

## CHAPTER III

### EXPERIMENT

#### 3.1 Materials

1. Plain paper : 80 gram A4

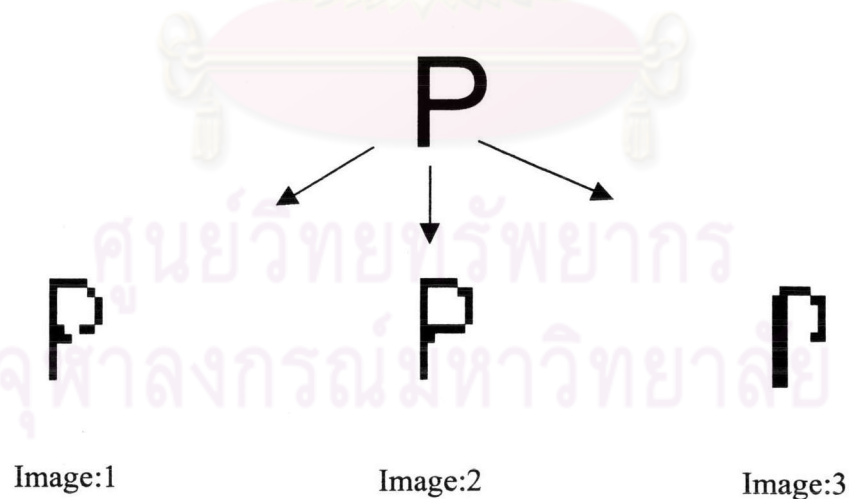
#### 3.2 Apparatus

1. Personal Computer : Pentium 100 MHz.
2. Personal Computer : Pentium III 1000 MHz.
3. Scanner : Microtek Scanmaker 3630
4. Inkjet printer : hp deskjet 670 C
5. Software Microsoft word : Microsoft Corporation 1997.
6. Software Microsoft Notepad : Microsoft Corporation 1999.
7. Software Scanner Twain : MGI Software Corporation 1999
8. Software Visual C++ v.6 : Microsoft Corporation 1998.

### 3 Experiment

#### 3.3.1 Code Algorithm Step

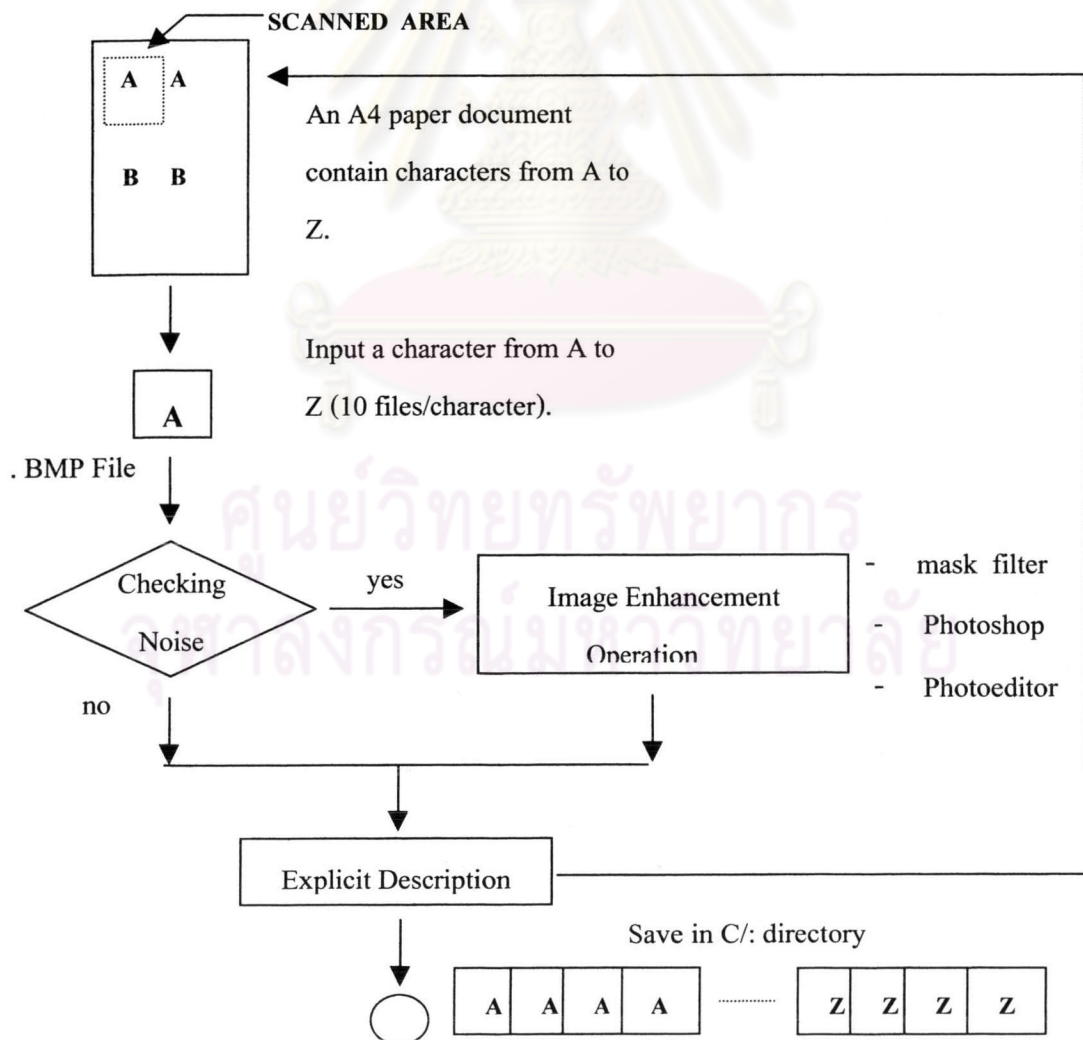
The image file (as shown in figure 3-1) was scanned from the same character at different times. It shows the different pixel arrangement and width of the characters as recognized at the boundaries of the objects in the image. This happens because the intensity of the fluorescent light reflected from the document is insufficient to be accurately recorded by the CCD sensor. These incorrect readings will distort the characters shape. These distorted letters will influence function of the algorithm. To make a proper algorithm function capable of analyzing 10 letters each for the alphabet from A-Z an average of the code information (as shown in Appendix A) must be established. So this step is intended to correct the code information of each character using image segmentation and feature extraction.



**Figure 3-1** Image of the character scanning at different period of time.

### 3.3.1.1 Image Segmentation Data Algorithm

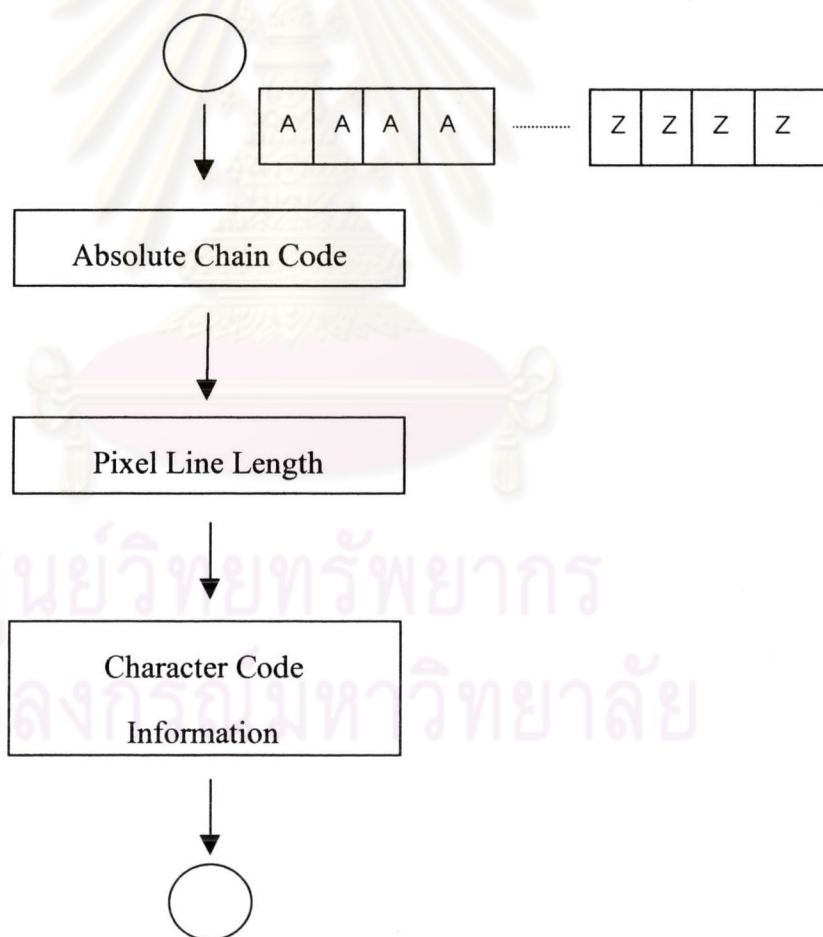
The process (Figure 3-1) started by setting the input scanning specification mode. Then scanning the characters or an object on the document. When an image was done the noise have to be check by go through an additional stage of image enhancement either by using a mask filter or an image editing software such as Adobe Photoshop and Photoeditor to eliminate the noise. This process was done for each letter of the English Alphabet from A-Z. Ten files were created for each letter. The code of all the characters was kept in directory C which served as a database. Then all of the characters went through the process of explicit description to create object boundaries.



**Figure 3-2** Flowchart of an algorithm in image segmentation part.

### 3.3.1.2 Feature Extraction Data Algorithm

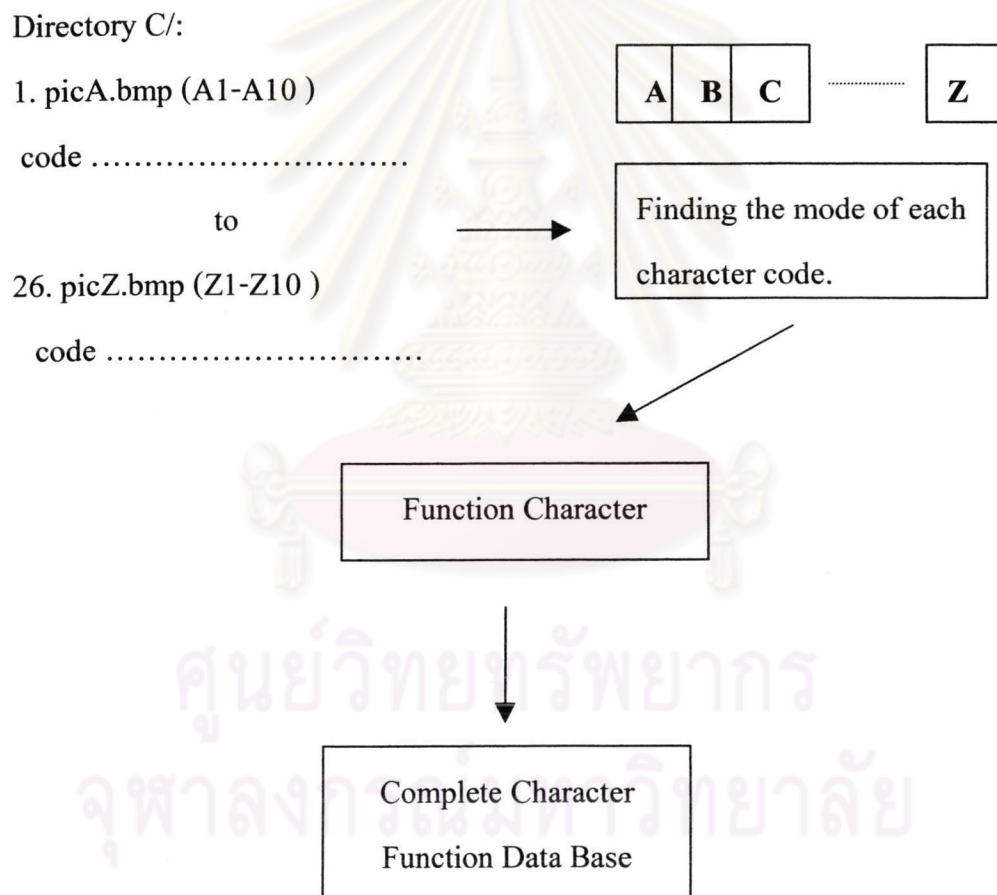
After the process of Explicit Description from Segmentation part. The pixel location information was used in the step of Absolute Chain Code to translate the code out of the pixel character structure (Figure 3-2). Next, the section of Pixel Line Length process would find a pixel distance or a length of each pixel line in the character. Finally, all of the character files that passed the process mentioned above would have their own code characteristic information.



**Figure 3-3** Flowchart of an algorithm in feature extraction part.

### 3.3.1.3 Object Classification Data Algorithm

After getting character code information from characters A to Z, the function of each character was made by finding the mode from the code information of each character (Figure 3-3). A complete character function database was then ready to be used.



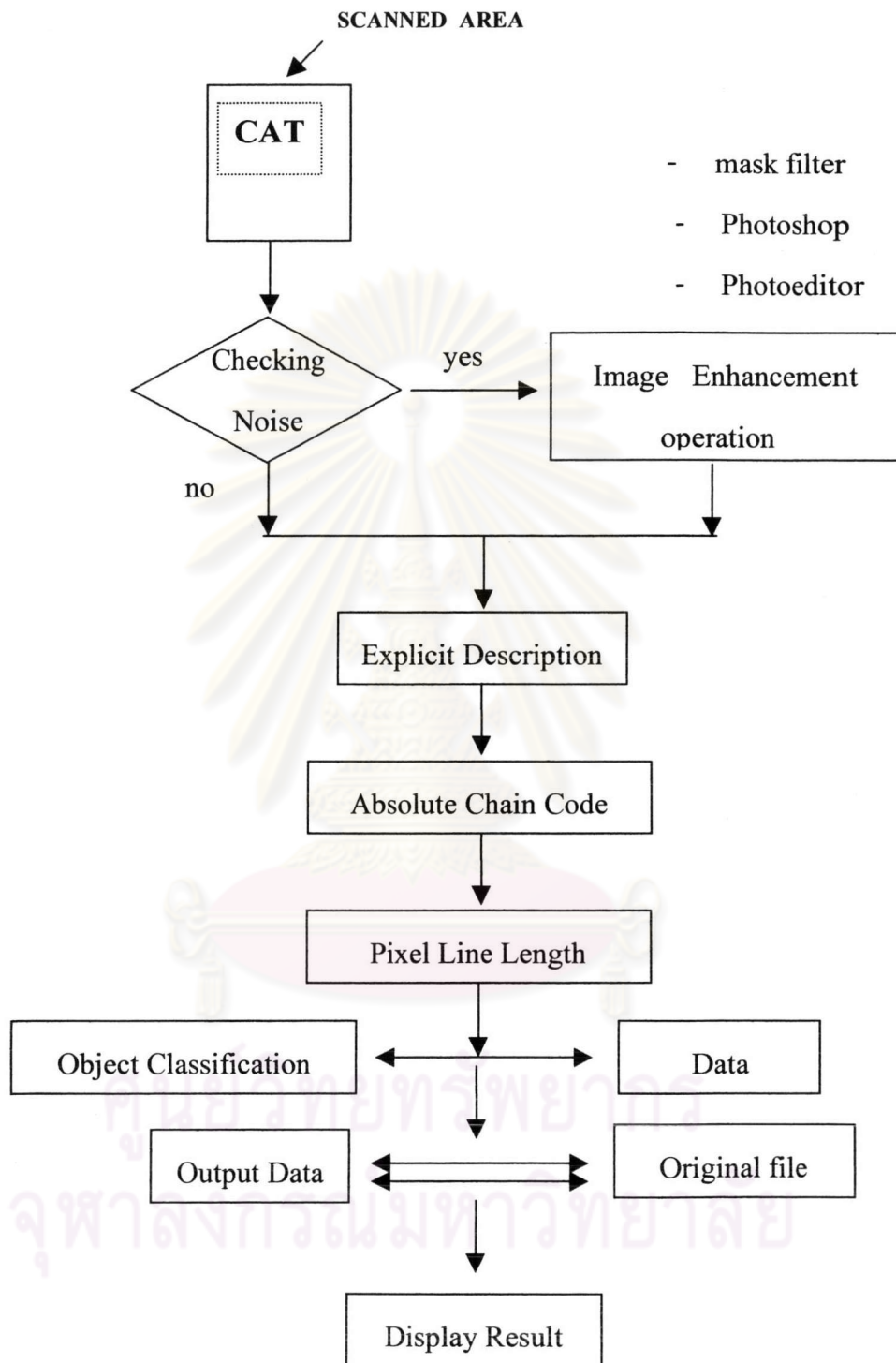
**Figure 3-4** Flowchart of an algorithm in object classification part.

### 3.3.2 Processing Algorithm step

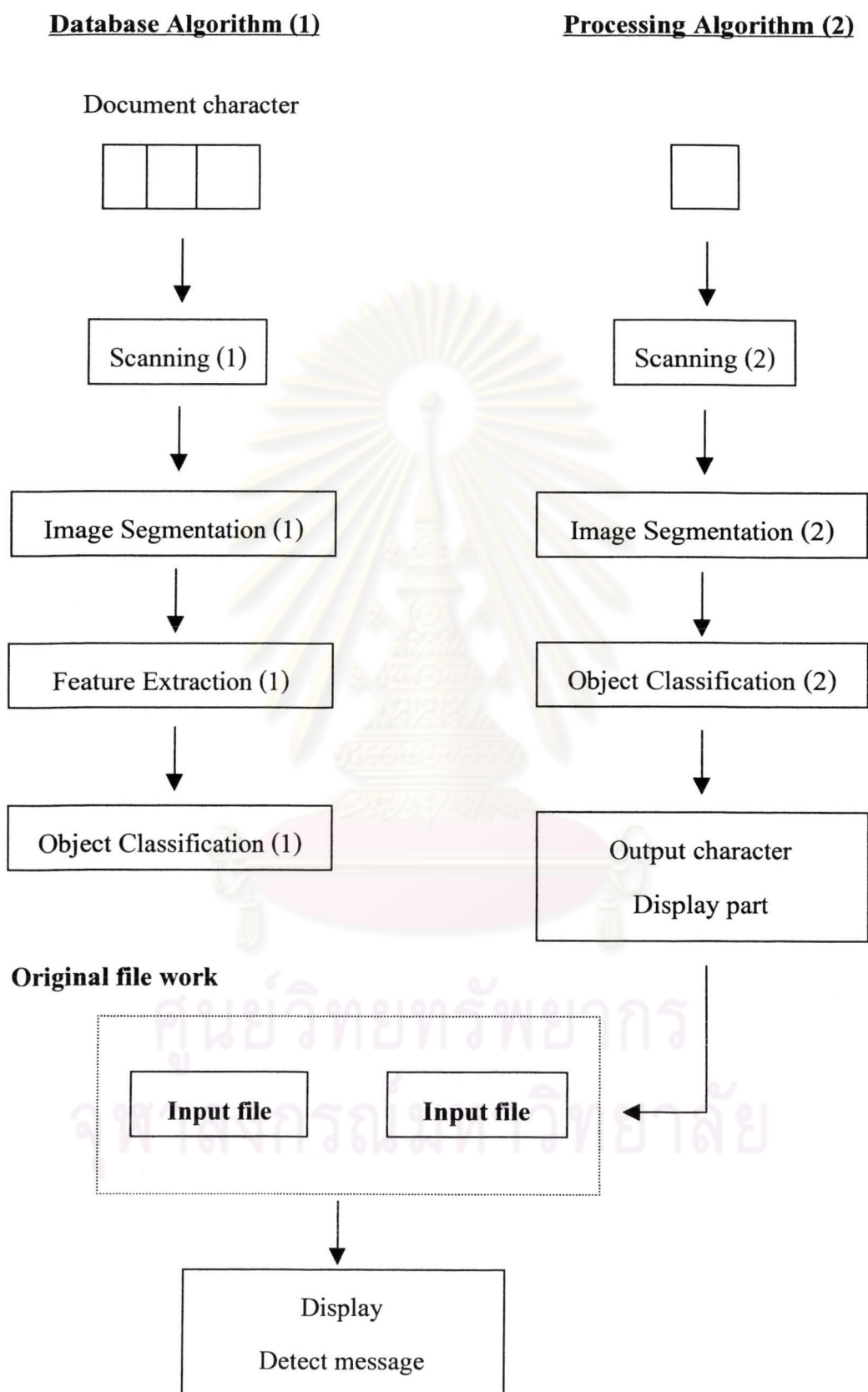
This part of processing was the stage of testing the operation. A complete character function from A to Z was kept and ready to be used in directory C. Then the stage of this algorithm part (Figure 3-4) started with scanning the letters on a document, having the same image input condition as Image Segmentation Data Algorithm part. Then the scanned image went through the stage of Explicit Description, Absolute Chain Code and Pixel Line Length. After all of the letters were extracted, the section of object classification classified the letter on document by comparing with character function from A to Z and then displayed the results on a monitor. The matching algorithm took part in matching the letters that were translated with the letters from the original digital file, and checking whether or not the position of the letters are correct .

The process was tested using a printed sheet of 10 documents containing characters, These documents were transformed to the form of digital files. Then each file went through a Processing Algorithm to determine efficiency of the algorithm.

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



**Figure 3-5** Flowchart of the process algorithm.



**Figure 3-6** Flowchart of a whole part algorithm.