

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

IREX II – IQCE

Iris Quality Calibration and Evaluation 2010
Concept, Evaluation Plan and API
Version 4.1

Elham Tabassi, Patrick Grother, Wayne Salamon

Image Group
Information Access Division
Information Technology Laboratory
National Institute of Standards and Technology



January 12, 2010

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Status of this Document

This is the final evaluation plan posted on December 24, 2009. NIST intends that the content of this document is fixed. However, NIST will update the document in response to specific technical issues. NIST may add background information. Comments and questions should be submitted to irex@nist.gov. A FAQ will be maintained in Annex B (to be developed).

UPDATEs

January 12, 2010:

Section 15.2 Linking has been corrected:

- RedHat Linux Enterprise 5 platforms based on later linux 2.6 kernels, with gcc 4.1.2-14 (PREFERRED), or
- The cygwin1 layer running on a Windows Server 2003 OS (might be upgraded to 2008) with linking done with gcc 3.4.4.

The development effort described in this document will benefit from larger iris image datasets. NIST has successfully engaged various organizations in accessing more images. Nevertheless we request organizations who can share their iris images with NIST to please contact elham.tabassi@nist.gov 301 975 5292.

We welcome any sort of iris images, but are particularly interested in images with impairments, and iris images where the capture setting parameters such as illumination, MTF, camera noise, etc are known.

- 1
- 2
- 3
- 4

1
2
3
4
5
6
7
8
9
10
11
12

Document History

0	First Draft, circulated for public comment	October 4, 2009
1	2 nd Draft, circulated for public comment	October 22, 2009
2	3 rd Draft, circulated for public review	November 27, 2009
3	4 th Final Draft posted	December 24, 2009

1

Intended Timeline

Aug -Sept, 2010	NIST Interagency Report
July, 2010	SC 37 Working Group 3 meeting in Malaysia
June 30, 2010	Preliminary report for SC 37 Working Group 3 meeting
June xx, 2010	M1 (US TAG to SC 37) meeting
May 31, 2010	Submission period ends
February 4, 2010	Deadline for submission of signed Annex A Participation Application to NIST (i.e. intend to participate – we need this for planning purposes)
January 26, 2010	Submission period begins. Iterations of submit -> evaluate -> report
January 18-22, 2010	SC 37 Working Group 3 meeting in Singapore
December 24, 2009	Release of final evaluation plan
November 27, 2009	3 rd draft posted. Editorial or technical fixes are welcome.
November 15, 2009	2 nd (and final) comment period ends.
October 22, 2009	Release of second draft evaluation plan, for comment.
October 18, 2009	Comment on 1 st draft due
October 4, 2009	Launching IQCE. Announcing website, ask for comment on IQCE test plan and API.

2

3 Nov 27, 2009: Timeline has been revised. To support research and development, IQCE will perform iterations of
4 submit -> evaluate -> report. Submission period begins in January 2010 and will end in May 2010. See clause 12.

5

6

Sponsors

7 NIST is grateful to the Department of Homeland Security's Science and Technology Directorate for supporting this
8 work. NIST thanks all organizations and institutions that contributed to NIST iris image corpora.

9

1	Table of Contents	
2	1. Overview	8
3	2. Scope	9
4	3. Normative references	10
5	4. Abbreviations	10
6	5. Motivation and background	10
7	6. Relation to other NIST tests	10
8	7. Relation to Standard	11
9	8. Audience and options for participation	11
10	9. Aspects of the test	13
11	9.1. Data	13
12	9.2. IQAA output	13
13	9.2.1. Scalar overall quality	15
14	9.2.2. Vector quality	15
15	9.3. Vector quality summarization	16
16	9.4. Measure performance	16
17	9.5. Measure efficiency	16
18	9.6. Measure robustness	16
19	9.7. Calibration curve	16
20	10. Number of submissions	17
21	11. Provision of sensor information to IQAAs	17
22	12. Phased testing	17
23	13. IQAA execution time	18
24	14. PC-based API Specification	18
25	14.1. Overview	18
26	14.2. Testing interface	18
27	14.2.1. Requirement	18
28	14.2.2. Sensor identifiers	18
29	14.2.3. Geometric, photometric or other alterations to images	18
30	14.2.4. Proprietary template creation	19
31	14.2.5. Proprietary templates comparison	20
32	14.2.6. Dissimilarity	21
33	14.2.7. Quality computation	21
34	14.2.8. Quality metric description	22
35	14.2.9. Implementation identifiers	22
36	15. Software and Documentation	22
37	15.1. SDK Library and Platform Requirements	22
38	15.2. Linking	22
39	15.3. Installation and Usage	22
40	15.4. Documentation	24
41	15.5. Modes of operation	24
42	16. Runtime behavior	24
43	16.1. Speed	24
44	16.2. Interactive behavior	24
45	16.3. Error codes and status messages	24
46	16.4. Exception Handling	24
47	16.5. External communication	24

1	16.6. Stateful behavior	24
2	17. References	25
3		
4		

Iris Quality Calibration and Evaluation (IQCE) Concept, Evaluation Plan, and API

1. Overview

Recent studies such as IREX I¹ showed that iris images captured at near infrared are viable biometrics for verification and identification. Like other biometrics, the performance of iris recognition algorithms drops when comparing images from imperfect sources (e.g. subject blinking) or under imperfect conditions (e.g. out of focus) [DOR_ICIP05, DAU_SMC07, KAL_WVU05, ZHO09]. Iris recognition technology is rapidly gaining acceptance and support in government identity management applications. However, the issue of iris image quality is not quantitatively defined and merits considerable research and evaluation. This project aims at closing this shortfall by enabling scientific progress in iris image quality definition and assessment. Specifically, IQCE seeks to

- identify iris image properties that are influential on recognition accuracy, and to quantify their effects;
- collect and document iris image quality metrics in any of the following forms:
 - mathematical equations (e.g. in an academic publication)
 - software implementation (e.g. open source code, or proprietary compiled libraries);
- evaluate iris image quality assessment algorithms to assess the state-of-the-art; and
- calibrate iris image quality assessment algorithms to expand marketplace of interoperable products.

The IQCE activity supports a new, formal, standard addressing iris quality. That standard, ISO/IEC 29794-6 *Iris Image Quality*, was initiated by the Working Group 3 of the ISO SC 37 committee in July 2009. The standard will define a vector of quality components each of which is some quantitative measure of a subject-specific or image-specific covariate. The current working draft (SC 37 N 3331) defines 19 image acquisition or subject covariates and 12 metrics for assessing the utility of an iris image. The 12 quality metrics are mostly quantification of the covariates. The table below lists covariates and quality metrics defined in SC 37 N 3331.

Table 1. Iris image or subject covariates and quality metrics defined in SC 37 N 3331

Iris acquisition covariates	Iris subject covariates	Quality metric
Dedicated illumination	Deviation from circularity in iris-sclera border and iris- pupil border	Contrast (iris-sclera and iris-pupil boundary contrast)
Defocus	Eye color	Gray scale density
Dynamic range	Eye wear	Image scale (margin between the iris boundary and closest edge of the image)
Motion Blur	Intrinsic iris-pupil contrast	Image orientation – head rotation
Noise (or camera sensitivity)	Intrinsic iris-sclera contrast	Image orientation – sight direction
Occlusion due to specular reflections	Occlusion due to eyelash/eyelid	Iris boundary shape
Optical distortion	Off-axis Orientation – head rotation	Iris size (Iris diameter)
Optical resolution	Off-axis Orientation – sight direction	Motion blur
Pixel aspect ratio	Pupil size	Pupil to iris ratio
Pixel sampling		Sharpness
		Signal-to-noise ratio

¹ See IREX I report at http://iris.nist.gov/irex/irex_report.pdf.

		Usable iris area
--	--	------------------

1
2 IQCE will evaluate these quality metrics' influence on iris recognition accuracy. In addition, any measure identified
3 as being influential on accuracy will be considered. The outcome will be a refined list of quality metrics and if
4 possible, with tolerance bounds for each². Furthermore, IQCE will evaluate various alternative methods for
5 measuring each metric with the goal to establish a precise, open and consensus/acceptable set of methods. This
6 ensures development of a clear, tested and implementable iris image quality by excluding or avoiding any over-
7 prescriptive and non-testable statements. The overall effect of IQCE and the ISO/IEC 29794-6 standard will be to
8 validate and support camera imaging properties and system design.

9
10 The IREX I results showed that iris image quality measures that produce overall scalar quality could predict
11 performance of iris recognition algorithms. IQCE aims to evaluate the effectiveness of image quality assessment
12 algorithms (IQAAs) in predicting the recognition accuracy of a mated comparison algorithm (from or identified by the
13 supplier of the IQAA), and of others' algorithms. Furthermore, per the IREX I result that quality scores are not
14 immediately interoperable, IQCE will establish a score calibration procedure for those IQAAs that do compute an
15 overall quality score.

16
17 This activity encourages participation from the main commercial providers and academic institutions, non-profit
18 research laboratories and consultancies for which mature or prototype implementations exist.

19
20 The NIST Iris Exchange (IREX) Program was initiated at NIST in support of an expanded marketplace of iris-based
21 applications based on standardized interoperable iris imagery. IREX I was primarily conducted in support of the
22 ISO/IEC 19794-6 standard, now under revision. It secondarily supports the recently completed ANSI/NIST ITL 1-2007
23 Type 17 standard, as derived from the ISO/IEC 19794-6:2005 parent [STD05], and future revisions thereof. The Iris
24 Quality Calibration and Evaluation (IQCE) is the second activity under the IREX iris image interoperability umbrella.

25 **2. Scope**

26 Specifically IQCE aims to:

- 27 – support development of *ISO/IEC 29794 Biometric sample quality – Part 6: Iris image data* by identifying
- 28 specific iris image properties that are influential on recognition accuracy, and quantifying their effects;
- 29 – collect and document iris image quality metrics;
- 30 – quantify the performance of iris image quality assessment algorithms (IQAA);
- 31 – quantify the efficiency of IQAAs (processing time);
- 32 – quantify the robustness of IQAAs;
- 33 – study the interoperability of IQAAs;
- 34 – study the generalizability of IQAAs; and
- 35 – calibration of IQAAs.

36 The test supports:

- 37 – IQAAs that produce scalar or vector³ quantification of iris quality;
- 38 – Standalone IQAAs (image in, quality out (class X in Table 3)); and
- 39 – IQAAs that are part of an iris proprietary template generation process (image in, proprietary template +
- 40 quality out (class Y or class Z in Table 3)).

² IQCE will quantify the influence of each metric on performance of supplied matchers. While IQCE aims to arrive at a conclusion on tolerance bounds for each metric, it is not guaranteed that tolerance bounds for each metric could be decisively inferred from the results.

³ A scalar estimate of iris image quality is a single number that is quantitatively related to matching accuracy. A vector representation of overall iris image quality could include estimates of several individual metrics, e.g., motion blur, sight direction, occlusion.

1 The primary output of this evaluation will be statements of performance including

- 2 – A refined list of iris image quality metrics (with tolerance bounds when possible);
- 3 – Documentation on how to compute quality metrics as either mathematical equations or software implementations (open source or proprietary compiled libraries);
- 4 – Measurements of effectiveness of IQAAs in predicting false-non-match-rate and/or false-match rate of a particular comparison algorithm (specified by the IQAA supplier), or a class of comparison algorithms;
- 5 – Measurements of processing time (throughput) of IQAAs; and
- 6 – Calibration curve per IQAA.

9 3. Normative references

10 The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

- 11 – ISO/IEC 29794-6:201X — Information technology — Biometric sample quality — Part 6: Iris image data
- 12 – ISO/IEC 29794-1:2009 — Information technology — Biometric sample quality — Part 1: Framework
- 13 – ISO/IEC 19794-6:201X — Information technology — Biometric data interchange formats — Part 6: Iris image.

18 4. Abbreviations

19 The abbreviations and acronyms of Table 2 are used in many parts of this document.

20 **Table 2. Abbreviations**

FMR	False match rate
FNMR	False non-match rate
IQAA	Image quality assessment algorithm
IQCE	Iris quality calibration and evaluation
IREX	Generic name for the series of NIST's iris interoperability program activities
ISO/IEC 19794	Multipart standard of "Biometric data interchange formats"
ISO/IEC 29794	Multipart standard of "Biometric sample quality"

21

22 5. Motivation and background

23 Quality measurement plays a vital role in improving biometric system accuracy and efficiency during the capture process (as a control-loop variable to initiate reacquisition), in database maintenance (sample update), in enterprise-wide quality assurance surveys, and in invocation of quality-directed processing of samples, including quality-based biometric fusion. Absence or neglecting quality measurement will adversely impact accuracy and efficiency of biometric recognition systems (e.g., verification and identification of individuals).

24 Biometric sample quality assessment algorithms are intended to produce quality scores that predict performance metrics such as false match or false non-match rates. Thus, quality scores should reflect the sensitivities and failure modes of matching algorithms. The term quality should not be solely attributable to the acquisition settings of the sample, such as image resolution, dimensions in pixels, or grayscale/color bit depth. However, such factors may affect sample utility and could contribute to the overall quality score. IQCE aims at improving iris recognition technology by examining what factors affect iris recognition and quantifying their effects.

25 Quality calibration aims at quality score interpretation and interoperability by relating quality scores to performance in terms of false match rate and false non-match rate. Quality calibration supports interoperability of quality scores.

38 6. Relation to other NIST tests

39 IREX I evaluation was conducted in cooperation with the iris recognition industry to demonstrate that standardized

1 image formats can be interoperable and compact. This is required for federated applications in which iris data is
 2 exchanged between interoperating systems, passed across bandwidth-limited networks, or stored on identity
 3 credentials. IREX I quantified the core algorithmic capability of nineteen recent iris recognition implementations
 4 from ten organizations. IREX I also studied the effect of iris image compression on error rates, confirming the
 5 findings of previous studies that increasing compression gives graduated increases in false rejection.

6
 7 Three IREX I participants reported scalar overall iris image quality scores on the standard range of [0-100]. IREX I
 8 examined the predictive power of iris image quality scores, and reported that two of the quality algorithms
 9 generated quality scores correlated with image-specific error rates. Also, IREX I showed lack of interoperability of
 10 quality scores, highlighting the need for calibration.

11
 12 The IREX activities are distinct from NIST's prior Iris Challenge Evaluations (ICE) [ICE06] and ongoing Multiple
 13 Biometric Grand Challenge activities, which have more basic research goals.

14 **7. Relation to Standard**

15 ISO/IEC JTC 1 biometric subcommittee (SC 37) initiated development of the iris image quality standard (ISO/IEC
 16 29794-6 [ISOIQ]) in July 2009. The scope of the iris image quality standard is to establish

- 17 – terms and definitions that are useful in the specification, characterization and evaluation of iris image quality,
- 18 – methods used to characterize and assess the quality of iris images,
- 19 – normative requirements on software and hardware producing iris images, and
- 20 – normative requirements on software and hardware measuring utility of iris images.

21
 22 Outside the scope are

- 23 – performance assessment of specific quality algorithms, and
- 24 – standardization of specific quality algorithms.

25
 26 The current working draft (SC 37 N 3331) defines 12 quality metrics thought to be influential on segmentation and
 27 matching accuracy. That means the standardized iris image quality will be a vector of quality components where
 28 quality components are measurements of subject or image covariates. The draft is organized by the distinction of
 29 quality covariates related to:

- 30 – the design and implementation of the image acquisition equipment and environment, and
- 31 – subject-specific or subject-influenced/controlled factors.

32
 33 IQCE supports the development of ISO/IEC 29794-6 by examining how each of the quality metrics defined in ISO/IEC
 34 SC 37 N 3331 affect matching performance, and quantifying their effect on matching accuracy. IQCE also aims to
 35 establish precise statements of how to compute each of the quality metrics; this could be mathematical equations,
 36 open source software implementations or proprietary compiled libraries (e.g. DLL). This activity ensures
 37 development of a clear, tested and implementable iris image quality by excluding or avoiding any over-prescriptive
 38 and non-testable statements.

39 **8. Audience and options for participation**

40 Commercial providers, universities, and non-profit research laboratories and consultancies with capabilities in the
 41 following area are invited to participate:

- 42 – Production of quality score (overall scalar or vector) from an iris image,
- 43 – Verification using iris images.

44
 45 Participants can choose from the options specified in Table 3.

46 **Table 3. Options for participation**

Class	Category	What to submit	What NIST will do	API requirement
X	Quality only	Submission of quality	Evaluate and calibrate IQAA in terms	14.2.7 Table 10

		algorithm only (see section 9.2)	of prediction of performance with other matching algorithms (class Z matchers).	
Y	Quality and matching algorithm	Submission of quality algorithm and mated proprietary-template-generator and a matcher	Evaluate and calibrate IQAA in terms of prediction of performance with the submitted (mated) matching algorithm ONLY. A class Y matcher is not used to evaluate class X IQAAs.	Quality computation: <ul style="list-style-type: none"> if quality computation is part of template generation: 14.2.4 Table 8 if standalone quality: 14.2.7 Table 10 Template generation: 14.2.4. Table 8 matching: 14.2.5 Table 9.
Z	Matching algorithm and proprietary template generation	Submission of a proprietary template generator and matcher. A class Z proprietary template generator may generate quality vectors.	Evaluate and calibrate quality vectors generated by submitted stand-alone IQAAs (class X) or quality vectors computed by classZ proprietary template generators.	Quality computation: <ul style="list-style-type: none"> if quality computation is part of template generation: 14.2.4 Table 8 if standalone quality: 14.2.7 Table 10 Template generation: 14.2.4. Table 8 Matching: 14.2.5 Table 9

1

2 Class X SDKs will compute quality scores to be evaluated on any matcher submitted in class Z. Input is an image, and
3 output is a vector of quality metrics.

4

5 Class Y is submission of a quality algorithm, a proprietary-template-generator and a matcher. A class Y matcher will
6 only be used to evaluate its mated IQAA. Vendors can submit a stand-alone quality measurement algorithm or
7 quality computation could be part of their proprietary template generation.

8

9 Class Z is submission of a proprietary template generator (with or without quality computation) and matcher. Class
10 Z template generator and matcher will be used for class X IQAA evaluations.

11 Class Z participants can choose to compute quality scores as part of their class Z proprietary-template-generator, or
12 submit a stand-alone IQAA as a class X. Quality scores computed by a class Z proprietary-template-generator will be
13 evaluated by other class Z matchers.

14

15 Participants could choose to submit up to two SDKs per class. These two could be for the two categories of slow and
16 fast, experimental and mature, or no-image-enhancement and with-image-enhancement. IQCE will examine
17 suitability of IQAAs for applications in which processing time is constrained by other factors, e.g. wait time per
18 passenger in US-VISIT entry check-point. The timing requirement for these two categories are described in section
19 13.

20 Matching algorithms (Class Y and Z) shall return a measure of the dissimilarity (see section 14.2.6) between the two
21 irises being compared.

22

23 December 24, 2009 – Clarification on participation classes:

24 1 If submitting quality algorithm only, you are a class X participant.

25 — API requirement: compute_quality_from_image_data() (14.2.7, Table 10)

26 2 If submitting matching algorithm only, you are a class Z participant.

27 — API requirement: convert_image_to_proprietary_template()(14.2.4, Table 8) and
28 match_proprietary_templates()(14.2.5, Table 9),

29 3 If submitting quality and matching, you are either class Y or class Z participant

- 1 3.1 Class Y: If you want your quality algorithm be evaluated against your matching algorithm only, you are a class
 2 Y participant. Class Y matcher is not used to evaluate class X or class Z quality scores.
- 3 3.2 Class Z: If you want your quality algorithm be evaluated against all possible (class Z) matchers, and your
 4 matcher be used for analysis and evaluation of other class X or Z quality algorithms you are a class Z
 5 participant.
- 6 In either case (class Y or class Z), quality computation can be part of:
- 7 – template generation (image in, quality and proprietary image out)
 - 8 ○ API requirement (for quality computation): `convert_image_to_proprietary_template()`(14.2.4,
 9 Table 8), or
 - 10 – standalone (image in, quality out)
 - 11 ○ API requirement (for quality computation) `compute_quality_from_image_data()`(14.2.7, Table 10)

12 Therefore classes Y and Z use these APIs:
 13 if quality computation is part of template generation:
 14 `convert_image_to_proprietary_template()` and `match_proprietary_templates()`(14.2.5, Table 9),
 15 otherwise
 16 `compute_quality_from_image_data()`, `convert_image_to_proprietary_template()` and
 17 `match_proprietary_templates()`

18 9. Aspects of the test

19 IQCE allows IQAAs to either generate scalar quality scores or a vector of quality components or both. A vector of
 20 quality could be a set of measurements of image properties or image covariates. Vector quality quantities could be
 21 used to specifically direct reacquisition attempts (e.g., camera settings) or direct enhancement of an image (e.g.,
 22 contrast adjustment). Scalar quality is described in section 9.2.1 and vector of quality components in section 9.2.2.

23 9.1. Data

24 NIST iris corpora are a mix of data collected by various sensors, under different conditions, and collected from a
 25 mixed population. IQCE will use a larger set of images than IREX I. Particularly, NIST intends to use images with
 26 specific controlled defects for this evaluation. These images may be the result of image manipulation by NIST (e.g.
 27 blurring images) or come to NIST from dedicated data collection efforts [CLARKSON].

28
 29 NIST will document the use of operational or laboratory data for various tests and add caveats to published reports
 30 that performance values are specific to the type and characteristic of images used and not representative of any
 31 deployment where imaging is systematically dissimilar to that reflected by the imagery used by IQCE.

32 9.2. IQAA output

33 Quality computation shall be done on uncompressed iris images in one of the following forms:

- 34 – Uncompressed raw image (out of sensor).
- 35 – Centered and cropped raster iris image, where the cropping operation should extend to no closer than 0.6
 36 iris radii from the iris in the horizontal directions, and 0.2 radii in the vertical directions. This is basically the
 37 uncompressed image data in an ISO/IEC 19794-6 KIND_CROPPED iris record.

38 Use of compressed images is not considered here, mainly because operationally, quality assessment is being done
 39 immediately after capture and before compressing the image. Compression, if applied to image samples, may
 40 reduce their quality but in properly designed systems such reduction should not be great enough to significantly
 41 affect performance.

42 IQAAs can choose to produce a scalar quality score indicating overall image quality, or a vector of quality
 43 components. However the output of IQAAs shall be a vector of integers with length 64. The overall iris quality (first
 44 element of IQAA output) shall be in the range of 0-100 as specified by ISO/IEC 29794 Biometric sample quality – Part
 45 1: Framework [ISOQ]. Zero is the lowest permissible value and 100 the highest. Other elements of the quality vector
 46 shall be one-byte unsigned integer. A value of 255 means that no quality score or quality component has been
 47 computed. NIST will initialize the vector elements to 255.

1 Oct 26, 09: Length of IQAA output is extended to 64. The first element is the overall image quality, 2-17 are for
 2 standardized quality metrics, 18-32 are reserved for future standardized quality metrics, 33-64 are for vendor
 3 defined quality metrics.

4
 5 Nov 25, 09: The 0-100 requirement for all quality components are lifted. Some metrics like pupil-iris-ratio or iris size
 6 are best represented by their actual measurement values. All quality metrics shall be non-negative values, as
 7 mandated by ISO/IEC 19794-6 [ISOIRIS].

8 For the rest of this document, the term “quality score” refers to overall image quality score and if computed, is
 9 always the first element of IQAAs quality vector.

10 For IQAAs producing scalar quality score, this value shall be the first element of the output array. The next 16
 11 elements are reserved for computation of standardized image quality metrics as defined in ISO/IEC 29794-6.
 12 Elements 18-32 are reserved for future standardized iris image quality metrics. Vendor-defined quality
 13 measurements shall be placed at positions 33-64. The structure of IQAAs output is shown in Table 4.

14 **Table 4. IQAAs output format.**
 15 **The range of each metric shall be [0,254], a value of 255 means that the quality metric is not computed.**

Position	Metric
1	Scalar overall quality
2	Gray level spread
3	Iris size (iris radius in pixel)
4	Pupil iris ratio (ratio of pupil diameter over iris diameter)
5	Usable iris area (percentage of usable iris area)
6	Iris-sclera contrast
7	Iris-pupil contrast
8	Iris sclera boundary shape (iris shape)
9	Iris pupil boundary shape (pupil shape)
10	Margin (image scale in N3331)
11	Sharpness (defocus)
12	Motion blur
13	Signal to noise ratio
14	Magnification
15	Head rotation
16	Gaze angle
17	Interlace
18- 32	Reserved for future standardized quality metric
33 ... 64	Vendor-defined quality measurements

16
 17 NOTE 1 Motion blur occurs because either the iris is moving (e.g., subject or subject’s eye moving), or the device is moving (e.g.,
 18 active tracking devices). These two causes of motion blur, give rise to potentially different methods of compensation.
 19 Participants can choose to submit two distinct metrics for motion blur, one at position 10 and another one in vendor-defined
 20 portion of IQAA output (i.e. positions 33-64).

21 December 24, 2009 Question to participants: should we add square root of the ratio of the pupil area to the iris area as a new
 22 measure to Table 14? This is a more precise measure for pupil-iris-ratio since computing radius for off-gaze images are
 23 problematic and not accurate. However, this measure adds the complexity of requiring segmentation prior to its computation.

Nov 23, 2009: Table 4 has been revised.

New to 2nd draft: margin, magnification, pupil shape , interlace .

Merged back visible iris and occlusion into usable iris area.

Renamed head orientation to head rotation, gray scale density to gray level spread.

November 22, 2009: Usable iris area are merged into usable iris area per CrossMatch and CSC comments.

October 22, 2009: Usable iris area is divided into Visible iris area and Occlusion per Iritech comment.

9.2.1. Scalar overall quality

For IQAAs producing scalar quality scores, the score shall be in the range of 0-100 as specified by ISO/IEC 29794 Biometric sample quality – Part 1: Framework [ISOQ]. Scalar quality will be the first element of the output array. Score of zero is the lowest iris quality value. An iris image with quality score of zero is an unusable iris image. Similarly, score of 100 is the highest iris quality value, where lowest recognition error (or none) is expected. A value of 255 means that quality score has not been computed. Table 5 shows an example of a valid IQAA output where IQAA is only producing a scalar overall quality.

NIST evaluates scalar quality scores by quantifying their ability to predict performance.

Table 5. Example of IQAAs scalar overall quality output

Position	1	2	...	32	33	34	64
	Standardized quality metrics				vendor-defined quality metrics			
Value	43	255	255	255	255	255	255

9.2.2. Vector quality

IQAAs may produce quality scores for specific image properties (covariates). In support of development of ISO/IEC 29794 Biometric sample quality – Part 6: Iris image [ISOQI], NIST accepts IQAAs that compute the image quality metrics listed in Table 4. If reporting these measurements, these shall take positions 2-17 of the IQAA output as illustrated in Table 4. Other image covariate quality scores may be reported and placed at positions 33-64. When reporting vendor defined quality metrics (positions 33-64), IQAAs can choose to disclose and document what image covariates are being measured (gray box IQAA) or not disclose any information on the content of a vector component, i.e. what image covariate has been measured (black box IQAA). Positions 18-32 are reserved for future standardized quality metrics.

Additionally, IQAAs may generate an overall iris image quality score. If a scalar quality score representing quality of the entire image is computed, the score shall be placed in the first element of the output array.

The first element (scalar quality), if computed, shall be in the range of 0-100 where 0 means lowest and 100 means the best. Other computed elements of a vector quality shall be in the range of 0-254. A value of 255 means the quality metric is not computed. Quality scores should be monotonic functions, that is the higher score means better quality for all except positions 2,4,14,17.

The 0-100 constrain on range of quality scores for elements 2-64 has been relaxed. The intent is to report "raw" measurement instead of the normalized (to 0-100) range. However, the constraint that quality component measures be a monotonic function of quality for all elements of the quality vector, except pupil-iris-ratio, remains. For metrics listed in positions 3,5,6,7,10,11,13 of Table 4, quality component measure is clearly a monotonic function. To ensure the monotonic behavior for other metrics:

- Positions 8, 9: It is expected that circular iris and pupil shape are the easiest to process, so closer to a circle should get a higher score.
- Position 12: less motion blur is desired so 254 – motion blur will be a monotonic function.
- Positions 15, 16: frontal is best, so closer to frontal gets higher score.
- Positions 2,14, 17 depend on what will be computed by the SDKs.

1 Table 6 shows an example of a valid IQAA output where IQAA is producing a scalar overall quality (1st element), a
 2 standardized quality measurement - iris-pupil contrast score (3rd element), and two vendor-specific quality
 3 measurements at positions 33, and 34 where quality measurement at position 33 is declared to be an assessment of
 4 image-homogeneity [ZHO_09] and no description is given for the other quality component (the 34th element).

5 NIST evaluates the goodness and effectiveness of the quality metrics by relating them to recognition error.
 6
 7

Table 6. Example of IQAAs gray box vector quality output

Position	1	2	3	4 ... 32	33	34	35 ... 64
	Standardized quality metrics				vendor-defined quality metrics		
Value	69	255	73	255	57	78	255

8
 9 API function get_quality_description() takes (as input) the position of a quality component in the vector quality and
 10 returns a character string describing the component. Additionally, for each or some of the quality metrics, either
 11 standardized or vendor defined, IQAAs could document the computation methodology. This could be mathematical
 12 equations, or software implementation (e.g. open source code, or proprietary compiled libraries).

13 9.3. Vector quality summarization

14 A set of measurements that constitute a quality vector will clearly convey more information than just a summary or
 15 overall scalar value. However, the vector in itself is not immediately useful for some application e.g. comparison
 16 against a required minimum threshold. Thus, it may be necessary and useful to establish a mapping $f: \mathbb{R}^N \rightarrow \mathbb{R}^1$ of an
 17 N-element quality vector to an actionable scalar overall quality value.

18 NIST will examine several strategies for mapping of the quality vector to false non-match rate i.e. $F(v) \rightarrow$ false non-
 19 match rate. This mapping is expected to be different for each matcher (or class of matchers).

20 9.4. Measure performance

21 NIST will run supplied IQAAs on all or a selection of images in NIST iris corpora. NIST evaluates the performance of
 22 an IQAA by relating its overall quality score with recognition error rates of the comparison algorithm (matcher)
 23 supplied or identified by the IQAA provider (see table 3 Class Y), or other iris matchers submitted to this test (see
 24 table 3 Class Z). NIST examines performance (e.g FNMR) degradation as a function of quality scores. Evaluation
 25 techniques include (but not limited to) those explored in IREX I, namely error-vs-reject curves and association of
 26 image-specific error rates and quality scores.

27 Furthermore, NIST will examine how the quality metrics for both enrolment and recognition images can be
 28 evaluated jointly to better predict performance. For example, the disparity between pupil-iris diameter ratio of
 29 enrolment and recognition images may be a better indicator of match performance than the pupil-iris ratio of either
 30 sample alone. Similarly, in the case of iris occlusion, it may be more accurate to consider the total occlusion
 31 produced by the union of the sets of occluded pixels in the enrolment and recognition images, rather than the
 32 occlusion (or visible iris) of just one of the images.

33 9.5. Measure efficiency

34 NIST will measure wall time (run time) per image for each IQAA. NIST will report summary statistics of timing
 35 measurements.

36 9.6. Measure robustness

37 NIST will report the fraction of images that fail to produce an output for each IQAA.

38 9.7. Calibration curve

39 NIST will calibrate quality scores to recognition error and will report calibration curves for IQAAs.

40 IREX I showed that images given the same quality score by two different IQAAs, exhibit different image false match
 41 rate or image false non-match rates per matching algorithm or for aggregate (of fusion) results of several matchers.
 42 That means error rates expected from a quality score (e.g. 43) varies for different IQAAs. In other words, quality

1 score of, for example 43, from one IQAA would result in similar performance of images with quality score of (for
2 example) 72 and not 43 from another IQAA. This clearly makes raw quality scores non-interoperable.

3 To facilitate interoperability of quality scores, NIST intends to provide a calibration curve, which would be a mapping
4 from quality scores to recognition error rates. Calibration curves map native quality scores to error (FNM) rates
5 observed for each quality value or range of values.

6 It is important to note that calibration curve could successfully be generated for IQAAs that produce quality scores
7 indicative of performance. No calibration curve could be generated for IQAAs that produce quality scores that are
8 not indicative of performance.

9 **10. Number of submissions**

10 Organizations may enter two SDKs per Class. This would allow, for example, "fast vs. slow", or "experimental vs.
11 mature" implementations to be tested. Organizations may submit in one or more Classes (see Table 3).

12 **11. Provision of sensor information to IQAAs**

13 For each image in the corpus, NIST will provide the manufacturer and model information to the image processing
14 functions provided in the SDK. This allows the implementation to tailor its algorithms to known properties of the
15 sensor (e.g. spectral properties of the illuminant). NIST is not, however, in possession of detailed sensor
16 specifications, and it is therefore incumbent on participants to acquire such information and to use it as they see fit.

17 **12. Phased testing**

18 To support research and development, IQCE will embed multiple rounds of testing. Once the test commences, NIST
19 will test implementations on a first-come-first-served basis and will return result to providers as expeditiously as
20 possible. NIST will return results to vendors as soon as they are produced and independently of the other status of
21 other providers' implementations. The results reports will expand as revised implementations are tested.

22 Providers may submit revised SDKs to NIST only after NIST provides results for the prior SDK. The frequency with
23 which a provider may submit SDKs to NIST will depend on the times needed for vendor preparation, transmission to
24 NIST, validation, execution and scoring at NIST, and vendor review and decision processes. At any point in time, the
25 maximum number of SDKs undergoing testing at NIST will be two. NIST will invite submission of revised SDKs when
26 testing of each prior SDK has been completed.

27
28 These test rounds are intended to support improved performance. Submission period is January 26 – May 31, 2010.

29
30 Each test will result in a "score-card" provided to the participant. The score-cards will

- 31 – be machine generated (i.e. scripted);
- 32 – be provided to participants with identification of their implementation;
- 33 – include results from other implementations, but will not identify the other providers; and
- 34 – be regenerated on-the-fly, primarily whenever any implementation completes testing, or when new
35 analysis is added.

36 NIST does not intend to release these test reports. NIST may release such information to the U.S. Government test
37 sponsors. While these reports are not intended to be made public, NIST can only request that agencies not release
38 this content.

39 NIST will produce a final public report by September 2010. The final test report will publish results for the best-
40 performing implementation. Other results may be included (e.g. in appendices) to show, for example, examples of
41 progress or tradeoffs.

42 Nov 27, 09: This section has been updated to allow rounds of testing.

13. IQAA execution time

1 QAAs shall produce an output in less than 50 milliseconds for submission in the “fast” category, and less than 1.0
2 second for “slow” category. The 50 milliseconds limit of the “fast” category supports quality assessment of video
3 frames in a camera.
4

5 October 22, 2009: The limit for fast category is reduced to 50 milliseconds (from 100 milliseconds) per CSC comment
6 that 100 milliseconds execution time does not allow per frame analysis.

14. PC-based API Specification

14.1. Overview

8 This section describes the IQCE API. All SDK's submitted to IQCE shall implement the functions below as required by
9 the Classes of participation listed in Table 3.
10

14.2. Testing interface

14.2.1. Requirement

11 IREXII IQCE participants shall submit an SDK, which presents the "C" prototyped interface given in the following
12 subsections.
13

14.2.2. Sensor identifiers

14 IQCE will use images from:

- 15 – a large corpus collected using the LG 3000,
- 16 – a larger corpus collected using the Securimetrics PIER camera,
- 17 – the smaller sequestered ICE 06 corpus of LG 2200 images, and
- 18 – an even smaller set of iris images with controlled specific image impairments collected by Clarkson
19 University using a Dalsa camera.
20
21

22 To support interoperable i.e. cross-sensor matching, the SDK will be provided the sensor identifier using the two
23 byte unsigned integer values in Table 7.
24

Table 7. Sensor Identifier

#	Sensor manufacture and model	Identifier
1	LG 2200	0x2A16
2	LG 3000	0x2A1E
3	LG 4000	0x2A26
4	Securimetrics PIER	0x1A03
5	Dalsa 4M30 infrared camera	0x3A01
6	Unknown or unspecified	0x0000

25 Presence of this table indicates NIST's intention to use images captured by these devices. NIST will revise this table
26 as other data becomes available.

27 **NIST is actively seeking to extend this to include other sources - please see NIST's call for images on Page 2.**

14.2.3. Geometric, photometric or other alterations to images

28 Quality scores shall be computed on the input image without any image alteration or manipulation.
29 SDKs can choose to alter images. Their success in compensating for certain image impairments will be reflected in
30 the matching accuracy. Participants are encouraged to disclose if and what image enhancement they perform.
31

32 SDKs should report if image has been altered or not by setting output parameter image_enhanced. If an image is
33 enhanced, quality vector shall be re-calculated for the altered image and be reported in positions 32+n where n<32
34 is the position of (the standard) quality metric computed on unaltered image. Any proprietary quality metrics shall
35 be reported in positions 51-64.

1
2 Computing and reporting quality vector on un-altered and altered images allow analysis the effect of
3 iteration/enhancement on other quality metrics.

4 **14.2.4. Proprietary template creation**

5 This function converts a raw iris image into an opaque proprietary template. Two options are provided - one to
6 convert an image into a generic enrollment or verification proprietary template and another to allow two functions,
7 one for enrollment and another for verification. This "output type" aspect will be respected in Table 9. It supports
8 matching algorithms that are asymmetric.

9 Table 8 is similar to Table 11 of IREX I API.

10
11

Table 8. IREX IQCE API proprietary template creation

Prototype	<pre>uint32_t convert_image_to_proprietary_template(const uint8_t *image_data, const uint16_t image_width, const uint16_t image_height, const uint8_t scan_type, const uint8_t image_format, const uint8_t intensity_depth, const uint8_t kind const uint8_t which_eye, const uint16_t nist_encoded_device_id, const uint32_t allocated_bytes, uint8_t output_type, uint32_t *template_size, uint8_t *proprietary_template, uint8_t *quality_vector, uint8_t image_enhanced);</pre>	
Description	<p>This function takes either a raw or a centered and cropped rectilinear image, and outputs a proprietary template.</p> <p>This function is first ran for a generic template output, if fails with return code 10, it will be ran for enrollment and verification template creation.</p> <p>The memory for the output proprietary 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 proprietary template that may be passed to the match_proprietary_templates function without error. That is this routine must internally encode "proprietary template creation failed" and the matcher must transparently handle this.</p>	
Input Parameters	image_data	The uncompressed image used for proprietary 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.
	scan_type	Progressive or interlaced. Values per [ISOIRIS].
	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.
	intensity_depth	
	kind	<p>KIND_VGA = 0 (0x00) An uncompressed iris image out of sensor. This is equivalent to the image data in a KIND_VGA iris image template as specified in [ISOIRIS].</p> <p>KIND_CROPPED = 1 (0x01) An uncompressed centered and cropped iris image. This is equivalent to the image data in a KIND_CROPPED iris image template as specified in [ISOIRIS].</p>
	which_eye	<p>EYE_UNDEF = 0 (0x00) EYE_RIGHT = 1 (0x01) EYE_LEFT = 2 (0x02) These are the values used in [ISOIRIS].</p>
	nist_encoded_device_id	A two byte unsigned integer value from Table 7.

	allocated_bytes	Number of bytes NIST allocated for the output proprietary template.
	output type	Generic or enrollment and verification. PROP_TEMPLATE_GENERIC = 0 (0x00) PROP_TEMPLATE_ENROL = 1 (0x01) PROP_TEMPLATE_VERIF = 2 (0x10)
Output Parameters	template_size	The size, in bytes, of the output proprietary template.
	proprietary_template	The output proprietary iris template. The format is entirely unregulated. Maximum allocated memory is 65,536.bytes.
	quality_vector	The output quality computation results, if quality computation is done at the same time as proprietary template creation (image in, proprietary and quality out). NIST allocates and initializes this vector to 255. If quality computation is not done at the time of proprietary template creation, all 64 elements of output quality vector shall be 255.
	image_enhanced	1: alteration and enhancement performed 0: no alteration/enhancement
Return Values	0	Success
	2	Elective refusal to produce a proprietary template (e.g. insufficient iris area)
	4	Elective refusal to process the input image
	6	Involuntary failure to extract features (e.g. could not find iris in the input-image)
	8	Not enough memory – need more than <i>allocated_bytes</i>
	10	Output type not supported
	12	Null pointer
	14	Bad arguments
	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. When the error code is "2", "4", or "6" this will be noted in the IREXII IQCE report.

4 14.2.5. Proprietary templates comparison

5 This function compares two proprietary templates and returns a real-valued dissimilarity score. This function is
6 identical to Template comparison (Table 12) of [IREXAPI].
7
8

Table 9. IREX IQCE API Proprietary templates matching

Prototype	uint32_t match_proprietary_templates(const uint8_t *verification_template, const uint16_t verification_template_size, const uint8_t *enrollment_template, const uint32_t enrollment_template_size, double *dissimilarity);	
Description	This function compares two opaque proprietary templates and outputs a non-negative comparison 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 proprietary templates are the result of a failed proprietary template generation (see Table 8), the dissimilarity score shall be -1 and the function return value shall be 2.	
Input Parameters	Verification_template	A proprietary template from convert_image_to_proprietaryTemplate().
	verification_template_size	The size, in bytes, of the input verification proprietary template $0 \leq N \leq 2^{16} - 1$
	enrollment_template	A proprietary template from convert_image_to_proprietaryTemplate().
	enrollment_template_size	The size, in bytes, of the input enrollment proprietary template $0 \leq N \leq 2^{16} - 1$
Output Parameters	dissimilarity	A dissimilarity score resulting from comparison of the proprietary templates. See section 14.2.6. Oct 26, 09: range specification for dissimilarity score is deleted.
Return Values	0	Success
	2	Either or both of the input proprietary templates were result of failed template generation.

	12	NULL pointer
	14	Bad arguments
	Other	Vendor-defined failure

14.2.6. Dissimilarity

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

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

There is no requirement for values to obey the metric (or distance) property e.g. symmetry ($d(x,y) = d(y,x)$) is not required.

14.2.7. Quality computation

This function generates a vector of quality components.

Table 10. IREX IQCE API Quality computation

Prototype	<pre>uint32 compute_quality_from_image_data(const uint8_t *image_data, const uint16_t image_width, const uint16_t image_height, const uint8_t scan_type, const uint8_t image_format, const uint8_t intensity_depth, const uint8_t kind, const uint8_t which_eye, const uint16_t nist_encoded_device_id, uint8_t *quality_vector);</pre>	
Description	<p>This function takes either an uncompressed raw (out of sensor) or a centered and cropped rectilinear image, and outputs quality metric(s).</p> <p>The memory for the output quality vector is allocated and each element is initialized to 255 before the call i.e. the implementation shall not allocate memory for the result.</p>	
Input Parameters	image_data	The uncompressed image used for quality computation.
	image_width	The number of pixels indicating the width of the image.
	image_height	The number of pixels indicating the height of the image.
	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 default intensity depth will be 8.
	intensity_depth	
	kind	<p>KIND_VGA = 0 (0x00) An uncompressed iris image out of sensor. This is equivalent to the image data in a KIND_VGA iris image template as specified in [ISOIRIS].</p> <p>KIND_CROPPED = 1 (0x01) An uncompressed centered and cropped iris image. This is equivalent to the image data in a KIND_CROPPED iris image template as specified in [ISOIRIS].</p>
which_eye	<p>EYE_UNDEF = 0 (0x00) EYE_RIGHT = 1 (0x01) EYE_LEFT = 2 (0x02) These are the values used in [ISOIRIS].</p>	
nist_encoded_device_id	A two byte unsigned integer value from Table 7	
Output Parameters	Quality vector	The output quality scores. This is an array of quality computations. The first element shall be the overall image quality score. Element 2-13 contain

		standardized quality metrics as listed in Table 4. Elements 14-32 are reserved for future use and shall be 255. The rest (element 33-64) can be proprietary vendor-defined quality metric values. Function <code>get_quality_description()</code> gives description of output quality metrics i.e. what image property has been reported. See section 14.2.8.
Return Values	0	Success
	2	Elective refusal to produce quality score
	4	Iris segmentation failure
	6	Involuntary failure to process the image (e.g. could not find iris in the input-image)
	12	NULL pointer
	14	Bad arguments
	Other	Vendor-defined failure

1
2
3
4
5
6
7

14.2.8. Quality metric description

This function takes the position of a quality component in the proprietary section of output quality vector (33-64) and the letter-code of an IQAA and returns the description of the output quality metric i.e. what image properties have been computed.

Table 11. IREX IQCE API Quality description

Prototype	<code>uint32_t get_quality_description(uint8_t quality_position, char *quality_description);</code>	
Description	This function takes an integer indicating the position of a quality metric (i.e. which element of the output vector) plus SDK identifier code and returns a character string of length 120 describing what image property is measured. NIST allocates 120 bytes memory for the output <code>quality_description</code> character string, that is the implementations shall not allocate memory for the result.	
Input Parameters	<code>quality_position</code>	Integer indication the position of quality metric generated by <code>compute_quality_from_image_data</code> . It has to be in the vendor defined proprietary section of quality vector..
Output Parameters	<code>quality_description</code>	Null-terminated string description of what image property is reported e.g. dilation, or null string if no information is disclosed.
Return Values	0	Success
	2	Failed
	12	NULL pointer
	14	Bad arguments
	Other	Vendor-defined failure

8
9
10
11
12
13

14.2.9. Implementation identifiers

The implementation shall support the self-identification function of Table 12. This function is required to support internal NIST book-keeping. The version numbers should be distinct between any versions, which offer different algorithmic functionality.

Table 12. IREX IQCE API `get_pids` function

Prototype	<code>uint32_t get_pid(uint32_t *nist_assigned_identifier, char *email_address);</code>	
Description	This function retrieves an identifier that the provider must request from NIST <code>irexII@nist.gov</code> , and hardwire into the source code. NIST will assign the identifier that will uniquely identify the supplier and the SDK version number.	
Output	<code>nist_assigned_identifier</code>	A PID which identifies the SDK under test. The memory for the identifier is

Parameters	nist_assigned_identifier	allocated by NIST's calling application, and shall not be allocated by the SDK.
	Email_address	Point of contact email address as nul-terminated ASCII string. NIST will allocate at least 64 bytes for this. SDK shall not allocate memory.
Return Values	0	Success
	12	NULL pointer
	14	Bad arguments
	Other	Vendor-defined failure

1

2 15. Software and Documentation

3 15.1. SDK Library and Platform Requirements

4 Participants shall provide NIST with binary code only (i.e. no source code). It is preferred that the SDK be submitted
5 in the form of a single static library file (i.e. ".LIB" for Windows or ".a" for Linux). However, dynamic and shared
6 library files are permitted.

7

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

12

13 15.2. Linking

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

- 16 – RedHat Linux Enterprise 5 platforms based on later linux 2.6 kernels, with gcc 4.1.2-14 (PREFERRED), or
- 17 – The cygwin⁴ layer running on a Windows Server 2003 OS (might be upgraded to 2008) with linking done
18 with gcc 3.4.4.

19 The compile and link command might be:

20

- 21 – gcc -o iqctest iqctest.c -L. -liQCE

22

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

28

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

32 15.3. Installation and Usage

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

36

37 The SDK's usage shall be unlimited. The SDK shall neither implement nor enforce any usage controls or limits based

⁴ 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 on licenses, execution date/time, number of executions, presence of temporary files, etc.

2
3 It is recommended that the SDK be installable using simple file copy methods, and not require the use of a separate
4 installation program. Contact the Test Liaison for prior approval if an installation program is absolutely necessary.

5 **15.4. Documentation**

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

9 **15.5. Modes of operation**

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

13 **16. Runtime behavior**

14 **16.1. Speed**

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

- 18 – The mean proprietary template match operation should not exceed 20 milliseconds.
- 19 – The mean proprietary template creation operation should not exceed 2.5 seconds.
- 20 – The above times are wall times (run times) and assume a dual processor 2.8 GHz Pentium-based PC (with
21 dual core). Participants have to notify NIST if their submitted SDKs are taking advantage of dual core.

22 **16.2. Interactive behavior**

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

26 **16.3. Error codes and status messages**

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

29 **16.4. Exception Handling**

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

32 **16.5. External communication**

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

38 **16.6. Stateful behavior**

39 All components in this test shall be stateless. This applies to quality computation, proprietary template creation and
40 matching. Thus, all functions should give identical output, for a given input, independent of the runtime history.
41 NIST will institute appropriate tests to detect stateful behavior. If detected, NIST will take appropriate steps,
42 including but not limited to, cessation of evaluation of all implementations from the supplier, notification to the
43 provider, and documentation of the activity in published reports.

1 **17. References**

2

Label	Document
ISOQ	ISO/IEC 29794 Biometric sample quality – Part 1: Framework (201X edition)
ISOIQ	ISO/IEC 29794 Biometric sample quality – Part 6: Iris image (201X edition)
ISOIRIS	ISO/IEC 19794 Biometric data interchange formats – Part 6: Iris image (201X edition)
STD05	2005 published version of ISO/IEC 19794 Biometric data interchange formats – Part 6: Iris image
ICE06	P. Jonathon Phillips, W. Todd Scruggs, Alice J. O’Toole, Patrick J. Flynn, Kevin W. Bowyer, Cathy L. Schott, and Matthew Sharpe. “FRVT 2006 and ICE 2006 large-scale experimental results”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 99(1), 2009.
IREXAPI	http://iris.nist.gov/irex/IREX08_conops_API_v11.pdf
DOR_ICIP05	V. Dorairaj, N. Schmid, and G. Fahmy, “Performance evaluation of non-ideal iris based recognition system implementing global ICA encoding” in Proc. IEEE ICIP, 2005, vol. 3, pp. 285–288
DAU_SMC07	J. Daugman, “New methods in iris recognition”, IEEE Trans. Systems, Man, Cybernetics B 37(5), pp. 1167 - 1175. 2007
KAL_WVU05	Kalka N., “Image quality assessment for iris biometric”, MS Thesis, College of Engineering and Mineral Resources at West Virginia University, 2005
ZHO_09	Z. Zhou, Y. Du, and C. Belcher, Transforming traditional iris recognition systems to work in nonideal situations”, IEEE Trans. On Industrial Electronics, Vol. 56, NO. 8, August 2009.
CLARKSON	S. Schuckers, “Iris data collection”

3

4

Annex A

Application to participate in IREX II IQCE

A.1 Who should participate

Providers of iris recognition technologies are invited to participate in IREX II. In addition, companies, research organizations, or universities that have developed mature prototypes or who research iris quality assessment or matching are invited to participate.

The algorithms and software need not be “operational,” nor a production system, nor commercially available. However, the system must, at a minimum, be a stable implementation capable of being “wrapped” (formatted) in the API specification that NIST has specified in section 14 of this evaluation.

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

A.2 How to participate

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

- Indicate via email a non-binding "Intention to Participate" - see the schedule on Page 4.
- Request an SDK ID from NIST (for use per section 14.2.9).
- Follow the instructions for cryptographic protection of your SDK here.
- http://iris.nist.gov/irex/NIST_biometrics_crypto.pdf
- Send a signed and fully completed copy of this entire Annex A, including the IREX II Application to Participate form below. This must identify, and include signatures from, the Responsible Parties as defined in section A.4
- Provide an SDK (Software Development Kit) library, which complies with the API (Application Programming Interface) specified in this document.

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

IREX II 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.
---	--

A.3 NIST activity

A.3.1 Initiation

Upon receipt of the signed Annex A form by NIST, the organization shall be classified as a “Participant”. NIST must receive the form during the submission period described in schedule on Page 4 of this document.

A.3.2 Supplier validation

Registered Participants will be provided with a small Validation Dataset available on the website <http://iris.nist.gov/irexII>.

Prior to submission of their SDK, the Participant must verify that their software executes on the validation data, and produces correct dissimilarity 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.
4 NIST will validate all submitted materials using the participant's public key, and the authenticity of that key will be
5 verified using the key fingerprint. This fingerprint must be submitted to NIST by writing it on the signed participation
6 agreement.

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

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

11 **A.3.4 Acceptance testing**

12 Software submitted shall implement the IREX API Specification of section 14.

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

16 **A.3.5 Limits of testing**

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

23 **A.4 Parties**

24 **A.4.1 Responsible Party**

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

26 **A.4.2 Point of contact**

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

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

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

33 **A.5 Access to IREX II validation data**

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

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

37 **A.6 Access to IREX II test data**

38 The IREX II Test Datasets are in some cases protected under the Privacy Act (5 U.S.C. 552a), and will be treated as
39 Sensitive but Unclassified and/or Law Enforcement Sensitive. IREX II Participants shall have no access to IREX II Test
40 Data, before, during or after the test.

41 **A.7 Reporting of results**

1 **A.7.1 Reports**

2 The Government will combine appropriate results into one or more IREX II reports. Together these will contain, at a
3 minimum, descriptive information concerning IREX II, descriptions of each experiment, and aggregate test results.
4 NIST will compute and report aggregate statistics including (but not limited to):

- 5 — Image quality computation, proprietary template generation, and matching timing statistics;
- 6 — Predictive power of quality scores;
- 7 — Methods to summarize quality components into an overall scalar image quality;
- 8 — Interoperability of quality scores;
- 9 — Calibration of quality scores;
- 10 — Generalizability of quality scores; and
- 11 — Robustness of IQAA.

12 NIST intends to publish results in one or more NIST Interagency Reports. The reports will contain

- 13 — names of participants; and
- 14 — results of all participants' implementations with attribution to the participants.

15 **A.7.2 Pre-publication review**

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

19 **A.7.3 Citation of the report**

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

24 **A.7.4 Rights and ownership of the data**

25 Any data generated, deduced, measured or otherwise obtained during IREX II (excepting the submitted SDK itself),
26 as well as any documentation required by the Government from the participants, becomes the property of the
27 Government. Participants will not possess a proprietary interest in the data and/or submitted documentation. The
28 data and documentation will be treated as sensitive information and only be used for the purposes of this evaluation

29 **A.8 Return of the supplied materials**

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

31 **A.9 Agreement to participate**

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

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

38 With the signing of this form, Participants realize that any test details and/or modifications that are provided in the
39 IREX website supersede the information on this form.

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

42

43

This form shall be completed by all suppliers electing to participate in the IREX II 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		

- 1 With my signature, I agree that this document is a sufficient description of the test to be conducted.
- 2 With my signature, I hereby request consideration as a Participant in the iris quality calibration and evaluation (IREX
- 3 II IQCE), and I am authorizing my company or organization to participate in IREX II according to the rules and
- 4 limitations listed in this document.
- 5 With my signature, I also state that I have the authority to accept the terms stated in this document.

6 _____

7 SIGNATURE OF SOFTWARE SUPPLIER RESPONSIBLE PARTY

DATE

8
9
10