

1 The pixels in the lower eyelid region should be substituted with a fixed mask value. This value shall be recorded on line 31
 2 of Table 6. The lower eyelid mask shall extend to the last (bottom) row of the image unless the iris itself lies in this row.
 3 The lower eyelid mask shall extend to the leftmost and rightmost columns of the image unless the iris itself lies there.
 4 At least one region shall be masked. For regions that are not masked, the corresponding Table 6 mask value shall be 0.
 5 Masks shall be understood as being four-connected regions of a single pixel value.

6 **6.4.2. Assigned values**

7 When masking is used, the non-iris region shall be masked with three values according to the following rules. If the upper
 8 and lower eyelid mask values are respectively U and L, with S the sclera mask value, then:

- 9 – $S \neq U$
- 10 – $S \neq L$
- 11 – $L \leq U$
- 12 – $1 \leq U \leq 255$, but without a specific reason to do otherwise, recommended value is $U = 128$
- 13 – $1 \leq L \leq 255$, but without a specific reason to do otherwise, recommended value is $L = 128$
- 14 – $1 \leq S \leq 255$, but without a specific reason to do otherwise, recommended value is $S = 200$

15 In each case, the special value 0 is reserved to indicate that the region has not been masked. This would support an ISO
 16 standard for raw images, and for images in which one or more regions might reasonably not be masked. The ongoing
 17 revision of ISO/IEC 19794-6 might adopt these rules and conventions.

18 **6.4.3. Mask-ROI transition blurring**

19 Regarding the pixels on the boundary between the mask and the original image, [CAM07] and [CAM08] include this
 20 statement: "the transition to eyelid substitution regions was locally smoothed by a [5x7] kernel to minimize the
 21 boundary's impact on the coding budget".

22 Therefore, the value of each non-masked pixel that is eight-connected to a masked pixel should be substituted by the
 23 result of the kernel neighborhood summation using "an isotropically binomial 7x7 kernel", K, as offered by L1:

$$K = 1/(64*64) \text{ uu}$$

where:

$$u = [1 \ 6 \ 15 \ 20 \ 15 \ 6 \ 1]$$

24 **6.5. Unsegmented polar records**

25 **6.5.1. Outer circle size**

26 The implementation should over-segment the iris. That is, it should assign the radius of the outer circle to be about 15%
 27 larger than that of the iris boundary when it is best approximated by a circle.

28 *NB: NIST will survey over this parameter (by re-segmenting original images). The 15% value is a default.*

29 **6.5.2. Inner circle size**

30 The implementation should under-segment the pupil. That is, it should assign the radius of the inner circle to be about
 31 20% smaller than that of the pupil boundary when it is best approximated by a circle.

32 *NB: NIST will survey over this parameter (by re-segmenting original images). The 20% value is a default.*

33 **6.6. Polar encoding**

34 The ISO/IEC 19794-5:2005 encoding procedure (subclause 6.3.2.7 "Polar conversion") shall be followed. Note that while
 35 the procedure allows for different iris and pupil centers, they shall be co-located here per the concentricity requirement
 36 of [IRI07]. NIST will release code to effect the forward- and reverse-polar interpolations.

1 **6.7. Auxiliary boundary data**

2 **6.7.1. Requirement**

3 IREX does not make provision for purely vendor-defined data (e.g. a proprietary template) in the IREX record because it is
 4 non-interoperable⁹. However, reflecting the fact that segmentation information can be very useful (and, operationally,
 5 could be computed during acquisition) the IREX record allows implementers of all of the formats to supplement the
 6 mandatory fields with either or both of the boundary encodings listed in sections 6.7.2 and 6.7.3.

7 **6.7.2. Ellipse encoded boundaries**

8 Implementers may elect to populate the 24 byte block of best-fit ellipses of the pupil and iris, i.e. Table 6 lines 39 to 50.

9 **6.7.3. FCC encoded boundaries**

10 Implementers may elect to populate either or both of the Freeman chain code blocks on Lines 51-54 of Table 6. This shall
 11 be indicated by setting the respective N values to N > 0 and by including data as specified in Table 5. This first block
 12 applies to the inner iris-pupil boundary. The second block applies to the iris-sclera boundary.

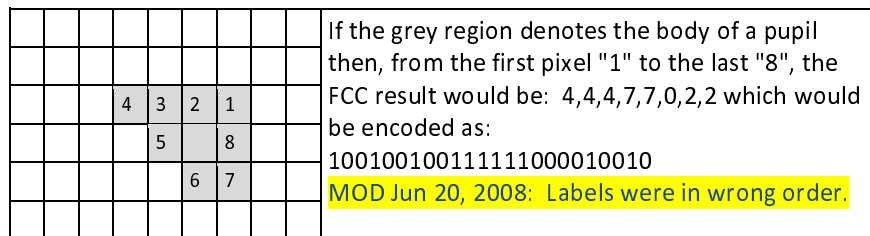
13 Each element of the FCC shall be stored immediately following its predecessor, without regard to byte boundaries. If
 14 necessary, the FCC shall be padded with trailing zeros to ensure that the result is an integral number of octets.

15 Eight-connected Freeman chain codes shall be used. These allow encoding of an arbitrary path in 3 bits per pixel. For an
 16 iris of radius 100 pixels, and a pupil of radius 40 pixels, such encoding would require, with a small header, around 220 and
 17 90 bytes respectively.

18 **Table 5 – Format for Freeman chain code**

#	Field	Length
1	X-coordinate of first pixel in closed path	2B
2	Y-coordinate of first pixel in closed path	2B
3	Number of elements in chain code	N
4	Bit-packed Freeman chain code, zero padded to nearest octet if (3N % 8 != 0)	≤ 3N/8 + 1

19 **Figure 1 – Example Freeman Chain Code**



20
 21
 22 **MOD Jun 20, 2008 added figure:** The standard directions of Figure 2 shall be used.

23 **Figure 2 – Freeman Chain Code Directions**

3	2	1
4	X	0
5	6	7

24

⁹ The issue of allowing Extended Data has been discussed with respect to other biometric standards and has been deprecated because it is only valuable when the producing and consuming implementations are manufactured by the same company.

1 **6.8. Inclusion of non-standard information**

2 The records shall contain only the information defined in this document. This restriction prohibits steganographic markup
3 of images. Interoperability is not assisted by proprietary hidden information.

Table 6 – Structure for the proposed formats for ISO/IEC 19794-6

	Section title and/or field name	L	IREX actual or required values	Remarks	
Iris Record Header (adapted from Table 2 of 37N2484 posted February 2008 as the latest revision text of ISO/IEC 19794-6)					
1.	Format Identifier	4 B	0x49495200	i.e. ASCII "IIR\0"	
2.	Version Number	4 B	0x30313500	'015\0' a non-standard interim value for IREX 08	
3.	Kind of imagery in this record	8b	0000 0001b = 1	Rectilinear without ROI masking without any cropping	
			0000 0011b = 3	Rectilinear without ROI masking but cropped so iris is about centered	
			0000 0111b = 7	[CAM07, CAM08] Rectilinear with masking and cropping	
			0001 0000b = 16	[IRI07] Unsegmented polar	
			0011 0000b = 48	Rectilinear reconstruction of [IRI07] unsegmented polar image	
4.	Record Length	4 B	$46 \leq L \leq 2^{32} - 1$	Total length	
5.	Capture Device ID	2 B		Value will be provided to SDK, which stores it here. Value will be one of those in Table 7.	
6.	Number of eyes imaged	1 B	1	In IREX this will always be 1	
7.	Record header length	2 B	46	In ISO drafts this was 45, but here line 3 adds 1 to make 46	
8.	Iris image properties (NB order)	Horz. orientation	2 b	ORIENTATION_BASE	Bits 0 and 1 i.e. 2 least significant bits (per see 6.4 in N2484)
9.		Vert. orientation	2 b	ORIENTATION_BASE	Bits 2 and 3
10.		Scan type	2 b	SCAN_TYPE_PROGRESSIVE SCAN_TYPE_INTERLACE_FRAME	Bits 4 and 5. One of these two values will be supplied to the IREX SDKs.
11.		Occlusions	1 b		Polar format only, see ISO/IEC 19794-6:2005
12.		Occlusion filling	1 b		Polar format only, see ISO/IEC 19794-6:2005
13.		Reserved for future use	8b	0	Most significant bits are reserved in ISO/IEC 19794-6
14.	Iris diameter (rectilinear)	2 B		Expected iris diameter in pixels. Optional: if not estimated value shall be 0.	
15.	Image format	2 B	IMAGEFORMAT_MONO_RAW	IREX SDKs are not permitted to compress, and not required to decompress. Color data will not be used. IMAGEFORMAT_MONO_RAW is the only allowed value.	
16.	Raw image width	2 B	> 0	For rectilinear, these values are the dimensions of the image data on the last line in this Table. For polar images these are the dimensions of the parent image from which the polar representation was extracted.	
17.	Raw image height	2 B	> 0		
18.	Intensity depth	1 B	8 bits		
19.	Image transformation (polar only)	1 B	1	TRANS_STD = 1 i.e. linear radial interpolation per [STD05, clause 6.5.4].	
20.	Device unique identifier	16 B	'0000000000000000'	16 character zeroes '0' = 0x30, not 0x00.	
Iris Biometric Subtype Header (Table 3 of 37N2484 posted February 2008 as the latest revision text of ISO/IEC 19794-6)					
21.	Eye	1 B	EYE_RIGHT, EYE_LEFT	NIST expects L/R will be known and supplied to SDK	
22.	Number of iris images of this eye	2 B	1	NIST will not present multiple images, so this is 1 here	
Iris Image Header (Table 4 of 37N2484 posted February 2008 as the latest revision text of ISO/IEC 19794-6)					
23.	Image length	4B	M > 0	Length of image data. For uncompressed 8bit greyscale data this is W x H x 1.	
24.	Image number	2 B	1	This field is an index starting at 1	
25.	Quality	1 B	$0 \leq Q \leq 100, 255$	This value shall be computed and stored here by the SDK	
26.	Quality algorithm vendor ID	2B		This value is optional in IREX. IBIA code may be used - NIST would build register.	
27.	Quality algorithm ID	2B		The value is optional in IREX	

IREX 2008

28.	Rotation angle of eye	2 B	0 ... 0xFFFF 0xFFFF =ROT_ANGLE_UNDEF	= (unsigned short) round(65536*angle/360) modulo 65536, where angle is measured counter-clockwise in degrees
29.	Rotation uncertainty	2 B	0 ... 0xFFFF 0xFFFF =ROT_UNCERTAIN_UNDEF	= (unsigned short) round(65536*uncertainty/180) where $0 \leq \text{uncertainty} < 180$ is measured in degrees and is the absolute value of maximum error
Fields supporting ROI-masked images				
30.	Mask value for upper eyelid	1B		These three fields are populated if Kind = 7. Otherwise all set to zero. See definitions and rules in section 6.4. Beyond IREX 08 the depth may need to be ≥ 8 bits.
31.	Mask value for lower eyelid	1B		
32.	Mask value for sclera	1B		
Fields supporting UNSEG polar images				
33.	Number of samples radially	2B	$0 < \text{NR}$	These six fields shall be populated if Kind = 16 or Kind = 48 (see line 3). Otherwise all set to 0. NR is the number of samples along a spoke NC is the number of spokes around the iris The polar image data on the last line of this table has height NR, width NC. Coordinate system is zero oriented with (0,0) at the top left corner. The inner and outer circle centers are concentric.
34.	Number of samples circumferentially	2B	$0 < \text{NC}$	
35.	X coordinate of inner + outer circle centers	2 B	$0 \leq x < W$	
36.	Y coordinate of inner + outer circle centers	2 B	$0 \leq y < H$	
37.	Inner circle radius	2 B	$0 < r$	
38.	Outer circle radius	2 B	$0 < r$	
Fields supporting all images				
39.	X coord of the center of the ellipse approximating the pupil boundary	2 B		Population of these six fields is allowed but not required for all "Kinds". If an implementation elects not to compute the ellipse then it shall assign X = 0xFFFF on line 39. Consumers of this data shall ignore these fields if X = 0xFFFF i.e. value on line 39 is out of bounds with respect to the width on line 16.
40.	Y coord of the center of the ellipse approximating the pupil boundary	2 B		
41.	X coord of the intersection pt. of the semi-major axis with the ellipse approximating the pupil	2 B		
42.	Y coord of the intersection pt. of the semi-major axis with the ellipse approximating the pupil	2 B		
43.	X coord of the intersection pt. of the semi-minor axis with the ellipse approximating the pupil	2 B		
44.	Y coord of the intersection pt. of the semi-minor axis with the ellipse approximating the pupil	2 B		
45.	X coord of the center of the ellipse approximating the iris boundary	2 B		Population of these six fields is allowed but not required for all "Kinds". If an implementation elects not to compute the ellipse then it shall assign X = 0xFFFF on line 45. Consumers of this data shall ignore these fields if X = 0xFFFF i.e. value on line 45 is out of bounds with respect to the width on line 16.
46.	Y coord of the center of the ellipse approximating the iris boundary	2 B		
47.	X coord of the intersection pt. of the semi-major axis with the ellipse approximating the iris	2 B		
48.	Y coord of the intersection pt. of the semi-major axis with the ellipse approximating the iris	2 B		
49.	X coord of the intersection pt. of the semi-minor axis with the ellipse approximating the iris	2 B		
50.	Y coord of the intersection pt. of the semi-minor axis with the ellipse approximating the iris	2 B		

Continued next page ...

IREX 2008

51.	Freeman code length for pupil-iris boundary	2 B	N	See section 6.7. This block allowed but not required for all "Kinds".
52.	Freeman code data for pupil-iris boundary	N		If N = 0, the sclera-iris FCC length value follows immediately
53.	Freeman code length for sclera-iris boundary	2 B	N	See section 6.7. This block allowed but not required for all "Kinds".
54.	Freeman code data for sclera-iris boundary	N		If N = 0, the image data follows immediately
55.	Image data	M		Length M encoded on line 23, in bytes

1

1 7. PC-based API specification

2 7.1. Overview

3 This section describes the IREX API. All SDK's submitted to IREX 08 shall implement the functions below here as required
4 by the classes of participation listed in Table 4.

5 7.2. Testing interface

6 7.2.1. Requirement

7 IREX participants shall submit an SDK which presents the "C" prototyped interface given in the following subsections.

8 7.2.2. Sensor identifiers

9 IREX will use images from:

- 10 – a large corpus collected using the LG 3000.
- 11 – a larger corpus collected using the Securimetrics PIER camera.
- 12 – the smaller sequestered ICE 06 corpus of LG 2200 images

13 **NIST is actively seeking to extend this to include other sources - please see NIST's call for images on Page 2.** To support
14 interoperable i.e. cross-sensor matching, the SDK will be told the sensor the two byte unsigned integer values in Table 7.

15 **Table 7 – Sensor identifiers**

#	Sensor Manufacturer and Model	Identifier
1	LG 2200	0x2A16
2	LG 3000	0x2A1E
3	LG 4000	0x2A26
4	Securimetrics PIER	0x1A03
5	Unknown or unspecified	0x0000

16 Presence on this table indicates NIST's intention to use images captured by these devices. NIST will revise this table as
17 other data becomes available.

18 7.2.3. Geometric, photometric or other alterations to images

19 It is at the discretion of the provider whether to alter the input images during the Table 6 record preparation. If a vendor
20 believed for instance that contrast enhancement would produce a more easy-to-recognize image then the
21 implementation is permitted to do this. A more important example is described in Section 2 of [CAM07,CAM08].

For those images in which the iris was partly outside of the original image frame, the missing pixels were replaced with black ones. For those in which the algorithms detected that the gaze was directed away from the camera, as gauged by projective deformation of the eye shape, a corrective affine transformation was automatically applied which effectively "rotated" the eye in its socket back into orthographic perspective on-axis with the camera.

22 Such steps are allowed and are likely to allow downstream feature extractors and matchers to give better performance.
23 NIST takes no position on whether these or other operations should be applied. NIST does however prohibit the
24 application of compression and recommends against any action which would blur the image. Note that vendors might
25 profitably implement local image processing steps (i.e. not requiring the entire original image) in the front-end of the
26 feature extraction routines.

27

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

1 7.2.5. Conversion of raw rectilinear imagery to cropped ROI-masked rectilinear

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

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

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

Prototype	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 image_format, const BYTE intensity_depth, const UINT16 nist_encoded_device_id, BYTE * cm_rectilinear_image);	
Description	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.	
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.
	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	
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

9

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

12

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	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 image_format, const BYTE intensity_depth, const UINT16 nist_encoded_device_id, const UNIT16 num_samples_radially, const UINT16 num_samples_circumferentially, BYTE * unseg_polar_image);	
Description	This function takes raw input image and outputs the corresponding unsegmented polar image. The coordinates of the pupil and iris centers are returned also. 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.	
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.
	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.	
Output Parameters	unseg_polar_image	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. — 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. — OR: Set Kind = 1, copy the input raster to the output record.
Return Value	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 failure.

9

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

3 7.2.7. Template creation

4 This function converts a Table 6 record into an opaque proprietary template. The function will need to look at the header
5 of the input record to determine the content, particularly the "Kind" value on Line 3. The function should return the
6 defined error code if it does not support certain Kinds.

7 **Table 11 – IREX API template creation**

Prototype	INT32 convert_image_to_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 not 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. June 25,2008: NIST will allocated 8KB before the function is called - if 8KB is not enough email us.
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

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

10 7.2.8. Template comparison

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

12 **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

1

2 7.2.9. Dissimilarity score

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

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

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

8 7.2.10. Implementation identifiers

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

12

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

13

14 7.3. Software and Documentation

15 7.3.1. SDK Library and Platform Requirements

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

20 If dynamic or shared library files are submitted, it is preferred that the API interface specified by this document be
21 implemented in a single "core" library file with the base filename 'libIREX' (for example, 'libIREX.dll' for Windows or
22 'libIREX.so' for Linux). Additional dynamic or shared library files may be submitted that support this "core" library file (i.e.
23 the "core" library file may have dependencies implemented in these other libraries).

1 **7.3.2. Linking**

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

- 4 — RedHat Linux Enterprise 4 or 5 platforms. (PREFERRED)
- 5 — The cygwin¹⁰ layer running on a Windows Server 2003 OS.

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

- 7 — gcc -o irextest irextest.c -L. -lIREX

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

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

15 **7.3.3. Installation and Usage**

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

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

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

23 **7.3.4. Documentation**

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

26 **7.3.5. Modes of operation**

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

30 **7.3.6. Watermarking of images**

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

32 **7.4. Runtime behavior**

33 **7.4.1. Speed**

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

- 37 — The mean template match operation should not exceed 20 milliseconds.
- 38 — The mean template creation operation should not exceed 2.5 seconds.

¹⁰ 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 — The mean iris segmentation operation (e.g. polar) should not exceed 2.5 seconds.

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

3 **7.4.2. Interactive behavior**

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

7 **7.4.3. Error codes and status messages**

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

10 **7.4.4. Exception Handling**

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

13 **7.4.5. External communication**

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

19 **7.4.6. Stateful behavior**

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

25 **8. References**

AN27	NIST Special Publication 500-271: American National Standard for Information Systems — <i>Data Format for the Interchange of Fingerprint, Facial, & 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. http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-685.pdf http://isotc.iso.org/livelink/livelink/6461927/JTC001-SC37-N-2125.pdf?func=doc.Fetch&nodeid=6461927
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. http://www.incits.org/tc_home/m1htm/2007xdocs/m1070456.pdf

IREX 2008

I379	American National Standard for Information Technology – <i>iris Image Format for Data Interchange</i> , ANSI/INCITS 379-2004, http://www.incits.org
IRI07	JTC001-SC37-N-2296 US NB Contribution on Compact Iris Format: SC37 Link . Posted 2007-11-03. This document was submitted to M1 for consideration as M1/07-0490 . Readers unable to access either link document should contact NIST . http://isotc.iso.org/livelink/livelink/6904418/JTC001-SC37-N-2296.pdf?func=doc.Fetch&nodeid=6904418 http://www.incits.org/tc_home/m1htm/2007xdocs/m1070490.pdf
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. http://www.incits.org/tc_home/m1htm/2007xdocs/m1070486.pdf
MINEX	P. Grother et al., <i>Performance and Interoperability of the INCITS 378 Template</i> , NIST IR 7296 http://fingerprint.nist.gov/minex04/minex_report.pdf
MOC	P. Grother and W. Salamon, <i>MINEX II - An Assessment of ISO/IEC 7816 Card-Based Match-on-Card Capabilities</i> http://fingerprint.nist.gov/minex/minexII/NIST_MOC_ISO_CC_interop_test_plan_1102.pdf
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 http://www.incits.org/tc_home/m1htm/2007xdocs/m1070488.pdf
PERFSTD	ISO/IEC FDIS 19795-4 — <i>Biometric Performance Testing and Reporting — Part 4: Interoperability Performance Testing</i> . Posted as document 37N2370 .
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 http://webstore.ansi.org/ or ISO.
UNSEG	D. Kim, <i>Introducing the Unsegmented Polar Format</i> , posted as m1070606 and m1070606rev to the M1 document register: http://m1.incits.org/m1htm/2007docs/m1docreg_2007.htm . 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: http://www.itl.nist.gov/iad/894.03/quality/workshop07/presentations.html as http://www.itl.nist.gov/iad/894.03/quality/workshop07/proc/Kim_Introducing_Unsegmented_Polar_Data_Format_for_NIST_Presentation_20071106_forPrinting.pdf
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.

Annex A

Application to participate in IREX

1
2

A.1 Who should participate

4 Providers of iris recognition technologies are invited to participate in IREX. In addition, companies, research
5 organizations, or universities that have developed mature prototypes or who research iris segmentation or matching are
6 invited to participate.

7 The algorithms and software need not be “operational,” nor a production system, nor commercially available. However,
8 the system must, at a minimum, be a stable implementation capable of being “wrapped” (formatted) in the API
9 specification that NIST has specified in section 7 for this evaluation.

10 Anonymous participation will not be permitted. This means that signatories to this Agreement acknowledge that they
11 understand that the results (see sections 5.10 and Annex A.7) of the evaluation of the software and/or hardware will be
12 published with attribution to their organization(s).

A.2 How to participate

14 Those wishing to participate in IREX testing must do all of the following, on the schedule listed on Page 4.

- 15 — MOD Jun 20, 2008: Indicate via email a non-binding "Intention to Participate" - see the schedule on Page 4.
- 16 — Request an SDK ID from NIST (for use per section 7.2.10).
- 17 — Follow the instructions for cryptographic protection of your SDK here.
18 http://iris.nist.gov/irex/NIST_biometrics_crypto.pdf
- 19 — Send a signed and fully completed copy of this entire Annex A, including the *IREX Application to Participate* form
20 below. This must identify, and include signatures from, the Responsible Parties as defined in section A.4
- 21 — Provide an SDK (Software Development Kit) library which complies with the API (Application Programmer Interface)
22 specified in this document.

23 The *IREX Application to Participate* shall be sent to:

IREX Test Liaison National Institute of Standards and Technology Information Access Division (894) 100 Bureau Drive A203/Tech225/Stop 8940 Gaithersburg, MD 20899-8940 USA	In cases where a courier needs a phone number please use NIST shipping and handling on: 301 -- 975 -- 6296.
--	---

24

A.3 NIST activity

A.3.1 Initiation

27 Upon receipt of the signed Annex A form by NIST, the organization shall be classified as a “Participant”. NIST must receive
28 the form during the submission period described in schedule on Page 2 of this document.

A.3.2 Supplier validation

30 Registered Participants will be provided with a small Validation Dataset available on the website <http://iris.nist.gov/irex>.
31 Prior to submission of their SDK, the Participant must to verify that their software executes on the validation data, and
32 produces correct similarity scores and templates.

1 **A.3.3 Submission of software to NIST**

2 NIST requires that all software submitted by the participants be signed and encrypted. Signing is done with the
3 participant's private key, and encrypting is done with the NIST public key, which is published on the IREX Web site. NIST
4 will validate all submitted materials using the participant's public key, and the authenticity of that key will be verified
5 using the key fingerprint. This fingerprint must be submitted to NIST by writing it on the signed participation agreement.

6 By encrypting the submissions, we ensure privacy; by signing the submission, we ensure authenticity (the software
7 actually belongs to the submitter). **NIST will not accept into IREX any submission that is not signed and encrypted. NIST
8 accepts no responsibility for anything that is transmitted to NIST that is not signed and encrypted.**

9 The detailed commands for signing and encrypting are given here: http://iris.nist.gov/irex/crypto_protection.pdf

10 **A.3.4 Acceptance testing**

11 Software submitted shall implement the IREX API Specification of section 7.

12 Upon receipt of the SDK and validation output, NIST will attempt to reproduce the same output by executing the SDK on
13 the validation imagery, using a NIST computer. In the event of disagreement in the output, or other difficulties, the
14 Participant will be notified.

15 **A.3.5 Limits of testing**

16 NIST will use the Participant's SDK software only for purposes related to the testing described in this document. The
17 provided software will also be used to resolve any errors identified subsequent to the test or publication of results. NIST
18 agrees not to use the Participants software for purposes other than indicated herein, without express permission by the
19 Participant. NIST reserves the right to conduct analyses of the output data and measurements beyond those described in
20 this document. NIST reserves the right to apply the software to images from sensors not enumerated in this document.

21 **A.4 Parties**

22 **A.4.1 Responsible Party**

23 The Responsible Party is an individual with the authority to commit the organization to the terms in this document.

24 **A.4.2 Point of contact**

25 The Point of Contact is an individual with detailed knowledge of the system applying for participation.

26 The IREX Liaison is the government point of contact for IREX. All correspondence should be directed to irex@nist.gov,
27 which will be received by the IREX Liaison and other IREX personnel.

28 These correspondences may be posted on the FAQ (Frequently Asked Questions) area of the <http://iris.nist.gov/irex> at the
29 discretion of the IREX Liaison. The identity of those persons or organizations whose correspondences lead to FAQ
30 postings will not be made public in the FAQ.

31 **A.5 Access to IREX validation data**

32 The IREX Validation Data is supplied to Participants to assist in preparing for IREX.

33 The images in the IREX Validation Data are representative of the IREX Test Data only in their format. Image quality,
34 collection device and other characteristics are likely to vary between the Validation and Test Datasets.

35 **A.6 Access to IREX test data**

36 The IREX Test Datasets are in some cases protected under the Privacy Act (5 U.S.C. 552a), and will be treated as Sensitive
37 but Unclassified and/or Law Enforcement Sensitive.

38 IREX Participants shall have no access to IREX Test Data, either before, during or after the test.

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.

IREX 2008

1

This form shall be completed by all suppliers electing to participate in the IREX evaluation.				
NIST assigned identifier for the supplied SDK.				
Responsible Party for supplier of iris segmentation, encoding and/or matching technologies.				
Company / Organization Name				
Title	First Name	MI	Last Name	Suffix
Street Address				
City		State	Zip	Country
Phone		Fax	Email	
Technical point of contact		Phone	Email	
Participant's public-key fingerprint (enter here)				
NIST's public-key fingerprint		846E 7008 996A E912 974C F8D7 1C7A 0F22 856B 9B28		

2 With my signature, I agree that this document is a sufficient description of the test to be conducted.

3 With my signature, I hereby request consideration as a Participant in the Iris Interoperability Exchange Test II (IREX), and I
 4 am authorizing my company or organization to participate in IREX according to the rules and limitations listed in this
 5 document.

6 With my signature, I also state that I have the authority to accept the terms stated in this document

7

8

9

10 SIGNATURE OF SOFTWARE SUPPLIER RESPONSIBLE PARTY

DATE

11