group. The presence of graphics also poses a great challenge among many skew correction methods. In the binarization of gray scale images which is a

frequent pre-processing step, the absence of bimodality in most input document images prevents an efficient use of global thresholding methods. Although more adaptation to the varying gray scale condition is achieved through the use of local adaptive thresholding technique, the requirement in the definition of a fixed local window size also constraints its application. Just like the binarization, color quantization is also a commonly use pre-processing step in processing color document images. The purpose is the same as the binarization process to reduce the representing state of each pixel in the input image. But it differs from the binarization process in the resulting number of states which is more than a binary state. Although there are many proposed methods in dealing with color quantization, they may not be suitable for the purpose of textual segmentation. In this context the main aim of the quantization process is to reduce the representing states to as low a number as possible to ease the computational load and yet retain a sufficiently large enough states to maintain the richness in color for the actual segmentation task. The method must also be efficient enough and leave the detailed segmentation task to a later process. The majority of the existing methods are either very efficient but perform too much quantization or too precise and lack in the processing efficiency.

There are three types of input document image. They are the binary, the gray scale and the color images. In the context of textual segmentation all three image types face some common challenges as well as difficulties peculiar to each individual type. The greatest challenge is in the processing of non-Manhattan layout documents. This is mainly due to the reliance on the utilization of the smearing and the XY-cutting concept which most methods use where the underlining assumption of these two approaches requires a horizontally aligned textual content. Although the Hough transform allows the estimation of the text orientation, its application is limited by the difficulty in the determination of an appropriate centre line and the angular steps in the analysis. Efficiency is also a general concern. The

most frequently used connected component analysis also encounters problems in the joined or broken character situation which has violated its fundamental objective to isolate individual characters. For the segmentation of text beyond the character level, most methods will need to employ again the smearing and the XY-cutting approaches. On top of the above mentioned problems, the requirement to perform detailed spatial analysis of the textual components in order to determine some type of inter-textual components distance threshold in all approaches also resulted in some rigidity in most methods. Document images with irregular text sizes, fonts and orientations always pose a problem for most of the existing methods.

In the handling of gray scale document images, binarization is a widely used pre-processing step in many methods. For document images with reverse text, binarization will not be suitable. There are also methods that perform direct segmentation from gray scale images capitalizing on the existence of multiple gray levels. Edge information is a popular achievable property from gray scale image and many direct segmentation methods utilize this information as the key factor to assist in the isolation of the textual content. Despite its popularity, difficulty arises in the determination of a suitable sensitivity level for the edge operator and the verification of the true edge point. Even after the correct extraction of the valid edge points, the alignment and the merging of the edge points for the isolation of textual region is still not an easy task. The assumption of a Manhattan document layout and the prior determination of inter-component spacing re-surface. Finally there are also methods that attempt to use the texture property to aid the segmentation task. High computational cost is the key problem in this category of segmentation method.

Lastly we have the color document image type. Although among the three different image types the number of proposed methods in the color textual segmentation domain is

not as high as the other two types, the use of color in document images have slowly gained its popularity. Just like the gray scale images, color quantization is often used as a pre-processing step attempting to reduce the number of color representations. Many color textual segmentation methods place a high emphasis on this pre-processing step trying to reduce the number of unique colors to a manageable number of color layers. Based on the generated color layers the same processing approaches (i.e. smearing, XY-cutting and connected component analysis) as in the binary or the gray scale images are applied to the respective color layers where the same problem in the requirement to have uniform horizontal document layout as discussed above exists. One new problem unique to this way of processing color images is the number of representing states. Due to the fact that color quantization is a category of feature-space based type of color segmentation/clustering method where the only consideration is within the color space and no spatial factor is used in the clustering process, very fragmented textual component is frequently the end product. As a result, a very intricate post-processing step is required to identify and merge components belonging to the same textual object. In order to solve this problem, there is a category of color segmentation methods that are based on domain. The main objective of these methods is the inclusion of spatial information while performing color clustering. In another words, both color and spatial factors are used at the same time while performing the textual segmentation. Nevertheless the majority of the proposed methods in the context of textual segmentation only attempt to incorporate some spatial information into a mainly feature-space based method. One of the main domain-based approaches is the region growing approach [207]. The advantage of this approach is the ability to take both color and spatial factors into consideration during region growing. Despite this benefit, it also suffers the problems of the sequential processing, the selection of suitable seed points and the determination of an appropriate growing criterion. A final difficulty that is shared by all color segmentation methods is the measurement of color distance. Till date there is still no

standard way in deriving an accurate color distance measurement. In view of the wide variety of color spaces and the subjectivity in determining the closest between colors, the task in measuring distance between colors gets even tougher.

1.2 Motivation in Pyramid Structure

Pyramid model has been around since the 1970's. It is basically a data structure holding image content in multiple coarser versions on different pyramid levels. There is a wide range of models from a simple regular structure with static horizontal and vertical configuration to a fully flexible structure with deviation in both horizontal and vertical layout to fit the input content. There are some applications of the pyramid model in textual segmentation. The majority of them employ the regular pyramid structure. Most of these studies still require connected component analysis in binary image and thus the assumption of disjoint components still exists [31, 48]. The main problem as reported in [56] is in the rigidity of the structure. Problem arises when it is used to segment elongated and non-uniform image objects. Although a later proposed linked regular pyramid model provides some flexibility in the vertical linkage, the inherited static horizontal layout from the regular model still restricts in its ability to adapt to the actual input content. The most flexible model is the irregular pyramid, but to the best of our knowledge there is yet any proposed method making use of such a model in textual segmentation. The majority of the irregular pyramid related papers mainly revolves around the structure and its formation issues. Not many have touched in the actual application of the structure. Only a few have attempted to apply the structure in the area of general segmentation. Most of these applications are just merely samples to illustrate the formation of the structure. The benefit of using the irregular pyramid model, especially in its local processing, hierarchical abstracting, content adapting, natural aggregating of image properties and the heuristic criteria application ability have yet to be explored in detail.

1.3 Our Contributions

In view of all the above problems and motivations, this thesis will suggest and report a series of solutions for document image processing using irregular pyramid model. The focus is on the segmentation of the textual component from the three types of input document images (i.e. binary, gray scale and color). The following will highlight our contributions in solving problems in each type of the input document images.

1.3.1 *Binary Input Document Images*

Although the first solution is developed from the consideration of binary document images, the solution is fundamental and it applies to the remaining two image types as well. In this solution, we make no assumption in the physical document layout. The algorithm has the ability to process document images with text of varying sizes, fonts and orientations. This will include texts within the same text group, sentence or even word. The input document images are always assumed to contain graphical objects. The flexibility in handling such situations allows our algorithm to completely discard the skew correction pre-processing step. The basic technique used in the segmentation is a bottom-up region growing approach from multiple seed points. No smearing, XY-cutting or Hough transform is utilized. As a result, the assumption of a Manhattan layout is no longer required. Our algorithm also does away with the connected component analysis. A major problem with the connected component method is that an extracted component may consist of multiple characters in the case of joined characters or fragments of a character in the case of broken character. This will create some complications during the recognition phase. On the contrary, our method will extract all components at the word's level regardless of whether there are joined or broken characters and thus simplify the recognition task by focusing only on word's recognition. The algorithm also extends beyond the word's level to extract logical

groups of words (e.g. sentences) with the ability to handle even varying word sizes and orientations within the same group. Although our proposed method still requires the assumption of inter-characters spacing to be smaller than the inter-words spacing, the actual distance need not to be pre-determined. As a result no spatial analysis is required to determine any distance threshold. The bottom-up natural clustering of neighboring regions from pyramid level to level will allow the growing of the character fragments/strokes into words and the growing of words into sentences systematically and heuristically in a concurrent manner. Different portions of this solution are presented in our three publications [65, 66, 67] and the detailed algorithm is further described in Chapter 4 and Chapter 5.

1.3.2 Gray Scale Input Document Images

Based on the same ability to process non-Manhattan layout in binary input document images, we continue to explore the handling of gray scale images. Our solution to the binarization problem is based on the local adaptive method, but the requirement to have a fixed local window size as in the other local thresholding methods is not needed. Differing from the usual sequence of performing binarization before actual segmentation, our proposed solution will perform a rough segmentation of the textual component including some background areas surrounding each word's contour forming a tightly bounded region. With all the isolated word regions, the algorithm will then perform binarization of the individual regions with the flexibility of using different thresholding methods for different regions. The binarization is achieved by using three simple thresholding methods and the best result is determined based on some deviation values. The final result is by combining the best binarized versions of the respective word regions. The key contribution of this proposed method is dispensing with the need for a fixed local window size while enjoying the flexibility and the adaptability of local thresholding. This is done by the deferment of the binarization process after the segmentation of a rough target region to facilitate local

thresholding without the interference from the other non-target regions. Our method also provides an alternative to the filtering of noise at various appropriate stages of the algorithm. No edge or texture property is employed. The proposed method is discussed in detail in Chapter 6 and it is published in [70].

1.3.3 Color Input Document Images

Unlike the majority of feature-space based methods that result in fragmented textual components, our proposed method utilizes a combination of feature-space based approach and domain-based approach. The former allows a fast clustering of “close” colors while the latter facilitates a detailed segmentation of the textual region. Our contributions are in five areas. The first is in the area of color measurement where a simple measurement method in the RGB color space is derived. The second is in the area of color quantization where an efficient method without the need for a color histogram is proposed. The third is in our region growing method where seeds are selected dynamically and repeatedly to suit the best local condition, which avoids the problem of having a fixed seed dominating the entire growing process. The problem of sequential processing encountered by the other region growing methods is also addressed by having multiple seeds to grow concurrently. The fourth area is in the adaptive determination of the growing criterion (i.e. closest color). Guarded against a largest possible color distance, each individual region will dynamically determine and compute its own color threshold to regulate the growing rate adapting to the varying local condition. The final contribution is a slight deviation from the color document images, where the ease in the alteration of some of the selection criteria allow the algorithm to also process gray scale document images. In contrast to the usual gray scale image processing, it allows the analysis of the varying gray scale component on different gray scale layers. This has enabled the processing of reverse text. It also avoids the complication in the analysis of neighboring components with different gray scale levels; especially when

the largest background region is isolated on a single layer. The solution is presented in Chapter 7 and it is published in [71].

1.3.4 Pyramid Structure

A special irregular pyramid structure with novel construction algorithms is proposed in this thesis to tailor to the need of textual segmentation in document images. Our main contributions are in five areas. First, this is the first attempt to use irregular pyramid structure to enable natural grouping of texts. This dispenses with the need for connected component processing and spatial analysis used in the traditional approach. The second is in the design of the surviving value which is the key attribute used in the selection of the survivors or seed points. Depending on the various specific requirements, different surviving value derivations are proposed. We have explored using the regional mass (i.e. number of foreground area) in [65, 66, 67], the gray scale intensity variance in [70], the number of large neighbor in [70] and the number of eligible neighbors in [71]. Each has its unique purpose contributing towards the subsequent processes. The third area is in the survivor selection process which is a departure from the usual irregular pyramid construction by inhibiting the participation of non-promising regions. This proposed modification is also supported by a later paper in [69] by Jolion with a slightly different motivation in relaxing the survivor selection rules. The fourth is in the child selection process. An alternative approach that allows the survivor to initiate the selection process is proposed for specific applications of segmentation as reported in [65, 66, 67, 70] to achieve a more accurate segmentation result. The fifth area is in the adoption of the different processing objectives on different pyramid levels. This is in contrast to the universal objective across all pyramid levels in the traditional pyramid construction. This strategy has served well in providing independent but concurrent processing of different regions of document images in text segmentation.

1.4 Thesis Outline

This thesis starts with the introduction of the importance and the various applications of document image processing, in particular textual segmentation. It is followed by the presentation of our research motivation in terms of document image processing and in the area of pyramid structure where some of the common problems faced by most of the existing methods are discussed. Chapter 2 will present the basic concept and construct of pyramid structure used in image processing. It will categorize and summarize the past literatures using pyramid structure in solving image processing problems. A general pyramid model is formally defined. Based on this model, the two main types of regular pyramid are described. Chapter 3 will focus on the irregular pyramid structure which is the main model we use in this thesis. The irregular pyramid construction process and some of the variations and considerations are discussed. The thesis continues to illustrate the use of the defined irregular pyramid model to solve problems faced in the segmentation of textual components from document images. It focuses on 4 main areas. Chapter 4 describes the first area which is the extraction of word components in varying sizes and orientations from binary document images where most methods have assumed horizontally alignment and constant size text. The work is published in [65, 66]. Chapter 5 talks about the second area which is the identification of logical grouping for document layout analysis. The work is published in [67]. Chapter 6 presents the third area which is the use of irregular pyramid to assist the adaptive thresholding of gray scale document images. This work is published in [70]. Finally Chapter 7 presents our solution in the extraction of texts from color document images as a compact region. This work is published in [71]. The thesis will finally discuss the issues of the storage requirement and the processing speed of using irregular pyramid in Chapter 8 and end with a conclusion and future directions in Chapter 9.

Chapter 2

Pyramid Structure

In this chapter we will introduce the basic concept of pyramid structure, the benefits and the various existing applications of the structure. In order to have a common ground to discuss the various pyramid structures, a generalized pyramid model is formally defined. The chapter will then continue to describe the various types of pyramid models where their pros and cons are discussed.

2.1 Basic Concept of Pyramid Structure

Pyramid is a form of image data structure that is used to hold the image content in multiple resolutions. The original image content is represented in successive levels of reduced resolution. Starting from the pyramid base holding the original image, each higher pyramid level holds a representative set of the image content of the lower level with a coarser resolution. Based on a suitable control of the reduction or contraction criteria, an image can be appropriately reduced in terms of its resolution and yet able to maintain the key content of the image. As a result the contraction process is also an abstraction or a summarization process. The abstraction of the content will continue until the pyramid apex, which becomes a single element. The spatial relationship among all pyramid elements are maintained either implicitly or explicitly during the formation process. Each element is aware of its direct surrounding neighboring elements and a group of elements on the immediate lower pyramid level that it represents. The former is the horizontal or the neighborhood relationship and the later is the vertical or the parent-child relationship. Based on these relationships, a 2-dimensional hierarchical structure is formed.

From the data content point of view, as described in [56], each pyramid data point can be interpreted as a measurement at a discrete point on the image plane or it can be treated as a representation of a region that partitions the image domain. From the application point of view, there are also two interpretations of the pyramid structure application abilities. The first is the decimating or the abstraction ability of the pyramid structure. A large image can be decimated into smaller sizes with lower resolutions which are equivalent to the summarization of image content into multiple versions with progressive abstraction. This has realized the possibility of processing the image in varying resolutions to increase computation efficiency and decrease analysis complexity. Due to the smaller image size, fewer computational steps are required. Appropriate resolution level can be selected to meet a specific analysis requirement depending on the level of details. The structure also allows fast identification of the target regions on a low resolution level to be followed by a more elaborate processing of the target regions at the higher resolution. The processing can also be done on multiple resolution levels and merge the outcomes at the end to yield the best result.

The second is the application of the “growing” ability. Although the pyramid structure formation is traditionally viewed as a decimation process, it can also be viewed as a growing process. Instead of focusing on the surviving elements on each pyramid level, the attention can be repositioned to the actual region represented by each surviving element. They are the regions formed by traversing down the parent-child link of each surviving element to the base pyramid level holding the original image. On each pyramid level the selection of the representative set to form the higher pyramid level are equivalent to the selection of seeds and the parent-child linkage is comparable to the growing of seeds. As we move up the pyramid levels smaller regions are grown by merging with neighboring regions to become

larger regions. With an appropriate definition of the representative set selection criteria and the parent-child linkage conditions, multiple regions can grow and merge concurrently within the structure towards the final and target configuration. This process is further illustrated in figures 1 to 4 where the elements on each pyramid level represented by the white spots and the image regions covered by the elements represented by various colors are super-imposed. On pyramid level 1 (i.e. Figure 1) there are 35 pyramid elements where each represents a small fragment of the word “gate”. As we move to pyramid level 2 (i.e. Figure 2) only 11 out of the 35 elements from level 1 are selected to survive on this level. In contrast to the decreasing number of pyramid elements, the actual regions on the base pyramid level represent by each surviving elements grow in terms of the regional size. This process continues on pyramid level 3 and eventually the entire word “gate” is formed on pyramid level 4 represented by a single pyramid element. The number of pyramid elements and the surviving elements onto the next pyramid level are shown in Table 1.

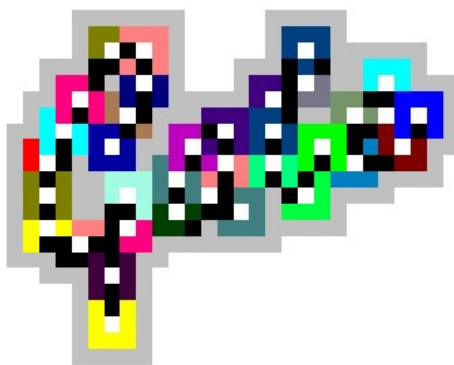


Figure 1. Pyramid level 1

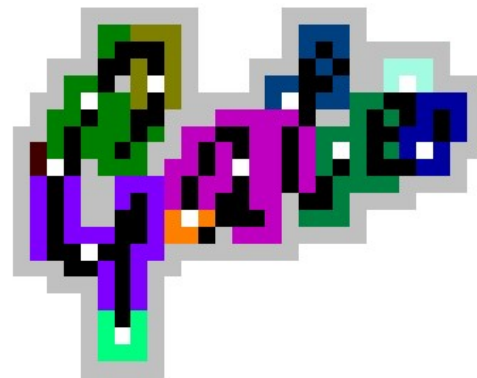


Figure 2. Pyramid level 2

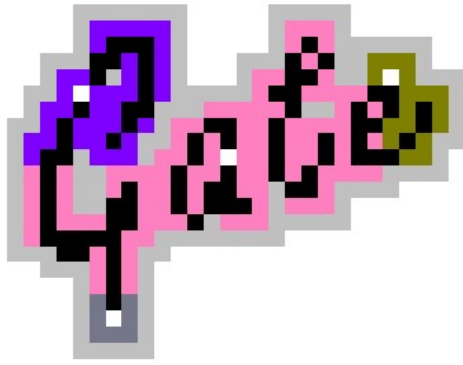


Figure 3. Pyramid level 3

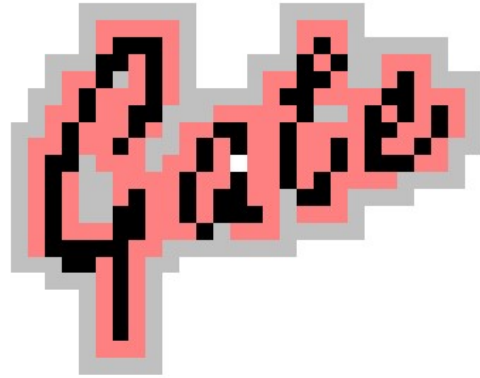


Figure 4. Pyramid level 4

Table 1. The gate image

Pyramid levels	Number of elements	Number of survivors
0	744	35
1	35	11
2	11	4
3	4	1
4	1	0

2.2 Application of Pyramid Structure

As early as 1971, researchers have already started to utilize the pyramid structure in saving processing time by working on the reduced resolution image. The savings in the processing time is clearly shown by Andelson et al [7] where the convolution with large weighting kernel can be simulated with the convolution in multiple reduced image resolutions. The computational saving also arises from the reduced analysis complexity in coarser images. The structure has provided the ability to handle problems at different levels of detail as explained in [13].

Pattern matching and plan-guided analysis and searching are two of the application examples that fully exploit these advantages. In pattern matching, the identification of a specified pattern can be done at a lower image resolution. As reported in [7] even with the

application of the match on all higher pyramid levels (i.e. except the base level) the cost is only one third that of searching on the original image. In another paper [9], the linked regular pyramid is used to perform region matching where the authors have shown that the approach is more robust than the standard moment-based method. This is also true in plan-guided searching applications where more efficient searching can be done with the pyramid structure. This is achieved by constraining the search area by first identifying those potential regions on a higher pyramid level which have a lower computation cost with a lower image resolution. More detailed analysis can then be performed on the lower pyramid level with higher resolution within the areas indicated by the results of the previous identification. This application is discussed in [7, 14, 16, 48, 77].

The structure also enables the processing of each pyramid level independently yielding different results and the outputs from the various levels can be integrated to complement each other shortfall to create a robust final result. Wu et al [138] describe the formation of a regular pyramid structure to assist the extraction of major edges in textual components. Each resolution level facilitates the filtering of varying degree of noises and the identification of edges belonging to the different text fonts and sizes. The outputs from the various pyramid levels are then combined to produce the final result.

Noise filtering is another advantage which is an inherited property of the pyramid structure. The structure has a natural ability in noise reduction during the image contraction process due to the low-pass filtering effect. In [7], the Laplacian pyramid is used in the removing of random noise. In [12] Jolion et al attempt to use the linked regular pyramid in processing images with low signal-to-noise ratios. On the same path, the structure is frequently used in the image smoothing application. Interesting smoothing result is obtained in [2] while maintaining clear boundary contrast among regions. Smoothing only occurs

within the interior of the region. As compared to the traditional smoothing operations this is difficult to achieve.

Because of the hierarchical construct of the structure, the pyramid structure is amenable to concurrent and parallel processing. Many pyramidal computer architecture systems [9, 10, 16] are introduced and described which allow concurrent formation of the pyramid structure. This has enabled the deployment of the pyramid structure in those applications that require real time tracking of moving objects. Tan and Martin [10] describe such a tracking system by processing the object concurrently in multiple non overlapping local windows within the traditional regular pyramid structure

The final most important property of a pyramid structure is in its local processing ability. The analysis and interpretation of global features can be achieved by local information accumulation and collection with the ability to even retain spatial relationship in the representation. Global objective is attained through local adaptation. This has permitted fast detection and extraction of global structures from an image which is a key requirement in image segmentation. The paper in [11] analyzes and describes the use of simple regular pyramid structures in the detection of global structures like similarity (i.e. bimodality), proximity (i.e. compact regions), continuation (i.e. smooth curves) and closure (i.e. blobs and ribbons) in binary images. Many proposed methods [1, 4, 5, 6] have based on this property to implement image segmentation algorithm. The method in [7] uses it in the estimation of the integrated property of a local region (i.e. texture).

In addition to the above mentioned applications, the structure is also used in the construction of image mosaics as in [7] where the objective is in the joining of different images with smooth boundary. It is also used to create realistic looking images [8]. Data

compression is another application which capitalizes on the ability of the structure in the systematic reduction of image data points [7].

2.3 The Pyramid Model

This section will define a formal pyramid model. The model is generalized to represent different types of pyramid structure. All subsequent sections will base on this model for the discussion of the various issues.

The input document image is represented by a series of pixels arranged in a rectangular coordinate of rows and columns (i.e. r and c). The total number of rows represents the image height while the width of the image is divided into columns. For easy reference, the row and column of each pixel are transformed into a unique index (i.e. p) calculated as shown below. Each pixel is uniquely identified as p ranging from 0 to the image size (i.e. not inclusive).

	C0	C1	C2	C3
R0	0	1	2	3
R1	4	5	6	7
R2	8	9	10	11

←

$$img(r, c) = \begin{cases} r = 0 \text{ to } imght - 1 \\ c = 0 \text{ to } imgwd - 1 \\ p = r * imgwd + c \\ 0 \leq p < imght * imgwd \end{cases} \quad (1)$$

Depending on the image format (i.e. binary, gray scale or color) each pixel p is associated with an intensity attribute Y_p , which can either be a single value or a vector of values. For a binary image the value of the attribute is either 0 or 1. For a gray scale image the value will typically fall into a range of 0 to 255. In the color image it is a combination of a triplet red, green and blue intensity, each normally having a range of 0 to 255. In this report we will treat 0 as the black intensity while 255 representing the white intensity.

$$\overrightarrow{Y_p} = \begin{cases} \text{binary} = [0] \mid [1] \\ \text{gray scale} = [v] \text{ where } 0 \leq v \leq 255 \\ \text{color} = r[v], g[v], b[v] \text{ where } 0 \leq v \leq 255 \end{cases} \quad (2)$$

$$\begin{aligned} \text{img}(r, c) &\Rightarrow \{L_i \mid i = 0 \text{ to number of pyramid level}\} \\ L_i &= \{\overrightarrow{D_{i,j}}\} \text{ where } j = 1 \text{ to } N_i \\ L_{i+1} &= \text{Transform}(L_i) \\ L_{i+1} &= \{\overrightarrow{D_{i+1,k}}\} \text{ where } \begin{cases} L_{i+1} \subset L_i \\ k = 1 \text{ to } N_{i+1} \\ N_{i+1} < N_i \end{cases} \end{aligned} \quad (3)$$

In a pyramid model the input image is represented in successive layers of pyramid levels as shown in equation 3. Each pyramid level L_i will hold a set of data points $D_{i,j}$ with j ranging from 1 to N_i representing the total number of data points on the pyramid level i . The pyramid base L_0 will have N_0 number of data points equal to the total number of pixels in the original input image. By the application of a transformation function, the lower pyramid data points are transformed into a smaller set of data points on the higher pyramid level. The higher pyramid level L_{i+1} will hold a proper subset of the data points from the lower level L_i . From a strict data structure point of view, no new data point is created or introduced. A representative set of data points from the lower pyramid level i are selected to form the data set $D_{i+1,j}$ of the higher pyramid level $i+1$. This reduction in data points (i.e. $N_{i+1} < N_i$) from a pyramid level to another level will continue as we move closer and closer to the pyramid apex. This process of pyramid size reduction will continue until either the pyramid apex where there is only 1 data point (i.e. full pyramid structure) or some intermediate pyramid level (i.e. a tapered pyramid with a flat top). The later will have to be determined through the satisfaction or convergence of some functions guiding the transformation.

$$\begin{aligned}
\overrightarrow{D_{i,j}} &= \left\{ d_{i,j}^p, \overrightarrow{d_{i,j}^y}, d_{i,j}^a, \overrightarrow{d_{i,j}^b}, \overrightarrow{d_{i,j}^c}, \dots \right\} \\
\overrightarrow{D_{i+1,k}} &= \left\{ \exists \overrightarrow{D_{i,j}} \mid \text{survive}(\overrightarrow{D_{i,j}}) \wedge (d_{i+1,k}^p = d_{i,j}^p) \right\}
\end{aligned} \tag{4}$$

In order to select the list of data points to be used for the next higher pyramid level, a surviving function is used as shown in equation 4. The selected data point is also known as the survivor. Data point $D_{i,j}$ on level i will survive to become a data point $D_{i+1,k}$ on level $i+1$ if it satisfies a ‘survive’ function. Both $D_{i,j}$ and $D_{i+1,k}$ can be viewed as the same vector data point having the same unique pixel index d^p (i.e. pointing to the same position in the original input image), which will be elaborated in the following paragraph.

The pyramid data point $D_{i,j}$, which differs from the image data point (i.e. pixel) associating only with an intensity value Y , is associated with a vector of attributes. There are two types of attribute. One is the unique attribute that will remain unique and unchanged from a pyramid level to another level. The other is the collective or derived attribute, which maintains and holds the collective value of a group of image regions formed by multiple pixels. The former enables the propagation of exclusive image information through the pyramid levels and the later allows the abstraction and encapsulation of image information. Among the many possible attributes, the pixel index d^p , the intensity value d^y , the area d^a , the neighborhood list d^b and the children list d^c are some of the common ones. Except for d^p which is a unique attribute, reflecting the absolute position of the pyramid data point in the original image, the remaining are the collective attributes that will vary from a pyramid level to another pyramid level reflecting the collective status of the data point. The purpose of having the pixel index attribute d^p is to allow the unique identification of all surviving data points on every pyramid level with respect to the original image. As the degree of image abstraction increases with lower resolution (i.e. on the higher pyramid level) where

the exact boundary of image objects are lost, this attribute becomes an essential linkage between the abstract and the original image versions.

Among the collective attributes, the neighborhood list d^b and the children list d^c are the most essential in maintaining the pyramid structure. The purpose of the neighborhood list is to maintain a list of neighboring data points on the same pyramid level. Both are vector attributes as defined in equation 5. The neighborhood list $d_{i,j}^b$ will contain all surrounding data points α_q that are adjacent to $D_{i,j}$ and share a common border. The children list will allow linkage of data points in two consecutive pyramid levels. Any data points β_r on the immediate lower pyramid level L_{i-1} that fulfills the criteria as a child of $D_{i,j}$ is maintained in the children list. The number of adjacent neighbors N_b and the number of children N_c will vary on different pyramid levels and according to the type of pyramid structure. These two attributes can be maintained as a simple array list holding the unique index number of the neighbor or child. Pointer is another alternative to maintain the linkage.

$$\begin{aligned}\overline{d_{i,j}^b} &= \left\{ \alpha_q \mid \alpha_q \in L_i, \alpha_q \neq \overline{D_{i,j}}, \text{adjacent}(\overline{D_{i,j}}, \alpha_q) \right\} \text{ where } q = 1 \text{ to } N_b \\ \overline{d_{i,j}^c} &= \left\{ \beta_r \mid \beta_r \in L_{i-1}, \text{child}(\overline{D_{i,j}}, \beta_r) \right\} \text{ where } (r = 1 \text{ to } N_c) \wedge (i \neq 0)\end{aligned}\tag{5}$$

Two other frequently used collective attributes are the intensity attribute d^y and the area attribute d^a as described in equation 6. The intensity value of a data point on a pyramid level is obtained by considering the intensity of its children data points (i.e. β_r) on a lower pyramid level through some type of averaging function. Just like the intensity attribute, the area attribute is also a collective value by summing the area attributes of all children data points.

$$\begin{aligned}
\overrightarrow{d_{i,j}^y} &= AveIntensity(\beta_r^y \mid \overrightarrow{\beta_r \in d_{i,j}^c}, \text{ for } r = 1 \text{ to } N_c \wedge i \neq 0) \\
d_{i,j}^a &= \sum_{r=1}^{N_c} (\beta_r^a \mid \overrightarrow{\beta_r \in d_{i,j}^c}) \text{ where } i \neq 0
\end{aligned} \tag{6}$$

The attributes mentioned above are just some of the common ones. The exact number and the type of attributes will vary according to the type of pyramid (e.g. regular or irregular) and the kind of abstraction detail required by different applications. Theoretically, we can have as many attributes as it is required. Nevertheless, there is a trade-off between the storage requirement and the detail of abstraction. More attributes mean more detailed image information can be held. But more attributes also translates into higher storage cost.

While the pyramid structure is physically reducing the number of representative data points of the image on each successive higher pyramid level, the existence of the collective attributes enables the abstraction of key image information. Instead of holding and analyzing the full size image, the pyramid structure provides an environment to heuristically abstract the required image information into a smaller size for analysis.

2.4 Types of Pyramid Structure

There are many types of pyramid structure. The two main categories are the regular and the irregular pyramid structure. In terms of the structural layout, the regular pyramid is always assumed to be a square layout/array with an equal number of rows and columns. Although the original input image may not be in any rectangular configuration (i.e. length \diamond width), the processing fundamental is based on a square grid. There are different ways in treating the image boundary for those input images with unequal dimensions. In contrast, an irregular pyramid structure cannot be defined by the dimension of a rectangular array. Due to the irregularity in the contraction of the varying image region, it is not possible to define

the structure according to an overall dimensional width or length of the image. Nevertheless, both types of pyramid structures follow the same general formation process which involves three main components. They are the input image L_i , an output image L_{i+1} and a transformation function T . Using the pyramid data points on the lower level as the input, the transformation function will produce a smaller number of data points on the next higher pyramid level.

$$L_{i+1} = T(L_i) \text{ where } |L_{i+1}| < |L_i| \quad (7)$$

2.4.1 Traditional Regular Pyramid

There are many variations of regular pyramid. The simplest kind is the traditional or non-overlapping regular pyramid structure. The structure and the size of the square array on each pyramid level will depend on a reduction ratio (i.e. R). R is defined as the number of times of reduction in terms of the dimension (i.e. length or width) of the square grid. Figure 5 shows the schematic of a pyramid structure with four pyramid levels. With a reduction ratio of 2, the dimension and the size of the image on each pyramid level are shown in Table 2. The dimension of the square grid on pyramid level i is R times longer than the dimension on level $i+1$. The image size N_i on pyramid level i is R^2 times larger than the image size N_{i+1} on level $i+1$.

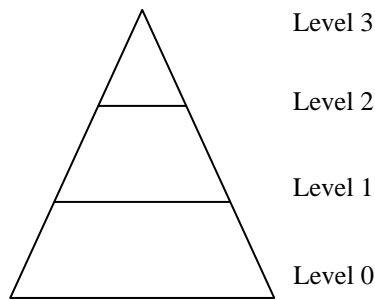


Figure 5. Schematic of a pyramid structure

Table 2. Pyramid dimension and size

Pyramid level	Length x Width	Array Size
3	1 x 1	1
2	2 x 2	4
1	4 x 4	16
0	8 x 8	64

In this structure any non-boundary pyramid data point on an arbitrary pyramid level i , excluding the base and the top pyramid levels, will have a definite R^2 number of children on level $i-1$ and a single parent on pyramid level $i+1$. Figure 6 shows the array layout for the pyramid levels 1, 2 and 3 of the same example as above. As shown by the color and the alphabet in each cell, a data point on level i (i.e. middle array) has four children on level $i-1$ (i.e. left array) and a single parent on level $i+1$ (i.e. right array). This transformation process can be viewed as a mapping process by shifting a local window enclosing groups of neighboring pixels across the lower pyramid image in a non-overlapping manner to produce the data points on the higher pyramid level.

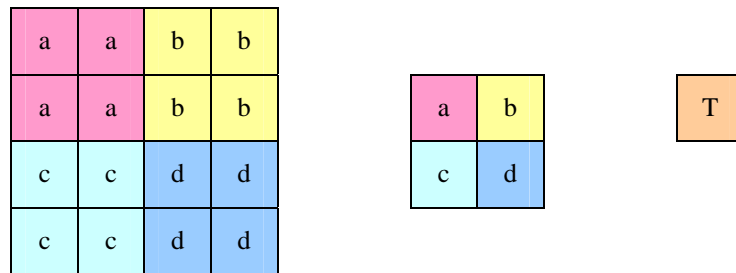


Figure 6. Regular pyramid structure on three levels (left: $i-1$, middle: i , right: $i+1$)

With this regular layout, the exact position of every parent/survivor, child and neighbor within the square grid can be defined precisely. As a result there is no requirement to explicitly create any physical linkage or maintain any children or neighbors list. The data

points on each pyramid level will only have a simple attribute list. They are the unique pixel attribute and the intensity attribute. Depending on how the pyramid formation algorithm is constructed, the value of the unique pixel attribute can either be derived as and when is required or retained within the attribute list of each data point.

$$\overline{D}_{i,j} = \left\{ d_{i,j}^p, \overline{d_{i,j}^y} \right\} \quad (8)$$

The pyramid formation process in this structure is a direct application of a simple transformation function. Based on the parent-child relationships arrangement, the content of a data point on the higher pyramid level is defined by the content of its group of children on the lower pyramid level. The parent's intensity value is derived through the application of a transformation function with the intensities of the children as the input. In [165] where the subject is the binary image, the authors utilize an OR-bit operation as the transformation function where the binary state of the data point on the higher pyramid level is determined by “oring” the binary states of all its children on the lower level. A black pixel appears on level $i+1$ if any of its children on level i is a black pixel. In gray scale images, the most commonly used transformation function is an averaging function where the average of the children intensity values will become the parent's intensity. This can be done with simple averaging [10] or a more elaborate averaging method with a Gaussian-like weighting function as in [7, 8].

In [7, 8] the authors introduce the Gaussian pyramid which is basically the traditional regular pyramid constructed by convoluting the lower pyramid level with a weighting function to produce the higher level. Capitalizing on the regularity of the traditional regular pyramid structure, the authors demonstrate the formation of Laplacian pyramid by the subtraction of successive Gaussian pyramid levels assisted by a series of “reduce” and

“expand” operations. A distributed tracking system based on the traditional regular pyramid structure is described in [10]. The use of multiple processing elements in the formation of the pyramid structure in parallel is demonstrated. Another method in [16] proposes a pyramidal computer architecture based on the traditional regular pyramid structure. The structure is used to perform segmentation of gray scale images by binarizing the image through recursive bottom-up detection of the bimodality within non-overlapping local window. The segmentation task is achieved by image thresholding.

A different application of the pyramid structure is proposed in [77], which utilizes the structure in isolation of background region in outdoor images. The method tries to estimate the dominant background color through the averaging effect in removing all foreground colors while building the pyramid structure bottom-up. With the derived background color threshold, the method will perform a top-down background labeling process by recursively analyzing the colors of all children. No further splitting will occur when a child region is labeled as a confirmed background region. The objective in the separation of foreground and background regions is achieved through the process. The paper proposed in [48] also utilizes such a bottom-up summarization of content, followed by a top-down traversal of the pyramid structure in the identification of text block in document (i.e. newspaper) layout analysis using the regular pyramid model.

Among all types of pyramid structure, the traditional regular pyramid is the most efficient in terms of the formation process. This is mainly due to the regularity of the structure where all the structure parameters are constant and derivable. This has enabled the process to be highly parallel and resulted in many parallel systems based on this structure. Nevertheless it is also the most inflexible structure when it is used in image segmentation. Due to the

rigidity in the horizontal (i.e. neighbors) and the vertical (i.e. parent-child) relationships, this structure will have problem segmenting regions with very irregular shapes and sizes.

2.4.2 Overlapped or Linked Regular Pyramid

The second type of regular pyramid is the overlapped or linked regular pyramid. The fundamental structure is the same as the traditional regular pyramid where the layout on each pyramid level is also assumed to be a square grid with equal contraction along both sides of the dimension. This will result in the same approach to reduce the image size in successive pyramid levels according to a reduction ratio R as shown in Table 2. The main difference is in the requirement to maintain an explicit parent-child linkage. Unlike the traditional regular pyramid where the parent-child relationships are implicitly defined within the structure, the change in the transformation process results in various possibilities in linking the children and parent data points. Instead of mapping the local window across non-overlapping groups of data points, the process will map the data points in an overlapping manner. This will not only alter the number of children, but will also increase the number of possible parents. Depending on the degree of overlap, the number of children and parents will vary. The commonly used 50% overlap (i.e. $\text{span}=2$) regular pyramid structure has 16 children and four potential parents. Figure 7, which is adapted from [9], shows an instance of the mapping layout (i.e. not a complete pyramid level). An arbitrary data point “1” (i.e. yellow color) on the higher pyramid level is derived by transforming a group of 16 children on the lower pyramid level indicated by using the same color and index number “1”. A second data point “2” on level $i+1$ is obtained by also taking 16 children on level i , but it will overlap 50% of the region covered by the children of data point “1”. This will again apply to data point “3” and data point “4”. Due to this overlapping of the children regions, each data point on the lower level will now has four potential parents on the higher level.

Only one of this higher pyramid level data points (i.e. the right array in Figure 7) will be eventually selected as the parent of the children.

1	1	1 2	1 2	2	2
1	1	1 2	1 2	2	2
1 3	1 3	1 2 3 4	1 2 3 4	2 4	2 4
1 3	1 3	1 2 3 4	1 2 3 4	2 4	2 4
3	3	3 4	3 4	4	4
3	3	3 4	3 4	4	4

1	2
3	4

Figure 7. A 50% overlap regular pyramid structure (i.e. left: level i , right: level $i+1$) showing parent-children relationships

In this layout the position of all neighbors and children are still derivable. But the location of the parent will vary depending on which of the four higher level data points are selected as the parent. In order to maintain this varying information, more pyramid data point attributes are added. The new additions are the father attribute $d_{i,j}^f$ and the area attribute $d_{i,j}^a$. The former will indicate the actual selected parent on the higher level and the later will retain the accumulated area of all children on the lower pyramid level.

$$\overrightarrow{D}_{i,j} = \left\{ d_{i,j}^p, \overrightarrow{d}_{i,j}^y, d_{i,j}^f, d_{i,j}^a \right\} \quad (9)$$

Unlike the traditional regular pyramid, this structure has a more elaborate transformation function. The function involves an iterative process which attempts to locate the closest parent for each child. In the following illustration, we will use the minimal gray scale intensity variance to define closeness. There are three key processes. The first is the initial estimate for the intensity value of each data point on the higher pyramid level where the

original parent-child relationship is used. The intensity value of each data point $D_{i+1,k}$ on level $i+1$ is defined as the average intensity of all its children β_r on the lower level i .

$$d_{i+1,k}^y = \frac{\sum_{r=1}^{16} \left(\beta_r^y \mid \beta_r \in \overline{d_{i+1,k}^c} \right)}{16} \quad (10)$$

The second process is the identification of the closest parent and the re-adjustment of the parent-child linkage. For each data point on the lower pyramid level the intensity variation between the child $D_{i,j}$ and each of the four parents δ_s are examined and the lowest variance is identified as shown in equation 11. The process will link the child to the new parent (i.e. updates the child's father attribute $d_{i,j}^f$). In situation where there is more than one parent with the minimum variance, the child will maintain the existing linkage if one of the minima is the old parent or else it will pick any of the minimum parents in random. The third process is to re-compute the intensity and the area attribute of all data points on the higher pyramid level by following the new linkages. The second and the third processes will iterate until there are no more changes in the linkage. In [1] ten iterations are reported to reach the convergence in their test sample.

$$d_{i,j}^f = \left\{ \begin{array}{l} \delta_s^p \mid \min \left(\left| d_{i,j}^y - \delta_s^y \right| \right) \forall \delta_s \text{ where } s = 1 \text{ to } 4 \\ \delta_s \in \text{parent of } \overline{D_{i,j}} \end{array} \right\} \quad (11)$$

In [1] the authors first introduce such a pyramid structure where they called it the linked pyramid structure to perform segmentation and smoothing of gray scale images. Interesting smoothing effect is achieved where smoothing only occurs within the interior of an image region and yet maintaining sharp image region boundary contrast against the background.

As compared to the normal smoothing operation by indiscriminately applying the averaging function to all pixels, the structure enables the isolation of homogenous regions and constraints the smoothing only within the interior of the isolated region. The similar group of authors present another paper in [2], focusing on the image smoothing process and suggest some variations in the initiation of the average intensity for the parent, the selection of parents, ties resolution, processing sequencing and the top level node or root node analysis to improve the transformation process in [1]. A modified version of [1] is also introduced by using weighted linking. The same structure is also used to perform image object boundary extraction in [3] by detecting and linking edges on successive pyramid levels.

A few observations are noted in this type of regular pyramid structure. The first is the disappearance or integration of non prominent regions (i.e. either in terms of size or intensity) into the background. Although the authors in [1] argue that such regions are most likely to be noises, they can also be valid regions. In [2] this effect is solved by changing the size and the layout of the children pixels used to obtain the initial estimate of the parent's gray scale intensity before the iteration by using non-overlapping group of 4x4 pixels. Despite the change, problem remains if a region lies in between two sets of four overlapping windows. If either both or a portion of the region is the minority within its respective averaging 4x4 window, the averaging effect will completely wipe the region off from further representation on the higher pyramid level. The problem is also reported in [4] as the island problem.

The second is the spatial discontinuity of the segmented regions where a single pyramid data point may represent multiple non-connected image objects. This is due to the high emphasis on the gray scale intensity homogeneity and the design in allowing regions that are

of some distance apart to group under the same parent as long as both are “close” in their intensity values. If the non-connected image objects are all standalone unique regions then such hierarchical representation is still acceptable. By appropriate top-down pyramid traversal, the regions are still extractable. Problems occur when fragments of the same image object are grouped under different parents which will happen in situation as described in the second observation. Further analysis and clustering of regions are required to identify the image object. This problem is also illustrated in [14] as the connectivity problem.

The final observation is the determination of the number of segments. Due to the rigidity of the structure where there is a fixed number of data points on each pyramid level. Regardless of the actual number of homogenous segments in the input image, the resulting number of segments on each pyramid level is always the same (i.e. 1, 4, 16, 64 and etc.). A prior knowledge of the number of segments is required in order to stop at the correct pyramid level for the extraction. Although some alternatives are suggested in [2], the requirement to know the number of region types in advance remains. As reported in [14] this limitation in the number of maximal roots will get worst as the image size increases.

Despite the observed problems as shown above, the flexibility of the linked regular pyramid structure as compared to its unlinked counterpart has attracted a lot of attentions and resulted in numerous publications. The majority of the proposed methods attempt to modify the basic linked regular pyramid structure as described above, either aiming to solve some of the problems as observed or suggesting new applications of the structure.

Reference [4] also reports some of the problems as observed earlier. Modification to the linked pyramid is suggested to use unforced linking where data points are allowed to stand alone without any parent if certain standard deviation value is not achieved. This will allow

the formation of root nodes on any pyramid level. Clustering algorithm is then applied to complete the segmentation task. In [5] further modification is done to the original linked pyramid structure by introducing weight that takes into consideration of the geometrical closeness between regions to derive the average intensity of the parent data points. Another modification is introduced in [6] by using both gray scale and edge information to produce three pyramid representation to achieve the final segmentation of the compact region. In a later year, a group of authors re-examine the linked pyramid structure in the context of region matching [9] where their approaches vary in the bounding of the number of iterations for the closest parent and the provision to prevent linking of non connected components. Reference [13] addresses the image boundary problem that occurs in all regular pyramids (i.e. pixels along the image border that cannot fit into the fixed mapping window). The solution is by increasing the overlap span at the last few iterations for the nearest parents. The paper also suggests ways in using a gray scale level and size threshold in the detection of root node. In [12] a modified version of the linked pyramid is used in the segmentation of images with very low signal-to-noise ratio. The literature in [15] also makes use of the structure in the detection and delineation of dot clusters by combining several segmentation results of the shifted version of the cluster.

As compared to the traditional regular pyramid, this structure has added the flexibility in the vertical relationships to allow parent-child linkages. This has enabled a better adaptation to the homogeneity criteria required in segmentation. Nevertheless the configuration of having a fixed horizontal relationship still results in some problems preventing its effective use for general image segmentation. Further problems are reported in [14]. The authors have illustrated the inability of the structure to segment objects of varying shapes and sizes; especially elongated objects. The shift-variant problem also prohibits such structure from being used as a general segmentation algorithm.

Chapter 3

Irregular Pyramid

In this chapter we will first review the two irregular pyramid models and some of the subsequent variants. As all existing irregular pyramid models were not designed for document image analysis, we will next propose a general framework of irregular pyramid for the use in text analysis in document images. Detailed mechanisms for the pyramid construction will be discussed. This framework will provide the ground work for subsequent research in the text segmentation in binary, gray scale and color document images. The discussions in the ensuing chapters on these research issues are built up on this ground work. Unlike the regular pyramid, irregular pyramid has attracted less attention in the research community. There are only a handful of proposed unique irregular pyramid models. There are also not many applications making use of the irregular pyramid structure. The two basic models are the stochastic and the adaptive irregular pyramid models. They have very similar pyramid formation process. The key difference is in the assignment of the surviving value for the decimation or the survivor selection process. The former utilizes a uniformly random number while the later derives the value from the image content. They are also other variants of these two models with slight variations in the pyramid formation process.

3.1 Types of Irregular Pyramid

The first irregular pyramid, known as the stochastic image pyramid, was proposed by Meer in 1989 [52]. The main objective is in the establishment of a pyramid model which requires only local information to enable parallel processing. The model makes use of the assignment of uniformly distributed random number to each data point to enable the

selection of local maxima data points (i.e. survivors) from the lower pyramid level to form the sampling grid of the higher pyramid level. The introduction of the local maxima concept with the criteria that no two neighbors survive together and that at least one survivor exists in the neighborhood of a non-survivor becomes the basis for the survivor selection process in all later irregular pyramid models. The idea of the non-survivors linking to the largest neighboring survivor (i.e. having the highest random number or surviving value) is also a frequently adopted convention in subsequent models. The main problem with this method, as pointed out by some of the later papers [14, 53, 55] is in its requirement to change the status of the initial selected survivors in order to meet the two local maxima selection criteria.

Based on the concept of the stochastic image pyramid, Montanvert et al [53, 55], conceptualized the irregular pyramid formation as a graph contraction problem where the selected vertices forming the higher pyramid level constitute the maximum independent set. Their work establishes the detailed pyramid construction steps and utilizes two binary state variables to address the need to make further adjustment to the status of the selected survivors as in the stochastic pyramid. Just like the stochastic image pyramid, the selection of the survivors is based on a stochastic decimation process using a random number. It differs from the previous model with the addition of a class criterion. The definition of class facilitates the survivor selection process where only neighbors of the same class can participate in the evaluation of the local maxima. This criterion also applies to the child selection process. The allocation of the non-survivors to a survivor will depend on two conditions. First, the survivor is the largest if there is more than one survivor in the neighborhood. Second, both non-survivor and survivor nodes must be of the same class. This modification has enabled the application of the model to segment binary and gray scale images. The paper also reveals several key problems faced by the model. The use of

different class definitions will directly affect the rate of convergence (i.e. pyramid contraction rate). The random selection of survivors yields varying sub-graphs and thus produces non-deterministic final results. The paper also discusses the problem in the merger of potentially “suitable” roots that may represent different homogenous regions. For the last problem the authors have analyzed various alternative solutions and proposed an evaluation method to examine the derived threshold value in successive pyramid levels for the determination of the root node. The uniqueness of this model is in its survivor and child selection which will depend on the outcome of a random value.

An alternative to the stochastic model is the adaptive irregular pyramid model proposed by Jolin and Montanvert [57] where the main difference is in the assignment of the surviving value. Instead of using a uniformly distributed random number as the surviving value, the model utilizes an interest operator to derive the surviving value from the image content. This has enabled the authors to apply one of the organizational principles such as the grouping by similarity. The paper uses the gray level variance as the surviving value. The survivor selection will be based on the determination of local minima of gray level variance. Once all such local minima are determined, the child selection process will start by allocating the non-survivors to a survivor with the closest mean gray level. The main idea in this model is to locate the least variance area in the image, which is usually the interior core region of an object, and to slowly allow the surrounding higher variance regions to join the core region as its children. This model has moved away from the stochastic nature of the previous model and allows a more directed pyramid construction process towards specific requirements by mean of content-based survivor and child selection.

In addition to the above works, Montanvert and Bertolino [58] also attempted to tackle the segmentation issue within the pyramid model as a region merging process where similar

adjacent regions, represented by neighboring vertices, are merged. The method also utilized the stochastic model in its decimation process with the help of a similarity measure according to the gray scale differences among adjacent vertices. The authors carried a step further to incorporate the contour information into the pyramid formation process for the segmentation task. Finally the paper has suggested an iterative process to measure the segmentation result on each pyramid level by computing the quadratic error between the pixels of the formed region and the pixels of the original image. The process can be repeated with varying Maximum Independent Set until the best result is obtained.

A generalized segmentation method utilizing irregular pyramid model is also introduced by the same authors [63]. The segmentation task is not achieved through the control of the bottom-up pyramid formation process, but rather by the top-down traversal of the already formed pyramid structure. Each pyramid data point, representing a group of pixels in the original image, constitutes a potential homogenous region. A homogenous region is defined as a region whose standard deviation of a certain attribute S is below a threshold value σ_m . If a homogenous region is not found, the process continues to examine all the children on the next subsequent lower pyramid level. A recursive and iterative algorithms are presented. The paper also proposes two possible definitions of the attribute S (i.e. a geometrical and an intensity property). As compared to the other proposed pyramid-based methods to handle image segmentation, it has postponed the root node detection task after the pyramid is formed with an additional top-down detection process.

Besides image segmentation, the irregular pyramid model is also used in other areas of applications. In [64] Elias and Laganier described the creation of a disparity pyramid in the stereoscopic image analysis. The paper followed the adaptive irregular pyramid model. The minimum disparity values between pixels of the two images under study (i.e. summation of

the intensity differences within a window) are used as the main pyramid data content. Survivor selection is based on the identification of the lowest disparity value in the neighborhood. The parent-child linkage is based on the evaluation of the nearest surviving cell in terms of the Euclidean's distance between the disparity vectors associated with the minimum disparity value (i.e. intensity variation). Since the aim is in disparity estimation, the pyramid root node is defined as one with a distance larger than a minimum distance threshold and of a certain minimum size (i.e. guarding against noise region). All root nodes will always survive with no interaction with the other nodes. The pyramid formation process will stop when all survivors are root nodes.

Several researchers focus their attention on the efficiency of the irregular pyramid formation process. Since the decimation process or the survivor selection process is one of the key contributors to the amount of the computation load, most of these papers have attempted to propose solutions to streamline this process. For instance, Horace et al [62] focus the attention on the irregular pyramid decimation process where two alternatives in the selection of the survivors (i.e. specified rate sampling and prioritized sampling) are introduced. The first involves a specified sampling rate, such that the usual iterative requirement in the survivor selection process is discarded. The method attempts to regularize the decimation ratio across regions by an adjustable sampling rate which can be controlled by the application. The second method uses prioritized sampling by giving a higher priority to those nodes with a larger number of neighbors to survive. The aim is to increase the number of non-survivors, decrease the number of survivors and eventually reduce the pyramid height. This will in turn increase the speed in the pyramid formation. The paper also introduces a new scheme by reducing the number of edges required to be examined while linking the non-survivors to the most similar survivor.

In a more recent paper [69], Jolion investigates the idea of discarding the iterative survivor selection process, aiming to increase the processing speed. The author re-examines the stochastic pyramid decimation process by relaxing the two rules traditionally used in governing the survivor selection process (i.e. No two neighbors survive together and the non-survivors must at least have a neighboring survivor). This approach has coincided and supported our suggestion presented in some of our earlier papers [65, 66, 67] in the need to violate the traditional selection rules to prevent non-promising regions from participating in the survivor selection process, but with a different motivation. Jolin's method (i.e. data driven decimation process) focuses the attention on the identification of the true local maximum and discards the need to iteratively locate other sub-optimal maxima on the same pyramid level. This will result in the survival of nodes which are the true local maximum as well as nodes with no local maximum in its neighborhood. The arrangement to allow the later type of nodes to survive has violated the two decimation rules and at the same time increases the number of survivors. In contrast to the prioritized sampling method as proposed by Horace et al [62], the height of the pyramid may thus increase. Nevertheless, through some experiments the proposed model is shown to execute faster than the stochastic pyramid model without much increases in the pyramid height. The disadvantage of the new model is its higher neighborhood count, thus incurring higher memory cost.

As can be seen from above, the majority of the irregular pyramid related papers mainly revolve around the structure and its formation issues. Not many have touched on the actual application of the structure. Only a few have attempted to apply the structure in the area of general segmentation. Most of these applications are just merely samples to illustrate the formation of the structure. The benefit of using irregular pyramid model, especially in its local processing, hierarchical abstracting, content adapting, natural aggregating of image properties and the heuristic criteria application ability have yet to be explored in full. No

paper has yet explored the detail to control the pyramid formation process for textual segmentation objective. This thesis will thus focus its attention on these areas. The next section will describe in detail a general irregular pyramid construction frame work where the subsequent chapters describing our proposed methods in solving the textual segmentation problem in binary, gray scale and color document images will follow.

3.2 Irregular Pyramid Construction Process

There are three main components involved in building an irregular pyramid structure. They are the input image L_i on a lower pyramid level, the output image L_{i+1} on the higher pyramid level where the number of data points in L_{i+1} is less than L_i and finally a transformation function T summarizing information on level i and producing a smaller set of data points to represent the image on level $i+1$.

$$L_{i+1} = T(L_i) \text{ where } |L_{i+1}| < |L_i| \quad (12)$$

As we can see, the key component is the function T . This function will determine the way and the rate of contraction for the input image. In an irregular pyramid the transformation function T is divided into three main steps. First is the selection of neighbors to explicitly determine the list of neighboring data points due to the irregularity of the regional structure. With reference to the pyramid model, this is an adjacency function. The second step is the survivor selection process where data points satisfying the ‘survive’ function are selected as the survivors or data points on the next higher pyramid level. The final step is the selection of children where non-survivors on the lower pyramid level are selected as the children of a survivor on the higher pyramid level (i.e. the child function). This will ensure a linkage from any pyramid level down to its original base image. The pseudo code for the entire construction process is shown in Figure 8.

```

1. Construct next level = true
2.  $i = 1$ 
3. While (Construct next level)
4. {  $L_i = \text{Create a new pyramid level}(L_{i-1})$ 
5.   Select neighbors( $L_i$ )
6.   Select survivors( $L_i$ )
7.   Select children( $L_i$ )
8.   Construct next level = Not StopEvaluation( $L_i$ )
9.    $i = i + 1$ 
10. }

```

Figure 8. Pyramid construction process.

3.2.1 *Creating a New Pyramid Level*

Pyramid construction will begin from the base level (i.e. L_0) where the original document image is used as the input. Each image pixel is represented by a single pyramid data point on the base level. The data point will retain the pixel intensity and has an area of 1. For the higher pyramid level (i.e. L_i where $i > 0$), all data points selected as the survivors on the immediate lower level are used to form the higher pyramid level. Other than the unique pixel index attribute d^p that remains unchanged (i.e. original pixel index), all the other attributes will reflect the collective value of its children on the lower pyramid level. The area attribute d^a is updated as the summation of all children's areas. The intensity attribute d^v will depend on the type of image format (i.e. binary, gray scale or color) and the averaging function with all the children intensities as the inputs. For a binary image there will not be any intensity attribute. For gray scale images two examples of the intensity averaging functions are shown below. Equation 14 shows a simple averaging function by calculating the intensity mean of all the children. Equation 15 takes the regional area of each child into consideration while computing the mean intensity. Larger child regions will have a higher contributing factor towards the final intensity value. This is a reasonable setting, as smaller regions may just be noise. Nevertheless, this assumption also relies on the correct segmentation of regions where small valid region may end up being absorbed by the larger

background region. The derivation of the intensity value is not restricted to the use of averaging function. The use of minimal, maximal, median or the mode function are also some other possible methods. For example in color images, due to the use of certain color space, averaging may not be suitable. The mode function is a more applicable method by selecting the most frequently occur color among the children.

$$AveIntensity(\overrightarrow{D_{i,j}}) = d_{i,j}^y \quad (13)$$

$$d_{i,j}^y = \frac{\sum_{r=1}^{N_c} (\beta_r^y)}{N_c} \text{ where } \beta_r \in \overrightarrow{d_{i,j}^c} \quad (14)$$

$$d_{i,j}^y = \frac{\sum_{r=1}^{N_c} (\beta_r^a \beta_r^y)}{\sum_{r=1}^{N_c} (\beta_r^a)} \text{ where } \beta_r \in \overrightarrow{d_{i,j}^c} \quad (15)$$

3.2.2 Selecting Neighbors

Once the new pyramid level is created, the first main step is the determination of neighborhood (i.e. line 5 in Figure 8). Since the original input image has a rectangular coordinate layout, either the 4-connectivity or 8-connectivity criterion can be used to build the neighborhood list of each data point. Through experiments, the 8-connectivity criterion is chosen as 4-connectivity will result in jagged boundaries. Thus the adjacency function for the definition of neighborhood list $\overrightarrow{d_{i,j}^b}$ is as follow.

(r-1,c-1)	(r-1,c)	(r-1,c+1)
(r,c-1)	(r,c)	(r,c+1)
(r+1,c-1)	(r+1,c)	(r+1,c+1)

$$\begin{aligned}
d_{i,j}^p &= (r, c) \\
8cc(d_{i,j}^p) &= \bigcup_{s=-l}^{s=l} (r+s, c+s) \\
adjacent(\overrightarrow{D_{i,j}}, \alpha_q) &= \{ \alpha_q \in L_i, \alpha_q \in 8cc(d_{i,j}^p) \}
\end{aligned} \tag{16}$$

The neighborhood list of a data point $D_{i,j}$ with pixel index $d_{i,j}^p$ positioned at row index r and column index c will comprise of all the eight connected data points sharing the same boundary as $D_{i,j}$. The majority of the interior data points will have the same number of neighbors (i.e. $N_b=8$), except for data points lying along the image boundary. Data point positions at one of the four image corners will have only three neighbors. The remaining data point positions along the boundary will have exactly six neighbors. Unlike the regular pyramid, boundary problem does not exist in irregular pyramid due to the flexibility of its structure to fit any size or shape of the input content.

On the higher pyramid level the regularity of rectangular coordinate will no longer exists. The formation of regions will become very irregular and thus have no fixed boundary border as in the base level for the 8-connectivity analysis. In order to determine whether two pyramid data points are neighbors, the algorithm has to analyze their children. Two data points $D_{i+1,k1}$ and $D_{i+1,k2}$ on pyramid level $i+1$ are neighbors if their children β_1 and β_2 on level i are neighbors.

$$adjacent(\overrightarrow{D_{i+1,k1}}, \overrightarrow{D_{i+1,k2}}) = \begin{cases} \exists \beta_1 \mid (\beta_1 \in L_i) \wedge (\beta_1 \in \overrightarrow{d_{i+1,k1}^c}) \\ \exists \beta_2 \mid (\beta_2 \in L_i) \wedge (\beta_2 \in \overrightarrow{d_{i+1,k2}^c}) \\ adjacent(\beta_1, \beta_2) \end{cases} \tag{17}$$

Based on the above criteria, there are three possible situations as shown below. The first situation is the most straightforward where both the data points on level $i+1$ are the direct promotion of the two neighboring data points which survive on level i . Since both are neighbors on the lower level i , they will remain as neighbors on level $i+1$. This is illustrated in Figure 9. Figure 10 demonstrates the second situation where both survivors are not the direct neighbor on the lower level, but they are linked (i.e. neighborhood link) through a third data point that is the child of either survivor. The last situation is when the children (i.e. y_1, y_2) of both survivors are neighbors in Figure 11.

$$adjacent(\beta_1, \beta_2) = \begin{cases} \text{case1: } (\beta_1^p = d_{i+1,k1}^p) \wedge (\beta_2^p = d_{i+1,k2}^p) \\ \text{case2: } ((\beta_1^p = d_{i+1,k1}^p) \wedge (\beta_2^p \neq d_{i+1,k2}^p)) \vee \\ \quad ((\beta_1^p \neq d_{i+1,k1}^p) \wedge (\beta_2^p = d_{i+1,k2}^p)) \\ \text{case3: } (\beta_1^p \neq d_{i+1,k1}^p) \wedge (\beta_2^p \neq d_{i+1,k2}^p) \end{cases} \quad (18)$$

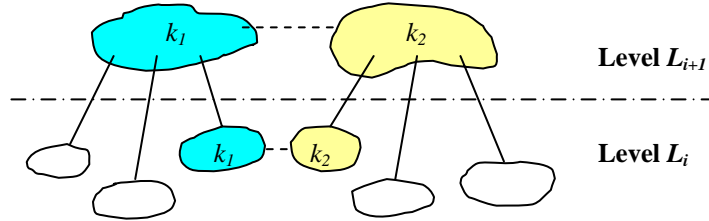


Figure 9. Case 1: Both children are the survivors on the lower pyramid level

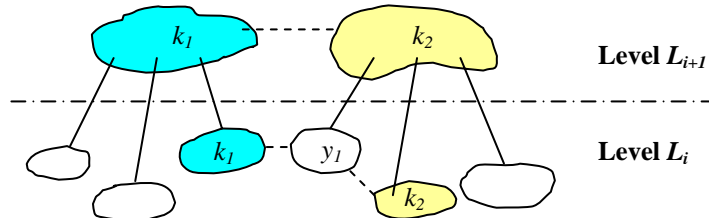


Figure 10. Case 2: Adjacency is defined through a 3rd region, which is either a child of k_1 or k_2

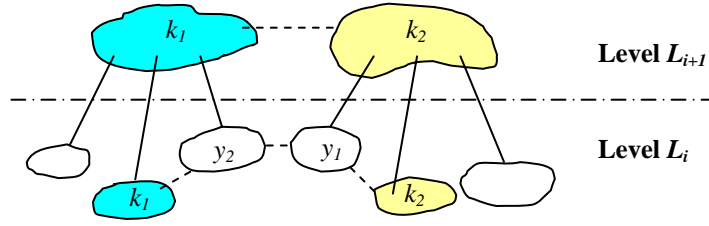


Figure 11. Case 3: Adjacency is defined through a 3rd and 4th regions

3.2.3 Selecting Survivors

The second step is the selection of survivors. This is the process in determining those data points that will survive to form the next higher pyramid level. Unlike the regular pyramid where the surviving candidates are fixed, the irregular pyramid requires explicit selection for the survivors. Based on the designated survivor selection process and criteria, appropriate data points are selected as the survivors. This is one of the key steps in the pyramid formation process. It is also one of the main factors that allow the irregular pyramid structure to have the maximum flexibility to adapt to the variability of the input content that the regular pyramid cannot fulfill [53]. The selection of an optimal set of survivors is required to yield ‘good’ resolution abstraction of the lower pyramid level. An appropriate set of survivors will support the suitable growing points (i.e. seed points). The maintenance of a low pyramid height to enhance processing speed can also be achieved by a careful selection of survivors with more neighbors [62].

α_1	α_2	α_3
α_8	$D_{i,j}$	α_4
α_7	α_6	α_5

$$\overrightarrow{D_{i,j}} = \{ \dots, d_{i,j}^s, \dots \}$$

$$survive(\overrightarrow{D_{i,j}}) = \begin{cases} d_{i,j}^s > \text{Max}(\alpha_q^s) \text{ for } q = 1 \text{ to } N_b \\ \alpha_q \in \text{neighbors of } \overrightarrow{D_{i,j}} \end{cases} \quad (19)$$

The above shows one of the most commonly used survivor selection process which is first proposed in [53]. It is the determination of a local maximum as the surviving data point. Without loss of generality, other criterion likes the local minimum is also a possible selection criterion. In order to facilitate the selection process, an addition attribute is required for each pyramid data point. It is called the surviving value (i.e. d^s). The value can simply be a random number, the gray scale value of the pixel or any derived value depending on the specific application requirement. Data point with the highest surviving value among all its neighbors α_q is considered as a local maximum and is thus eligible to survive. Unlike the regular pyramid where data points on fixed regular interval are selected as the survivors, this process facilitates the flexibility in allowing the most suitable data point to survive. It also has the advantage of using only local information (i.e. the immediate neighboring data points) in the analysis process and thus enables parallelism due to the use of only local information.

Almost all the proposed irregular pyramid models utilized this as the survivor selection process with some form of variations. The main drive of this process is based on the computation requirement. The process attempts to attain maximum decimation and yet to have enough remaining data points yielding a good representation of the current pyramid level on the next higher pyramid level. It can be viewed as a process trying to select a maximum independent set (i.e. MIS) where the pyramid data points representing the vertices and their neighboring relationships standing as the edges between vertices. The extraction of the maximum independent set will produce a maximum number of unique pyramid data points forming the next higher pyramid level. In order for this process to work, two rules or conditions as shown below have to be satisfied.

1. No two neighbors will survive together.
2. Non-survivor will have at least one neighboring survivor.

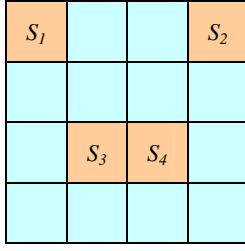


Figure 12. Violating rule 1

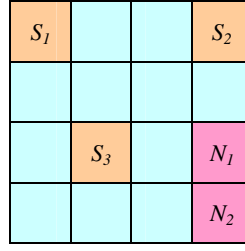


Figure 13. Violating rule 2

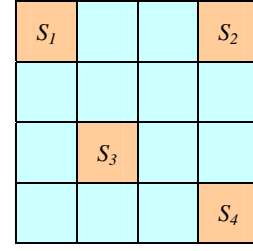


Figure 14. MIS

Both conditions are to ensure the satisfaction of MIS extraction. The first condition ensures that no edges from level i will appear on level $i+1$. Since no neighboring data points or vertices can survive together, it is guaranteed that no old edges will ever appear on the new pyramid level. The existence of old edges will result in the selection of dependent pairs of vertices. Figure 12 shows an example of the violation of rule 1 where the data points S_3 and S_4 survives together and thus carrying the old edge between the two points. The second condition is to ensure that every non-survivor will have at least one survivor in its neighborhood and thus to guarantee that no vertex is left out from the MIS. In Figure 13 due to the non-existence of any survivor in the neighborhood of N_1 and N_2 , the selected three survivors do not constitute a MIS. Figure 14 shows a complete MIS where both conditions are satisfied. The four survivors S_1 , S_2 , S_3 and S_4 will make up a suitable representation of all the 16 data points. A paper [62] even proposed a repetitive computation process to obtain the optimal MIS.

Despite the ability of the process to extract the MIS, problems occur when there is no real local maximum in the neighborhood. Due to the failure to locate some of the data points as a local maximum as the survivor, certain non-survivors will have no neighboring survivor as shown in Figure 13. Many papers [57, 62, 69] have reported this problem. The problem is less obvious when there is a high degree of randomness in the surviving values among the neighbors as in the stochastic model where uniformly distributed random number is used as the surviving value. The problem becomes more apparent in the adaptive pyramid model

when the image content is used as the surviving value where there are more consistency and continuity in the surviving value among neighbors as seen in Figure 15, Figure 16 and Figure 17. The figures show three examples of a 5x5 image, demonstrating the problem of the absence of real maximum. The displayed numbers are the surviving values which can represent the actual gray value of the pixel or the pixel gray value variance among its neighborhood.

4	5	6	5	4
5	7	8	7	5
6	8	9	8	6
5	7	8	7	5
4	5	6	5	4

Figure 15. Gaussian

5	6	7	8	9
5	6	7	8	9
5	6	7	8	9
5	6	7	8	9
5	6	7	8	9

Figure 16. Stepwise

5	5	5	5	5
5	5	5	5	5
5	5	5	5	5
5	5	5	5	5
5	5	5	5	5

Figure 17. Flat

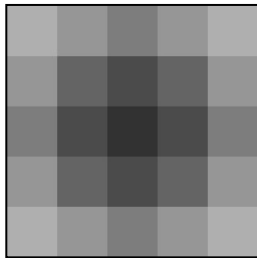


Figure 18. Gaussian

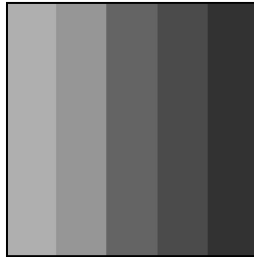


Figure 19. Stepwise



Figure 20. Flat

The first image (i.e. Figure 15) is a Gaussian like pattern. It has only one local maximum (i.e. 9). All other data points have at least a neighboring data point having a surviving value larger than itself and thus cannot be a local maximum. Figure 16 and Figure 17 show a stepwise and a flat pattern image. Both have no local maximum and thus no survivor. Without a survivor the pyramid construction process cannot derive the next pyramid level. Figure 18, Figure 19 and Figure 20 show the corresponding graphical representation of each example. The image pattern in Figure 18 illustrates a very common occurring pattern in document images where a slowly fading gray scale level occurs surrounding the core subject

region. Figure 19 shows a stepwise pattern that usually occurs at the boundary of an image object. Figure 20 represents the background region of an image which is usually the largest area.

The solution to this problem is an iterative selection process. The key to the process is to mark the neighbors of the selected local maximum as the confirmed non-survivors. This is to prevent them from participating in the evaluation of other local maxima in the next iteration. Figure 21 shows the flow of the process.

```

1:  $k = 0$ 
2:  $d_{i,j}^{suv}(k) = d_{i,j}^{xsuv}(k) = 0 \quad \forall \overrightarrow{D_{i,j}}$ 
3: Do
4: { For each data point  $\overrightarrow{D_{i,j}}$  where  $\overline{d_{i,j}^{suv}(k)} \wedge \overline{d_{i,j}^{xsuv}(k)}$ 
5:   { If  $d_{i,j}^s > \max(\alpha_q^s \cdot \overline{\alpha_q^{xsuv}(k)}) \mid \alpha_q \in \overline{d_{i,j}^b} \quad \forall q = 1 \text{ to } N_b$ 
6:     {  $d_{i,j}^{suv}(k) = 1$  }
7:   }
8:    $k = k + 1$ 
9:   For each data point  $\overrightarrow{D_{i,j}}$ 
10:  { if  $d_{i,j}^{suv}(k-1) = 1$ 
11:    {  $d_{i,j}^{suv}(k) = 1$ 
12:      For each neighbor  $\alpha_q \in \overline{d_{i,j}^b}$  {  $\alpha_q^{xsuv}(k) = 1$  }
13:    }
14:  }
15: } Until  $(d_{i,j}^{suv}(k) \vee d_{i,j}^{xsuv}(k)) \quad \forall \overrightarrow{D_{i,j}}$ 

```

Figure 21. The local maxima selection process

The selection process will maintain two status flags. The first (i.e. $d_{i,j}^{suv}$ or α_q^{suv}) will indicate whether the current data point is selected as a local maximum or survivor. The second (i.e. $d_{i,j}^{xsuv}$ or α_q^{xsuv}) will reflect whether the data point is a confirmed non-survivor. A

data point is a confirmed non-survivor when one of its neighbors is selected as a survivor. Both flags are initialized to false at the beginning of the process (i.e. line 2). The process will iterate through each pyramid data point which is neither a survivor (i.e. $d_{i,j}^{surv} = 1$) nor a confirmed non-survivor (i.e. $d_{i,j}^{xsurv} = 1$). A data point is considered as a local maximum if it has the largest surviving value among all its eligible neighbors α_q . A neighbor is eligible to participate in the selection process if it is not a confirmed non-survivor where its α_q^{xsurv} is false (i.e. there is no survivor in its neighborhood). Once a data point is determined to be the local maximum, its survivor status flag $d_{i,j}^{surv}$ is switched to true (i.e. line 6). In the preparation for the next iteration, a surviving data point will remain as a survivor (i.e. line 11) and all its neighboring data points will become the confirmed non-survivors (i.e. line 12). This will prevent the neighboring data points of a survivor from participating in the selection of other local maximum in the next iteration. The entire process will repeat until all data points become either a survivor or a confirmed non-survivor (i.e. line 15).

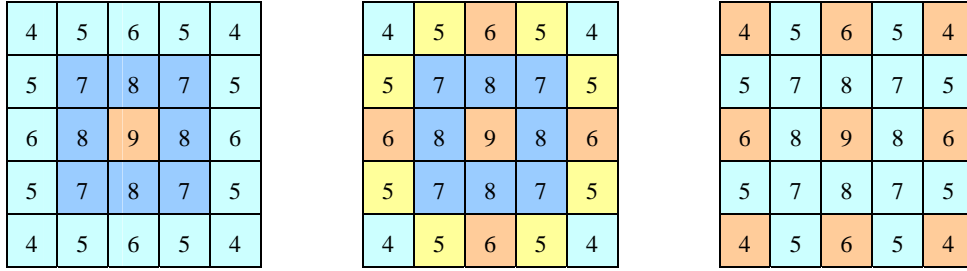


Figure 22. Application of the local maxima selection process to the image in Figure 15

Figure 22 demonstrates the flow of the process using the sample in Figure 15. Only one local maximum (i.e. 9) is selected in the first iteration (i.e. left). The neighboring data points (i.e. 7, 8) are marked as confirmed non-survivors. This will enable the second iteration to pick the four survivors having the surviving value 6. The final iteration will select the

remaining data points at the four corners to be the survivors with no contender to be the local maximum.

Although the above solution has solved the problem of absence of a true local maximum by allowing the selection of a sub-optimal maximum, it has created a new problem where the number of iterations (i.e. 3 for the above example) to reach stability is non-deterministic. There is no way to control and determine the number of iterations to reach full participation by all data points to be classified as either a survivor or a non-survivor. As reported in Meer's the stochastic model [52], through some experiments the number of iterations to reach stability is at most five. In the adaptive model by Jolin et al [57], however, depending on the assignment of the surviving value, it requires a greater number of iterations in the first few pyramid levels. In terms of the computation time, the iterative process does contribute towards the slowing down of the pyramid formation process. Jolin [69] has suggested to totally discard the iterative process and to postpone the selection of the sub-optimal maximum to the higher pyramid level. Despite the problem it is a suitable survivor selection process making use of only local information.

Another problem remains in Figure 16 and Figure 17 where no local maximum can be selected. This is due to the way the local maximum is defined as the largest surviving value (i.e. $d_{i,j}^s > \max(\alpha_q^s)$). The simplest solution is to allow an equal condition for the local maximum (i.e. \geq). This will allow data points with an equal largest surviving value to be selected as the survivor. Figure 23 shows the selection process for the local maximum in three iterations for the image in Figure 16. The application of the same process to Figure 17 will result in the same number of local maximum selected as the survivors. An alternative is to pre-process the input image before the pyramid construction process where all identical neighboring pixels are merged as a single component. With this process, the input image

will no longer be in a regular pixels format. As a result this will involve the alteration of the pyramid formation process to begin the formation from an intermediate pyramid level rather than the pyramid base level. The detail of this process is further discussed in Chapter 7.



Figure 23. Application of the local maximal selection process to the image in Figure 16

The idea of obtaining and using the MIS to represent the abstraction of an image is a very fundamental concept. Further modification in the process and the features are required in specific applications. This is achieved by a careful selection of the appropriate surviving value and the designation of suitable selection criteria. In our context of text segmentation and extraction, the survivor selection process will need to be very precise and robust. As compared to the segmentation of graphical objects, the textual objects are much smaller in terms of its size and are positioned closer together and thus easily affected by the neighboring noise. An accurate identification of suitable surviving points are required to support proper grow of region into textual object. In our proposed methods a few changes are made to this tradition model. The first is the derivation of the surviving value. Depending on the specific application the design of the surviving value is altered accordingly. In the processing of the binary document images the main objective is to allow the growing of the textual region starting from the inner core of the character or word. Thus the regional mass (i.e. number of foreground pixel) is used as the surviving value. Pyramid region which encloses a larger foreground area constitutes a better surviving region or seed point. In the gray scale images, two different types of surviving values are utilized

depending on the growing stage on different pyramid levels. The uses of the gray scale intensity variance among neighboring regions as the surviving value on the base pyramid level is to locate good textual boundary. On the higher pyramid level the surviving value is changed to reflect the number of neighbors which show potential of being the word's fragment. This has demonstrated the flexibility of the structure to accept different criteria on different pyramid levels. In the color document images, the surviving value is derived by collecting the number of votes from the neighbors that satisfy a set of criteria.

The second change to the survivor selection process is the violation of the two selection rules as stated above. The motivation of the alteration is mainly based on a practical issue. Since the majority of the region in the document image is the background region where our objective is the textual region. It will be a waste of storage space and processing time to allow these regions to propagate from pyramid level to level. In addition the existence of these regions may also interfere and affect the accurate growing of the textual region. There is also situation when a region already formed the target object and no further growing is necessary. As a result some regions may not be allowed to participate in the survivor selection process and thus there will be situation where either or both the selection rules will be violated. The details are shown in Chapter 4, 5, 6 and 7.

3.2.4 *Selecting Children*

The third step is the selection of children. This is the process in establishing the parent-child linkage. In the pyramid structure, the selected survivors are used to form the higher pyramid level and its neighboring non-survivors will remain on the current pyramid level. In order to maintain the linkage between the successive pyramid levels, each survivor on the higher pyramid level is linked to a group of non-survivors on the lower pyramid level. The former is called the parent and the later are the children of the parent. Unlike the regular

pyramid where such vertical relationship is implied within the model, irregular pyramid structure has to explicitly construct the relationship. This explicit establishment of the relationship is the other key factor, besides the selection of survivor, in allowing the flexibility of the irregular pyramid to adapt to the input content. Based on the concept of decimating, a correct parent-child linkage will allow the top-down traversal of the pyramid structure to perform finer analysis of the area of interest. Based on the concept of growing, the linking of the survivor to its neighboring non-survivors is comparable to the growing of a survivor or the seed into its surrounding regions.

There are two alternatives for this process. The first alternative is shown in Figure 24. It is the most frequently used traditional selection scheme. It allows the non-survivors to initiate the linking task. Each non-survivor will attempt to link to its largest neighboring survivor (i.e. λ). Without loss of generality, this child selection criterion can also be based on the minimum or the closest in terms of any type of measurements (e.g. gray scale level). The process will evaluate all the neighbors α_q of each non-survivor and locate a survivor λ with the highest surviving value or meeting the selection criteria. If such a survivor exists, the non-survivor $D_{i,j}$ will assign itself to the child list of the identified survivor $d_{i+l,k}^c$.

```

1: For each data point  $\overrightarrow{D_{i,j}}$ 
2: { If Not survive( $\overrightarrow{D_{i,j}}$ )
3:   { If  $\exists \lambda \mid \lambda = \max(\alpha_q^s \in \overrightarrow{d_{i,j}^b} \wedge \text{survive}(\alpha_q) \forall q = 1 \text{ to } N_b)$ 
4:     {  $\exists \overrightarrow{D_{i+l,k}} \mid d_{i+l,k}^p = \lambda^p$ 
5:        $\overrightarrow{d_{i+l,k}^c} = \overrightarrow{d_{i+l,k}^c} \cup \overrightarrow{D_{i,j}}$ 
6:     }
7:   }
8: }
```

Figure 24. First child selection process

The second alternative is our own modified version as shown in Figure 25. The scheme is to use the data points on the newly formed pyramid level or the selected survivors as the focal points and to allow the survivors to initiate the linking to the surrounding non-survivors. For each data point $D_{i+1,k}$ on the pyramid level $i+1$ there is a corresponding data point $D_{i,j}$ on the lower pyramid level i , which is the survivor. The first child for $D_{i+1,k}$ will be the corresponding data point $D_{i,j}$, which is itself (i.e. line 3). The process will then iterate through all the neighboring points of $D_{i,j}$ and include any neighbors α_q which has no other survivors in its neighborhood in the child list (i.e. lines 8). In situations where there are other neighboring survivors λ , α_q will only be included in the child list if $D_{i,j}$ has the largest surviving value (i.e. line 5,6). In order to avoid circumstances where a non-survivor has neighboring survivors with equal surviving values and as a result becomes an orphan, the equal condition is included in line 6. The non-survivor will link to the first encountered survivor.

```

1: For each data point  $\overrightarrow{D_{i+1,k}}$ 
2: {  $\exists \overrightarrow{D_{i,j}} \mid survive(\overrightarrow{D_{i,j}}) \wedge d_{i,j}^p = d_{i+1,k}^p$ 
3:    $\overrightarrow{d_{i+1,k}^c} = \overrightarrow{D_{i,j}}$ 
4:   For each neighbor  $\alpha_q \in \overrightarrow{d_{i,j}^b} \forall q = 1 \text{ to } N_b$ 
5:     { If  $\exists \lambda \in \overrightarrow{\alpha_q^b} \wedge survive(\lambda) \wedge \lambda \neq \overrightarrow{D_{i,j}}$ 
6:       { If  $d_{i,j}^s \geq \max(\lambda^s \forall \lambda)$  {  $\overrightarrow{d_{i+1,k}^c} = \overrightarrow{d_{i+1,k}^c} \cup \alpha_q$  } }
7:       Else
8:         {  $\overrightarrow{d_{i+1,k}^c} = \overrightarrow{d_{i+1,k}^c} \cup \alpha_q$  }
9:     }
10: }
```

Figure 25. Second child selection process

Both alternatives will achieve the objective in the identification of the most appropriate surrounding neighbors on the lower pyramid level to become the children of a parent on the

higher pyramid level. Nevertheless, they differ in their computation complexity and the uniqueness in the processing order. Between the two alternatives, the first way is a simpler process. The child identification task is in the order of $O(N)$ where N is the number of neighbors surrounding the non-survivors. The second alternative is of the order $O(NM)$ where N is the number of neighbors surrounding the survivors and M is the number of neighbors of the neighbors of the survivors. The increase in the computation steps is due to the requirement for an additional inner loop to examine the neighbors of the neighbors of the survivor again to identify the best match for the non-survivor. If this requirement can be removed, then the processing order will be the same for both alternatives. In some applications, this requirement can be relaxed to give way to other more important child selection considerations as in one of our papers [70].

The advantage of using the second alternative is in the use of each survivor as the pivot point. In the first alternative, the survivors have no control in the linking to which neighboring child or the number of children. It is the children or the non-survivors themselves to decide the linkage. The survivor or the parent will only have a complete children list status at the end of the process. If there is a specific requirement in the number of children or the layout of the children surrounding the survivor, then the first alternative cannot be fulfilled. Section 6.4.1 presents such a situation where detailed analysis of the neighboring regions is required. The first alternative can never be used in this situation. For a simple child selection criterion the first alternative is sufficient. For a more complex child selection criterion which requires detailed analysis of the survivor surrounding the second alternative must be used. It allows the selection process to evaluate all neighboring non-survivors at the same time before making the linking decision. For the ease of processing, the child selection process is performed on the same pyramid level as the selection of the survivors. The result is temporarily retained until the next higher pyramid level is formed.

3.2.5 *Stopping Criteria*

Once the child selection process is finished, a complete pyramid level is formed. The selected survivors will become the data points on the new pyramid level. The child list of each data point is updated. The respective unique and collective attributes, as described in the section 2.3, of each new pyramid data points are also updated. A new pyramid level creation will begin where the selection of the neighbors, the selection of the survivors and the determination of the child list processes will repeat. In the traditional pyramid model, this process will continue until it reaches the pyramid apex with only one data point. In most applications especially those that utilize the growing ability of the pyramid model, it may not be necessary or possible to reach a pyramid apex. Frequently the formation process will stop at some intermediate level, resulting in a pyramid structure with a flat top. This will be the case in image segmentation where the last pyramid level will hold multiple pyramid data points each representing one homogenous region. In order to determine this pyramid level, a stopping criterion is required to stop the formation process. There are two ways to define the stopping condition. The first is to define the condition locally where each pyramid data point will determine itself whether to stop the formation process by evaluating itself or/and the surrounding neighbors. A data point which decides to stop is called a root node. The formation of a root node may take place on any pyramid level. The pyramid formation process will stop when all the data points become the root nodes. There are many other considerations in the handling of root node which will be discussed in the next section. The second way is to define a global stopping condition. A function is derived to evaluate all pyramid data points either on the current pyramid level or on consecutive pyramid levels to determine the suitability to stop the pyramid formation process.

3.2.6 *Handling of Root Nodes*

The accurate identification of a root node is an essential task as failure to detect the root node may result in under segmentation of objects or the merging of unique objects. In spite of its importance, it is not an easy task. In the course of root node determination, there are three challenges. The first is the definition of a root node formation condition that should be general enough to cater for all types of image objects and yet discriminating enough to avoid noisy objects. The definition will largely depend on the specific application requirement. The second is an efficient way to determine on which pyramid level the root node will appear. The ideal scenario will be a complete representation of all root nodes on a single pyramid level. In contrast, this scenario seldom happens while processing real images where the variation in objects sizes, contrasts and positions will affect the object formation rate. As a result, root nodes will usually appear on multiple pyramid levels. Due to this problem the third challenge is the treatment of the root nodes during the pyramid formation process and how they interact with other non root nodes. Montanvert et al [55] point out three possible treatments. The first is to have no treatment (i.e. the root node is treated the same as the other nodes). This will result in the disappearance of root node and merger with other nodes at higher pyramid levels. The second is to allow the identified root node to always survive and continue to interact with other nodes. This will enable the surviving root node to claim other neighboring non-survivors and as a result forming a new root node again. The result of this treatment is the formation of multiple root nodes representing the same image object. The final treatment is to allow the detected root node to survive and prevent it from interacting with the other nodes. According to the authors, this may result in over segmentation of the image with too many roots. According to our view this may not be always the case. In our model we utilize the last alternative. Depending on the root node definition, a suitable definition may result in just the right degree of segmentation. An alternative to the root node detection during the pyramid formation process is to first allow a

complete pyramid structure formation and then perform a top-down analysis of the pyramid structure again to identify root nodes. It is a possible solution used in [63].

3.3 Irregular Pyramid in Textual Segmentation

In the ensuing chapters we will describe the problems in textual segmentation and extraction faced by using the traditional methods and our proposed solutions according to the three types of input document images (i.e. binary, gray scale and color). It will discuss in detail the solution, using the irregular pyramid structure, to each problem. The solutions will be based on the basic irregular pyramid model as described in chapter 3. As all the solutions are focused on the segmentation of textual region within the pyramid structure, the discussion will use a pyramid region or a pyramid data point to represent an element in the pyramid structure. Each pyramid region corresponds to one physical pyramid data point on a specific pyramid level. The traversal within the pyramid structure starting from the pyramid data point down to the base pyramid level will produce a pyramid region in the original image. Some papers [64, 69] refer to the pyramid region as the receptive field of the pyramid data point. In each of the following four chapters, we will first survey related works on the problem in question, and show how the current methods fail to address certain issues. We will then present our irregular pyramid solution to meet the challenge. Various issues will be discussed and experimental results will be presented to validate our method.

Chapter 4

Word Segmentation in Binary Imaged Documents

This chapter describes a method that we have published in two conferences [65, 66] to detect word groups in binary imaged documents, using irregular pyramid. The irregular pyramid model is engaged to solve three common problems still faced by the document image processing community in the segmentation and extraction of the textual component from a document image. These problems are the segmentation of text with broken characters, text in varying sizes and text in arbitrary orientations. Our method does not utilize connected component analysis where it will fail with joined or broken characters and can only detect text at the character level. Our method also avoids the need to define any fixed text size or distance threshold for the segmentation of text and the aggregation of characters into words. The novelty of our method is its inclusion of strategic background information in the analysis where most techniques have discarded, except in the White Tiles approach to perform text skew estimation using the background region as proposed by A. Antonacopoulos in [140]. Our method differs in the use of both foreground (i.e. text area) and portion of background (i.e. white area) regions in the analysis. The fundamental principle of the method is based on 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 definition of “closeness” is achieved with no specified measurement of the inter-character or inter-word distance. The method also defines and proposes the use of the word’s density in the measurement of word’s formation status. The result produced by the method is shown in the experimental section illustrating the ability of our method to

correctly group words of different sizes, fonts, arrangements and orientations with or without broken characters.

4.1 Related Works

One of the key pre-processing steps in most of the textual segmentation methods is the document skew correction process. Almost all proposed methods either assume a strict Manhattan document layout or require some form of skew correction before any further processing in segmentation or layout analysis. As reported by Cattoni, Coianiz, Messelodi and Modena in their paper [142], skew estimation remains as the most important pre-processing steps in document image processing. Despite the many proposed skew estimation methods, problems remain in terms of the accuracy and the strong assumptions regarding the input domains. The majority of the methods assume a clearly dominant skew angle for the entire document. Although there are some methods that allow multiple skews within the document, they are still subject to the requirement of a common skew angle within the same text group. A typical example is in O’Gorman proposed page layout analysis algorithm [121] that will segment document images into rectangular blocks of text. Although the algorithm has the ability to segment text of varying orientation, it still requires the text within the same block to have a common flow of direction. The presence of graphics poses a great challenge among many of these methods. Not many proposed methods can handle document with text in greatly varying orientations among words within the same sentence or even among characters of the same word. Existing skew correction techniques will have problems processing document images from real scenes, web pages, advertisements and maps which have become important sources of target images.

In the main segmentation stage our aim is in the segmentation of the textual components. There are many proposed methods in the segmentation of document images. Most of them

focus their attentions on the isolation of characters. Some methods extend the segmentation to merge characters into words or work directly in the segmentation of words. There are also methods that process up to the sentence or the paragraph level which cross beyond simply textual segmentation into the document layout analysis area. Among all the proposed methods a few common basic approaches are observed. They are the run-length smearing algorithm (i.e. RLSA) as proposed in [109], the XY-cut method as proposed in [111], the Hough transform as utilized in [113] and the frequently used connected component analysis. Almost all proposed document image segmentation techniques utilize a combination of common approaches or devise methods based on the modification of these approaches. In the run-length smearing algorithm continuous stream of binary pixels are examined horizontally column by column and vertically row by row according to the image rectangular layout. Sections of black pixels that are separated less than a certain number of white pixels are smeared to become a continuous stretch of black pixels. Iterative merging of pixels in both directions will enable the grouping of pixels into character and characters into word. The XY-cut method works in an opposite way. Instead of starting from the pixels level to grow into text objects, the method starts from the image level and performs recursive X-Y cutting of the image region into smaller and smaller regions until it reaches the text object. As in the RLSA the cutting is done horizontally and vertically with respect to the rectangular image layout. In order to determine the points of cutting, projection profile histograms in both directions are used. An example is in [90] where both horizontal and vertical projection profile histograms are constructed to determine individual text lines and words. Irregular text sizes or non-uniform horizontal alignments of the text contents will result in some complications in the determination of the cutting points. As we can see, both of these traditional techniques require the assumption of a strict Manhattan document layout in order for the methods to work. Any skewing in the angle of the document image may result in an undesirable output. As a result skew correction as discussed above is

always a required pre-processing step before the actual segmentation. Many proposed methods utilize a hybrid of these two common approaches. A combination of the splitting and merging operations, basing on the same concept as in the cutting and the smearing techniques, is used in [131] where the image is first split into many overlapping columns and it is then followed by a line extraction/merging process where a histogram is used to determine an appropriate height of the text line in order to correctly extract the different horizontally aligned text lines. The paper in [127, 162, 211] also makes an assumption of a horizontally aligned textual component in order to facilitate the detailed analysis of the distance between regions. The labeling algorithm [111] requires the alteration in the image horizontal scale to facilitate the extraction.

The Hough transform approach attempts to extrapolate the direction of a group of text objects by accumulating the pixels surrounding a region into many bins, each defined by a straight line rotating at a certain angular step iteratively. The direction of the straight line that has accumulated the most pixels defines the text orientation. This method is mainly applied to the detection of text group orientation and not the actual segmentation of the text regions. The segmentation of the textual region will still require either of the above methods. Despite its ability to detect text direction, the effective usage of this method will depend in the determination of the centre line which will be complicated by the presence of ascenders, descenders and text of varying sizes. It also depends on the definition of the angular steps. More steps will yield a more accurate result, but also translates into a higher computational cost. Some of the proposed methods [115, 136], make use of this technique in the detection of text in varying directions.

The final basic approach is the connected component analysis. It is the most frequently used technique for binary input document images. The foreground object which is

represented by the black pixels is formed through the analysis of either the 4-connecting neighbors or the 8-connecting neighbors. All joined neighbors that satisfy either of the criteria (i.e. 4cc or 8cc) is merged as a single object. This technique is usually used in the isolation of the character component due to the continuous joined text stroke of the English text, except in some characters like the “i” and the “j”. They are formed by two disjoint components. Some methods [94] have proposed solutions to this problem, but most of the segmentation methods ignore this problem. The greatest problem with this technique is the assumption of the absence of disjoint and non-broken characters. Problems occur when multiple characters are joined which may be due to the existence of noise in between the characters where the joined multiple characters are treated as a single character. A single character may split into multiple fragments if for some reason (e.g. scanning) the text strokes are broken and thus the fragmented portions may each be treated as a standalone character. This effect is not desirable where confusion will start to arise during the character recognition phase. The technique will only handle character level segmentation. In order for the identified character to form into words, again either the RLSA or the XY-cut techniques have to be employed. The method proposed in [102] tries to down scale the connected component image by $\frac{1}{4}$ the original size to identify word groups in the horizontal direction. This is equivalent to the smearing of characters into word in the horizontal direction. Another technique proposed in [128] attempts to first construct bounding boxes enclosing each connected components and follow by a top-down recursive X-Y cut decomposition of the document images.

In addition to the above mentioned problems of the restriction in the text orientation and the disjoint or broken character, methods that are based on these four basic approaches also need to fix or to compute some type of inter-textual components distance threshold. The inter-character spacing will also need to be determined prior to the formation of words. The

inter-word spacing will be required in the formation of sentences. As a result detailed spatial analysis of the image content is required even before the content is segmented. Some other problems are the required assumption of uniform text fonts and sizes. A mixture of regular and italic text fonts may create problems in the XY-cutting. Irregular text sizes within the same group of text components will create problems in the correct determination of text orientation using the Hough transform and the smearing of characters into words using RLSA. A detail estimation of the text height, width and the inter-character spacing is required in [96] in order to effectively separate the text from the drawing object. Reference [90] also requires the use of distance threshold to ensure the correct extraction of horizontal text bounding boxes. The estimation of inter-char spacing and text height is also needed in [211] to enable proper merging of characters into sentence of color document images. Although a method proposed in [139] has achieved the extraction of textual component in various orientations using multi-stage relaxation approach, it also has an implicit requirement in the assumption of similar text width and height within the same text group and has very high computational cost.

Most of the pyramid-based methods make use of a regular pyramid structure. Most of these studies require connected component analysis [48]. A strong assumption of disjoint components is needed to ensure correct extraction of text images [31]. In this thesis work, no connected component analysis is required to extract textual components on the character's level. Our method will extract all components at the word's level regardless of whether there are joined or broken characters and thus simplify the recognition task by focusing only on word's recognition. The aggregation of pixels into characters and character into words is done through the natural grouping of pixels. This proposed algorithm has no assumption in terms of the sizes and orientations of the textual components in the input images or even within the same text group.

4.2 Fundamental Concepts

The proposed method relies on three fundamental concepts. They are the involvement of non-text image areas in the analysis, the definition of “closeness” and the definition of word’s density.

4.2.1 *Inclusion of Background Information*

One of the novel features of this proposed algorithm is the inclusion of background (i.e. non-text) image area in the analysis. Background information is usually considered not important and discarded in most of the text extraction techniques. Attention is thus normally directed to text areas (i.e. black pixels) and there is no holistic view of the entire picture involving the non-text areas. For document images containing regular lines of text, this is fine [109], but when there are irregular alignments and orientations of text such as in advertising documents, detection of words in correct grouping becomes difficult and requires extensive analysis of spatial relationship among characters and words [101]. In the present work, however, we capture crucial information about the spatial distances between text image objects among the non-text area in an irregular pyramid structure. The structure provides a direct and natural grouping of words in any arrangement and orientation without the need for spatial reasoning among the words.

With the involvement of the background information, we can now view a text image as a combination of multiple irregular smaller regions as shown in Figure 26 with different colors. Some regions may contain text information while others may be empty. A potential word group can thus be viewed as a bigger region containing smaller regions holding fragments of a word group including the empty regions. The concatenation of these region fragments forms a word group. With multiple fragments of region that belong to different

word groups, the problem in clustering these fragments to the right word group occurs. In our algorithm, the problem is solved by introducing the concept of “closeness”.



Figure 26. A word group formed by multiple words' fragments

4.2.2 *Concept of “closeness”*

A human reader can easily identify different logical text groups from a text image if the text information within the group is close to each other, in terms of the spatial distance (i.e. in all directions), as compared to other text areas. The distance between characters is smaller within a word group, as compared to distance between two separate word groups. If we view a text image as a combination of multiple regions, then a logical text group is defined as a region enclosing various sub-regions that are “close” to each other. Some of these sub-regions may contain text area. Others may be empty. While processing text with a common font, size and orientation, we can find a value in defining ‘closeness’. The papers in [159, 166] have explored the possibility to compute such a value in the extraction activity with some degree of success. Nevertheless, the method is quite restrictive in terms of the kind of documents 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”.

Two regions are considered “close” if they appear in the immediate surrounding of each other. No computation of the physical distance is required. To be close, regions just need to be around in the neighborhood. In cases where there are more than two regions, the “closeness” property among regions is defined with respect to a reference point or the pivot region. A group of regions are considered “close” if there are “close” to a pivot region. We can also view such a region as a central pulling force in pulling all other surrounding regions together, which are considered “close” to it. 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”. By using an irregular pyramid structure, such concept of “closeness” can be implemented. The pyramid structure with successively condensing image resolution allows the “closeness” among local regions to grow progressively from level to level into the “closeness” among the global region. Figure 27 and Figure 28 show two examples of “closeness”.

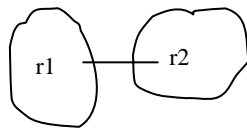


Figure 27. “closeness” between two regions

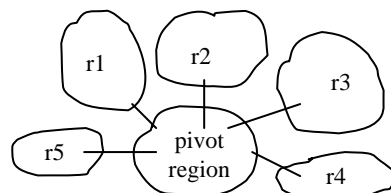


Figure 28. “closeness” among multiple regions

4.2.3 Density of a Word Region

A word region is defined as a tightly bounded area enclosing the content of the word formed by multiple characters that are placed relatively “close” to each other as compared to distance to the character in other words. Figure 29 shows an example of two word regions. The area of a word region is the total number of pixels within the region. This will include both foreground and background pixels. The total area for the word “dispensing” is the

number of yellow pixels (i.e. background) plus the number of black pixels (i.e. foreground). The mass of a word region is the total number of foreground black pixels of all characters forming the word. The density of a region is thus defined as the total mass enclosed by the region over the actual area of the region. The density value of a region that constitutes a word group reflects the size of the area enclosing the specific word's mass. Our observation reveals that all regions containing word groups have a similar density value regardless of the regions' mass. A word region having a bigger mass requires a larger area to enclose the word. Smaller mass regions will have smaller enclosing areas. As a result the density value of the word region or word group is relatively stable. In [168], the authors also make use of the definition of the density value in the identification of text components. It differs from our density definition in the use of a fixed circular local window to define the bounding area. Our bounding area is an irregular region enclosing the textual components which has better adaptability to varying text component sizes and layouts.



Figure 29. Example of two word groups

4.3 Pyramid Model

In order to achieve the objective in the detection of word groups in a binary document image, the basic irregular pyramid model as described in section 3.2 is reconfigured to meet the requirement. The first configuration is in the pyramid data point attribute list. On top of the usual pixel index, the neighborhood list, the children list and the surviving value attribute, two other attributes are added. They are the area attribute $d_{i,j}^a$, the mass attribute

$d_{i,j}^m$ and the root node status attribute $d_{i,j}^r$. The area and the mass attributes are used to facilitate the computation of the pyramid region density or the ultimate word group density as discussed above. The root node status attribute is used to indicate whether the current region constitutes a word group. The root node status attribute has a binary state. If the current node is determined to be a word group, it will have a value of 1. Otherwise the value is 0. Since there is not much intensity information to process, the intensity attribute is discarded.

$$\overrightarrow{D}_{i,j} = \left\{ d_{i,j}^p, \overrightarrow{d_{i,j}^b}, \overrightarrow{d_{i,j}^c}, d_{i,j}^s, d_{i,j}^a, d_{i,j}^m, d_{i,j}^r \right\} \quad (20)$$

The area attribute $d_{i,j}^a$ is derived by simply accumulating the total area of all the children β_r . It reflects the total number of pixels that a pyramid data point or region represents in the original image regardless of whether it is the foreground or the background area. Each pyramid data point on the base level (i.e. level 0) has an area of 1.

$$d_{i,j}^a = \begin{cases} 1, & \text{if } i=0 \\ \sum_{r=1}^{N_c} (\beta_r^a \mid \beta_r \in \overrightarrow{d_{i,j}^c}), & \text{otherwise} \end{cases} \quad (21)$$

Unlike the area attribute the mass attribute $d_{i,j}^m$ only reflects the size of the foreground region. On the base pyramid level, the mass attribute of those pyramid data points having the foreground pixel will have a mass of 1. Those representing the background pixels will have no mass. For the higher pyramid level (i.e. level > 0), the mass attribute is derived by summing all the mass attributes of the children. This attribute will reflect the size of the foreground area within a pyramid region as shown in equation 22.

$$d_{i,j}^m = \begin{cases} 1, & \text{if } i = 0 \wedge \text{img}(p) = \text{black} \\ 0, & \text{if } i = 0 \wedge \text{img}(p) = \text{white} \\ \sum_{r=1}^{N_c} (\beta_r^m \mid \beta_r \in \overrightarrow{d_{i,j}^c}), & \text{otherwise} \end{cases} \quad (22)$$

The second configuration is the derivation of the surviving value. The surviving value $d_{i,j}^s$ of a pyramid region $D_{i,j}$ is computed by summing its own mass $d_{i,j}^m$ and the mass of all its neighboring regions α_q as indicated in equation 23. The motivation behind such computation is to identify the most suitable pivot region to assist in the application of the concept of “closeness”. Larger surviving value reflects that the region either has a bigger mass (i.e. the core of the textual area) or it has more neighboring regions with mass (i.e. fragments of a word group). Both constitute a good survivor candidate. Only region with mass will have a surviving value. Since the main aim is to perform text extraction, the survivor selection process will focus on regions with mass. Nevertheless, empty regions are also processed under a different role. The empty region will participate in the definition of the region’s density and act as a bridging region between parts of a word group by presenting itself as the child of the survivor. The proposed method needs the complete involvement from all the pixel points.

$$d_{i,j}^s = d_{i,j}^m + \sum_{q=1}^{N_b} (\alpha_q^m \mid \alpha_q \in \overrightarrow{d_{i,j}^b}) \text{ where } d_{i,j}^m > 0 \quad (23)$$

4.4 Pyramid Formation

The pyramid formation process is almost the same as the basic model with some modifications to suit the current objective. The process will go through the pyramid level

creation, neighborhood determination, survivor selection, child selection and finally the stopping criteria evaluation. The main differences are in the survivor selection, child selection and the definition of the stopping criteria.

4.4.1 Selection of Survivors

There are two types of survivors. The first type is the local maximum. A region survives if there is no other neighboring region with a larger surviving value. It reflects a higher possibility to be the center of a word group. Its neighboring regions with a lower surviving value are more likely to be the fragments of the word group. Thus the survivor acts as a good pivot region. The second type of survivor belongs to those regions with no surviving values and not a neighbor to any survivor. They reflect the non-existence of any neighboring foreground areas. These regions are the background area. In order to allow their continue participation in the growing of text region to act as a bridging regions, they are allowed to survive. In spite of this, they are prohibited from participating in the child selection process. This is to prevent such regions from growing at the same rate as the other promising regions containing foreground information and later create an un-necessary interference. In most document images, they are a large group of such regions. In order not to waste too much physical memory in propagating this group of pyramid data points from one pyramid level to the next pyramid level, they are retained in a separate structure and shared among all pyramid levels. The majority of such regions will remain in this structure, except those in the neighborhood of the survivor (i.e. the local maximum) and its surrounding non-survivors. This group of regions (i.e. survivor with no surviving value) will also be promoted onto the next pyramid level. Figure 29 shows the existence of such regions, surrounding the words “dispensing” and “tissues”, in gray color. The purpose is to allow the continued growing of the text region into its surrounding within the pyramid structure without the need to reference to an external structure. The process to transfer this group of

regions into the pyramid structure will happen at every pyramid creation step for all higher pyramid levels. This modification to the basic pyramid formation process has enabled the continuation in the usual formation process and yet preserved the precise physical memory.

4.4.2 *Selection of Children*

The purpose of this process is to establish the parent-child link. In our context to perform segmentation of the textual region, it can also be viewed as the growing of the selected survivors or seeds into the neighboring regions. In order to better adapt to the conditions in the various growing stages of a word region, different sets of the selection criteria are used in different pyramid levels. Since on pyramid levels 0 and 1, the possibility of locating word groups is very low and as a result no special evaluation is required. A survivor will claim all neighboring non-survivors if there is no other survivor claiming the same non-survivor. In situation where two survivors are claiming the same non-survivor, the non-survivor is assigned to the survivor with a larger surviving value. On the lower pyramid level, the effect of such claiming rule is not obvious. Once the region grows to a bigger size on a higher pyramid level, the rule of allowing a region with larger surviving value to claim more non-survivors than a region with smaller surviving value will become obvious. A region with a larger surviving value is more likely to be the centre of a word group. As such, in allowing it to pull in more neighbors, it will promote the possibility of forming a word group.

Once the process reaches level 2 and above, the likelihood to locate word group increases and thus a more elaborate child selection process is used. The criteria will gear more towards the determination of the word group formation. In general, a potential word region has no neighboring regions with mass and has a density below a density threshold T_d . The derivation of T_d is discussed in section 4.4.3. A word region must have enough enclosing

area. As a result, a region with a high mass-to-area ratio will not be considered as a word region. Figure 30 shows the detailed child selection algorithm.

```

1. For each survivor
2. { if (exist neighboring regions with mass)
3.   { for (all neighbors with mass)
4.     { assign neighbor as a child to the survivor }
5.     for (all neighbors with no mass and new_density <  $T_d$ )
6.       { assign neighbor as a child to the survivor }
7.   }
8.   else
9.     { for (all survivor neighbors and new_density <  $T_d$ )
10.      { assign neighbor as a child to the survivor }
11.    }
12. }
```

Figure 30. Detailed child selection algorithm

On this level we will see two categories of survivor. The first category pertains to those survivors which have neighbors with mass (i.e. line 2-7). The other category contains survivors with no neighboring mass (i.e. line 9-11). These two categories of survivors are mutually exclusive. At any instance only one can occur. Intuitively, the first category belongs to those survivors which are still in pieces and form parts of a word group in the neighborhood. The second category belongs to those that may have already formed a word group and there exists no other mass in its surrounding. For the first category the algorithm is again divided into two parts. Unlike previously, these two parts are processed one after the other. The algorithm will focus on those neighbors with mass (i.e. line 3-4). The neighbor is pulled in as a child immediately. In doing so, the same validations as those perform on level 1 and below will also carry out. Once all neighbors with mass are examined, the algorithm will continue to process the remaining neighbor with zero mass (i.e. line 5-6). Unlike the previous two levels where even the blank regions are taken in with no question asked, from this level onwards special care must be taken in order to avoid overgrowing of regions into other word groups. This may result in the overlapping of more than

one word group. Two different word groups may thus wrongly merge. In order to control the growth, density is used. For each addition of a neighboring region, the density of the new region is computed. Blank region will continue to be added until the density threshold T_d is reached.

In the second category (i.e. line 9-11), survivors are surrounded by a blank region. Since there is no other neighboring mass, such surviving regions may have already formed a word group. The majority of the regions falling under this scenario is a correct word group and thus no further pulling of neighbors is required. Nevertheless, there is a small percentage of regions that are only part of the word group. Density is again used to force these remaining regions to find the correct word group. As in the second part of the first category, the survivor will continue to pull in the blank region until the new region reaches the density threshold T_d .

4.4.3 *Stopping Criteria*

Since in this context the main purpose is to extract the word groups, the formation process will stop when there is no possibility to detect any more word groups. As discussed in the child selection process, the detection of a word group is by evaluating the density of a region (i.e. $< T_d$) and the region's neighborhood (i.e. no neighboring regions with mass). Although the density value of each word region is relatively stable, it does vary slightly in the word groups with varying characters spacing. Word groups with a larger inter-character spacing will yield lower density than one having characters placed closer together. With the same character size, the former will have a bigger enclosing area than the later and thus resulting in a lower density value. In order to reduce the effect of this variation, the average density of all word groups is used as the density threshold T_d . The threshold is further refined dynamically by re-computing a new threshold $T_d(i)$ for each pyramid level i .

$$T_d(i) = \frac{\sum \left(\frac{d_{i,j}^m}{d_{i,j}^a} \right) \forall \overrightarrow{D_{i,j}} \text{ where } d_{i,j}^r = 1}{\left| \forall \overrightarrow{D_{i,j}} \text{ where } d_{i,j}^r = 1 \right|} \quad (24)$$

The threshold $T_d(i)$ is computed at the pyramid creation stage of each pyramid level. Starting from level 2 where an estimate of the density threshold is computed by taking the density average of all regions with no neighboring mass. In our experiment, most of the word groups can be identified after the construction of pyramid level 2. From this point onwards the above derivation for the density threshold is used. As more word groups are formed on each pyramid level, the density threshold will converge and reflect a more accurate value. This density value can thus become a better target density in the formation of word groups. If the majority of the regions require a certain density value to form word groups, then the remaining regions will most likely follow the same ratio.

In order to reach the configuration of having all word groups on the same final pyramid level, root nodes will always survive with no interaction with the other regions. This will dispense with the need to have a post-processing to traverse the pyramid structure again to locate the root nodes on each pyramid level. This configuration also allows the evaluation of the number of formed word groups on the consecutive pyramid levels to determinate the final pyramid level.

4.5 Experimental Results

The first test case is shown in Figure 31 where word groups of different sizes, fonts, alignments and orientations are used. The size of the image is 541 x 298 pixels. There are a total of 94 words. Figure 32 is the final output image (i.e. the receptive field image) with

different coloured regions covering different word groups. Figure 33 is a difference output representation of using rectangular boxes to enclose the extracted word group. Table 3 shows the detail of the output results on different resolution levels. The number of identified word groups on each level is shown. From level 2 onwards the possibility in forming word group is higher and is thus reflected in the slowing down of the reduction in the number of extracted word groups. Once there is no likelihood in forming new word groups the process stops (i.e. level 8). By examining the density column, we can see that the density value re-adjusts itself from level to level, as more and more word groups are formed on each level. The number of remaining pixels includes the surviving pixels and those un-processed background areas. Thus on level 8 there are 94 surviving pixels and 13,513 remaining white pixels.

Table 3. Output result at the various resolution levels

Level	Number of extracted words	Density	Number of remaining pixels
0	-	-	39226
1	-	-	24762
2	1243	0.1949	14771
3	426	0.2469	13938
4	215	0.2510	13702
5	127	0.2566	13563
6	98	0.2578	13517
7	94	0.2572	13513
8	94	0.2572	13513

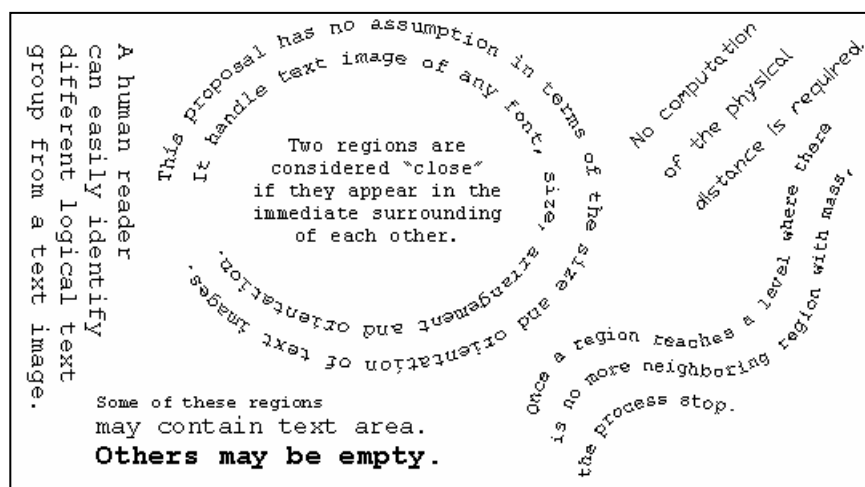


Figure 31. Original text image (541 x 298 pixels)

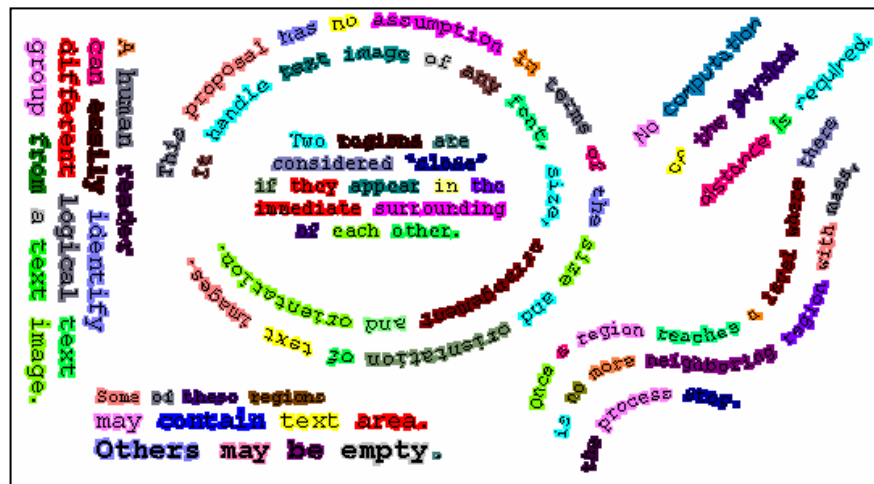


Figure 32. Output result showing various extracted word groups

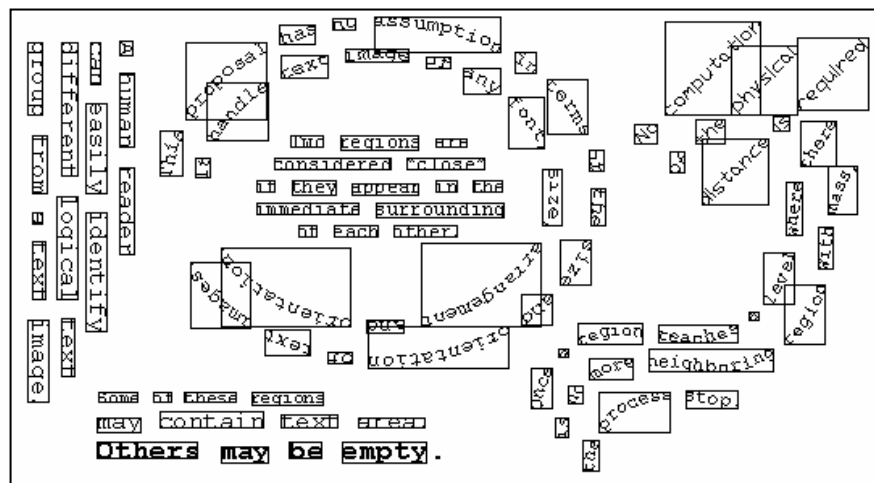


Figure 33. Different representation of the output result

Reported below are some other test cases of scanned real images from newspaper cutting.

In order for the system to focus only on the detection of logical word groups, large graphical objects are removed from the image through a pre-processing step (i.e. filtering by using the component size). The image is then subject to a thresholding algorithm to convert into a binary image before the actual detection process begins. Figure 34 is obtained from a product's packaging material. The image contains word groups of varying sizes and alignments. The image size is 194 x 183 pixels and there are a total of 19 word groups. Figure 35 is the final output with different coloured regions covering different word groups.

In this test image all word groups are correctly identified. One can observe that the middle word groups are all joined. By using connected component analysis they will be extracted as 5 separate character components. Together with the other extracted valid character components these components will be rejected during the character recognition phase. Instead of worrying about such connected characters situation in an environment where the expectation is to extract only character components, our method isolates all regions as the word components. This will simplify and streamline the recognition phase to only focus on the recognition of word.

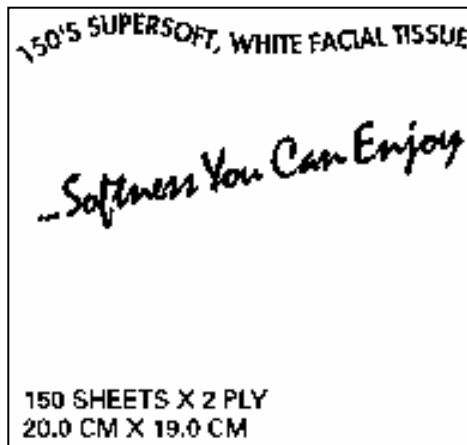


Figure 34. Tissue box test sample



Figure 35. Segmentation result of Figure 34

Figure 36 shows another test image of a newspaper advertisement for toys. The image size is 359 x 244 pixels and there are a total of 45 word groups. Figure 37 is the final output using rectangular boxes to enclose the extracted word groups. In this test image 2 word groups (i.e. “toys” and “Accessories”) are wrongly identified and the 3 reversed symbol and texts (i.e. “\$”, “1” and “onwards”) are treated as background objects. The last test image shown in Figure 38 is from a pamphlet showing the route map of a local train system. The size of the image is 869 x 741 pixels. All the dashed lines shown in the image are removed by the size filter as graphical objects (i.e. most of the dashed lines are joined as a single

object). There are a total of 73 words in varying alignments and orientations. Figure 39 is the final output. In this test image, the number of correct word groups is 73. The majority of these word groups are correctly extracted except for a few where two separate words are detected as one. Of the 69 word groups detected, there are 65 correctly identified words and 4 that are wrongly identified. All four groups contained double words. This is due to the fact that parts of the word are too close to each other and it has exceeded the tolerance level of our algorithm.



Figure 36. Newspaper cutting: toys

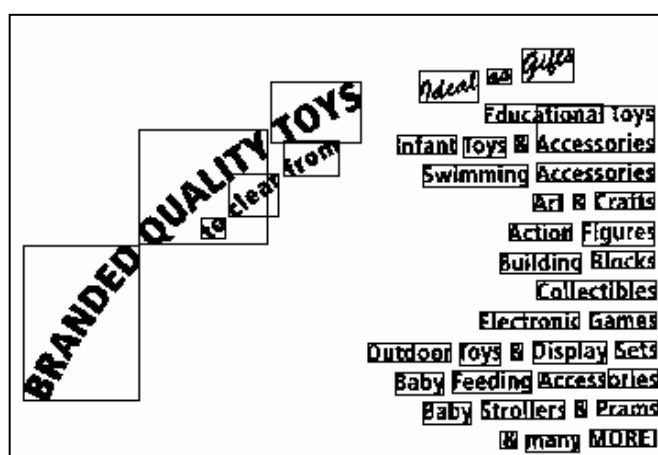


Figure 37. Segmentation result of Figure 36

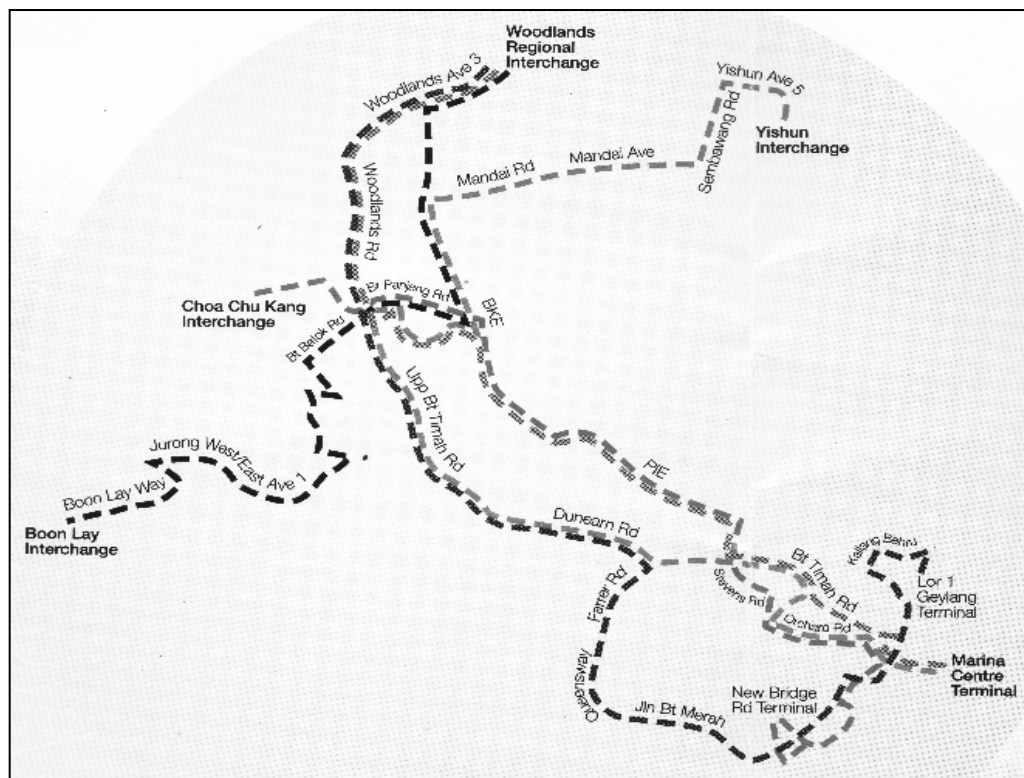


Figure 38. The route map test sample

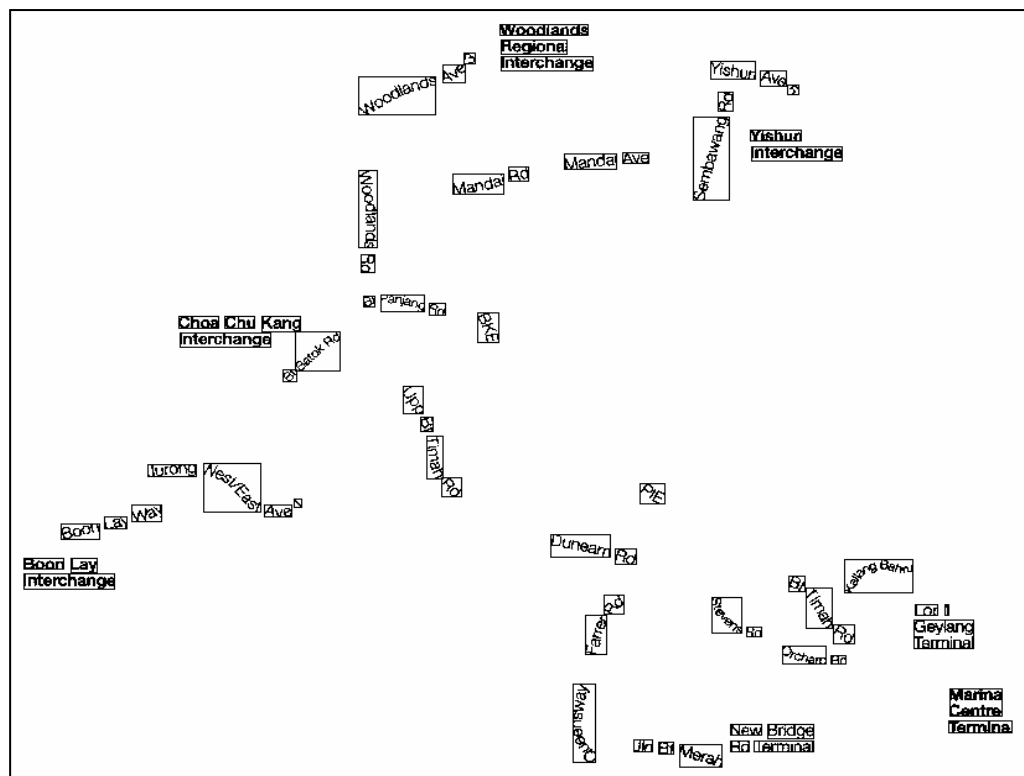


Figure 39. Segmentation result of Figure 38

4.6 Summary and Discussion

We propose a new approach to word group's detection using an irregular pyramid structure. The method differs from other approaches in that it captures background information (i.e. non-text area) and forms irregular regions in a hierarchical manner through the concept of "closeness". The irregular pyramid structure in effect contains the layout information of the entire document such that layout analysis is straightforward by navigating the pyramid levels. The proposed method is able to extract word groups with varying sizes, fonts, alignments and orientations, as found in such documents as maps, advertising flyers and book covers. In fact, the method works well with conventional text documents too. It is not constrained by any specific layout or text density as the system is self adjusting in gauging the text density. The contribution of this method is in the application of the irregular pyramid model in word level segmentation where no others have attempted. We have also proposed a solution in solving the broken or joined text problem in text extraction using the connected component analysis method. The alternative solution in extracting text with varying sizes, fonts and orientations without any need to perform detailed spatial analysis or distance threshold fixing are also one of our contributions. Finally the definition of word density enables the detection of word formation which is not an easy task in many other methods.

In the next chapter we will present the continuation of this work to extract multiple word groups to form phrases and paragraphs. An added feature that will help this task is the computation of "word growing direction". As we examine the formation of word regions from multiple smaller regions, the trend of the growing direction through the pyramid levels can be determined. This will allow finding of road names and river names along the winding lines on the maps.

Chapter 5

Identification of Textual Layout

This chapter will present the result of our continued work on a further enhancement to the previous proposed algorithm as described in Chapter 4. The method is published in [67]. 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 and orientation as discussed before with the new addition of even varying layouts on the same document image. No assumption is made on any specified document type. The algorithm is based on the same irregular pyramid model with some addition attributes and processes. The method also relies on the same concept of the inclusion of background information and 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. Modification to the concept of word's density is adopted here. The method also introduces two new concepts. The concept in the “majority win” strategy that is more suitable under the greatly varying environment than a constant threshold value and the concept of the directional uniformity and continuity.

5.1 Fundamental Concepts

The five fundamental concepts that this method relies upon are the inclusion of background information, the concept of “closeness”, the density of a word region, the majority “win” strategy and the directional uniformity and continuity among words in a

sentence. The first two concepts are as described in the preceding chapter. The other three will be explained in detail below.

5.1.1 *Density of a Word Region*

The reader may recall that, 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 word fragments. 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 previously 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, which is a new concept to be adopted in this chapter, is used as a criterion to determine a correct word formation among a group of neighboring words. The detail of this concept is explained below.

5.1.2 Majority “win” Strategy

Since our algorithm has no restriction to the kind of documents it can process, the variation over the text image feature will vary greatly. The possibility of making a 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.

5.1.3 Directional Uniformity and Continuity

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. An example is shown in Figure 40 where logical group of words exhibit directional uniformity and the red line defines the directional path. This is a common scenario observed in most text documents.



Figure 40. Directional uniformity

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 the words' alignment is, continuity among words will still exist to allow a human reader to perceive this group of words. Figure 41 and Figure 42 show 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.



Figure 41. Directional continuity in a sentence

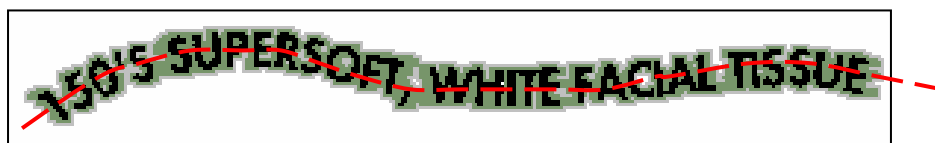


Figure 42. Directional continuity in a sentence

5.2 Pyramid Model

There are a few changes in the configuration of the pyramid model. The changes are the addition of more pyramid data point attributes. They are the “pulling status” attribute $d_{i,j}^{ps}$ and a vector attribute $\overrightarrow{d_{i,j}^{dm}}$ holding eight directional mass entries.

$$\overrightarrow{D_{i,j}} = \left\{ d_{i,j}^p, \overrightarrow{d_{i,j}^b}, \overrightarrow{d_{i,j}^c}, d_{i,j}^s, d_{i,j}^a, d_{i,j}^m, d_{i,j}^r, d_{i,j}^{ps}, \overrightarrow{d_{i,j}^{dm}} \right\} \quad (25)$$

The pulling status attribute $d_{i,j}^{ps}$ is used by the child selection process to manage the selection of the surrounding non-survivors as the children of a survivor. It is also viewed as a growing process where the survivor or the seed region is grown into the surrounding regions by “pulling” those eligible neighboring regions for merging. Depending on the binary state of the attribute, the process will follow one of the two ways in pulling the neighboring regions. They are the ‘special’ pulling where only surrounding neighbors with mass are considered and the “general” pulling which will claim all neighboring regions (i.e. mass and non-mass).

$$d_{i,j}^{ps} = \begin{cases} 1, & \text{"special" pulling} \\ 0, & \text{"general" pulling} \end{cases} \quad (26)$$

In order to assist the extraction of sentences, an additional attribute called the growing directional mass/weight $\overrightarrow{d_{i,j}^{dm}}$ 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 as shown in Figure 43. Just like the 8-connectivity directions, it comprises of top, top-right, right, bottom-right, bottom, bottom-left, left and top-left.

$$\overrightarrow{d_{i,j}^{dm}} = \{dm_{i,j}^t, dm_{i,j}^{tr}, dm_{i,j}^r, dm_{i,j}^{br}, dm_{i,j}^b, dm_{i,j}^{bl}, dm_{i,j}^l, dm_{i,j}^{tl}\} \quad (27)$$

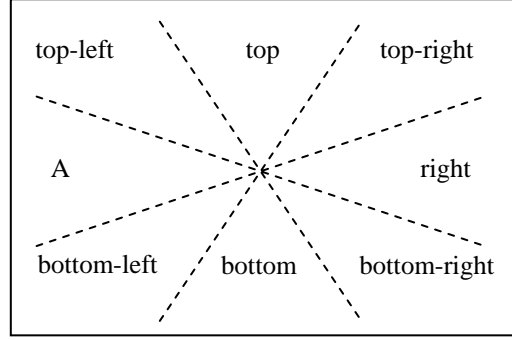


Figure 43. The eight growing directions

As a survivor is promoted to a higher pyramid level, it retains its original directional weight from the lower level. Using this set of weights as the base, it analyzes all the child regions β_r . By taking the center of mass of the overall region (i.e. all regions covered by the survivor) as the pivot point, the direction of where the child is located $dir(\beta_r)$ 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 $dm_{i,j}^x$ in a specified direction x with the mass of all children in the same direction. It will retain the maximal value.

$$dm_{i,j}^x = \max\left(dm_{i,j}^x, \beta_r^m \mid \forall \beta_r \mid \beta_r \in \overrightarrow{d_{i,j}^c} \wedge dir(\beta_r) = x\right) \quad (28)$$

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.

5.3 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. Figure 44 contains the pseudo code of the main algorithm for the word extraction. As compared to the proposed method in the previous chapter, we have revised some of the old procedures and added a few new procedures. The major revision is in the child selection process. The new additions are the assignment and the adjustment of a pulling status as shown in line 8 and 9 of Figure 44.

1. Create pyramid base level with (original image)
2. Select survivors
3. Select children for each survivor
4. For (each pyramid level where number of pyramid data points > 1 and
5. more word groups continue to form in the last pyramid level)
6. { Create pyramid higher level with (previously formed level)
7. Update the survivor neighborhood list
8. Assign pulling status (ie. general/special) to each region
9. Adjust the pulling status (ie. “smoothing”)
10. Assign surviving value to each region
11. Select survivors for the next higher pyramid level
12. Select children for each survivor
13. }

Figure 44. The new algorithm for words and sentences segmentation

5.3.1 Word Extraction Process

The objective of this process 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. The strategy is first to identify the potential center of a word group (i.e. local maximum among the neighboring regions). With this central region as the survivor we assign the neighboring non-surviving regions (i.e. 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 survivor and child selection criteria in our algorithm are set as follows. The survivor selection criterion is to allow a 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 child selection process, the current algorithm has made some modifications in the criteria and the procedure as described below.

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 the “general” claiming of neighboring regions. The motivation 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 fragments into the neighborhood. Starting from level 2 and onwards, a more restrictive child selection process is used and we call this the “special” claiming. The motivation is to slow down the claiming process and claim only those necessary neighbors. A survivor will first claim all neighboring regions with mass. A non-mass region will only be considered if the surviving region density is below a pyramid level density threshold obtained through the averaging of all formed word groups (i.e. root nodes). The growing towards the background area (i.e. no mass) will only continue if the region of the survivor is too dense as compared to the average density of the already formed word groups.

Two assumptions are used in the process. The first assumption is that no word groups will ever form before level 2 and thus the process will encourage unrestrictive growing. The growing process will slow down from level 2 and onwards as the likelihood to locate a word group is higher. Based on this assumption the initial estimate of the density threshold used in the detection of the word groups is derived. The process treats this value as a good estimate and requires the compliance of all regional densities towards this density threshold. Although the threshold is dynamically updated at every pyramid level, the starting point is fixed at pyramid level 2. The second assumption is that the density of all word groups will always fall within the globally derived pyramid level density threshold without exception.

After some experiments with more test images we discover that the problems of “over growing” and “under growing” occur in some word regions. “Over growing” 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. Figure 45 shows an overgrown word group where all individual words are erroneously grouped together as one “word”. It also shows an under grown word where part of the letter “i” and the letter “B” are detached from the word groups.

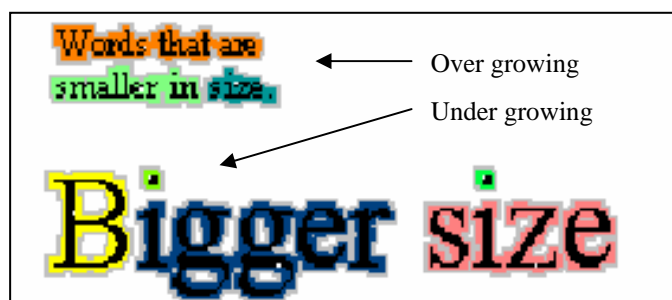


Figure 45. Problem with over/under growing

After some analysis of the results we discover that the key reason to the above problems is in our assumptions. Although the assumptions work for the majority of the word groups, cases that fall outside these assumptions will lead to the above problems. The first case is in those exceptionally small text regions. Some of these regions may already form a word group before pyramid level 2. Since “general” claiming is used before pyramid level 2 where a region will continue to grow un-restrictively into the surrounding, the already formed small word region will continue to grow outwards and the chances of growing into other word groups will occur. This is the result of “over growing”. The problem lies in the timing in switching from “general” to “special” claiming. The second case is in those very large text regions. As compared to the regular regions, the formation of such large regions may still be in a very initial stage even at pyramid level 2. 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. The use of the estimated density threshold at level 2 will also inhibit their growth. In addition the varying local conditions or the surrounding areas of each region may be quite different. In using a globally determined threshold value, the process will not be able to adapt to all situations.

In order to solve this problem, the algorithm is amended to include a “pulling” status flag $d_{i,j}^{ps}$ for each region. The flag will indicate whether a survivor should use “general” pulling or “special” pulling to claim its neighboring children. In “general” pulling, the survivor will claim all neighboring regions which include both the mass and non-mass regions. In “special” pulling only neighbors with mass are considered. The flag will allow individual survivors or seed regions to grow independently from each other depending on their own local conditions. Instead of every one following a rigid global decision, the survivors will make their own local decisions. In order to make the decision, the survivor will analyze its

immediate neighbors. Density is again used to make the decision, but in a different way as before. For each survivor $D_{i,j}$ two density values are computed. The first density f_1 is computed by merging the surviving region and all its surrounding neighbors α_q with mass. This value will reflect the new overall density level if only the neighboring mass regions are claimed. The second density f_2 is the average density of all neighbors with mass, excluding the survivor. This value will indicate the density level in the surrounding regions of the survivor.

$$f_1(\overrightarrow{D_{i,j}}) = \frac{d_{i,j}^m + \sum_{q=1}^{N_b} \alpha_q^m}{d_{i,j}^a + \sum_{q=1}^{N_b} \alpha_q^a} \text{ where } \alpha_q \in \overrightarrow{d_{i,j}^b} \wedge \alpha_q^m > 0 \quad (29)$$

$$f_2(\overrightarrow{D_{i,j}}) = \frac{\sum_{q=1}^{N_b} \frac{\alpha_q^m}{\alpha_q^a}}{|\forall \alpha_q|} \text{ where } \alpha_q \in \overrightarrow{d_{i,j}^b} \wedge \alpha_q^m > 0 \quad (30)$$

Both density values are used to determine the “pulling” mode for the survivor (i.e. general or special). If the density of the region f_1 formed by merging the survivors and all its surrounding mass neighbors stay within an acceptable range ω of the average density f_2 of all its neighboring mass regions, then the region is considered stable and thus should assign the “special” pulling status. This will slow down the growing process. On the contrary the regions are unstable and thus the survivor should assign the “general” pulling status to allow for more un-restrictive growing of the region. This is described in equation 31. In our experiment ω is fixed at 0.2. This has yielded a $\pm 20\%$ tolerance range. The assignment of the pulling status for each survivor is achieved in the newly added function on line 7 of Figure 44.

$$d_{i,j}^{ps} = \begin{cases} 1, & \text{if } (1-\omega) \cdot f_2 \leq f_1 \leq (1+\omega) \cdot f_2 \\ 0, & \text{otherwise} \end{cases} \quad (31)$$

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 (i.e. line 9 in Figure 44). 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 46 is the result of the new algorithm. As compared to the results generated by the previous algorithm in Figure 45, the over and under growing problems are solved and all word groups are segmented properly.



Figure 46. Result with the amended algorithm

5.3.2 Sentence Extraction Process

Once the word extraction process stops when there is no possibility to find more word groups (i.e. the same number of word groups on two consecutive levels), the extraction of sentence will begin. The new objective is to continue to grow the word region in order for

words that belong 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 (i.e. pull in more blank regions), but only in 2 directions. They are the directions with the highest mass value in the directional mass attribute. It reflects that the formed word group is oriented along the 2 directions. An example is shown below where the directional mass of the image in Figure 47 is shown in Table 4. The two largest masses are in the top-right and bottom-left directions which reflect the correct directional path of the word “BUS”.

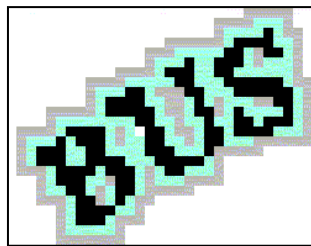


Figure 47. Sample image for directional mass

Table 4. Directional mass/weight

Direction	Mass
Top	6
Top-right	53
Right	2
Bot-right	8
Bot	2
Bot-left	26
Lelf	8
Top-left	4

There exist cases where no clear directional path can be found. This usually occurs in words that are very short in length (e.g. 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 (i.e. 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.

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. Problems will occur if we have the long word group as shown in Figure 48. 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 (i.e. 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.

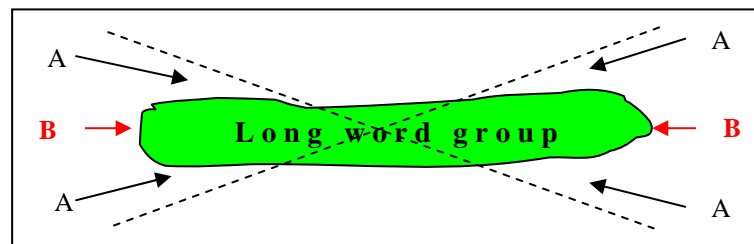


Figure 48. Targeted growing direction

As we can see in the sentence extraction main algorithm in Figure 49, 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 mass 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.

1. 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))
2. { Create pyramid higher level with (previously formed level)
3. Update the survivor neighborhood list
4. Assign pulling status (sentence) to each region
5. Assign surviving value to each region
6. Select survivors for the next higher pyramid level
7. Select children (sentence) for each survivor
8. }

Figure 49. The sentence extraction algorithm

5.4 Experimental Results

We now report some of our test cases. The first test sample is used to illustrate the results produce in the various pyramid levels. Figure 50 demonstrates the word formation stages. Each colored region represents the area covered by a physical pyramid data point. Starting from the top image which is the result on pyramid level 1, the formation of word groups progressively emerge in successive higher pyramid levels until the final formation of five unique word groups on pyramid level 5. The same is shown for the sentence extraction stages in Figure 51 where the various word groups are merged in consecutive pyramid levels. The second test case is an advertisement poster with text of varying sizes in the same sentence and aligned in a non-traditional orientation (i.e. non-horizontal). The result has demonstrated the capability of the algorithm to extract words of different sizes on the same document and even with varying orientations. Figure 52 shows the result of the word extraction. Figure 53 illustrates the merging of words to form their respective sentences. All word groups are correctly merged to the correct sentence. The third test case is a newspaper advertisement for toys. Figure 54 and Figure 55 show the output results. All sentences are correctly identified including the three sloping texts, represented by the varying colored

regions. The fourth and the last test cases are the route map of a local train system and an advertisement flyer for a camera. Figure 56 and Figure 57 show the output results for the fourth test case. All words and sentences are correctly identified represented by a bounding box for each word and sentence. The result for the final test case is shown in Figure 58.

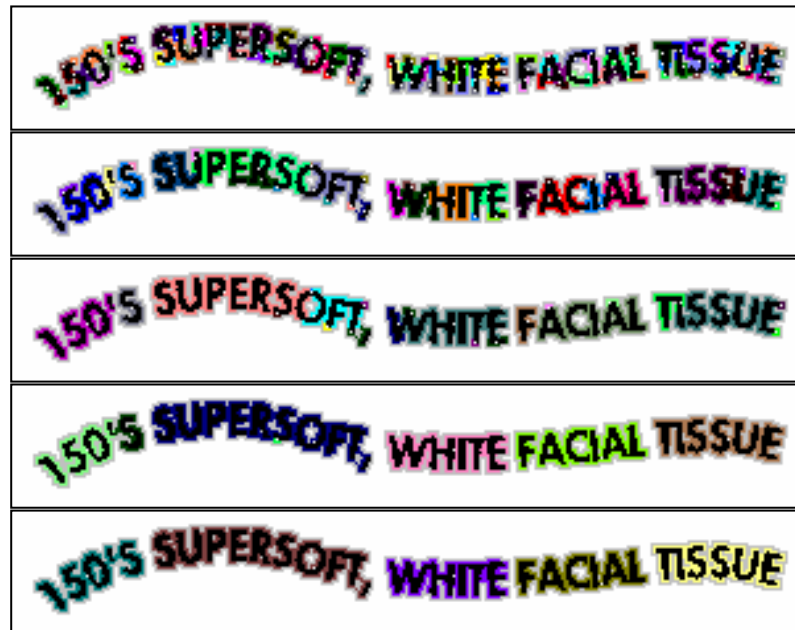


Figure 50. Sample of the word formation stages



Figure 51. Sample of the sentence extraction stages



Figure 52. Advertisement poster after word extraction



Figure 53. Advertisement poster after sentence extraction



Figure 54. Toys advertisement after word extraction



Figure 55. Toys advertisement after sentence extraction

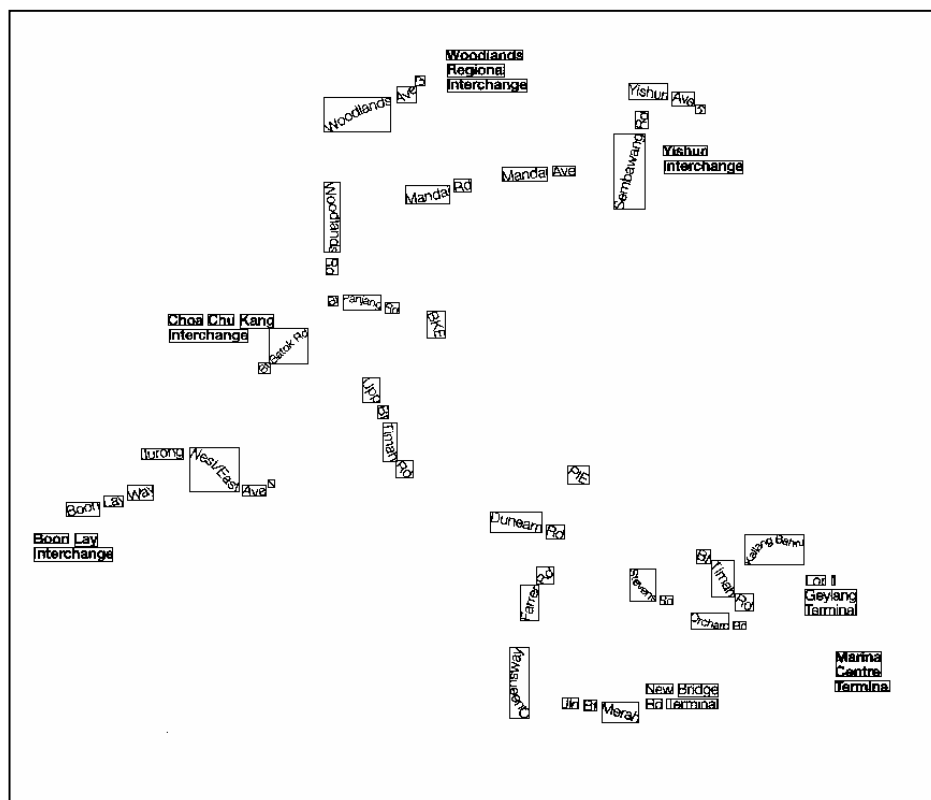


Figure 56. Route map after word extraction

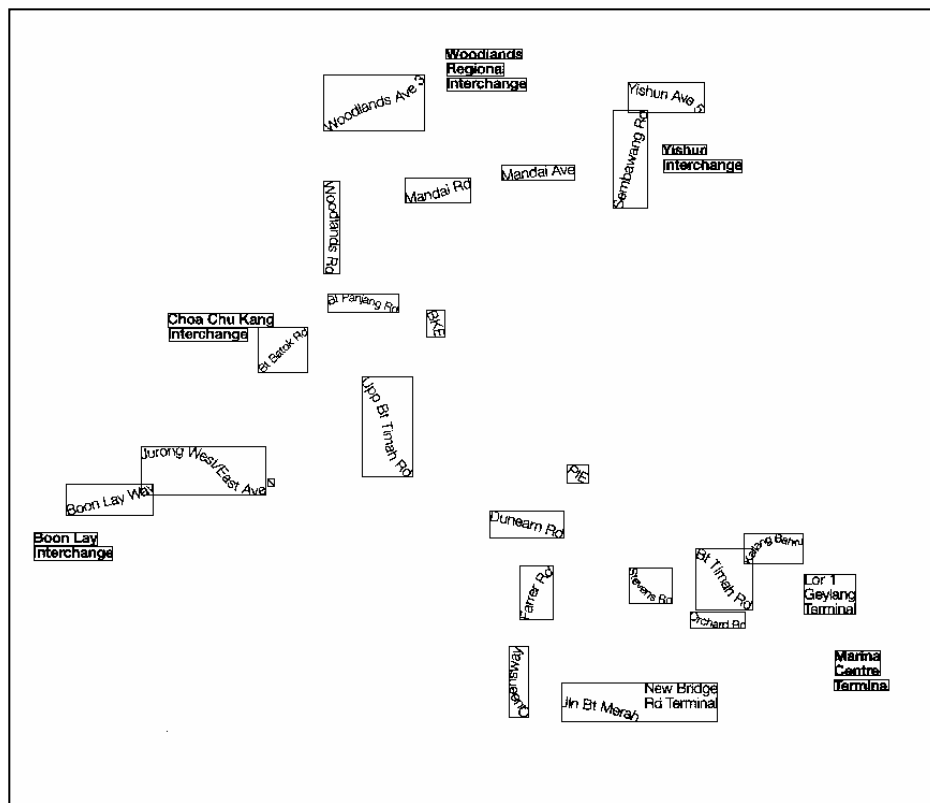


Figure 57. Route map after sentence extraction



Figure 58. Camera advertisement after word and sentence extraction

5.5 Summary and Discussion

This chapter has proposed a revised method to perform word and sentence extraction from imaged documents with large variation in the text size, font, orientation and layout within the same document. The entire algorithm is based on the irregular pyramid structure with the application of five 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 has illustrated the ability to process words of varying orientations and layout where many existing techniques have avoided. It has also demonstrated the advantage of our method in the segmentation of the various textual types (i.e. word and sentence) in clearly distinct stages (i.e. on different pyramid levels). In contrast to method in [139], this is difficult to achieve.

Chapter 6

Adaptive Thresholding in Gray Scale Images

Compared to binary images that most text extraction methods work on, gray scale images provide much more information for the extraction task. On the other hand complication also arises in determining the subject textual content from its background region (i.e. thresholding) before the actual text extraction process can begin. Differing from the usual sequence of processes where document images are binarized before the actual text extraction, this chapter proposes a new method by first segmenting individual subject areas with the help of an irregular pyramid to be followed by the binarization process. This permits the focus of attention only on the appropriate subject areas for the binarization process before text recognition. The new method overcomes the difficulty in global binarization to find a single value to fit all. It avoids the common problem in most local thresholding techniques of finding a suitable window size. In addition, a solution in the handling of noise is also suggested. As shown in the experimental results, our method performs well in both text segmentation and binarization by varying the sequence of processing.

6.1 Related Works

Most of the traditional document processing algorithms are based on binary image as the inputs. With the reduction in the storage cost, more and more document images are stored in gray scale or color format. The processing of these input images has become an important issue. Even if the segmentation task can be performed with the gray scale images, the

segmented objects must still be binarized before any recognition task can begin. As a result binarization remains as an essential pre-processing step. The accuracy in the textual recognition will rely on the validity of the binarized result. For gray scale images, there are abundances of thresholding algorithms as reported in [186, 197]. They can be categorized as either global or local binarization methods. A few famous global methods are the Otsu technique [172] and the Entropy method [173] where the attempt is to compute a single threshold value from the entire image content. The basic idea is to locate the bimodality in the gray scale levels of the input image. But as reported in many papers, this is not an easy task. With today's document images where many creative uses of gray level representing different components of the document content, the likelihood to locate bimodality is even lower. The second category is the local thresholding method, instead of finding a single threshold value to fit all the situations, the method will attempt to derive threshold value locally adapting to the varying local conditions. For document images with varying gray scale intensities, especially with the existence of graphical object, the local thresholding method is a more appropriate choice. In addition the local method can achieve a better thresholding precision that is more suitable for the target textual object which is smaller in size and closer in proximity. The adaptive binarization method [192] is one of such types where the image is analyzed within a pre-defined window size and the result from the analysis is used to select an appropriate binarization algorithm. In Fu Chang's method [159], a pre-determined window size is also used as the target region where a pixel is categorized as the foreground or the background region by examining how far away its own intensity is from the two extreme intensity values within the window area. The question arises here as to what a suitable window size should be. The size of the local window becomes a crucial factor in determining the accuracy of the thresholding result. As reported by Trier and Jain in their paper [186], too big a local window size will fall back to the same problem as in the global method. Too small a window size which is smaller than the target object will result in

an invalid derivation of threshold value that may turn true foreground area into background region. In view of this, we propose an algorithm which is also based on a local threshold, but we have solved the window size problem by deferring the binarization process to the end when individual subject areas (i.e. word regions) have been extracted. This will permit our algorithm to focus on the actual area of interest. It allows our algorithm to use a simpler thresholding method as the image feature variations are confined to the subject area rather than the entire image. In addition, the flexibility to select different binarization techniques to fit different regions also exists.

Most of the proposed methods work on binary document images. For gray scale document images, the bulk of the methods will perform the binarization of the images as discussed above before the actual segmentation of the binarized image. Regardless of which binarization methods (i.e. global or local), the handling of reverse text is always a problem for those methods that use binarization as a pre-processing step. Nevertheless there are also methods that achieve the segmentation directly from the gray scale image. The majority of the methods utilize the edge information as a guide to detect the textual regions as in [86, 138, 158, 165, 171]. As reported in [158] the higher concentration of edge pixels and the larger gradient magnitudes of the edge pixels within a local block enable an easier detection of textual region. One common problem with these methods is the verification of the true edge point. Depending on the sensitivity of the edge operator and the setting of a gray scale threshold, many false edges may appear and the work involved in verifying the true edge points are not an easy task. In addition, the same old problems as the other methods will re-surface. The task in the alignment of edges, the merging of edges and the identification of the textual regions in-between edges will still require a strong assumption of Manhattan layout and a prior determination of inter-component spacing. Hence, this approach of using edges will create even more complications than the simple connected component analysis.

Texture is another type of information utilized by some methods. One of the key motivations in using texture information is the ability to differentiate between text and non-text regions where textual region exhibits periodic structure in the horizontal direction for characters within a text line and in the vertical direction for text lines within a paragraph. Nevertheless the difficulty in designing an appropriate texture classifier and the high computational cost prohibits its popularity. Some surveys of the various texture based methods are presented in [148, 169]. There are also methods that do not belong to any of the above-mentioned categories. A method proposed by S.W. Lee et al [133] attempts to process gray scale document images by the use of vertical and horizontal projection profiles to pre-segment the character components. It is followed by a topographical features analysis to extract individual characters. Another proposed method in [125] also utilizes the topographical analysis to identify the various features (ie. saddle, ridge, ravine and hillside) to achieve the extraction task. This method works well for images without noise and is quite computationally expensive to obtain the various features for the extraction.

In our proposed method, we are using an irregular pyramid structure whose advantage is its natural clustering of neighboring regions from pyramid level to level. This allows our algorithm to process and extract text of varying sizes, fonts, layouts and orientations, thus avoiding the usual constraint on the projection profile method. It also permits and tolerates the existence of noise during the extraction process.

6.2 The Algorithm

Figure 59 shows the flow of the entire algorithm. Gray scale document image with a mixture of graphical objects and text with some noise is expected as the input. The output from the algorithm is a list of extracted words which is noise free and binarized. This will

enable the immediate incorporation of text recognition to transform the document image into searchable text format.

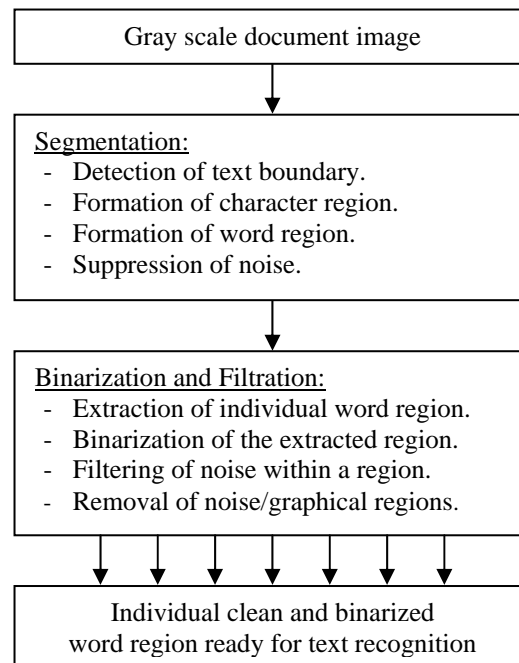


Figure 59. Flow diagram for the entire algorithm

The two main stages in the algorithm are the text segmentation stage and the binarization and filtration stage where each extracted text region is binarized and cleaned. With the help of an irregular pyramid, textual content from the document image will be segmented. The segmentation will start from the text boundary and grow inward to form character regions. Once a character region is identified, the process will begin again to merge characters into the final word regions. In this stage potential noise regions are suppressed. This will prevent such regions from interfering with the segmentation of other more promising regions. Once there is no more text region, the algorithm will switch to the binarization and filtration stage. In this stage individual isolated word regions are extracted. With the focus only on the extracted subject text area and a small amount of its immediate surrounding region, appropriate thresholding method will then be selected to binarize such region to extract the

actual textual content. Filtering of noise and big graphical object will also be done at this stage. Removal of noise is first carried out within individual local regions. With the assumption of the majority of the resulting regions being a valid textual region, undesirable regions with greatly varying features from the majority are discarded as noise/graphics. The following sections will describe in detail the process of the two main stages. The description will begin with the segmentation process where the pyramid model configuration is discussed. This is followed by the pyramid formation process that produces the subject isolated textual region on the final pyramid level. The discussion will then proceed to the thresholding and filtering of the isolated region in the second stage.

6.3 Pyramid Model

The segmentation of the textual content is based on the irregular pyramid model. The attributes required in this model are the usual group of unique pixel, neighbors list, children list, surviving value and the root node status with the addition of an intensity, local threshold and a noise status attribute. Unlike the previous model, an intensity attribute is required in the current model to retain the gray scale level of each region with 0 representing the black intensity and 255 representing the white intensity.

$$\overrightarrow{D_{i,j}} = \left\{ d_{i,j}^p, d_{i,j}^y, d_{i,j}^a, \overrightarrow{d_{i,j}^b}, \overrightarrow{d_{i,j}^c}, d_{i,j}^s, d_{i,j}^r, d_{i,j}^t, d_{i,j}^n \right\} \quad (32)$$

$$d_{i,j}^y = \left\{ v \mid 0 \leq v \leq 255 \right\} \quad (33)$$

There are two different ways in computing the surviving value in this model. This is catered to the varying objectives on the different pyramid levels. On the base pyramid level the objective is to locate the image object boundary where there is a larger contrast in the gray scale intensity level against the surrounding. The surviving value is thus defined as the

average gray level variance between the pyramid data point $d_{0,j}^y$ and all its immediate neighbors α_q^y . This value will reflect the degree of intensity variation within a local region and thus is also called the local contrast value (i.e. LC). On pyramid level 1 and above, the way in computing the surviving value changes. The value will reflect the total number of neighboring region with area greater than 1. The purpose is to encourage those regions with more neighbors to survive. This will exclude those neighboring regions with an area of 1 which are most likely the noise regions.

$$d_{i,j}^s = \begin{cases} \frac{\sum_{q=1}^{N_b} |d_{0,j}^y - \alpha_q^y|}{N_b} \text{ where } \alpha_q \in \overrightarrow{d_{0,j}^b}, & \text{if } i=0 \\ \text{count}(\alpha_q | \alpha_q \in \overrightarrow{d_{i,j}^b} \wedge \alpha_q^a > 1), & \text{otherwise} \end{cases} \quad (34)$$

The local threshold attribute $d_{i,j}^{lt}$ is used to retain the maximum intensity value found within a subject pyramid region (i.e. 0-black and 255-white). It will reflect the highest possible gray scale level that a subject region will have or the lightest possible intensity that it will appear in the image. It acts as a local threshold to identify the foreground region from the background area. Any intensity value beyond this level or lighter than this level is considered as the background region. Those below and equal to this intensity value are considered as the foreground area (i.e. darker than the lightest intensity value). This attribute is first estimated while constructing the base pyramid level and propagates through the pyramid levels. The local threshold value of a new pyramid data point $D_{i+1,k}$ on level $i+1$ will inherit the maximum local threshold value among the corresponding survivor $d_{i,j}^{lt}$ and all the non-survivors/children (i.e. β_r^{lt}) on the lower pyramid level i .

$$d_{i+1,k}^{lt} = \begin{cases} d_{i+1,k}^p = d_{i,j}^p \\ \max(d_{i,j}^{lt}, \beta_r^{lt}) \quad \forall \beta_r \in \overline{d_{i+1,k}^c} \end{cases} \quad (35)$$

The final additional attribute is the noise status attribute $d_{i,j}^n$. This attribute will assist in the handling of noise region within the pyramid model. Once a region is classified as noise, its noise status attribute will become true. This will allow special treatment of such region during the pyramid formation process. Pyramid data points which are classified as noise can be the children of other survivors, but they are not allowed to claim their own children. On one hand this will eliminate the existence of holes on the higher pyramid level due to the absence of noise regions. On the other hand it also avoids the unnecessary growing of such regions into size that may affect the subject region.

6.4 Segmentation

The objective of this stage is to segment the input image into multiple regions. Although non text regions will also be segmented through the process, they are removed in the later stage. The majority of the regions are assumed to be the subject textual area. The segmentation task is achieved through the construction of pyramid structure as shown in Figure 60. Subject areas are isolated while pyramid levels are being constructed. In the current method the segmentation process is divided into two pyramid formation stages. The formation of the base pyramid level (i.e. Line 1-5) and the construction of all higher pyramid levels (i.e. Line 6-17).

1. Create the base pyramid level.
2. Select neighbors (8connectivity pixels).
3. Assign surviving values (local contrast values).
4. Select survivors.
5. Select children (2 regions clustering).
6. while (exist more isolated word regions).
7. { Create next pyramid level.
8. Select neighbors (if children are neighbors).
9. Assign surviving values (# of mass neighbors).
10. Select survivors.
11. Select children.
12. – If (formation of character)
13. { claim only neighboring regions with darker
14. intensity than the local threshold value. }
15. – else (formation of word)
16. { claim all neighboring regions. }
17. }

Figure 60. Pyramid construction process

6.4.1 Base Pyramid Level Formation

Pyramid construction will begin from the base level where the original document image is used as the input. The main purpose on this base pyramid level formation stage is to locate and construct suitable boundary regions. With these boundary regions as the pivot regions, it will allow the subsequent pyramid formation stages to grow inward into the core area of the subject regions. The first task in this stage is the detection of boundary point where such point will have higher intensity contrast against its background area. The local contrast value (i.e. *LC*) as described in section 6.3 is used as the surviving value. The point which is the local maximum with the highest contrast is selected as the survivor. Such a point or pixel will have the greatest likelihood of being the boundary pixel. Nevertheless this process also produces another type of undesirable local maximum which has very small intensity variance against its surrounding. Most of these pixels are discovered to be just small noises belonging to the background area. In order to avoid the interference of these pixels in the segmentation process on the higher pyramid level, they are marked as noise. To facilitate the identification of such noise pixels, a global contrast value (i.e. *GC*) is used. The

value is derived by taking the overall average of all the local contrast, which is also the surviving value on pyramid level 0, with value greater than 0. The purpose of avoiding $LC=0$ is to prevent such flat regions from affecting the estimated global contrast value (i.e. GC). This average will reflect the degree of contrast that the majority of the subject regions have against the background area. The global contrast value will act as a good baseline to discriminate against the noise regions or regions with very low contrast (e.g. background graphics).

$$GC = \frac{\sum_{j=1}^{N_0} LC_{0,j}}{\text{count}(LC_{0,j} > 0)} \text{ where } LC_{0,j} > 0 \quad (36)$$

Once the appropriate boundary pixels or the survivors are selected, the child selection process will attempt to grow from this boundary point into a boundary region. The aim here is for the survivor (i.e. boundary pixel) to claim only pixels belonging to the foreground area. In order to claim the right set of foreground neighboring pixels, further analysis is required before the actual claiming. The child selection process is split into three stages as shown in Figure 61. From line 1 to 8 is the first labeling stage. Line 9 is the connected component analysis stage and finally the actual claiming stage from line 10 to 12.

- | |
|--|
| <ol style="list-style-type: none"> 1. Locate the lightest intensity within the 3x3 region 2. Locate the darkest intensity within the 3x3 region 3. For each pixel within the 3x3 region 4. { If pixel intensity closer to lightest intensity 5. Label pixel as background 6. Else 7. Label pixel as foreground 8. } 9. Perform 8-connected component analysis(foreground) 10. If number of foreground component=1 11. { Marked all foreground pixels as children 12. Locate the lightest intensity among the foreground pixels 13. } |
|--|

Figure 61. Child selection process for the base pyramid level

In the labeling stage, each pixel within the 3x3 local regions is labeled as either the foreground or the background pixel. The criterion we used for this labeling task is borrowed from Fu Chang's [159] local thresholding method where the 2 extreme intensity values (i.e. darkest and the lightest) are used to threshold the textual area from its background. In our algorithm this technique is modified and applied within the local region to label each pixel. The darkest and the lightest intensity values within the 3x3 local regions are identified (i.e. line 1-2). A pixel is labeled as foreground if its intensity is closer to the darkest intensity value. Otherwise, it is labeled as the background pixel if its intensity is nearer to the lightest intensity value. After the labeling process, all pixels within the local region are put through a connected component analysis which is the second stage. The analysis is based on the 8-connectivity criterion and the process will locate the number of connected foreground component within the 3x3 region. The process attempts to locate a clear layout separation between the foreground and the background area. If the selected survivor is a true boundary pixel, then all the foreground pixels should cluster as a single component. Some examples of the valid boundary pixel layout are shown in Figure 62. However if the foreground pixels are clustered into fragments then the boundary status of the selected survivor will be in question. Figure 63 shows some of the invalid boundary pixel layout samples.

B	B	F
B	F	F
B	B	F

B	F	F
B	F	F
B	B	B

B	F	B
B	F	B
B	F	B

Figure 62. Samples of the valid boundary pixel layouts

B	F	F
B	B	B
F	F	F

B	F	F
B	B	F
F	B	B

F	B	B
F	B	F
B	B	F

Figure 63. Sample of the invalid boundary pixel layouts

Based on the connected component analysis the final stage in line 10-13 to claim the neighboring pixels as children will begin. If there are only two components (i.e. foreground and background), it is considered a valid boundary region and the darker group or the foreground area is claimed by the survivor as its children. If the connected component analysis produces more than 1 foreground connected component, the survivor is discarded with no selection of any children. Such regions will not be a good candidate as the boundary region. While claiming the foreground area, the process also attempts to identify the lightest intensity value among all the foreground pixels. This value will act as a local threshold between the subject foreground region and its surrounding background. This value is recorded in the local threshold $d_{i,j}^l$ attribute. After this base pyramid level formation stage some pyramid data points may become orphan where the non-survivors have no neighboring survivors. The treatment in this pyramid model is to allow all these data points to survive onto the next pyramid level as a stand alone data points with no children. This will avoid the existence of holes on the higher pyramid level.

The child selection process used in this stage belongs to the second alternative as described in section 3.2.4. Due to the requirement to perform detailed analysis of the survivor surrounding environment, the usual way in allowing the non-survivor to initiate the linkage with the most suitable survivor can not be used. In order to improve the computation efficiency there is no further verification of whether is there a more suitable survivor in the neighborhood of a non-survivor. Since the main aim in the current segmentation stage is to locate an estimated subject region to facilitate a more accurate processing of the region at a later stage, the constraint is removed consciously to avoid excessive computational load.

6.4.2 Higher Pyramid Level Formation

Once the boundary regions are located, the next task will be the formation of the character regions and followed by the word regions. These objectives are achieved while constructing the subsequent pyramid levels. The flow of the process is shown in Figure 60 from lines 6 to 17. On the higher pyramid level, the assignment of the surviving value changes from the local contrast value to the number of neighboring regions with areas greater than 1. The motivation here is to speed up the merging process among neighboring boundary regions. As a result a region with more non-unit neighbors is given a higher priority to survive. Once the appropriate surviving region is selected, the child selection process (i.e. lines 11-16) will start to evaluate its neighboring regions. Neighbors are selected if their intensity values are darker than the local threshold value $d_{i,j}^n$ of the survivor which is recorded during the base pyramid level construction process. The reason for this criterion is to encourage regions to grow towards the foreground textual area rather than the background region. Local threshold value (i.e. lightest intensity value among the boundary regions) is used to ensure the growing is in the right direction. After a few pyramid levels, the child selection process will start to run out of neighboring regions satisfying this condition. This is a signal to indicate that an isolated region (i.e. character) is formed. Once the character region is formed, the algorithm will relax its child selection criteria to claim all neighboring regions (i.e. lines 15-16). The new task now is to grow the isolated character region into its neighborhood until multiple character regions are merged to create a second and larger isolated region (i.e. word).

6.5 Binarization and Filtration

Once there is no more new isolated region formed on two consecutive pyramid levels, the segmentation process ends. The algorithm will switch to the next stage of processing where

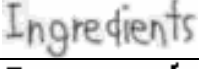
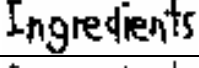
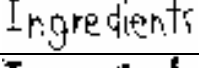

individual word regions are extracted. With the focus only on each region, three different global thresholding methods are used. They are the average intensity thresholding method, binarization based on the local threshold value as described section 6.3 and the high-low method we have used for the boundary detection. In order to determine which method produces the best result, the three resulting binary images of a word region are put through a connected component analysis and a deviation value is computed for each image by the following formula.

$$\begin{aligned}
 CCAveArea &= \frac{\sum AreaCC_x}{NumCC} \\
 Deviation\ value &= \frac{\sum |CCAveArea - AreaCC_x|}{NumCC \cdot CCAveArea}
 \end{aligned} \tag{37}$$

For each connected component x found in the binary image of the word region, the area of each component (i.e. $AreaCC_x$) is counted and the overall component area average (i.e. $CCAveArea$) is obtained. With this average as the pivot point, the total area deviation of each component x from this point is calculated and averaged by dividing the total area deviation by the number of components (i.e. $NumCC$). Finally this value is normalized by the component area average to allow comparison among the three methods. The deviation value reflects how stable all the connected components are in each image in terms of the component area. A low deviation value shows that most of the components are of relatively equal size. A high deviation value represents the existence of greatly varying component size within the image. With the assumption that the size of most characters within a word will not vary significantly, a low deviation value will translate into a good thresholding method which enables the majority of the characters to be isolated as a component. This is a very crucial factor for good text recognition. Table 5 shows a sample of the thresholding selection process. The average threshold method is selected in this case due to its lowest

deviation value. The high-low method produces a very fragmented image that shows the problem of over-thresholding. In contrast the local threshold method is under-thresholded as most character regions are wrongly merged as a single component.

Table 5. Thresholding method selection

Threshold Types	Images	Number of CC	Deviation values
Original			
Average		10	0.30
High-low		17	0.61
Local		9	0.91

Once each extracted region is binarized, filtering of noise within each region will begin. The average component area computed in the previous step is used again to filter any component size that is very small (i.e. below 10% of the average size). Such components are usually the noise regions. The final step is the removal of noise or big graphical object region. Filtering by the area method is again used.

6.6 Experimental Results

The following are some sample test cases extracted from a total of 30 images we have tested. Figure 64 shows the entire process for a small test image. The left hand side is the output extracted from some of the pyramid levels. Starting from the original image and down to the final pyramid level where 3 regions are identified. The right hand side shows the output after individual words are extracted, binarized and finally the removal of small noise. The dot of the character “i” is removed as small graphical component. As we can see, during the pyramid construction process some of the noises are suppressed (i.e. from 2nd left to 3rd left).

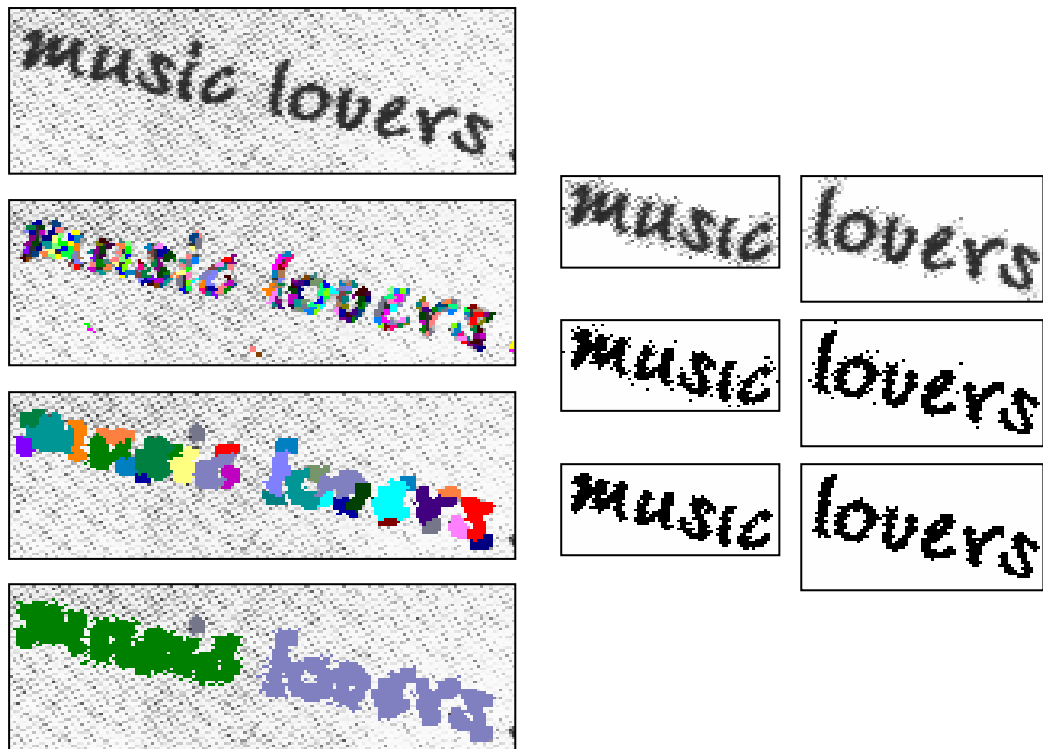


Figure 64. Test sample with noise

Figure 65 and Figure 66 shows two other test cases. On the left is an image with big graphical content and the textual area in this image has a lower contrast against its background area. In addition to the existence of noise, the right image has textual content with much darker intensity as compared to its background region. Both test cases demonstrate the ability of our algorithm to process image document with the existence of graphic/noise and even with varying intensity contrast of its textual content from the background area.

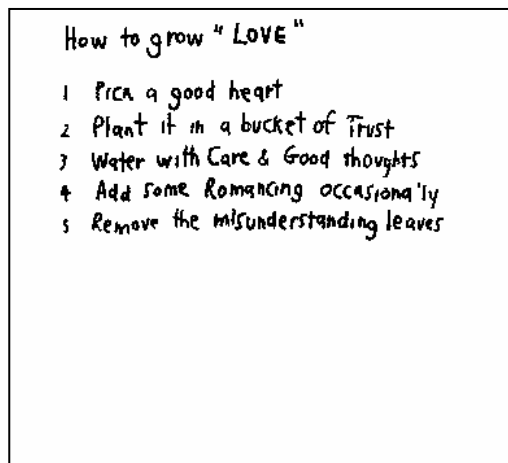


Figure 65. Sample image with graphic

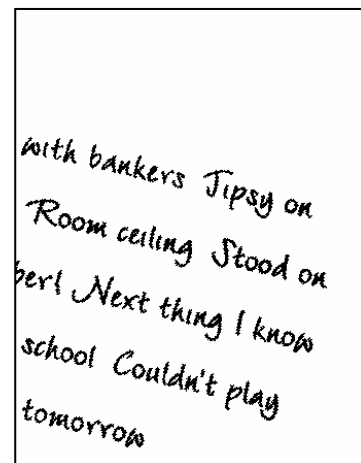
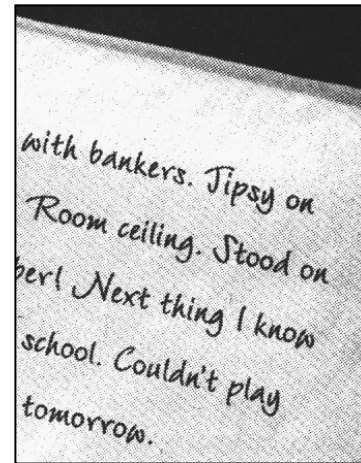


Figure 66. Sample image with noise

Figure 67 shows a test image with a very low textual content contrast. The resulting segmented word regions are represented as bounding boxes at the middle image. The final

binarized version is shown at the bottom image reflecting the combination results obtained by selecting the most appropriate thresholding method for each word region.

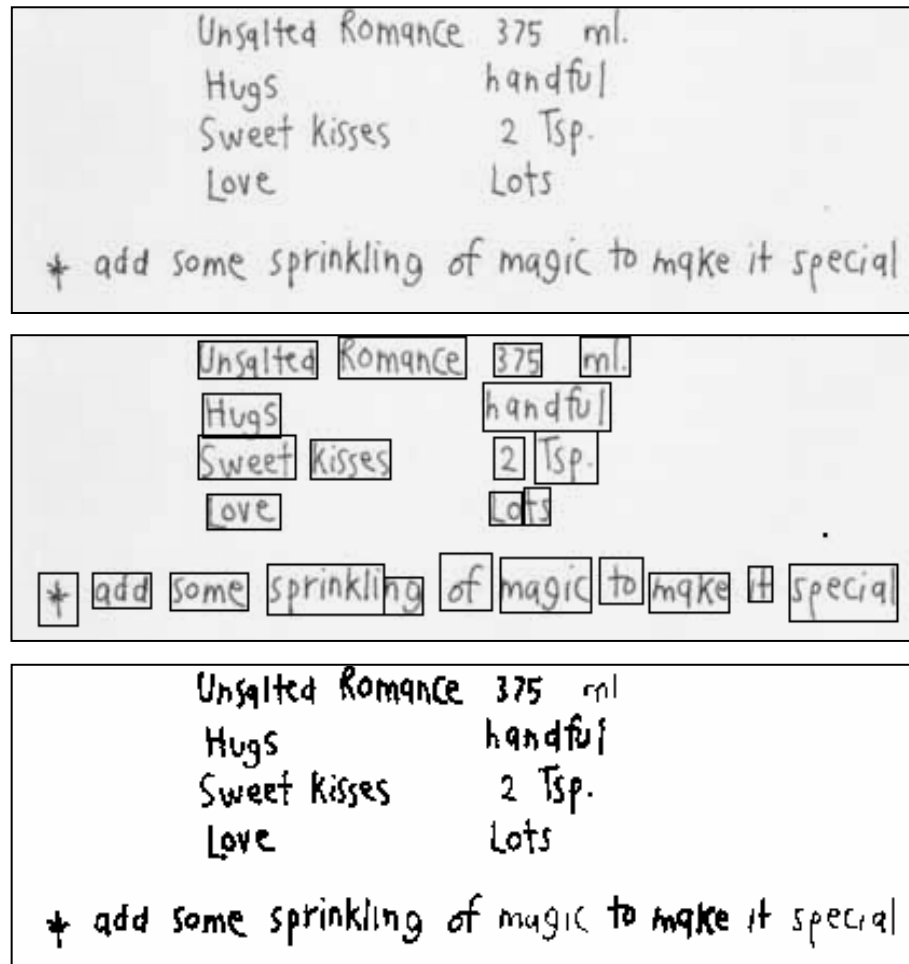


Figure 67. Test image with low contrast

Figure 68 demonstrates the ability of our algorithm to extract even textual content of varying orientation from gray scale image. Each segmented word region is represented as a bounding box. Figure 69 shows the results obtained from an image with some background graphics. Finally a test image with varying background contrasts is shown in Figure 70. The results produce by our method and two other very popular thresholding methods (i.e. Otsu and the Entropy) are presented. The result demonstrates the ability of our method in removing background noises whereas noises remain in the results produced by the Otsu and

the Entropy method. From the result of 30 test cases we have experimented, our recall rate is 85% and a precision rate of 87%. Recall is defined as the number of correct words detected by the system divided by the number of words present in the document. Precision is defined as the number of correct word detection divided by the number of words picked up by the system.

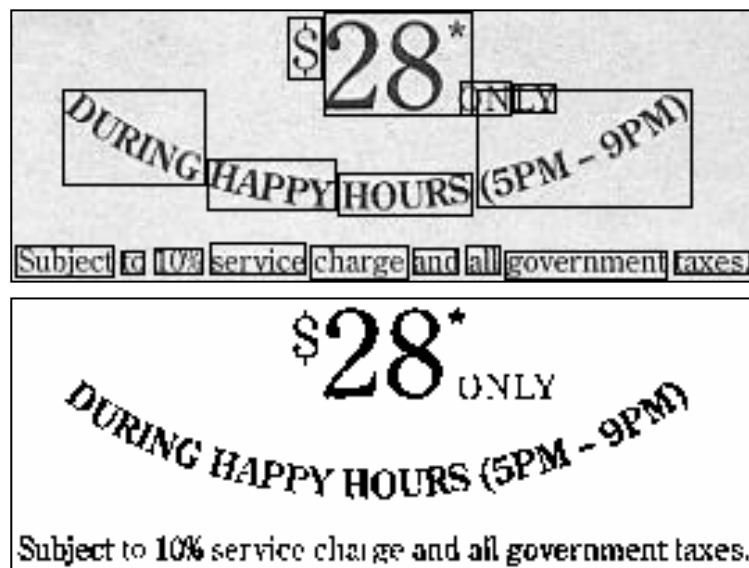


Figure 68. Test image with varying orientation

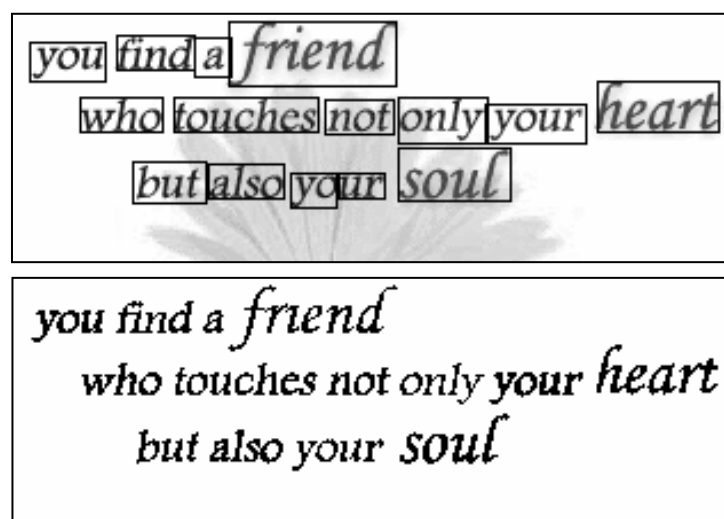


Figure 69. Test image with background graphics

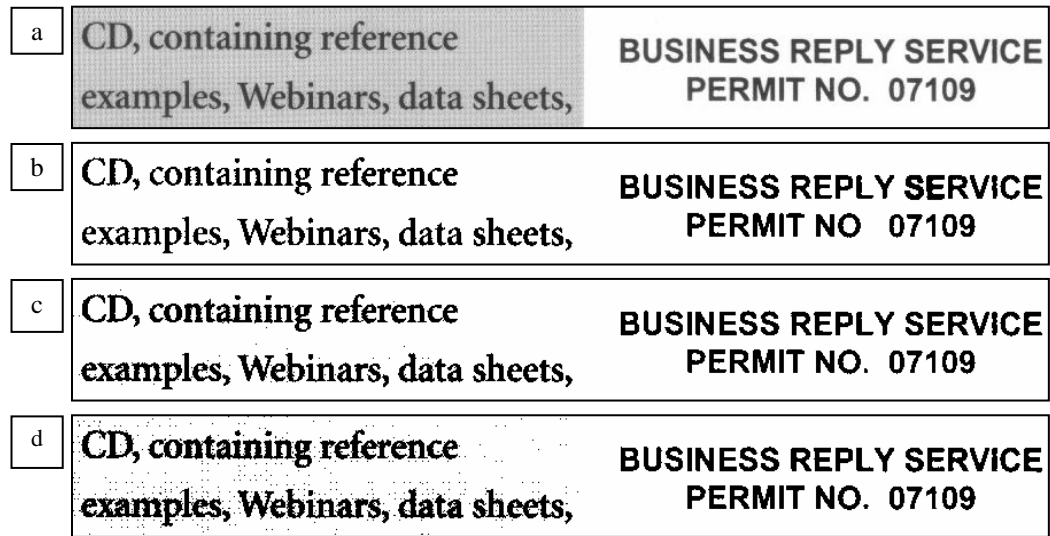


Figure 70. a:Original image, b:Our algorithm, c:Otsu thresholding, d:Entropy thresholding

6.7 Summary and Discussion

We have proposed a new methodology to segment textual regions from gray scale image document. Our technique defers the usual pre-processing steps in doing binarization and filtering of noise/graphical object till the post-processing steps after the actual textual subject regions are extracted. Such changes have been proven to be effective as shown in our experimental results where focus of attention is achieved thus allowing a simpler binarization/filtering process and also the flexibility to use different methods for different subject regions. The paper also introduces a new way in using the irregular pyramid to perform the extraction from gray scale images. The process has shown to be effective in identifying the subject boundary region, followed by the full extraction of the complete word. In addition, the ability to extract text from interfering background objects and also text of varying orientations are also shown in the test cases.

Chapter 7

Textual Segmentation from Color Document Images

This chapter presents the result of an adaptive region growing segmentation technique for color document images using an irregular pyramid structure. It is published in [71]. The emphasis is in the segmentation of textual components for subsequent extraction in document analysis. The segmentation is done in the RGB color space. A simple color distance measurement and a category of color thresholds are derived. The proposed method utilizes a hybrid approach where color feature based clustering followed by detailed region based segmentation is performed. Clustering is done by merging image color points surrounding a color seed selected dynamically. The clustered regions are then put through a detailed segmentation process where an irregular pyramid structure is utilized. Dynamic and repeating selection of the most suitable seed region, fitting changing local condition during the segmentation, is implemented. The growing of regions is done through the use of multiple seeds growing concurrently. The algorithm is evaluated according to 2 factors and compared with an existing method. The result is encouraging and demonstrates the ability and efficiency of our algorithm in achieving the segmentation task.

As compared to the binary and gray scale document images, color document images contain much richer set of information. The use of varying colors allows the subject textual area to be distinguishable from the background and non-subject regions. On the one hand, the color attribute provides an additional avenue for the extraction of textual components. On the other hand, it also introduces new complexity and difficulties. First is the variety of color spaces that can be used where each has its pros and cons. No single space is general

enough for all uses. Second is the distance measurement problem. Till date there is yet a standard and precise way of measuring color distance, which is a crucial parameter for all segmentation tasks. Third is the storage requirement. In a frequently used 24-bit true color image, the storage requirement will increase by 3 times that of a gray scale image. Due to this increase in the representation dimension, the number of unique color points will also increase to 16 millions. This will intensify the processing complexity. Nevertheless, there are various proposed methods attempting to overcome the problem and complexity in order to benefit from the advantage of using the color attribute in the segmentation process.

7.1 Related Works

Color document images have gained increasing popularity in its usage. There are many proposed techniques for color segmentation. As categorized by [216], color segmentation can be divided into feature-space based, image-domain based and physics based techniques. Feature-based methods focus their attention only on the color features where color similarity is the key and only criterion to segment image content. Color quantization as discussed in the following paragraph belongs to the feature-space based method. Spatial relationship among color is ignored. This has resulted in a problem where the segmented regions are usually fragmented. Extra and elaborate post-processing is required to retain the compactness of the regions. Image-domain based methods belong to a category of methods that take spatial factors into consideration. The technique utilizes both color and spatial factors in its homogeneity evaluation. Physics based techniques are mainly used to process real scene images where the physical models of the reflections properties of materials are utilized. Despite the large number of proposed color segmentation algorithms, only a handful of them have directly addressed the document image processing domain with the focus in text segmentation and extraction. The key requirement in this domain is not so much in attempting to find the best approximation in terms of the color features. Rather, the

emphasis is on how well the segmentation process can achieve the retention of major document components (i.e. text or non-text) and at the same time realizes the compactness within each component. The challenge is to have just a sufficient number of unique colors for the former and minimizing the color uniqueness to attain the later.

Color quantization has the same effect as the gray scale binarization process to reduce the representing state of the image content to simplify the processing task. In terms of the color domain, it may also be due to the requirement in adapting to a specific hardware constraint. There are two main types of color format. They are the GIF format which has 256 unique colors and the JPEG format that supports true color of 24-bit (i.e. 16 million colors). In view of the number of unique colors in the true color format the need to reduce the representing state is definitely more pressing than the GIF format (TIF and PNG are also becoming popular in recent years). Unlike in gray scale binarization, color quantization will reduce to a certain number of representing states other than the binary state. This will enable the retention of the richness in using color to differentiate between different image objects during the segmentation stage. The dividing line between color quantization and color segmentation is very thin. By accurately quantizing an input image, it will result in the correct segmentation of the image object. Moreover this will require finer control of the quantization process. We will leave this process to the segmentation stage and focus on the reduction of the most similar color point in this pre-processing step. On the one hand, the quantization process will be required to reduce as much unique color points as possible to reduce the segmentation work load. On the other hand it must also not reduce to such an extent that it will affect the subsequent proper segmentation of the image. There are several common quantization methods like the bit-dropping technique, the median cut algorithm, the popularity algorithm and the Octrees method. Some of these methods are reported in [201, 206]. On one extreme is the bit-dropping technique used by Jain and Yu in their paper

[85] where every truncation in the lowest order bit of the color bands (i.e. R,G,B) will reduce the intensity level of each band by half. As a result a truncation of the lowest order 6bits will result in 64 unique colors. Although it is the simplest and the fastest, it is also the most inaccurate method due to the “forced” clusterization of color points without considering the actual color distribution of the input image. On the other end we have the histogram-based method where the actual color content of the image is analyzed to obtain the most occurrence color points, to be followed by allowing the remaining color points to join with the nearest peak color. Despite the improved accuracy, the use of the histogram has created additional computation load. There is also other form of quantization method. A. Antonacopoulos and D. Karatzas [81] proposed the use of the usual connected component method, amended to work within the color space to merge pixels of similar color.

As in the gray scale domain, the majority of the proposed methods utilize the feature-space based color quantization method as a pre-processing step which is followed by the usual textual segmentation technique in performing smearing, connected component analysis or the XY-cutting. The main difference is in the execution of these techniques on different color layers. Jain and Yu [85] use the connected component method in the localization of texts from images and videos after quantizing the color image with the bit-dropping method. In [211] a histogram based technique is used in the quantization of color. It follows by a XY-cutting approach in discarding the homogenous background region and finally the merging of the remaining regions within the same color cluster. Connected component method and the RLSA techniques with horizontally aligned text assumption are also used in [202, 207]. Besides sharing the same problem as in the processing of the binary and the gray scale images in the requirement of a Manhattan layout, a new problem which generates fragmented textual result surfaces. The method proposed in [76] is a clear illustration of the problem faced in using feature-space based technique where a very

intricate post-processing stage in performing connected component analysis is required to merge fragments of textual region.

In order to solve the problem faced in the feature-space based technique; there are some proposed methods that attempt to incorporate the spatial factor while performing feature-space based color segmentation. In [217], an interesting color clustering concept is used in the RGB color space. It divides the RGB Cartesian space into multiple numbers of fixed size cubes where each cube will hold the occurrence of pixels having the color defined within the cube. A pointer chain is then constructed by analyzing the 26 potential neighboring cubes to locate the local maximum with the highest number of occurring pixels. It performs the clustering in a three dimensional space. In order to take into consideration of the spatial factor a 4th dimension is added. The additional dimension is defined by dividing the image plane into horizontal strips where each strip will contain a fixed number of image rows. Each bin in the 4th dimensional space will now contain the pixel occurrence of a specified color range located along a certain strip. Although it overcomes some of the problems faced due to the lack of spatial factor, the effectiveness is restricted to the size of the cube and the widths of the strips. The accuracy of the segmentation result will also depend on where the color space and the image plane are divided. Despite the problem it is an efficient method capitalizing on the efficiency and simplicity of using histogram and at the same time incorporating the spatial factor in the clustering progress. Another conceptually similar system proposed in [79] also attempts to incorporate spatial information into a feature-based type of color clustering by computing a spatial proximity among colors within a local region.

Under the image domain based color segmentation category, splitting/merging and region growing are two main techniques. The common processing steps are the selection of a seed

region, the growing or splitting of regions from this seed point, the merging of homogenous regions and a stopping criterion for growing or splitting. In terms of textual segmentation, this is not a frequently used technique despite its ability to truly take both color and spatial factors into consideration in segmentation. By nature this is a sequential process where each pixel and all its neighbors have to be evaluated. The processing order becomes critical at points with the same homogeneity value. The selection of a suitable seed region is another problem where the initial selected seed region may dominate the growing or splitting process. Although many proposed methods attempt to solve this problem by making the best selection, the suitability of a region being a seed point does change during the segmentation process. A final problem is in the determination of an appropriate growing criterion (e.g. color distance) which dictates the growing path of a region. Most methods choose to use an empirical value or leave this as a program parameter for the user to specify. These problems are reported in [208, 216, 221]. One example of the region growing approach in textual segmentation is shown in [89] where a general region-based method of growing a region into the nearest neighbors with the least differences in the color intensity without any explicit selection of seed point is implemented. One common difficulty in such method is the determination of a segmentation threshold used to define constitution of a homogenous region. Another example as proposed by D. Karatzas and A. Antonacopoulos [83] employs a fuzzy approach by using a derived connections ratio reflecting the degree of connectivity and the color “closeness” within the $L^*a^*b^*$ color space between quantized components as the inputs to produce the propinquity values for all possible pair of components. Merger will then begin with the sorted list of propinquity values.

In contrast to all the above methods, our contributions are in 4 areas. First is in the area of color measurement where a simple measurement method in the RGB color space is derived as described in section 7.2. Second is in the area of color quantization where an efficient

method without the need of a color histogram is proposed in section 7.3.1. Third is in our region growing method where seeds are selected dynamically and repeatedly to suit the best local condition, which avoids the problem of having a fixed seed dominating the entire growing process. The problem of sequential processing encountered by the other region growing methods is also addressed by having multiple seeds to grow concurrently. The fourth area is in our use of the irregular pyramid structure which differs from the traditional pyramid in that it constructs the pyramid from an intermediate level instead of the original base level in pixel format. It has greatly enhanced the processing speed. The Last two contributions are described in section 7.3.3.

7.2 Color Space and Distance Measurement

In color segmentation the RGB color space is most commonly used where each color is represented by a triplet red, green and blue intensity. HSI is another common color space where a color is characterized by the degree of Hue, Saturation and Intensity variance. Another category of color space is based on the CIE color model. The main aim of this model is to provide a uniform color spacing that facilitates direct measurement of color distance. $L^*a^*b^*$ is one of such color space. While selecting a color space for image segmentation, the key consideration is the ability to have an accurate and efficient way to measure color distance. Color distance is used as a measurement of color similarity where pixels/regions satisfying a certain degree of color homogeneity are grouped to form a cluster. In this aspect the CIE $L^*a^*b^*$ color space seems to be the most promising where the color distance can be computed directly from the Euclidean distance of the Lab coordinates (i.e. ΔE). In spite of this, not many proposed methods make use of this color space. This may be due to the complexity of its conversion process from the RGB color space and also some controversy in its accuracy. In HSI color space, color distance is frequently measured along the individual axis separately. Although the Hue component alone can be used to

measure color similarity as in [77], it is not sufficient for detailed segmentation. Both Saturation and Intensity value must also be utilized for finer segmentation results as in [76]. In addition to this requirement to analyze the three axes separately, a further complication exists when the Saturation value is low where all colors look almost the same despite varying Hue value. This is reported both in [216] and [77]. In view of these problems we have decided to use the RGB color space. It is efficient because no conversion is required. Although it also suffers from the non-uniformity problem where the same distance between two color points within the color space may be perceptually quite different in different parts of the space, within a certain color threshold it is still definable in terms of color consistency.

In order to analyze the color distance measurement in the RGB color space for the definition of color similarity, we have conducted an experiment. The experiment starts with a pivot color. It will then randomly generate 620 non-repeating variation of color points with the same distance from the pivot color computed by using the same distance function (i.e. Euclidean or Manhattan). The color point is a 20x20pixels square which is about the size of a 12point character. All colors are then visually inspected by 10 human subjects to determine its similarity. Each observer will vote for one of the 7 categories as shown in Table 6. This process is then repeated for color distance, in the range of 10 to 500 by a step of 10, computed by different distance measurements. The final result is obtained by taking the majority vote. The result of the experiment reveals that the Manhattan distance is a better distance measurement where the generated color points exhibit a more stable visual color similarity. In contrast, the Euclidean distance measurement will produce a wider variation of color perception with the same color distance. This finding shows that color is formed by the additive of the varying red, green and blue intensity and not so much of the physical Euclidean distance between the color points. The various categories of threshold

limit obtained through the experiment are shown in Table 6. It is categorized into 4 main groups. The first group belongs to those below 71 where the same color is observed with a very low intensity variance. The second group ranges from 71 to 120 where the color appears to be from the same color series (e.g. dark/light brown) with varying degree of intensity. The third group ranges from 121 to 190 where different colors are observed with varying color ranges. Color above 190 becomes quite random and thus is considered as undefined and cannot be interpreted.

Table 6. Categories of color threshold limits

Threshold	Visual inspection result
10 to 30	Same color.
31 to 70	Same color, low intensity variance.
71 to 90	Same color series.
91 to 120	Same color series, low intensity variance.
121 to 150	Difference color, small color range.
151 to 190	Difference color, wider color range.
Above 190	Very random occurring color.

Based on this experimental result, the following color distance function is derived as shown in equation 38. The function will compute the total absolute variation of the respective RGB values between 2 color vectors (i.e. $\overrightarrow{C_x}$ and $\overrightarrow{C_y}$). The further additive factor σ is to discriminate between well distributed color variance among all RGB values and those with un-even variance distribution. The former reflects better color consistency than the later. If the distance is within the threshold T_l then the 2 colors are considered “close”. Otherwise they are treated as 2 unique colors. Although the use of a single threshold to determine the “closeness” between two colors may not be the most precise way of color measurement in the RGB color space, in our context for text segmentation it is more than sufficient. In [210], the authors also make use of a human perception evaluation of color differences to guide the color clustering process.

$$\begin{aligned}
dist(\vec{C_x}, \vec{C_y}) &= r' + g' + b' + \sigma \\
r' &= |C_x^r - C_y^r|, g' = |C_x^g - C_y^g|, b' = |C_x^b - C_y^b| \\
\sigma &= (|r' - g'| + |r' - b'| + |g' - b'|) / 3 \\
close(\vec{C_x}, \vec{C_y}, T_l) &= \begin{cases} true, & \text{if } dist(\vec{C_x}, \vec{C_y}) < T_l \\ false, & \text{otherwise} \end{cases}
\end{aligned} \tag{38}$$

7.3 Proposed Method

Our proposed method is a combination of color feature based and region based color segmentation process. The algorithm utilizes a color feature based technique to perform fast segmentation of regions with very close colors in a pre-processing stage. Based on the result, region-based growing method is then employed to perform detailed segmentation of the remaining regions taking both color and spatial factors into consideration.

7.3.1 Pre-processing Stage

In a 24-bit true color input image, the number of unique colors will frequently exceed half of the image size. Most of these colors are perceptually close and cannot be differentiated by human beings. In our study of the RGB color space, a color variance of 30 and below will fall into this category. As a result the pre-processing stage will attempt to aggregate colors within a boundary of $T_l=15$ surrounding the pivot color as a cluster. This will ensure that the maximum color distance among all colors inside the cluster is within 30. Due to the usually large number of color points, we employ a simple and yet efficient way of clustering. The process will loop through the entire image. As it moves, pivot color points are identified. A color point with variance exceeding 2 times the T_l limit is inserted as a new pivot color. Those within the color limit are clustered with the closest pivot color. The

efficiency and accuracy of this process lie in between fixed partitioning of the color space as in bit-dropping technique and the selection of a color seed by giving preference to a bigger region [210]. Our proposed method is more efficient without the pre-requisite to build a histogram for the selection of a suitable color seed. It is also more accurate than the bit-dropping technique by building the pivot color list dynamically as it loops through the image. As compared to bit-dropping where a fixed partition is used regardless of the actual color distribution, this process will avoid non-existent color points. The final output from this stage is a group of pixel clusters having color similarity within a limit of 30.

7.3.2 Pyramid Model

The pyramid data point attributes used in this model is almost the same as the model used in the previous section to process gray scale document images. The differences are in the way the intensity value is represented, the derivation of the surviving value and the computation and the application of the local threshold attribute. There is no noise status attribute in this model.

$$\overrightarrow{D_{i,j}} = \left\{ d_{i,j}^p, \overline{d_{i,j}^y}, d_{i,j}^a, \overline{d_{i,j}^b}, \overline{d_{i,j}^c}, d_{i,j}^s, d_{i,j}^r, d_{i,j}^t \right\} \quad (39)$$

Since the inputs are the color document images, the intensity attribute will become a vector attribute holding the red, green and blue intensity value with 256 intensity levels. The computation of the color distance between two pyramid data points will follow the Manhattan color distance measurement as described in the previous section.

$$\overline{d_{i,j}^y} = \left\{ Y_{i,j}^r, Y_{i,j}^g, Y_{i,j}^b \right\} \quad (40)$$

In this model the surviving value is derived by allowing each pyramid data point to vote for its closest neighbor satisfying an eligibility function. The function will evaluate the closeness between two data points. At the end of the voting process, the surviving attribute of each pyramid data point will reflect the total number of “close” neighbors. This will enable those data points with more neighborhood support to survive and eventually claims these “close” neighbors as children.

In the previous processing of gray scale document images, the local threshold attribute is used to retain the lightest intensity value which defines the foreground region. Any pixels with intensities darker than this value are considered as the foreground regions. In the current model the local threshold attribute has a different definition and also varies in its application. Although it is also a threshold value that is determined locally within the surrounding neighbors, it is computed in a different way. The local threshold value of a pyramid data point $D_{i,j}$ is defined in equation 41 as the minimal between a local contrast value and another threshold value T_3 . Just like the local contrast value uses in the previous method, it is obtained by taking the average of all the color distances between the color vector of a pivot pyramid data point $\overrightarrow{d_{i,j}^y}$ and the color vectors of all the surrounding neighbors $\overrightarrow{\alpha_q^y}$. The value reflects the degree of color intensity variation within a local region. Nevertheless the computed intensity variation value must be confined within a certain range. This is due to the fact that color distance beyond a certain range will become meaningless as discussed in section 7.2. As a result an upper limit T_3 is required in the definition of a sensible local contrast value.

$$d_{i,j}^{lt} = \min \left(\frac{\sum_{q=1}^{N_b} \text{dist}(\overrightarrow{d_{i,j}^y}, \overrightarrow{\alpha_q^y})}{N_b} \text{ where } \alpha_q \in \overrightarrow{d_{i,j}^b}, T_3 \right) \quad (41)$$

The simplest way to define the upper limit T_3 is to select a suitable threshold limit from Table 6 where the various categories of color threshold limits are identified. Although it is a convenient method, the method is too rigid where the selected value may not cater for the various input images with varying degree of color contrasts. The question of which category of the color threshold is considered as an appropriate upper limit also exists. In order to have a suitable upper bound value that can apply to all images, T_3 is determined globally according to the content of each input image. As shown below in equation 42, T_3 is defined as the overall average color variance among all pixels P for the entire image plus the standard deviation among all the variances. In order to have a good estimate for T_3 , flat regions with zero variance and regions whose variance are beyond the 190 threshold as stated in Table 6 are ignored.

$$T_3 = \frac{\sum_{p=0}^{\text{ImgSize}} \sum_m^{8cc} \left\{ \nu \mid \nu = \text{dist}(\overrightarrow{Y_p} - \overrightarrow{Y_m}) \text{ where } 190 \geq \nu > 0 \right\}}{\text{total number of } \nu} + \text{stdev}(\forall \nu) \quad (42)$$

Unlike in our previous model where the local threshold attribute is used directly in the selection of children, in this model it is used in the definition of color “closeness” between regions. Two regions are considered “close” in color if their color distance is below this local color threshold. Further elaboration in the use of this local threshold attribute will be presented in section 7.4.

7.3.3 Detailed Segmentation Stage

This stage will perform a detailed analysis of the resulting clustered regions from the pre-processing stage and continue to merge regions having a larger color variance. Region growing within an irregular pyramid structure is used as a means to perform clustering. In the current context we can view the pyramid building process as a way to perform image segmentation by growing seed regions. The selection of a survivor is equivalent to the selection of a seed region. The claiming of non-survivors by the survivor as its children is comparable to the growing of the seed. In contrast to a regular pyramid, this effect can be achieved through the use of irregular pyramid structure because of its flexibility in survivor selection and the ability to perform selective claiming of suitable neighboring non-survivors. Intended segmentation result can thus be obtained through a suitable definition of the seed selection criteria and the appropriate designation of growing or claiming rules. In our algorithm the selecting and the growing of seed regions will follow three basic eligibility criteria as defined in equation 43.

$$eligible(\overrightarrow{D_{i,j}}, X) = \begin{cases} true, & \text{if } X \in \overrightarrow{d_{i,j}^b} \wedge \\ & close(\overrightarrow{d_{i,j}^y}, X^y, d_{i,j}^h) \wedge \\ & d_{i,j}^a \geq X^a \\ false, & \text{Otherwise} \end{cases} \quad (43)$$

A region is eligible to participate in the selection and the growth processes if it satisfies all three eligibility criteria. The selection and the growth of a seed region will only occur among regions that exhibit spatial adjacency, color closeness and size inferiority. An arbitrary region X is eligible to participate in the process initiated by a pivot region $D_{i,j}$, if it is one of the pivot region's neighbors (i.e. $\overrightarrow{d_{i,j}^b}$). A neighboring region will only be

considered if the color variance between itself X^y and the pivot region $\overrightarrow{d_{i,j}^y}$ falls within a local threshold $d_{i,j}^{lt}$ where the superscript “y” denotes color. Finally only smaller or equal size neighbor is evaluated where the superscript “a” represents size.

Unlike the traditional pyramid structure where the base pyramid level is the original input image in pixel format, our proposed algorithm will begin the pyramid in a region format (i.e. group of pixels). After the initial pre-processing stage, regions with small color variance are formed. Connected component analysis is then used to identify individual regions from the respective color clusters. These extracted regions will form the base of the pyramid. This change in using the intermediate result avoids the processing of the first few pyramid levels which are the most time consuming. After the formation of a pyramid base and the determination of neighborhood relationship, the survivor/seed selection process will begin. The seed selection process is based on the surviving value as defined in the pyramid model section (i.e. 7.3.2). The surviving value of each region will reflect the total number of “close” neighbors that satisfy the 3 eligibility criteria. The process will then determine a local maximum having the largest number of neighborhood’s vote. Region that has the largest number of “suitable” neighbors is a good seed candidate that will enable maximum “healthy” growing of the seed region into its surrounding.

With the selected seed, the growing of region will begin. In a traditional irregular pyramid construction process, this is the child selection stage where each survivor will claim a set of suitable surrounding non-survivors as its children. In our context, we will treat this stage as the growing of the selected seed regions into the eligible surrounding neighbors. Instead of having the seed actively growing into its neighbors, our algorithm allows the neighbors to take an active role in searching for the most suitable seed. The first alternative

in the selection of children is used in this method where the non-survivor will act as the pivot point to locate the closest survivor. A non-surviving region X will become part of a seed S , if S is the nearest to the region X among all eligible seed regions α of X as shown in equation 44. In order not to confuse the α as the neighbors of D_{ij} , we use S to represent the surviving seed point that is nearest to X . Region X will evaluate all its surrounding seed regions α , which it is eligible to evaluate, and merge with one having the lowest color variance. This will maximize the color closeness among regions that are merged. Due to the use of the eligibility function, there is no requirement to set an upper limit for the minimal color distance.

$$nearest(S, X) = \left\{ S = \alpha \mid \begin{array}{l} \min(dist(X^y, \alpha^y)) \forall \alpha \mid eligible(\alpha, X) \\ \alpha \in neighbouring \text{ seeds / survivors of } X \end{array} \right\} \quad (44)$$

After this stage the entire process to construct another new pyramid level will repeat. A newly formed region $R_{i+1,k}$ on pyramid level $i+1$ will encapsulate a seed region $R_{i,j}$, and a group of non-surviving regions β_r on the lower level i . All the non-survivors β_r are guaranteed to be the nearest eligible neighboring regions.

$$R_{i+1,k} = \{R_{i,j} \cup \beta_r\} \text{ where } \left\{ \begin{array}{l} R_{i,j} \in L_i \cap L_{i+1} \\ \beta_r \in L_i \\ r = 1 \text{ to } |R_{i+1,k}| - 1 \\ eligible(R_{i,j}, \beta_r) \\ nearest(R_{i,j}, \beta_r) \end{array} \right. \quad (45)$$

The pyramid construction process will stop when the reduction rate is below 0.1. Reduction rate is defined as the decreasing rate in the number of newly created regions on the next higher pyramid level. Through experiment we observe that 0.1 is the break-even point where any continuation in the growing process beyond this stage will not yield any noticeable improvement in the segmentation result. The outputs from the final pyramid level are the various connected components on the respective color layers. The next stage is the text extraction process where the connected components on each respective color layers are extracted and their textual status is verified. The verification is done through a simple component's size, width and height consistency check to determine its textual identity.

7.4 Threshold Derivation

This section will describe the motivation behind the derivation of the local threshold value $d_{i,j}^{lt}$ which is used in the eligibility function to determine the color closeness of two regions. Two regions are “close”, if their color differences fall below this threshold. It is different from T_l used in the pre-processing stage (i.e. section 7.3.1) which is an empirical value. This threshold is dynamically and locally determined by taking the average of all color variances between a pivot region and all its neighbors. This has enabled a good adaptation to the varying color contrast conditions across image regions. The design of the formula is based on the assumption that regions belonging to the same image object will have a lower color contrast than those in different image objects. During the initial growing stage of a region, the majority of the neighboring regions may belong to the same object. The threshold will provide an unrestrictive and yet steady growth of the regions. As the growing reaches its maturity (i.e. reaching the boundary) the majority of the regions with homogenous background will stop at this threshold value (i.e. average variance). For the

remaining regions with a complex background, the upper color limit of T_3 is used to avoid excessive over segmentation of region.

7.5 Experimental Results

Evaluation of color segmentation is not an easy task. There exists no general methodology to evaluate the correctness of a color segmentation result. The validity of the resulting colors and the segmented regions will vary according to the human perception and the original intent. As a result the measurement is more qualitative rather than quantitative. The existence of the wide variety of color space representations and the utilized segmentation techniques also add on to the evaluation complexity. This has also led to the difficulty in comparative study among algorithms. Since the ultimate aim of our proposed algorithm is to achieve color document image segmentation in preserving both the major graphical and textual components with the later as the focus, we will evaluate the result based on 2 factors. The most important factor is to evaluate how well the algorithm can ensure spatial compactness in the segmented textual region. The second factor will assess the effectiveness in the retention of the original image content (i.e. non-text) after segmentation. The evaluation method for the first factor is by counting the number of correctly extracted textual components from the respective color layers that is visually recognizable. The second factor is measured by visually counting the number of retained major features in the image. For doing a comparative study we will use the “pointer” method [217] as described in section 7.1.

Figure 71 shows the test sample of a logo extracted on the web. After color segmentation and text extraction, all recognizable text regions are detected with minor over segmentation for the character “A” (i.e. part c and part f). Figure 71b shows the segmentation result of the

“pointer” method where only the centre bigger texts (i.e. a1tp) are extracted as shown in Figure 71e. All smaller texts along the circular path are classified as noise. As shown in Figure 71d, the “pointer” method has failed to group pixels within the character as a single color cluster. As a result, pixels belonging to the same character are fragmented into multiple color layers. Without further connected component analysis, each fragment of the character will be too small in size to be classified as text. The lack in the color value continuity for pointer chasing may be the main cause for the failure. Figure 72 is a test sample of an advertisement on the web for onions. As shown in Figure 72f all text regions are extracted correctly by our method. In contrast, the “pointer” method produces results as illustrated in Figure 72b and Figure 72d showing components on 2 different color layers holding the main bulk of the textual contents. In order to better demonstrate the degree of fragmentation, these images are used instead of the image after text extraction where most of the components are removed as graphic. As seen in these 2 images, the textual regions are split into 2 main color clusters. Although visually the representative colors for the 2 layers are close, both co-exist as the color peaks in their own local region in the feature space and thus they are clustered as 2 separate color layers. Figure 73 demonstrates the same effect for another test sample where the word “On” and the character “n” in “volcanoes” are segmented into multiple color layers as shown in Figure 73c, Figure 73d and Figure 73e. Our method has proven again to be effective with the correct and full extraction as shown in Figure 73g. Figure 74a shows a test sample of a poster with the majority of textual areas having been segmented properly by both methods, except for the character “e” in the word “Views” in Figure 74d (i.e. “pointer” method). Due to the close proximity in terms of color for both text and background, the character is completely integrated into the background object. On the contrary, our method identifies the character properly with no problem (i.e. Figure 74g). The middle test sample: weather service logo in Figure 74b demonstrates the ability of our algorithm in preserving major regional contents (i.e. woman and tower) as

shown in Figure 74h whereas the “pointer” method has resulted in both regions almost being absorbed into the background area (i.e. Figure 74e). This may be due to the problem of using histogram where the color pixel count for both regions is small. The possibility of such a weak color becoming a local maximum is very low. As such it is absorbed by a neighboring stronger color in terms of the area coverage. The last sample in Figure 74c shows an image having colors that are perceptually very close. This has stretched the “pointer” method to its extreme where the result is a single cluster of color. In contrast, our method performs satisfactorily.

The results after evaluating 38 images according to the 2 factors are shown in Table 7 which confirms that our algorithm achieved the intended task. For textual components, our method achieved a 84% identification rate (i.e. 128/152) as compared to the 70% attained by the “pointer” method. The majority of the results are satisfactory except for images with very high color variance among the textual fragments where the aim of compactness cannot be obtained.

Table 7. Evaluation result

Factors	Original	Pyramid	Pointer
Text	152	128 (84%)	107 (70%)
Non-text	93	86 (92%)	72 (77%)

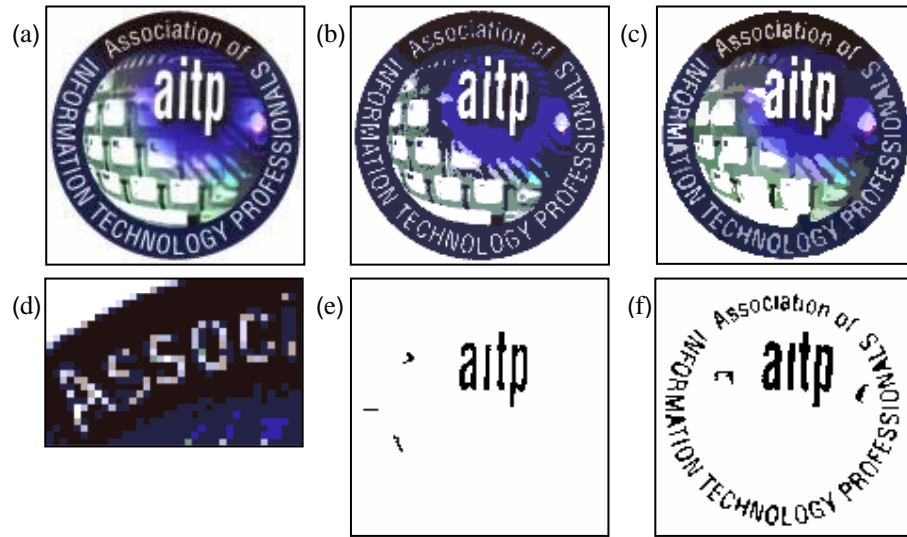


Figure 71. Test sample : logo

(a) original image, (b) pointer segmentation, (c) pyramid segmentation,
(d) zoom-in of image b, (e) pointer text extraction, (f) pyramid text extraction



Figure 72. Test sample : onions advertisement

(a) original image, (b) pointer segmentation color layer 1,
(c) pointer segmentation, (d) pointer segmentation color layer 2,
(e) pyramid segmentation, (f) pyramid text extraction

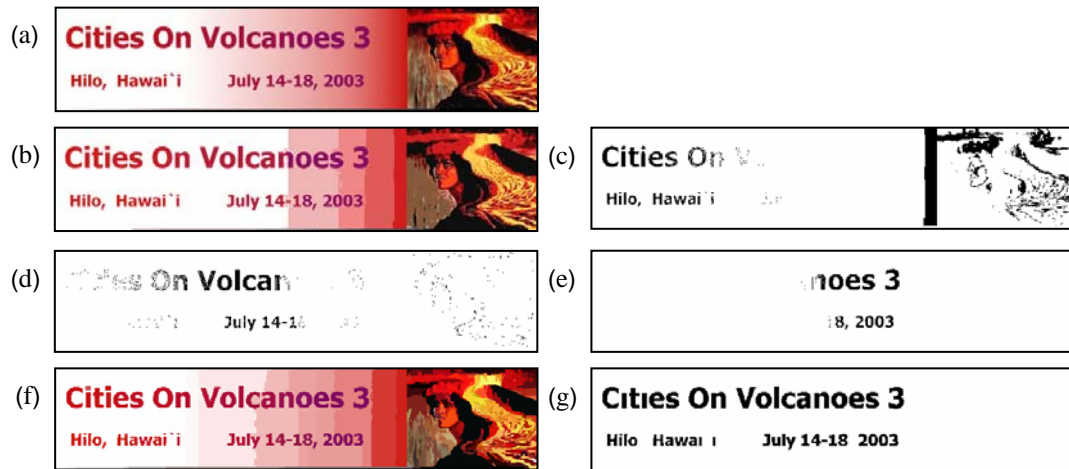


Figure 73. Test sample : volcanoes advertisement
 (a) original image, (b) pointer segmentation,
 (c),(d),(e) pointer segmentation color layers,
 (f) pyramid segmentation, (g) pyramid text extraction

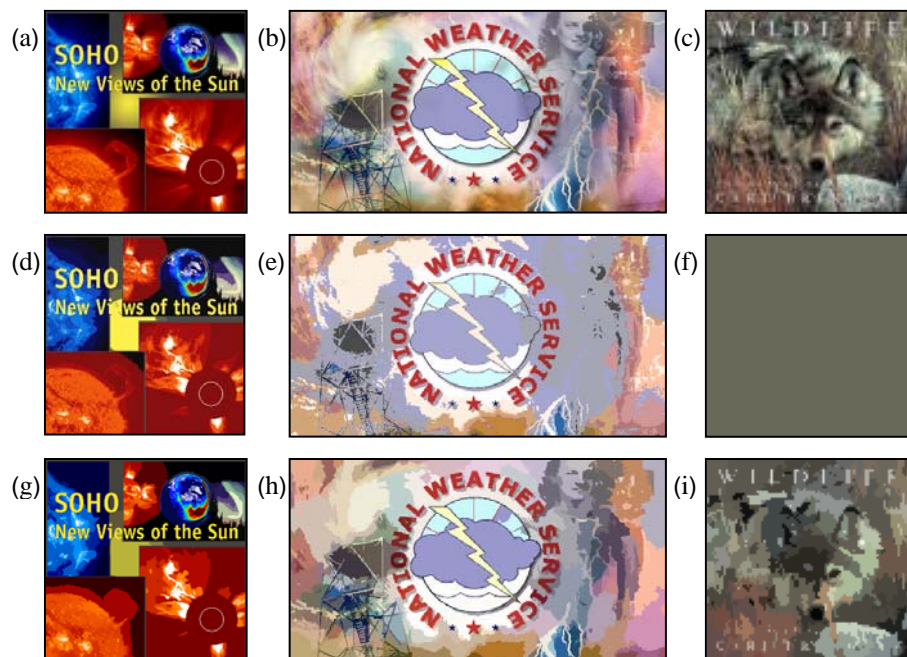


Figure 74. Three test samples: Poster, Weather service logo, Wildlife magazine
 (a) (b) (c) original image,
 (d) (e) (f) pointer segmentation,
 (g) (h) (i) pyramid segmentation

Although the main subject of the proposed method is color document images, it can easily be modified to also process gray scale images. The only change to the algorithm is to replace the color distance measurement to a gray scale intensity distance measurement which is simply the absolute gray scale level difference between two regions. Due to the ability of the algorithm to produce result in the form of multiple gray scale layers with each layer holding their own set of connected components, the problem in binarizing or segmenting gray scale images with the existence of reverse text is solved. Our method was recently used for text extraction from name card images. It has been found that the name card designs are getting more and more fanciful with complex background color and reverse contrast text. A commercially available name card scanner [223] for name card text extraction and database construction has encountered problems due to such fanciful design. Our method was used for testing and has been found to produce promising results. The following are some examples. Figure 75 shows an image containing texts in different gray scale intensity including some reverse contrast texts on a background with varying gray scale level. Figure 76 presents the final segmentation result. On the left is the original segmentation result by merging all gray scale layers into a single image. On the right in Figure 76 shows the result after some noise removal where those components with extreme sizes (i.e. very large and very small components) are removed. Figure 77 shows the subject components on the various gray scale layers after the pre-processing and the detailed segmentation process. Figure 78 shows another test sample containing text in reverse contrast. As demonstrated, the texts are identified correctly by the algorithm. An interesting test sample in Figure 79 where there exist some background pattern. Instead of badly affected by these patterns, the algorithm manages to segment most of the textual regions. When comparing the result of the final test sample in Figure 80 to the result produced by the Otsu's thresholding method in Figure 81, the uses of our adaptive growing method in producing result in multiple layers is also proven to be a better thresholding method.

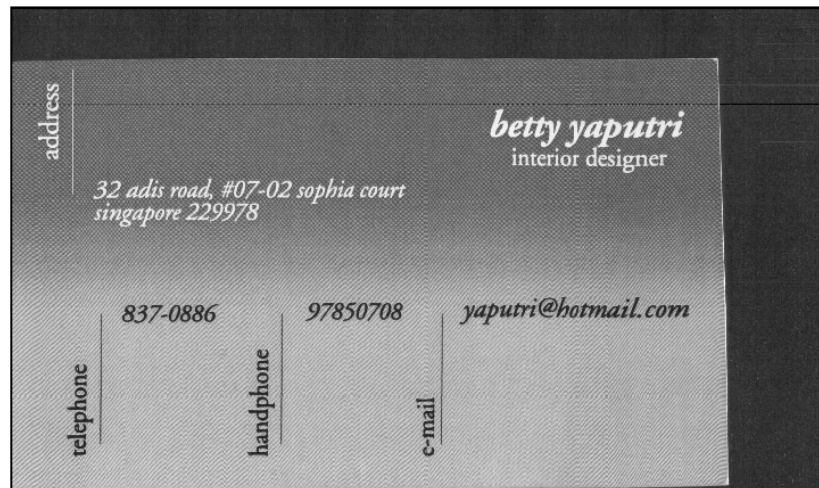


Figure 75. Gray scale test sample 1: original image



Figure 76. Segmentation result for Figure 75 (left: original, right: after some noise filtering)

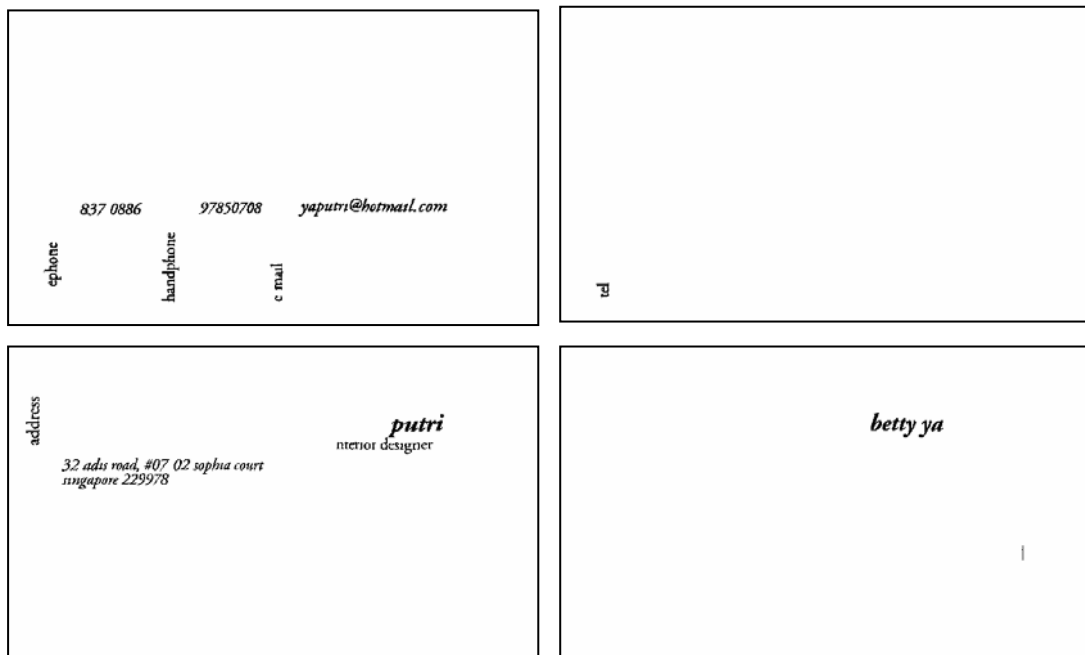


Figure 77. The subject connected components on the various gray scale layers for Figure 75



Figure 78. Gray scale test sample 2 (top: original, bottom: result)



Figure 79. Gray scale test sample 3 (top: original, bottom: result)



Figure 80. Gray scale test sample 4 (top: original, bottom: result)

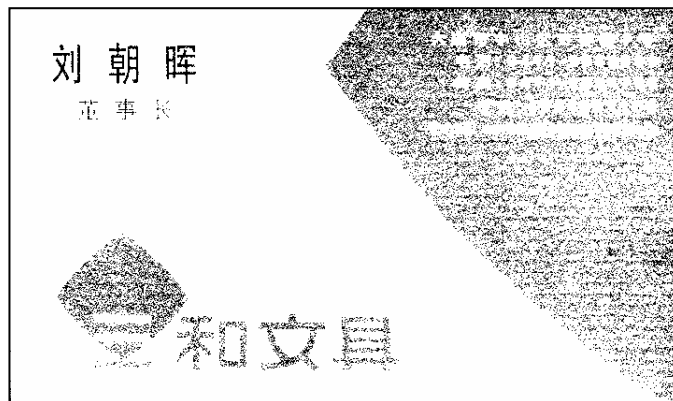


Figure 81. Binarization result produces by the Otsu's method

7.6 Summary and Discussion

We have proposed a novel color segmentation technique to be used for text extraction. The segmentation is done in the RGB color space. A simple color distance measurement (i.e. Manhattan) and category of color thresholds are derived. The proposed algorithm is divided into two stages. The initial pre-processing stage utilizes a dynamic seed selection and clustering process with very close color range. The clustered regions are then used as an input to the detailed segmentation process (i.e. 2nd stage) where both color and spatial factors are considered. The segmentation is based on a region growing technique. Irregular pyramid is used for the segmentation. It differs from other region-growing methods in its selection of seeds and the way regions are grown. The algorithm is evaluated according to 2 factors and compared with the “pointer” method [217]. The results have demonstrated the ability of our algorithm in achieving segmentation for text extraction.

In addition we have also demonstrated the simplicity in switching the algorithm to process gray scale images. Our method has also been found useful for name card text extraction which can solve the problem of complex background color and reverse contrast text.

Chapter 8

The Storage Requirement and the Processing Speed Analysis

While our irregular pyramid has been shown to provide a novel solution to text segmentation problem, one concern of the choice of the pyramid structure is the computational cost involved. This chapter will analyze the storage requirement of our irregular pyramid operation with the view in estimating the system's complexity. Experiments are carried out to confirm our storage requirement analysis. Processing time will also be measured in the same experiments to assess the system's computational cost.

8.1 Storage Requirement Analysis

We will first examine the regular pyramid's storage requirement and then adaptive irregular pyramid before we move on to our irregular pyramid structure.

8.1.1 Regular Pyramid Model

In a regular pyramid structure, the number of data points on each pyramid level is based on a fixed dimensional reduction ratio R as described in section 2.4.1. A reduction ratio of two will translate into a reduction in image size by R^2 . This will result in a pyramid size of 64, 16, 4 and 1 from the lower pyramid level to the final pyramid apex as shown in Table 2. As a result the number of data points on each pyramid level is constant with respect to the reduction ratio. For a reduction ratio of 2 the total number of pyramid data points, including the base level, is approximately 1.33 times that of the input image size as shown in equation 46. The computation is shown as follow where the first term is the number of data points on the base pyramid level, follow by the number of data points on each respective subsequent

level and finally the pyramid apex which has a single data point. For a reduction ratio of 3 the total number is 1.17 times of the original image.

$$\text{Total number of data points} = \frac{N_0}{4^0} + \frac{N_0}{4^1} + \frac{N_0}{4^2} + \frac{N_0}{4^3} + \frac{N_0}{4^4} + \dots + 1 \quad (46)$$

where N_0 = number of data points on pyramid level 0

8.1.2 Adaptive Irregular Pyramid Model

As compared to the regular pyramid, the irregular pyramid will only have a constant number of data point on the base and the first pyramid level. The number of data points on the base pyramid level is the total number of pixels in the input image (i.e. image size). Data points on the first pyramid level will depend on the survivor selection process. The selection is based on the identification of a maximal independent set satisfying two selection rules (i.e. “No two neighbors will survive together” and “Non-survivor will have at least one neighboring survivor”). If we assume a strict compliance of these two rules in the selection of the survivors then we are able to compute an estimated number of survivors on the first pyramid level. Figure 82 shows three possible survivor selection scenarios in a 5x5 image. The character “S” in bold represents the selected survivors and the character “N” represent the non-survivors. The left image shows the worst case situation where there are a total of nine selected survivors to form the next higher pyramid level. The middle image shows the average situation with six survivors. The right image demonstrates the best case scenario with only four survivors. From these three images we can see that the selection process attains its maximal decimation when a single survivor is surrounded by eight non-survivors. On the other hand the worst selection is when there are only three surrounding non-survivors.

S	N	S	N	S
N	N	N	N	N
S	N	S	N	S
N	N	N	N	N
S	N	S	N	S

N	S	N	S	N
N	N	N	N	N
N	S	N	S	N
N	N	N	N	N
N	S	N	S	N

N	N	N	N	N
N	S	N	S	N
N	N	N	N	N
N	S	N	S	N
N	N	N	N	N

Figure 82. Three possible survivor selection scenarios (left: worst, middle: average, right: best)

By a quick estimation, we find that the total number of pyramid data points will fall within the range defined by the best and the worst case as follows.

$$N_I = \begin{cases} N_0/9 & \text{for the best case} \\ N_0/4 & \text{for the worst case} \end{cases} \quad (47)$$

We carried out an experiment on the adaptive irregular pyramid model, which always fulfills both the survivor selection rules at all time with the total intensity variance among neighboring regions as the surviving value. Table 8 shows the summary collected from a total of 45 images of various types (i.e. all texts, text and graphic and all graphics) and sizes. The second column shows the average size reduction ratios among all input images on the various pyramid levels. The third column shows the standard deviation among the average and the last two columns reflect the minimal and the maximal size reduction ratios among the 45 input images.

Table 8. Results obtained by using various input images

Levels	Average	Standard deviation	Minimal	Maximal
0	5.0	0.32	4.4	5.3
1	4.4	0.20	4.1	4.7
2	4.3	0.14	4.1	4.5
3	4.2	0.15	3.8	4.4
4	3.9	0.28	3.4	4.5
5	3.5	0.43	2.9	4.3
6	2.6	0.47	1.9	3.3
7	4.9	4.25	1.9	16.0

The reduction ratio on the first pyramid level confirms our above analysis which stays in between the worst case of 4 and the best case of 9. Based on this result we can make two observations. The first observation is that the size reduction rate for all images falls within a very narrow range of standard deviation values. Despite the varying image types and sizes, the reduction rate on each pyramid level is fairly similar among all the images. The second observation is a gradual slowing down of the reduction rate as in Figure 83 except for a sudden jump of the reduction ratio in the last pyramid level indicating the merger of all remaining data points.

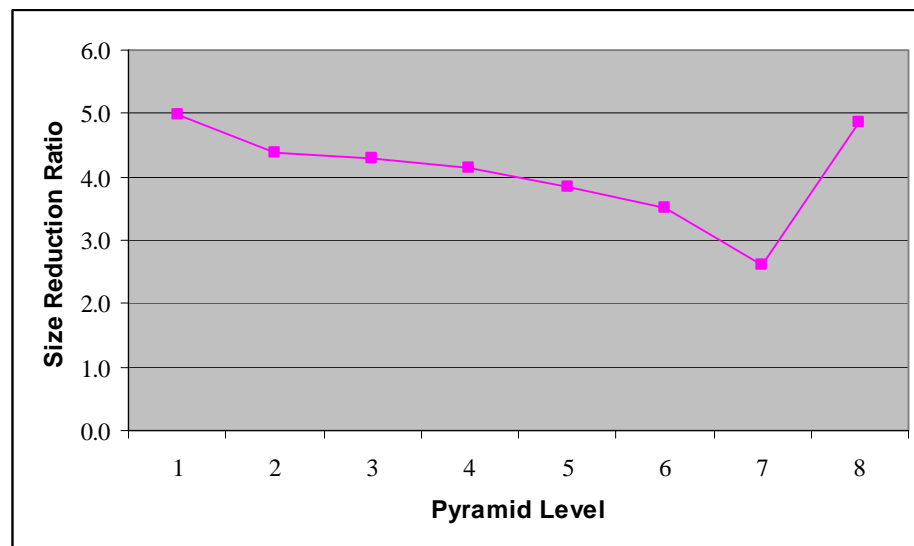


Figure 83. Size reduction ratio graph for different images

By using the experimental data in Table 8 we will attempt to make an estimate of the total pyramid data points in this model by summing the reciprocal of all average reduction ratios as shown in equation 48. The total summation (i.e. including the 1) will result in a factor which reflects the size of the data points with respect to the size of the original image on all pyramid levels. For the first level we will use the worst case situation (i.e. 4). All remaining reduction ratios will be based on the average values in the experiment. Since there is a very narrow deviation in most of these average values (i.e. level 1 to 6), it should

be a very close representation of most cases. Due to the fact that the possible number of data points on the last pyramid levels is very low, the inclusion of their average reduction ratios in the computation will not affect much of the estimation. The estimated factor is 1.32 which is very close to the regular pyramid model.

$$\text{Total number of data points} = \left(1 + \sum_{i=1}^{\text{Levels}} \left(\prod_{j=1}^i (1/r_j) \right) \right) \cdot N_0 \quad (48)$$

where r_j = the reduction ratio at level j

8.1.3 Our Irregular Pyramid Model

Our irregular pyramid is quite different from the adaptive model without going into the details of the various difference, we will highlight three main differences from the adaptive version. The first difference is its departure from the standard survivor selection rule. There are two main categories of survivors. The first category is the legitimate survivors that follow both the selection rules. The second category is the non-legitimate survivors that violate the first selection rule (i.e. “No two neighbors will survivor together”). Due to the permission to allow the co-existence of the survivors in the neighborhood, this category of survivors will contribute to the increase in the total number of surviving data points. Under this category one group of the survivors are the root nodes and the other group of survivors are identified by the algorithm as the confirmed non-target regions. The second difference is the starting point of the irregular pyramid from a higher level rather than from the base. The third difference is the premature termination of our irregular pyramid before reaching the apex. From the above observation, the first difference could mean more processing in our irregular pyramid than the adaptive version due to the duplicated processing of survivors and neighbors. This however may be compensated by the second and third differences as

they will result in computational saving from unnecessary processing of the lower and the upper levels of the pyramid. Because of the high variability in the reduction ratio in the reduction ratio in our model, we carried out a similar experiment on our model as before.

Table 9. An example of a test image using our model

Levels	Data points	Survivors	Size reduction ratios	Non-legitimate survivors	Percentage of non-legitimate survivors
0	28800	3148	9.15	-	
1	3148	1085	2.90	663	61%
2	1085	697	1.60	571	82%
3	697	565	1.20	528	93%
4	565	482	1.20	461	96%
5	482	334	1.40	326	98%
6	334	126	2.70	123	98%
7	126	98	1.30	95	97%
8	98	98	1.00	98	100%

Table 10. Results obtained by using various input images

Levels	Average	Standard deviation	Minimal	Maximal
0	11.6	6.89	3.1	29.7
1	2.5	0.73	1.3	3.6
2	1.5	0.25	1.1	1.9
3	1.3	0.20	1.0	1.6
4	1.2	0.13	1.0	1.4
5	1.2	0.14	1.0	1.4

Table 9 shows the experimental result of a single test sample. On the first pyramid level, a reduction ratio as high as 9.15 was achieved by the clustering process. This ratio varies very greatly among different images even with the same clustering criteria as seen in Table 10. As we move up the pyramid levels the reduction ratio decreases. In comparison with the previous two models, it has a lower reduction ratio in the subsequent higher pyramid levels. This is mainly due to those non-legitimate survivors shown in the second last column of Table 9 due to the departure from the standard selection rule as just explained above. The percentage of those non-legitimate survivors is fairly high and it increases towards the final

pyramid level. If the method of storing these survivors outside of the pyramid structure is employed, a better reduction ratio can be obtained. As reflected in Table 10, among the tested input images, larger deviation in the reduction ratio occurs on the second pyramid level while the remaining levels maintain at a moderate deviation range. The majority of the images reach the final formation stage at pyramid level 5 with a common stopping condition. Some images even stop before the 5th pyramid level as seen under the “minimal” column in Table 10 where a reduction ratio of 1 is reported with no further contraction in the pyramid data point. In view of the greater variable condition in our model we will compute the number of estimated data point in terms of three values (i.e. best, average and worst). Since the reduction ratio on the first pyramid level has the largest deviation, we will make use of the maximal, average and the minimal reduction ratios for the best, average and worst scenarios estimation respectively. The remaining levels are based on the average value. The comparison data are shown in Table 11.

Table 11. Summary of the estimated number of pyramid data points

Model	Number of times of the input image size
Regular	1.33(R=2), 1.17(R=3)
Adaptive	1.32
Our model	1.08(best), 1.20(average), 1.76(worst)

8.2 A Rough Estimation of Complexity

The above storage requirement analysis gives us some idea to do a rough estimation of the complexity of our model, despite the high variability in the neighbor/survivor selection processes. While the analysis shows an increase of data points ranging from 8% to 76% in our model, the increase in the computational cost is not a mere correspondence to this factor. In fact, between two consecutive levels the main computational cost arises from the selection operation between the survivors and children. If we assume a constant reduction

ratio r and an image size N and let $\rho = 1/r$, then the rough computational cost of the operation between level 0 and level 1 will be $O(N \times \rho N) = O(\rho N^2)$. However, this computational cost decreases rapidly with the pyramid building towards the higher level. Thus the aggregate cost from the progressive pair-wise operation between successive levels will be as shown in equation 49. Thus the irregular pyramid operation cost basically has a polynomial growth, which is comparable to the traditional methods that require connected computation analysis and spatial analysis among components [116, 139].

$$\begin{aligned}
& O(\rho N^2 + \rho N \times \rho N \times \rho + \rho^2 N \times \rho^2 N \times \rho + \rho^3 N \times \rho^3 N \times \rho + \dots) \\
& = O(\rho N^2 (1 + \rho^2 + \rho^4 + \rho^6 + \dots)) \\
& = O\left(\frac{\rho N^2}{1 - \rho^2}\right) = O(N^2) \quad \text{where } \rho < 1
\end{aligned} \tag{49}$$

8.3 Processing Speed Analysis

The previous section was merely a rough and simplified estimation with an assumption of a constant reduction ratio. In reality, the actual computational cost will largely depend on the content of the image, which ultimately determines the various complexes processing between successive pyramid levels. As such, in this section, we will use some experimental data to illustrate the processing speed in using our irregular pyramid model in comparison with a conventional method. In this experiment, we will use the model for color document images described in Chapter 7. The main objective in this section is to present a general idea of the kind of processing speed in using our irregular pyramid model. Table 12 shows the image size and the processing speeds for the “pointer” method and our pyramid model as described in Chapter 7 for the various color images. In general the pyramid model has a higher processing speed due to the finer analysis of the image content as compared to the

“pointer” method. The speeds are recorded in spite of the fragmentation problem in the segmented textual content for the “pointer” method. Figure 84 shows the graph by plotting the image sizes against the processing speeds in both methods. For smaller image size, the processing speed is relatively similar in both methods. There are even cases where our model is faster than the “pointer” method. For a larger image size the pyramid model will have a higher processing speed. Nevertheless it is still within a tolerable limit. As observed in the last data point in Figure 84, the processing speed is not directly proportional to the image size. There is situation where the processing speed can be even lower than the smaller image size if majority of the image regions have similar colors.

Table 12. Processing speeds for the various images (Pentium IV – 1.8GHz)

Test Sample	Size (pixels)	Pointer method (sec)	Pyramid method (sec)
“Wildlife” Figure 74c	7,480	0.40	0.38
“Infosurf”	9,072	0.71	0.76
“aitp” Figure 71	18,496	1.18	1.67
Liverpool	31,185	1.84	2.57
“Planet”	41,160	2.30	0.97
“sweet” Figure 72	67,584	5.23	7.21
“Cities” Figure 73	76,500	6.23	12.16
“Soho” Figure 74a	82,944	5.30	17.16
“Newsfront” Figure 74b	133,500	13.63	23.96
Texture	170,340	12.70	20.74

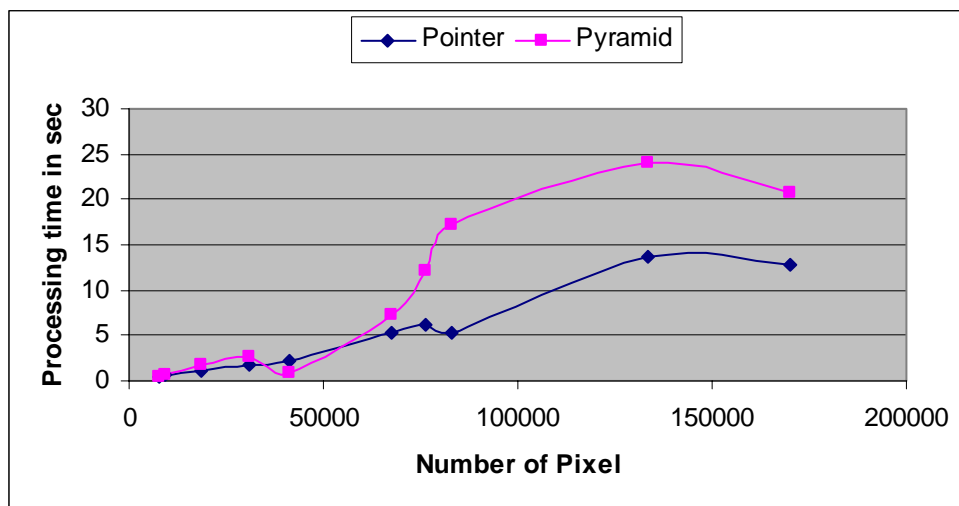


Figure 84. Processing speeds for the various images arranged according to image sizes

Chapter 9

Conclusions and Future Directions

In this thesis we have addressed several issues of text segmentation in document image processing. Most document image analysis systems assume Manhattan layout of text. To date, there are not many satisfactory solutions to deal with documents containing sparse text in variable sizes and irregular alignments such as in pamphlets and advertisements. The adaptive binarization of gray scale document images also faced the problem in the need to pre-determine a fixed local window size. Color documents involving text on complex background also present another problem. In this we have proposed the use of irregular pyramid to address these problems. After the introductory chapter and two survey chapters on regular and irregular pyramids, we present our irregular pyramid solutions in chapters 4 to 7. In Chapter 4 we propose the use of our pyramid model to provide a natural aggregation of word components of any sizes, fonts and orientations to solve the problem faced by most of the traditional methods. These methods generally assume Manhattan document layout and require complicated inter-textual component distance analysis. In Chapter 5 we extend our method in the segmentation of logical text groups with varying words' orientation. This has provided solution to the detection of non-uniform logical grouping of text in contrast to the usual rectangular block layout segmentation approach in most traditional methods. In the processing of gray scale document images we have suggested the deferment of the binarization process after the segmentation of a rough textual region as described in Chapter 6. This has not only dispensed with the need to pre-determine a fixed local window size as in most adaptive thresholding methods, it also permit a more focused thresholding process on the targeted textual region to achieve a better binarization process. Finally in Chapter 7

our proposed use of a concurrent region growing method within the pyramid structure enables the segmentation of color images in ensuring the extraction of a compact textual region which most other methods cannot achieve. We also demonstrated the ease in the alteration of our algorithm to solve the reverse contrast text problem faces in many gray scale document image processing methods.

In Chapter 8 we present the storage requirement in using our irregular pyramid model and a brief estimation of its complexity with some measurements of its processing speed. As illustrated in the chapter, although the storage requirement is slightly higher than the regular pyramid model in the worst case scenario, depending on the design of the selection criteria and the nature of the input images it is of comparable size in the average case. In the processing speed, our method has about the same efficiency as the traditional method. For the larger image size, our method will take moderately longer time. In spite of this increase in the processing time, it is still within a tolerable limit. This slight increase in the storage requirement and the processing efficiency, however, is compensated by the novel solution offered by our method. In fact it is well known that pyramid structure is amenable to parallel processing [9, 10, 16]. With advances in computer technology such as the recent PC clusters, our irregular pyramid structure can be implemented in a parallel computing platform. The computational cost will thus not be an issue.

The fascinating aspect of an irregular pyramid structure is its close resemblance to the natural evolutionary theory. A single pixel resides within an input image surrounded by some neighboring pixels where each has its own unique property. Due to the “closeness” of certain properties some are pulled together to form a region. These newly formed regions inherit new property by summarizing or through some form of agreement among all parties within the regions. Again each region will have a new group of neighboring regions and

through the interaction among neighboring regions with the same or a different type of “closeness” criteria they are merged again to form a larger region. This will continue and evolve until the final formation of the targeted region. This flexibility in the pyramid structure to manipulate the image information that allows an asynchronous and autonomous processing of individual processes within a hierarchical structure is not achievable in many other methods. The structure has provided a very flexible processing environment and yet bounding the information within a constant structure. The thesis has demonstrated this ability of the pyramid model through the various proposed methods in solving difficult document image processing problems.

Although our methods have been shown to be able to solve many of the problems that the traditional techniques cannot achieve, just like any other methods our methods also have some limitations. Despite the ability to avoid the pre-determination of fixed distance threshold, the correct segmentation of word regions must still rely in the assumption of larger inter-words spacing than inter-characters spacing within the same word. Although this is a common and reasonable assumption, even human reader requires this setting to identify different words. Word regions will not be correctly segmented if the inter-word distance is the same or smaller than the inter-character distance. Another limitation is in the processing of joined text and graphical components. Due to the bottom-up approach we have employed in the aggregation of pixels into text, the growing of the text regions may continue to expand into the area of the graphical component. This will happen if both components have interconnected foreground pixels in the case of binary image and very close intensities in the case of gray scale or color images. In this thesis we have only focused on the segmentation and the extraction of the textual content. The task to filter graphical objects is not the focus of the present work. In all our methods the filtering of graphical objects is achieved by a simple area filtering method where a component size threshold is picked to discard big

graphical objects which is often a minority in number as compared to the majority text components. Due to this assumption, very large text size which belongs to the minority group within the document may also be discarded as graphical object (e.g. large newspaper heading). In view of this, further work can be done in future in the identification of text and non-text objects. Instead of using the current simple area filtering method, graphical components may also be identified in the irregular pyramid structure on an appropriate pyramid level and processed accordingly. Another area that can be done in future is in the realignment of texts into a horizontal direction to allow for recognition. The information kept in the pyramid for various components can be used for future processing, such as the correction and the realignment of skewed or curved text line.

Bibliography

Regular Pyramid model

1. P. J. Burt, T.H. Hong and A. Rosenfeld, "Segmentation and estimation of image region properties through cooperative hierarchical computation", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-11, No. 12, Dec 1981, pp. 802-809.
2. T.H. Hong, K.A. Narayanan, S. Peleg and A. Rosenfeld, "Image smoothing and segmentation by multiresolution pixel linking: further experiments and extensions", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-12, No. 5, Sep/Oct 1982, pp. 611-622.
3. T.H. Hong, M. Shneier and A. Rosenfeld, "Border extraction using linked edge pyramids", IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-12, No. 5, Sep/Oct 1982, pp. 660-668.
4. H.J. Antonisse, "Image segmentation in pyramids", Computer Graphics and Image Processing 19, 1982, pp. 367-383.
5. T.H. Hong and A. Rosenfeld, "Compact region extraction using weighted pixel linking in a pyramid", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, No. 2, Mar 1984, pp. 222-229.
6. T.H. Hong and M. Shneier, "Extracting compact objects using linked pyramids", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, No. 2, Mar 1984, pp. 229-237.
7. E.H. Adelson, C.H. Anderson, J.R. Bergen, P.J. Burt and J.M. Ogden "Pyramid methods in image processing", RCA Engineer, 29-6, Nov/Dec 1984, pp. 33-41.
8. J.M. Ogden, E.H. Adelson, J.R. Bergen and P.J. Burt, "Pyramid-based computer graphics", RCA Engineer, 30-5, Sep/Oct 1985, pp. 4-15.
9. W.I. Grosky and R. Jain, "A pyramid-based approach to segmentation applied to region matching", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 5, Sep 1986, pp. 639-650.
10. C.L. Tan and W.N. Martin, "An analysis of a distributed multiresolution vision system", pattern Recognition, Vol. 22, No. 3, 1989, pp. 257-265.
11. A. Rosenfeld, "Pyramid algorithms for finding global structures in images", Information Sciences 50, 1990, pp. 23-34.
12. J.M. Jolion, P. Meer and A. Rosenfeld, "Border delineation in image pyramids by concurrent tree growing", Pattern Recognition Letters 11, 1990, pp. 107-115.
13. S. Baronti, A. Casini and F. Lotti, "Variable pyramid structures for image segmentation", Computer Vision, Graphics and Image Processing 49, 1990, pp. 346-356.
14. M. Bister, J. Cornelis and A. Rosenfeld, "A critical view of pyramid segmentation algorithms", Pattern Recognition Letters 11, 1990, pp. 605-617.
15. C.A. Sher and A. Rosenfeld, "Pyramid cluster detection and delineation by consensus", pattern Recognition Letters 12, 1991, pp. 477-482.
16. G. Bongiovanni, L. Cinque, S. Leviald and A. Rosenfeld, "Image segmentation by a multiresolution approach", Pattern Recognition, Vol. 26, No. 12, 1993, pp. 1845-1854.
17. M.G. Kim, I. Dinstein and L. Shaw, "A prototype filter design approach to pyramid generation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 12, Dec 1993, pp. 1233-1240.
18. P.K. Biswas, J. Mukherjee and B.N. Chatterji, "Component labeling pyramid architecture", Pattern Recognition, Vol. 26, No. 7, 1993, pp. 1099-1115.
19. C.L. Tan and S.K.K. Loh, "Efficient edge detection using hierarchical structures", Pattern Recognition, Vol. 26, No. 1, 1993, pp. 127-135.
20. S.W.C. Lam and Horace H.S. Ip, "Structure texture segmentation using irregular pyramid", Pattern Recognition Letters 15, 1994, pp. 691-698.
21. C.L. Tan, C.M. Pang and W.N. Martin, "Transputer Implementation of a Multiple Agent Model for Object Tracking", Pattern Recognition Letters, V. 16, pp. 1197-1203, 1995.

22. D. Prewer, "Connectionist pyramid powered perceptual organization: visual grouping with hierarchical structures of neural networks", Honours report, The University of Melbourne, Nov 1995.
23. L. Cinque, S. Leviald and A. Rosenfeld, "Fast pyramidal algorithms for image thresholding", Pattern Recognition, Vol. 28, No. 6, 1995, pp. 901-906.
24. P.F.M. Nacken, "Image segmentation by connectivity preserving relinking in hierarchical graph structures", Pattern Recognition, Vol. 28, Vol. 6, 1995, pp. 907-920.
25. Borowy, M., Jolion, J.M., "A pyramidal framework for fast feature detection", Proc. 4th Int. Workshop on Parallel Image Analysis, 1995, pp. 193-202
26. L.Cinque, S.Levialdi and A.Rosenfeld, "Fast Pyramidal Algorithms for Image Thresholding", Pattern Recognition, V. 28, No. 6, pp. 901-906, 1995.
27. Hui Cheng, Charles A. Bouman, and Jan P. Allebach, "Multiscale Document Segmentation," IS&T 50th Annual Conference, Cambridge, MA, 18th-23rd May 1997, pp. 417-425.
28. C.H. Lee and L.H. Chen, "A fast motion estimation algorithm based on the block sum pyramid", IEEE Transactions on Image Processing, Vol. 6, No. 11, Nov 1997, pp. 1587-1591.
29. A.S. Wright and S.T. Acton, "Watershed pyramids for edge detection", In Proceedings of the 1997 International Conference on Image Processing, 1997.
30. P.S. Wu and M. Li, "Pyramid edge detection based on stack filter", Pattern Recognition Letters 18, 1997, pp. 239-248.
31. A.S.Wright and S.T.Acton, "Watershed Pyramids for Edge Detection", Proceedings of the 1997 International Conference on Image Processing (ICIP'97), 1997.
32. V.Cantoni, L.Lombardi, G. Manzini and L.Cinque, "Page Segmentation using a Pyramidal Architecture", Proceedings of the 1997 Computer Architectures for Machine Perception (CAMP'97), pp. 195-199, Oct 1997.
33. M.Li and P.S.Wu, "Pyramid Edge Detection for Color Images", Optical Engineering, V. 36, No. 5, May 1997.
34. C.H.Lee and L.H.Chen, "A Fast Motion Estimation Algorithm Based on the Block Sum Pyramid", IEEE Transactions on Image Processing, V. 6, No. 11, Nov 1997.
35. C.L.Tan and P.O.Ng, "Text extraction using pyramid", Pattern Recognition, Vol. 31, No. 1, 63-72 (1998).
36. A. Rosenfeld and C.Y. Sher, "Detecting image primitives using feature pyramids", Journal of Information Sciences 107, 1998, pp. 127-147.
37. F. Ziliani, B. Jensen, "Unsupervised segmentation using modified pyramidal linking approach", Proceedings of the 5th IEEE International Conference on Image Processing (ICIP'98), Vol. 3, Chicago, USA, 4th-7th Oct 1998, pp. 303-307.
38. P. Bertolino, S. Ribas, "Image sequence segmentation by a single evolutionary graph pyramid", In Graph Based Representations in Pattern Recognition, 1998, pp. 93-100
39. A.Rosenfeld and C.Y.Sher, "Detecting Image Primitives using Feature Pyramids", Journal of Information Sciences, V. 107, pp. 127-147, 1998.
40. Zoltan Tomori, Jozef Marcin and Peter Vilim, "Pyramidal Seeded Region Growing Algorithm and Its Use in Image Segmentation", CAIP, 1999, pp. 395-402
41. G. Borgefors, G. Ramella, G. and Sanniti di Baja, "Permanence-based shape decomposition in binary pyramids", Proc. 10th International Conference on Image Analysis and Processing (ICIAP'99), Venice, Italy, Sep 1999, pp. 38-43.
42. C.L.Tan, B.Yuan, W.Huang, Q.Wang and Z.Zhang, "Text/Graphics Separation using Agent-based Pyramid Operation", International Conference in Document Analysis and Recognition, 1999.
43. P. Brigger, F. Muller, K. Illgner and M. Unser, "Centered pyramids", IEEE Transactions on Image Processing, Vol. 8, No. 9, Sep 1999, pp. 1254-1264.
44. M. Baatz and A. Schape, "Multiresolution segmentation: an optimization approach for high quality multi-scale image segmentation", AGIT 2000.
45. D. Prewer and L. Kitchen, "Weighted linked pyramids and soft segmentation of colour images", ACCV, Vol. 2, Jan 2000, pp. 989-994.

46. E. Sharon, A. Brandt, and R. Basri, "Fast Multiscale Image Segmentation" in IEEE Proc. of Computer Vision and Pattern Recognition (CVPR '00), Vol. I, Hilton Head, SC, June 2000, pp. 70-77.
47. D. Prewer and L. Kitchen, "Soft image segmentation by weighted linked pyramid", Pattern Recognition Letters, Vol. 22, No. 2, 2001, pp. 123-132.
48. C.L. Tan, Z. Zhang, "Text block segmentation using pyramid structure", SPIE Document Recognition and Retrieval, Vol. 8, January 24-25, 2001, San Jose, USA, pp. 297-306.
49. Wei Yu and Jason Fritts, "A Hierarchical Image Segmentation Algorithm," International Conference on Multimedia and Expo (ICME 2002), Lausanne, Switzerland, Aug 2002, pp. 221-224.
50. Rubio TJ, Bandera A, Urdiales C and Sandoval F, "A hierarchical context-based textured image segmentation algorithm for aerial images", in Proceeding of the 2nd International workshop on texture analysis and synthesis, 1st Jun 2002, Denmark, pp. 117-122.
51. A. Kosir and J.F. Tasic, "Pyramid segmentation parameters estimation based on image total variation", In proceedings of IEEE Conference Eurocon 2003.

Irregular pyramid model

52. P. Meer, "Stochastic image pyramids", Comp. Vision, Graphics and Image Proc, Vol. 45, No. 3, 1989, pp. 269-294.
53. A. Montanvert and P. Meer, "Irregular tessellation based image analysis", In Proceedings of the 10th International Conference on Pattern Recognition, Vol. I, Jun 1990, pp. 474-479.
54. W.G. Kropatsch, "Irregular pyramids", Proceedings of the 15th OAGM meeting in Klagenfurt, April 24th-26th 1991, pp. 39-50.
55. A. Montanvert, P. Meer and A. Rosenfeld, "Hierarchical image analysis using irregular tessellations", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 4, April 1991, pp. 307-316.
56. W.G. Kropatsch and A. Montanvert, "Irregular versus regular pyramid structures", Geometrical problems of Image Processing, 1991, pp. 11-22.
57. J.M. Jolion and A. Montanvert, "The adaptive pyramid: a framework for 2D image analysis", CVGIP: Image Understanding, Vol. 55, No. 3, May 1992, pp. 339-348.
58. A. Montanvert and P. Bertolino, "Irregular pyramids for parallel image segmentation", Pattern Recognition (OAGM), May 1992, pp. 13-34.
59. H. Macho and W.G. Kropatsch, "Finding Connected Components with Dual Irregular Pyramids", In Franc Solina and Walter G. Kropatsch editors, Visual Modules, Proc of the OAGM and 1st SDVR Workshop, 1995, pp. 313-321.
60. W.G. Kropatsch and H. Macho, "Finding the structure of connected components using dual irregular pyramids", OAGM, 1995.
61. W.G Kropatsch and S.B. Yacoub, "A revision of pyramid segmentation", ICPR, 1996, pp. 477-481.
62. Horace H.S. Ip and Stephen W.C.Lam, "Alternative strategies for irregular pyramid construction", Image and Vision Computing 14, 1996, pp. 297-304.
63. P. Bertolino and A. Montanvert. "Multiresolution segmentation using the irregular pyramid", In proceedings of the ICIP 96, Lausanne, 17th-19th Sep 1996, pp. 257-260.
64. R. Elias and R. Laganriere, "The disparity pyramid: an irregular pyramid approach for stereoscopic image analysis", Vision Interface, May 1999, pp. 352-359.
65. P.K. Loo and C.L.Tan, "Word Extraction using Irregular Pyramid", Document Recognition and Retrieval VII Conference, SPIE, 2001 at San Jose, CA, USA.
66. P.K.Loo and C.L.Tan, "Detection of Word Group based on Irregular Pyramid", 6th International Conference on Document Analysis and Recognition, Sep 10th-13th 2001 at Seattle, Washington, USA.
67. P.K.Loo and C.L.Tan, "Word and sentence extraction using irregular pyramid", 5th International Workshop on Document Analysis Systems, Aug 19th-21st 2002 at Princeton, New Jersey, USA.
68. M. Saib, Y. Haxhimusa and R. Glantz, "Building irregular graph pyramid using dual graph contraction", Technical report, Pattern Recognition and Image processing group, Institute of Computer Aided Automation, Vienna University of Technology, Jun 2002.

69. J.M. Jolion, "Stochastic pyramid revisited", Pattern Recognition Letters 24, 2003, pp. 1035-1042.
70. P.K.Loo and C.L.Tan "Using Irregular Pyramid for Text segmentation and Binarization of Gray Scale images", Proceedings of the 7th International Conference on Document Analysis and Recognition, Vol. 1, Aug 2003, pp. 594-598.
71. P.K.Loo and C.L.Tan, "Adaptive Region Growing Color Segmentation for Text using Irregular Pyramid", International Workshop on Document Analysis Systems, Sep 2004, USA.

Detection of textual content in real scene images

72. J. Barroso, A. Rafael, E.L. Dagless and J. Bulas-Cruz., "Number Plate Reading Using Computer Vision", IEEE International Symposium on Industrial Electronics, 1997.
73. Y. Liu, T. Yamamura, N. Ohnishi, and N. Sugie, "Detecting Characters in Grey-Scale Scene Image", Lecture Notes in Computer Science (LNCS), Jan 1998, pp. 1352:153-160.
74. J. Ohya, A. Shio and S. Akamatsu, "Recognizing Characters in Scene Images", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, pp. 294-308, Mar 1998.
75. S. Messelodi and C.M. Modena, "Automatic Identification and Skew Estimation of Text Lines in Real Scene Images", Pattern Recognition, Vol. 32, Nov 1999, pp. 791-810.
76. H. Wang, "Automatic Character Location and Segmentation in Color Scene Images", Proceedings of the 11th International Conference on Image Analysis and Processing (ICIAP'01), 2001.
77. S.Lefevre, L.Mercier, V.Tiberghien and N.Vincent, "Multiresolution Color Image Segmentation Applied to Background Extraction in Outdoor Images", IS&T European Conference on Color in Graphics, Image and Vision, pp. 363-367, April 2002.

Textual extraction from web images

78. J Zhou and D. Lopresti, "Extracting Text from WWW Images", In 4th International Conference on Document Analysis and Recognition (ICDAR), Vol. 1, pp. 248-252, Aug 1997.
79. D. Lopresti and J.Zhou, "Locating and recognizing text in WWW Images", Information Retrieval, Vol. 2, pp. 177-206, 2000.
80. T.Kanungo and C.H.Lee, "What Fraction of Images on the Web Contain Text?", 5th International Workshop on Web Document Analysis, 2001.
81. A.Antonacopoulos and D.Karatzas, "Text extraction from web images based on human perception and fuzzy inference", 5th International Workshop on Web Document Analysis, 2001.
82. E.V.Munson and Y.Tsymbalenko, "To Search for Images on the Web, Look at the Text , Then Look at the Images", 5th International Workshop on Web Document Analysis, 2001.
83. D. Karatzas and A. Antonacopoulos, "Two Approaches for Text Segmentation in Web Images", In proceedings of the 7th International Conference on Document Analysis and Recognition, 2003.

Detection of textual content from video images

84. R. Lienhart, "Automatic Text Recognition for Video Indexing", in Proceeding ACM Multimedia, Boston, MA, Nov 1996, pp. 11-20.
85. A. K. Jain and B. Yu, "Automatic text Location in Image and Video Frames", pattern Recognition, Vol. 31, No. 12, 1998, pp. 2055-2076.
86. Osamn Hori, "A Video Text Extraction Method for Character Recognition", 5th International Conference on Document Analysis and Recognition, Sep 1999, pp. 25-28.
87. T.Sato, T. Kanade, E. Hughes, M. Smith and S. -i Satoh, "Video OCR Indexing Digital News Libraries by Recognition of Superimposed Caption", Multimedia System, Vol. 9, No. 5, 1999, pp. 385-395.
88. Y. Zhong, H.J. Zhang and A.K. Jain, "Automatic Caption Localization in Compressed Video", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 22, No. 4, pp. 385-392, April 2000.
89. A. Miene, Th. Hermes and G. Ioannidis, "Extracting Textual Inserts from Digital Videos", 6th International Conference on Document Analysis and Recognition, Sep 2001.

90. R. Lienhart, A. Wernicke, "Localizing and Segmenting Text in Images and Videos", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, No. 4, April 2002.

Textual extraction in engineering drawing applications

91. D.N. Ying, E.J. Wang, L. Ye, W. Li and Y. Wang, "A Study on Automatic Input and Recognition of Engineering Drawing", Proceeding CAD/Graphics, Hang Zhou, China, Sep 1991, pp. 478-481.
92. L.Csink, "On Integrating paper-based general Electronic diagrams into a CAD environment", pattern Recognition, pp. 56-62, May 1992.
93. CP. Lai and R. Kasturi, "Detection of Dimension Sets in Engineering Drawings", IEEE Transaction Pattern Analysis & Machine Intelligence, Vol. 16, no. 8, pp. 848-855, 1994.
94. Z. Lu, "Detection of Text Region from Digital Engineering Drawings", IEEE Transactions on Pattern Analysis and machine Intelligence, Vol. 20, pp. 431-439, April 1998.
95. M.Zhao, Y.Yang and H.Yan, "An Adaptive Thresholding Method for Binarization of Blueprint Images", Pattern Recognition letters, V. 21, pp.927-943, 2000.
96. C.H. Tsai and Y.L. Chi, "An Extractor for Understanding Text Strings from Digital Engineering Drawings", Proceedings of SCI 2001/ISAS 2001, World Multi-Conference on Systemics, Cybernetics and Informatics, Vol. XIV, Orlando, Florida, 2001

Textual segmentation in form processing

97. B.Yu and A.K.Jain, "A Generic System for Form Dropout", IEEE Transactions on Pattern Analysis and Machine Intelligence, V. 18, No. 11, Nov 1996.
98. I. Aksak, Ch. Feist, V.Kiiko, R. Knoefel, V. Matsello, V. Oganovskij, M. Schesinger, D. Schlesinger and G. Stanke, "Extraction of Filled-in Data from Color Forms", Lecture Notes in Computing Science (LNCS), Vol. 1296, pp. 98-105, Sep 1997.
99. S. Djeziri, F. Noubouud and R. Plamondon, "Extraction of Signature from Check background based on a filiformity Criterion", IEEE Transaction Image Processing, Vol. 7, No. 10, pp. 1425-1438, oct 1998.
100. W.S. Wong, N. Sherkat and T. Allen, "Use of Color in Form Layout Analysis", 6th International Conference On Document Analysis and recognition (ICDAR 2001), Seattle, Sep 2001.

Recovery of textual content from document images for archiving

101. K.S.Kiernan, "Digital Image Processing and the Beowulf Manuscript", Literary and Linguistic Computing 6, pp. 20-27, 1991.
102. Hideyuki Negishi, Jien Kato, Hiroyuki Hase and Toyohide Watanabe, "Character Extraction from Noisy Background for an Automatic Reference System", In Proc. 5th Int. Conf. On Document Analysis and Recogn. (ICDAR), 1999, pp. 143-146.
103. Y.Yang and H.yan, "An Adaptive Logical method for Binarization of Degraded Document Images", Pattern Recognition, V. 33, pp. 787-807, 2000.
104. Z.Zhang and C.L.Tan, "Recovery of Distorted Document Images from Bound Volumes", 6th International Conference on Document Analysis and Recognition (ICDAR '01), Sep 2001, pp. 429-433.
105. G.Leedham, S.Varma, A.Patankar and V.Govindaraju, "Separating text and Background in Degraded Document Images-A Comparison of Global Thresholding techniques for Multi-Stage Thresholding", proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition, 2002.

Textual segmentation in newspaper document

106. D. Wang and S.N. Srihari, "Classification of Newspaper Image Blocks using texture Analysis", Computer Vision Graphics and Image Processing, Vol. 47, pp. 327-352, Jan 1989.
107. P.E.Mitchell and H.Yan, "Newspaper Document Analysis Featuring Connected Line Segmentation", 6th International Conference on Document Analysis and Recognition, 2001, pp. 1181-1185.

108. C. L. Tan and Q. H. Liu, "Extraction of newspaper headlines from microfilm for automatic indexing", International Journal on Document Analysis and Recognition, Vol.6, no.3, pp.201-210, March 2004.

Document image text extraction and layout analysis

109. K.Y.Wong, R.G.Casy and F.M.Wahl, "Document analysis system", IBM J. Res. Development, Vol 26, 642-656 (1982).
110. F.M. Wahl, K.Y. Wong and R.G. Casey, "Block Segmentation and Text Extraction in Mixed Text/Image Documents", Computer Graphic Image Processing, Vol. 20, 1982, pp. 375-390.
111. G. Nagy and S. Seth, "A Prototype Document Image Analysis System for Technical Journals", In Proceedings of the International Conference on Pattern Recognition, 1984, pp. 347-349.
112. G.Nagy and S.Seth, "Hierarchical representation of optically scanned documents", In Proc. 7th Int. Conf. Pattern Recognition. (ICPR), 1984, pp. 347-349.
113. A. Rastogi and S.N. Srihari, "Recognizing textual blocks in document images using the Hough transform", TR 86-01, Dept. of CS, SUNY at Buffalo, 1986.
114. Srihari, S.N., "Document Image Understanding", Proceedings of ACM-IEEE C/S Fall Joint Computer Conference, Dallas, TX, November, 1986, pp. 87-96
115. L.A. Fetcher and R. Kasturi, "A Robust Algorithm for text String Separation from Mixed Text/Graphics Images", IEEE Transactions on Pattern Analysis & Machine Intelligence, Vol. 10, no. 6, pp. 910-918, 1988.
116. Y.Ishitani, "Document Image Analysis with Cooperative Interaction Between Layout Analysis and Logical Structure Analysis", Document layout Interpretation and Its Application proceeding, DLIA, 1991.
117. S. Srihari, S. Lam, V. Govindaraju, R. Srihari, J. Hull, and E. Yair. "Document understanding: Research directions", Technical Report CEDAR-TR-92-1, SUNY Buffalo - CEDAR, May 1992.
118. T. Pavlidis and J. Zhou, "Page Segmentation and Classification", Computer Vision Graphics and Image Processing, Vol. 54(6), pp. 484-496, Nov 1992.
119. D.S. Bloomberg, "Multi-resolution Morphological Analysis of Document Images", Proceeding SPIE Visual Communication Image Processing, Vol. 1818, 1992, pp.648-662.
120. A.K. Jain and S. Bhattacharjee, "Text Segmentation using Gabor Filters for Automatic Document Processing", Machine Vision and Applications, Vol. 5(3), pp. 169-184, 1992.
121. L.O'Gorman, "The Document Spectrum for Page Layout Analysis", IEEE Transactions on Pattern Analysis and Machine Intelligence, V. 15, No. 11, Nov 1993.
122. M. Kamel and A. Zhao "Extraction of Binary Character/Graphics Images from Gray Scale Document Images", CVGIP : Graph Models and Image Processing, Vol. 55, No. 3, pp. 203-217, 1993.
123. K.K. Chin and J. Saniie, "Morphological Processing for Feature Extraction", Proceeding SPIE, Vol. 2030, 1993, pp. 288-302.
124. M.Kamel and A.Zhao, "Extraction of Binary Character/Graphics Images from Grayscale Document Images", Graphical Models and Image processing, V. 55, No. 3, May 1993.
125. Li-Wang, Theo Pavlidis, "Direct Gray-Scale Extraction of features for Character recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 10, Oct 1993, pp. 1053-1067.
126. O. Deforges and D. Barba, "A Fast Multi-resolution Text-line and non Text-line Structures Extraction and Discrimination Scheme for Document Image Analysis", In ICIP Proceedings, Vol. 1, Aug 1994, pp.134-138.
127. Y. Lu and A.C. Tisler, "Gray Scale Filtering for Line and Word Segmentation", proceedings of the 3rd International Conference on Document Analysis and Recognition, 1995.
128. J. Ha, R. M. Haralick and I. T. Phillips, "Recursive X-Y Cut using Bounding Boxes of Connected Components", Proceedings of the 3rd International Conference on Document Analysis and Recognition, 1995.
129. K.C. Fan, L.S. Wang and Y.K. Wang, "Page Segmentation and Identification for Intelligent Signal Processing", Signal Process, Vol. 45, pp. 329-346, 1995.
130. N.G.Bourbakis, "A Methodology of Seperating Images from text Using an OCR Approach", proceedings of the 1996 IEEE International Joint Symposia on Intelligence and Systems, 1996.

131. P.Parodi and G.Piccioli, "A Fast and Flexible Statistical method for Text Extraction in Document Pages", Proceedings of the 1996 Conference on Computer Vision and Pattern recognition, 1996.
132. Y.Y.Tang, S.W.Lee and C.Y.Suen, "Automatic Document Processing: A Survey", Pattern Recognition, V. 29, No. 12, pp. 1931-1952, 1996.
133. S-W Lee, D-J Lee, H-S Park, "A New Methodology for Gray-Scale Character Segmentation and Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 12, Dec 1996, pp. 1045-1050.
134. J.Liang, I.T.Phillips, J.Ha and R.M.Haralick, "Document Zone Classification Using Sizes of Connected-components", Proceedings of the SPIE, V. 2660, 1996.
135. R.G.Casey and E.Lecolinet, "A Survey of Methods and Strategies in Character Segmentation", IEEE Transactions on pattern Analysis and Machine Intelligence, V. 18, No. 7, July 1996.
136. U. Pal and B. B. Chaudhuri, "Automatic separation of words in Indian multi-lingual multi-script documents", In Proc. 4thICDAR, pp. 576-579, 1997.
137. Doermann, "The retrieval of document images: a brief survey," Proceedings of the Fourth International Conference on Document Analysis and Recognition, 1997. vol.2, pp: 945 -949
138. V. Wu, R. Manmatha and E.M. Riseman, "Finding Text in Images", in Proceeding 2nd ACM International Conference Digital Libraries, Philadelphia, PA, July 1997.
139. H.Hase, T.Shinokawa, M.Yoneda, M.Sakai and H.Maruyama, "Character String Extraction by Multi-stage Relaxation", 4th International Conference on Document Analysis and Recognition, 18-20 August 1997, pp 298-302
140. A. Antonacopoulos, "Local Skew Angle Estimation from Background Space in Text Regions", Proceedings of the 4th International Conference on Document Analysis and Recognition (ICDAR'97), Ulm, Germany, August 18-20, 1997, pp. 684-688
141. A.K. Jain and B. Yu, "Document representation and its Application to Page Decomposition", IEEE Transaction Pattern Analysis and Machine Intelligence, Vol. 20, pp. 294-308, Mar 1998.
142. R.Cattoni, T.Coianiz, S.Messelodi and C.M.Modena, "Geometric Layout Analysis Techniques for Document Image Understanding: a Review", 1998.
143. L. O'Gorman and R. Kasturi: "Document Image Analysis: An Executive Briefing", IEEE Computer Society Press 1998
144. O.Okun, M.Pietikainen and J.Sauvola, "Robust Skew Estimation on Low-resolution Document Images", Proceeding 5th International Conference on Document Analysis and Recognition, pp. 621-624, 1999.
145. C. Di Ruberto, G. Rodriguez, and S. Vitulano, "Image segmentation by texture analysis", In Proceedings of International Conference on Image Analysis and Processing, pages 376-381, Los Alamitos, CA, 1999. IEEE Computer Society.
146. Y. Rui, T. S. Huang, and S.-F. Chang. "Image retrieval: Current techniques, promising directions and open issues" Journal of Visual Communication and Image Representation, March 1999.
147. C.H. Chan, L.F. Pau and P.S.P. Wang, "Handbook of Pattern Recognition and Computer Vision", (2nd edition), 1999.
148. O.Okun and M.Pietikainen, "A Survey of Texture-based methods for Document Layout Analysis", Proc. of Workshop on Texture Analysis in Machine Vision (WTAMV'99), June 14-15 1999, Oulu, Finland, pp. 137-148.
149. R. Malik and S.A. Chin, "Extraction of text in images", In Proceedings of the International Conference on Information Intelligence and Systems, Bethesda, MD, USA, pages 534-537, 1999.
150. Dae-Seok Ryu, Sun-Mee Kang and Seong-Whan Lee, "Parameter-Independent Geometric Document Layout Analysis, ICPR 2000, pp. 4397-4400
151. Y.M.Y. Hassan and L.J. Jaram, "Morphological Text Extraction from Images", IEEE Transactions on Image Processing, Vol. 9, No. 11, Nov 2000.
152. J. Patrick Bixler, "Tracking Text in Mixed-mode Documents", Proceedings of the ACM Conference on Document Processing Systems, Santa Fe, New Mexico, United States, pp. 17-185, 2000.
153. Y.Wang, I.T.Phillips and R.Haralick, "Statistical-based Approach to Word Segmentation", Proceedings of the International Conference on Pattern Recognition, 2000.

154. P.Clark and M.Mirmehdi, "Finding Text regions using Localised measures", Machine Vision Conference, pp. 675-684, Sep 2000.
155. G.Nagy, "Twenty years of Document Image Analysis in PAMI", IEEE Transactions on Pattern Analysis and machine Intelligent, V. 22, No. 1, pp. 38-62, Jan 2000.
156. G.Harit, S.Chaudhury, P.Gupta, N.Vohra and S.D.Joshi, "A Model Guided Document Image Analysis Scheme", Proceedings of the International Conference on Document Analysis and Recognition, 2001
157. H.Yan, "Detection of Curved text Path based on the Fuzzy Curve-tracing (FCT) Algorithm", ICDAR 2001.
158. M. Pietikainen and O. Kun, "Edge-based Method for Text Extraction from Complex Document Image", Proceeding 6th International Conference on Document Analysis and Recognition (ICDAR2001), Seattle, WA, USA, pp. 286-291, Sep 2001.
159. Fu Chang, "Retrieving information from document images: problems and solutions", International Journal on Document Analysis and Recognition, Springer-Verlag, 2001, pp. 46-55.
160. B.Waked, C.Y.Suen and S.Bergler, "Segmenting Document Images using Diagonal White Runs and vertical Edges", In Proceedings of the Sixth International Conference on Document Analysis and Recognition, Seattle, Washington, September 2001.
161. S.W.Lee and D.S.Ryu, "Parameter-Free Geometric Document Layout Analysis", IEEE Transactions on Pattern Analysis and Machine Intelligent, V. 23, N. 11, Nov 2001.
162. J. Duong, M. Lote, H. Emptos and C.Y. Suen, "Extraction of Text Areas in Printed Document Images", Proceedings of the 2001 ACM Symposium on Document Engineering, Atlanta, Georgia, USA, pp. 157-165, 2001.
163. 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.
164. R. Cao and C.L. Tan, "Separation of overlapping text from graphics", International Conference on Document Analysis and Recognition, ICDAR 2001, 10-13 Sept 2001, Seattle, USA, pp. 44-48.
165. Q. Yuan, and C.L. Tan, "Text Extraction from Gray Scale Document Images Using Edge Information", Proceedings of the International Conference on Document Analysis and Recognition, ICDAR'01, September 10-13, 2001, Seattle, USA, pp. 302-306.
166. N.-V.Marti and H. Bunke, "Text Line Segmentation and Word Recognition in a System for General Writer Independent Handwriting Recognition", ICDAR, 2001.
167. C.L.Tan, W.Huang, Z.Yu and Y.Xu, "Imaged Document Text Retrieval without OCR", IEEE Transactions on Pattern Analysis and Machine Intelligence, V. 24, No. 6, June 2002.
168. D.X. Zhong, "Extraction of Embedded and/or Line-touching Character-like Objects", Pattern Recognition, Vol. 35, pp. 2453-2466, 2002.
169. J.Zhang and T.Tan, "Brief Review of Invariant Texture Analysis Methods", pattern Recognition, V. 35, pp. 735-747, 2002.
170. S. Mao, A. Rosenfeld and T. Kanungo, "Document Structure Analysis Algorithms: A Literature Survey", Proceedings SPIE Electronic Imaging, Vol. 5010, Jan 2003, pp. 197-207.
171. J.Fan, "Text Extraction via an Edge-bounded Averaging and a Parametric Character Model", Electronic Imaging (SPIE), San Jose, Jan 2003.

Gray scale image thresholding

172. N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", IEEE Transactions on System, man, and Cybernetics, V. SMC-9, No. 1, Jan 1979.
173. T.Pun, "Entropic Thresholding, A New Approach", Computer Graphics and Image Processing, V. 16, pp. 210-239, 1981.
174. J.M.White and G.D.Rohrer, "Image Thresholding for Optical Character Recognition and Other Application requiring Character Image Extraction", IBM Journal Resource Development, V. 27, No. 4, July 1983.
175. J.M. White and G.D. Rohrer, "Image Thresholding for Optical Character Recognition and Other ...", IBM Journal Resource Development, Vol. 27, No. 4, pp. 400-411, 1983.

176. J.N.Kapur, P.K.Sahoo and A.K.C.Wong, "A New method for Gray-level Picture Thresholding Using the Entropy of the Histogram", *Computer Vision, Graphics, and Image Processing*, V. 29, pp. 273-285, 1985.
177. J. Bernsen, "Dynamic Thresholding of Grey-level Images", *Proceedings of International Conference Pattern Recognition*, Paris, France, 1986, pp. 1251-1255.
178. J. Kittler and J. Illingworth, "Minimum Error Thresholding", *Pattern Recognition*, Vol. 19, No. 1, pp. 41-47, 1986.
179. A.S.Abutaleb, "Automatic Thresholding of Gray-Level Pictures Using Two-Dimensional Entropy", *Computer Vision, Graphics, and Image Processing*, V. 47, No. 1, pp. 22-32, 1989.
180. S.D.Yanowitz and A.M.Bruckstein, "A New Method for Image Segmentation", *Computer Vision, Graphics, and Image Processing*, V. 46, No. 1, pp. 82-95, 1989.
181. W.S. Baird, S.E. Jones and S.J. Fortune, "Image Segmentation by Shape Directed Covers", *Proceeding of International Conference in Pattern Recognition*, pp. 820-825, 1990.
182. Lawrence O'Gorman, "Binarization and Multi-thresholding of Document Images using Connectivity", *Computer Vision, Graphics & Image Processing*, Vol. 56(6), 1994, pp. 494-506.
183. M.S.Chang, S.M.Kang, W.S.Rho, H.G.Kim and D.J.Kim, "Improved Binarization Algorithm for Document Image by Histogram and Edge Detection", *proceedings of the 3rd International Conference on Document Analysis and Recognition*, 1995.
184. M.L.G.Althouse and C.I.Chang, "Image Segmentation by Local Entropy methods", *proceedings of the 1995 International Conference on Image Processing*, 1995.
185. O.D.Trier and T.Taxt, "Improvement of 'Integrated Function Algorithm' for Binarization of Document Images", *Pattern Recognition Letters*, V. 16, pp. 277-283, 1995.
186. O. D. Trier and A. K. Jain, "Goal-directed evaluation of binarization methods", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 12, December 1995, pp. 1191-1201.
187. A.T.Abak, U.Baris and B.Sankur, "The performance Evaluation of Thresholding Algorithms for Optical Character Recognition", *IEEE*, 1997.
188. Y.Liu and S.N.Srihari, "Document Image Binarization based on texture features", *IEEE Transactions on Pattern Analysis and Machine Intelligent*, V. 19, No. 5, May 1997.
189. J. Sauvola, T. Seppanen, S. Haapakoski, and M.Pietikainen, "Adaptive Document Binarization," pp.147-152, *ICDAR 97*, Ulm, Germany, 1997.
190. A.E.Savakis, "Adaptive Document Image Thresholding Using Foreground and Background Clustering", *Proceedings of International Conference on Image Processing*, 1998.
191. Y.Solihin and C.G.Leedham, "Integral Ratio: A New Class of Global thresholding techniques for Handwriting Images", *IEEE Transactions on pattern Analysis and Machine Intelligent*, V. 21, No. 8, Aug 1999.
192. J.Sauvola and M.Pietikainen, "Adaptive Document Image Binarization", *Pattern Recognition*, V. 33, pp. 225-236, 2000.
193. F.Chang, "Retrieving Information from Document Images: Problems and Solutions", *IJDAR*, 2001.
194. A.D.Woud and M.Kamel, "Binarization of Document Images Using Image Dependent Model", *proceeding of the 6th International Conference on Document Analysis and Recognition*, 2001.
195. S.Rodtook and Y.Rangsanseri, "Adaptive Thresholding of Document Images Based on Laplacian Sign", *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'01)*, 2001.
196. D.Sylwester and S.Seth, "Adaptive Segmentation of Document Images", *Proceedings of the 6th International Conference on Document Analysis and Recognition*, 2001.
197. B.Sankur and M.Sezgin, B. Sankur, M. Sezgin, "Image Thresholding Techniques: A Survey over Categories", *Pattern Recognition*, 2001.
198. N.Bonnet, J.Cutrona and M.Herbin, "A 'no-threshold' histogram-based Image Segmentation Method", *Pattern Recognition*, V. 35, pp.2319-2322, 2002.
199. I.K.Kim, D.W.Jung and R.H.Park, "Document Image Binarization based on Topographic Analysis Using a Water Flow Model", *Pattern Recognition*, V. 35, pp. 265-277, 2002.

Processing of color document images

200. H. Wong and H. Yan, "Text Extraction from Color Map Images", *Journal Electron Imaging*, Vol. 3, No. 4, pp. 390-396, 1994.
201. Zhiang Xiang and Gregory Joy, "Color Image Quantization by Agglomerative Clustering", *IEEE Computer Graphics*, May 1994, pp 44-48.
202. Y. Zhong, K. Karu and A.K. Jain, "Locating Text in Complex Color Images", *Pattern Recognition*, Vol. 28, No. 10, pp. 1523-1535, 1995.
203. H.M. Suen and J.F. Wang, "Text String Extraction from Images of Color-printed Documents", *Proceeding Inst. Elect. Eng. Vis., Image Signal Process.*, Vol. 143, No. 4, pp. 210-216, 1996.
204. L. Velho, J. Gomes and M.V.R. Sobreiro, "Color Image Quantization by Pairwise Clustering", *proceedings of SIBGRAP'96*, pp. 203-210, Oct 1997.
205. H.M. Suen and J.F. Wang, "Segmentation of Uniform-Colored text from Colour Graphics Background", *IEEE Proceeding Vision Image Signal Processing*, Vol. 144, No. 6, pp.317-322, 1997.
206. P. Scheunders, "A Comparison of Clustering Algorithms Applied to Color Image Quantization", *Pattern Recognition Letter*, pp. 1379-1384, 1997.
207. A. Tremeau and N. Borel, "A Region Growing and Merging Algorithm to Color Segmentation", *Pattern Recognition*, Vol. 30, No. 7, 1997, pp. 1191-1203.
208. A. Mehnert and O. Jackway, "An Improved seeded region growing algorithm", *Pattern Recognition Letters* 18, 1997, pp. 1065-1071.
209. W.Y. Chen and S.Y. Chen, "Adaptive Page Segmentation for Color Technical Journals Cover Images", *Images and Vision Computing*, Vol. 16, No. 12, pp. 855-877, Aug 1998.
210. Y.H.Gong, G.Proietti and C.Faloutsos, "Image Indexing and Retrieval Based on Human Perceptual Color Clustering", *Computer Vision and Pattern recognition*, 1998.
211. K. Sobottka, H. Bunke and H. Kronenberg, "Identification of Text on Colored Book and Journal Covers", *In Proceedings of the 5th International Conference on Document Analysis and Recognition*, pp. 57-62, Sep 1999.
212. Y.Deng, B.S.Manjunath and H.Shin "Color image segmentation", *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '99*, Fort Collins, CO, vol.2, pp.446-51, June 1999.
213. P.K. Kim, "Automatic Text Location in Complex Color Images Using Color Quantization", *Proceeding of the IEEE Region 10 Conference (TENCON 99)*, Vol. 1, pp. 629-632, 1999.
214. H. Hase, T. Shinokawa, M. Yoneda, M. Sakai, and H. Maruyama, "Character String Extraction from a Color Document", *Proc. of the 5th International Conference on Document Analysis and Recognition*, Bangalore, India, 1999, pp.75-78.
215. D.X. Zhong, "Color Space Analysis and Color Image Segmentation", *VIP2000, Pan-Sydney Area Workshop on Visual Information Processing*, December 2000.
216. L.Lucchese and S.K.Mitra, "Color Image Segmentation: A State-of-the-Art Survey", *Image Processing, Vision and Pattern recognition, Proceeding of the India National Science Academy*, Vol. 67, A, No. 2, Mar 2001, pp. 207-221.
217. T. Perroud, K. Sobottka and H. Bunke, "Text Extraction from Color Documents – Clustering Approaches in Three and Four Dimensions", *6th International Conference on Document Analysis and recognition*, Seattle, Sep 2001.
218. H.Hase, M.Yoneda, T.Shinokawa and C.Y.Suen, "Alignment of Free layout Color texts for Character Recognition", *6th International Conference on Document Analysis and Recognition*, Seattle, Sep 2001.
219. T.Q. Chen and Yi Lu, "Color Image Segmentation – An Innovative Approach", *Pattern Recognition*, Vol. 35, pp. 395-405, 2002.
220. C. Strouthopoulos, N. Papamarkos and A.E. Atsalakis, "Text Extraction in Complex Color Documents", *Pattern Recognition*, vol. 35, pp. 1743-1758, 2002.
221. H.D. Cheng, X.H. Jiang and J. Wang, "Color Image Segmentation based on Homogram Thresholding and Region Merging", *Pattern Recognition*, Vol. 35, pp. 373-393, 2002.

222. A.S.Nugroho, S.Kuroyanagi and A.Iwata, "An Algorithm for Locating Characters in Color Image using Stroke Analysis Neural Network", Proceeding of the 9th International Conference on Neural Information Processing 9ICONIP'02), V.4, pp. 2132-2136, Nov 18-22, 2002.

Web sites

223. Hotcard Technology Pte Ltd, <http://www.hotcardtech.com/>