Khmer Segmentation Study Report

25 September 2008

Prepared by: Mr. ING LENG IENG

Cambodia Country Component

PAN Localization project

PAN Localization Cambodia (PLC) of IDRC

Table of Contents

1. Khmer Script

1.1. Introduction to Khmer Script

Khmer, the official language of Cambodia, is one of the earliest writing systems used in Southeast Asia. It belongs to the Mon-Khmer group of Austro-Asiatic languages. The Khmer script, known as Abugida, developed from Pallava script of India before the 7th century. Ultimately, it was the descendant of ancient Brahmi script of India [1]. The Khmer Script is written from left to right and downwards when horizontal space runs out [2]. Khmer can be clearly distinguished from its neighboring languages which are mostly tonal languages in the way that it is a syllabic alphabet. In other words, one letter represents a syllable in the form of a consonant which followed by an inherent vowel. Moreover, a Khmer word is the result of a cluster of Khmer letters in the form of ligatures. Normally, a word consists of a consonant with the inherent vowel to the left or right, sometimes followed by another consonant to the most right-side. Furthermore, a Khmer word may consist of two superscripts and/or two subscripts at most. Superscripts are those dependent vowels that appear on top of the main body, whereas subscripts are those which appear below the main body and are divided into two forms, i.e. some are dependent vowels and some are the smaller versions of, though not all, the consonants. The followings are some of the examples illustrating Khmer words written in one of the commonly used Khmer font **Limon S1** with some possible combinations of a normal consonant, vowels, superscripts, and subscripts.

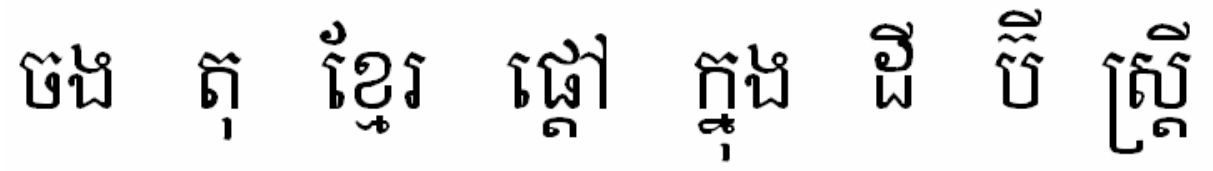


Figure 1.1 *Khmer words written in one of the Khmer fonts, Limon S1*

1.2. Types of Khmer font

Prior to the emergence of Khmer Unicode, many non-unicode standard fonts had been developed. In general, they are categorized into few categories such as Limon fonts, Khek fonts from Khek Bros [3], Zero-space fonts, to name just a few.

1.3. Widely used fonts

Although many Khmer fonts had been developed, most of them were created for specific purpose. Some institutions, organizations, etc. created their own fonts for in-house usage. As the result, the number of fonts increased dramatically from year to year. However, the most widely used fonts ever seen in Cambodia are Limon fonts. In particular, Limon S1 with the size of 22 and

Limon R1 with larger size used for title have been used in ordinary documents as well as some formal documents notwithstanding people may use such other fonts developed within the same category as Limon S2, Limon S3, etc.

2. Khmer OCR Scope

2.1. Selection of font for OCR

Initially, we select Limon S1 size 22 as our research and development since the font is widely used for documenting Khmer text in many organizations, private institutions as well as some government institutions. Likewise, newspaper and books also have the same font used for printing purpose.

2.2. Scope

The study covers segmentation details of selected font and the recognition system of Main Body only. The output of recognition will be the identity of 138 Main Bodies.

3. OCR Modules

There are six significant modules in the Khmer OCR.

- Module 1: Line Separation
- Module 2: Text Band Calculation and Character Separation
- Module 3: Text Band Adjustment and Segmentation
- Module 4: Input Data Preparation
- Module 5: Recognition System
- Module 6: Mapping

3.1. Line Separation

The process is to separate the text of a page into each separate line. The idea is to go through in a downward direction. The process starts from the top to the bottom of the page by looking for the white-space between each line. Once two horizontal white-spaces are found, the black lines in between is considered to be one line. The process ends whenever the last line of the page is reached.

3.2. Text Band Calculation and Character Separation

3.2.1. Terminologies

- **Text Band**: the top and bottom margins of the Main Bodies are called Text Band. Text Band comprises of Start of the Text Band and End of the Text Band. The following example depicts Start of the Text Band and End of the Text Band.

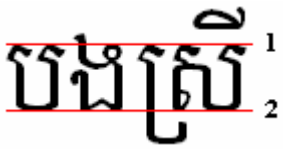


Figure 3.1 *Text Band, 1 - Start of the Text Band, 2 - End of the Text Band*

- **Character**: each vertically separated component is called a character. A character may comprise of many individual letters.

Figure 3.2 *Character*

3.2.2. Process

Prior to the Character separation, Text Band has been calculated for the whole line. For the Start of the Text Band, we take the average which is the result of total top positions of each Character divided by total numbers of Character per line. Not different from Start of the Text Band, the End of Text Band is the result of total bottom positions of each Character divided by total numbers of Character of that line.

Character Segmentation is the process of getting each separable Character from the sentence. Characters are seen as separable by the vertical white-space between them.

Figure 3.3 *Separable characters*

3.3. Text Band Adjustment and Segmentation

3.3.1. Terminologies

Some terminologies have been used throughout our research and development.

- **Main Body**: any body appears within the Text Band is the Main Body. It can be a consonant, an independent vowel, or a dependent vowel.

Figure 3.4 *Main Bodies*

From left to right – Consonant, Independent vowel, Dependent vowel

- **SuperScript**: any body appears above the Main Body is called SuperScript. It is usually a dependent vowel. However, a part of a dependent vowel can also be a SuperScript. Furthermore, double-quotes are also considered as the SuperScript due to their positions, i.e. above the Text Band.

Figure 3.5 *SuperScript*

- **SubScript**: the isolated black body that appears below the Main Body is called SubScript. It can be a dependent vowel or the smaller version of the consonant.

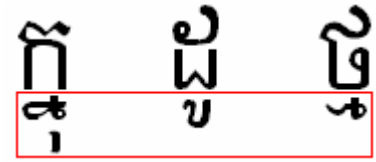


Figure 3.6 *SubScript*

- **CCDown**: it is an overlapping identity that covers both Main Body and SubScript.

Figure 3.7 *CCDown*

- **CC**: it is an overlapping identity that covers from SubScript to SuperScript. CC is the term used to identify the Complex Character.

Figure 3.8 *CC*

- **Width Intensity:** the total number of black pixels per vertical line is called Width Intensity.

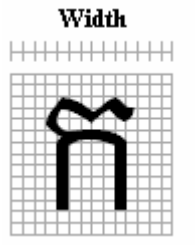


Figure 3.9 *Width Intensity*

- **Height Intensity:** the total number of black pixels per horizontal line is called Height Intensity.

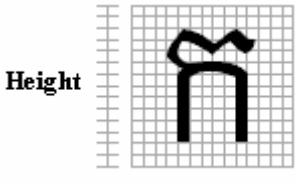


Figure 3.10 *Height Intensity*

3.3.2. Process

Text Band adjustment is needed before each character is separated into atomic components. The idea of adjusting the Text Band is to guarantee that every individual Character takes up its own Text Band more accurately.

In the Segmentation module, six crucial steps are taken.

3.3.2.1. CCDown

3.3.2.1.1. Detection Methodology

In this step, there are three processes. Initially, we find number of bodies and their corresponding positions. Secondly, we detect CCDown, get number of Main Bodies, and store the correct information about the whole Character. Finally, we store information about position of each Main Body.

3.3.2.1.2. Algorithmic Approaches

First, we find the Width Intensity of the Character with the height from Start of the Text Band to End of the Text Band. Then, we find the number of bodies by looking for any component that is in between two vertical white-spaces, while at the same time storing the starting position and the ending position of each body. After that, CCDown is to be found according to the number of bodies. If there is one body, we assume that there is no CCDown because CCDown will never appear alone in the sentence. If there is more than one body, we continue looking for CCDown by finding the Height Intensity of each body from top to the bottom, and continue looking from the End of Text Band till the end of the Character. As the result, we will assume that the body is a CCDown when it has the straight series of more than 10 black pixels attached to the bottom of it because less than 11 pixels is considered to be a SubScript or nothing below the Main Body.

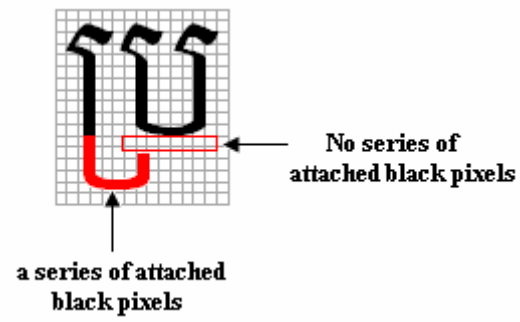


Figure 3.11 *CCDown Detection Algorithm Approaches*

3.3.2.2. CC

3.3.2.2.1. Detection Methodology

CC Detection can only be done when the information about the whole Character contains a CCDown. The important thing is that CC will only have its position on the right side of the whole Character. Therefore, there are two cases for detecting the CC.

1. **First case (Two bodies present):**

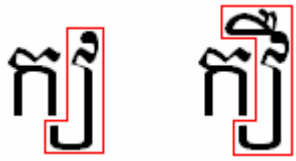


Figure 3.12 *CC, first case (Two bodies present)*

2. **Second case (Three bodies present):**

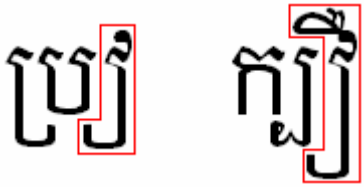


Figure 3.13 *CC, second case (Three bodies present)*

3.3.2.2.2. Algorithm Approaches

As above mentioned, CC should be detected only when there is a CCDown attached to right side of the Character. If this is true, we find the Height Intensity of that CCDown. Then, within the CCDown position, we continue finding if there are series of black pixels attached to the top of it. Finally, CCDown would be automatically identified as a CC if there are the series of black pixels present.

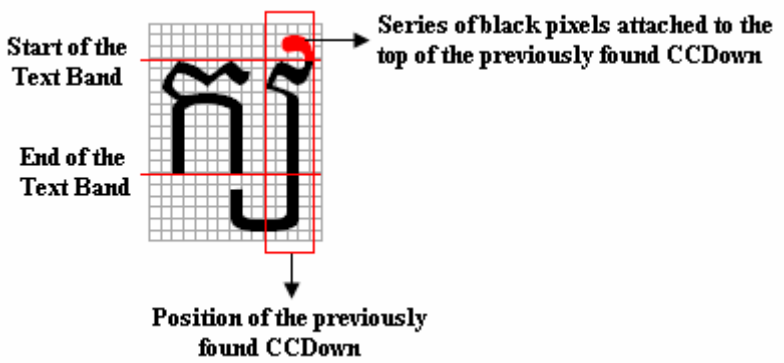


Figure 3.14 *CC Detection Algorithm Approaches*

3.3.2.3. SuperScript

3.3.2.3.1. Detection & Extraction Methodology

SuperScript detection works only when there is no CC presents because the writing will not be accepted when both SuperScript and CC present at the same time.

There are four main cases in the SuperScript Detection.

1. **Case 1 (No single body presents)**

This is the special case when Khmer Double Quote presents. Since the position of this shape is above the Start of the Text Band, it is considered as the SuperScript in accordance with the defined terminology (See section 3.3.1 Terminologies, SuperScript).

Figure 3.15 *SuperScript, first case (No single body presents)*

2. **Case 2 (One body presents)**

In this case, we count up the separated components in the vertical direction. Therefore, we divide it into three sub-cases:

Figure 3.16 *SuperScript, Second case (One body presents), Sub-case #1 (One component found)*

Figure 3.17 *SuperScript, Second case (One body presents), Sub-case #2 (Two components found)*

Figure 3.18 *SuperScript, Second case (One body presents), Sub-case #3 (Three components found)*

3. **Case 3 (Two bodies present)**

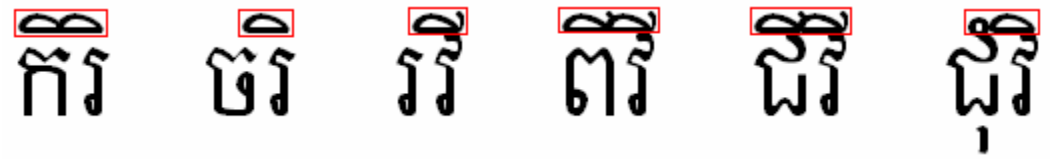


Figure 3.19 *SuperScript, Third case (Two bodies present)*

4. Case 4 (Three bodies present)

In this case, SuperScripts are detected in distinctive ways according to the Character information. Therefore, we divide the case into three sub-cases.

- Sub-case #1 (MMM [M: Main Body])

  In this case, the SuperScripts are detected by checking for the white-space between the Main Body and themselves. From left to right, we try to detect the white-space between each SuperScript and its corresponding Main Body. Once the white-space is found, the first black line to the top of the image after the white-space will be the bottom of all the SuperScripts.

Figure 3.20 *SuperScript, Fourth case (Three bodies present), Sub-case #1 (MMM)*

- Sub-case #2 (cMM [c: CCDown, M: Main Body])

  In this case, SuperScript detections can be done by using simple technique, i.e. downwardly checking from the top for the black component between two white-spaces. The black component found will be the SuperScript. The only special case that may present is the parallel SuperScripts which means there are more than one SuperScript presents on the same level.

Figure 3.21 *SuperScript, Fourth case (Three bodies present), Sub-case #2 (cMM), No parallel SuperScripts*

Figure 3.22 *SuperScript, Fourth case (Three bodies present), Sub-case #2 (cMM), Parallel SuperScripts*

- Sub-case #3 (McM [c: CCDown, M: Main Body])

  In this case, we start looking for the top and the bottom of the SuperScript with the assumption that it resides on both the second and the third body. Then, we will judge if there is a SuperScript depending on its bottom position. Therefore, two possibilities may occur.

  - One possibility is that the bottom position of the SuperScript is above the Start of the Text Band which means there must be a SuperScript presents. Then, we start counting if there are parallel SuperScripts.

Figure 3.23 *SuperScript, Fourth case (Three bodies present), Sub-case #3 (McM), Possibility #1 (No Parallel SuperScripts)*

Figure 3.24 *SuperScript, Fourth case (Three bodies present), Sub-case #3 (McM), Possibility #1 (Parallel SuperScripts)*

  - Another possibility is that there is a SuperScript attached to the third body or none is presents.

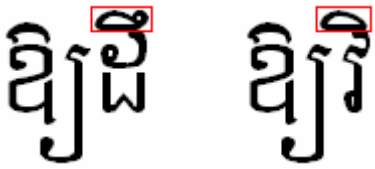


Figure 3.25 *SuperScript, Fourth case (Three bodies present), Sub-case #3 (McM),*

Possibility #2 (SuperScript and the Thrid Main Body are attached)

Figure 3.26 *SuperScript, Fourth case (Three bodies present), Sub-case #3 (McM), Possibility #2 (No SuperScript presents)*

3.3.2.3.2. Algorithm Approaches

SuperScript detection is done in accordance with number of bodies. Thus, there are two main cases for this approach.

1. **Case 1 (No single body presents)**

This case solely applies to Khmer Double Quote which by definition should be a SuperScript because of its position (above the Start of the Text Band). The extraction can be simply done by, from top to the bottom of the image, finding the black component between two white-spaces. The first black line will be the top position, and the last black line will be the bottom position. Likewise, from left to right, the black component between two white-spaces is being sought. Hence, the first vertical black line will be the left position, and the last vertical black line will be the right position.

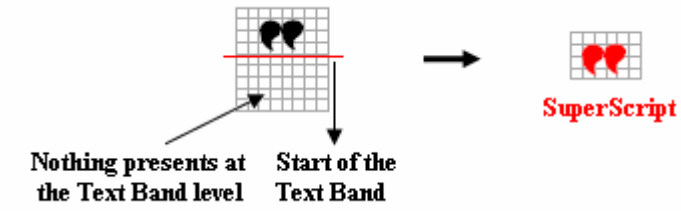


Figure 3.27 *SuperScript Detection, Case 1 (No single body presents), Open Double Quote*

In Khmer font (Limon S1), there are two Double Quotes such as open and close Double Quotes. So, the extraction of both will be applied using the same technique as illustrated above.

2. **Case 2 (One body presents)**

In this case, we initially start horizontally looking for how many separated black components there are from the top of the image to where the middle of the Text Band is. Then, there are three sub-cases as the followings:

- Case 2, sub-case 1 (One separated black component counted)

  Two possibilities may occur in this case.

  - One possibility is that there is no SuperScript.

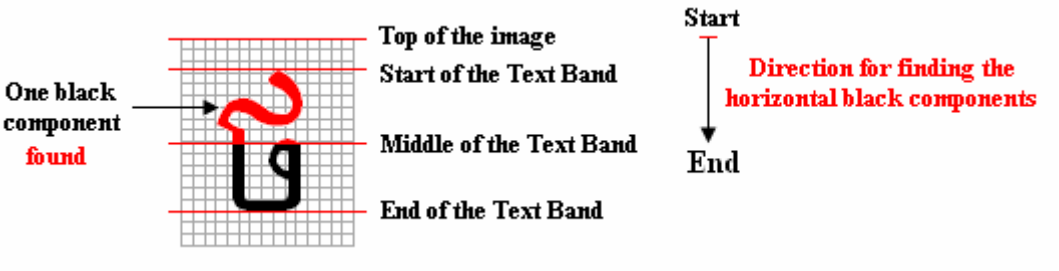


Figure 3.28 *SuperScript Detection, Case 2, Sub-case 1 (No SuperScript presents)*

  - The other possibility is that SuperScript and Main Body are either connected by noise or overlapping pixel. Components connected by noise are automatically separated when there are at most two pixels noise. However, for overlapping components there are two attempts for this. The first attempt is to check for the SuperScript that has the total width equal to or less than 1 or 2 pixels of that of the Main Body. If found, the pointer is moving downwardly through the image to find the minimum point which will be the bottom of the SuperScript. If the first attempt failed, the second attempt would be addressed by checking for the SuperScript that has the total width less than 75% of the Main Body.

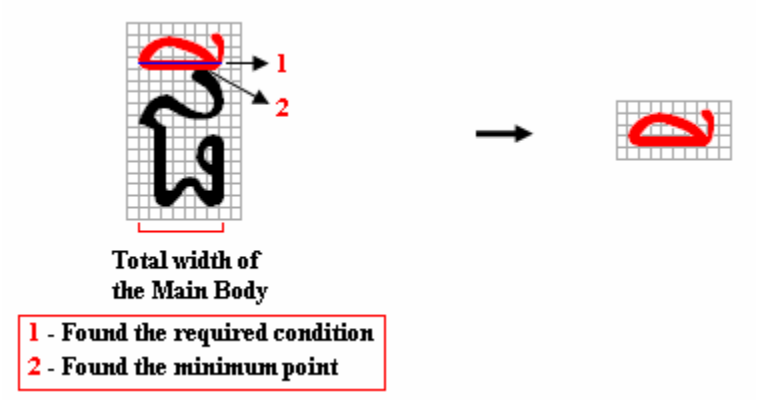


Figure 3.29 *SuperScript Detection, Case 2, Sub-case 1 (SuperScript and Main Body are attached #1)*

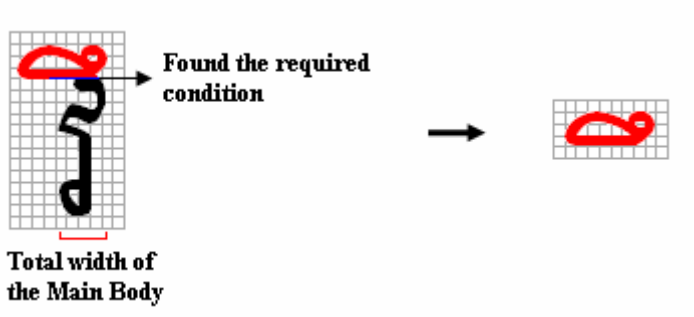


Figure 3.30 *SuperScript Detection, Case 2, Sub-case 1*
(SuperScript and Main Body are attached #2)

- Case 1, sub-case 2 (Two separated black components counted)
  In this sub-case, the SuperScript and the Main Body are obviously separated.

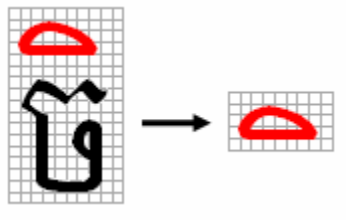


Figure 3.31 *SuperScript Detection, Case 2, Sub-case 2*
(SuperScript and Main Body are clearly separated)

- Case 1, sub-case 3 (Three separated black components counted)
  In this case, there are two SuperScripts present and they are clearly separated.

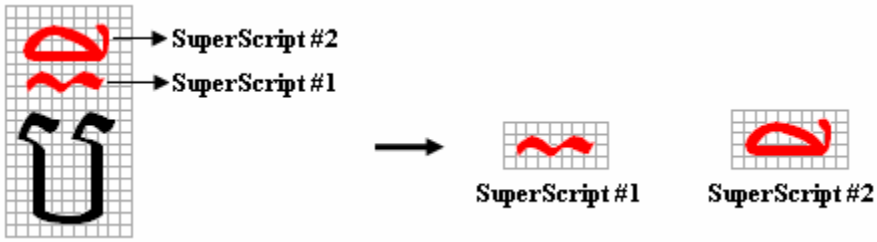


Figure 3.32 *SuperScript Detection, Case 2, Sub-case 3*
(SuperScript #1 and SuperScript #2 present)

3. Case 3 (Two bodies present)

In this case, we divide it into two sub-cases. The first sub-case is concerned with any Character that has only one Main Body. The second sub-case is concerned with any Character that has two Main Bodies.

- Case 3, Sub-case 1 (One Main Body presents)

In this case, there must be a CCDown which is either on the left side or the right side of the Main Body. Hence, two smaller cases are being addressed.

- If there is a CCDown on the left and Main Body on the right, we initially find the horizontal black components on the Main Body (right) side. Then, if there is only one black component found, we apply the same technique as described in Case #1, Sub-case 1 (One separated black component counted). However, if there are two black components found, that means the SuperScript presents without attaching to the Main Body; thus, we extract it out of the Character.

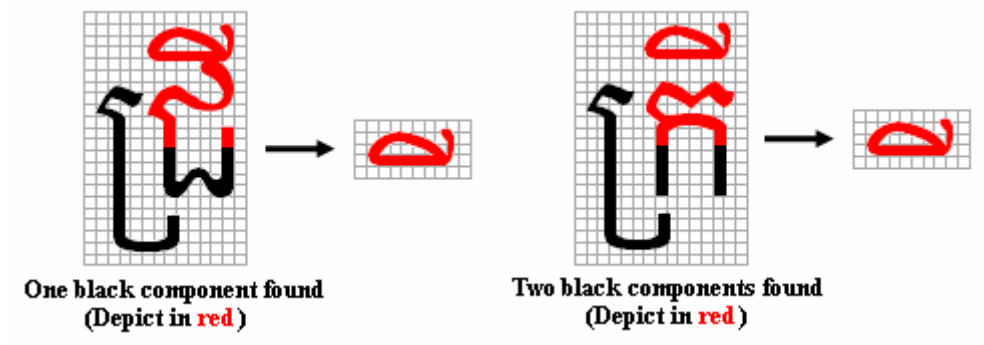


Figure 3.33 *SuperScript Detection, Case 3, Sub-case 1*
(CCDown's on the left and Main Body's on the right)

- If the CCDown is on the right and Main Body is on the left, SuperScript may reside on either the CCDown or the Main Body. In standard writing, SuperScript resides only on the CCDown in this case but since some typists never care about this and they are the majority, such the case as SuperScript resides on the Main Body side will inevitably occur. Therefore, both of the cases are taken into this study to guarantee that every possible case is discovered.

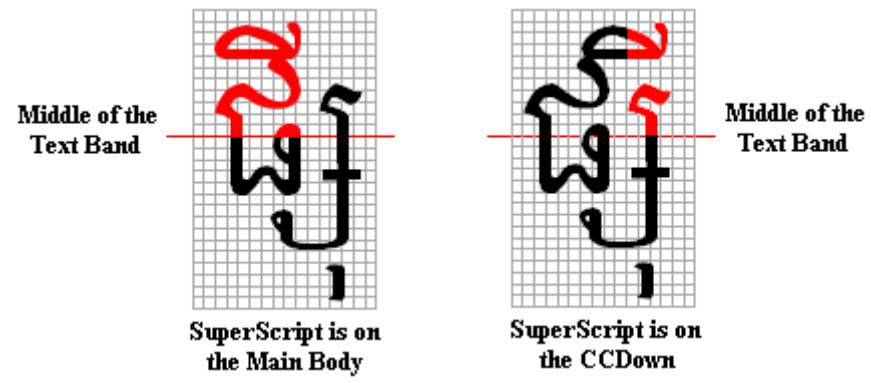


Figure 3.34 *SuperScript Detection, Case 3, Sub-case 1*

(Main Body's on the left and CCDown's on the right)

- Case 3, Sub-case 2 (Two Main Bodies present)

In this case, the only benchmark for extracting the SuperScript is to find the white-space between SuperScript and Main Body. From left to right, Main Body1 is being checked first followed by the Main Body2. Then, different techniques will be applied for different cases.

- If there is no white-space on both Main Bodies, two possible cases can occur. SuperScript may or may not present in this case. So, to check up whether there is a SuperScript, we start looking for black pixels in between the two bodies from the middle of Text Band up until the top of the image. If the white pixels line we reach is the top of the image, that means there is no SuperScript presents; otherwise, the SuperScript is extracted from that line up.

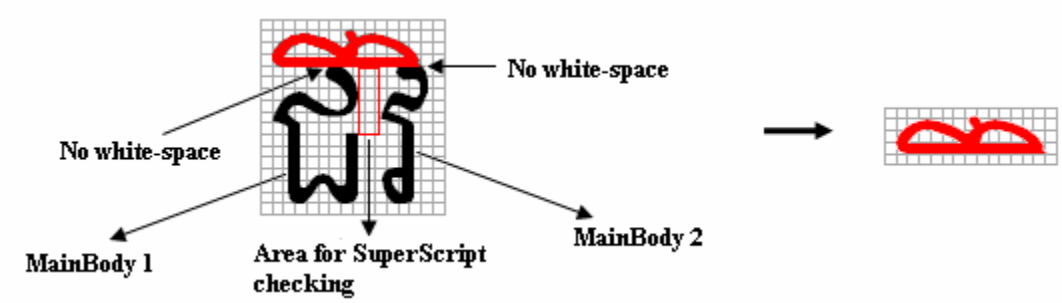


Figure 3.35 *SuperScript Detection, Case 3, Sub-case 2 (Two Main Bodies present), No white-space between Main Bodies*

- If there are white-spaces on both Main Bodies, the bottom of the SuperScript will be the first black line from the white-space above the second Main Body. And the extraction methodology will be done from this line up to the top of the image. Moreover, SuperScript may be separated into two if there is a vertical white-space between these two SuperScripts.

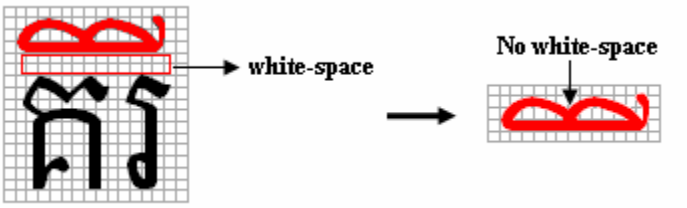


Figure 3.36 *SuperScript Detection, Case 3, Sub-case 2 (Two Main Bodies present), White-spaces between Main Bodies, SuperScript are inseparable*

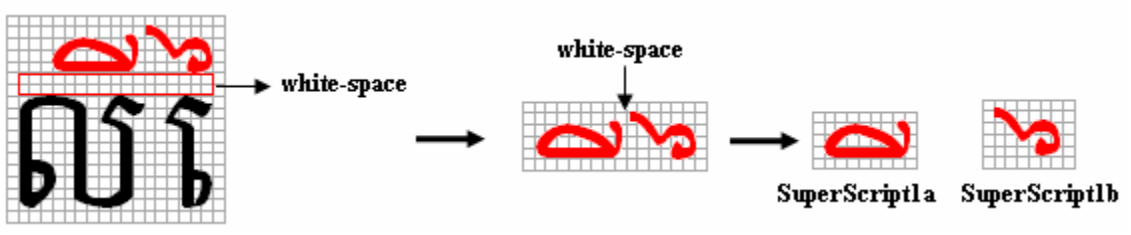


Figure 3.37 *SuperScript Detection, Case 3, Sub-case 2 (Two Main Bodies present), White-spaces between Main Bodies, SuperScripts are separable*

- If there is a white-space either on the Main Body1 or Main Body2, the extraction will be done on the basis of where the white-space is.

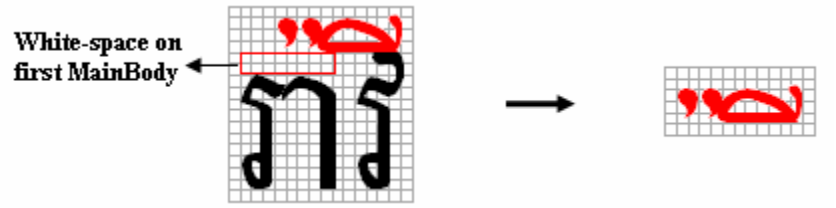


Figure 3.38 *SuperScript Detection, Case 3, Sub-case 2 (Two Main Bodies present), White-spaces above the first Main Body*

4. Case 4 (Three bodies present)

In this case, we divide the case into three distinctive cases. The first case deals with the Character that has three Main Bodies. The second case deals with any Character which is composed of the CCDown on the left and two Main Bodies on the right. The final case deals with the Character that is comprised of one Main Body on the left, followed by a CCDown in the middle, and one more Main Body on the right.

- Case 4, Sub-case 1 (MMM, [M: Main Body])

  In this case, we assume that of all the three bodies, there must be at least a white-space between an individual SuperScript and its corresponding Main Body. Hence, we initially check for the white-space on the first body. If it is found, we stop there and find the first black line which will be the bottom position of all the parallel SuperScripts. Then, we find the top position of the SuperScript by going from the top of the image down until the first black line is met. Likewise, in the left-to-right direction we continue finding its left and right position. However, if there is no white-space between the first Main Body and its above SuperScript, the second Main Body and its corresponding SuperScript will be checked using the same technique as the previous one.

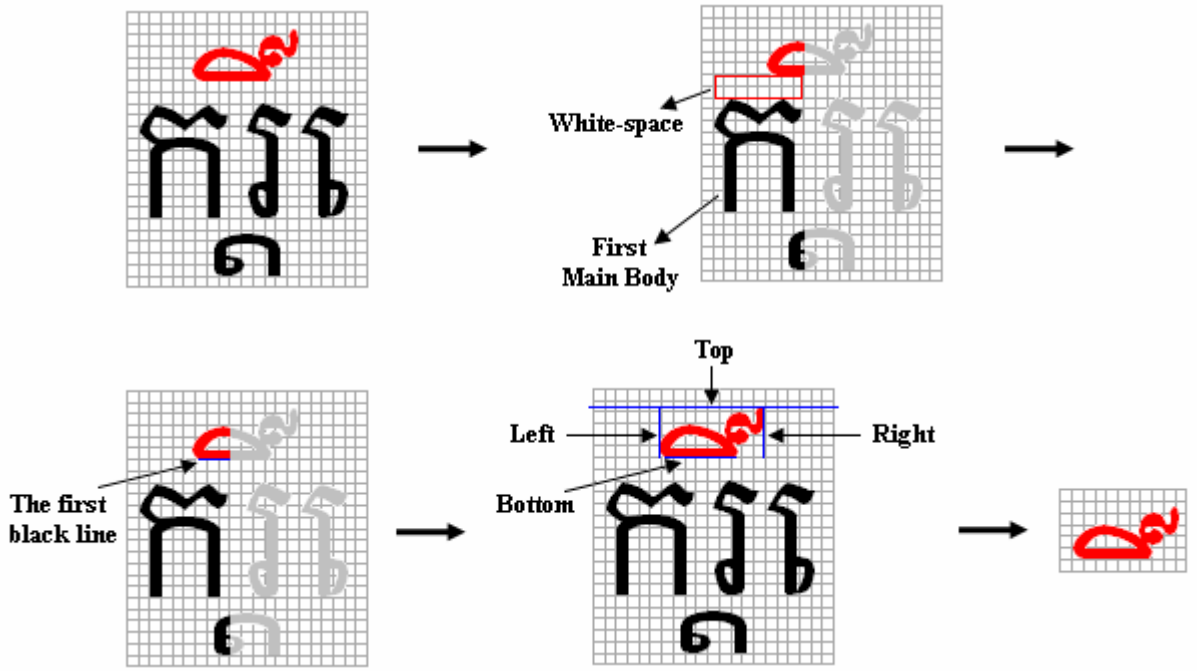


Figure 3.39 *SuperScript Detection, Case 4, Sub-case 1 (MMM),*
A white-space line detected on the first Main Body

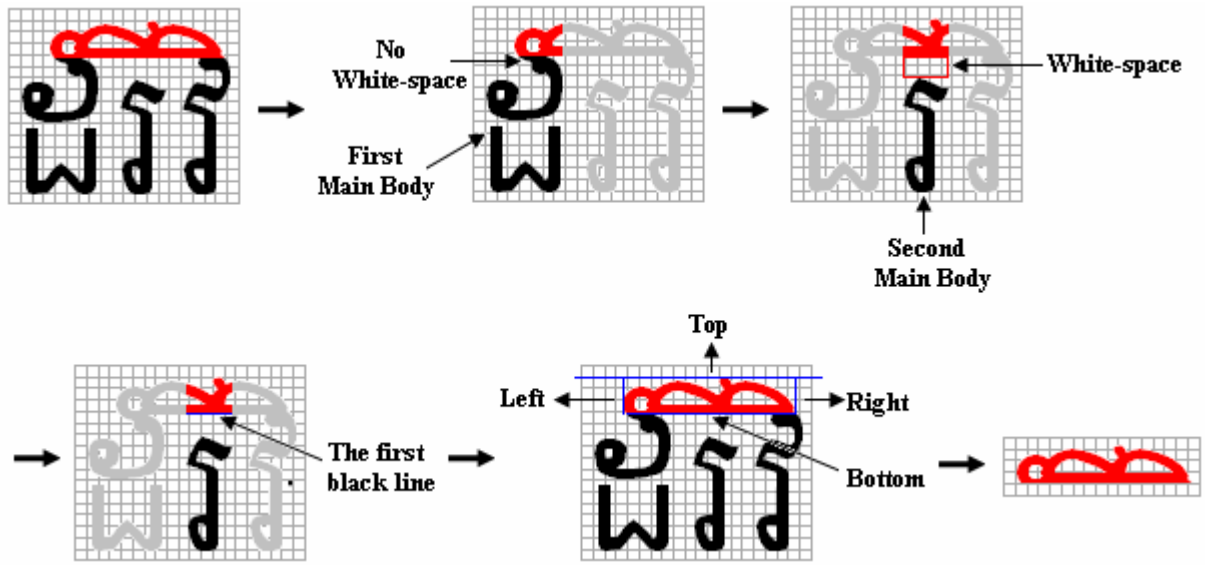


Figure 3.40 *SuperScript Detection, Case 4, Sub-case 1 (MMM),*
A white-space line detected on the second Main Body

- Case 4, Sub-case 2 (cMM, [c: CCDown, M: Main Body])

In this case, we initially find the top and bottom position of the SuperScript from the top of the image. Then, we get number of parallel SuperScripts within the new top and bottom position found. If there is only one SuperScript, we extract it by finding

the left and right position. Else, if there is more than one, we start subsequently extract each from left to right.

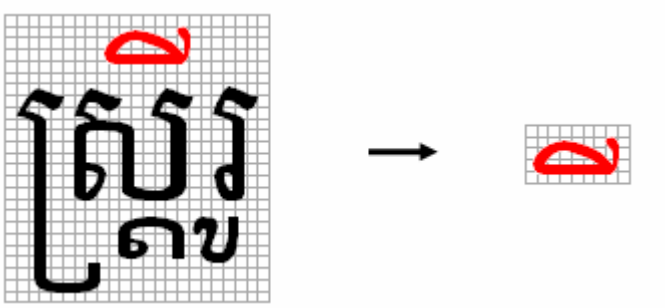


Figure 3.41 *SuperScript Detection, Case 4, Sub-case 2 (cMM), One SuperScript presents*

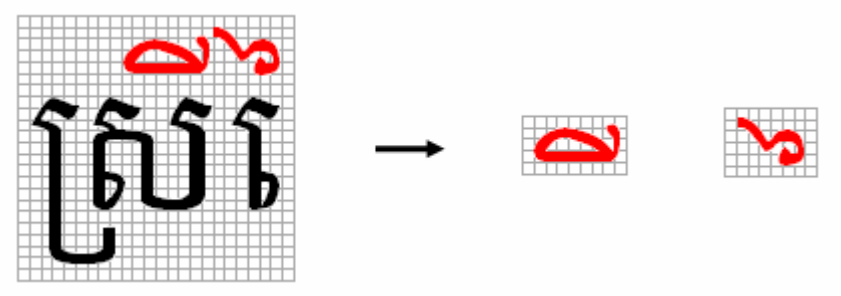


Figure 3.42 *SuperScript Detection, Case 4, Sub-case 2 (cMM), Parallel SuperScripts present*

- Case 4, Sub-case 3 (McM, [c: CCDown, M: Main Body])

This case may contain complexity; however, we start detecting the SuperScript with the simplest technique. In other words, we start finding the top and the bottom position of it. After that, we compare the bottom position with the Start of the Text Band. Then, there are two cases concerning this comparison. One case occurs when the bottom position is less than or equal to the Start of the Text Band. The other case occurs when the bottom position is greater than the Start of the Text Band.

- *Case 4, Sub-case 3.1 (The bottom position <= the Start of the Text Band)*

  In this case, we initially start counting the number of parallel SuperScripts. If there is only one SuperScript, we extract it. Otherwise, the very left SuperScript needs to be checked for genuine identity since in some Characters it is just a part of the first Main Body.

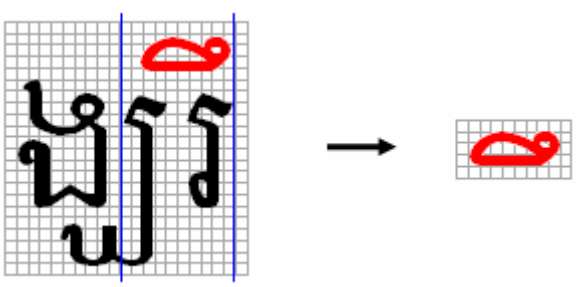


Figure 3.43 *SuperScript Detection, Case 4, Sub-case 3.1 (SuperScript and Main Body are not attached), No Parallel SuperScripts present*

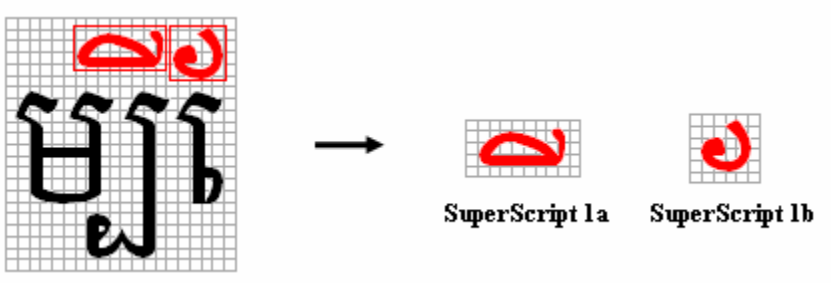


Figure 3.44 *SuperScript Detection, Case 4, Sub-case 3.1 (SuperScript and Main Body are not attached), Parallel SuperScripts present*

- *Case 4, Sub-case 3.2 (The bottom position > the Start of the Text Band)*

  In this case, the very left SuperScript may or may not be a real SuperScript. Hence, we check it for verification by looking for the white-space below it. If there is a white-space, it is a genuine SuperScript; otherwise, it is not because in this case the first SuperScript and the Main Body below cannot attach.

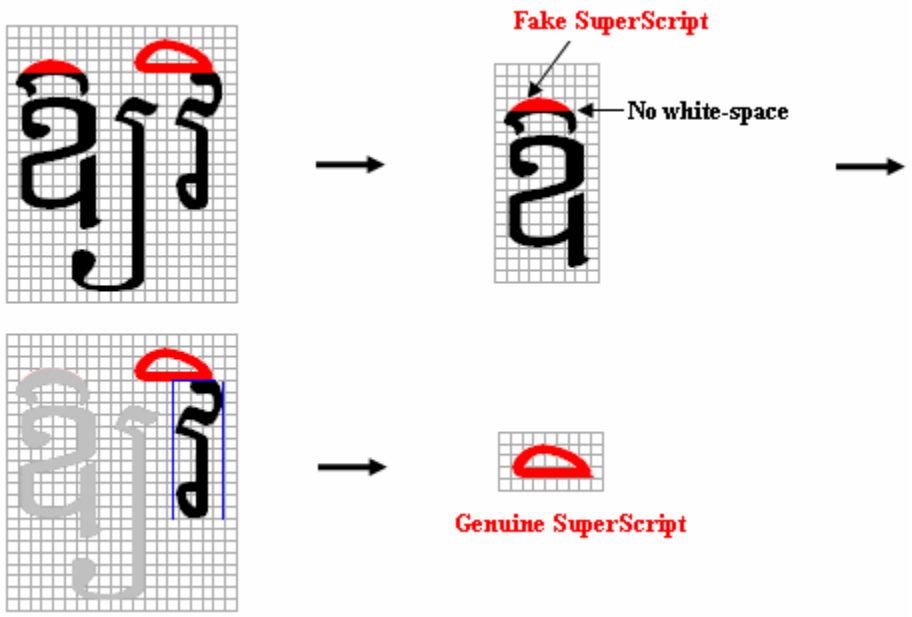


Figure 3.45 *SuperScript Detection, Case 4, Sub-case 3.2 (SuperScript and Main Body are attached), Parallel SuperScripts found but one is fake*

3.3.2.4. SubScript

3.3.2.4.1. Detection & Extraction Methodology

SubScript detection is done in three main cases. The first case is concerned with any Character that has only one body. The second case is concerned with any Character that has two bodies. And the last case is concerned with any Character that is composed of three bodies.

1. **Case 1 (One body presents)**

In this case, we initially count the horizontal black components from the middle of Text Band to the bottom of the image. Hence, three sub-cases have been addressed.

- Case 1, Sub-case 1 (one black component found)

  This case means that there is no SubScript presents in the Character because one black component found is a part of the Main Body, not the SubScript.

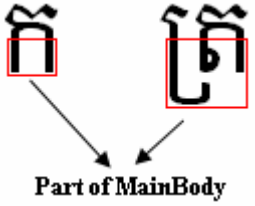


Figure 3.46 *SubScript Detection, Case 1, Sub-case 1 (One black component found), Neither SubScript1 nor SubScript2 presents*

- Case 1, Sub-case 2 (Two black components found)

  In this case, only one SubScript presents in the Character. One component is the part of the Main Body and the other is the SubScript1.

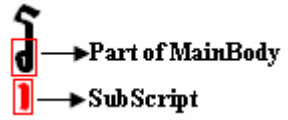


Figure 3.47 *SubScript Detection, Case 1, Sub-case 2 (Two black components found), Only SubScript1 presents*

- Case 1, Sub-case 3 (three black components found)

  This is the case when there are two SubScripts present.

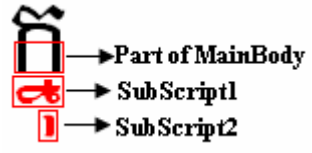


Figure 3.48 *SubScript Detection, Case 1, Sub-case 3 (Three black components found), Two SubScripts present*

2. Case 2 (Two bodies present)

In this case, we divide it into three sub-cases.

- Case 2, Sub-case 1 (Main Body & Main Body)

  In this case, the Character consists of two Main Bodies. By counting the horizontal black components from the middle of Text Band down to the bottom of the image, three mini-cases are being addressed.

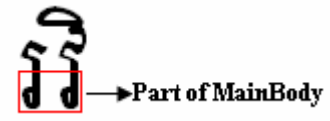


Figure 3.49 *SubScript Detection, Case 2 (Two bodies present), Sub-case 1.1 (One black component found)*

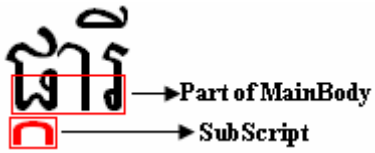


Figure 3.50 *SubScript Detection, Case 2 (Two bodies present), Sub-case 1.2 (Two black components found)*

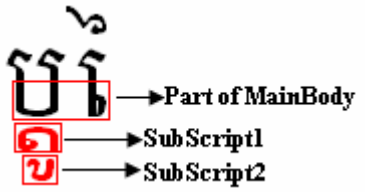


Figure 3.51 *SubScript Detection, Case 2 (Two bodies present), Sub-case 1.3 (Three black components found)*

- Case 2, Sub-case 2 (Main Body & CCDown)

In this case, we initially get the number of horizontal black components from the Character. Then, we divide the case into two mini-cases.

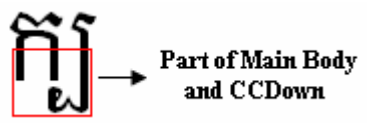


Figure 3.52 *SubScript Detection, Case 2 (Two bodies present), Sub-case 2.1 (One black component found)*

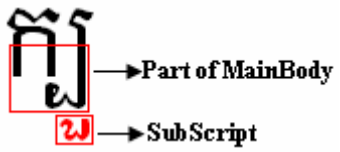


Figure 3.53 *SubScript Detection, Case 2 (Two bodies present), Sub-case 2.2 (Two black components found)*

- Case 2, Sub-case 3 (CCDown & Main Body)

In this case, horizontal black components are also counted. Thus, two mini-cases are being addressed.

- *Case 2, Sub-case 3.1 (One horizontal black component found)*

  In this case, there are two possible cases. Hence, we vertically check for the black components in the left-to-right direction. When only one black component found, we assume that there is no single SubScript presents because the found one is the part of the CCDown. In contrast, if we found two black components, we will continue counting for the black components in a horizontal direction on the second component.

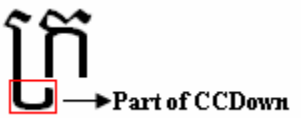


Figure 3.54 *SubScript Detection, Case 2 (Two bodies present), Sub-case 3.1 (One black component found), No SubScript*

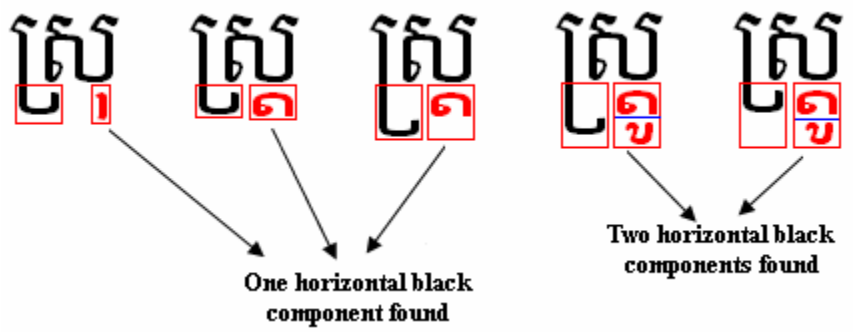


Figure 3.55 *SubScript Detection, Case 2 (Two bodies present), Sub-case 3.1 (One black components found), SubScript(s) present*

- *Case 2, Sub-case 3.2 (Two horizontal black components found)*

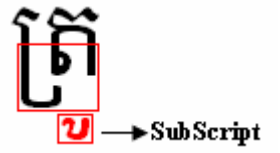


Figure 3.56 *SubScript Detection, Case 2 (Two bodies present), Sub-case 3.2 (Two black components found), One SubScript presents*

3. Case 3 (Three bodies present)

In this case, we divide it into three sub-cases.

- Case 3, Sub-case 1 (MMM, [M – Main Body])

  In this case, SubScript detection starts from below the End of the Text Band until the bottom of the image.

Figure 3.57 *SubScript Detection, Case 3, Sub-case 1 (MMM)*

- *Case 3, Sub-case 2 (McM, [M – Main Body], c – CCDown)*

In this case, only the third body may have a SubScript. Hence, the checking will be done on that body.

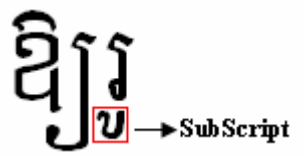


Figure 3.58 *SubScript Detection, Case 3, Sub-case 2 (McM)*

- *Case 3, Sub-case 3 (cMM, [c – CCDown , M – Main Body])*

Initially, we get the number of vertical black components from below the End of the Text Band to the bottom of the image. The number should be greater than one for extraction to be done because if the number returns one, it means that the black component found is the part of CCDown only. The following depicts two parallel SubScripts found in the Character, and the number of vertical black components is three.

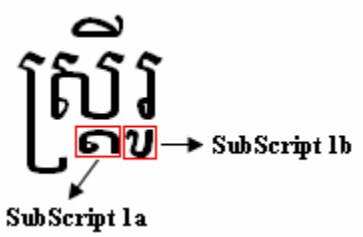


Figure 3.59 *SubScript Detection, Case 3, Sub-case 3 (cMM)*

3.3.2.4.2. Algorithm Approaches

SubScript detection is the last detection we do for the whole Character. We divide this approach into three main cases.

1. **Case 1 (One body presents)**

Initially, we start looking for horizontal black components from the middle of Text Band to the bottom of the image. So, three sub-cases have been addressed.

- Case 1, Sub-case 1 (One black component found)

In this case, we assume that there is no SubScript presents in the Character since the black component found is either the part of Main Body alone or the part of both Main Body and CCDown.

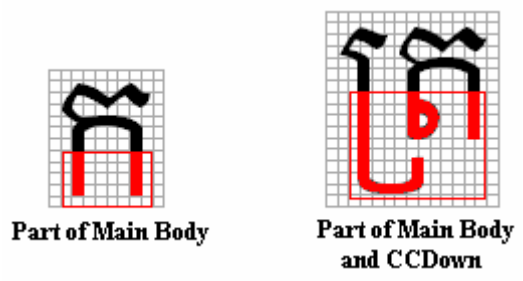


Figure 3.60 *SubScript Detection, Case 1 (One body presents), Sub-case 1 (One black component found), No SubScript presents*

- Case 1, Sub-case 2 (Two black components found)

In this case, only one SubScript is found because the first black component belongs to the Main Body, while the second is the SubScript part.

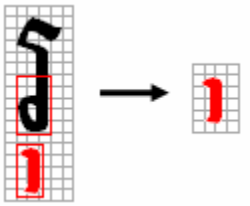


Figure 3.61 *SubScript Detection, Case 1 (One body presents), Sub-case 2 (Two black components found), One SubScript presents*

- Case 1, Sub-case 3 (Three black components found)

In this case, two SubScripts are found. SubScript2 is extracted first followed by SubScript1 at last.

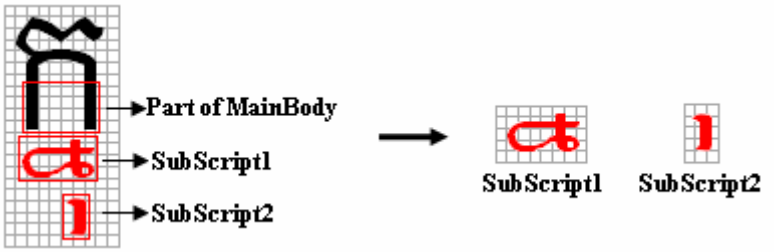


Figure 3.62 *SubScript Detection, Case 1 (One body presents), Sub-case 2 (Three black components found), Two SubScripts present*

2. Case 2 (Two bodies present)

In this case, we basically have to look at the Character information, and thus four possible sub-cases have been addressed.

- Case 2, Sub-case 1 (Main Body & Main Body)

This is the case when there are two Main Bodies in the Character. Therefore, this approach is done on the same basis as Case 1 (one body presents).

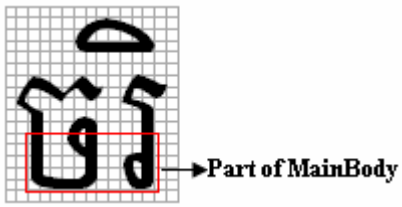


Figure 3.63 *SubScript Detection, Case 2 (Two bodies present), Sub-case 1 (Main Body & Main Body), No SubScript presents*

However, when there are two components found, there are two possibilities. One possibility is that there is only one SubScript presents. The other possibility is that two SubScripts may present in parallel, i.e. they appear on the same line.

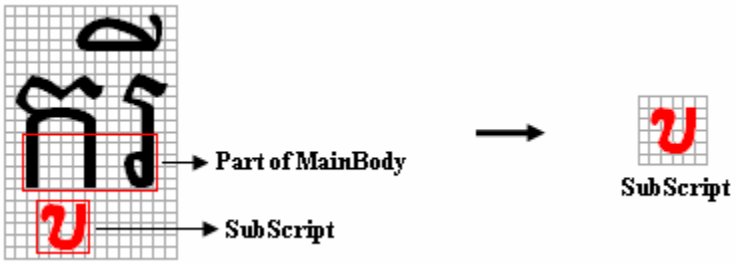


Figure 3.64 *SubScript Detection, Case 2 (Two bodies present), Sub-case 1 (Main Body & Main Body), Possibility 1 (One SubScript presents)*

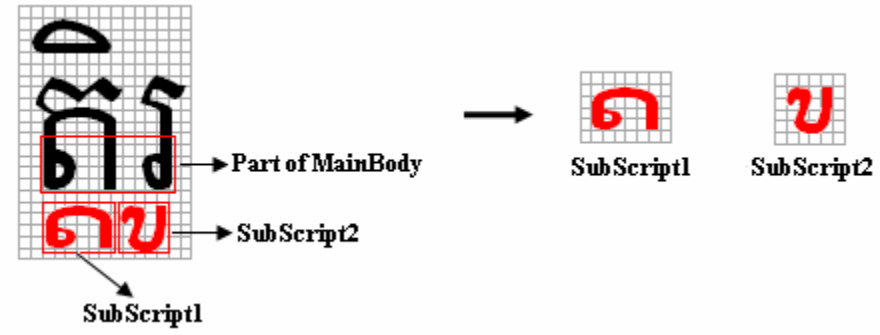


Figure 3.65 *SubScript Detection, Case 2 (Two bodies present), Sub-case 1 (Main Body & Main Body), Possibility #2 (Two SubScripts present in parallel)*

- Case 2, Sub-case 2 (Main Body & CCDown)

In this case, the Character consists of Main Body on the left and CCDown on the right. The detection approach is the same as the previous case, i.e. looking for

horizontal black components from the middle of the Text Band down to the bottom of the image. Hence, two mini-cases have been addressed.

- *Case 3, Sub-case 2.1 (One black component found)*

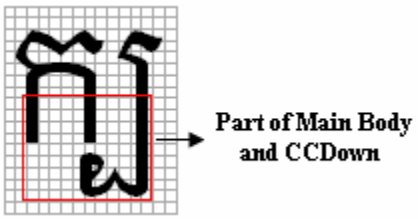


Figure 3.66 *SubScript Detection, Case 2 (Two bodies present), Sub-case 2.1 (One black component found), No SubScript presents*

- *Case 3, Sub-case 2.2 (Two black components found)*

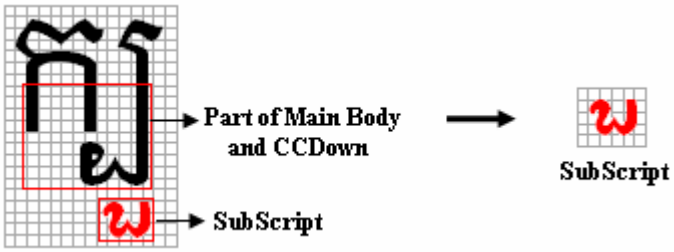


Figure 3.67 *SubScript Detection, Case 2 (Two bodies present), Sub-case 2.2 (Two black components found), One SubScript presents*

- Case 2, Sub-case 3 (CCDown & Main Body)

Not different from the previous case, we initially look for horizontal black components from the middle of Text Band down to the bottom of the image. Then, we take the number of horizontal black components as the condition. As the result, two mini-cases have been addressed.

- *Case 2, Sub-case 3.1 (One black component found)*

  In this case, there are two possibilities; therefore, looking for vertical black components from the End of Text Band down to the bottom of the image is needed. Then, if the number of vertical black component is equal to one, that means there is no SubScript presents in the Character. Otherwise, there will be one or two SubScripts present.

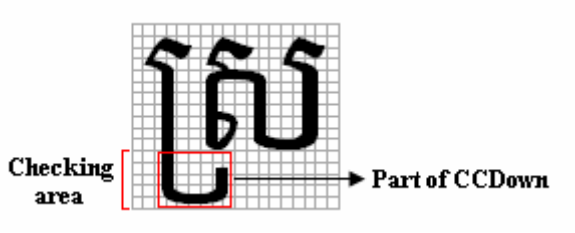


Figure 3.68 *SubScript Detection, Case 2 (Two bodies present),*
Sub-case 3.1 (One horizontal black component found), No SubScript presents

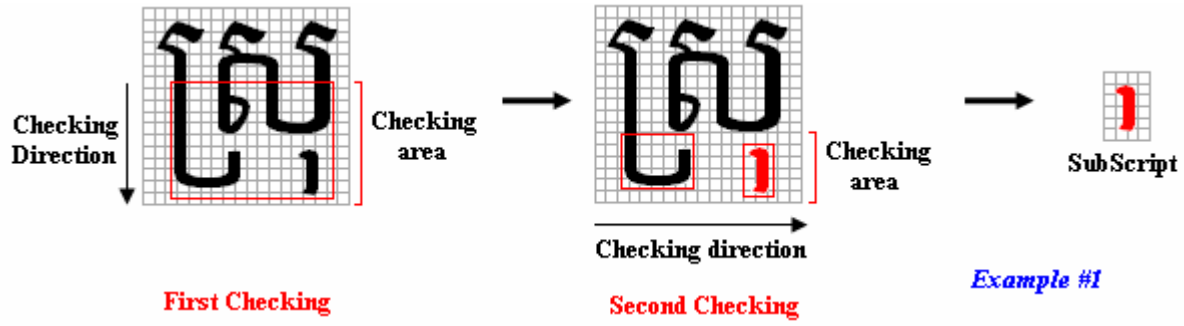


Figure 3.69 *SubScript Detection, Case 2 (Two bodies present),*
Sub-case 3.1 (One horizontal black component found), One SubScript presents

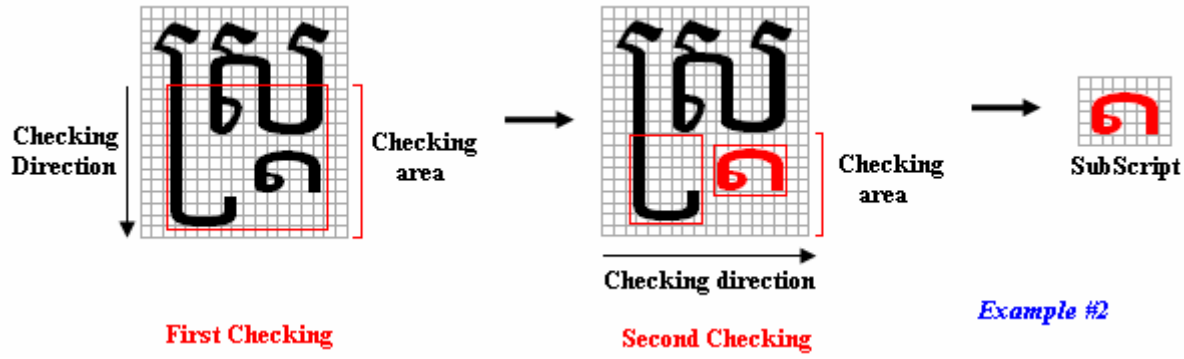


Figure 3.70 *SubScript Detection, Case 2 (Two bodies present),*
Sub-case 3.1 (One horizontal black component found), One SubScript presents

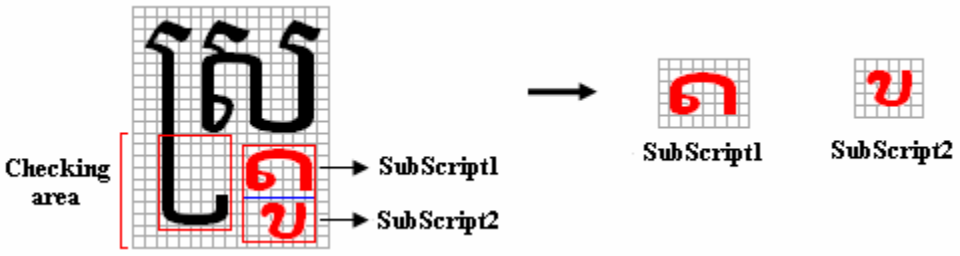


Figure 3.71 *SubScript Detection, Case 2 (Two bodies present),*

Sub-case 3.1 (One horizontal black component found), Two SubScripts present

- *Case 2, Sub-case 3.2 (Two black components found)*

  This case occurs when there is a SubScript presents below the CCDown.

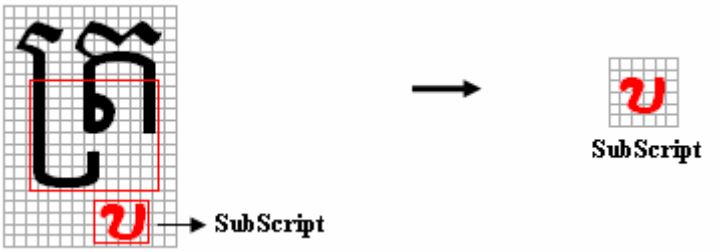


Figure 3.72 *SubScript Detection, Case 2 (Two bodies present), Sub-case 3.2 (Two horizontal black components found), One SubScript presents*

3. Case 3 (Three bodies present)

In this case, SubScript can occur in three divergent cases.

- Case 3, Sub-case 1 (MMM, [M – Main Body])

  In this case, we start detecting the SubScript from below the End of the Text Band down until the bottom of the image. The technique is to find the black component between two white-spaces.

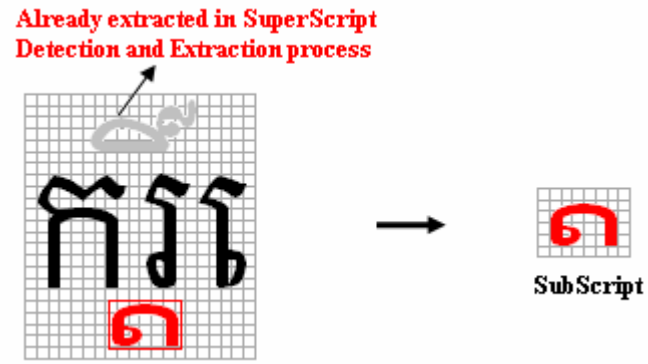


Figure 3.73 *SubScript Detection, Case 3 (Three bodies present), Sub-case 1 (MMM)*

- Case 3, Sub-case 2 (McM, [c – CCDown, M – Main Body])

  In this case, the only possible SubScript will present is the one that resides below the third body. Therefore, we check for the SubScript within the specific boundary, i.e. from the right boundary of the second body to the right boundary of the third body. Once the top and the bottom position are found, the right position will be detected, and then the left position at last.

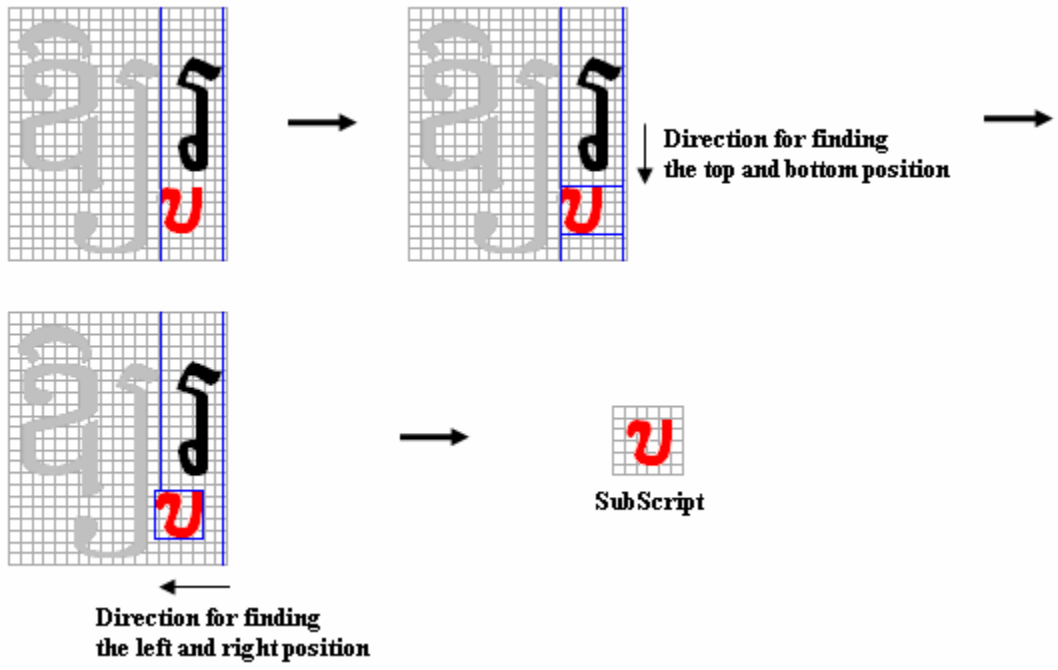


Figure 3.74 *SubScript Detection, Case 3 (Three bodies present), Sub-case 2 (McM)*

- Case 3, Sub-case 3 (cMM, [c – CCDown, M – Main Body])

  In this case, we initially get the number of parallel SubScript below the End of the Text Band. Then, the detection and extraction processes will be done on the basis of that number. If we get one, there is no SubScript since that is the part of the CCDown. If we get two, there is one SubScript. If we get three, there are two SubScripts. Particularly, in this case we detect and extract the SubScript from right to left which is a far cry from the other processes.

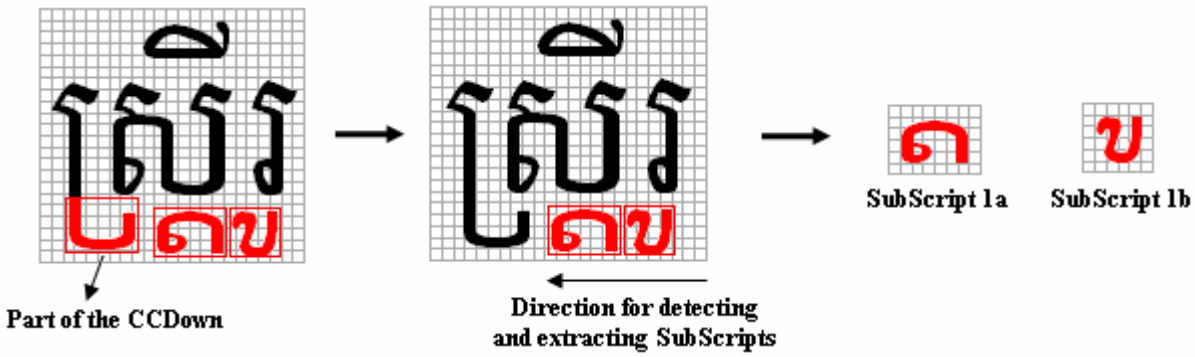


Figure 3.75 *SubScript Detection, Case 3 (Three bodies present), Sub-case 3 (cMM), Three parallel SubScripts detected, one is part of the CCDown*

3.3.2.5. Main Body

3.3.2.5.1. Extraction Methodology

Main Body extraction is done on the basis of two cases. The first case deals with any Character that consists of less than three bodies, while the second case deals with any Character which has three bodies.

1. Case 1 (Less than three bodies)

Since the Character that has two bodies can be the combination of various bodies such the combination of Main Body and Main Body, Main Body and CC, CCDown and Main Body, etc., we have found some common characteristics, and divide it into two sub-cases. The first sub-case deals alone with any Character that consists of Main Body and CC. The rest will be dealt with in the second sub-case.

- Case 1, Sub-case 1 (Main Body and CC)

  The idea starts with the checking of white-space between the Main Body and the CC. Then, if we reach the top of the image, we can guarantee that the Main Body and the CC don't have any overlapping black pixel above them. In this case, we extract the Main Body out of the Character with our ex-technique, i.e. find the top of the Main Body from the top of the Character, and continue to find the bottom of the Main Body in a downward direction. On the other hand, if we don't reach the top of the image regardless of whether white-spaces are detected, we are sure that Main Body and the CC have some overlapping black pixels above them. Therefore, Main Body is extracted using two techniques. The first technique is dealt with the case when the Main Body and the CC are connected. In this case, the Main Body extraction is done on the basis of finding horizontal white-spaces between the two bodies in an upward direction until where the overlapping pixels are, and that is the top of the Main Body. The second technique is applied to the case when the Main Body and the CC are not connected notwithstanding there are some overlapping black pixels above them. In this case, we start downwardly finding the black line from below the overlapping pixels. The first black pixel line will be the top of the Main Body.

  The following possible cases will be detected:

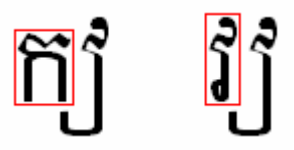


Figure 3.76 *Main Body Extraction, Case 1, Sub-case 1 (Main Body and CC), no overlapping black pixels above the two bodies*

Figure 3.77 *Main Body Extraction, Case 1, Sub-case 1 (Main Body and CC), overlapping black pixels present, Main Body and CC are connected*

Figure 3.78 *Main Body Extraction, Case 1, Sub-case 1 (Main Body and CC), overlapping black pixels present, Main Body and CC are not connected*

- Case 1, Sub-case 2 (Other cases)

  In this case, the Main Body extraction is done on the basis of information from the Main Body Indexes (a global variable used in the application), and thus each Main Body will be subsequently extracted.

Figure 3.79 *Main Body Extraction, Case 1, Sub-case 2 (Other cases)*

2. Case 2 (Three bodies)

This is broken into five sub-cases.

- Case 2, Sub-case 1 (MMM, [M – Main Body])

  The extraction in this case is very simple. From left to right, each Main Body is subsequently extracted. The process starts with the finding of the left and right

position. After that we will get the top and bottom position. Finally, each Main Body will be extracted according to all the positions found.

Figure 3.80 *Main Body Extraction, Case 2, Sub-case 1 (MMM)*

- Case 2, Sub-case 2 (McM, [c – CCDown, M – Main Body])

The process is similar to the previous case (Sub-case 1, MMM), but the only difference is that the second body which is a CCDown will not be extracted.

Figure 3.81 *Main Body Extraction, Case 2, Sub-case 2 (McM)*

- Case 2, Sub-case 3 (McC, [M – Main Body, c – CCDown, C – CC])

In this case, only the first body will be extracted since the rest are CCDown and CC.

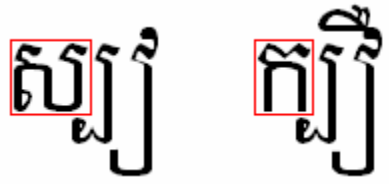


Figure 3.82 *Main Body Extraction, Case 2, Sub-case 3 (McC)*

- Case 2, Sub-case 4 (cMM, [c – CCDown, M – Main Body])

In this case, two bodies will be extracted. The first body is ignored because it is a CCDown.

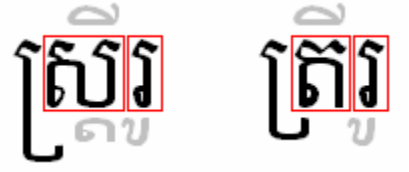


Figure 3.83 *Main Body Extraction, Case 2, Sub-case 4 (cMM)*

- Case 2, Sub-case 5 (cMC, [c – CCDown, M – Main Body, C – CC])

In this case, the second body which is the Main Body will be extracted.

Figure 3.84 *Main Body Extraction, Case 2, Sub-case 5 (cMC)*

3.3.2.5.2. Algorithm Approaches

1. Case 1 (Less than three bodies)

As mentioned earlier, there are two sub-cases for the first case. The first sub-case is when there are Main Body and CC. The second sub-case is any other case beside the first sub-case.

- Case 1, Sub-case 1 (Main Body and CC)

  Initially, we start finding the left and the right boundaries of the Main Body by, from left-to-right, looking for vertical black line. The first vertical black line will be the left boundary of the Main Body. Then, we continue looking for the vertical white line, and thus the first vertical white line will be the right boundary of the Main Body. After that, we get the Height Intensity of the gap between the Main Body and the CC. By starting from the Start of the Text Band up to the top of the image, we start looking for the horizontal black line. Then, there are two possible cases.

  - *Case 1, Sub-case 1.1:*

    The first possibility is when we reach the top of the image. In this case, we can conclude that Main Body and the CC don't have any overlapping pixel. Therefore, the top of the Main Body can be found by starting looking for the first horizontal black line from the top of the image. After the top of the Main Body is found, we continue looking for the first horizontal white line.

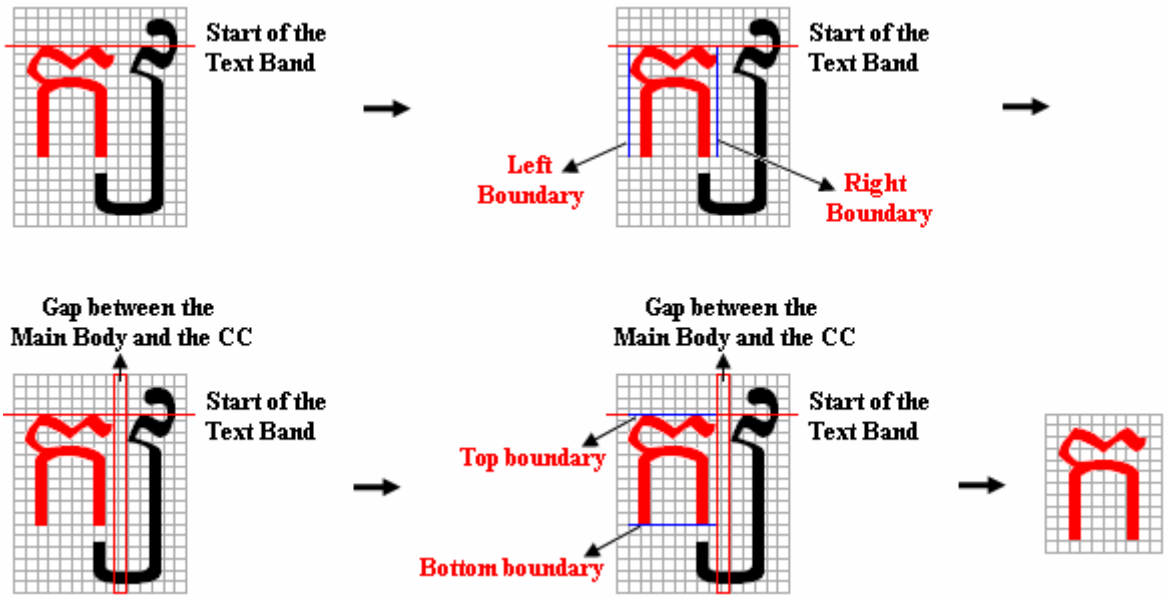


Figure 3.85 *Main Body Extraction, Case 1(Main Body and CC), Sub-case 1 (No overlapping pixels between the Main Body and the CC)*

- *Case 1, Sub-case 1.2:*

The second possibility is when we reach the horizontal black line somewhere before we reach the top of the image. That line indicates the overlapping pixels between the Main Body and the CC. The strategy is to first of all check for the white-space one pixel below the first black line we reached at the Main Body position. Therefore, two mini-cases may occur.

- o *Sub-case 1.2.1 (White-spaces found between the Main Body and the CC)*

In this case, the Main Body and the CC are not connected, i.e. there are horizontal white-spaces between the Main Body and the overlapping pixels of the CC. So, the top of the Main Body can be found by looking for the first horizontal black line from where the bottom of the overlapping line is to the End of the Text Band. Then, the bottom of the Main Body can be found by continuing looking for the first horizontal white line.

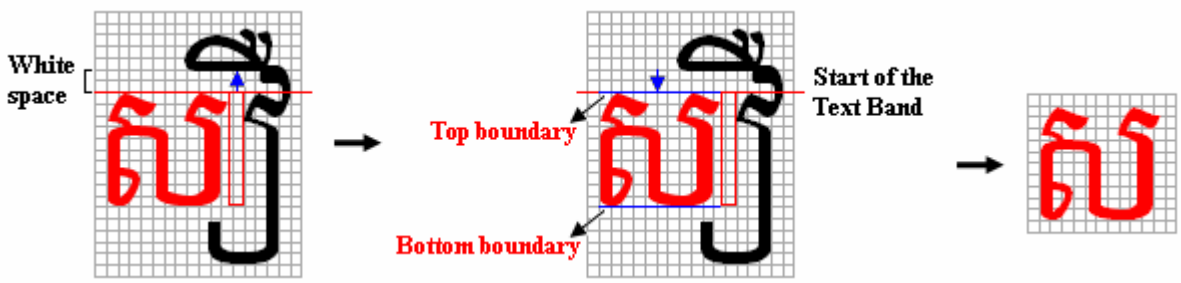


Figure 3.86 *Main Body Extraction, Case 1(Main Body and CC), Sub-case 2 (Overlapping pixels between the Main Body and the CC), Mini-case 1 (White-space between the Main Body and the CC)*

- *Sub-case 1.2.2 (No white-spaces found between the Main Body and the CC)*

In this case, the Main Body and the CC are connected, i.e. there is no white-space between them. Therefore, the line that is one pixel below the first horizontal black line of the overlapping lines will be the top of the Main Body. The bottom of the Main Body can be found by continue, from the top down to the End of Text Band, looking for the first white line.

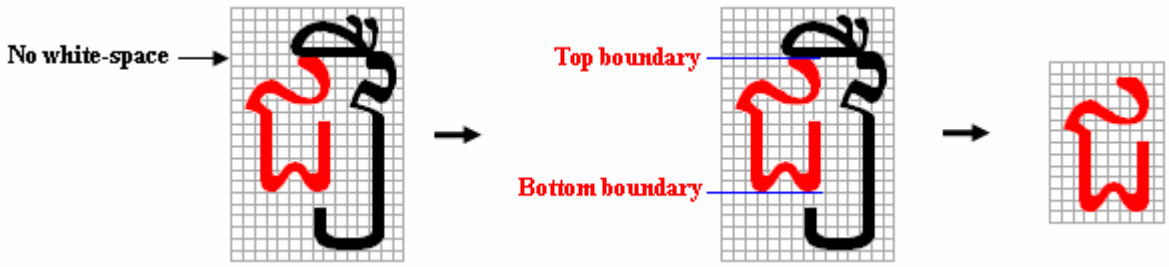


Figure 3.87 *Main Body Extraction, Case 1(Main Body and CC), Sub-case 2 (Overlapping pixels between the Main Body and the CC), Mini-case 2 (No White-space between the Main Body and the CC)*

- Case 1, Sub-case 2 (Other cases)

In this case, the extraction will be done on the basis of number of Main Bodies. If there is more than one Main Body, each will be subsequently extracted from left to right. The followings depict the extraction process.

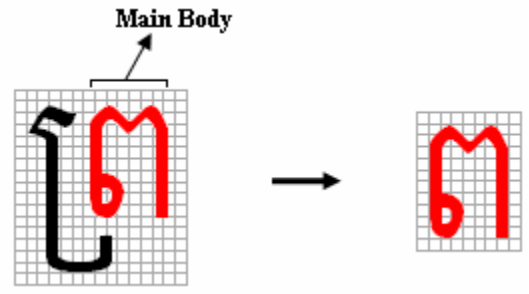


Figure 3.88 *Main Body Extraction, Case 1, Sub-case 2 (Other cases), One Main Body*

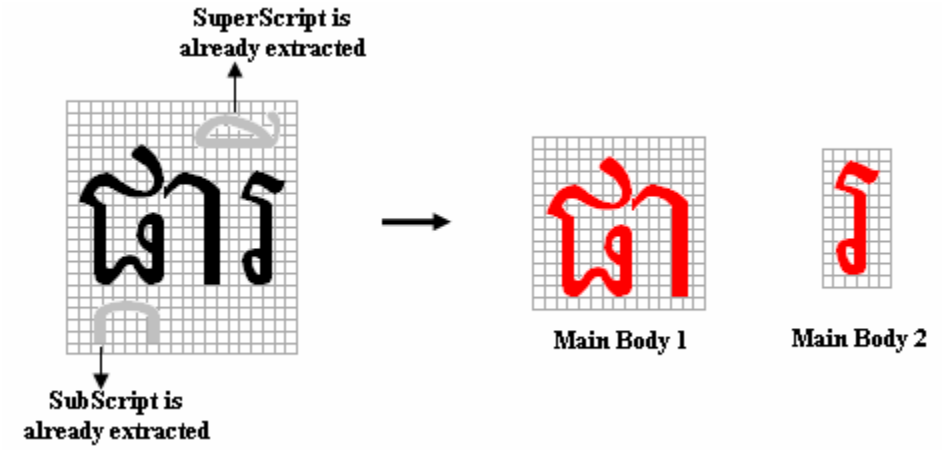


Figure 3.89 *Main Body Extraction, Case 1, Sub-case 2 (Other cases),, Two Main Bodies*

2. Case 2 (Three bodies)

- Case 2, Sub-case 1 (MMM, [M: Main Body])

  Initially, we get the left and right boundaries of the first body. Then, we get the top and bottom boundaries. After that, the extraction will be done accordingly. Moreover, in order to find the left position of the next Main Body, the right position of the previous one is used.

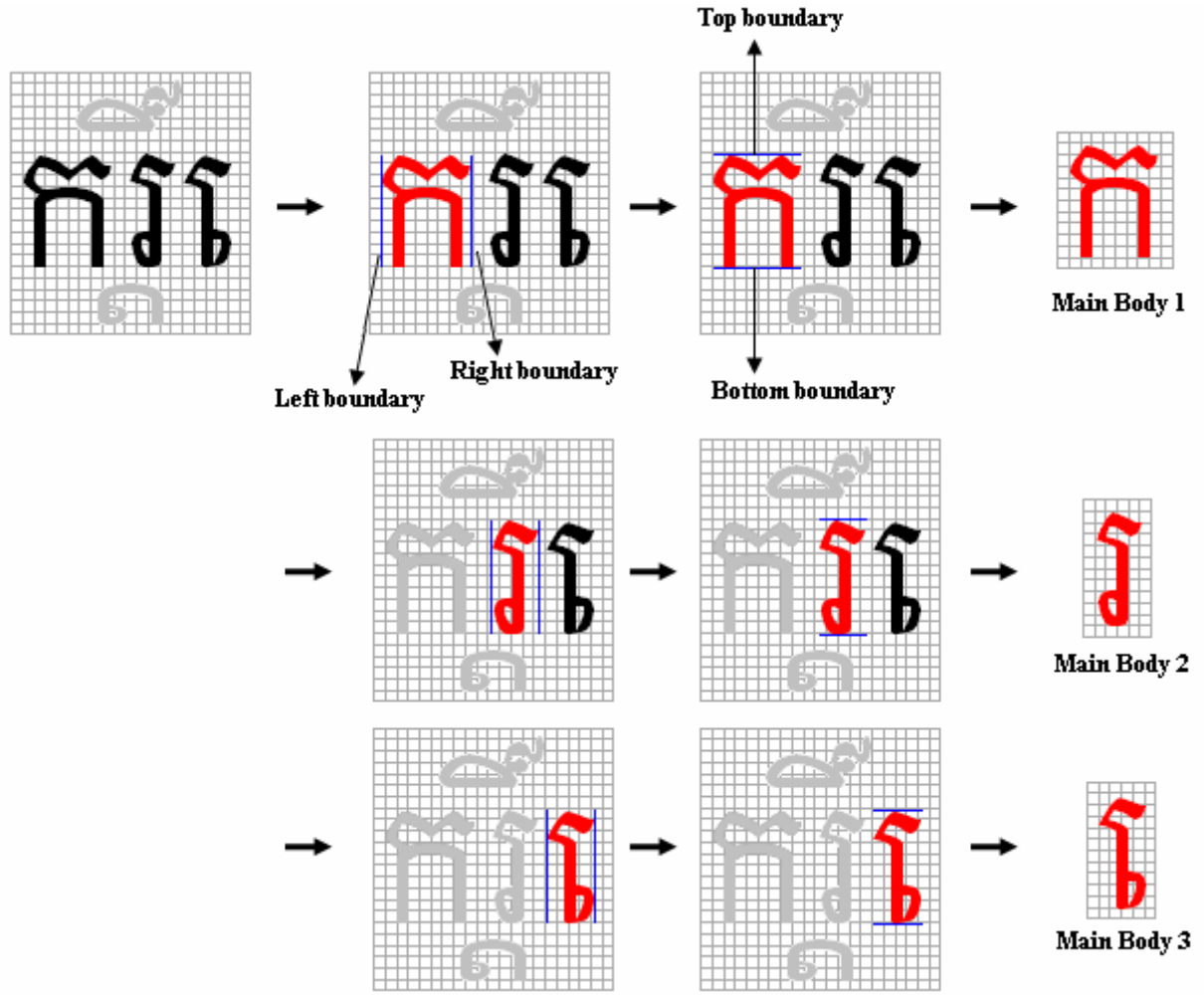


Figure 3.90 *Main Body Extraction, Case 2, Sub-case 1(MMM)*

- Case 2, Sub-case 2 (McM, [c – CCDown, M – Main Body])

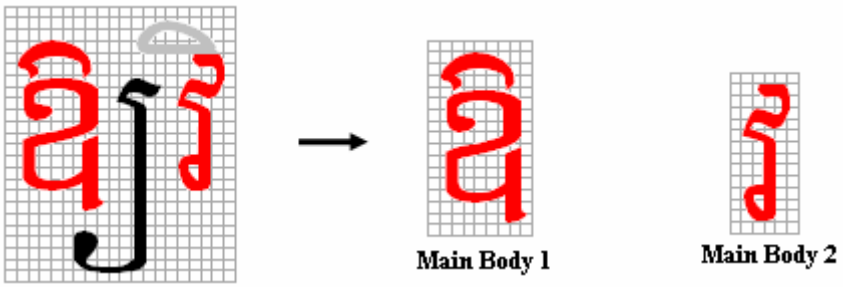


Figure 3.91 *Main Body Extraction, Case 2, Sub-case 2(McM)*

- Case 2, Sub-case 3 (McC, [M – Main Body, c – CCDown, C – CC])

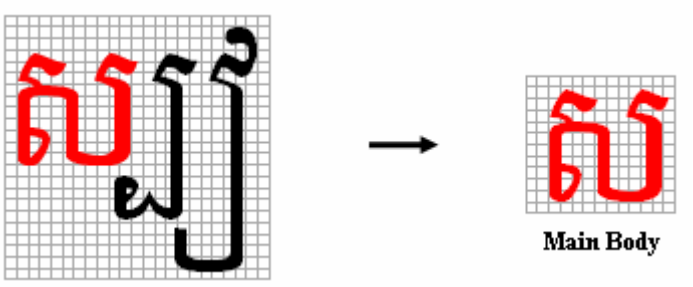


Figure 3.92 *Main Body Extraction, Case 2, Sub-case 3(McC)*

- Case 2, Sub-case 4 (cMM, [c – CCDown, M – Main Body])

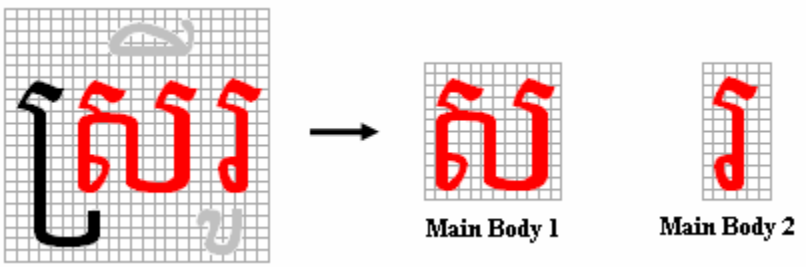


Figure 3.93 *Main Body Extraction, Case 2, Sub-case 4(cMM)*

- Case 2, Sub-case 5 (cMC, [c – CCDown, M – Main Body, C – CC])

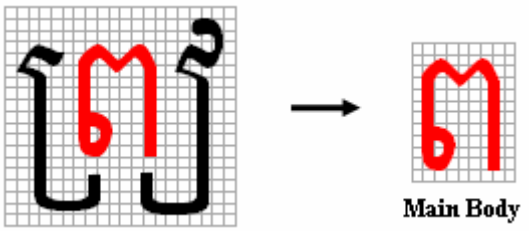


Figure 3.94 *Main Body Extraction, Case 2, Sub-case 5(cMC)*

3.3.2.6. CCDown and CC Extraction

3.3.2.6.1. Extraction Methodology

Since the remaining shape will be the shape of the CCDown or the CC, the extraction of each can be done by using the same methodology. The extraction is applied by using the basic technique used in the other extraction processes. In other words, it involves the finding of the shape's boundary. Additionally, in order to distinguish between the CCDown and the CC, a condition has to be checked so that the naming convention of the output is correct. Furthermore,

any Character that has two bodies will be addressed in different case from that which has three bodies.

3.3.2.6.2. Algorithm Approaches

As mentioned above, we have to find the boundary of the shape. Therefore, the extraction involves finding the top, bottom, left, and right boundary of the shape. However, the extraction only works when the information of the Character contains either CCDown or the CC.

1. **Case 1 (Two bodies)**

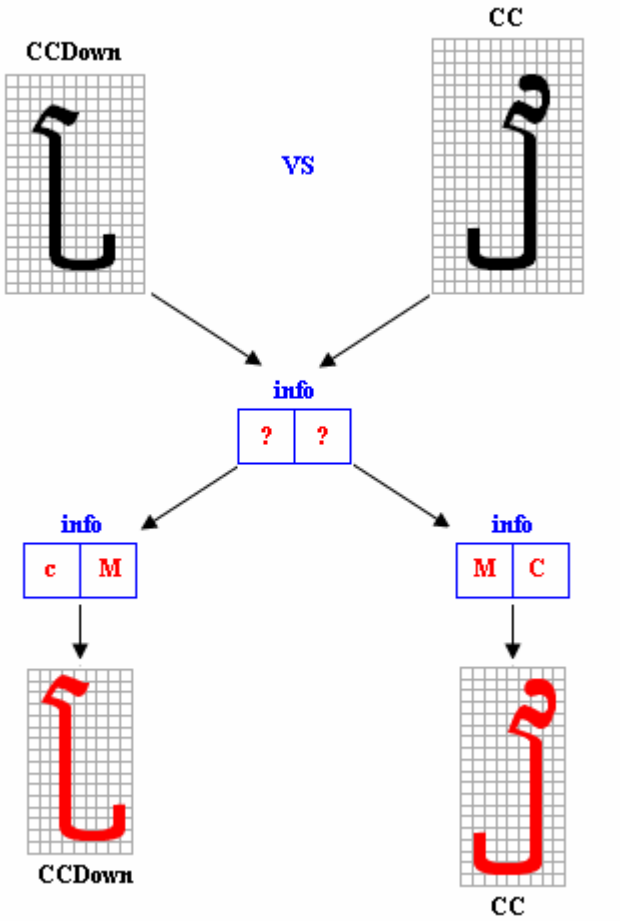


Figure 3.95 *CCDown and CC Extraction, [c – CCDown, M – Main Body, C – CC]*

In the above case, if the information about the Character indicates that there is a CCDown, the extracted shape will be identified as the CCDown. Otherwise, it will be identified as the CC.

2. Case 2 (Three bodies)

For CCDown extraction, there are two cases. The first case deals with any Character that has CCDown on the second body. The second case deals with those that have CCDown on the first body.

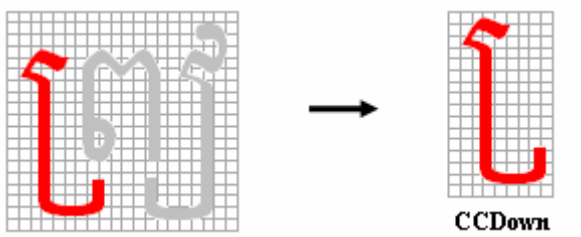


Figure 3.96 *CCDown and CC Extraction, Case 2, CCDown on the first body*

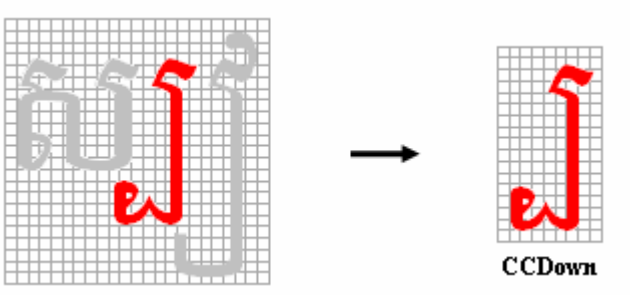


Figure 3.97 *CCDown and CC Extraction, Case 2, CCDown on the second body*

After that, CC will be extracted. Since it always resides on the third body, this is the last extraction process. Therefore, the extraction is not a real challenge. What we have to do is to find its boundaries (top, bottom, left, and right).

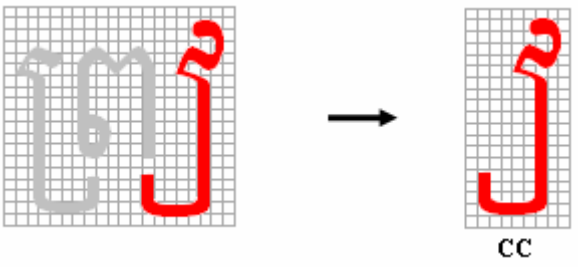


Figure 3.98 *CCDown and CC Extraction, Case 2, CC (on the third body)*

4. Results

We did experiments on ten scanned pages with skew-free and noise-free images. All the pages were printed in Limon S1 font, size 22, and scanned with the resolution of 300 dpi. The following is the result of the experiments.

Total Characters: 16164

	Total	Segmented	Errors
Main Body	16318	16262	56
SuperScript	3075	3022	53
SubScript	2167	2140	27
CCDown	965	949	16
CC	147	99	48

Overall, the main cause of error segmented Main Bodies is the error in extracting the SuperScript. The upper part of the Main Body is confused as the SuperScript, and thus it is extracted, leaving the Main Body with the incomplete shape. Another error usually occurs when Double quote presents. It is segmented as a SuperScript which should be the Main Body. For SubScripts, most of the Subscripts at the first level are missing. The others are not extracted when presents with either the CCDown or the CC. Furthermore, the most critical part is with the identification of the CCDown and the CC. Most of the time, CC is identified as the CCDown because of the Text Band.

In short, Main Body extraction can have a better result once SuperScript extraction is improved. In order to improve the SuperScript extraction, we should consider the part of code where noise is detected since the SuperScript is wrongly extracted once noise is misidentified. Moreover, SubScript extraction may require a better technique to identify because most of them are missing after the extraction. Futhermore, the CCDown and the CC problems can also be improved by correctly identifies the Text Band because the only method we propose to distinguish between the two requires the correct Text Band identification.

5. References

[1] Khmer alphabet, http://www.omniglot.com/writing/khmer.htm

[2] Khmer, http://www.ancientscripts.com/khmer.html

[3] http://www.khekbros.com/index.shtml

Appendixes

Appendix A: Main Body (138)

3	1	2	3
	ក	កា	កៅ
6	4	5	6
	ខ	ខា	ខៅ
9	7	8	9
	គ	គា	គៅ
12	10	11	12
	ឃ	ឃា	ឃៅ
15	13	14	15
	ង	ងា	ងៅ
18	16	17	18
	ច	ចា	ចៅ
21	19	20	21
	ឆ	ឆា	ឆៅ

26	22	23	24	25	26
	ជ	ជា	ជៅ	ជំ	ជះ

29	27	28	29
	ឈ	ឈា	ឈៅ
32	30	31	32

	ಣ	ಣಾ	ಣಿ
37	33 ಡ	34 ಡಾ	35 ಡಿ	36 ಡೀ	37 ಡು
42	38 ಪ	39 ಪಾ	40 ಪಿ	41 ಪೀ	42 ಪು
45	43 ಫ	44 ಫಾ	45 ಫಿ
48	46 ಬ	47 ಬಾ	48 ಬಿ
51	49 ಭ	50 ಭಾ	51 ಭಿ
53	52 ಮ	53 ಮಾ
58	54 ಯ	55 ಯಾ	56 ಯಿ	57 ಯೀ	58 ಯು
61	59 ರ	60 ರಾ	61 ರಿ
64	62	63	64

	ធ	ធា	ធៅ
67	65	66	67
	ន	នា	នៅ
70	68	69	70
	ប	ផ	ផៅ

75	71	72	73	74	75
	ផ	ផា	ផៅ	ផំ	ផ័

78	76	77	78
	ព	ពា	ពៅ
	79	80	81
	ភ	ភា	ភៅ
84	82	83	84
	ម	មា	មៅ
87	85	86	87
	យ	យា	យៅ
90	88	89	90
	រ	រា	រៅ
93	91	92	93
	ល	លា	លៅ

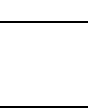
98	94	95	96	97	98

	រ	រា	រ៉	រិ	រី
101	99	100	101		
	ស	សា	សៅ		
104	102	103	104		
	ហ	ហា	ហៅ		
107	105	106	107		
	អ	អា	អៅ		
108	ឥ				
109	ព				
110	ឆ				
111	ឋ				
112	ឯ				
113	ឍ				
114	ដ៏				
115	◌				

116	ๅ
117	เ
118	๏
119	ฯ
120	ๆ
121	๗
122	ๅ
123	๙
124	ฝำ
125	ฝำ
126	ฝำ
127	ฝำ
128	ภำ
129	ภำ

130	
131	
132	
133	
134	
135	
136	
137	
138	

Appendix B: SuperScript (25)

1	
2	
3	
4	
5	
6	
7	

8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	

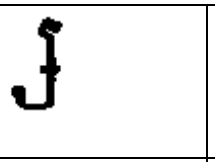
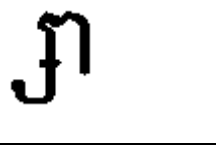
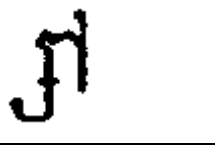
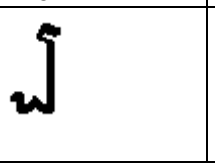
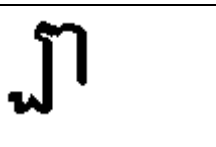
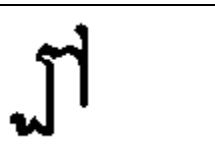
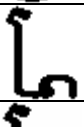
Appendix C: SubScript (31)

1				
2				
3				
4				
5				
6				
7				
11	8	9	10	11
				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				

28	
29	
30	
31	

Appendix D: CCDown (25)

2	1			2	
					
4	3			4	
					
6	5			6	
					
8	7			8	
					
10	9			10	
					
15	11	12		13	14	15
						
18	16		17		18	

			
21	19	20	21
			
22			
23			
24			
25			

Appendix E: CC (14)

1		
3	2	3
		
5	4	5
		
6		

7	
8	
9	
10	
11	
12	
13	
14