NIST Special Database 4

Fingerprint Database

C.I.Watson and C.L. Wilson

National Institute of Standards and Technology
Advanced Systems Division
Image Recognition Group
March 17, 1992

1.0 INRODUCTION

This report describes the NIST Fingerprint database, *NIST Special Database 4*, which contains 8-bit gray scale images of randomly selected fingerprints. The database is being distributed for use in the development and testing of automated fingerprint classification systems on a common set of images. The CD-ROM contains 4000 (2000 pairs) fingerprints stored in NIST's IHead raster data format and compressed using a modified JPEG lossless [1] compression algorithm. Each print is 512 X 512 pixels with 32 rows of white space at the bottom of the print (see Appendix A). Approximately 636 Megabytes of storage are needed when the prints are compressed where as 1.1 Gigabytes are needed when uncompressed (1.6 : 1 average compression ratio).

The fingerprints are classified into one of five categories (L = left loop, W = whirl, R = right loop, T = tented arch, and A = arch) with an equal number of prints from each class (400). Each filename contains a reference to the hand and digit number so the classes can be converted to other classification techniques (i.e. radial and ulnar). All classes are stored in the NIST IHead **id** field of each file, allowing for comparison with hypothesized classes.

2.0 MODIFIED JPEG LOSSLESS COMPRESSION

The compression used was developed from techniques outlined in the WG10 "JPEG" (draft) standard for 8-bit gray scale images with modifications to the compressed data format. The NIST IHead format already contained most of the information needed in the decompression algorithm, so the JPEG compressed data format was modified to contain only the information needed when reconstructing the Huffman code tables and identifying the type of predictor used in the coding process. Codes used to compress and decompress the images are still developed per the draft standard, but only applied to 8-bit gray scale images.

The standard uses a differential coding scheme and allows for seven possible ways of predicting a pixel value. Tests showed that predictor number 4 provided the best compression on up to 99.9% of the fingerprint images; therefore, this predictor was used to compress all of the images.

3.0 DATABASE REFLECTANCE CALIBRATION

The reflectance values for the fingerprint dataset in *NIST Special Database 4* was calibrated using a reflection step table [3]. A plot of the reflectance values obtained using this step table is shown in Appendix B. Also shown on the plots and below is an equation used to predict the reflectance of a given datapoint. The plot in Appendix B shows that this predicted reflectance closely follows the actual reflectance obtained using the reflection step table.

$$predicted \text{ \% } reflectance = 10 + (.32 * grayscale \text{ } pixel \text{ } value)$$

4.0 FINGERPRINT FILE FORMAT

Image file formats and effective data compression and decompression are critical to the usefulness of image archives. Each fingerprint was digitized in 8-bit gray scale form at 19.6850 pixels/mm (500 pixels/inch), 2-dimensionally compressed using a modified JPEG lossless algorithm, and temporarily archived onto computer magnetic mass storage. Once all prints were digitized, the images were mastered and replicated onto ISO-9660 formatted CD-ROM discs for permanent archiving and distribution.

After digitization, certain attributes of an image are required to correctly interpret the 1-dimensional pixel data as a 2-dimensional image. Examples of such attributes are the pixel width and pixel height of the image. These attributes can be stored in a machine readable header prefixed to the raster bit stream. A program which manipulates the raster data of an image is able to first read the header and determine the proper interpretation of the data which follows it.

Numerous image formats exist, but most image formats are proprietary. Some are widely supported on small personal computers and others on larger workstations. A header format named IHead has been developed for use as a general purpose image interchange format. The IHead header is an open image format which can be universally implemented across heterogeneous computer architectures and environments. Both documentation and source code for the IHead format are publicly available and included with this database. IHead has been designed with an extensive set of attributes in order to adequately represent both binary and gray level images, to represent images captured from different scanners and cameras, and to satisfy the image requirements of diversified applications including, but not limited to, image archival/retrieval, character recognition, and fingerprint classification. Figure 1 illustrates the IHead format.

```
┌─────────────────────────────────────┐
│          Header Length              │
├─────────────────────────────────────┤
│                                     │
│      ASCII Format Image Header      │
│                                     │
├─────────────────────────────────────┤
│                                     │
│                                     │
│    8-bit Gray Scale Raster Stream   │
│                                     │
│   110101001101001111010010110 . . . │
│                                     │
│                                     │
│  • Representing the digital scan across the │
│         page left to right, top to bottom.  │
│  • 8 bits to a pixel                │
│  • 256 levels of gray               │
│  • 1 Pixel is packed into a single byte │
│         of memory.                  │
│                                     │
└─────────────────────────────────────┘
```
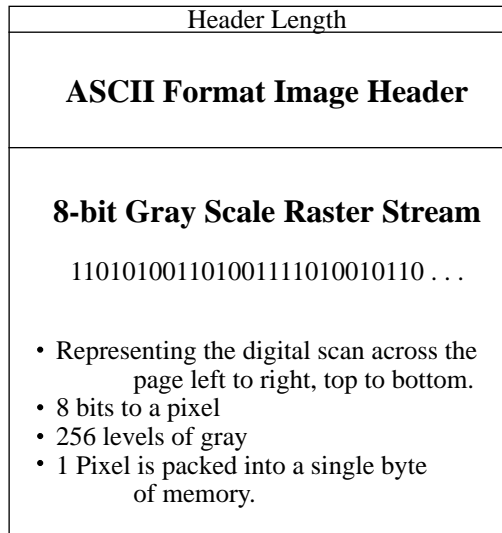
Figure 1: An illustration of the Ihead raster file format.

Since the header is represented by the ASCII character set, IHead has been successfully ported and tested on several systems including UNIX workstations and servers, DOS personal computers, and VMS mainframes. All attribute fields in the IHead structure are of fixed length with all multiple character fields null-terminated, allowing the fields to be loaded into main memory in two distinct ways. The IHead attribute fields can be parsed as individual characters and null-terminated strings, an input/output format common in the 'C' programming language, or the header can be read into main memory using record-oriented input/output. A fixed-length field containing the size in bytes of the header is prefixed to the front of an IHead image file as shown in Figure 1.

```
/**********************************************************
        File Name: IHead.h
        Package:   NIST Internal Image Header
        Author:    Michael D. Garris
        Date:      2/08/90
**********************************************************/
/* Defines used by the ihead structure */
#define IHDR_SIZE        288      /* len of hdr record (always even bytes) */
#define SHORT_CHARS      8        /* # of ASCII chars to represent a short */
#define BUFSIZE          80       /* default buffer size */
#define DATELEN          26       /* character length of data string */

typedef struct ihead{
  char id[BUFSIZE];                   /* identification/comment field */
  char created[DATELEN];              /* date created */
  char width[SHORT_CHARS];            /* pixel width of image */
  char height[SHORT_CHARS];           /* pixel height of image */
  char depth[SHORT_CHARS];            /* bits per pixel */
  char density[SHORT_CHARS];          /* pixels per inch */
  char compress[SHORT_CHARS];         /* compression code */
  char complen[SHORT_CHARS];          /* compressed data length */
  char align[SHORT_CHARS];            /* scanline multiple: 8|16|32 */
  char unitsize[SHORT_CHARS];         /* bit size of image memory units */
  char sigbit;                        /* 0->sigbit first | 1->sigbit last */
  char byte_order;                    /* 0->highlow | 1->lowhigh*/
  char pix_offset[SHORT_CHARS];       /* pixel column offset */
  char whitepix[SHORT_CHARS];         /* intensity of white pixel */
  char issigned;                      /* 0->unsigned data | 1->signed data */
  char rm_cm;                         /* 0->row maj | 1->column maj */
  char tb_bt;                         /* 0->top2bottom | 1->bottom2top */
  char lr_rl;                         /* 0->left2right | 1->right2left */
  char parent[BUFSIZE];               /* parent image file */
  char par_x[SHORT_CHARS];            /* from x pixel in parent */
  char par_y[SHORT_CHARS];            /* from y pixel in parent */
}IHEAD;
```

Figure 2: The IHead 'C' programming language structure definition.

The IHead structure definition written in the 'C' programming language is listed in Figure 2. Figure 3 lists the header values from an IHead file corresponding to the structure members listed in Figure 2. This header information belongs to the database file **f0001_01.pct** (see Figure A.1 in Appendix A). Referencing the structure members listed in Figure 2, the first attribute field of IHead is the identification field, **id**. This field uniquely identifies the image file, typically by a file name. The identification field in this example not only contains the image's file name, but also the classification of the fingerprint, any references to another classification, and the sex of the individual. This convention enables an image recognition system's hypothesized classification to be automatically scored against the actual classification printed in the **id** field.

```
IMAGE FILE HEADER
~~~~~~~~~~~~~~~~~
Identity            : f0001_01.pct W M
Header Size         : 288 (bytes)
Date Created        : MON OCT 28 17:25:45 1991
Width               : 512 (pixels)
Height              : 512 (pixels)
Bits per Pixel      : 8
Resolution          : 500 (ppi)
Compression         : 6 (code)
Compress Length     : 160602 (bytes)
Scan Alignment      : 8 (bits)
Image Data Unit     : 8 (bits)
Byte Order          : High-Low
MSBit               : First
Column Offset       : 0 (pixels)
White Pixel         : 255
Data Units          : Unsigned
Scan Order          : Row Major,
                      Top to Bottom,
                      Left to Right
Parent              : a0591.pct
X Origin            : 0 (pixels)
Y Origin            : 0 (pixels)
```

Figure 3: The IHead values for the fingerprint data file **f0001_01.pct**.

The attribute field, **created**, is the date on which the image was captured or digitized. The next three fields hold the image's pixel **width**, **height**, and **depth**. A binary image has a pixel depth of 1 whereas a gray scale image containing 256 possible shades of gray has a pixel depth of 8. The attribute field, **density**, contains the scan resolution of the image; in this case, 19.6850 pixels/mm (500 pixels/inch). The next two fields deal with compression.

In the IHead format, images may be compressed with virtually any algorithm. Whether the image is compressed or not, the IHead is always uncompressed. This enables header interpretation and manipulation without the overhead of decompression. The **compress** field is an integer flag which signifies which compression technique, if any, has been applied to the raster image data which follows the header. If the compression code is zero, then the image data is not compressed, and the data dimensions: width, height, and depth, are sufficient to load the image into main memory. However, if the compression code is nonzero, then the **complen** field must be used in addition to the image's pixel dimensions. For example, the image described in Figure 3 has a compression code of 6. This signifies that modified JPEG lossless compression has been applied to the image data prior to file creation. In order to load the compressed image data into main memory, the value in **complen** is used to determine the size of the compressed block of image data.

Once the compressed image data has been loaded into memory, JPEG lossless decompression can be used to produce an image which has the pixel dimensions consistent with those stored in its header. Using JPEG lossless compression and this compression scheme on the images in this database, an average compression ratio of 1.6 to 1 was achieved.

The attribute field, **align**, stores the alignment boundary to which scan lines of pixels are padded. Pixel values of 8-bit gray scale images are stored 1 pixel (or 8 bits) to a byte, so the images will automatically align to an even byte boundary.

The next three attribute fields identify data interchanging issues among heterogeneous computer architectures and displays. The **unitsize** field specifies how many contiguous bits are bundled into a single unit by the digitizer. The **sigbit** field specifies the order in which bits of significance are stored within each unit; most significant bit first or least significant bit first. The last of these three fields is the **byte_order** field. If **unitsize** is a multiple of bytes, then this field specifies the order in which bytes occur within the unit. Given these three attributes, data incompatibilities across computer hardware and data format assumptions within application software can be identified and effectively dealt with.

The **pix_offset** attribute defines a pixel displacement from the left edge of the raster image data to where a particular image's significant image information begins. The **whitepix** attribute defines the value assigned to the color white. For example, the gray scale image described in Figure 3 is gray print on a white background and the value of the white pixel is 255. This field is particularly useful to image display routines. The **issigned** field is required to specify whether the units of an image are signed or unsigned. This attribute determines whether an image with a pixel depth of 8, should have pixel values interpreted in the range of -128 to +127, or 0 to 255. The orientation of the raster scan may also vary among different digitizers. The attribute field, **rm_cm**, specifies whether the digitizer captured the image in row-major order or column-major order. Whether the scan lines of an image were accumulated from top to bottom, or bottom to top, is specified by the field, **tb_bt**, and whether left to right, or right to left, is specified by the field, **rl_lr**.

The final attributes in IHead provide a single historical link from the current image to its **parent** image. The images used in this database were mixed and renamed from their original filenames and the 'link' to the original filename was stored in the **parent** field. The **par_x** and **par_y** fields contain the origin, upper left hand corner pixel coordinate, from where the extraction took place from the parent image. These fields provide a historical thread through successive generations of images and subimages. We believe that the IHead image format contains the minimal amount of ancillary information required to successfully manage binary and gray scale images.

5.0 DATABASE CONTENT AND ORGANIZATION

NIST Special Database 4 contains 4000 8-bit gray scale fingerprint images stored in the **data** directory (see Figure 4). The images, which use approximately 636 Megabytes of storage compressed and 1.1 Gigabytes of storage uncompressed, are distributed on an ISO-9660 formatted CD-ROM and compressed using a modified JPEG lossless compression algorithm. Included with the fingerprint data are software and documentation.
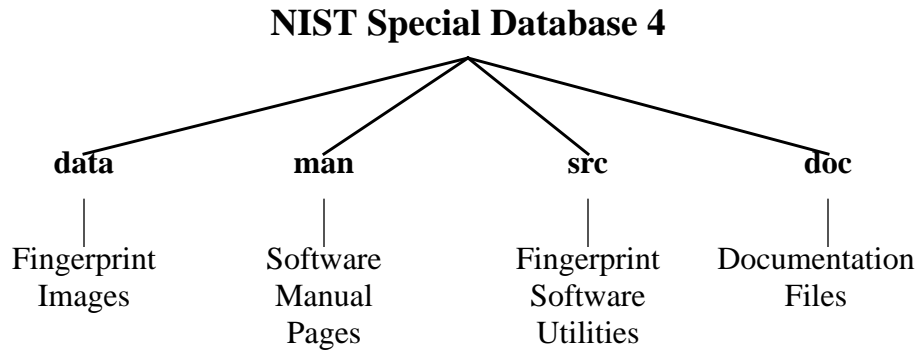
NIST Special Database 4

```
                        data        man        src        doc
                          |          |          |          |
                    Fingerprint   Software   Fingerprint  Documentation
                      Images      Manual      Software      Files
                                  Pages       Utilities
```

Figure 4: Top level directory tree for *NIST Special Database 4*.

5.1 Database File Hierarchy

The top level of the file structure contains four directories **doc**, **src**, **man**, and **data**. The code needed to decompress and use the image data is contained in the **src** directory with man pages for the source code stored in the **man** directories. Documentation for the CD-ROM is in the **doc** directory. The **data** directory contains the fingerprint images stored in 8 subdirectories for easier access (see Figure 5). Each subdirectory contains 250 fingerprint pairs (two different rollings of the same fingerprint). Fingerprints are stored with filenames containing one letter, four digits, an under score then two more digits, and a ".pct" extension. The first character in the filename is always an "f" or "s" distinguishing the first and second rollings of each fingerprint. The next four characters indicate the print number (i.e. 0001 - 2000), and the final two digits represent the finger number in the same order as on a fingerprint card (see Figure 6).
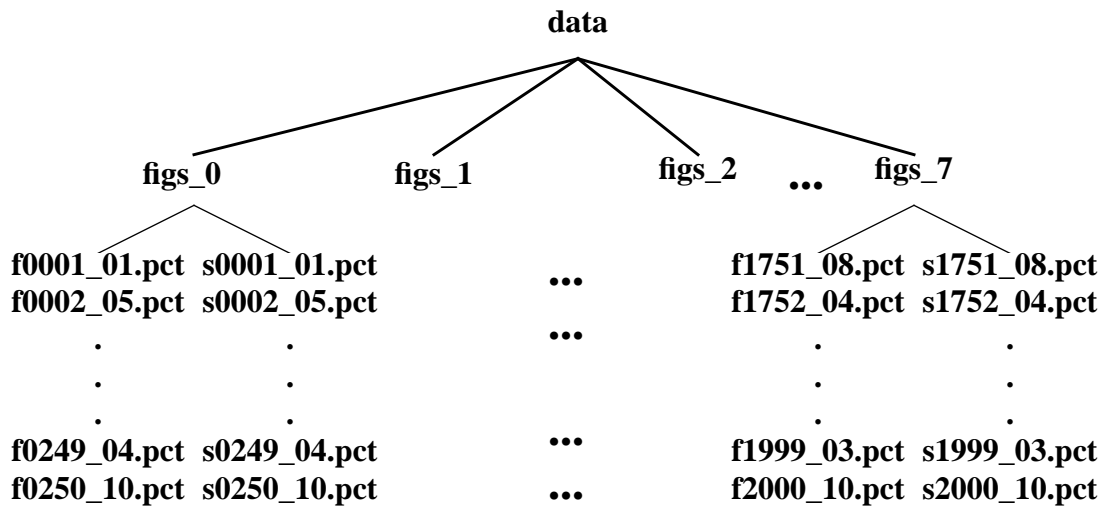
```
                                 data

           figs_0        figs_1       figs_2   ...   figs_7

     f0001_01.pct  s0001_01.pct              f1751_08.pct  s1751_08.pct
     f0002_05.pct  s0002_05.pct    ...       f1752_04.pct  s1752_04.pct
          .             .          ...            .             .
          .             .                         .             .
          .             .          ...            .             .
     f0249_04.pct  s0249_04.pct               f1999_03.pct  s1999_03.pct
     f0250_10.pct  s0250_10.pct    ...        f2000_10.pct  s2000_10.pct
```

Figure 5: Arrangement of fingerprint files for *NIST Special Database 4*.

1. R. Thumb	2. R. Index	3. R. Middle	4. R. Ring	5. R. Little
6. L. Thumb	7. L. Index	8. L. Middle	9. L. Ring	10. L. Little

Figure 6: Layout of fingerprint card numbers.

5.2 Cross-Referenced Fingerprints

The cross-referencing of a fingerprint is caused by a variety of ambiguities such as a scar occurring in the fingerprint, the quality of the print rolling, or the print having ridge structures characteristic of two different classes. The cross-referenced prints could easily cause a wrong classification when used in testing an automatic classification system, but could provide a challenge in the later stages of development. *NIST Special Database 4* contains 350 fingerprint pairs (17.5%) whose classifications are cross-referenced to a second classification (see Appendix C for a listing of files with cross-references). The **id** fields of these prints contain the fingerprint classifications immediately followed by the cross-referenced class. A good example is the first file listed in Appendix C (**f0008_10.pct**, shown in Figure A.2 of Appendix A). The fingerprint is classified as a tented arch; however, the ridge structure is also similar to that of a left loop so a cross-reference of left loop is given in this file's **id** field (see Figure 7).

.

```
                    IMAGE FILE HEADER
                    ~~~~~~~~~~~~~~~~~
                    Identity         : f0008_10.pct TL F
                    Header Size      : 288 (bytes)
                    Date Created     : THU FEB 20 10:10:47 1992
                    Width            : 512 (pixels)
                    Height           : 512 (pixels)
                    Bits per Pixel   : 8
                    Resolution       : 500 (ppi)
                    Compression      : 6 (code)
                    Compress Length  : 144320 (bytes)
                    Scan Alignment   : 8 (bits)
                    Image Data Unit  : 8 (bits)
                    Byte Order       : High-Low
                    MSBit            : First
                    Column Offset    : 0 (pixels)
                    White Pixel      : 255
                    Data Units       : Unsigned
                    Scan Order       : Row Major,
                                        Top to Bottom,
                                        Left to Right
                    Parent           : a221a.pct
                    X Origin         : 0 (pixels)
                    Y Origin         : 0 (pixels)
```

Figure 7: The IHead values for the fingerprint data file **f0008_10.pct.**

6.0 SOFTWARE FOR ACCESSING DATABASE

Included with the fingerprint images are documentation and software written in the 'C' programming language. Four programs are included in the **src** directory: **dumpihdr**, **ihdr2sun**, **sunalign**, and **dcplljpg**. These routines are provided as an example to software developers of how IHead images can be manipulated and used. Descriptions of these programs and their subroutines are given below as well as in the included man pages located in the **man** directory. Copies of the manual pages are also included in Appendix D.

6.1 Compilation

CD-ROM, *NIST Special Database 4*, is a read only storage medium. The files in the **src** directory must be copied to a read-writable partition prior to compiling. After copying these files, executable binaries can be produced by invoking the UNIX utility **make** to execute the included makefile. An example of this command follows.

make -f **makefile.mak**

6.2 Dumpihdr <Ihead file>

Dumpihdr is a program which reads an image's IHead data from the given file and formats the header data into a report which is printed to standard output. The report shown in Figure 3 was generated using this utility. The main routine for **dumpihdr** is found in the file **dumpihdr.c** and calls the external function **readihdr()**.

Readihdr() is a function responsible for loading an image's IHead data from a file into main memory. This routine allocates, reads, and returns the header information from an open image file in an initialized IHead structure. This function is found in the file **ihead.c**. The IHead structure definition is listed in Figure 2 and is found in the file **ihead.h**

6.3 Ihdr2sun <Ihead file>

Ihdr2sun converts an image from NIST IHead format to Sun rasterfile format. **Ihdr2sun** loads an IHead formatted image from a file into main memory and writes the raster data to a new file appending the data to a Sun rasterfile header. The main routine for this program is found in the file **ihdr2sun.c** and calls the external function **ReadIheadRaster()** which is found in the file **rasterio.c**.

ReadIheadRaster() is the procedure responsible for loading an IHead image from a file into main memory. This routine reads the image's header data returning an initialized IHead structure by calling **readihdr()**. In addition, the image's raster data is returned to the caller uncompressed. The images in this database have been 2-dimensionally compressed using a modified JPEG lossless compression algorithm, therefore **ReadIheadRaster()** invokes the external procedure **jpglldcp()** which is responsible for decompressing the raster data. Upon completion, **ReadIheadRaster()** returns an initialized IHead structure, the uncompressed raster data, the image's width and height in pixels, and pixel depth.

Jpglldcp() accepts image raster data compressed using the modified JPEG lossless compression algorithm and returns the uncompressed image raster data. **Jpglldcp()** was developed using techniques described in the WG10 "JPEG" (draft) standard [1] and adapted for use with this database. Source code for the algorithm is found in **jpglldcp.c**.

6.4 Dcplljpg <lossless JPEG compressed file>

Dcplljpg is a program which decompresses a fingerprint image file (approximately 4.5 seconds per image on a scientific workstation) that was compressed using the modified JPEG compression routine. The routine accepts a compressed image in NIST IHead format and writes the uncompressed image to the same filename using the NIST IHead format. The main routine is found in **dcplljpg.c** and calls the external functions **ReadIheadRaster()** (see section 6.3 for ReadIheadRaster description) and **writeihdrfile()**.

Writeihdrfile() is a routine that writes an IHead image into a file. This routine opens the passed filename and writes the given IHead structure and corresponding data to the file. **Writeihdrfile()** is found in the src file **rasterio.c**.

References

[1] WG10 "JPEG", committee draft ISO/IEC CD 10198-1, "Digital Compression and Coding of Continuous-Tone Still Images," March 3, 1991.

[2] M.D. Garris, "Design and Collection of a Handwriting Sample Image Database," Social Science Computing Journal, Vol. 10, to be published, 1992.

[3] National Bureau of Standards, "Standard Reference Materials," Reflection Step Table 2601.

Appendix A: Database Fingerprint Image Samples

Figure A.1: Fingerprint file **f0001_01.pct** from *NIST Special Database 4*.
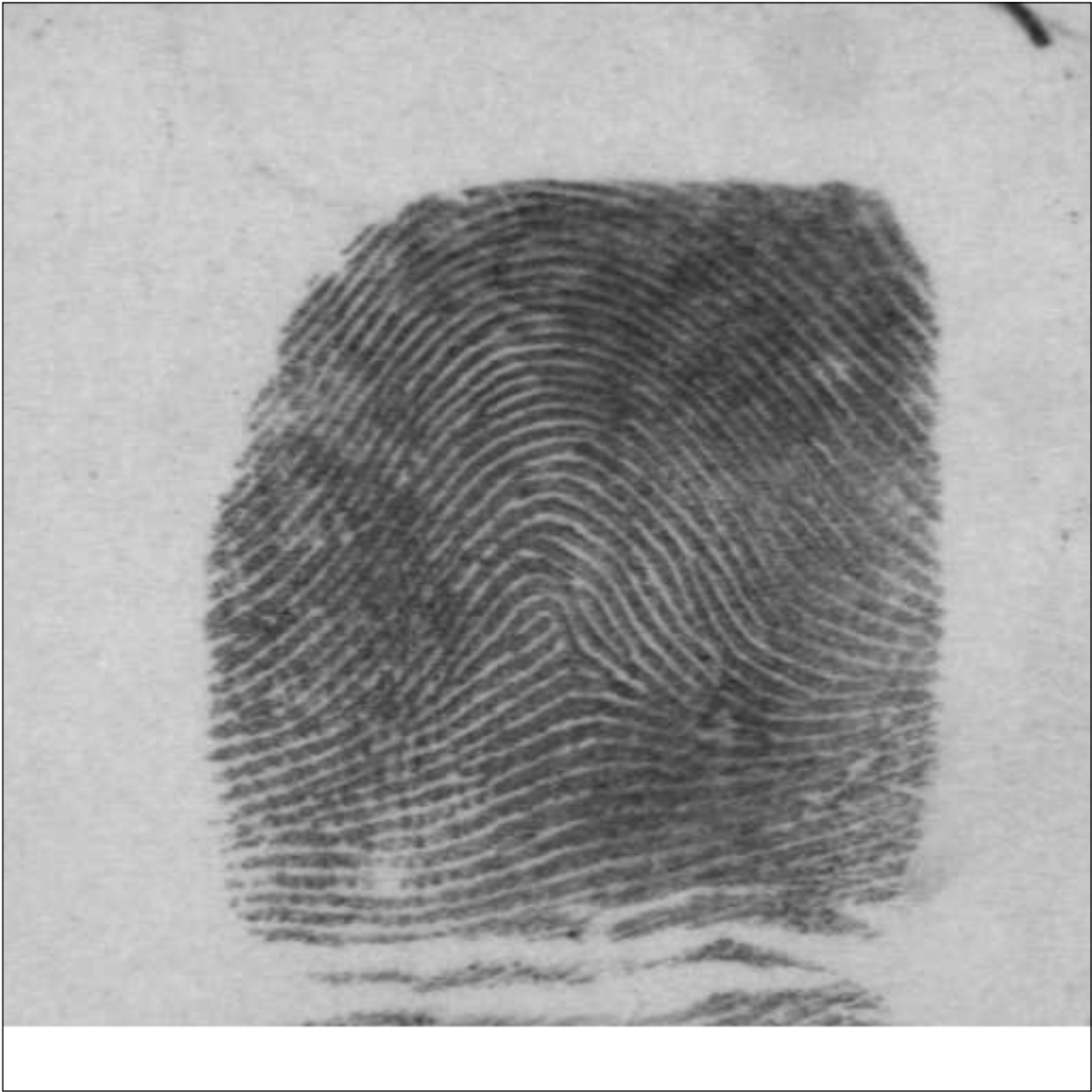
Figure A.2: Fingerprint file **f0008_10.pct** from *NIST Special Database 4*.

Appendix B: Database Reflectance and Resolution Calibration

Contact Craig Watson for Reflectance Plot
e-mail => craig@magi.ncsl.nist.gov
phone => (301) 975-4402

Database Resolution Calibration

The resolution of the scanner used to create the database was calibrated using a NBS 1010A resolution chart[1]. A portion of this image is shown below (Figure B.2) after being magnified for viewing (Note: Printing has reduced the overall quality of this image.). The scan of the table showed that the vertical resolution was approximately 8.0 line pairs/mm and the horizontal resolution was approximately 6.3 line pairs/mm.
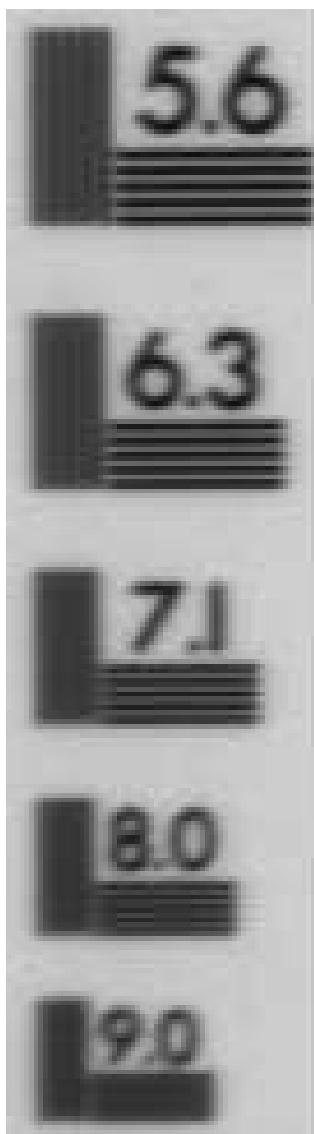
Figure B.2: Scan of Resolution Calibration Table.

1. National Bureau of Standards, Microcopy Resolution Test Chart, Standard Reference Material 1010a, ANSI and ISO test chart No. 2.

Appendix C: Chart of Referenced Fingerprints

f0008_10.pct (crid) = L :: s0008_10.pct (crid) = L
f0026_03.pct (crid) = R :: s0026_03.pct (crid) = R
f0028_08.pct (crid) = L :: s0028_08.pct (crid) = L
f0036_04.pct (crid) = R :: s0036_04.pct (crid) = R
f0038_02.pct (crid) = L :: s0038_02.pct (crid) = L
f0039_03.pct (crid) = R :: s0039_03.pct (crid) = R
f0047_01.pct (crid) = A :: s0047_01.pct (crid) = A
f0049_03.pct (crid) = R :: s0049_03.pct (crid) = R
f0050_01.pct (crid) = A :: s0050_01.pct (crid) = A
f0053_04.pct (crid) = R :: s0053_04.pct (crid) = R
f0054_03.pct (crid) = R :: s0054_03.pct (crid) = R
f0070_10.pct (crid) = T :: s0070_10.pct (crid) = T
f0073_07.pct (crid) = L :: s0073_07.pct (crid) = L
f0076_07.pct (crid) = R :: s0076_07.pct (crid) = R
f0087_03.pct (crid) = A :: s0087_03.pct (crid) = A
f0091_10.pct (crid) = T :: s0091_10.pct (crid) = T
f0103_10.pct (crid) = T :: s0103_10.pct (crid) = T
f0105_03.pct (crid) = R :: s0105_03.pct (crid) = R
f0108_03.pct (crid) = R :: s0108_03.pct (crid) = R
f0110_08.pct (crid) = T :: s0110_08.pct (crid) = T
f0112_02.pct (crid) = R :: s0112_02.pct (crid) = R
f0123_06.pct (crid) = T :: s0123_06.pct (crid) = T
f0126_10.pct (crid) = A :: s0126_10.pct (crid) = A
f0127_04.pct (crid) = R :: s0127_04.pct (crid) = R
f0132_07.pct (crid) = L :: s0132_07.pct (crid) = L
f0134_03.pct (crid) = R :: s0134_03.pct (crid) = R
f0140_08.pct (crid) = A :: s0140_08.pct (crid) = A
f0146_08.pct (crid) = L :: s0146_08.pct (crid) = L
f0154_08.pct (crid) = T :: s0154_08.pct (crid) = T
f0160_08.pct (crid) = A :: s0160_08.pct (crid) = A
f0163_03.pct (crid) = R :: s0163_03.pct (crid) = R
f0171_08.pct (crid) = T :: s0171_08.pct (crid) = T
f0172_08.pct (crid) = R :: s0172_08.pct (crid) = R
f0177_02.pct (crid) = R :: s0177_02.pct (crid) = R
f0178_02.pct (crid) = A :: s0178_02.pct (crid) = A
f0186_04.pct (crid) = A :: s0186_04.pct (crid) = A
f0187_08.pct (crid) = L :: s0187_08.pct (crid) = L
f0195_07.pct (crid) = L :: s0195_07.pct (crid) = L
f0207_05.pct (crid) = T :: s0207_05.pct (crid) = T
f0208_03.pct (crid) = A :: s0208_03.pct (crid) = A
f0210_09.pct (crid) = T :: s0210_09.pct (crid) = T
f0212_04.pct (crid) = T :: s0212_04.pct (crid) = T
f0217_02.pct (crid) = R :: s0217_02.pct (crid) = R
f0225_10.pct (crid) = L :: s0225_10.pct (crid) = L
f0229_08.pct (crid) = A :: s0229_08.pct (crid) = A
f0233_08.pct (crid) = L :: s0233_08.pct (crid) = L

f0241_04.pct (crid) = T :: s0241_04.pct (crid) = T
f0243_08.pct (crid) = A :: s0243_08.pct (crid) = A
f0246_02.pct (crid) = R :: s0246_02.pct (crid) = R
f0259_10.pct (crid) = T :: s0259_10.pct (crid) = T
f0260_02.pct (crid) = L :: s0260_02.pct (crid) = L
f0267_07.pct (crid) = T :: s0267_07.pct (crid) = T
f0268_09.pct (crid) = T :: s0268_09.pct (crid) = T
f0280_02.pct (crid) = R :: s0280_02.pct (crid) = R
f0282_03.pct (crid) = A :: s0282_03.pct (crid) = A
f0283_02.pct (crid) = A :: s0283_02.pct (crid) = A
f0295_07.pct (crid) = L :: s0295_07.pct (crid) = L
f0300_08.pct (crid) = A :: s0300_08.pct (crid) = A
f0303_03.pct (crid) = R :: s0303_03.pct (crid) = R
f0304_08.pct (crid) = A :: s0304_08.pct (crid) = A
f0314_04.pct (crid) = T :: s0314_04.pct (crid) = T
f0316_03.pct (crid) = R :: s0316_03.pct (crid) = R
f0317_05.pct (crid) = T :: s0317_05.pct (crid) = T
f0321_05.pct (crid) = A :: s0321_05.pct (crid) = A
f0344_06.pct (crid) = A :: s0344_06.pct (crid) = A
f0347_02.pct (crid) = L :: s0347_02.pct (crid) = L
f0348_08.pct (crid) = T :: s0348_08.pct (crid) = T
f0349_09.pct (crid) = T :: s0349_09.pct (crid) = T
f0351_08.pct (crid) = L :: s0351_08.pct (crid) = L
f0353_03.pct (crid) = R :: s0353_03.pct (crid) = R
f0355_08.pct (crid) = A :: s0355_08.pct (crid) = A
f0371_05.pct (crid) = R :: s0371_05.pct (crid) = R
f0386_02.pct (crid) = R :: s0386_02.pct (crid) = R
f0394_07.pct (crid) = A :: s0394_07.pct (crid) = A
f0400_02.pct (crid) = R :: s0400_02.pct (crid) = R
f0403_07.pct (crid) = A :: s0403_07.pct (crid) = A
f0406_04.pct (crid) = A :: s0406_04.pct (crid) = A
f0419_08.pct (crid) = L :: s0419_08.pct (crid) = L
f0423_02.pct (crid) = R :: s0423_02.pct (crid) = R
f0424_03.pct (crid) = R :: s0424_03.pct (crid) = R
f0429_04.pct (crid) = T :: s0429_04.pct (crid) = T
f0435_10.pct (crid) = A :: s0435_10.pct (crid) = A
f0456_06.pct (crid) = L :: s0456_06.pct (crid) = L
f0461_03.pct (crid) = R :: s0461_03.pct (crid) = R
f0466_10.pct (crid) = L :: s0466_10.pct (crid) = L
f0469_07.pct (crid) = A :: s0469_07.pct (crid) = A
f0486_10.pct (crid) = L :: s0486_10.pct (crid) = L
f0500_02.pct (crid) = T :: s0500_02.pct (crid) = T
f0509_07.pct (crid) = T :: s0509_07.pct (crid) = T
f0510_07.pct (crid) = L :: s0510_07.pct (crid) = L
f0513_07.pct (crid) = L :: s0513_07.pct (crid) = L
f0516_02.pct (crid) = R :: s0516_02.pct (crid) = R

f0537_09.pct (crid) = L :: s0537_09.pct (crid) = L
f0539_02.pct (crid) = R :: s0539_02.pct (crid) = R
f0540_08.pct (crid) = T :: s0540_08.pct (crid) = T
f0550_02.pct (crid) = R :: s0550_02.pct (crid) = R
f0551_08.pct (crid) = R :: s0551_08.pct (crid) = R
f0559_09.pct (crid) = T :: s0559_09.pct (crid) = T
f0560_08.pct (crid) = A :: s0560_08.pct (crid) = A
f0561_08.pct (crid) = A :: s0561_08.pct (crid) = A
f0568_07.pct (crid) = A :: s0568_07.pct (crid) = A
f0571_02.pct (crid) = A :: s0571_02.pct (crid) = A
f0574_09.pct (crid) = R :: s0574_09.pct (crid) = R
f0581_04.pct (crid) = R :: s0581_04.pct (crid) = R
f0586_02.pct (crid) = T :: s0586_02.pct (crid) = T
f0595_08.pct (crid) = A :: s0595_08.pct (crid) = A
f0598_08.pct (crid) = A :: s0598_08.pct (crid) = A
f0610_04.pct (crid) = R :: s0610_04.pct (crid) = R
f0616_08.pct (crid) = L :: s0616_08.pct (crid) = L
f0617_01.pct (crid) = A :: s0617_01.pct (crid) = A
f0624_03.pct (crid) = A :: s0624_03.pct (crid) = A
f0646_09.pct (crid) = T :: s0646_09.pct (crid) = T
f0659_10.pct (crid) = L :: s0659_10.pct (crid) = L
f0660_02.pct (crid) = R :: s0660_02.pct (crid) = R
f0664_03.pct (crid) = T :: s0664_03.pct (crid) = T
f0668_02.pct (crid) = R :: s0668_02.pct (crid) = R
f0674_04.pct (crid) = R :: s0674_04.pct (crid) = R
f0676_02.pct (crid) = R :: s0676_02.pct (crid) = R
f0684_02.pct (crid) = R :: s0684_02.pct (crid) = R
f0692_02.pct (crid) = L :: s0692_02.pct (crid) = L
f0693_03.pct (crid) = A :: s0693_03.pct (crid) = A
f0697_07.pct (crid) = L :: s0697_07.pct (crid) = L
f0698_08.pct (crid) = R :: s0698_08.pct (crid) = R
f0702_08.pct (crid) = L :: s0702_08.pct (crid) = L
f0703_02.pct (crid) = A :: s0703_02.pct (crid) = A
f0708_03.pct (crid) = R :: s0708_03.pct (crid) = R
f0713_03.pct (crid) = T :: s0713_03.pct (crid) = T
f0721_03.pct (crid) = R :: s0721_03.pct (crid) = R
f0727_07.pct (crid) = L :: s0727_07.pct (crid) = L
f0728_09.pct (crid) = A :: s0728_09.pct (crid) = A
f0731_02.pct (crid) = R :: s0731_02.pct (crid) = R
f0733_08.pct (crid) = L :: s0733_08.pct (crid) = L
f0741_03.pct (crid) = A :: s0741_03.pct (crid) = A
f0742_07.pct (crid) = L :: s0742_07.pct (crid) = L
f0743_07.pct (crid) = R :: s0743_07.pct (crid) = R
f0744_08.pct (crid) = R :: s0744_08.pct (crid) = R
f0746_10.pct (crid) = L :: s0746_10.pct (crid) = L
f0757_10.pct (crid) = T :: s0757_10.pct (crid) = T

f0758_07.pct (crid) = A :: s0758_07.pct (crid) = A
f0760_06.pct (crid) = L :: s0760_06.pct (crid) = L
f0765_07.pct (crid) = R :: s0765_07.pct (crid) = R
f0769_04.pct (crid) = T :: s0769_04.pct (crid) = T
f0771_03.pct (crid) = R :: s0771_03.pct (crid) = R
f0780_05.pct (crid) = R :: s0780_05.pct (crid) = R
f0781_02.pct (crid) = T :: s0781_02.pct (crid) = T
f0796_08.pct (crid) = A :: s0796_08.pct (crid) = A
f0804_03.pct (crid) = A :: s0804_03.pct (crid) = A
f0806_04.pct (crid) = T :: s0806_04.pct (crid) = T
f0844_03.pct (crid) = R :: s0844_03.pct (crid) = R
f0850_07.pct (crid) = A :: s0850_07.pct (crid) = A
f0856_02.pct (crid) = R :: s0856_02.pct (crid) = R
f0862_08.pct (crid) = L :: s0862_08.pct (crid) = L
f0864_05.pct (crid) = R :: s0864_05.pct (crid) = R
f0868_07.pct (crid) = R :: s0868_07.pct (crid) = R
f0877_02.pct (crid) = T :: s0877_02.pct (crid) = T
f0884_03.pct (crid) = R :: s0884_03.pct (crid) = R
f0891_07.pct (crid) = L :: s0891_07.pct (crid) = L
f0894_06.pct (crid) = A :: s0894_06.pct (crid) = A
f0897_08.pct (crid) = R :: s0897_08.pct (crid) = R
f0905_03.pct (crid) = R :: s0905_03.pct (crid) = R
f0909_07.pct (crid) = L :: s0909_07.pct (crid) = L
f0910_07.pct (crid) = R :: s0910_07.pct (crid) = R
f0916_09.pct (crid) = T :: s0916_09.pct (crid) = T
f0922_07.pct (crid) = L :: s0922_07.pct (crid) = L
f0934_10.pct (crid) = W :: s0934_10.pct (crid) = W
f0938_06.pct (crid) = T :: s0938_06.pct (crid) = T
f0941_02.pct (crid) = R :: s0941_02.pct (crid) = R
f0948_02.pct (crid) = T :: s0948_02.pct (crid) = T
f0950_02.pct (crid) = R :: s0950_02.pct (crid) = R
f0958_09.pct (crid) = L :: s0958_09.pct (crid) = L
f0959_07.pct (crid) = R :: s0959_07.pct (crid) = R
f0964_03.pct (crid) = R :: s0964_03.pct (crid) = R
f0983_03.pct (crid) = T :: s0983_03.pct (crid) = T
f0993_08.pct (crid) = A :: s0993_08.pct (crid) = A
f0994_04.pct (crid) = A :: s0994_04.pct (crid) = A
f0998_06.pct (crid) = T :: s0998_06.pct (crid) = T
f1000_07.pct (crid) = A :: s1000_07.pct (crid) = A
f1001_05.pct (crid) = R :: s1001_05.pct (crid) = R
f1003_04.pct (crid) = R :: s1003_04.pct (crid) = R
f1008_08.pct (crid) = A :: s1008_08.pct (crid) = A
f1010_02.pct (crid) = R :: s1010_02.pct (crid) = R
f1042_03.pct (crid) = R :: s1042_03.pct (crid) = R
f1045_04.pct (crid) = R :: s1045_04.pct (crid) = R
f1074_04.pct (crid) = R :: s1074_04.pct (crid) = R

f1082_08.pct (crid) = L :: s1082_08.pct (crid) = L
f1083_02.pct (crid) = A :: s1083_02.pct (crid) = A
f1087_02.pct (crid) = R :: s1087_02.pct (crid) = R
f1088_03.pct (crid) = R :: s1088_03.pct (crid) = R
f1095_09.pct (crid) = T :: s1095_09.pct (crid) = T
f1098_02.pct (crid) = R :: s1098_02.pct (crid) = R
f1099_05.pct (crid) = T :: s1099_05.pct (crid) = T
f1106_07.pct (crid) = A :: s1106_07.pct (crid) = A
f1115_03.pct (crid) = R :: s1115_03.pct (crid) = R
f1119_02.pct (crid) = R :: s1119_02.pct (crid) = R
f1123_08.pct (crid) = L :: s1123_08.pct (crid) = L
f1138_10.pct (crid) = T :: s1138_10.pct (crid) = T
f1143_03.pct (crid) = R :: s1143_03.pct (crid) = R
f1146_03.pct (crid) = R :: s1146_03.pct (crid) = R
f1149_09.pct (crid) = R :: s1149_09.pct (crid) = R
f1163_03.pct (crid) = L :: s1163_03.pct (crid) = L
f1174_07.pct (crid) = R :: s1174_07.pct (crid) = R
f1179_10.pct (crid) = L :: s1179_10.pct (crid) = L
f1181_10.pct (crid) = A :: s1181_10.pct (crid) = A
f1182_02.pct (crid) = A :: s1182_02.pct (crid) = A
f1185_03.pct (crid) = R :: s1185_03.pct (crid) = R
f1186_07.pct (crid) = L :: s1186_07.pct (crid) = L
f1190_04.pct (crid) = T :: s1190_04.pct (crid) = T
f1203_04.pct (crid) = R :: s1203_04.pct (crid) = R
f1206_02.pct (crid) = R :: s1206_02.pct (crid) = R
f1207_02.pct (crid) = R :: s1207_02.pct (crid) = R
f1214_02.pct (crid) = L :: s1214_02.pct (crid) = L
f1220_02.pct (crid) = R :: s1220_02.pct (crid) = R
f1228_02.pct (crid) = A :: s1228_02.pct (crid) = A
f1230_04.pct (crid) = T :: s1230_04.pct (crid) = T
f1232_09.pct (crid) = T :: s1232_09.pct (crid) = T
f1233_04.pct (crid) = W :: s1233_04.pct (crid) = W
f1234_08.pct (crid) = L :: s1234_08.pct (crid) = L
f1238_09.pct (crid) = T :: s1238_09.pct (crid) = T
f1240_03.pct (crid) = R :: s1240_03.pct (crid) = R
f1249_08.pct (crid) = L :: s1249_08.pct (crid) = L
f1255_07.pct (crid) = R :: s1255_07.pct (crid) = R
f1261_07.pct (crid) = A :: s1261_07.pct (crid) = A
f1268_02.pct (crid) = R :: s1268_02.pct (crid) = R
f1277_08.pct (crid) = T :: s1277_08.pct (crid) = T
f1292_07.pct (crid) = T :: s1292_07.pct (crid) = T
f1294_07.pct (crid) = L :: s1294_07.pct (crid) = L
f1300_08.pct (crid) = L :: s1300_08.pct (crid) = L
f1306_08.pct (crid) = L :: s1306_08.pct (crid) = L
f1316_09.pct (crid) = T :: s1316_09.pct (crid) = T
f1317_07.pct (crid) = L :: s1317_07.pct (crid) = L

f1323_03.pct (crid) = A :: s1323_03.pct (crid) = A
f1326_07.pct (crid) = A :: s1326_07.pct (crid) = A
f1330_03.pct (crid) = T :: s1330_03.pct (crid) = T
f1333_08.pct (crid) = L :: s1333_08.pct (crid) = L
f1335_08.pct (crid) = L :: s1335_08.pct (crid) = L
f1341_03.pct (crid) = R :: s1341_03.pct (crid) = R
f1360_02.pct (crid) = R :: s1360_02.pct (crid) = R
f1363_08.pct (crid) = R :: s1363_08.pct (crid) = R
f1366_06.pct (crid) = L :: s1366_06.pct (crid) = L
f1367_02.pct (crid) = R :: s1367_02.pct (crid) = R
f1371_08.pct (crid) = R :: s1371_08.pct (crid) = R
f1384_07.pct (crid) = R :: s1384_07.pct (crid) = R
f1395_02.pct (crid) = T :: s1395_02.pct (crid) = T
f1396_07.pct (crid) = L :: s1396_07.pct (crid) = L
f1399_06.pct (crid) = A :: s1399_06.pct (crid) = A
f1413_07.pct (crid) = R :: s1413_07.pct (crid) = R
f1433_07.pct (crid) = R :: s1433_07.pct (crid) = R
f1437_07.pct (crid) = L :: s1437_07.pct (crid) = L
f1442_07.pct (crid) = R :: s1442_07.pct (crid) = R
f1448_10.pct (crid) = T :: s1448_10.pct (crid) = T
f1449_07.pct (crid) = A :: s1449_07.pct (crid) = A
f1455_03.pct (crid) = R :: s1455_03.pct (crid) = R
f1472_08.pct (crid) = L :: s1472_08.pct (crid) = L
f1479_10.pct (crid) = A :: s1479_10.pct (crid) = A
f1484_02.pct (crid) = A :: s1484_02.pct (crid) = A
f1486_03.pct (crid) = R :: s1486_03.pct (crid) = R
f1498_03.pct (crid) = L :: s1498_03.pct (crid) = L
f1502_04.pct (crid) = R :: s1502_04.pct (crid) = R
f1508_10.pct (crid) = L :: s1508_10.pct (crid) = L
f1513_06.pct (crid) = L :: s1513_06.pct (crid) = L
f1514_01.pct (crid) = A :: s1514_01.pct (crid) = A
f1515_02.pct (crid) = A :: s1515_02.pct (crid) = A
f1518_02.pct (crid) = L :: s1518_02.pct (crid) = L
f1523_05.pct (crid) = W :: s1523_05.pct (crid) = W
f1524_08.pct (crid) = R :: s1524_08.pct (crid) = R
f1536_02.pct (crid) = L :: s1536_02.pct (crid) = L
f1543_03.pct (crid) = R :: s1543_03.pct (crid) = R
f1544_05.pct (crid) = R :: s1544_05.pct (crid) = R
f1553_08.pct (crid) = L :: s1553_08.pct (crid) = L
f1558_07.pct (crid) = R :: s1558_07.pct (crid) = R
f1560_02.pct (crid) = L :: s1560_02.pct (crid) = L
f1570_09.pct (crid) = T :: s1570_09.pct (crid) = T
f1575_08.pct (crid) = L :: s1575_08.pct (crid) = L
f1576_06.pct (crid) = L :: s1576_06.pct (crid) = L
f1584_03.pct (crid) = R :: s1584_03.pct (crid) = R
f1588_07.pct (crid) = R :: s1588_07.pct (crid) = R

f1592_02.pct (crid) = R :: s1592_02.pct (crid) = R
f1593_04.pct (crid) = T :: s1593_04.pct (crid) = T
f1599_04.pct (crid) = T :: s1599_04.pct (crid) = T
f1603_08.pct (crid) = A :: s1603_08.pct (crid) = A
f1606_03.pct (crid) = A :: s1606_03.pct (crid) = A
f1607_08.pct (crid) = A :: s1607_08.pct (crid) = A
f1629_08.pct (crid) = L :: s1629_08.pct (crid) = L
f1634_09.pct (crid) = L :: s1634_09.pct (crid) = L
f1635_02.pct (crid) = L :: s1635_02.pct (crid) = L
f1639_06.pct (crid) = A :: s1639_06.pct (crid) = A
f1663_08.pct (crid) = L :: s1663_08.pct (crid) = L
f1669_03.pct (crid) = R :: s1669_03.pct (crid) = R
f1674_02.pct (crid) = L :: s1674_02.pct (crid) = L
f1679_09.pct (crid) = T :: s1679_09.pct (crid) = T
f1682_05.pct (crid) = R :: s1682_05.pct (crid) = R
f1684_03.pct (crid) = T :: s1684_03.pct (crid) = T
f1689_08.pct (crid) = R :: s1689_08.pct (crid) = R
f1693_08.pct (crid) = L :: s1693_08.pct (crid) = L
f1699_07.pct (crid) = A :: s1699_07.pct (crid) = A
f1702_03.pct (crid) = R :: s1702_03.pct (crid) = R
f1719_02.pct (crid) = A :: s1719_02.pct (crid) = A
f1726_02.pct (crid) = R :: s1726_02.pct (crid) = R
f1729_02.pct (crid) = A :: s1729_02.pct (crid) = A
f1732_04.pct (crid) = R :: s1732_04.pct (crid) = R
f1733_02.pct (crid) = R :: s1733_02.pct (crid) = R
f1736_08.pct (crid) = L :: s1736_08.pct (crid) = L
f1740_07.pct (crid) = T :: s1740_07.pct (crid) = T
f1745_09.pct (crid) = L :: s1745_09.pct (crid) = L
f1748_02.pct (crid) = L :: s1748_02.pct (crid) = L
f1751_08.pct (crid) = A :: s1751_08.pct (crid) = A
f1759_09.pct (crid) = L :: s1759_09.pct (crid) = L
f1763_10.pct (crid) = T :: s1763_10.pct (crid) = T
f1766_08.pct (crid) = A :: s1766_08.pct (crid) = A
f1772_04.pct (crid) = R :: s1772_04.pct (crid) = R
f1773_07.pct (crid) = A :: s1773_07.pct (crid) = A
f1791_03.pct (crid) = R :: s1791_03.pct (crid) = R
f1793_07.pct (crid) = R :: s1793_07.pct (crid) = R
f1801_03.pct (crid) = A :: s1801_03.pct (crid) = A
f1809_08.pct (crid) = T :: s1809_08.pct (crid) = T
f1816_03.pct (crid) = R :: s1816_03.pct (crid) = R
f1817_07.pct (crid) = R :: s1817_07.pct (crid) = R
f1831_08.pct (crid) = L :: s1831_08.pct (crid) = L
f1834_02.pct (crid) = L :: s1834_02.pct (crid) = L
f1836_02.pct (crid) = L :: s1836_02.pct (crid) = L
f1843_10.pct (crid) = L :: s1843_10.pct (crid) = L
f1861_07.pct (crid) = R :: s1861_07.pct (crid) = R

f1862_09.pct (crid) = T :: s1862_09.pct (crid) = T
f1864_07.pct (crid) = A :: s1864_07.pct (crid) = A
f1868_02.pct (crid) = L :: s1868_02.pct (crid) = L
f1878_05.pct (crid) = R :: s1878_05.pct (crid) = R
f1879_02.pct (crid) = R :: s1879_02.pct (crid) = R
f1882_03.pct (crid) = R :: s1882_03.pct (crid) = R
f1885_03.pct (crid) = R :: s1885_03.pct (crid) = R
f1897_07.pct (crid) = L :: s1897_07.pct (crid) = L
f1905_08.pct (crid) = A :: s1905_08.pct (crid) = A
f1916_09.pct (crid) = A :: s1916_09.pct (crid) = A
f1925_03.pct (crid) = T :: s1925_03.pct (crid) = T
f1928_02.pct (crid) = L :: s1928_02.pct (crid) = L
f1931_01.pct (crid) = R :: s1931_01.pct (crid) = R
f1932_09.pct (crid) = T :: s1932_09.pct (crid) = T
f1934_07.pct (crid) = L :: s1934_07.pct (crid) = L
f1945_10.pct (crid) = L :: s1945_10.pct (crid) = L
f1947_07.pct (crid) = A :: s1947_07.pct (crid) = A
f1952_02.pct (crid) = A :: s1952_02.pct (crid) = A
f1953_03.pct (crid) = A :: s1953_03.pct (crid) = A
f1960_06.pct (crid) = T :: s1960_06.pct (crid) = T
f1966_07.pct (crid) = R :: s1966_07.pct (crid) = R
f1969_03.pct (crid) = T :: s1969_03.pct (crid) = T
f1971_04.pct (crid) = T :: s1971_04.pct (crid) = T
f1972_03.pct (crid) = R :: s1972_03.pct (crid) = R
f1975_02.pct (crid) = R :: s1975_02.pct (crid) = R
f1981_08.pct (crid) = T :: s1981_08.pct (crid) = T
f1984_08.pct (crid) = L :: s1984_08.pct (crid) = L
f2000_10.pct (crid) = L :: s2000_10.pct (crid) = L

Appendix D: Manual Pages for Database Source Code

Contact Craig Watson for Manual Pages
e-mail => craig@magi.ncsl.nist.gov
phone => (301) 975-4402