

1  
2  
3  
4  
5  
6 **IREX**

7 An Evaluation-based Program for the Development of Compact  
8 Interoperable ISO/IEC 19794-6 Standardized Iris Images  
9

10  
11  
12 **Iris Exchange (IREX) Evaluation 2008**

13 **Concept, Evaluation Plan and API**

14  
15  
16  
17  
18  
19  
20  
21  
22 **Patrick Grother**

23 **NIST**

24 **September 11, 2008**

25  
26  
27  
28













































## 1 7.2.4. Conversion of raw rectilinear imagery to cropped rectilinear

2 To assess viability of the proposed standard crop-only format, participating submissions to IREX 08 shall convert a raw  
3 raster iris image into a cropped raster iris image, and write the result as a Table 6 instance of Kind = 3. To do this the  
4 implementation will need to find the iris center and crop symmetrically around it. The implementation shall not compress  
5 the image data.

6 June 25, 2008. The implementation shall not rotate the iris: Instead if rotation is detected it shall be recorded in the  
7 appropriate fields of Table 6, i.e. lines 28-29. Template generators and/or matchers should heed such values.

8 The provided SDK shall implement the API call specified in Table 8.

9 **Table 8 – IREX API for preparation of cropped rectilinear records**

Prototype	<pre> INT32 convert_raster_to_cropped_rectilinear( const BYTE *uncompressed_raster_data, const UINT16 image_width, const UINT16 image_height, const BYTE horz_orientation, const BYTE vert_orientation, const BYTE scan_type, const BYTE which_eye, // August 28, 2008: New parameter const BYTE image_format, const BYTE intensity_depth, const UINT16 nist_encoded_device_id, INT16 *bbox_topleft_x, // June 25, 08: These two parameters are new and are introduced to allow NIST INT16 *bbox_topleft_y, // to survey over crop sizes. July 28, 2008: They're are SIGNED (x,y) so can be negative. const UINT32 allocated_bytes, // August 28, 2008: New parameter added for safety. BYTE *c_rectilinear_image); </pre>	
Description	<p>This function takes a raw input image and outputs a corresponding cropped image. The memory for the output structure is allocated before the call i.e. the implementation shall not allocate memory for the result. The function returns either success (0) or failure (non-zero). Failure indicates a failure to convert the image. The result will nevertheless be a conformant instance with zero irides and zero images (0 on Line 22 of Table 6)</p>	
Input Parameters	uncompressed_raster_data	The uncompressed raw image used for template creation.
	image_width	The number of pixels indicating the width of the image.
	image_height	The number of pixels indicating the height of the image.
	horz_orientation	NIST anticipates setting these values to ORIENTATION_BASE, per [STD05, 6.5.4].
	vert_orientation	
	scan_type	Progressive or interlaced. Values per the standard.
	which_eye	EYE_UNDEF = 0 (0x00) EYE_RIGHT = 1 (0x01) EYE_LEFT = 2 (0x02) These are the values used in ISO/IEC 19794-6:2005.
	image_format	NIST anticipates using only unprocessed uncompressed 8 bit grayscale data, so the image format will be 0x0002, and the intensity depth will be 8, both per [STD05].
	intensity_depth	
	nist_encoded_device_id	A two byte unsigned integer value from Table 7
allocated_bytes	Number of bytes NIST allocated for the output image record	
Output Parameters	c_rectilinear_image	The output record, per Table 6 with Kind = 3
	bbox_topleft_x	X coordinate in the original image from which the crop was prepared
	bbox_topleft_y	Y coordinate in the original image from which the crop was prepared
Return Values	0	Success
	2	Elective refusal to generate this Kind of output record
	4	Involuntary failure to make output record - e.g. could not find iris-sclera boundary.
	6	Elective refusal to produce an output record - e.g. on quality grounds.
	8	Cannot parse input data (i.e. assertion that input record is non-conformant)
	Other	Vendor-defined

10

1 The number of times a non-zero error codes is returned will be counted, reported and appropriately factored into  
2 analyses.

### 3 7.2.5. Conversion of raw rectilinear imagery to cropped ROI-masked rectilinear

4 To assess viability of the proposed standard's cropped-and-ROI-masked format, participating submissions to IREX 08 shall  
5 convert a raw raster iris image into a cropped raster iris image, replace eyelids and sclera with fixed pixel values, and write  
6 the result as a Table 6 instance of Kind = 7. To do this the implementation will need to find the iris center and crop  
7 symmetrically around it, and find the eyelids and iris-sclera boundaries. The implementation shall not compress the  
8 output image data.

9 The function shall be implemented with the API call specified in Table 9.

10 **Table 9 – IREX API for preparation of cropped-and-ROI-masked records**

Prototype	<pre> INT32 convert_raster_to_cropped_and_masked_rectilinear( const BYTE *uncompressed_raster_data, const UINT16 image_width, const UINT16 image_height, const BYTE horz_orientation, const BYTE vert_orientation, const BYTE scan_type, const BYTE which_eye, // New August 28, 2008. const BYTE image_format, const BYTE intensity_depth, const UINT16 nist_encoded_device_id, const UINT32 allocated_bytes, // August 28, 2008: New parameter added for safety. BYTE * cm_rectilinear_image); </pre>	
Description	<p>This function takes a raw input image and outputs the corresponding cropped and ROI-masked image. The memory for the output structure is allocated before the call i.e. the implementation shall not allocate memory for the result. The function returns either success (0) or failure (non-zero). Failure indicates a failure to convert the image. The result will nevertheless be a conformant instance with zero irides and zero images (0 on Line 22 of Table 6)</p>	
Input Parameters	uncompressed_raster_data	The uncompressed raw image used for template creation.
	image_width	The number of pixels indicating the width of the image.
	image_height	The number of pixels indicating the height of the image.
	horz_orientation	NIST anticipates setting these values to ORIENTATION_BASE, per [STD05, 6.5.4].
	vert_orientation	
	scan_type	Progressive or interlaced. Values per the standard.
	which_eye	EYE_UNDEF = 0 (0x00) EYE_RIGHT = 1 (0x01) EYE_LEFT = 2 (0x02) These are the values used in ISO/IEC 19794-6:2005.
	image_format	NIST anticipates using only unprocessed uncompressed 8 bit grayscale data, so the image format will be 0x0002, and the intensity depth will be 8, both per [STD05].
	intensity_depth	
	nist_encoded_device_id	A two byte unsigned integer value from Table 7
allocated_bytes	Number of bytes NIST allocated for the output image record	
Output Parameters	cm_rectilinear_image	The output cropped and ROI-masked image, per Table 6 with Kind = 7
Return Values	0	Success
	2	Elective refusal to generate this Kind of output record
	4	Involuntary failure to make output record - e.g. could not find iris-sclera boundary.
	6	Elective refusal to produce an output record - e.g. on quality grounds.
	8	Cannot parse input data (i.e. assertion that input record is non-conformant)
	Other	Vendor-defined

11

12 The number of times a non-zero error code is returned will be counted, reported and appropriately factored into  
13 analyses.

14

## 1 7.2.6. Conversion of raw rectilinear imagery to unsegmented polar

2 To assess viability of the proposed standard's unsegmented polar format, participating submissions to IREX 08 shall  
 3 convert a raw raster iris image into an unsegmented polar image and write the result as a Table 6 instance of Kind = 16.  
 4 To do this the implementation will need to find concentric circles in the pupil and outside the iris, and to execute the  
 5 forward polar transformation (using NIST will provide reference polar transformation code). The implementation shall  
 6 not compress the output image data.

7 The function shall be implemented with the API call specified in Table 10.

8 **Table 10 – IREX API for creation of unsegmented polar records**

Prototype	<pre>INT32 convert_raster_to_unsegmented_polar( const BYTE *uncompressed_raster_data, const UINT16 image_width, const UINT16 image_height, const BYTE horz_orientation, const BYTE vert_orientation, const BYTE scan_type, const BYTE which_eye,      // New August 28, 2008. const BYTE image_format, const BYTE intensity_depth, const UINT16 nist_encoded_device_id, const UNIT16 num_samples_radially,      // Aug 6 2008: If either of these input parameters are 0, the implem- const UINT16 num_samples_circumferentially, // entation should decide suitable values or use published defaults. const UIN32 allocated_bytes,           // August 28, 2008: New parameter added for safety. BYTE * unseg_polar_image);</pre>	
Description	<p>This function takes raw input image and outputs the corresponding unsegmented polar image. The coordinates of the pupil and iris centers are returned also.</p> <p>The memory for the template is allocated before the call i.e. the implementation shall not allocate memory for the result. The function returns either success (0) or failure (non-zero). Failure indicates a failure to convert the image. The result will nevertheless be a conformant instance with zero irides and zero images (0 on Line 22 of Table 6)</p>	
Input Parameters	uncompressed_raster_data	The uncompressed raw image used for template creation.
	image_width	The number of pixels indicating the width of the input image.
	image_height	The number of pixels indicating the height of the input image.
	horz_orientation	NIST anticipates setting these values to ORIENTATION_BASE, per [STD05, 6.5.4].
	vert_orientation	
	scan_type	Progressive or interlaced. Values per the standard.
	which_eye	EYE_UNDEF = 0 (0x00) EYE_RIGHT = 1 (0x01) EYE_LEFT = 2 (0x02) These are the values used in ISO/IEC 19794-6:2005.
	image_format	NIST anticipates using only unprocessed uncompressed 8 bit grayscale data, so the image format will be 0x0002, and the intensity depth will be 8, both per [STD05].
	intensity_depth	
	nist_encoded_device_id	A two byte unsigned integer value from Table 7
	num_samples_radially	The number of sample along a spoke. The output polar data shall have this height.
	num_samples_circumferentially	The number of "spokes" around the iris. The output polar data shall have this width.
allocated_bytes	Number of bytes NIST allocated for the output image record	
Output Parameters	unseg_polar_image	<p>The output record, per Table 6, with the values on lines 33 - 38 set correctly. The Kind on line 3 and the image data on line 55 shall be set as follows.</p> <ul style="list-style-type: none"> <li>— EITHER: Set Kind = 16 and insert proper polar format image data. You can use the NIST code available for the interpolation needed to execute the forward polar transform.</li> <li>— OR : Set Kind = 1, copy the input raster to the output record.</li> </ul>
	Return Value	<p>0 Success</p> <p>2 Elective refusal to generate this Kind of output record</p>



	4	Involuntary failure to make output record - e.g. could not find iris-sclera boundary.
	6	Elective refusal to produce an output record - e.g. on quality grounds.
	8	Cannot parse input data (i.e. assertion that input record is non-conformant)
	Other	Vendor-defined failure.

1

2 The number of times a non-zero error code is returned will be counted, reported and appropriately factored into  
3 analyses.

#### 4 7.2.7. Template creation

5 These functions convert a Table 6 record into an opaque proprietary template. The functions will need to look at the  
6 header of the input record to determine the content, particularly the "Kind" value on Line 3. The functions should return  
7 the defined error code if it does not support certain Kinds. **Sep 7, 2008: Two options are provided - one to convert an  
8 image into a generic enrollment or verification template and another to allow two functions one for enrollment and  
9 another for verification. This "template role" aspect will be respected in Table 12. It supports matching algorithms that  
10 are asymmetric. Your choice of Option 1 or 2 must be indicated in a ".h" header file that you supply.**

11

**Table 11 – IREX API template creation**

Prototype OPTION #1	INT32 convert_image_to_template( const BYTE *input_record, UINT16 *template_size, BYTE *proprietary_template);	
Prototype OPTION #2 Added Sep 7 2008 Two functions here - both functions must be implemented	INT32 convert_image_to_enrollment_template( const BYTE *input_record, UINT16 *template_size, BYTE *proprietary_template);  INT32 convert_image_to_verification_template( const BYTE *input_record, UINT16 *template_size, BYTE *proprietary_template);	
Description	This function takes either a rectilinear image, or an ROI-masked image, or an unsegmented polar image, and outputs a proprietary template. The implementation should inspect the input header to determine which kind of imagery is being provided, per the version number values given in section 6.2. The memory for the output template is allocated before the call i.e. the implementation shall not allocate memory for the result. In all cases, even when unable to extract features, the output shall be a template record that may be passed to the match_templates function without error. That is this routine must internally encode "template creation failed" and the matcher must transparently handle this.	
Input Parameters	input_record	An instance of Table 6. Implementations must alter their behavior according to the Kind of image. The implementation shall support these values: Kind = 1 Kind = 3 Kind = 7 Kind = 48 The SDK does <b>not</b> have to support Kind = 16 because NIST will execute any needed reverse polar transforms using NIST code to make Kind = 48 instances.
Output Parameters	Template_size	The size, in bytes, of the output template
	proprietary_template	The output template. The format is entirely unregulated. <b>June 25,2008: NIST will allocated 8KB before the function is called - if 8KB is not enough email us.</b>
Return Value	0	Success
	2	Elective refusal to process this Kind of input record
	4	Involuntary failure to extract features (e.g. could not find iris in the input-image)
	6	Elective refusal to produce a template (e.g. insufficient iris area)
	8	Cannot parse input data (i.e. assertion that input record is non-conformant)
	Other	Vendor-defined failure

1 The number of times a non-zero error code is returned will be counted, reported and appropriately factored into  
 2 analyses. When the error code is "2" this will be noted in the IREX report.

### 3 7.2.8. Template comparison

4 This function compares two proprietary templates and returns a real-valued distance score.

5 **Table 12 – IREX API template matching**

Prototype	INT32 match_templates( const BYTE *verification_template, const UINT16 verification_template_size, const BYTE *enrollment_template, const UINT16 enrollment_template_size, double *dissimilarity);	
Description	This function compares two opaque proprietary templates and outputs a non-negative match score.  The returned score is a non-negative distance measure. It need not satisfy the metric properties. NIST will allocate memory for this parameter before the call. When either or both of the input templates are the result of a failed template generation (see Table 11), the dissimilarity score shall be -1 and the function return value shall be 2.	
Input Parameters	verification_template	A template from create_template().
	verification_template_size	The size, in bytes, of the input verification template $0 \leq N \leq 2^{16} - 1$
	enrollment_template	A template from create_template().
	enrollment_template_size	The size, in bytes, of the input enrollment template $0 \leq N \leq 2^{16} - 1$
Output Parameters	dissimilarity	A dissimilarity score resulting from comparison of the templates, on the range [0,DBL_MAX]. See section 7.2.9.
Return Value	0	Success
	2	Either or both of the input templates were result of failed feature extraction
	Other	Vendor-defined failure

6

### 7 7.2.9. Dissimilarity score

8 The template comparison function shall return a measure of the dissimilarity between the persons whose iris data is  
 9 contained in the two templates. So, smaller values indicate more likelihood that the two samples are from the same  
 10 person. This deviates from many prior NIST tests which have used "larger-is-more-genuine" semantics.

11 There is no requirement for the scores to be Hamming distances.

12 There is no requirement for values to obey the metric property.

### 13 7.2.10. Implementation identifiers

14 The implementation shall support the self-identification function of Table 13. This function is required to support internal  
 15 NIST book-keeping. The version numbers should be distinct between any versions which offer different algorithmic  
 16 functionality.

17 **Table 13 – IREX API get\_pids function**

Prototype	INT32 get_pid(UINT32 *nist_assigned_identifier, char *email_address);	
Description	This function retrieves an identifier that the provider must request from NIST irex@nist.gov, and hardwire into the source code. NIST will assign the identifier that will uniquely identify the supplier and the SDK version number.	
Output Parameters	nist_assigned_identifier	A PID which identifies the SDK under test. The memory for the identifier is allocated by NIST's calling application, and shall not be allocated by the SDK.
	email_address	Point of contact email address as null terminated ASCII string. NIST will allocate at least 64 bytes for this. SDK shall not allocate.
Return Value	0	Success
	Other	Vendor-defined failure

18

## 1 **7.3. Software and Documentation**

### 2 **7.3.1. SDK Library and Platform Requirements**

3 Participants shall provide NIST with binary code only (i.e. no source code). Header files ( “.h”) are allowed, but these shall  
4 not contain intellectual property of the company nor any material that is otherwise proprietary. It is preferred that the  
5 SDK be submitted in the form of a single static library file (ie. “.LIB” for Windows or “.a” for Linux). However, dynamic and  
6 shared library files are permitted.

7 If dynamic or shared library files are submitted, it is preferred that the API interface specified by this document be  
8 implemented in a single “core” library file with the base filename ‘libIREX’ (for example, ‘libIREX.dll’ for Windows or  
9 ‘libIREX.so’ for Linux). Additional dynamic or shared library files may be submitted that support this “core” library file (i.e.  
10 the “core” library file may have dependencies implemented in these other libraries).

### 11 **7.3.2. Linking**

12 NIST will link the provided library file(s) to a C language test driver application developed by NIST. The runtime  
13 environment shall be either

- 14 – RedHat Linux Enterprise 4 or 5 platforms. (PREFERRED)
- 15 – The cygwin<sup>10</sup> layer running on a Windows Server 2003 OS.

16 Both will use GNU's gcc compiler, version 3.3.3. These use libc. The link command might be:

- 17 – gcc -o irextest irextest.c -L. -lIREX

18 Participants are required to provide their library in a format that is linkable using GCC with the NIST test driver, which is  
19 compiled with GCC. All compilation and testing will be performed on x86 platforms. Thus, participants are strongly  
20 advised to verify library-level compatibility with GCC (on an equivalent platform) prior to submitting their software to  
21 NIST to avoid linkage problems later on (e.g. symbol name and calling convention mismatches, incorrect binary file  
22 formats, etc.).

23 Dependencies on external dynamic/shared libraries such as compiler-specific development environment libraries are  
24 discouraged. If absolutely necessary, external libraries must be provided to NIST upon prior approval by the Test Liaison.

### 25 **7.3.3. Installation and Usage**

26 The SDK must install easily (i.e. one installation step with no participant interaction required) to be tested, and shall be  
27 executable on any number of machines without requiring additional machine-specific license control procedures or  
28 activation.

29 The SDK's usage shall be unlimited. The SDK shall neither implement nor enforce any usage controls or limits based on  
30 licenses, execution date/time, number of executions, presence of temporary files, etc.

31 It is recommended that the SDK be installable using simple file copy methods, and not require the use of a separate  
32 installation program. Contact the Test Liaison for prior approval if an installation program is absolutely necessary.

### 33 **7.3.4. Documentation**

34 Participants shall provide complete documentation of the SDK and detail any additional functionality or behavior beyond  
35 that specified here. The documentation must define all (non-zero) vendor-defined error or warning return codes.

### 36 **7.3.5. Modes of operation**

37 Individual SDKs provided shall not include multiple “modes” of operation, or algorithm variations. No switches or options  
38 will be tolerated within one library. For example, the use of two different “coders” by an iris feature extractor must be  
39 split across two separate SDK libraries, and two separate submissions.

---

<sup>10</sup> According to <http://www.cygwin.com/> is a Linux-like environment for Windows. It consists of two parts: A DLL (cygwin1.dll) which acts as a Linux API emulation layer providing substantial Linux API functionality; a collection of tools which provide Linux look and feel.

### 1 **7.3.6. Watermarking of images**

2 The SDK functions shall not watermark or otherwise steganographically mark up the images.

## 3 **7.4. Runtime behavior**

### 4 **7.4.1. Speed**

5 The following limits are instituted to constrain NIST's total IREX computational workload. The absolute times are probably  
6 less relevant than any relative trends. Deviations above these limits will be allowed but note that timing statistics will be  
7 reported.

- 8 — The mean template match operation should not exceed 20 milliseconds.
- 9 — The mean template creation operation should not exceed 2.5 seconds.
- 10 — The mean iris segmentation operation (e.g. polar) should not exceed 2.5 seconds.

11 The above times assume a vanilla a 2GHz Pentium IV.

### 12 **7.4.2. Interactive behavior**

13 The SDK will be tested in non-interactive “batch” mode (i.e. without terminal support). Thus, the submitted library shall  
14 not use any interactive functions such as graphical user interface (GUI) calls, or any other calls which require terminal  
15 interaction e.g. reads from “standard input”.

### 16 **7.4.3. Error codes and status messages**

17 The SDK will be tested in non-interactive “batch” mod, without terminal support. Thus, the submitted library shall run  
18 quietly, i.e. it should not write messages to "standard error" and shall not write to “standard output”.

### 19 **7.4.4. Exception Handling**

20 The application should include error/exception handling so that in the case of a fatal error, the return code is still  
21 provided to the calling application.

### 22 **7.4.5. External communication**

23 Processes running on NIST hosts shall not side-effect the runtime environment in any manner, except for memory  
24 allocation and release. Implementations shall not write any data to external resource (e.g. server, file, connection, or  
25 other process), nor read from such. If detected, NIST will take appropriate steps, including but not limited to, cessation of  
26 evaluation of all implementations from the supplier, notification to the provider, and documentation of the activity in  
27 published reports.

### 28 **7.4.6. Stateful behavior**

29 All components in this test shall be stateless. This applies to segmentation, feature extraction and matching. Thus, all  
30 functions should give identical output, for a given input, independent of the runtime history. NIST will institute  
31 appropriate tests to detect stateful behavior. If detected, NIST will take appropriate steps, including but not limited to,  
32 cessation of evaluation of all implementations from the supplier, notification to the provider, and documentation of the  
33 activity in published reports.

## 34 **8. References**

AN27	NIST Special Publication 500-271: American National Standard for Information Systems — <i>Data Format for the Interchange of Fingerprint, Facial, &amp; Other Biometric Information – Part 1.</i> (ANSI/NIST ITL 1-2007). Approved April 20, 2007.
BATH06	S. Rakshit and D. M. Monro, Effects of Sampling and Compression on Human Iris Verification, Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 2, No. II, pp. 337-340, Toulouse, May 2006
BATH07	Soumyadip Rakshit and Donald M. Monro, <i>An Evaluation of Image Sampling and Compression for Human Iris Recognition</i> , IEEE Transactions On Information Forensics And Security, Vol. 2, No. 3, September 2007

CAM07	JTC001-SC37-N-2125 — <i>UK Contribution on the effect of severe image compression on iris recognition performance</i> . May 29, 2007. This document was submitted as a UK contribution to SC 37 Working Group 3. It is available as: John Daugman and Cathryn Downing, <i>Effect of severe image compression, on iris recognition performance</i> . Technical Report No. 685, University of Cambridge, Computer Laboratory, UCAM-CL-TR-685, ISSN 1476-2986, May 07. <a href="http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-685.pdf">http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-685.pdf</a> <a href="http://isotc.iso.org/livelink/livelink/6461927/JTC001-SC37-N-2125.pdf?func=doc.Fetch&amp;nodeid=6461927">http://isotc.iso.org/livelink/livelink/6461927/JTC001-SC37-N-2125.pdf?func=doc.Fetch&amp;nodeid=6461927</a>
CAM08	Also very similar material to [CAM07] can be found in: John Daugman and Cathryn Downing, <i>Effect of Severe Image Compression on Iris Recognition Performance</i> , IEEE Trans on Information Forensics and Security, Vol. 3, No. 1, March 2008.
DHS	J. Mayer-Splain for DHS, <i>DHS Technical Contribution in Support of Retaining the Iris Polar Image Format</i> . August 31, 2007. This document is a password protected contribution toward the US position on the revision of 19794-6:2005. <a href="http://www.incits.org/tc_home/m1htm/2007xdocs/m1070456.pdf">http://www.incits.org/tc_home/m1htm/2007xdocs/m1070456.pdf</a>
I379	American National Standard for Information Technology – <i>iris Image Format for Data Interchange</i> , ANSI/INCITS 379-2004, <a href="http://www.incits.org">http://www.incits.org</a>
IRI07	JTC001-SC37-N-2296 US NB Contribution on Compact Iris Format: <a href="#">SC37 Link</a> . Posted 2007-11-03. This document was submitted to M1 for consideration as <a href="#">M1/07-0490</a> . Readers unable to access either link document should <a href="#">contact NIST</a> . <a href="http://isotc.iso.org/livelink/livelink/6904418/JTC001-SC37-N-2296.pdf?func=doc.Fetch&amp;nodeid=6904418">http://isotc.iso.org/livelink/livelink/6904418/JTC001-SC37-N-2296.pdf?func=doc.Fetch&amp;nodeid=6904418</a> <a href="http://www.incits.org/tc_home/m1htm/2007xdocs/m1070490.pdf">http://www.incits.org/tc_home/m1htm/2007xdocs/m1070490.pdf</a>
LG	S. Shah, <i>Comment on Iris Boundary Determination (Polar Representation) and Polar Format</i> , September 7, 2007. This document is a password protected US contribution to the January 2008, SC37/WG3 meeting in Tel Aviv. <a href="http://www.incits.org/tc_home/m1htm/2007xdocs/m1070486.pdf">http://www.incits.org/tc_home/m1htm/2007xdocs/m1070486.pdf</a>
MINEX	P. Grother et al., <i>Performance and Interoperability of the INCITS 378 Template</i> , NIST IR 7296 <a href="http://fingerprint.nist.gov/minex04/minex_report.pdf">http://fingerprint.nist.gov/minex04/minex_report.pdf</a>
MOC	P. Grother and W. Salamon, <i>MINEX II - An Assessment of ISO/IEC 7816 Card-Based Match-on-Card Capabilities</i> <a href="http://fingerprint.nist.gov/minex/minexII/NIST_MOC_ISO_CC_interop_test_plan_1102.pdf">http://fingerprint.nist.gov/minex/minexII/NIST_MOC_ISO_CC_interop_test_plan_1102.pdf</a>
N2059	JTC001-SC37-N-2059 — <i>German National Body contribution on the revision project on Iris Image Data standard ISO/IEC 19794-6</i> . April 20, 2007
N2124	JTC001-SC37-N-2124 — <i>UK Contribution on a defect in ISO/IEC 19794-6 polar iris image format</i> . May 29, 2007
N2226	JTC001-SC37-N-2226 — <i>Base document for revision of ISO/IEC 19794-6, Information technology: Biometric data interchange formats – Part 6: Iris image data</i>
NIST	P. Grother, <i>NIST comments toward US position on N2226, revision of ISO/IEC 19794-6</i> , September 7, 2007 <a href="http://www.incits.org/tc_home/m1htm/2007xdocs/m1070488.pdf">http://www.incits.org/tc_home/m1htm/2007xdocs/m1070488.pdf</a>
PERFSTD	ISO/IEC FDIS 19795-4 — <i>Biometric Performance Testing and Reporting — Part 4: Interoperability Performance Testing</i> . Posted as <a href="#">document 37N2370</a> .
PROC	Hugo Proença and Luís A. Alexandre, <i>Iris Recognition: An Analysis of the Aliasing Problem in the Iris Normalization Stage</i>
STD05	ISO/IEC 19794-6:2005 — <i>Information technology — Biometric data interchange formats — Part 6: Iris image data</i> The standard was published in 2005, and can be purchased from ANSI at <a href="http://webstore.ansi.org/">http://webstore.ansi.org/</a> or ISO.
UNSEG	D. Kim, <i>Introducing the Unsegmented Polar Format</i> , posted as m1070606 and m1070606rev to the M1 document register: <a href="http://m1.incits.org/m1htm/2007docs/m1docreg_2007.htm">http://m1.incits.org/m1htm/2007docs/m1docreg_2007.htm</a> . These documents are password protected US contributions to the January 2008, SC37/WG3 meeting in Tel Aviv. Substantially the same information was presented to the Second NIST Quality Workshop on November 8, 2007 and is linked here: <a href="http://www.itl.nist.gov/iad/894.03/quality/workshop07/presentations.html">http://www.itl.nist.gov/iad/894.03/quality/workshop07/presentations.html</a> as <a href="http://www.itl.nist.gov/iad/894.03/quality/workshop07/proc/Kim_Introducing_Unsegmented_Polar_Data_Format_for_NIST_Presentation_20071106_forPrinting.pdf">http://www.itl.nist.gov/iad/894.03/quality/workshop07/proc/Kim_Introducing_Unsegmented_Polar_Data_Format_for_NIST_Presentation_20071106_forPrinting.pdf</a>
USNA	Robert W. Ives, Bradford L. Bonney, and Delores M. Etter, <i>Effect of Image Compression on Iris Recognition</i> , IMTC 2005 – Instrumentation and Measurement Technology Conference, Ottawa, Canada, 17-19 May, 2005.





## 1 **A.7 Reporting of results**

### 2 **A.7.1 Reports**

3 The Government will combine appropriate results into one or more IREX reports. Together these will contain, at a  
4 minimum, descriptive information concerning IREX, descriptions of each experiment, and aggregate test results. NIST will  
5 include

- 6 – DET performance metrics as the primary indicators of one-to-one verification accuracy,
- 7 – ISO/IEC 19795-4 interoperability matrices as the primary measures of interoperability, and
- 8 – Image generation, template generation, and matching timing statistics.

9 NIST may compute and report other aggregate statistics.

10 MOD Jun 20, 2008: NIST intends to publish results in one or more NIST Interagency Reports. The reports will contain

- 11 – contain the names of participants,
- 12 – contain the results of all participants' implementations with attribution to the participants.

13

### 14 **A.7.2 Pre-publication review**

15 Participants will have an opportunity to review and comment on the reports. Participants' comments will be either  
16 incorporated into the main body of the report (if it is decided NIST reported in error) or published as an addendum.  
17 Comments will be attributed to the participant.

### 18 **A.7.3 Citation of the report**

19 Subsequent to publication of our reports Participants may decide to use the results for their own purposes. Such results  
20 shall be accompanied by the following phrase: "Results shown from the Iris Exchange Test (IREX) do not constitute  
21 endorsement of any particular system by the U. S. Government." Such results shall also be accompanied by the URL of  
22 the IREX Report on the IREX website, <http://iris.nist.gov/irex>.

### 23 **A.7.4 Rights and ownership of the data**

24 Any data generated, deduced, measured or otherwise obtained during IREX (excepting the submitted SDK itself), as well  
25 as any documentation required by the Government from the participants, becomes the property of the Government.  
26 Participants will not possess a proprietary interest in the data and/or submitted documentation.

## 27 **A.8 Return of the supplied materials**

28 NIST will not return any supplied software, documentation, or other material to vendors.

## 29 **A.9 Agreement to participate**

30 With the signing of this form, Participants attest that they will not file any IREX-related claim against IREX Sponsors,  
31 Supporters, staff, contractors, or agency of the U.S. Government, or otherwise seek compensation for any equipment,  
32 materials, supplies, information, travel, labor and/or other participant provided services.

33 The Government is not bound or obligated to follow any recommendations that may be submitted by the Participant. The  
34 United States Government, or any individual agency, is not bound, nor is it obligated, in any way to give any special  
35 consideration to IREX Participants on future contracts, grants or other activities.

36 With the signing of this form, Participants realize that any test details and/or modifications that are provided in the [IREX](#)  
37 [website](#) supersede the information on this form.

38 With the signing of this form, Participants realize that they cannot withdraw from the IREX without their participation and  
39 withdrawal being documented in the IREX Final Report.



