Initial Report on

Online Sinhala Handwritten Character Recognition Tool

Country Component		**Report no.**	
Phase no.		**Report Ref no.**	

Prepared By: _____
(Name)

Designation: _____

Project Leader: _____
(Name)

Signature: _____

Date: _____

TABLE OF CONTENTS

1. Summary

The primary goal of this research is to develop a practical isolated Sinhala character recognizer using DTW algorithm. The DTW classifier is a template based classifier; i.e. an unknown sample entered to the system is compared with all templates in the system. The classification is carried out by calculating the DTW distance between the templates and the entered query. Upon identification of the strokes, a list of possible characters will be calculated according to the sorted DTW distance. The most probable character will be determined based on the DTW distance; however confusing characters will be displayed so that the users can select the correct character in such situations.

1.1. Current Status of OLCR project

1. A comprehensive literature survey was carried out.
2. A data capture application has been developed.
    a. A sample application "PocketSignature", written in C# (.NET 2005), distributed by Microsoft was further modified to capture online handwritten data.
3. A data analysis and classification tool was developed
    a. This application was developed using Visual Basic .NET (2005)

1.2 Future Work

1. Handwritten Data Collection

2. Data Cleaning and Data Classification

    a. Erroneous data samples have be identified by visual inspection of handwritten data using Online Handwritten Character Analysis Tool (OHCAT). Spurious strokes, erroneous placing and lifting of the stylus etc may lead to errors. The erroneous samples have to be eliminated manually prior to template generation process.

    b. Initially, individual stroke identification and classification has to be carried out. Then, a Finite State Automata can be implemented in order to recognize a character by identifying its component strokes. Therefore, individual strokes have to be

extracted and isolated from the character data samples. Stroke classification can be done with the help of OHCAT.

3. Separation of Training and Testing data
    a. Cleaned data have to be classified mainly into two groups, for training and for testing. (Appropriate numbers have to be decided.)
    b. Testing Data set have to be further partitioned into two groups (Group A and Group B).

4. Template Generation
    a. Optimal set of strokes have to be selected from the training data set for generating templates. In order to generate templates for a certain stroke, similar strokes have to be determined and then those similar strokes have to be averaged or merged to obtain the final set of templates. Clustering techniques can be used to find similar samples (ie. by clustering samples using the DTW distance).
    b. Template generation can be carried out by merging (or taking the average of) similar stroke clusters.

5. Evaluation of Templates
    a. By changing various parameters of the DTW algorithm, recognition accuracy have to be calculated with Group A testing data. The optimal parameter values have to be determined at this stage. The generated templates have to be manually examined by an expert. Erroneous templates have to be re-generated as described in the step 4. Until a stable and good recognition performance yields, this procedure has to be performed iteratively.

6. Evaluation of the Classifier (OLCR system)
    a. Evaluate the performance of the OLCR system using Group B testing data.

7. Implement DTW classifier using Visual C++ for the Windows CE platform. Optimize algorithms and deploy the software in Pocket-PC devices.

8. Document Project Report.

2. Data Capture Application

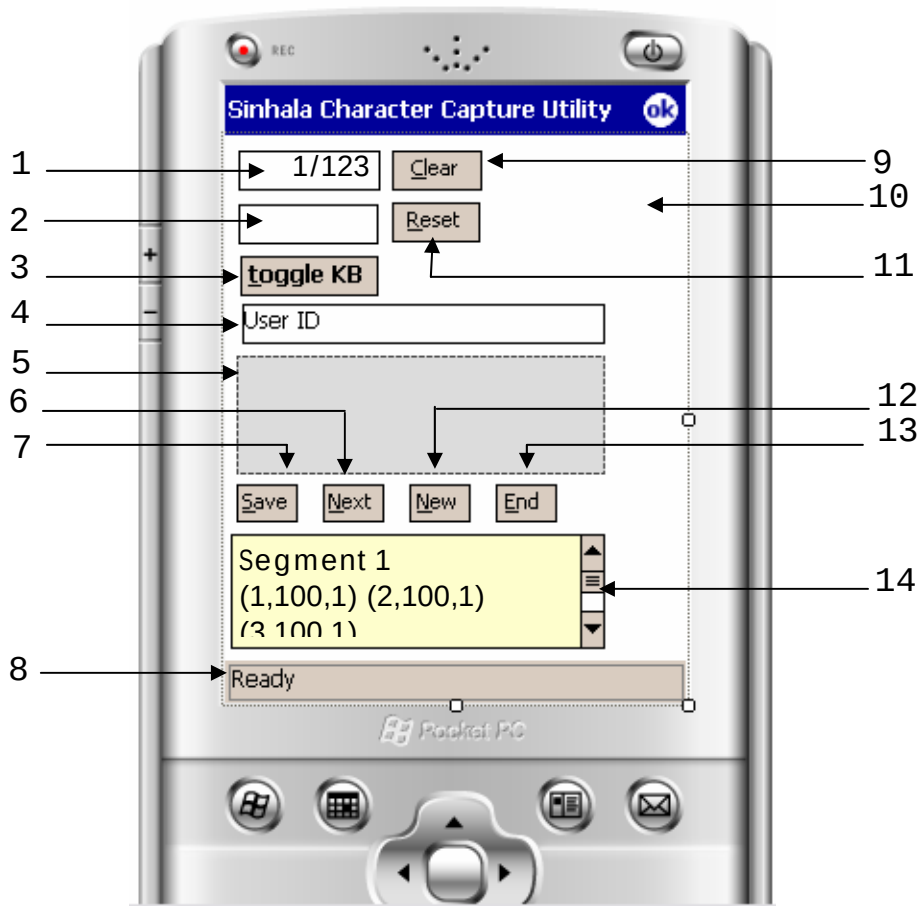

Fig. 1 A-Screen-shot of Data Capture Application

1. Current Sample Number Display
2. Reset Index Input Box
3. On/Off pop-up keyboard button
4. Sample (File) Name Input Text Box
5. Data Capture Area
6. Next Button
7. Save Button
8. Status Bar
9. Clear Button
10. Current Character Display
11. Reset button
12. Initialize New Sample Capture Session button
13. Save Captured Data button
14. Captured Data information Display Area

The Data Capture Application can be use to capture online handwritten character data and store them in an InkML compatible XML file format. Current version captures 154 characters (or glyphs) and each character will be captured 8 times. Therefore, all together 1232 samples will be captured at a single session (see Annexure I).

2.1 New Sample Capturing Procedure

1. Enter a sample name in the "Sample File Name Input Box" (2). Captured data will be stored in an XML created in the storage card of the pocket pc device.

   Note: Name of the XML file will be the sample name entered in this input box.

2. Click "Initialize New Sample Capture Session button" (12).

   Now you are ready to capture online handwritten character data. The "Current Sample Number Display" will show the index of the current sample that is being captured and the total number of samples remaining to be captured. Please note that each character has to be captured 8 times (5 for training data and 3 for testing data). The character to be captured is shown in the "Current Character Display" (1). However, due to some rendering issues related to Windows CE platform in pocket pc devices some Sinhala characters may not display properly.

3. In the "Data Capture Area" (5), the desired character has to be written in a stylus. A single character has to be written and it should lie within the boundaries of the "Data Capture Area". Guidelines are provided in the "Data Capture Area". A character may contain multiple strokes, however all strokes belonging to a single character must be written within the boundaries of the "Data Capture Area". Some statistics about the captured data will be shown in the "Captured Data information Display Area" (14). These information include, the number of strokes in the character and the captured coordinates of the stylus in (x,y,t) format. Here, x and *y* represent the coordinates of the stylus and t represents the time in milliseconds.

Note: If you are not satisfied with the character that you've written using the stylus, click "Clear" button at this stage and re-draw the character before saving it (see step 4 below).

4. Once you have written the character in the “Data Capture Area” (5), Click “Save” (7) button to append stroke/character data to the XML file. The “Status Bar” will display the saving status of the data.

5. Now click “Next”(6) button in order to capture next character.

6. Repeat above 3-5 steps until you capture all samples.

7. Click “End”(13) button to finalize the data capturing session and generate complete XML containing captured data.

8. Transfer the XML containing captured data to a PC using ActiveSync or SD card reader.

2.1.1 Additional Functions and Features

1. Enter a starting index in the “Reset Index Input Box” and click “Reset” button to start re-capturing characters starting from a given index.

2. Click “On/Off Pop-Up Keyboard” (3) button to stop interfering on-screen keyboard in the process of data capturing.

3. Online Handwritten Character Analysis Tool

A tool was developed in order to manually separate and classify individual strokes of characters. Furthermore, this application can itself be used to do further analysis and comparison of individual strokes of different characters (see Fig 2).

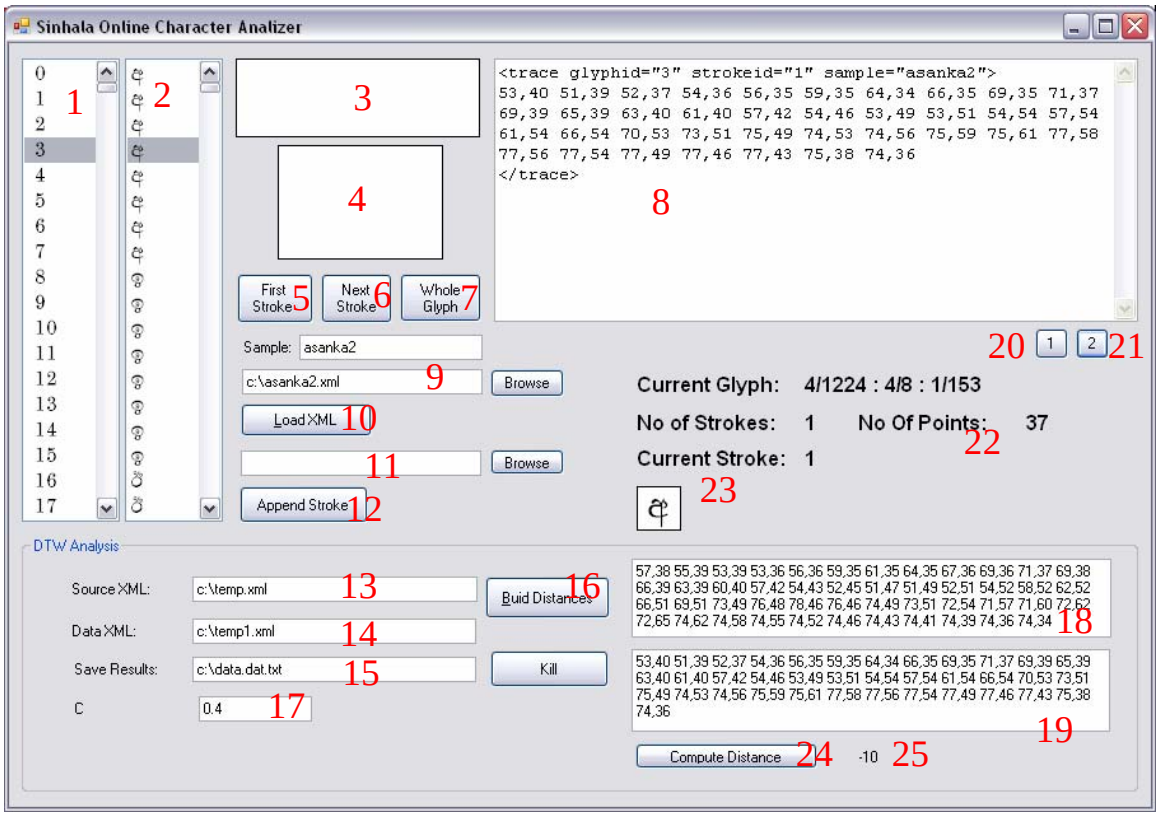

Fig. 2. A Screen-shot of Online Handwritten Character Analysis Tool

Controller's Description

1. Character List Box
2. Glyph (Index) List Box
3. Character Display Area
4. Modified Character/Stroke Display Area
5. First Stroke Button
6. Next Stroke Button
7. Whole Glyph Button
8. Append Stroke Button
9. Source XML Input Path for Advanced DTW analysis

10. Reference XML Input Path for Advanced DTW Analysis
11. Output Path for Advanced DTW Analysis
12. Perform Advanced DTW Analysis Button
13. C - Parameter Input Text Box
14. DTW Comparison Data Input Box 1
15. DTW Comparison Data Input Box 2
16. Copy Source XML into Data Input Box 1 Button
17. Copy Source XML into Data Input Box 2 Button
18. Character/Stroke Information Display Area
19. Current Glyph/Character Display
20. Compute DTW Distance Button
21. DTW Distance Display

Using Online Handwritten Character Analysis Tool

1. Enter full path of the XML (InkML generated by data capture application) to be analyzed in the "Source XML Path Input Text Box" (9).

2. Click "Load XML" button (10) to load captured data sample into application.

   Character indexes and the corresponding (expected) glyphs will be loaded and shown in the "Character List Box" (1) and "Glyph (Index) List Box" (2) respectively.

3. Now, select either a character Index or a glyph from "Character List Box" (1) or "Glyph (Index) List Box" (2) in order to analyze or separate into strokes.

   "Character Display Area" (3) and "Modified Character/Stroke Display Area" (4) will re-draw the captured character/stroke data on a PC screen. "Modified Character/Stroke Display Area" (4) will display individual strokes of a character or entire character in the middle of the area by applying the first moment normalization algorithm. "XML Code Display Area" (8) will display the XML code corresponding to a single stroke or an entire character. Various statistics about the selected stroke/will be displayed in the "Character/Stroke Information Display Area" (22) while the current (expected) glyph will be displayed in the "Current Glyph/Character Display" (23).

4. In order to traverse in between individual strokes of a character “First Stroke”, “Next Stroke” and “Whole Glyph” Buttons (5,6,7) can be used.

Saving Strokes (Stroke Classification)

5. Selected coordinate data of a stroke can be appended to an XML file. In order to create an output XML file, enter a full path (along with a name for the XML) in the "Output XML Path Input Text Box". Then, click "Append Stroke Button" to save selected stroke data into the given XML. This step can be used to classify individual strokes into groups.

Stroke Distance Analysis using Dynamic Time Warping (DTW) Algorithm

In this project, it is expected to implement template matching technique using DTW algorithm as the fine classifier. The algorithm has been implemented in this analysis tool and it can be used to calculate matching distance between given two strokes.

6. Click "Copy Source XML into Data Input Box 1 Button" (20) to copy current stroke into "DTW Comparison Data Input Box 1" (18).

7. Click "Copy Source XML into Data Input Box 2 Button" (21) to copy current stroke into "DTW Comparison Data Input Box 2" (19).

8. Click "Compute DTW Distance Button" to calculate DTW distance between given strokes in the above steps (6,7).

   DTW distance will appear in "DTW Distance Display".

9. Change the value in "C - Parameter Input Text Box" and re-calculate DTW distance (8) in order to find an optimized value for C.

Note:
Advance XML based distance calculation is not yet optimized and it may take couple of hours to process an entire XML file.

Annexure- I – Character Selection

1.අ
2.ඉ
3.ඊ
4.උ
5.ඍ
6.එ
7.ඔ
8.ඕ
9.ං
10.ඃ
11.ක
12.කු
13.කූ
14.ක්‍ර
15.ඛ
16.ඛ
17.ඛි
18.ඛී
19.ග
20.ගු
21.ගූ
22.ග්‍ර
23.ඝ
24.ඝ්‍ර
25.ඩ
26.ඟ
27.ඟු
28.ඟූ
29.ච
30.ඡ
31.ඡි
32.ඡී
33.ජ
34.ජී
35.ජී
36.ජ
37.ජී
38.ජී
39.ඣ
40.ඣි
41.ඤ
42.ඤු
43.ඥ
44.ඦ
45.ට
46.ට්‍ර
47.ඨ
48.ඨි
49.ඨී
50.ඩ
51.ඩී
52.ඩී
53.ඪ
54.ඪ
55.ඪි
56.ඪී
57.ණ
58.ණී
59.ණ
60.ණු
61.ණූ
62.ණී
63.ණී
64.ඬ
65.ඬ
66.ඬි
67.ඬී
68.ත
69.තු
70.තූ
71.ත්‍ර
72.ථ
73.ථී
74.ථී
75.ද
76.දා
77.ද්‍ය
78.දු
79.දූ
80.ද්‍ර
81.ධ
82.ධි
83.ධි
84.ධී
85.න
86.නු
87.නූ
88.ඳ
89.ඳු
90.ඳූ
91.ප
92.ඵ
93.ඵී
94.ඵී
95.බ
96.බ්
97.බි
98.බී
99.භ
100.භු
101.භූ
102.භ්‍ර
103.ම
104.ම්
105.මි
106.මී
107.ඹ
108.ඹි
109.ඹි
110.ඹී
111.ය
112.ර
113.රි
114.රී
115.ර්
116.ර්
117.ල
118.ලු
119.ලූ
120.ව
121.වි
122.වි
123.වී
124.ශ
125.ශු
126.ශූ
127.ශ්‍ර
128.ෂ
129.ස
130.හ
131.හු
132.හූ
133.ළ
134.ළු
135.ෆ
136.ෆු
137.ෆූ
138.ෆ්‍ර
139.්
140.ා
141.ැ
142.ෑ
143.ි
144.ී
145.ු
146.ූ
147.ෘ
148.ෙ
149.ෛ
150.ෛ
151.ක්ෂ
152.ඣ
153.ඳ
154.්‍ය

Notes:

1. When writing characters '೭' and '೭ಽ' (76 and 77) alternative glyphs have to be used.
2. ೮ and ೯ (115 and 116) should be written in a single stroke.

This character set was selected with the help of a linguist. Majority of the characters comprise of a single stroke. Characteristics (differences) of the individual strokes was the basic criteria in selecting characters to be included in the sample data set.