ANSI/NIST Fingerprint Standard Update Workshop

Dec. 5-6, 2005

Inclusion of JP2 in Type 10 Records

ANSI/NIST-ITL 1-2000

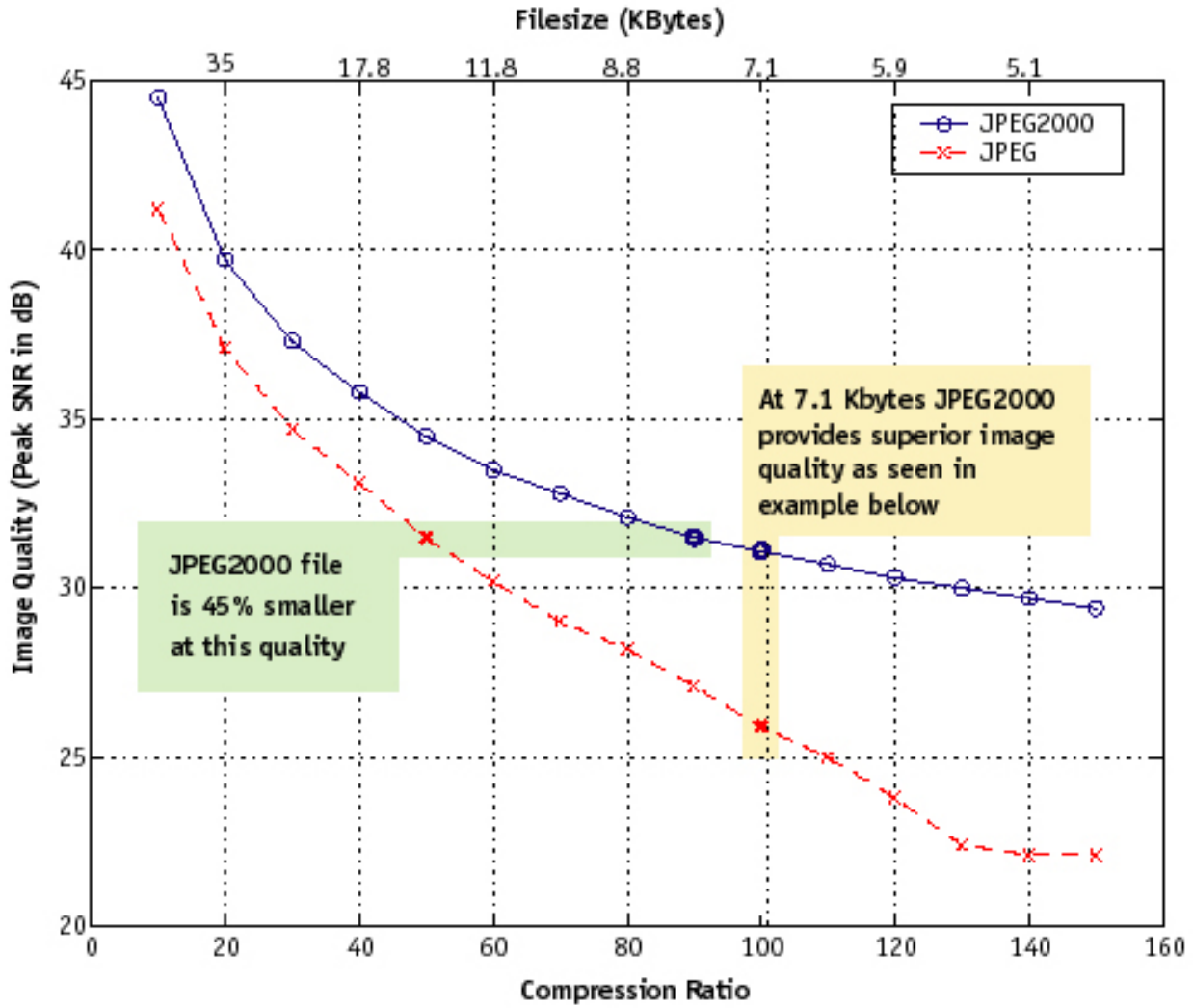
Rob Mungovan, Aware, Inc.

JPEG2000 support in Type 10 Records

- Field 10.11, (CGA)- Compression algorithm
  - Currently supports “NONE”, “JPEGB” (baseline JPEG) and “JPEGL” (lossless mode of JPEG)

- Recommendation:
  - We suggest the inclusion of two new entries
  - “JP2” for lossy JPEG2000 and “JP2L” for for lossless mode of JPEG2000

Better Compression than JPEG



JPEG2000 Scalability: Layered by Resolution and Quality

-----Resolution -→

←------ Quality ----------

Lossless Compression is Part of the Standard

Original: 722 Kbytes

Compressed Identical: 277 Kbytes

Support for Region of Interest

Image compressed at 50:1; all data loss is in the region outside the face.

Proposed Wording

Filters: The 9-7 irreversible filters [11] should be used for lossy mode; however for handheld devices (fixed point processors), the 5-3 reversible filters may be used instead. The 5-3 reversible filters shall be used for lossless mode. A compliant decoder shall be able to decode code streams created through both filters.

Number of resolution levels: The image shall be encoded using enough resolution levels to ensure that a thumbnail with max(width, height) <= 64 is available in the image. Example: a 640x480 image shall be encoded with 5 resolution levels, which enables sub-resolution decodes of 320x240, 160x120, 80x60, and 40x30.

Proposed Wording

Resolution as the dominant progression: JPEG2000 allows five progression orders - LRCP, RLCP, RPCL, PCRL and CPRL. The RLCP progression order (resolution, layer, component, position) shall be used since it best facilitates decode and display of lower resolution derivative images by remote networked devices. Through the RLPC progression order, the code stream shall be formatted so that the resolution information of the image is the first data made available to a decoder in a streaming mode of operation.

Bits per Channel: The number of bits per channel for encoders and decoders shall be 8-16 bits.

Proposed Wording

Single tile images: Facial images shall be encoded using only single tile to avoid tiling artefacts.

JPEG200 quality layers: The image shall be encoded using at least 10 quality layers to enable quality progressive decoding or sub-quality image extraction.

Region of Interest Encoding is allowed: This encoding method is a useful way to compress a facial image to a small size, while retaining sufficient image quality within the specified ROI to perform either human or automated identification.

Add Support for UTF-8

- Committee Members

Bonny Scheier, Sabre

Tony Misslin, Identix

Dale Hapeman, DOD-BFC

Ralph Lessman, CrossMatch

Scott Hills, Aware

Rob Mungovan, Aware

Use of UTF-8 in User Defined Fields

- We suggest this primarily as a way to better support international character sets
- UTF-8 is the 8 bit variant of Unicode
- Unicode is the common standard encoding scheme for international characters and text
- It supports most of the world's characters sets; including…
  - Ethiopic, Cherokee, Canadian Aboriginal Syllabics, Myanmar, Khmer, Mongolian and Braille.

UTF-8 in User Defined Fields

- Current method in ANSI/NIST-ITL 1-2000 to support international character sets is as follows:
  - Described in sections 7.2.3 and 8.1.15
  - Field 1.015 “Directory of Character Sets”- a table of character sets used in the given transaction file
  - Example: index = 000, 7 bit ASCII is used, index = 002, 16 bit unicode is used
  - Within the larger file, the occurrence of the ASCII string 0x02 followed by an equal sign “=“ is used to denote a new character set

UTF-8 in User Defined Fields

- 0x02=002 This means that Unicode 16 bit data follows
- This character set remains active until ASCII 0x03 is encountered or until the next ASCII information separator is encountered
- This non-ASCII text must be base-64 encoded

- Problems with this scheme:
  - Overly complex
  - Not necessary
  - Base-64 encoding expands the data

Recommendation

- Allow UTF-8 in place of 7 bit ASCII for all user-defined fields (not in type 1, type 4, type 8, etc).
- UTF-8 has been a common method for updating 7 bit ASCII systems to support international applications
- ASCII characters map directly to UTF-8
- This change would have no impact on existing systems or domains that do not use UTF-8.
- UTF-8 data can be read/written with existing parsers
- No syntactical changes are are required
- This scheme is in place already in certain locations

Wording Changes, section 6.1

- **page 7, paragraph 2**

The second paragraph restricts text or character data in Type 2 and Type 9 through Type 16 records to 7-bit ASCII code. This should be modified to allow 8-bit UTF-8 characters in all user defined fields within these records.

Proposed wording change to paragraph 2:

After the first sentence:

“The text or character data in Type-2, and Type-9 through Type-16 records will normally be recorded using 7-bit ASCII code in variable length fields with specified upper limits on the size of the fields.”

… Add this…

“Eight bit UTF-8 characters shall be allowed to support international character sets for all user defined fields in all record types. By definition this excludes record types 1, 3, 4, 5, 6 and 8.”

Wording Changes, Section 6.1

- **Section 6.1, page 7, paragraph 3**

The third paragraph should be changed to suggest UTF-8 as the preferred way of storing data that cannot be represented in 7-bit ASCII.

Proposed wording change to paragraph 3:

After the first sentence:

"For data interchange between non-English speaking agencies, character sets other than 7-bit ASCII may be used in textual fields contained in Type-2 and Type-9 through Type-16 records."

… Add this…

"UTF-8 is the preferred method of storing textual data that cannot be represented as 7 bit ASCII."

Changes, Section 7.2.3

- **End of section 7.2.3, page 14**

… add this text…

"Usage of UTF-8 is allowed as an alternative to the technique that requires the usage of the ASCII "STX" and "ETX" characters to signify the beginning or end of of international characters. UTF-8 is only allowed to be used for user defined fields, for example, record type 2, and the UDF fields (200-998) of record types 10, 13, 14, and 15."

… and add this text…

"Even though there is no overlap within the character sets used with UTF-8, UTF-8 should be registered in the type 1 record within the DCS field 1.15 (Directory of Character Sets). "

Changes, Table 4

- **Table 4, page 16**

… add a new character set index …

New Character Set Index: “003”

New Character Set Name: “UTF-8”

New Description: “8-bit Unicode”

Change reserved indices: “004-127”

Changes, Section 8.1.15

- **Section 8.1.15, filed 1.015, page 16**

… after the second sentence add this text…

"In the case of the use of UTF-8, the third optional information item can be used to hold the specific version of the character set used with UTF-8, so that the human terminal can be switched to the right font family."

Example

1.015:003<US>UTF-8<US>Chinese BIG5<GS>"

Add Event Tracking Data Field

- Committee Members

Bonny Scheier, Sabre

Tony Misslin, Identix

Dale Hapeman, DOD-BFC

Ralph Lessman, CrossMatch

Scott Hills, Aware

Rob Mungovan, Aware

Add Event Tracking Data Field

- Purpose: To track and log location (via GPS) and time of events related to a transaction (ANSI/NIST file) during its lifetime.

- Strategy: Keep it simple. Make it flexible. Provide a field and syntax, but make it optional.

- Rationale: In certain architectures many machines may send, receive, or modify the data.
  - This information will be useful to determine time, location and type of data processing.
  - This information will be useful for diagnosing problems and understanding system performance.

Events of Interest

- Events might include such things as
  - Original creation of the transaction
  - Validation (error checking)
  - Receipt and/or forwarding by an intermediate device
  - Changes to the priority (PRY) of the transaction
  - Changes to the addressing of the transaction- including changes to TCN, TCR, ORI, or DAI
  - Insertion/removal of finger, palm, or facial images
  - Changes to textual information content; perhaps related to changes from one domain's format to another (I.e. RCMP to FBI)
  - Final archival of the transaction.

Proposed Changes to the Standard

- Add a new, optional, type 1 field
- Add a new section
  - 8.1.16 Field 1.016 Event Tracking Field (ETF)

- Each event is to be stored in a separate subfield, starting with the initial event.

- Each subfield will consist of 5 data items.

Field 1.016, Five Data Items

- GMT- Greenwich mean time. Identical definition to the current file 1.014 to enable backward compatibility.
- AGN- Identifier of the agency which has recorded the event.
- LAT- Latitude of the location at which the event occurs in decimal degrees. The latitude value should be based on the WGS-84 datum.
- LON- Latitude of the location at which the event occurs in decimal degrees. The latitude value should be based on the WGS-84 datum.
- EVT- Free text description of the event.

Example

- **1.016:19991120235745Z<US>DCFBIWA6Z<US>39.144336<US>-77.216461 <US>Transaction Created <RS>20001022244222Z<US>MASP<US>45.243456<US>23.457895<US>Transaction received and validated<FS>**

Descriptive text for section 8.1.16

“This optional field is intended to provide a consistent, flexible, cross domain format, for tracking events related to the transaction during its lifetime. Each full entry in the ETF field will include the time, agency, location and a text description of an event that affects the transaction.”

“Each event would be stored in a separate subfield starting with the initial event.”

Add description of 5 field items and example as shown on previous slide.