Slap Fingerprint Segmentation Evaluation 2004

API Specification

Overview

The Slap Fingerprint Segmentation Evaluation 2004 will investigate the accuracy of fingerprint image segmentation systems for use with multi-finger slap images.
The segmentation application to be tested is assumed to run on Windows 2000 or Red Hat Linux 7.2, on x86 platforms. Other options must be approved by the Test Liaison. The application must have a command-line interface as specified in this document; no other user interface is permissible.
The segmentation application must be capable of taking as input either a Wavelet Scalar Quantization (WSQ) compressed slap image, or an uncompressed raw slap image, and outputting up to four uncompressed raw segmented images, each corresponding to one of the individual fingers pictured in the slap. It is recommended that the segmentation application return segmentation quality values, indicating the likelihood that that image was correctly segmented.

1 Application Inputs

1.1 Slap Image Files

The segmentation application must be capable of processing multi-finger slap images stored either in

- Wavelet Scalar Quantization (WSQ) compressed format, or
- Uncompressed raw 8-bit (one byte per pixel) grayscale format.

These formats are defined further below.

1.1.1 Resolution and Dimensions

All images for this test shall employ 500 PPI resolution (horizontal and vertical).
In almost all cases, the dimensions of slap images are within the maxima defined by ANSI/INCITS 381[2]: width less than 83.8 mm (3.3 in), and height less than 76.2 mm (3.0 in).
In a few cases, the dimensions may exceed those values, up to a maximum of 101.6 mm (4.0 in), or 2000 pixels.

1.1.2 Slap Image Filenames

Multi-finger slap image files shall be specified in the command line either by relative pathnames, or fully-qualified pathnames. Unix-style forward slashes ("/") shall be used, not Windows-style backward slashes ("\"). For example,
/share/data/slap999.wsq

The *root* filename is defined as the filename without the path or extension. For example,
slap999

Filenames will be limited to alphanumeric characters and underscores (i.e. they will not contain blanks). Symbolic links or Windows shortcuts will not be used.

1.1.3 WSQ File Format

WSQ files are compliant with the document WSQ Gray-Scale Fingerprint Image Compression Specification.[4]
NIST has public domain WSQ tools in NIST Fingerprint Image Software (NFIS). [1] A variety of commercial vendors of WSQ tools can easily be found by doing an Internet search, e.g. for "wsq compression software."

1.1.4 Raw File Format

Raw 8-bit grayscale image files are canonically encoded with black equal to 0, white equal to 255, etc.; stored left to right, top to bottom, with one 8-bit byte per pixel. The number of bytes in a file is exactly the image height * image width, as measured in pixels; there is no header.

1.2 Input Parameters

The following information shall be provided as parameters to the segmentation application:

Hand identifier[-h]
: specifies R (right), L (left) or U (unknown) hand corresponding to the specified slap image.

Source [-s]
: specifies the source of the fingerprint image: L (livescan), P (paper), or U (unspecified; could be livescan or paper).

Output path [-o]
: optionally specifies the directory to which all output files shall be written. The default is the directory in which the application is currently executing.

Slap image filename
: specifies a file containing either a WSQ compressed, or uncompressed 8-bit grayscale slap image to segment. In the latter case, the height and width of the image in pixels shall follow as additional parameters.

The parameters (if present) will always be in the stated order. Parameters will be separated by spaces or tabs.

1.3 Example Command-line Usage

The following are examples of how the input parameters may be specified to the segmentation application (using “Unix-like” command-line usage syntax). In the following examples, items within “[ ]” are optional.
The application should be named “segment” (Linux) or “segment.exe” (Windows).
Usage:

```
segment -h ID –s SOURCE [-o DIR] image.wsq
```

or

```
segment -h ID –s SOURCE [-o DIR] image.raw HEIGHT WIDTH
```

-h ID
: Hand identifier (R=right, L=left, U=unknown) (no other cases; uppercase only)

-s SOURCE
: fingerprint image source (L=livescan, P=paper, U=unspecified) (no other cases; uppercase only)

-o DIR
: Output directory for segmented image files

image.wsq
: WSQ compressed image filename

image.raw HEIGHT WIDTH

Raw image filename with height and width in pixels

2 Application Outputs

2.1 Segmented Image Files

For each slap image file that is input, up to four separate segmented image files may be produced, depending upon the number of individual fingers that the segmentation system is able to find in the slap image. Unless otherwise specified by the application input parameter, *Output Path*, the segmented image files shall be written to the current execution directory.

2.1.1 Segmented Image File Format

Segmented images shall be stored in uncompressed, raw 8-bit (one byte per pixel) grayscale format and shall preserve the original resolution (PPI) of the slap image. The segmented image shall be encoded canonically (black equal to 0, white equal to 255, etc.) and stored left to right, top to bottom.

2.1.2 Segmented Image Filenames

Segmented image files shall be written with the root filename (as defined in 1.1.2 Slap Image Filenames) concatenated with _{seg_pos} followed by _{finger_pos} and the extension *.raw*.

Seg_pos

The segmented position is defined to begin at the capital letter "A", and increment for each additional segmented image created from the slap. The seg_pos letters must progress from left to right for the fingers in the slap image. For example, for a left slap, the little finger would be assigned a seg_pos of A, while for a right slap, the index finger would be assigned a seg_pos of A.

Finger_pos

The finger position is the position code defined in Table 6 of the standard ANSI/NIST-ITL 1-2000 [3]:

00 Unknown
01 Right thumb
02 Right index
03 Right middle
04 Right ring
05 Right little
06 Left thumb
07 Left index
08 Left middle
09 Left ring
10 Left little

Finger_pos shall be written as two digits with a leading zero (if necessary). For example,

```
slap999_A_10.raw
slap999_B_09.raw
slap999_C_08.raw
slap999_D_07.raw
```

Note that if the position code cannot be determined by the segmentation application, it shall set *finger_pos* equal to 0, for example,

```
slap999_A_00.raw
```

2.1.3 Dimensions

It is assumed that the output files are unlikely to have width or height greater than 1.5 inches (750 pixels), but any width or height up to 4 inches (2000 pixels) is permissible.
Output files with width or height less than 0.1 inches (50 pixels) will be ignored during analysis.

2.2 Meta-information Files

Along with outputting a set of (up to four) segmented image files for a single slap image, the segmentation application shall also produce a single meta-information file for each segmented image. Unless otherwise specified by the application input parameter, *Output Path*, the meta-information files shall be written to the current execution directory. The meta-information file is stored with a filename equal to the segmented image filename (see 2.1.2 Segmented Image Filenames) with the extension .nsm replacing .raw. For example,

```
slap999_A_10.nsm
slap999_B_09.nsm
slap999_C_08.nsm
slap999_D_07.nsm
```

The meta-information file shall contain a series of attribute-value pairs, formatted one per line. Minimally it must contain the following attribute-value pairs:

PIX_HEIGHT=value

value is the segmented image height in pixels

PIX_WIDTH=value

value is the segmented image width in pixels

The following attribute-value pairs may be output by the segmentation system, but are not required. We request that these be output if practical.

SEG_QUAL=value

value is the quality of the segmentation, in a user-defined numeric scale. A higher segmentation value must correspond to a higher likelihood that that image was correctly segmented. The values can be integers or decimals.

ORIGINAL_ROTATION=value

value is the amount of rotation of the original (input) finger from vertical, in degrees. Positive values are clockwise rotation, and negative values are counter-clockwise. The values can be integers or decimals.

OUTPUT_ROTATED=(TRUE or FALSE)

If the output images are rotated relative to the input, OUTPUT_ROTATED should be set to TRUE.

BOUNDINGBOX=(X,Y)topleft, (X,Y)topright, (X,Y)bottomleft, (X,Y)bottomright

The four corners of a rectangle bounding the individual segmented finger, measured in pixels from the top left corner of the input image.

For example,

```
PIX_HEIGHT=200
PIX_WIDTH=120
SEG_QUAL=0.82
ORIGINAL_ROTATION=-30
OUTPUT_ROTATED=FALSE
BOUNDINGBOX=(100,100),(400,100),(100,400),(400,400)
```

2.3 Error Codes and Handling

The segmentation application shall exit with a return code of zero on success. The participant must provide documentation of all (non-zero) error or warning codes (see section 3.3, Documentation).
The application should include error/exception handling so that in the case of a crash, the return code is still provided to the calling shell.
We request that the following return codes be used:

Return code	Explanation
0	Success
1	Unable to read input file
2	Unable to open/decompress input WSQ file
3	Unable to write output files
10	0 fingers could be segmented
11	Only 1 finger could be segmented
12	Only 2 fingers could be segmented
13	Only 3 fingers could be segmented
20 - 63	Application-specific fatal errors (explained in documentation)
64 - 127	Application-specific non-fatal warnings (explained in documentation)

All errors, warnings and informational messages shall be limited to output displayed via standard output or standard error. No GUI-type dialog windows are permitted.

3 Software and Documentation

3.1 Application type and platform

The application provided shall be command-line driven, and capable of being run in non-interactive “batch mode.” No graphical user interface (GUI) is permitted.
Test participants shall provide NIST with binaries only (i.e. no source code) for their segmentation application. Testing of segmentation systems will be performed on commercial, off-the-shelf PCs. Applications running on Red Hat Linux 7.2 or Microsoft Windows 2000 are preferred; other operating systems must be approved by the Test Liaison.

3.2 Installation

Segmentation software must install and run easily to be evaluated.
It is recommended that the application shall be immediately executable without use of an installation program. Please contact the Test Liaison if an installation program is absolutely necessary.
The application shall be executable on any number of machines without requiring additional machine-specific license control procedures or activation.
It is preferred that the application be packaged as a single executable file. If external libraries (such as DLLs) are necessary, they should work from the application directory, and not require installation in another location.

3.3 Documentation

Complete documentation of application usage shall be provided, and shall detail any additional functionality or behavior beyond what is specified in this document.
The documentation must define all error and warning codes.
The documentation must define the range of segmentation quality values, at least to the extent of defining the minimum and maximum values. If thresholds are known, they should be documented: e.g. “values above 0.8 are almost always correctly segmented, and values below 0.2 are almost always incorrect”.

3.4 Speed

Software that runs excessively slowly cannot be evaluated. On average, segmentation software should take ***much*** less than ten (10) seconds to segment a slap image (using a 1ghz Pentium III). Due to resource limitations, software that takes longer than that may not be evaluated. Processing speed will be noted but will not be a primary evaluation criterion.

3.5 Sample Data and Results

Participants must test their software using the SlapSeg04 Sample data, and email these results to the Test Liaison for validation before sending software to NIST for evaluation.

References

[1] *NIST Fingerprint Image Software (NFIS)*
http://www.nist.gov/itl/iad/ig/nbis.cfm
[2] *American National Standard for Information Technology - Finger Image-Based Data Interchange*, ANSI/INCITS 381-2004, May 13, 2004, www.incits.org
[3] R. M. McCabe, "ANSI/NIST-ITL 1-2000 Data Format for the Interchange of Fingerprint, Facial, and Scar Mark & Tattoo (SMT) Information"

[4] *WSQ Gray-Scale Fingerprint Image Compression Specification*, IAFIS-IC-0110 (V3) Criminal Justice Information Services Division, Federal Bureau of Investigation, December 1997