NEUROtechnology

Iris identification for stand-alone or Web solutions

VeriEye SDK

VeriEye SDK

Iris identification for stand-alone and Web solutions

Document updated on ***September 27, 2017***

CONTENTS

VeriEye iris identification technology is designed for biometric systems developers and integrators. The technology includes many proprietary solutions that enable robust iris enrollment under various conditions and fast iris matching in 1-to-1 and 1-to-many modes.

VeriEye is available as a software development kit that allows development of stand-alone and Web-based solutions on Microsoft Windows, Linux, Mac OS X, iOS and Android platforms.

- Rapid and accurate iris identification, proven by NIST IREX.
- Robust recognition, even with gazing-away eyes or narrowed eyelids.
- Original proprietary algorithm solves the limitations and drawbacks of existing state-of-the-art algorithms.
- Available as multiplatform SDK that supports multiple programming languages.
- Reasonable prices, flexible licensing and free customer support.

Features and Capabilities

Performance numbers are provided for a PC with Intel Core 2 Q9400 processor (2.67 GHz).

Neurotechnology began research and development in the field of eye iris biometrics in 1994. In 2008, Neurotechnology released **VeriEye iris recognition algorithm**. The next year VeriEye was **recognized by NIST** as one of the most reliably accurate iris recognition algorithms.

The proprietary algorithm implements advanced iris segmentation, enrollment and matching using robust digital image processing algorithms:

- **Robust iris detection.** Irises are detected even when there are obstructions to the image, visual noise and/or different levels of illumination. Lighting reflections, eyelids and eyelashes obstructions are eliminated. Images with narrowed eyelids or eyes that are gazing away are also accepted.
- **Automatic interlacing detection and correction** results in maximum quality of iris features templates from moving iris images.
- **Gazing-away eyes** are correctly detected on images, segmented and transformed as if it were looking directly into the camera (see Figure 1).
- **Correct iris segmentation** is obtained even under these conditions:
  - **Perfect circles fail.** VeriEye uses active shape models that more precisely model the contours of the eye, as iris boundaries are not modeled by perfect circles.
  - **The centers of the iris inner and outer boundaries are different** (see Figure 2). The iris inner boundary and its center are marked in red, the iris outer boundary and its center are marked in green.
  - **Iris boundaries are definitely not circles and even not ellipses** (see Figure 3) and especially in gazing-away iris images.
  - **Iris boundaries seem to be perfect circles.** The recognition quality can still be improved if boundaries are found more precisely (see Figure 4). Note these slight imperfections when compared to perfect circular white contours.
  - **Iris is partially occluded by eyelids.** The upper and lower lids are marked in red and green correspondingly (see Figure 5).
- **Fast matching.** Configurable matching speed varies from 60,000 to **548,000 comparisons per second** on a PC. See technical specifications for more details.
- **Reliability.** VeriEye 10.0 algorithm has shown excellent recognition accuracy during the NIST IREX evaluations, as well as during testing on publicly available datasets. (see *Reliability and Performance Tests Results* section).

Figure 1

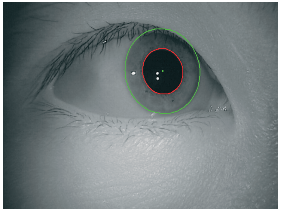


Figure 2

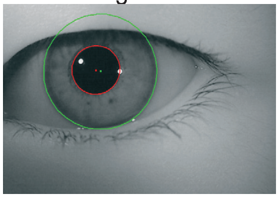


Figure 3

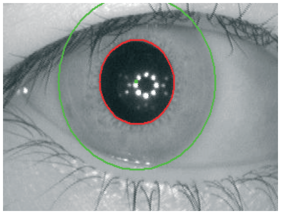


Figure 4

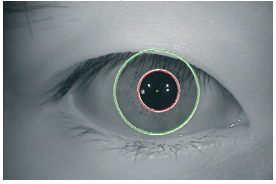
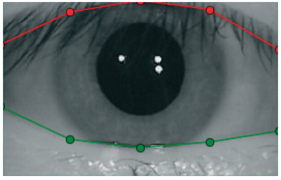


Figure 5



*All presented iris images are taken from CASIA Iris Image Database V2.0 and CASIA Iris Image Database V3.0 collected by the Chinese Academy of Sciences Institute of Automation (CASIA) (**http://www.cbsr.ia.ac.cn/english/IrisDatabases.asp**).*

Contents of VeriEye 10.0 Standard SDK and Extended SDK

VeriEye SDK is based on VeriEye iris recognition technology and is intended for biometric systems developers and integrators. The SDK allows rapid development of biometric applications using functionality from the VeriEye algorithm that ensures relable fast iris identification. VeriEye can be easily integrated into the customer's security system. The integrator has complete control over SDK data input and output.

VeriEye SDK includes the Device Manager library for working with the supported iris cameras. Integrators can also write **plug-ins to support their iris cameras** or other devices using the plug-in framework provided with the Device Manager.

VeriEye is available as the following SDKs:

- **VeriEye 10.0 Standard SDK** is designed for PC-based, embedded or mobile biometric application development. It includes Iris Matcher and Extractor component licenses, programming samples and tutorials, iris scanner support modules and software documentation. The SDK allows the development of biometric applications for Microsoft Windows, Linux, Mac OS X, iOS or Android operating systems.
- **VeriEye 10.0 Extended SDK** is designed for biometric **web-based** and network application development. It includes all features and components of the Standard SDK. Additionally, the SDK contains Iris Client component licenses for PCs and mobile / embedded devices, sample client applications, tutorials and a **ready-to-use matching server** component.

The table below compares VeriEye Standard SDK and VeriEye Extended SDK. See the licensing model for more information on specific license types.

Component licenses included with a specific SDK		
	VeriEye 10.0 Standard SDK	VeriEye 10.0 Extended SDK
• Iris Matcher	1 single computer license	1 single computer license
• Embedded Iris Matcher for Android	1 single computer license	1 single computer license
• Embedded Iris Matcher for iOS	1 single computer license	1 single computer license
• Embedded Iris Matcher for ARM Linux	1 single computer license	1 single computer license
• Iris Client [(1)]		3 single computer licenses
• Embedded Iris Client for Android		3 single computer licenses
• Embedded Iris Client for iOS		3 single computer licenses
• Embedded Iris Client for ARM Linux		3 single computer licenses
• Iris Extractor	1 single computer license	1 single computer license
• Embedded Iris Extractor for Android	1 single computer license	1 single computer license
• Embedded Iris Extractor for iOS	1 single computer license	1 single computer license
• Embedded Iris Extractor for ARM Linux	1 single computer license	1 single computer license
• Matching Server		+

(1) Iris Client component includes Iris Extractor and Iris BSS components, which can be also obtained separately.

VeriEye 10.0 SDK includes programming samples and tutorials that show how to use the components of the SDK to perform face template extraction or matching against other templates. The samples and tutorials are available for these programming languages and platforms:

	Windows	Linux	Mac OS X	iOS	Android
Programming samples and tutorials					
• C/C++	+	+	+		
• Objective-C				+	
• C#	+				
• Java	+	+	+		+
• Visual Basic .NET	+				

Biometric Components Description

Iris Matcher

The Iris Matcher performs iris template matching in 1-to-1 (verification) and 1-to-many (identification) modes on PC or Mac platform. Also the Iris Matcher component includes fused matching algorithm that allows to increase template matching reliability by:

- matching templates that contain 2 iris records;
- matching templates that contain fingerprint, face, voiceprint and/or iris records (note that matching fingerprints, faces and voiceprints requires Fingerprint Matcher, Face Matcher and Voice Matcher components correspondingly - see *VeriFinger SDK, VeriLook SDK* and *VeriSpeak SDK* brochures for more information);

The Iris Matcher component matches **40,000 irises per second** and is designed to be used in **desktop** or mobile biometric systems, which run on PCs or laptops with at least Intel **Core 2 Q9400** (2.67 GHz) processor.

One Iris Matcher license is included with VeriEye 10.0 Standard SDK and VeriEye 10.0 Extended SDK. More licenses for this component can be purchased any time by VeriEye 10.0 SDK customers.

Embedded Iris Matcher

The Embedded Iris Matcher has the same functionality, as the Iris Matcher. It matches **3,000 irises per second** and is designed to be used in **embedded** or **mobile** biometric systems, which run on ARM Linux, **Android** or **iOS** devices. The Android devices should be based on at least **Snapdragon S4** system-on-chip (**Krait 300** processor with 4 cores running at 1.51 GHz).

One Embedded Iris Matcher license for each of Android, iOS and ARM Linux platforms is included with VeriEye 10.0 Standard SDK and VeriEye 10.0 Extended SDK. More licenses for this component can be purchased any time by VeriEye 10.0 SDK customers.

Iris Client

The Iris Client component is a combination of the Iris Extractor and Iris BSS components. It is designed for the systems that need to support all functionality of the mentioned components on the same PC. Using these licenses allows to optimize component license costs as well as reduce license management.

The Iris Client extracts a single iris template in **0.6 seconds**. The specified performance requires a **PC or laptop** with at least Intel **Core 2 Q9400** (2.67 GHz) processor.

Three licenses for the Iris Client component are included with VeriEye 10.0 Extended SDK. More licenses for this component can be purchased any time by VeriEye 10.0 Extended SDK customers.

Embedded Iris Client

The Embedded Iris Client component has the same functionality as the Iris Client and is designed to run on ARM Linux, **Android** or **iOS** devices. The Android devices should be based on at least **Snapdragon S4** system-on-chip (**Krait 300** processor with 4 cores running at 1.51 GHz). The component extracts a single iris template in **1.2 seconds**.

Three licenses for the Embedded Iris Client component for each of Android, iOS and ARM Linux platforms are included with VeriEye 10.0 Extended SDK. More licenses for this component can be purchased any time by VeriEye 10.0 Extended SDK customers.

Iris Extractor

Iris Extractor creates iris templates from eye images.

The component extracts a single iris template in **1.34 seconds**. The specified performance requires a **PC or laptop** with at least Intel **Core 2 Q9400** (2.67 GHz) processor.

One Iris Extractor license is included with VeriEye 10.0 Standard SDK and VeriEye 10.0 Extended SDK. More licenses for this component can be purchased any time by VeriEye 10.0 SDK customers.

Embedded Iris Extractor

The Embedded Iris Extractor has the same functionality as the Iris Extractor and is designed to be run on ARM Linux, **Android** or **iOS** devices. The Android devices should be based on at least **Snapdragon S4** system-on-chip (**Krait 300** processor with 4 cores running at 1.51 GHz). The component extracts a single iris template in **1.34 seconds**.

One Embedded Iris Extractor license is included with VeriEye 10.0 Standard SDK and VeriEye 10.0 Extended SDK. More licenses for this component can be purchased any time by VeriEye 10.0 SDK customers.

Iris BSS (Biometric Standards Support)

The Iris BSS (Biometric Standards Support) component allows to integrate support for iris image format standards and additional image formats with new or existing biometric systems based on VeriEye SDK.

These biometric standards are supported:

- **BioAPI 2.0** (ISO/IEC 19784-1:2006) (Framework and Biometric Service Provider for iris identification engine)
- **CBEFF V1.2 (ANSI INCITS 398-2008)** (Common Biometric Exchange Formats Framework)
- **CBEFF V2.0 (ISO/IEC 19785-1:2006 with Amd. 1:2010, 19785-3:2007 with Amd. 1:2010)** (Common Biometric Exchange Formats Framework)
- **ISO/IEC 19794-6:2005** (Biometric Data Interchange Formats - Iris Image Data)
- **ISO/IEC 19794-6:2011** with **Cor. 1:2012**
- **ISO/IEC 29794-6:2015** (Biometric Sample Quality - Iris Image Data)
- **ANSI/INCITS 379-2004** (Iris Image Interchange Format)
- **ANSI/NIST-ITL 1-2007** (Data Format for the Interchange of Fingerprint, Facial, & Other Biometric Information)
- **ANSI/NIST-ITL 1a-2009** (Data Format for the Interchange of Fingerprint, Facial, & Other Biometric Information)
- **ANSI/NIST-ITL 1-2011** (Data Format for the Interchange of Fingerprint, Facial, & Other Biometric Information)
- **ANSI/NIST-ITL 1-2011 Update:2013 Edition 2** (Data Format for the Interchange of Fingerprint, Facial, & Other Biometric Information)
- **ANSI/NIST-ITL 1-2011 Update:2015** (Data Format for the Interchange of Fingerprint, Facial, & Other Biometric Information)

The component is designed for applications that run on hardware with at least Intel **Core 2 Q9400** (2.67 GHz) processor. It can be used from C/C++, C# and Java applications on all supported platforms. .NET wrappers of Windows libraries are provided for .NET developers.

Licenses for the Iris BSS component can be purchased anytime by VeriEye 10.0 Extended SDK customers.

Matching Server

The Matching Server is ready-to-use software intended for building moderate size web-based and other network-based systems like local single- or multi-biometric identification system. The Server software runs on a server PC and allows to perform the biometric template matching on server side using Iris Matcher component.

Multi-biometric matching can be enabled by running components for iris, fingerprint, face and voiceprint matching on the same machine.

Client communication module that allows sending a task to the Matching Server, querying status of the task, getting the results and removing the task from server, is included with MegaMatcher 10.0 SDK, VeriFinger 10.0 SDK, VeriLook 10.0 SDK, VeriSpeak 10.0 SDK and VeriEye 10.0 SDK. This module hides all low level communications and provides high-level API for the developer.

The components and database support modules with source codes included for Matching Server component are listed in the table below. Custom modules for working with other databases can also be developed by integrator and used with the Matching Server software.

The table below shows what components are available with Matching Server software.

Components	Microsoft Windows 32 & 64 bit	Linux 32 & 64 bit	Mac OS X
• Matching server software	+	+	+
• Server administration tool API	+	+	
• Source code of sample web server software	+		
Database support modules			
• Microsoft SQL Server	+		
• PostgreSQL	+	+	
• MySQL	+	+	
• Oracle	+	+	
• SQLite	+	+	+
Programming samples			
• C# client	+		
• Visual Basic .NET client	+		
• Java web client	+	+	+
Programming tutorials			
• C/C++	+	+	
• C#	+		
• Visual Basic .NET	+		

The Matching Server component requires a **special license** that allows to run the component on all machines that run the fingerprint, face, iris, voiceprint or palm print matching components obtained by an integrator. The Matching Server software is included with VeriEye 10.0 Extended SDK.

Also the Matching Server component is included with these Neurotechnology SDKs (see their brochures for more info):

- MegaMatcher 10.0 Standard or MegaMatcher 10.0 Extended SDK;
- VeriFinger 10.0 Extended SDK;
- VeriLook 10.0 Extended SDK.
- VeriSpeak 10.0 Extended SDK.

Supported Iris Cameras

The table below explains which iris scanners are supported under different versions of Microsoft Windows.

Integrators or scanner manufacturers can also write **plug-ins** for the Device Manager from the VeriEye SDK to **support their iris cameras** using the provided plug-in framework. The SDK documentation contains more information about the plug-in framework.

	Microsoft Windows Vista / 7 / 8		Linux (32-bit only)	Android
	32 bit	64 bit		
• CMITech BMT-20 / EMX-30	+	+		
• Credence ID Trident				+[2]
• Cross Match I Scan 2	+	+[1]		
• Iritech IriShield USB MK 2120U / IriShield-USB BK 2121U	+	+	+	+
• Iritech IriMagic1000BK	+	+[1]		
• VistaFA2 / VistaFA2E / VistaEY2 / VistaEY2-02 / VistaEY2R / VistaEY2H	+	+		

Notes:
(1) Can be used on 64-bit OS, but only in 32-bit applications.
(2) The device has pre-installed Android OS

System requirements

There are specific requirements for each platform which will run VeriEye-based applications.

Microsoft Windows platform requirements

- Microsoft Windows **Vista** / **7** / **8** / **10**, 32-bit or 64-bit.
    - Note that some iris scanners will work only from 32-bit applications.
    - Windows XP is no longer supported in this version of the SDK. If your product requires to support Windows XP, you may consider the previous version of the SDK. Please contact us for more information.
- PC or laptop with **x86 (32-bit)** or **x86-64 (64-bit)** compatible processors.
    - 2 GHz or better processor is recommended.
    - **SSE2 support is required.** Processors that do not support SSE2 cannot run the VeriEye algorithm. Please check if a particular processor model supports SSE2 instruction set.
- At least **128 MB of free RAM** should be available for the application. Additional RAM is required for applications that perform 1-to-many identification, as all biometric templates need to be stored in RAM for matching. For example, **50,000 templates** (each with 2 iris records) require about **223 MB of additional RAM**.
- **Free space on hard disk drive (HDD):**
    - at least 1 GB required for the development.
    - 100 MB required for VeriEye components deployment.
    - Additional space would be required in these cases:
        - VeriEye does not require the original eye image to be stored for the matching; only the templates need to be stored. However, storing eye irises images on hard drive for the potential future usage is recommended.
        - Usually a database engine runs on a separate computer (back-end server). However, DB engine can be installed on the same computer for standalone applications. In this case HDD space for templates storage must be available. For example, 50,000 templates (each containing 2 iris records) stored using a relational database would require about 223 MB of free HDD space.
- **Iris scanner.** VeriEye SDK includes support modules for several iris scanners under Microsoft Windows platform. Integrators can also write plug-ins to support their iris scanners using the plug-in framework provided with the Device Manager from the VeriEye SDK.
- **Database engine** or connection with it. VeriEye templates can be saved into any DB (including files) supporting binary data saving. VeriEye Extended SDK contains the following support modules for Matching Server on Microsoft Windows platform: **Microsoft SQL Server**, **MySQL**, **Oracle**, **PostgreSQL** and **SQLite.**
- **Network/LAN connection (TCP/IP)** for client/server applications. Also, network connection is required for using Matching server component (included in VeriEye Extended SDK). Communication with Matching server is not encrypted, therefore, if communication must be secured, a dedicated network (not accessible outside the system) or a secured network (such as VPN; VPN must be configured using operating system or third party tools) is recommended.
- Microsoft **.NET framework 3.5** or newer (for .NET components usage).
- One of following **development environments** for application development:
    - Microsoft Visual Studio 2012 or newer (for application development under C/C++, C#, Visual Basic .Net)
    - Sun Java 1.6 SDK or later.

Android platform requirements

- A smartphone or tablet that is running **Android 4.4 (API level 19)** OS or newer.
    - API level 22 is the recommended **target** for code compilation.
    - If you have a custom Android-based device or development board, contact us to find out if it is supported.
- ARM-based **1.5 GHz processor recommended** for iris processing in the specified time. Slower processors may be also used, but the iris processing will take longer time.
- At least **20 MB of free RAM** should be available for the application. Additional RAM is required for applications that perform 1-to-many identification, as all biometric templates need to be stored in RAM for matching. For example, **1,000 templates** (each with 1 iris records) require about **2.5 MB of additional RAM**.
- **Free storage** space (built-in flash or external memory card):
    - 30 MB required for embedded iris components deployment for each separate application.
    - Additional space would be required if an application needs to store original iris images. VeriEye does not require the original image to be stored for the matching; only the templates need to be stored.
- **Iris scanner.**
    - VeriEye SDK includes support modules for several iris scanners under Android platform.
    - Iris images in **BMP**, **JPG** or **PNG** formats can be processed thus almost any third-party iris capturing hardware can be used with the VeriEye technology if it generates images in the mentioned formats.
    - Integrators may also write **plug-ins to support their iris cameras** using the plug-in framework provided with the Device Manager from the VeriEye SDK. The integrators should note, that the most accurate iris recognition is achievable only when iris images are captured with near-infrared cameras and appropriate illumination. However, it is still possible to recognize irises with reasonable accuracy, when the irises are captured with cameras, which are built in smartphones or tablets, using proper illumination and focus, and choosing proper environment. See our testing results for details.
- **Network/LAN connection (TCP/IP)** for client/server applications. Also, network connection is required for using Matching server component (included in VeriEye Extended SDK). Communication with Matching server is not encrypted, therefore, if communication must be secured, a dedicated network (not accessible outside the system) or a secured network (such as VPN; VPN must be configured using operating system or third party tools) is recommended.
- **PC-side development** environment requirements:
    - Java SE JDK 6 (or higher)
    - Eclipse Indigo (3.7) IDE
    - Android development environment (at least API level 19 required)
    - One of the following build automation systems:
        - Apache Maven 3.1.x or newer
        - Gradle 2.10 or newer
    - Internet connection for activating VeriEye component licenses

iOS platform requirements

- One of the following devices, running **iOS 8.0** or newer:
  - **iPhone 5** or newer iPhone.
  - **iPad 2** or newer iPad, including iPad Mini and iPad Air models.
  - **iPod Touch 6th Generation** or newer iPod.
- At least **20 MB of free RAM** should be available for the application. Additional RAM is required for applications that perform 1-to-many identification, as all biometric templates need to be stored in RAM for matching. For example, **1,000 templates** (each with 1 iris records) require about **2.5 MB of additional RAM**.
- **Free storage** space (built-in flash or external memory card):
  - 30 MB required for embedded iris components deployment for each separate application.
  - Additional space would be required if an application needs to store original iris images. VeriEye does not require the original image to be stored for the matching; only the templates need to be stored.
- **Iris scanner.**
  - At the moment iris scanner support on iOS platform should be implemented by integrators. The integrators should note, that the most accurate iris recognition is achievable only when iris images are captured with near-infrared cameras and appropriate illumination. However, it is still possible to recognize irises with reasonable accuracy, when the irises are captured with cameras, which are built in smartphones or tablets, using proper illumination and focus, and choosing proper environment. See our testing results for details.
  - Iris images in **BMP**, **JPG** or **PNG** formats can be processed thus almost any third-party iris capturing hardware can be used with the VeriEye technology if it generates images in the mentioned formats.
- **Network/LAN connection (TCP/IP)** for client/server applications. Also, network connection is required for using Matching server component (included in VeriEye Extended SDK). Communication with Matching server is not encrypted, therefore, if communication must be secured, a dedicated network (not accessible outside the system) or a secured network (such as VPN; VPN must be configured using operating system or third party tools) is recommended.
- **Development environment** requirements:
  - a Mac running Mac OS X 10.10.x or newer.
  - Xcode 6.4 or newer.

Mac OS X platform requirements

- A Mac running **Mac OS X 10.7.x** or newer. 2 GHz or better processor is recommended.
- At least **128 MB of free RAM** should be available for the application. Additional RAM is required for applications that perform 1-to-many identification, as all biometric templates need to be stored in RAM for matching. For example, **50,000 templates** (each with 2 iris records) require about **223 MB of additional RAM**.
- **Free space on hard disk drive (HDD):**
  - at least 1 GB required for the development.
  - 100 MB required for VeriEye components deployment.
  - Additional space would be required in these cases:
    - VeriEye does not require the original eye image to be stored for the matching; only the templates need to be stored. However, storing eye irises images on hard drive for the potential future usage is recommended.
    - Usually a database engine runs on a separate computer (back-end server). However, DB engine can be installed on the same computer for standalone applications. In this case HDD space for templates storage must be available. For example, 50,000 templates (each containing 2 iris records) stored using a relational database would require about 223 MB of free HDD space. Also, the database engine itself requires HDD space for running. Please refer to HDD space requirements from the database engine providers.
- **Iris scanner.**
  - At the moment iris scanner support on Mac OS X platform should be implemented by integrators. The integrators should note, that the most accurate iris recognition is achievable only when iris images are captured with near-infrared cameras and appropriate illumination. However, it is still possible to recognize irises with reasonable accuracy, when the irises are captured with cameras, which are built in smartphones or tablets, using proper illumination and focus, and choosing proper environment. See our testing results for details.
  - Iris images in **BMP**, **JPG** or **PNG** formats can be processed thus almost any third-party iris capturing hardware can be used with the VeriEye technology if it generates images in the mentioned formats.
- **Database engine** or connection with it. VeriEye templates can be saved into any DB (including files) supporting binary data saving. VeriEye Extended SDK contains **SQLite** support modules for Matching Server on Mac OS X platform**.**
- **Network/LAN connection (TCP/IP)** for client/server applications. Also, network connection is required for using Matching server component (included in VeriEye Extended SDK). Communication with Matching server is not encrypted, therefore, if communication must be secured, a dedicated network (not accessible outside the system) or a secured network (such as VPN; VPN must be configured using operating system or third party tools) is recommended.
- Specific requirements for **application development**:
  - XCode 4.3 or newer
  - wxWidgets 3.0.0 or newer libs and dev packages (to build and run SDK samples and applications based on them)
  - Qt 4.8 or newer libs, dev and qmake packages (to build and run SDK samples and applications based on them)
  - GNU Make 3.81 or newer (to build samples and tutorials development)
  - Sun Java 1.6 SDK or later.

Linux x86 / x86-64 platform requirements

- Linux 2.6 or newer kernel (32-bit or 64-bit) is required. **Linux 3.0 kernel** or newer is recommended.
- PC or laptop with **x86 (32-bit)** or **x86-64 (64-bit)** compatible processors.
  - 2 GHz or better processor is recommended.
  - **SSE2 support is required.** Processors that do not support SSE2 cannot run the VeriEye algorithm. Please check if a particular processor model supports SSE2 instruction set.
- At least **128 MB of free RAM** should be available for the application. Additional RAM is required for applications that perform 1-to-many identification, as all biometric templates need to be stored in RAM for matching. For example, **50,000 templates** (each with 2 iris records) require about **223 MB of additional RAM**.
- **Free space on hard disk drive (HDD):**
  - at least 1 GB required for the development.
  - 100 MB required for VeriEye components deployment.
  - Additional space would be required in these cases:
    - VeriEye does not require the original eye image to be stored for the matching; only the templates need to be stored. However, storing eye irises images on hard drive for the potential future usage is recommended.
    - Usually a database engine runs on a separate computer (back-end server). However, DB engine can be installed on the same computer for standalone applications. In this case HDD space for templates storage must be available.
- **Iris scanner.**
  - VeriEye SDK includes support modules for several iris scanners under Linux platform.
  - Iris images in **BMP**, **JPG** or **PNG** formats can be processed thus almost any third-party iris capturing hardware can be used with the VeriEye technology if it generates images in the mentioned formats.
  - Integrators may also write **plug-ins to support their iris cameras** using the plug-in framework provided with the Device Manager from the VeriEye SDK. The integrators should note, that the most accurate iris recognition is achievable only when iris images are captured with near-infrared cameras and appropriate illumination. However, it is still possible to recognize irises with reasonable accuracy, when the irises are captured with cameras, which are built in smartphones or tablets, using proper illumination and focus, and choosing proper environment. See our testing results for details.
- glibc 2.11.3 library or newer
- **Database engine** or connection with it. VeriEye templates can be saved into any DB (including files) supporting binary data saving. VeriEye Extended SDK contains **MySQL, Oracle, PostgreSQL** and **SQLite** support modules for Matching Server on Linux x86 / x86-64 platforms**.**
- **Network/LAN connection (TCP/IP)** for client/server applications. Also, network connection is required for using Matching server component (included in VeriEye Extended SDK). Communication with Matching server is not encrypted, therefore, if communication must be secured, a dedicated network (not accessible outside the system) or a secured network (such as VPN; VPN must be configured using operating system or third party tools) is recommended.
- Specific requirements for **application development**:
  - wxWidgets 3.0.0 or newer libs and dev packages (to build and run SDK samples and applications based on them)
  - Qt 4.8 or newer libs, dev and qmake packages (to build and run SDK samples and applications based on them)
  - GCC-4.4.x or newer
  - GNU Make 3.81 or newer (to build samples and tutorials development)
  - Sun Java 1.6 SDK or later.
  - pkg-config-0.21 or newer (optional; only for Matching Server database support modules compilation)

ARM Linux platform requirements

We recommend to contact us and report the specifications of a target device to find out if it will be suitable for running VeriEye-based applications. There is a list of common requirements for ARM Linux platform:

- A device with ARM-based processor, running **Linux 3.2 kernel** or newer.
- ARM-based **1.5 GHz processor recommended** for iris processing in the specified time. .
  - **ARMHF** architecture (**EABI 32-bit hard-float ARMv7**) is required.
  - Lower clock-rate processors may be also used, but the iris processing will take longer time
- At least **20 MB of free RAM** should be available for the application. Additional RAM is required for applications that perform 1-to-many identification, as all biometric templates need to be stored in RAM for matching. For example, **1,000 templates** (each with 1 iris records) require about **2.5 MB of additional RAM**.
- **Free storage** space (built-in flash or external memory card):
  - 100 MB required for embedded iris components deployment for each separate application.
  - Additional space would be required if an application needs to store original iris images. VeriEye does not require the original image to be stored for the matching; only the templates need to be stored.
- **Iris scanner.**
  - At the moment iris scanner support on ARM Linux platform should be implemented by integrators. The integrators should note, that the most accurate iris recognition is achievable only when iris images are captured with near-infrared cameras and appropriate illumination. However, it is still possible to recognize irises with reasonable accuracy, when the irises are captured with cameras, which are built in smartphones or tablets, using proper illumination and focus, and choosing proper environment. See our testing results for details.
  - Iris images in **BMP**, **JPG** or **PNG** formats can be processed thus almost any third-party iris capturing hardware can be used with the VeriEye technology if it generates images in the mentioned formats.
- glibc 2.13 library or newer
- libstdc++-v3 4.7.2 or newer.
- **Network/LAN connection (TCP/IP)** for client/server applications. Also, network connection is required for using Matching server component (included in VeriEye Extended SDK). Communication with Matching server is not encrypted, therefore, if communication must be secured, a dedicated network (not accessible outside the system) or a secured network (such as VPN; VPN must be configured using operating system or third party tools) is recommended.
- **Development environment** requirements:
  - GCC-4.4.x or newer
  - GNU Make 3.81 or newer
  - JDK 1.6 or later

Technical Specifications

64 pixels is the minimal radius of circle containing full iris texture, that is required for iris template extraction.

Near-infrared spectral region is recommended for iris image capture.

All iris templates should be loaded into RAM before identification, thus the maximum iris template database size is limited by the amount of available RAM.

VeriEye biometric template extraction and matching algorithm is designed to run on **multi-core processors** allowing to reach maximum possible performance on the used hardware.

VeriEye 10.0 iris engine specifications				
	Embedded / mobile platform [1]		PC-based platform [2]	
Template extraction components	**Embedded Iris Extractor**	**Embedded Iris Client**	**Iris Extractor**	**Iris Client**
Iris template extraction time (seconds)	1.34	1.20	1.34	0.60
Template matching components	**Embedded Iris Matcher**		**Iris Matcher**	
Matching speed (Irises per second)	3,000		40,000	
Single iris record size in a template (bytes)	2,348			

Notes:

(1) Requires to be run on iOS or Android devices based on at least Snapdragon S4 system-on-chip with Krait 300 processor (4 cores, 1.51 GHz).

(2) Requires to be run on PC or laptop with at least Intel Core 2 Q9400 quad-core processor (2.67 GHz) to reach the specified performance.

Reliability Test Results

We present the testing results to show VeriEye 10.0 template matching algorithm reliability evaluations. Iris images from several **standard datasets** were used for testing, thus the testing results can be compared with testing results of other algorithms. All datasets contained iris images with 640 x 480 pixels size.

Iris image datasets used for VeriEye 10.0 algorithm testing

	ICE2005 Exp1	ND-IRIS-0405	IRISDB1600	MobileIrisV10
Image count	1,425	64,980	24,361	3,290
Subject count	124	356	624	70
Unique iris count	124	712	1231	135
Session count	1 - 31	4 - 291	1 - 40	6 - 42

Notes:

1. The ICE2005 dataset was collected by the National Institute of Standards and Technology (NIST). Near-infrared spectrum equipment was used for iris capture. ICE2005 Exp1 is a subset, which contains right iris images.
2. The ND-IRIS-0405 was collected by the University of Notre Dame. Near-infrared spectrum equipment was used for iris capture.
3. The IRISDB1600 was collected by the University of Bath. Near-infrared spectrum equipment was used for iris capture. The full IRISDB1600 dataset contains 31,997 images (image size 1280x960 pixels), which represented 799 unique persons and 1,598 unique irises. A subset used in this test was preprocessed similar to NIST IREX experiments – the images were downsampled to 640x480 via 2x2 neighborhood averaging, and all images containing irises with diameters larger than 340 pixels were removed.
4. The MobileIrisV10 dataset was collected by the Warsaw University of Technology. The iris image collection was performed using regular, **visible light spectrum** camera built-in into Apple iPhone 5S smartphone. Colored images were collected with the camera. The images were resized to 640x480 pixels and converted to grayscale. See the scientific paper (*http://zbum.ia.pw.edu.pl/PAPERS/Wilga2015_SmartphoneIris.pdf*) for more details.

Two tests were performed with each database:

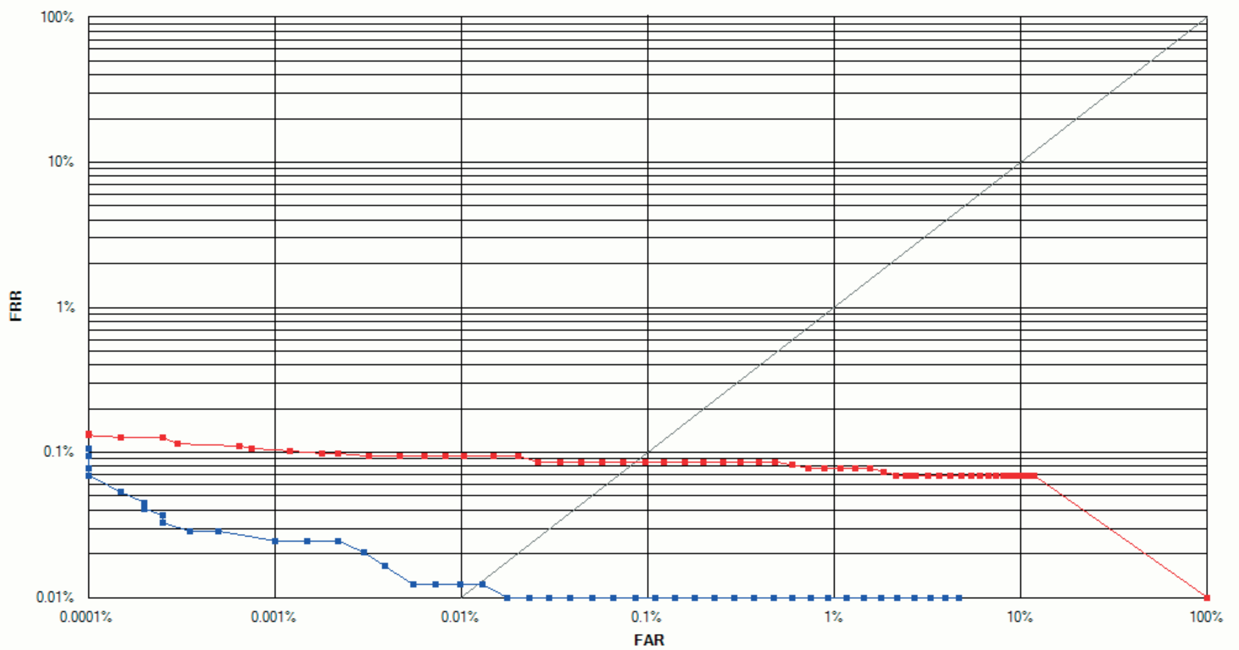
- **Test 1** maximized **matching accuracy**. VeriEye 10.0 algorithm reliability in this test is shown as **blue curves** on the ROC charts.
- **Test 2** maximized **matching speed**. VeriEye 10.0 algorithm reliability in this test is shown as **red curves** on the ROC charts.

The iris rotation tolerance was set to **±15°** in all tests.

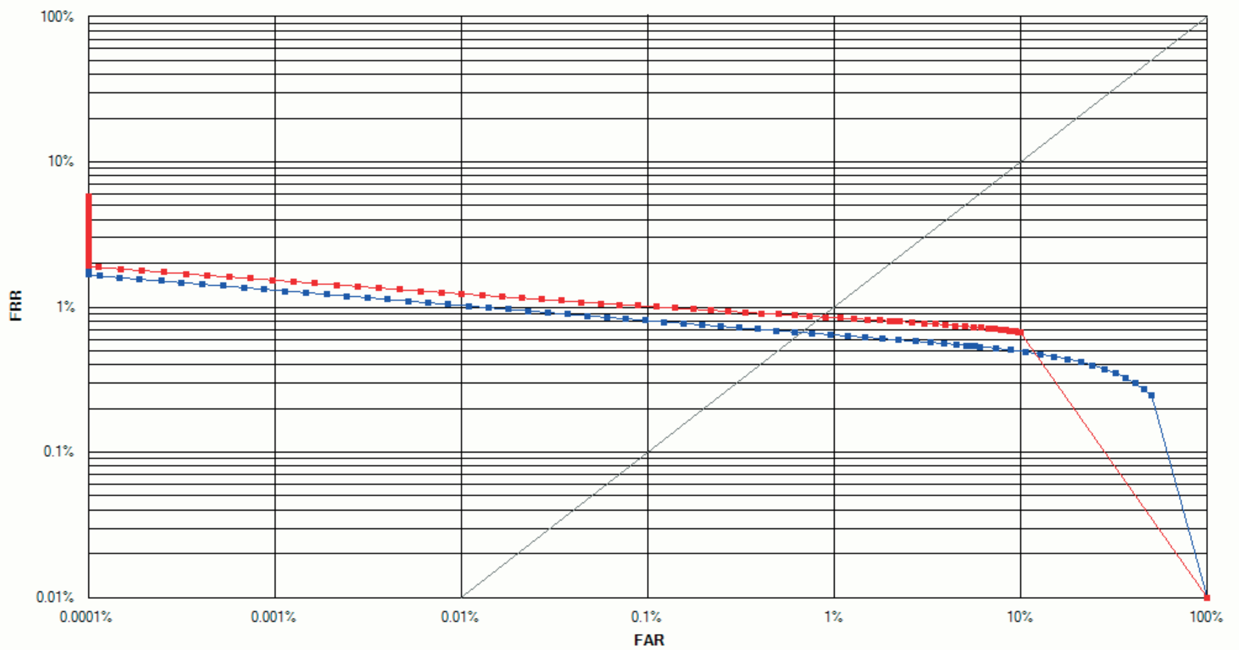
VeriEye 10.0 algorithm reliability testing results

	ICE2005 Exp1		ND-IRIS-0405		IRISDB1600		MobileIrisV10	
	Test 1	Test 2	Test 1	Test 2	Test 1	Test 2	Test 1	Test 2
FRR at 0.01 % FAR	0.0123 %	0.0942 %	1.0420 %	1.2590 %	0.0373 %	0.0496 %	0.0527 %	1.1490 %
FRR at 0.001 % FAR	0.0246 %	0.1064 %	1.3240 %	1.5300 %	0.0453 %	0.0587 %	0.0527 %	1.3380 %

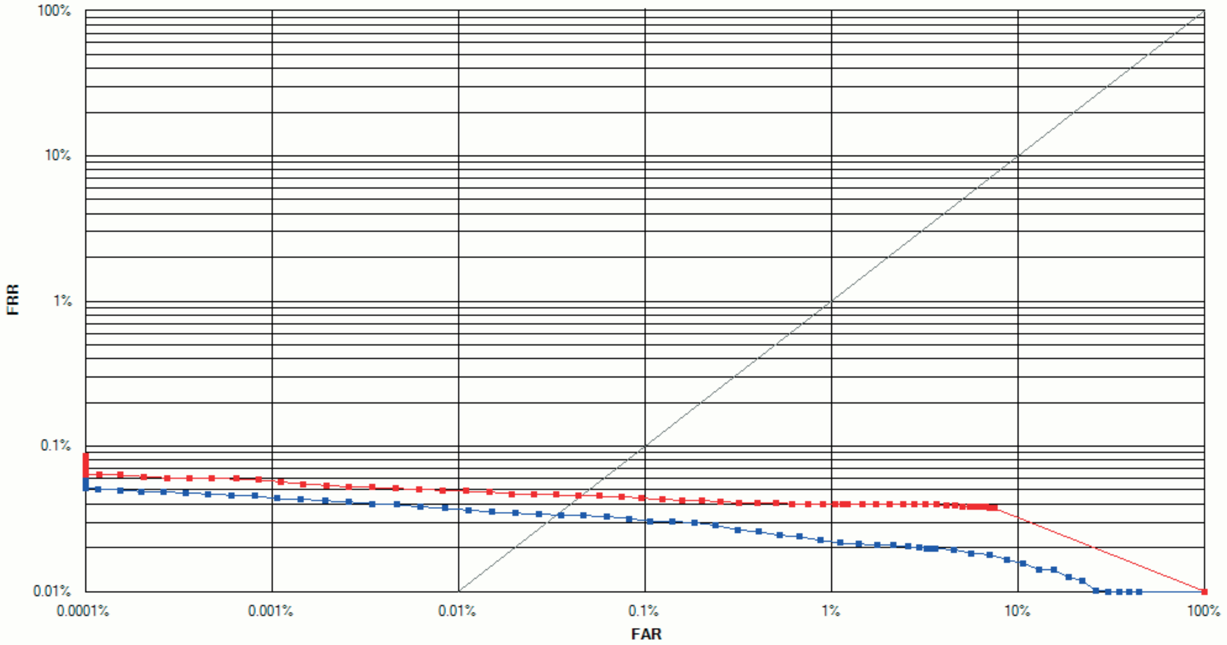
Receiver operation characteristic (**ROC**) curves are usually used to demonstrate the recognition quality of an algorithm. ROC curves show the dependence of false rejection rate (**FRR**) on the false acceptance rate (**FAR**). The ROC charts are on the next page.



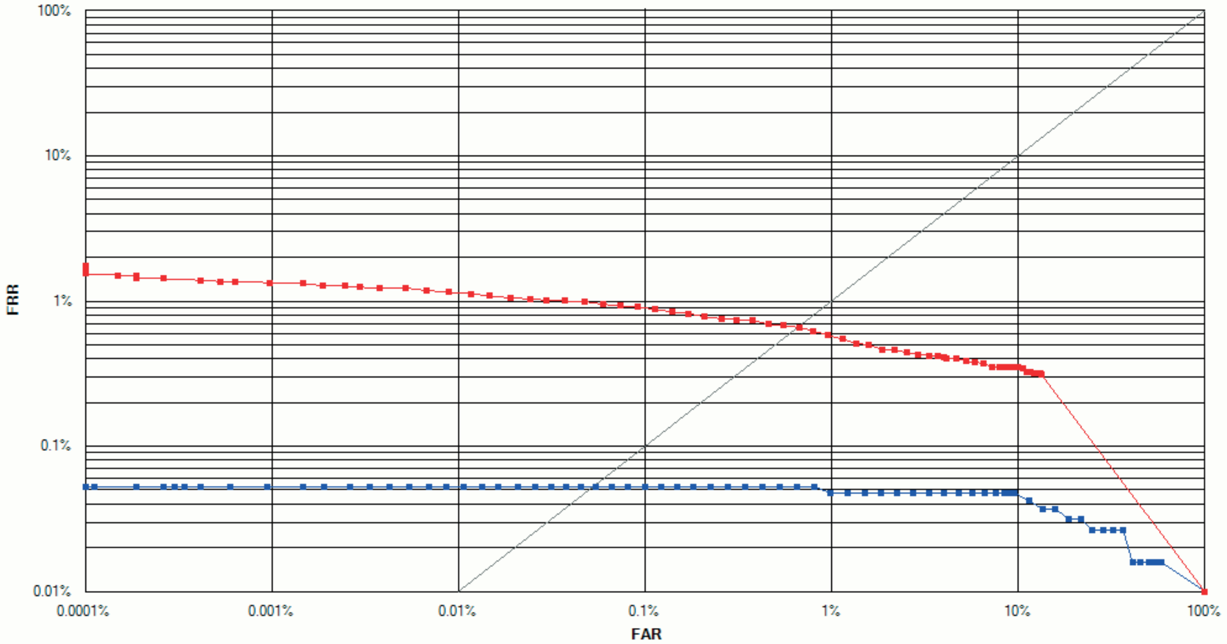
VeriEye 10.0 SDK matching engine with iris templates from ICE2005 Exp1 (Right Iris) database (captured in near-infrared spectrum):
🟥 **Maximized matching speed scenario**
🟦 **Maximized matching accuracy scenario**



VeriEye 10.0 SDK matching engine with iris templates from ND-IRIS-0405 database, University of Notre Dame (captured in near-infrared spectrum):
🟥 **Maximized matching speed scenario**
🟦 **Maximized matching accuracy scenario**



VeriEye 10.0 SDK matching engine with iris templates from IRISDB 1600 database, University of Bath (captured in near-infrared spectrum):
🟥 **Maximized matching speed scenario**
🟦 **Maximized matching accuracy scenario**



VeriEye 10.0 SDK matching engine with iris templates from MobileIrisV10 DB, Warsaw University of Technology (captured in visible light spectrum):
🟥 **Maximized matching speed scenario**
🟦 **Maximized matching accuracy scenario**

VeriEye Demo, Trial SDK and Related Products

VeriEye **algorithm demo** application and VeriEye **30-day SDK Trial** are available for downloading at ***www.neurotechnology.com/download.html.***

These products are related to VeriEye SDK:

- **MegaMatcher SDK** – intended for development of AFIS or multi-biometric iris, fingerprint, face and voice identification products. See “MegaMatcher SDK” brochure for more information.
- **MegaMatcher On Card SDK** – a product for iris, fingerprint and face matching on smart cards. See “MegaMatcher On Card“ brochure for more information.
- **NCheck Bio Attendance** – an end-user employee attendance management application designed as ready-to-use time and attendance system with biometric iris, face and fingerprint identification; VeriEye iris recognition algorithm is used in the application for checking person identity. See “NCheck Bio Attendance” brochure for more information.

Licensing VeriEye SDK

Product Development

An integrator should obtain either a VeriEye 10.0 Standard SDK (EUR 339) or VeriEye 10.0 Extended SDK (EUR 859) to develop a end-user product based on VeriEye technology. The SDK needs to be purchased just once and may be used for all projects and by all the developers within the integrator's company.

See the "Contents of VeriEye Standard SDK and Extended SDK" chapter (page 4) for the list of component licenses included with VeriEye 10.0 Standard and VeriEye 10.0 Extended SDK.

Integrators can obtain additional component licenses if more component licenses are required for the development process.

Product Deployment

To deploy their developed products, an integrator need obtain licenses of components for every **computer or device**, where component will be installed together with integrator's product. See Product Advisor to find out what specific components will be needed for the deployment of your system. Integrators can purchase additional VeriEye component licenses if required at anytime.

License activation options

The components are copy-protected. The following license activation options are available:

- **Serial numbers** are used to activate licenses for particular VeriEye components on particular computer or device. The activation is done via the Internet or by email. After activation the network connection is not required for single computer license usage.
  Notes:
  1. Activation by serial number is **not suitable for iOS and ARM-Linux** platforms, except BeagleBone Black and Raspberry Pi 3 devices..
  2. Activation by serial number is **not suitable for virtual environments**.
- **Internet activation.** A special **license file** is stored on a computer or a mobile or embedded device; the license file allows to run particular VeriEye components on that computer or device after **checking** the license over the Internet. **Internet connection** should be available periodically for a short amount of time. A single computer license can be **transferred** to another computer or device by moving the license file there and waiting until the previous activation expires.
- **Volume License Manager.** Licenses may be stored in a volume license manager **dongle**. License activation using volume license manager may be performed without connection to the Internet and is suitable for virtual environments. Volume license manager is **used on site by integrators or end users** to manage licenses for VeriEye components in the following ways:
  1. **Activating single computer licenses** – An installation license for a VeriEye component will be activated for use on a particular computer. The number of available licenses in the license manager will be decreased by the number of activated licenses. .
  2. **Managing single computer licenses via a LAN or the Internet** – The license manager allows the management of installation licenses for VeriEye components across multiple computers or mobile/embedded devices in a LAN or over the Internet. The number of managed licenses is limited by the number of licenses in the license manager. No license activation is required and the license quantity is not decreased. Once issued, the license is assigned to a specific computer or device on the network.
  3. **Using license manager as a dongle** – A volume license manager containing at least one license for a VeriEye component may be used as a dongle, allowing the VeriEye component to run on the particular computer where the dongle is attached.

Licenses Validity

All SDK and component licenses are perpetual and do not have expiration. There are no annual fee or any other fees except license purchasing fee. It is possible to move licenses from one computer or device to another. Neurotechnology provides a way to renew the license if the computer undergoes changes due to technical maintenance.

Licensing Agreement

The Licensing Agreement (*http://neurotechnology.com/mm_100_sla.html*) contains all licensing terms and conditions.

Note that you unambiguously accept this agreement by placing an order using Neurotechnology online ordering service or by email or other means of communications. Please read the agreement before making an order.

Other licensing options

- **VAR License.** The above described licensing model is intended for **end-user** product developers. Integrators who want to develop and sell a VeriEye-based development tool (with API, programming possibilities, programming samples, etc.), must obtain permission from Neurotechnology and **sign** a special VAR agreement. For more information please contact us.
- **Enterprise License.** The VeriEye enterprise license allows an **unlimited use** of VeriEye components in end-user products for a specific territory, market segment or project. Specific restrictions would be included in the licensing agreement. The enterprise license price depends on the application size and the number of potential users of the application within the designated territory, market segment or project. For more information please contact us.

Prices for VeriEye Products

- The prices are **effective June 21, 2017**. The prices may change in the future, so please **download and review the latest version** of the brochure before making an order.
- Quantity discounts do not accumulate over time.
- Prices do not include local import duties or taxes.
- Product shipping costs depend on delivery country.
- Customers with Solution Partner status are eligible for product discounts.

VeriEye SDK	
VeriEye 10.0 Standard SDK	€ 339.00
VeriEye 10.0 Standard SDK	€ 859.00

Iris Client component concurrent licenses	
Price per license	€ 650.00

Iris components for PCs (prices per single computer license)

Quantity	Iris Extractor	Iris Client [(1)]	Iris Matcher	Iris BSS [(1)]
1 - 9	€ 30.00	€ 38.00	€ 38.00	€ 10.00
10 - 19	€ 22.00	€ 28.00	€ 28.00	€ 7.50
20 - 49	€ 19.00	€ 25.00	€ 25.00	€ 6.50
50 - 99	€ 17.00	€ 22.00	€ 22.00	€ 5.50
100 - 199	€ 15.00	€ 19.00	€ 19.00	€ 5.00
200 - 499	€ 13.00	€ 17.00	€ 17.00	€ 4.50
500 - 999	€ 12.00	€ 15.00	€ 15.00	€ 4.00
1000 - 1999	€ 11.00	€ 13.00	€ 13.00	€ 3.50
2000 - 3999	€ 10.00	€ 12.00	€ 12.00	€ 3.10
4000 - 7999	€ 9.00	€ 11.00	€ 11.00	€ 2.80
8000 and more	Please contact us for more information			

Embedded iris components for Android devices (prices per single computer license)

Quantity	Embedded Iris Extractor	Embedded Iris Client [(1)]	Embedded Iris Matcher
1-9	€ 20.00	€ 25.00	€ 25.00
10-19	€ 15.00	€ 18.00	€ 18.00
20-49	€ 13.00	€ 16.00	€ 16.00
50-99	€ 11.00	€ 14.00	€ 14.00
100-199	€ 10.00	€ 12.50	€ 12.50
200-499	€ 9.00	€ 11.00	€ 11.00
500-999	€ 8.00	€ 10.00	€ 10.00
1000-1999	€ 7.00	€ 9.00	€ 9.00
2000-3999	€ 6.40	€ 8.00	€ 8.00
4000-7999	€ 5.80	€ 7.00	€ 7.00
8000 and more	Please contact us for more information		

License management	
Volume license manager	€ 16.00

(1) These components are not available for VeriEye Standard SDK customers.

VeriEye products can be ordered:

- online, at ***www.neurotechnology.com/cgi-bin/order.cgi***
- via a local Neurotechnology distributor; the list of distributors is available at ***www.neurotechnology.com/distributors.html***