IREX II – IQCE

Iris Quality Calibration and Evaluation 2010

Concept, Evaluation Plan and API

Version 4.4

Elham Tabassi, Patrick Grother, Wayne Salamon

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

April 26, 2010

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.

January 26, 2010:
Typos fixed in Table 8, 9, and 11. Changes are marked green.

Feb 24, 2010:
Table 7 is updated + Calification text has been added to Table 11. Changes are highlighted in purple.

April 26, 2010:
A new column is added to Table 7, indicating the test dataset, which contains images captures with each device.

Handling of KIND_VGA images with margin smaller than 0,6R and 0,2R (Tables 9 and 10). The current draft of ISOIRIS (N3620) specifies horizontal margin of >0,6 R and vertical margin of >0,2R. NIST might exclude images of KIND_VGA with margin less than 0,6R and 0,2R from IQCE analysis. **Comments on handling images in our current test corpora that do not satisfy this margin requirement is requested.**

Clarification on values for scan-type in Tables 9 and 10.
IQCE will use two values for scan-type: Progressive (1) or interlaced (2). Current draft of ISOIRIS (N3620 – Table 4) has four values for scan type: corrected (0), progressive (1), interlace frame (2), and interlace filed (3), with definitions for interlace field and frame given in Table 7 of N3620.

Changes are highlighted in pink.

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.

Document History

0	First Draft, circulated for public comment	October 4, 2009
1	2nd Draft, circulated for public comment	October 22, 2009
2	3rd Draft, circulated for public review	November 27, 2009
3	4th Final Draft posted	December 24, 2009
4	Version 4.1	January 12, 2010
5	Version 4.2	January 26, 2010
6	Version 4.3	February 24, 2010
7	Version 4.4	April 26, 2010

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	3rd draft posted. Editorial or technical fixes are welcome.
November 15, 2009	2nd (and final) comment period ends.
October 22, 2009	Release of second draft evaluation plan, for comment.
October 18, 2009	Comment on 1st draft due
October 4, 2009	Launching IQCE. Announcing website, ask for comment on IQCE test plan and API.

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

Sponsors

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

Table of Contents

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

1. Overview

Recent studies such as IREX I[1] 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

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

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

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

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

2. Scope

Specifically IQCE aims to:

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

The test supports:

- IQAAs that produce scalar or vector[3] quantification of iris quality;
- Standalone IQAAs (image in, quality out (class X in Table 3)); and
- IQAAs that are part of an iris proprietary template generation process (image in, proprietary template + 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.

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

- A refined list of iris image quality metrics (with tolerance bounds when possible);
- Documentation on how to compute quality metrics as either mathematical equations or software implementations (open source or proprietary compiled libraries);
- 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;
- Measurements of processing time (throughput) of IQAAs; and
- Calibration curve per IQAA.

3. Normative references

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.

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

4. Abbreviations

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

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"

5. Motivation and background

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).

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.

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.

6. Relation to other NIST tests

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

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

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

7. Relation to Standard

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

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

Outside the scope are
— performance assessment of specific quality algorithms, and
— standardization of specific quality algorithms.

The current working draft (SC 37 N 3331) defines 12 quality metrics thought to be influential on segmentation and matching accuracy. That means the standardized iris image quality will be a vector of quality components where quality components are measurements of subject or image covariates. The draft is organized by the distinction of quality covariates related to:
— the design and implementation of the image acquisition equipment and environment, and
— subject-specific or subject-influenced/controlled factors.

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

8. Audience and options for participation

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

— Production of quality score (overall scalar or vector) from an iris image,
— Verification using iris images.

Participants can choose from the options specified in Table 3.

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: • 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: • 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

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

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

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

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

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

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

December 24, 2009 – Clarification on participation classes:

1 If submitting quality algorithm only, you are a class X participant.
   – API requirement: compute_quality_from_image_data() (14.2.7, Table 10)

2 If submitting matching algorithm only, you are a class Z participant.
   – API requirement: convert_image_to_proprietary_template()(14.2.4, Table 8) and match_proprietary_templates()(14.2.5, Table 9),

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

3.1 Class Y: If you want your quality algorithm be evaluated against your matching algorithm only, you are a class Y participant. Class Y matcher is not used to evaluate class X or class Z quality scores.

3.2 Class Z: If you want your quality algorithm be evaluated against all possible (class Z) matchers, and your matcher be used for analysis and evaluation of other class X or Z quality algorithms you are a class Z participant.

In either case (class Y or class Z), quality computation can be part of:

- template generation (image in, quality and proprietary image out)
  - API requirement (for quality computation): convert_image_to_proprietary_template()(14.2.4, Table 8), or
- standalone (image in, quality out)
  - API requirement (for quality computation) compute_quality_from_image_data()(14.2.7, Table 10)

Therefore classes Y and Z use these APIs:
if quality computation is part of template generation:
convert_image_to_proprietary_template() and match_proprietary_templates()(14.2.5, Table 9),
otherwise
compute_quality_from_image_data(), convert_image_to_proprietary_template() and match_proprietary_templates()

9. Aspects of the test

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

9.1. Data

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

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

9.2. IQAA output

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

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

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

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

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

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

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

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

Table 4. IQAAs output format.

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

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., active tracking devices). These two causes of motion blur, give rise to potentially different methods of compensation. Participants can choose to submit two distinct metrics for motion blur, one at position 10 and another one in vendor-defined portion of IQAA output (i.e. positions 33-64).

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 measure to Table 14? This is a more precise measure for pupil-iris-ratio since computing radius for off-gaze images are 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.

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

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

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

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

9.3. Vector quality summarization

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

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

9.4. Measure performance

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

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

9.5. Measure efficiency

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

9.6. Measure robustness

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

9.7. Calibration curve

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

IREX I showed that images given the same quality score by two different IQAAs, exhibit different image false match rate or image false non-match rates per matching algorithm or for aggregate (of fusion) results of several matchers. That means error rates expected from a quality score (e.g. 43) varies for different IQAAs. In other words, quality score of, for example 43, from one IQAA would result in similar performance of images with quality score of (for example) 72 and not 43 from another IQAA. This clearly makes raw quality scores non-interoperable.

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

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

10. Number of submissions

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

11. Provision of sensor information to IQAAs

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

12. Phased testing

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

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

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

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

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

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

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

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

13. IQAA execution time

IQAAs shall produce an output in less than 50 milliseconds for submission in the "fast" category, and less than 1.0 second for "slow" category. The 50 milliseconds limit of the "fast" category supports quality assessment of video frames in a camera.

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

14. PC-based API Specification

14.1. Overview

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

14.2. Testing interface

14.2.1. Requirement

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

14.2.2. Sensor identifiers

IQCE will use images from:

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

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

Table 7. Sensor Identifier

#	Sensor manufacture and model	Identifier	Dataset containing images collected with this sensor
1	LG 2200	0x2A16	ICE2006
2	LG 3000	0x2A1E	
3	LG 4000	0x2A26	
4	Securimetrics PIER	0x1A03	OPS
5	Dalsa 4M30 infrared camera	0x3A01	Clarkson
6	ISG LightWise LW-1.3-S-139	0x4A02	BATH
7	CASIA-IRISV3	0x5A15	CASIA-IrisV3_Interval
8	OKI IRISPASS-h	0x5A1F	CASIA-IrisV3-Lamp CASIA-IrisV3-Twin
9	OKI IRISPASS EQ5016A	0x6A29	Clarkson
10	Unknown or unspecified	0x0000	

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

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

Quality scores shall be computed on the input image without any image alteration or manipulation.

SDKs can choose to alter images. Their success in compensating for certain image impairments will be reflected in the matching accuracy. Participants are encouraged to disclose if and what image enhancement they perform.

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

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

14.2.4. Proprietary template creation

This function converts a raw iris image into an opaque proprietary template. Two options are provided - one to convert an image into a generic enrollment or verification proprietary template and another to allow two functions, one for enrollment and another for verification. This "output type" aspect will be respected in Table 9. It supports matching algorithms that are asymmetric.
Table 8 is similar to Table 11 of IREX I API.

Table 8. IREX IQCE API proprietary template creation

Prototype	uint32_t convert_image _to_ proprietary_template(const unit8_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);	
Description	This function takes either a raw or a centered and cropped rectilinear image, and outputs a proprietary template. 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. 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.	
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 (1) or interlaced (2). Current draft of ISOIRIS (N3620 – Table 4) has four values for scan type: corrected (0), progressive (1), interlace frame (2), and interlace filed (3), with definitions for interlace field and frame given in Table 7. IQCE will use value 2 to indicate interlaced scan-type.
	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	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]. The current draft of ISOIRIS (N3620) specifies horizontal margin of ≥0,6 R and vertical margin of ≥0,2R. Comments on handling images in our current test corpora that do not satisfy this margin requirement is requested. NIST might exclude images of KIND_VGA with margin less than 0,6R and 0,2R from IQCE analysis. 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].
	which_eye	EYE_UNDEF = 0 (0x00) EYE_RIGHT = 1 (0x01) EYE_LEFT = 2 (0x02) These are the values used in [ISOIRIS].
	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 *allocated_bytes*
	10	Output type not supported
	12	Null pointer
	14	Bad arguments
	Other	Vendor-defined failure

The number of times a non-zero error code is returned will be counted, reported and appropriately factored into analyses. When the error code is "2", “4”, or “6” this will be noted in the IREXII IQCE report.

14.2.5. Proprietary templates comparison

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

Table 9. IREX IQCE API Proprietary templates matching

Prototype	uint32_t match_ proprietary_templates(const uint8_t *verification_template, const uint32_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	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, uint8_t *image_enhanced);	
Description	This function takes either an uncompressed raw (out of sensor) or a centered and cropped rectilinear image, and outputs quality metric(s). 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.	
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 (1) or interlaced (2). Current draft of ISOIRIS (N3620 – Table 4) has four values for scan type: corrected (0), progressive (1), interlace frame (2), and interlace filed (3), with definitions for interlace field and frame given in Table 7. IQCE will use value 2 to indicate interlaced scan type.
	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	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]. The current draft of ISOIRIS (N3620) specifies horizontal margin of ≥0,6 R and vertical margin of ≥0,2R. Comments on handling images in our current test corpora that do not satisfy this margin requirement is requested. NIST might exclude images of KIND_VGA with margin less than 0,6R and 0,2R from IQCE analysis. 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].
	which_eye	EYE_UNDEF = 0 (0x00) EYE_RIGHT = 1 (0x01) EYE_LEFT = 2 (0x02) These are the values used in [ISOIRIS].
	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 get_quality_description() gives description of output quality metrics i.e. what image property has been reported. See section 14.2.8.
	image_enhanced	1: alteration and enhancement performed 0: no alteration/enhancement
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

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	uint32_t get_quality_description(uint8_t quality_position, char *quality_description);
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 including the nul character describing what image property is measured.

	NIST allocates 120 bytes memory for the output quality_description character string, that is the implementations shall not allocate memory for the result.	
Input Parameters	quality_position	Integer indication the position of quality metric generated by compute_quality_from_image_data. It will be in the vendor defined proprietary section of quality vector..
Output Parameters	quality_description	Null-terminated string description of what image property is reported e.g. dilation, or empty string if no information is disclosed.
Return Values	0	Success
	2	Failed
	12	Null pointer
	14	Bad arguments
	Other	Vendor-defined failure

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	uint32_t get_pid(uint32_t *nist_assigned_identifier, char *email_address);	
Description	This function retrieves an identifier that the provider must request from NIST irexII@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 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

15. Software and Documentation

15.1. SDK Library and Platform Requirements

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

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

15.2. Linking

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

— RedHat Linux Enterprise 5 platforms based on later linux 2.6 kernels, with gcc 4.1.2-14 (PREFERRED), or

- The cygwin[4] layer running on a Windows Server 2003 OS (might be upgraded to 2008) with linking done with gcc 3.4.4.

The compile and link command might be:

- gcc –o iqcetest iqcetest.c -L. –lIQCE

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

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

15.3. Installation and Usage

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

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

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

15.4. Documentation

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

15.5. Modes of operation

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

16. Runtime behavior

16.1. Speed

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

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

[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.

16.2. Interactive behavior

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

16.3. Error codes and status messages

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

16.4. Exception Handling

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

16.5. External communication

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

16.6. Stateful behavior

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

17. References

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	Stephanie Schuckers, Paulo Meyer Lopez, Peter Johnson, Nadezhda Sazonova, Fang Hua, Rick Lazarick, Chris Miles, Elham Tabassi, Edward Sazonov, Arun Ross, Lawrence Hornak, Quality--Face / Iris Research Ensemble (Q-FIRE) Dataset Overview, Technical Report, Clarkson University, Dept of Electrical and Computer Engineering, 2010. Stephanie Schuckers, Paulo Meyer Lopez, Peter Johnson, Nadezhda Sazonova, Fang Hua, Rick Lazarick, Chris Miles, Elham Tabassi, Edward Sazonov, Arun Ross, Lawrence Hornak, Quality--Face / Iris Research Ensemble (Q-FIRE) Data Collection Steps, Technical Report, Clarkson University, Dept of Electrical and Computer Engineering, 2010. http://www.citer.wvu.edu/quality_faceirisresearchensembleclarkson

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 A207/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.

A.3.3 Submission of software to NIST

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

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

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

A.3.4 Acceptance testing

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

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

A.3.5 Limits of testing

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

A.4 Parties

A.4.1 Responsible Party

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

A.4.2 Point of contact

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

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

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

A.5 Access to IREX II validation data

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

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

A.6 Access to IREX II test data

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

A.7 Reporting of results

A.7.1 Reports

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

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

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

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

A.7.2 Pre-publication review

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

A.7.3 Citation of the report

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

A.7.4 Rights and ownership of the data

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

A.8 Return of the supplied materials

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

A.9 Agreement to participate

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

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

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

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

<table>
<tr><td colspan="7">This form shall be completed by all suppliers electing to participate in the IREX II evaluation.</td></tr>
<tr><td colspan="7">NIST assigned identifier for the supplied SDK.</td></tr>
<tr><td colspan="7">Responsible Party for supplier of iris segmentation, encoding and/or matching technologies.</td></tr>
<tr><td colspan="7">Company / Organization Name</td></tr>
<tr><td>Title</td><td colspan="2">First Name</td><td>MI</td><td colspan="2">Last Name</td><td>Suffix</td></tr>
<tr><td colspan="7">Street Address</td></tr>
<tr><td colspan="3">City</td><td>State</td><td>Zip</td><td colspan="2">Country</td></tr>
<tr><td colspan="2">Phone</td><td colspan="2">Fax</td><td colspan="3">Email</td></tr>
<tr><td colspan="2">Technical point of contact</td><td colspan="2">Phone</td><td colspan="3">Email</td></tr>
<tr><td colspan="2">Participant's public-key fingerprint
(Enter here)</td><td colspan="5"></td></tr>
<tr><td colspan="2">NIST’s public-key fingerprint</td><td colspan="5">846E 7008 996A E912 974C F8D7 1C7A 0F22 856B 9B28</td></tr>
</table>

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

With my signature, I hereby request consideration as a Participant in the iris quality calibration and evaluation (IREX II IQCE), and I am authorizing my company or organization to participate in IREX II according to the rules and limitations listed in this document.

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

SIGNATURE OF SOFTWARE SUPPLIER RESPONSIBLE PARTY DATE