# *NIST Special Database 2*
# Structured Forms Database

**D. L. Dimmick, M. D. Garris, and C. L. Wilson**

National Institute of Standards and Technology
Advanced Systems Division
Image Recognition Group
December 1, 1991

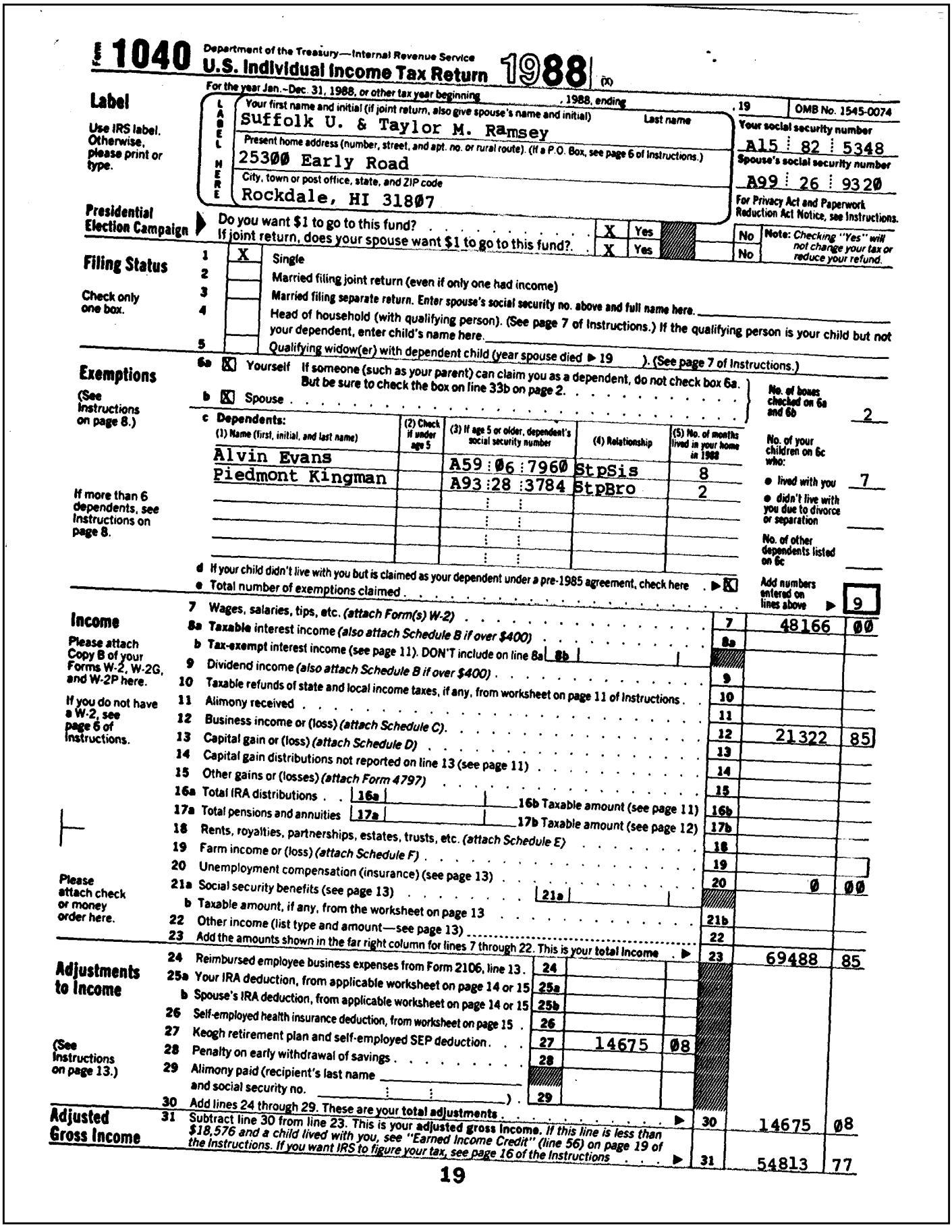
## 1.0 Introduction

This report describes the NIST Structured Forms Reference Set database, *NIST Special Database 2*, containing binary images of synthesized documents. Databases of this magnitude are necessary to further the research and development of automated document processing systems. This database is being distributed as a reference data set to be used by developers of document recognition and data capture systems to test and report results on a common corpus of images derived from structured forms containing machine printed data. The structured forms used in this database are 12 different tax forms from the IRS 1040 Package X for the year 1988. These include Forms 1040, 2106, 2441, 4562, and 6251 together with Schedules A, B, C, D, E, F, and SE. Eight of these forms contain two pages or form faces making a total of 20 different form faces represented in the database.

The database contains 5,590 full page images of completed tax forms. Each image is stored in the bilevel black and white raster format defined in Section 2.2. The images in this database appear to be real forms prepared by individuals but the images have been automatically derived and synthesized using a computer.

## 2.0 Image Synthesis

The entry field values on these forms have been automatically generated by a computer in order to make the data available without the danger of distributing privileged tax information. The computer derived entry field values are synthesized as images from one or more fonts of machine printed data. An image of an entry field value is produced by combining images of each character in the value. An entry field image is then inserted in a selected location within the corresponding field within a form image. The image data entered in a field in this way has been translated and rotated by small amounts to simulate effects of imperfect printing and imperfect alignment of a form in the printing device. Multiple examples of the digital representation of each character are used so that the pattern of the binary pixels representing each character is not consistently replicated but varies as it would in a sample of real tax forms. Both the form templates and the character examples are digitized at 300 pixels per inch binary. Figure 1 displays a synthesized tax form.



Form 1040 Department of the Treasury—Internal Revenue Service U.S. Individual Income Tax Return 1988 (X)

For the year Jan.–Dec. 31, 1988, or other tax year beginning , 1988, ending , 19 OMB No. 1545-0074

Label — Use IRS label. Otherwise, please print or type.

Your first name and initial (if joint return, also give spouse's name and initial) Last name: Suffolk U. & Taylor M. Ramsey

Present home address (number, street, and apt. no. or rural route). (If a P.O. Box, see page 6 of Instructions.): 25300 Early Road

City, town or post office, state, and ZIP code: Rockdale, HI 31807

Your social security number: A15 : 82 : 5348

Spouse's social security number: A99 : 26 : 9320

For Privacy Act and Paperwork Reduction Act Notice, see Instructions.

Presidential Election Campaign — Do you want $1 to go to this fund? X Yes No. If joint return, does your spouse want $1 to go to this fund? X Yes No. Note: Checking "Yes" will not change your tax or reduce your refund.

Filing Status — Check only one box.

1 X Single
2 Married filing joint return (even if only one had income)
3 Married filing separate return. Enter spouse's social security no. above and full name here.
4 Head of household (with qualifying person). (See page 7 of Instructions.) If the qualifying person is your child but not your dependent, enter child's name here.
5 Qualifying widow(er) with dependent child (year spouse died ▶ 19 ). (See page 7 of Instructions.)

Exemptions (See Instructions on page 8.)

6a X Yourself. If someone (such as your parent) can claim you as a dependent, do not check box 6a. But be sure to check the box on line 33b on page 2.
b X Spouse
c Dependents:

| (1) Name (first, initial, and last name) | (2) Check if under age 5 | (3) If age 5 or older, dependent's social security number | (4) Relationship | (5) No. of months lived in your home in 1988 |
|---|---|---|---|---|
| Alvin Evans | | A59 : 06 : 7960 | StpSis | 8 |
| Piedmont Kingman | | A93 : 28 : 3784 | StpBro | 2 |
| | | : : | | |
| | | : : | | |
| | | : : | | |
| | | : : | | |

If more than 6 dependents, see Instructions on page 8.

No. of boxes checked on 6a and 6b: 2
No. of your children on 6c who: • lived with you: 7 • didn't live with you due to divorce or separation:
No. of other dependents listed on 6c:
d If your child didn't live with you but is claimed as your dependent under a pre-1985 agreement, check here ▶ X
e Total number of exemptions claimed. Add numbers entered on lines above ▶: 9

Income — Please attach Copy B of your Forms W-2, W-2G, and W-2P here. If you do not have a W-2, see page 6 of Instructions.

| | | | |
|---|---|---|---|
| 7 | Wages, salaries, tips, etc. (attach Form(s) W-2) | 7 | 48166 00 |
| 8a | Taxable interest income (also attach Schedule B if over $400) | 8a | |
| b | Tax-exempt interest income (see page 11). DON'T include on line 8a 8b | | |
| 9 | Dividend income (also attach Schedule B if over $400) | 9 | |
| 10 | Taxable refunds of state and local income taxes, if any, from worksheet on page 11 of Instructions | 10 | |
| 11 | Alimony received | 11 | |
| 12 | Business income or (loss) (attach Schedule C) | 12 | 21322 85 |
| 13 | Capital gain or (loss) (attach Schedule D) | 13 | |
| 14 | Capital gain distributions not reported on line 13 (see page 11) | 14 | |
| 15 | Other gains or (losses) (attach Form 4797) | 15 | |
| 16a | Total IRA distributions 16a 16b Taxable amount (see page 11) | 16b | |
| 17a | Total pensions and annuities 17a 17b Taxable amount (see page 12) | 17b | |
| 18 | Rents, royalties, partnerships, estates, trusts, etc. (attach Schedule E) | 18 | |
| 19 | Farm income or (loss) (attach Schedule F) | 19 | |
| 20 | Unemployment compensation (insurance) (see page 13) | 20 | 0 00 |
| 21a | Social security benefits (see page 13) 21a | | |
| b | Taxable amount, if any, from the worksheet on page 13 | 21b | |
| 22 | Other income (list type and amount—see page 13) | 22 | |
| 23 | Add the amounts shown in the far right column for lines 7 through 22. This is your total income ▶ | 23 | 69488 85 |

Please attach check or money order here.

Adjustments to Income (See Instructions on page 13.)

| | | | | |
|---|---|---|---|---|
| 24 | Reimbursed employee business expenses from Form 2106, line 13 | 24 | | |
| 25a | Your IRA deduction, from applicable worksheet on page 14 or 15 | 25a | | |
| b | Spouse's IRA deduction, from applicable worksheet on page 14 or 15 | 25b | | |
| 26 | Self-employed health insurance deduction, from worksheet on page 15 | 26 | | |
| 27 | Keogh retirement plan and self-employed SEP deduction | 27 | 14675 08 | |
| 28 | Penalty on early withdrawal of savings | 28 | | |
| 29 | Alimony paid (recipient's last name and social security no. ) | 29 | | |
| 30 | Add lines 24 through 29. These are your total adjustments ▶ | 30 | 14675 08 | |

Adjusted Gross Income

| | | | |
|---|---|---|---|
| 31 | Subtract line 30 from line 23. This is your adjusted gross income. If this line is less than $18,576 and a child lived with you, see "Earned Income Credit" (line 56) on page 19 of the Instructions. If you want IRS to figure your tax, see page 16 of the Instructions ▶ | 31 | 54813 77 |

19

FIGURE 1. A representative image file of a completed form in *NIST Special Database 2.*

## 2.1 Answer File Formats

| | |
|---|---|
| 1040_1 | |
| 1040_1_L_H1_V1 | |
| 1040_1_L_H2_V1 | |
| 1040_1_L_H3_V1 | |
| 1040_1_L_H1_V2 | Suffolk U. & Taylor M. Ramsey |
| 1040_1_L_H2_V2 | A15 82 5348 |
| 1040_1_L_H1_V3 | 25300 Early Road |
| 1040_1_L_H2_V3 | A99 26 9320 |
| 1040_1_L_H1_V4 | Rockdale, HI 31807 |
| 1040_1_L_H1_V5 | _ICON_ |
| 1040_1_L_H2_V5 | |
| 1040_1_L_H1_V6 | _ICON_ |
| 1040_1_L_H2_V6 | |
| 1040_1_1 | _ICON_ |
| 1040_1_2 | |
| 1040_1_3_H1 | |
| 1040_1_3_H2 | |
| 1040_1_4_H1 | |
| 1040_1_4_H2 | |
| 1040_1_5_H1 | |
| 1040_1_5_H2 | |
| 1040_1_6a_H1 | _ICON_ |
| 1040_1_6b_H1 | _ICON_ |
| 1040_1_6b_H2 | 2 |
| 1040_1_6c_H1_V1 | Alvin Evans |
| 1040_1_6c_H2_V1 | |
| 1040_1_6c_H3_V1 | A59 06 7960 |
| 1040_1_6c_H4_V1 | StpSis |
| 1040_1_6c_H5_V1 | 8 |
| 1040_1_6c_H6_V1 | 7 |
| 1040_1_6c_H1_V2 | Piedmont Kingman |
| 1040_1_6c_H2_V2 | |
| 1040_1_6c_H3_V2 | A93 28 3784 |
| 1040_1_6c_H4_V2 | StpBro |
| 1040_1_6c_H5_V2 | 2 |
| 1040_1_6c_H6_V2 | |
| 1040_1_6c_H1_V3 | |
| 1040_1_6c_H2_V3 | |
| 1040_1_6c_H3_V3 | |
| 1040_1_6c_H4_V3 | |
| 1040_1_6c_H5_V3 | |
| 1040_1_6c_H6_V3 | |
| 1040_1_6c_H1_V4 | |
| 1040_1_6c_H2_V4 | |
| 1040_1_6c_H3_V4 | |
| 1040_1_6c_H4_V4 | |
| 1040_1_6c_H5_V4 | |
| 1040_1_6c_H1_V5 | |
| 1040_1_6c_H2_V5 | |
| 1040_1_6c_H3_V5 | |
| 1040_1_6c_H4_V5 | |
| 1040_1_6c_H5_V5 | |
| 1040_1_6c_H1_V6 | |
| 1040_1_6c_H2_V6 | |
| 1040_1_6c_H3_V6 | |
| 1040_1_6c_H4_V6 | |
| 1040_1_6c_H5_V6 | |
| 1040_1_6d | _ICON_ |
| 1040_1_6e 9 | |
| 1040_1_7 | 48166 00 |
| 1040_1_8a | |
| 1040_1_8b | |
| 1040_1_9 | |
| 1040_1_10 | |
| 1040_1_11 | |
| 1040_1_12 | 21322 85 |
| 1040_1_13 | |
| 1040_1_14 | |
| 1040_1_15 | |
| 1040_1_16a | |
| 1040_1_16b | |
| 1040_1_17a | |
| 1040_1_17b | |
| 1040_1_18 | |
| 1040_1_19 | |
| 1040_1_20 | 0 00 |
| 1040_1_21a | |
| 1040_1_21b | |
| 1040_1_22_H1 | |
| 1040_1_22_H2 | |
| 1040_1_23 | 69488 85 |
| 1040_1_24 | |
| 1040_1_25a | |
| 1040_1_25b | |
| 1040_1_26 | |
| 1040_1_27 | 14675 08 |
| 1040_1_28 | |
| 1040_1_29_V1 | |
| 1040_1_29_H1_V2 | |
| 1040_1_29_H2_V2 | |
| 1040_1_30 | 14675 08 |
| 1040_1_31 | 54813 77 |

FIGURE 2. The answer file for the image shown in Figure 1.

The values entered on the forms have been derived by a computer. These entry field values are stored separately from the image in an ASCII text file. This text file, one per completed structured form image, serves as an answer file which can be used to score the values hypothesized by a recognition system. An example of one of the answer files in the database is listed in Figure 2. These text files are the ground truth against which recognition responses may be compared.

The information in Figure 2 has been listed in two adjacent text columns. The first line in this file contains the identification of the form face in the referenced image. *NIST Special Database 2* contains multiple form faces and therefore can be used for testing the forms identification ability of a document recognition system. The form type identification can be used to compute a system's accuracy in correctly identifying the form face contained in an image.

Each successive line in the answer file is an entry field identification and entry field value pair. The field identification string uniquely identifies which entry field is being referenced on a structured form. The field identifications used in this database are labeled on the form faces contained in Appendix A. The entry field value may be empty or it may contain a computer derived value. Empty entry field values model sparsely filled forms. An entry field value containing the token string "_ICON_" represents the existence of non-character information. Examples of this type of information includes box check marks and signatures. Any other value listed for an entry field references the precise character information entered into the form image.

### 2.2 Image File format

Image file formats and effective data compression and decompression are critical to the usefulness of image archives. Each page of a completed form face was synthesized at 300 dots per inch binary, 2-dimensionally compressed using CCITT Group 4, and temporarily archived onto computer magnetic mass storage. Once all forms were synthesized, the images were mastered and replicated onto ISO-9660 formatted CD-ROM discs for permanent archiving and distribution.

In this application, a raster image is a digital encoding of light reflected from discrete points on a scanned form. The 2-dimensional area of the form is divided into discrete locations according to the resolution of a specified grid. Each cell of this grid is represented by a single bit value 0 or 1 called a pixel; 0 represents a cell predominately white, 1 represents a cell predominately black. This 2-dimensional sampling grid is then stored as a 1-dimensional vector of pixel values in raster order, left to right, top to bottom. Successive scan lines (top to bottom), contain the values of a single row of pixels from the grid concatenated together.

After digitization, certain attributes of an image are required to be known 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 is used to manipulate the raster data of an image, is able to first read the header and determine the proper interpretation of the data which follows it. Figure 3 illustrates this file format.

A header format named IHead has been developed for use as an image interchange format. Numerous image formats exist; some are widely supported on small personal computers, others supported on larger workstations; most are proprietary formats, few are public domain. 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 ex-

tensive 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.

IHead has been successfully ported and tested on several systems including UNIX workstations and servers, DOS personal computers, and VMS mainframes. The attribute fields in IHead can be loaded into main memory in two distinct ways. Since the attributes are represented by the ASCII character set, the attribute fields may be parsed as null-terminated strings, an input/output format common in the ‘C’ programming language. IHead can also be read into main memory using record-oriented input/output. The fixed length of the header is prefixed to the front of the header as shown in Figure 3. The IHead structure definition as written in the ‘C’ programming language is listed in Figure 4.

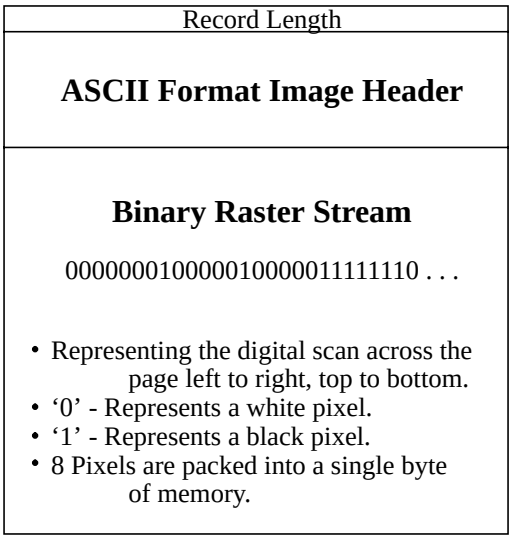



FIGURE 3. An illustration of the IHead raster file format.

```
/*************************************************************
        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 4. IHead C language definition.

Figure 5 lists the header values from an IHead file corresponding to the structure members listed in Figure 4. This header information belongs to the isolated box image displayed in Figure 6. Referencing the structure members listed in Figure 4, 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 reference string the writer was instructed to print in the box. The reference string is delimited by double quotes.

IMAGE FILE HEADER
~~~~~~~~~~~~~~~~~~

| | |
|---|---|
| Identity | : box_03.pct “0123456789” |
| Header Size | : 288 (bytes) |
| Date Created | : Thu Jan 4 17:34:21 1990 |
| Width | : 656 (pixels) |
| Height | : 135 (pixels) |
| Bits per Pixel | : 1 |
| Resolution | : 300 (ppi) |
| Compression | : 2 (code) |
| Compress Length | : 874 (bytes) |
| Scan Alignment | : 16 (bits) |
| Image Data Unit | : 16 (bits) |
| Byte Order | : High-Low |
| MSBit | : First |
| Column Offset | : 0 (pixels) |
| White Pixel | : 0 |
| Data Units | : Unsigned |
| Scan Order | : Row Major,<br>Top to Bottom,<br>Left to Right |
| Parent | : hsf_0/f0000_14/f0000_14.pct |
| X Origin | : 192 (pixels) |
| Y Origin | : 732 (pixels) |

FIGURE 5. The IHead values for the isolated subimage displayed in Figure 6.

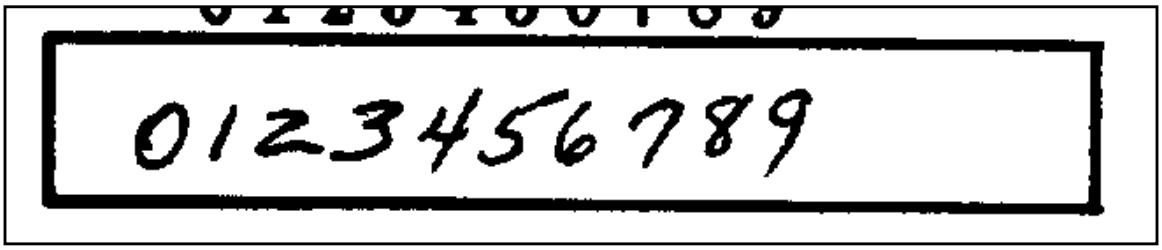



FIGURE 6. An IHead image of an isolated box.

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, 300 dots per inch. The next two fields deal with compression.

In the IHead format, images may be compressed with virtually any algorithm. The IHead is always uncompressed, even if the image data is compressed. 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 5 has a compression code of 2. This signifies that CCITT Group 4 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 load the compressed block of data into main memory. Once the compressed image data has been loaded into memory, CCITT Group 4 decompression can be used to produce an image which has the pixel dimensions consistent with those stored in its header. Using CCITT Group 4 compression and this compression scheme on the images in this database, a compression ratio of 10 to 1 was achieved.

The attribute field, **align**, stores the alignment boundary to which scan lines of pixels are padded. Pixel values of binary images are stored 8 pixels (or bits) to a byte. Most images, however, are not an even multiple of 8 pixels in width. In order to minimize the overhead of ending a previous scan line and beginning the next scan line within the same byte, a number of padded pixels are provided in order to extend the previous scan line to an even byte boundary. Some digitizers extend this padding of pixels out to an even multiple of 8 pixels, other digitizers extend this padding of pixels out to an even multiple of 16 pixels. This field stores the image's pixel alignment value used in padding out the ends of raster scan lines.

The next three attribute fields identify binary interchanging issues among heterogeneous computer architectures and displays. The **unitsize** field specifies how many contiguous pixel values 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, binary incompatibilities across computer hardware and binary 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 binary image described in Figure 5 is black text on a white background and the value of the white pixels is 0. 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 one from which the current image was derived or extracted. In Figure 5, the **parent** field contains the full path name to the image from which the image displayed in Figure 6 was extracted. The **par_x** and **par_y** fields contain the origin point (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. The IHead image format contains the minimal amount of ancillary information required to successfully manage binary and gray scale images.

## 3.0 Data Base Content and Organization

*NIST Special Database 2* contains 5,590 full page images of completed structured forms and correspondingly contains 5,590 ASCII text answer files. The database is approximately 610 Megabytes in size and is distributed on an ISO-9660 formatted CD-ROM disc. The binary images in the database have been 2-dimensionally compressed. Uncompressed the database would require 5.9 Gigabytes of storage.

### 3.1 Hierarchy

Figure 7 illustrates the top level directory tree in the database. The directories **doc**, **man**, and **src**, contain documentation and utilities necessary to manipulate the image data on the CD discussed in Section 4. The **data** directory contains files of images and entry field value answer files described in Section 2.0. The organization of these files is illustrated in Figure 8.

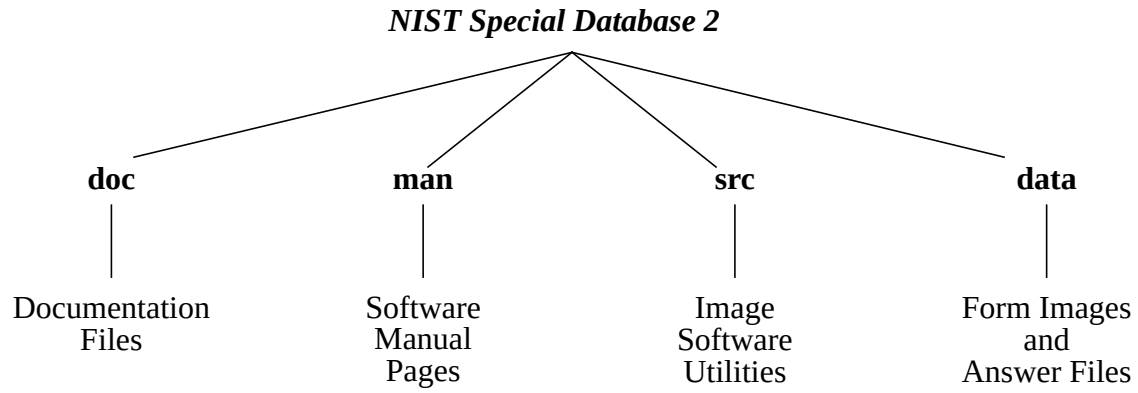



FIGURE 7. The top level directory tree in the database.

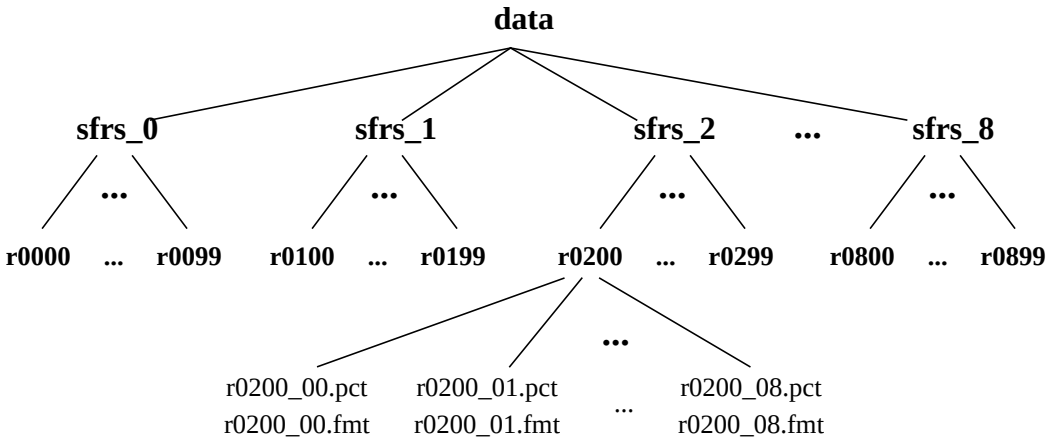



FIGURE 8. The file organization of the form images and answer files contained in *NIST Special Database 2*.

There are 5,590 full page images of completed forms distributed across 8 subdirectories within **data**. The subdirectories **sfrs_0**, **sfrs_1**, through **sfrs_8** each contain 100 synthesized tax submissions comprised of a random collection of completed form faces generated by a computer. There-

fore there are 900 total tax submissions in this database. Each submission is represented as a directory. An example of a submission directory **r0200** is illustrated in Figure 8. In this example, the directory **sfrs_2** contains the 100 submission directories **r0200** through **r0299**. The images associated with submission 200 are stored in the subdirectory **r0200**. There are on average 6.2 form images stored in a submission directory. In Figure 8, **r0200** contains 9 synthesized form faces stored as the files **r0200_00.pct**, **r0200_01.pct**, through **r0200_08.pct** where the last two digits in the file name uniquely index the form images. For each form face image, there is a corresponding answer file. The answer file for the image **r0200_00.pct** is **r0200_00.fmt**, **r0200_01.pct** is **r0200_01.fmt**, and so on. In this way 5,590 form images are stored on the CD with their 5,590 corresponding answer files accounting for 11,180 individual files in all.

### 4.0 Source Code For Data Base Access

In addition to images and answer files, the database contains documentation and software written in the 'C' programming language. Source code for 3 different programs: **dumpihdr**, **ihdr2sun**, and **sunalign** is included within the top level database directory **src**. These programs, their supporting subroutines, and associated file names are described below. These routines are provided as an example to software developers of how IHead images may be manipulated. Manual pages are included in Appendix B and are located in the top level database directory **man**.

### 4.1 Compilation

CD-ROM is a read-only storage medium; this requires the files located in the directory **src** to be copied to a read-writable partition prior to compilation. Once these files have been copied, executable binaries can be produced by invoking the UNIX utility **make**. A command line example follows.

# make -f **makefile.mak**

### 4.2 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 5 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 4 and is found in the file **ihead.h**.

### 4.3 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 **readihdrfile()** which is found in the file **loadihdr.c**.

**Readihdrfile()** is a 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 CCITT Group 4, therefore **readihdrfile()** invokes the external procedure **grp4decomp()** which decompresses the raster data. Upon completion, **readihdrfile()** returns an initialized IHead structure, the uncompressed raster data, and the image's width and height in pixels. **Grp4decomp()** was developed by the CALS Test Network and adapted by NIST for use with this database and is found in the file **g4decomp.c** [1,2].

### 4.4 Sunalign <Sun rasterfile>

**Sunalign** is a program which ensures the Sun rasterfile passed has scanlines of length equal to a even multiple of 16 bits. It has been found that some Sun rasterfile applications assume scanlines which end on an even word boundary. IHead images may contain scanlines which do not conform to this assumption. Therefore, it may be necessary to run **sunalign** on an image which has been converted using **ihdr2sun**. The main routine for this program is found in the file **sunalign.c**.

## 5.0 Entry Field Documentation Tables.

The final set of information provided with this database is a collection of tables. These tables contain general knowledge about each entry field found on a structured form. This knowledge can be applied by system developers to guide the recognition process of their document processing system. These tables specify the data type and context associated with each entry field found on the form faces labeled in Appendix A. Formatted copies of these tables are included in Appendix C and are found in the directory **tables** within the top level database directory **doc**.

Appendix C contains 20 different tables, 1 for each of the 20 different form faces found in this database. Each line in these tables references an unique entry field from the corresponding form face. Entry fields are described by three columns of information. The first column in these tables contains entry field identifiers, the second column contains entry field data types, and the third column contains each entry field's associated context. Figures 9 and 10 list the possible entry field data types and contexts contained on the structured form faces used in this database.

| TAG | DEFINITION |
|---|---|
| A, CA | Alphanumeric Fields |
| F, FF, FP, FPER, FU | Floating Point Fields |
| I | Integer Fields |
| ICON | Non-Character Fields (box markings, signatures) |

FIGURE 9. The set of possible entry field data types.

| TAG | DEFINITION |
|---|---|
| DATA | Generic Data |
| NAME | Names of People |
| SSN | Social Security Numbers |

FIGURE 10. The set of possible entry field contexts.

## References

[1] Department of Defense, “Military Specification - Raster Graphics Representation in Binary Format, Requirements for, MIL-R-28002,” 20 Dec 1988.

[2] CCITT, “Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus, Fascicle VII.3 - Rec. T.6,” 1984.

# Appendix A: Labeled Form Faces

# Appendix B: Manual Pages for Supplied Software

## Appendix C: Example Tables