NEUROtechnology

Biometrical and Artificial Intelligence Technologies

SentiVeillance 6.0 Algorithm Demo

Users's guide

Introduction

About SentiVeillance SDK

SentiVeillance is a software development kit (SDK) for biometric face identification, pedestrian detection, object recognition and tracking using live video streams from high-resolution digital surveillance cameras. SentiVeillance not only identifies faces in a crowd, it detects both people and objects while they are moving through the video frame, and it can detect if a person or object is entering or leaving a predefined boundary area, triggering an alert to the system monitor. This smart monitoring technology can be used for a variety of applications, including retail and commercial areas, entrance monitoring and counting, automated time-attendance systems, transportation, law enforcement and security systems.

SentiVeillance includes a robust, dynamic face model that can adapt to visual appearance changes as subjects move across the scene. Based on the biometric facial recognition engine from Neurotechnology, SentiVeillance provides fast face detection, improved face feature determination, gender determination and emotion detection, including detection of smiles, closed-eyes, open-mouths, clear and dark glasses. SentiVeillance matches face images against internal databases, such as authorized personnel or criminal watch lists, and the associated application can then lock or unlock a door, put a record into a time/attendance system or immediately trigger a report or alert for recognized (or un-recognized) faces.

In situations when video surveillance is monitoring a very large area or when cameras are in a position that does not allow some faces to be visible, SentiVeillance uses moving object detection and classification to identify objects vs. people in the scene. In addition to person identification, it can be used to identify and monitor objects such as cars, buses and trucks in the camera's view.

With smart monitoring technology, if a moving person or object has crossed a defined border or entered/left a specific monitored area or region, SentiVeillance can send an immediate alert notification and trigger appropriate action if a restricted or sensitive area has been compromised. This ability to detect and track both moving people and objects allows SentiVeillance to be used in a much wider range of applications, such as event security or police applications like the automatic detection of cars passing through red lights or stop signs.

SentiVeillance continues tracking of subjects even when their faces briefly disappear from the frame or when they are partially blocked by other objects or even other faces (a common problem while tracking multiple faces). The face recognition algorithm enables detection of faces with up to 45 degrees out-of-plane rotation in yaw angle and uses motion prediction models to re-localize faces that have undergone full occlusion, such as when a subject has been fully obstructed by a wall and emerged on the other side. The dynamic face model allows the system to efficiently and reliably track faces in front of complex backgrounds and ensures that subjects can be localized in all video frames, even under strenuous conditions.

SentiVeillance enables one computer to process images from multiple cameras and also provides connections between SentiVeillance units deployed on different computers, synchronizing the databases so that the system works as a whole within the logical network. It enables real-time face detection, extraction and matching by providing embedded parallelization of all VeriLook functions for improved performance on multi-core, multi-processor systems.

SentiVeillance can be used in two different modes: automatic and manual. When automatic mode is used SentiVeillance is working with the default subjects tracking algorithm. But when the manual mode is selected SenitVeillance allows users to perform manual camera's calibration and to define custom objects classes. For instance user can define such classes as "vehicle", "subject" or "group". The object class is defined by the limits of an area and speed. These different object classes are tracked separately in video footage or live video. Also, it is possible to assign a separate event for different object classes in the surveillance zone. For instance, only when a vehicle enters the zone, the event handler is triggered.

This guide is dedicated to face tracking and recognition in digital surveillance cameras. The guide covers such topics as adding subjects to watch list (aka enrollment), face tracking and recognition, camera calibration, GUI of the demo program and some notes on parameters and other settings.

The demo program can accept video streams from different sources like digital surveillance cameras, IP cameras, video files, USB cameras and etc.

For more information about SentiVeillance SDK, download a free trial version of SDK from Neurotechnology website and read documentation in "Documentation" folder ("SentiVeillance SDK.pdf"). In this document you'll also find "Samples" section which explains how to use the same demo application with IP cameras.

System requirements

Recommended requirements for the system:

- At least Intel Core 2 CPU with 4 cores running at 2.66 GHz or equivalent multi-core processor from other manufacturer
- At least 1 GB of free RAM
- Microsoft Windows specific requirements:
  - Microsoft Vista / 7 / 8 / 10 / Server 2008 / Server 2008 R2 / Server 2012, 64-bit
  - Microsoft .NET framework 3.5 or later (for .NET components)
  - Microsoft DirectX 9.0 or later
  - 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

Note: Windows 8 UI (Windows Store) applications are not supported.

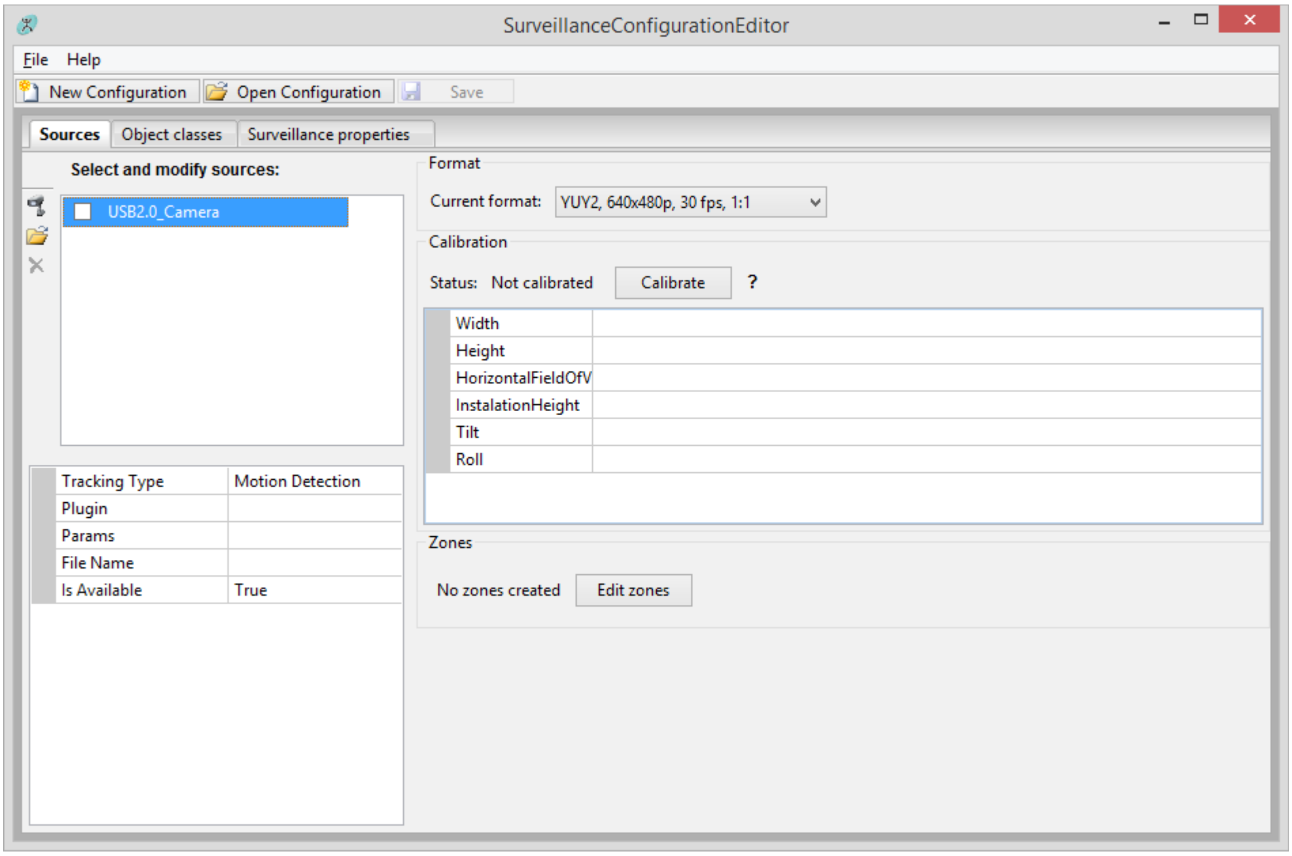
- Linux specific requirements:
  - Ubuntu 16.04 // note: possibly also 14.04, will confirm later
  - GStreamer with gst-vaapi plugins installed for hardware accelerated video decoding
  - libgudev
  - wxWidgets 3.0.0 or newer libs and dev packages (to build and run SDK samples and applications based on them)
  - Sun Java 1.6 SDK or later (for application development with Java)

Using Algorithm Demo application

Configuration Editor

Before you start using SentiVeillance sample application, it is required to calibrate camera. Camera calibration and remapping allows to fix distortion, thus avoiding errors when trying to determine the object's relative position and distance from a camera. Furthermore, with calibration you may determine the relation between camera's natural units (or pixels) and the real world units (e.g. meters). Calibration is performed by entering such parameters as horizontal field of view, camera's installation height, tilt and roll. Normally, camera calibration is performed only once after it was installed.

SentiVeillance algorithm demo package on the root directory includes the special calibration tool called *SentiVeillanceConfigurationEditorDemo.exe*. When this tool is started this window appears:



Current format: YUY2, 640x480p, 30 fps, 1:1

Status: Not calibrated

Width	
Height	
HorizontalFieldOfV	
InstalationHeight	
Tilt	
Roll	

No zones created

Tracking Type	Motion Detection
Plugin	
Params	
File Name	
Is Available	True

On the left pane Sources for video can be selected or existing sources modified. On the right pane camera calibration properties are displayed. Also *Object* classes and Surveillance properties tabs are used respectively for defining new object classes for tracking and for changing SentiVeillance properties.

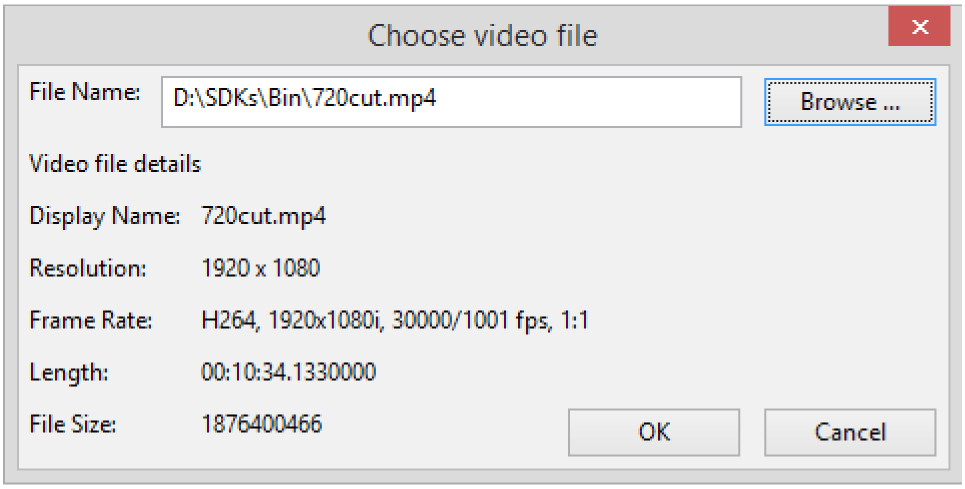
Let's review these panes in more detail.

Sources

Sources tab is a starting point when an application was launched. In this tab all available video sources for surveillance are listed down. As you can see in the screenshot above, an USB camera has been detected already. We will add two more sources - an IP camera (with a live video) and video file (with a recorded video).

Adding video file as a surveillance source

Press Add from video file... button on the very left corner of the main window and in pop-up window select video file (or enter full path to it).



File Name: D:\SDKs\Bin\720cut.mp4

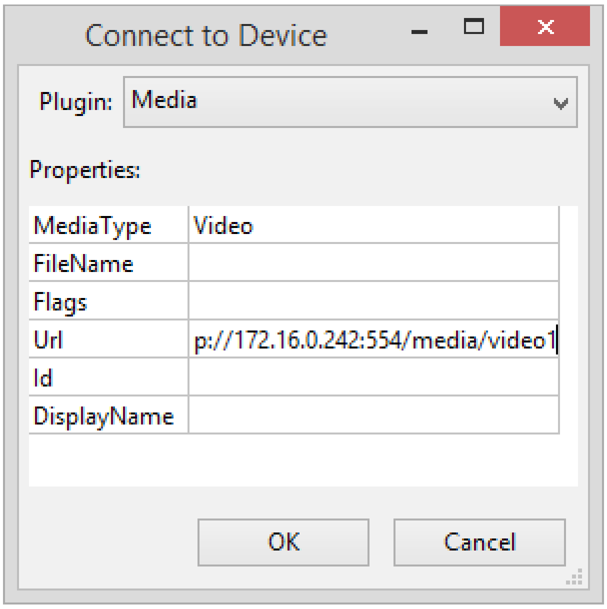
Video file details

Display Name:	720cut.mp4
Resolution:	1920 x 1080
Frame Rate:	H264, 1920x1080i, 30000/1001 fps, 1:1
Length:	00:10:34.1330000
File Size:	1876400466

Video file will be added into Sources pane and further camera calibration profile or zone will be assigned to this source.

Adding IP camera as a video source

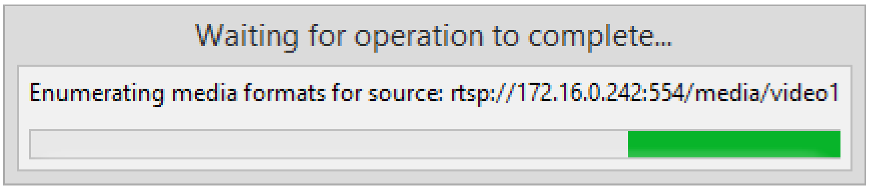
When using SentiVeillance application, it is normal practice to use live video streaming commonly from an IP camera. IP camera or other type of supported media device is added by choosing *Connect...* on the very left corner of the main window.



Plugin: Media

Properties:

MediaType	Video
FileName	
Flags	
Url	p://172.16.0.242:554/media/video1
Id	
DisplayName	



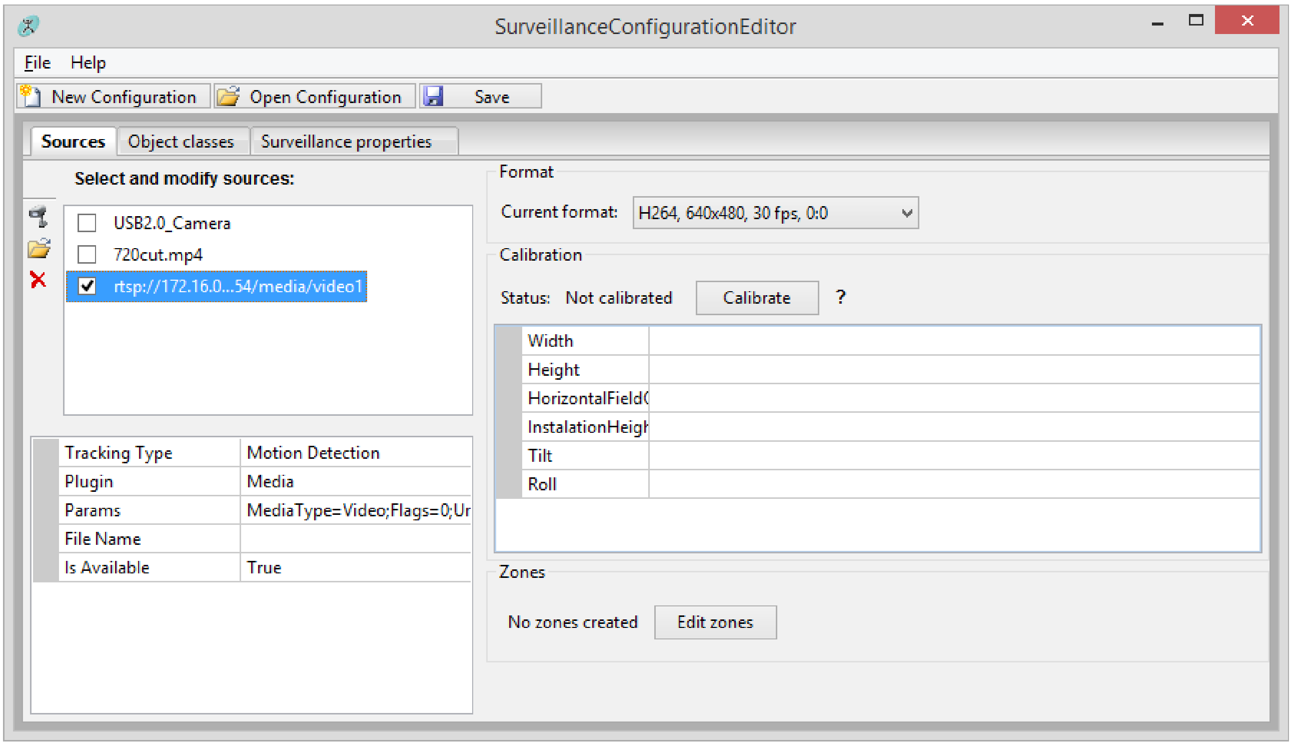
Waiting for operation to complete...

Enumerating media formats for source: rtsp://172.16.0.242:554/media/video1

User should enter camera's IP address, username, password and other properties if required.

Camera calibration

When USB camera was detected and video source and IP camera added, we will see such window:



Select and modify sources:

□ USB2.0_Camera
□ 720cut.mp4
☑ rtsp://172.16.0...54/media/video1

Tracking Type	Motion Detection
Plugin	Media
Params	MediaType=Video;Flags=0;Ur
File Name	
Is Available	True

Format

Current format: H264, 640x480, 30 fps, 0:0

Calibration

Status: Not calibrated

Width	
Height	
HorizontalField(	
InstalationHeigh	
Tilt	
Roll	

Zones

No zones created

Let's choose IP camera and calibrate it. **Note**: calibration is optional, but it is useful to calibrate camera. Without calibration the system will be able to distinguish pedestrians from other objects. With calibration it is possible to create and use custom classes.

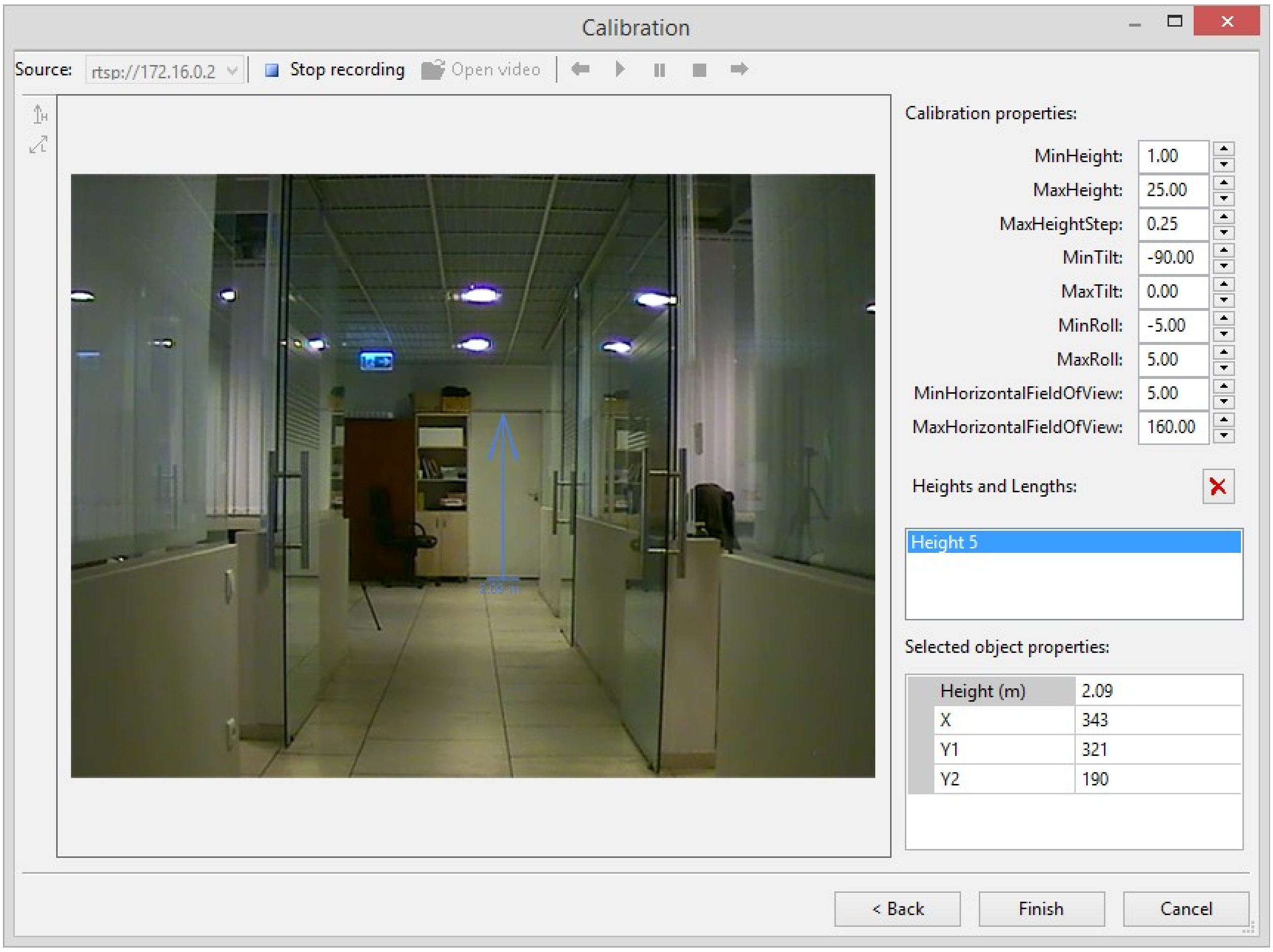
Press *Calibrate* button and in other window select calibration mode: automatic or manual.

Note: camera calibration is required only when user-defined classes are used. If you use only pedestrians, objects and faces tracking, you can skip camera calibration.

Automatic calibration

Automatic calibration will let you to determine camera parameters automatically by entering heights and lengths of the objects in video. Heights (of pedestrians, traffic lights, doors, or other objects touching the ground) should be provided by clicking the ground, top of the object and entering the actual measurement. Lengths/widths (of cars, roads, road markings, or other objects with both length ends on the ground), should be provided by clicking two points on the ground and entering the actual measurements.

This window shows live video from an IP camera calibration:



Calibration properties:

MinHeight:	1.00
MaxHeight:	25.00
MaxHeightStep:	0.25
MinTilt:	-90.00
MaxTilt:	0.00
MinRoll:	-5.00
MaxRoll:	5.00
MinHorizontalFieldOfView:	5.00
MaxHorizontalFieldOfView:	160.00

Heights and Lengths:

Height 5

Selected object properties:

Height (m)	2.09
X	343
Y1	321
Y2	190

When calibration is started, video is recorded into a file. The top bar contains video controls which can be used to start, stop, play or open a recorded video, and to step frames backward and forward besides.

The screenshot above has already a measured object - the doors. Blue arrow, which was selected from the left pane, was used to mark both ends of the doors and to enter the height. The right bottom pane is used for entering the exact height/length or to edit X/Y coordinates.

Also user can edit calibration properties on the right[1]:

- *MinHeight/MaxHeight* - minimal and maximal values for the height of camera's installation.
- *MaxHeightStep* - maximal height increase in meters.
- *MinTilt/MaxTilt* - minimal and maximal values in degree of camera's tilt.
- *MinRoll/MaxRoll* - minimal and maximal values in degree of camera's roll.
- *MinHorizontalFieldOfView/MaxHorizontalFieldOfView* - minimal and maximal values in degree of camera's horizontal field of view.

[1] These calibration properties shouldn't be accurate. Although these properties are used, the calibration algorithm calculates the accurate measurements basically using the precise sizes of objects. So if you do not know the precise height of camera's installation height, you can compare it with your height and, for instance, specify min and max height values 1.8 and 5 meters, respectively.

Whereas the screenshot above shows only one measured object, it is recommended to measure more objects in different distances from each other in distinct parts of a frame. It is a good idea to measure, let's say, 5 objects in a real scene and enter precise height/legth measurements to the Configuration editor.

Manual calibration

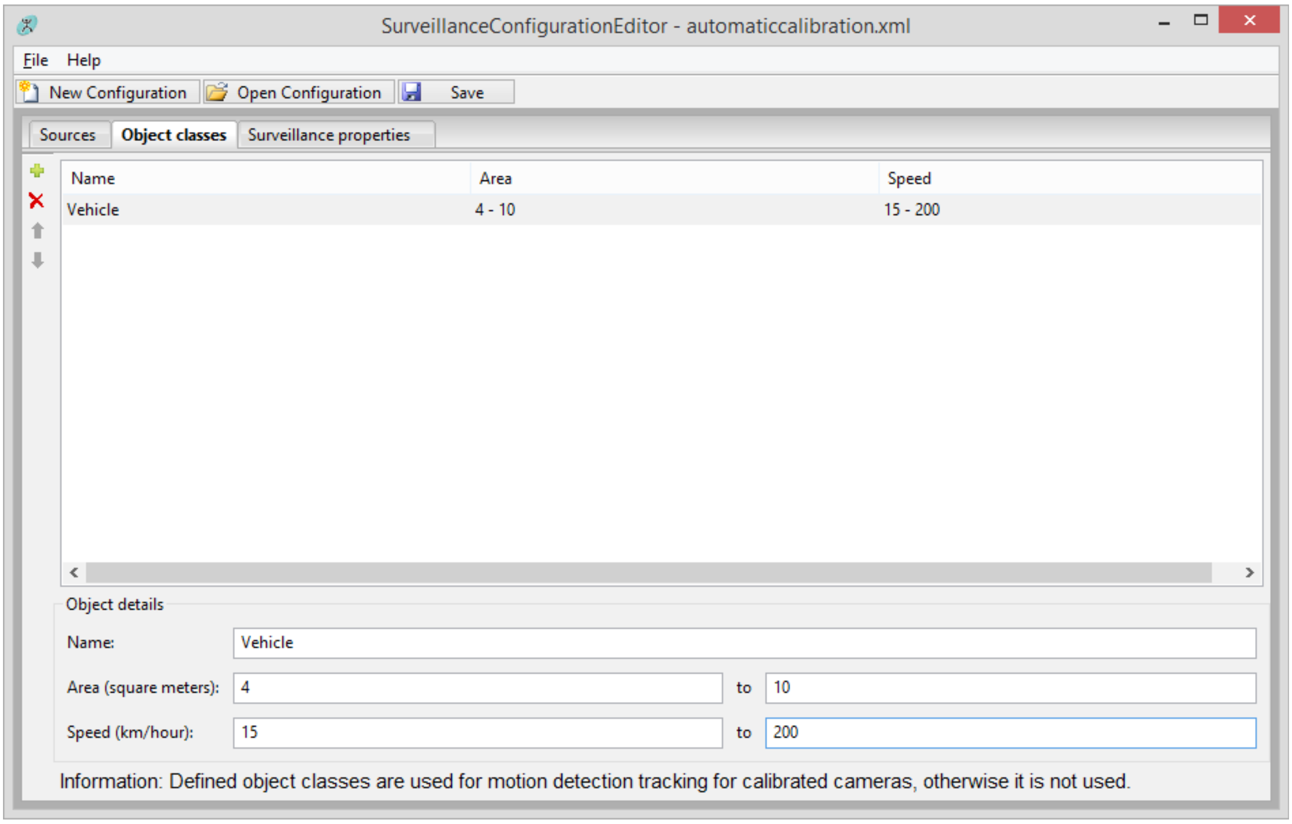
For the most users automatic camera calibration should be used and SentiVeillance algorithm will determine these values based on the precise size of objects in a frame. Manual calibration is used when precise measurements of camera installation height in meters, tilt relative to the ground in degrees (0 - camera looking down, 90 - camera looking horizontally), horizontal field of view in degrees and roll to the left or right in degrees (usually it is close to 0) are known.

For both manual and automatic calibration data is saved by pressing *Finish* button.

Object classes

Users can define various objects (such as vehicles, subjects, group of people, etc.) which are different in size, shape or other characteristics. The SentiVeillance recognizes these objects and tracks them separately. Object classes are defined by specifying its area and speed limits. Note: object classes can be used for motion detection only for calibrated cameras.

The picture shows that the class *Vehicle* was created:



Name	Area	Speed
Vehicle	4 - 10	15 - 200

Object details

Name: Vehicle

Area (square meters): 4 to 10

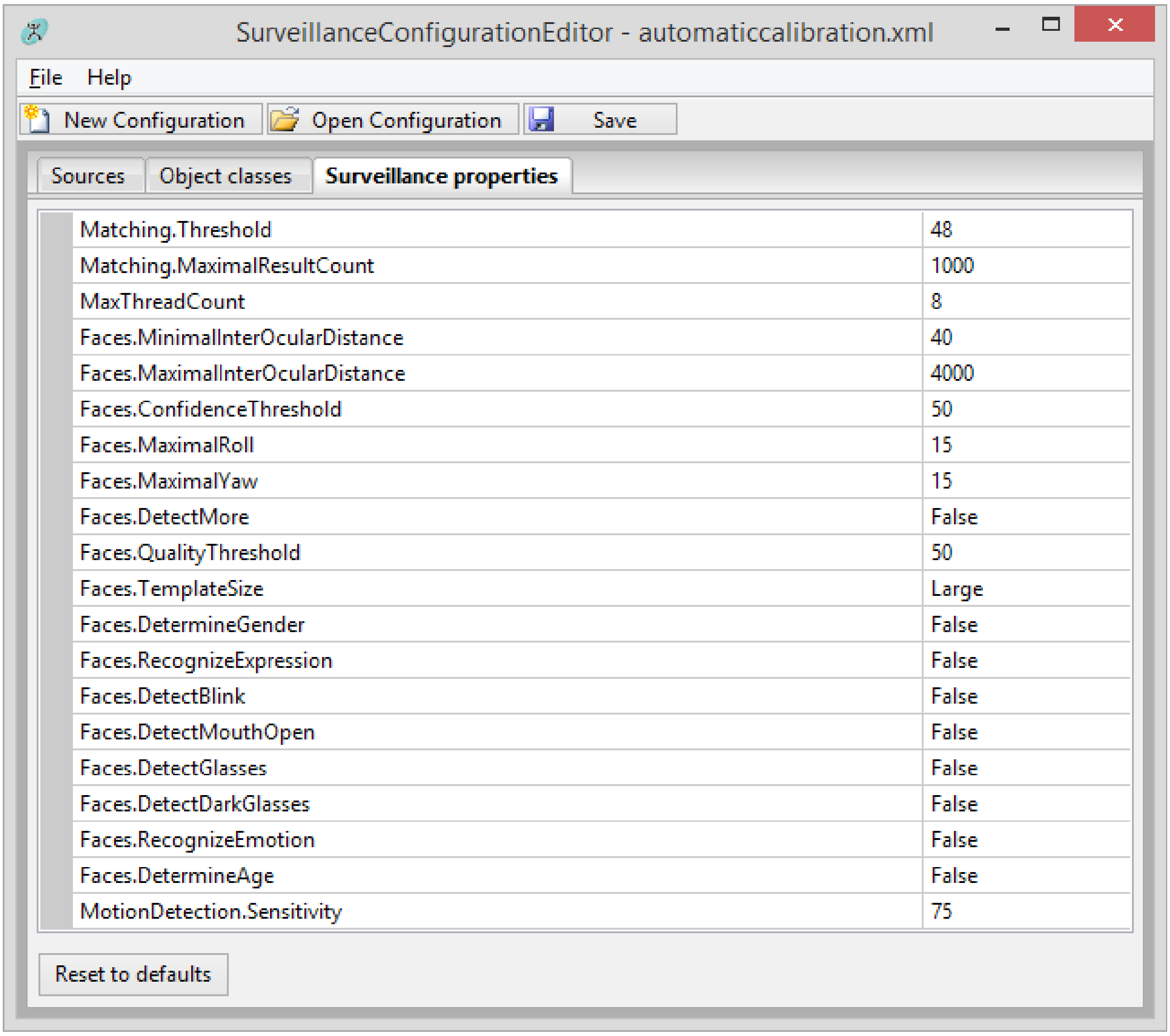
Speed (km/hour): 15 to 200

Information: Defined object classes are used for motion detection tracking for calibrated cameras, otherwise it is not used.

Only vehicles occupying 4 to 10 square meters and driving 15-200 kmph will be assigned to the class Vehicle. You should note that at the moment only speed and area characteristics are used. So if you want to define two similar classes like a bus and a truck, SentiVeillance would probably not work correctly, because these two objects are similar in area and speed.

Surveillance properties

These properties can be changed:



Matching.Threshold	48
Matching.MaximalResultCount	1000
MaxThreadCount	8
Faces.MinimalInterOcularDistance	40
Faces.MaximalInterOcularDistance	4000
Faces.ConfidenceThreshold	50
Faces.MaximalRoll	15
Faces.MaximalYaw	15
Faces.DetectMore	False
Faces.QualityThreshold	50
Faces.TemplateSize	Large
Faces.DetermineGender	False
Faces.RecognizeExpression	False
Faces.DetectBlink	False
Faces.DetectMouthOpen	False
Faces.DetectGlasses	False
Faces.DetectDarkGlasses	False
Faces.RecognizeEmotion	False
Faces.DetermineAge	False
MotionDetection.Sensitivity	75

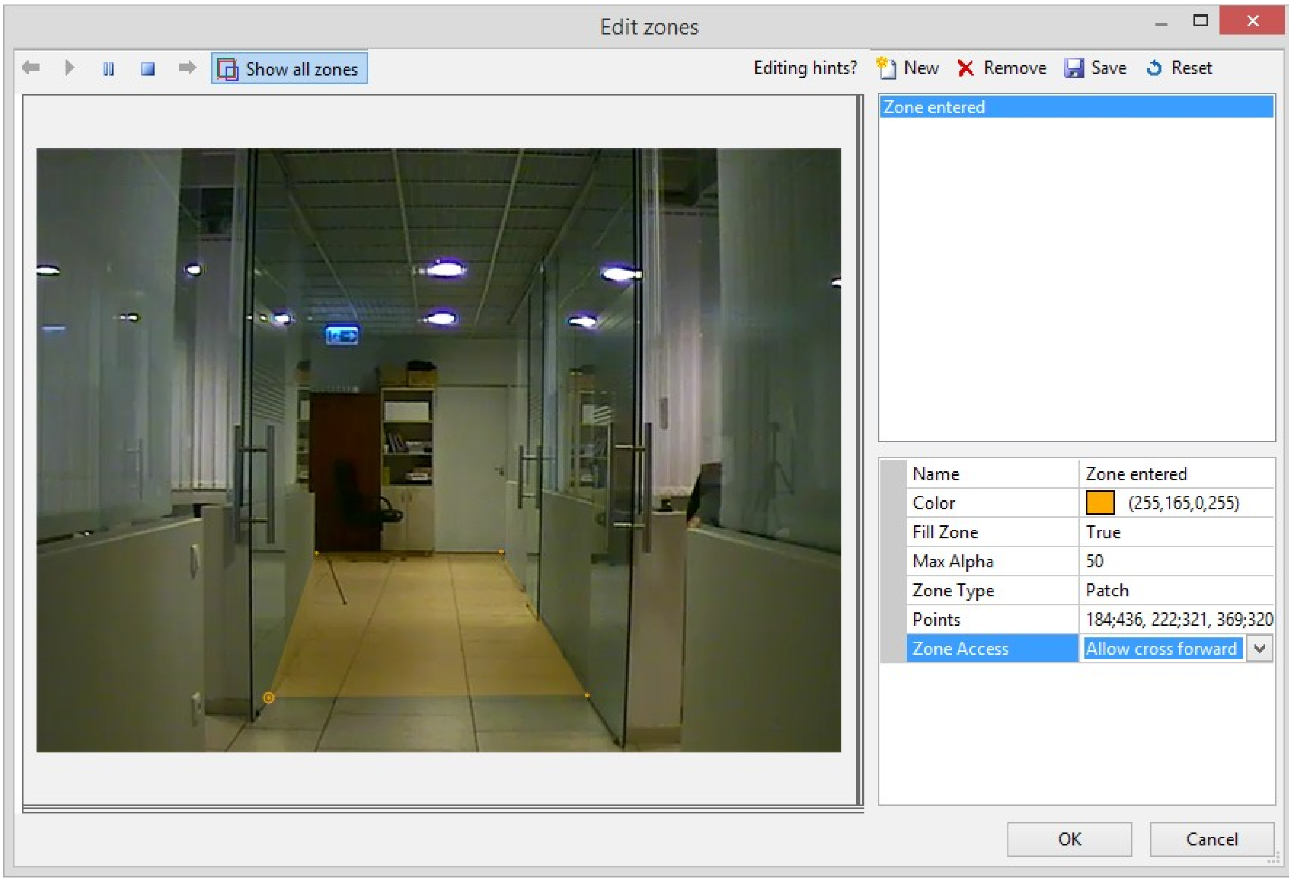
- *Matching.Threshold* - specifies matching threshold. Value should be more than 0.
- *Maximal result count* - sets the maximal number of matching results.
- *MaxThreadCount* - maximum threads count which can be used.
- *Faces.MinimalInterOcularDistance* - minimum distance between eyes.
- *Faces.MaxinalInterOcularDistance* - maximum distance between eyes.
- *Faces.MaximalRoll* - this value is used to define the maximum value in degrees of tilted face image which can be enrolled or identified.
- *Faces.MaximalYaw* - this value is used to define the maximum yaw value in degrees of face image which can be enrolled or identified.

- *Faces.QualityThreshold* - sets a quality threshold. If a face image's quality threshold is less than specified value, the face image will be rejected.
- *Faces.TemplateSize* - size of face templates. It is recommended to use large template size for better accuracy.
- *Faces.DetermineGender* - if true, gender will be determined.
- *Faces.RecognizeExpression* - if true, face expressions will be recognized.
- *Faces.DetectBlink* - if true, eyes blinking will be detected.
- *Faces.DetectMouthOpen* - if true, open mouth will be detected.
- *Faces.DetectGlasses* - if true, glasses on a face will be detected.
- *Faces.DetectDarkGlasses* - if true, dark glasses on a face will be detected.
- *Faces.RecognizeEmotion* - if true, face emotions will be recognized.
- *Faces.DetermineAge* - if true, face recognition algorithm tries to determine subject's age.
- *MotionDetection.Sensitivty* - motion detection sensitivity value.

Edit zones

SentiVeillance allows user to specify surveillance zone. A surveillance zone is a virtual polygon or a line, separating the camera's view. An event is an object crossing such line, or entering/leaving a polygon. A zone involved in an event is referred to as a relevant zone.

New zones are created and existed ones edit when video source is chosen and Edit zones button on the main window is pressed. Such window appears:



Name	Zone entered
Color	🟧 (255,165,0,255)
Fill Zone	True
Max Alpha	50
Zone Type	Patch
Points	184;436, 222;321, 369;320
Zone Access	Allow cross forward

The zone was drawn on a live video stream from an IP camera and is shown in an orange color. The new zone is created by selecting New on the right pane. Then you can draw a polygon (should select Patch as zone type) or a tripwire (should select Tripwire as zone type) on the left pane. This will be the zone. A polygon or a tripwire is drawn by adding or removing points. This can be done by mouse click or manually adding point coordinates.

User can change a zone name, color, type and zone access on the right pane. Also zone point can be manually added or edited by selecting Points.

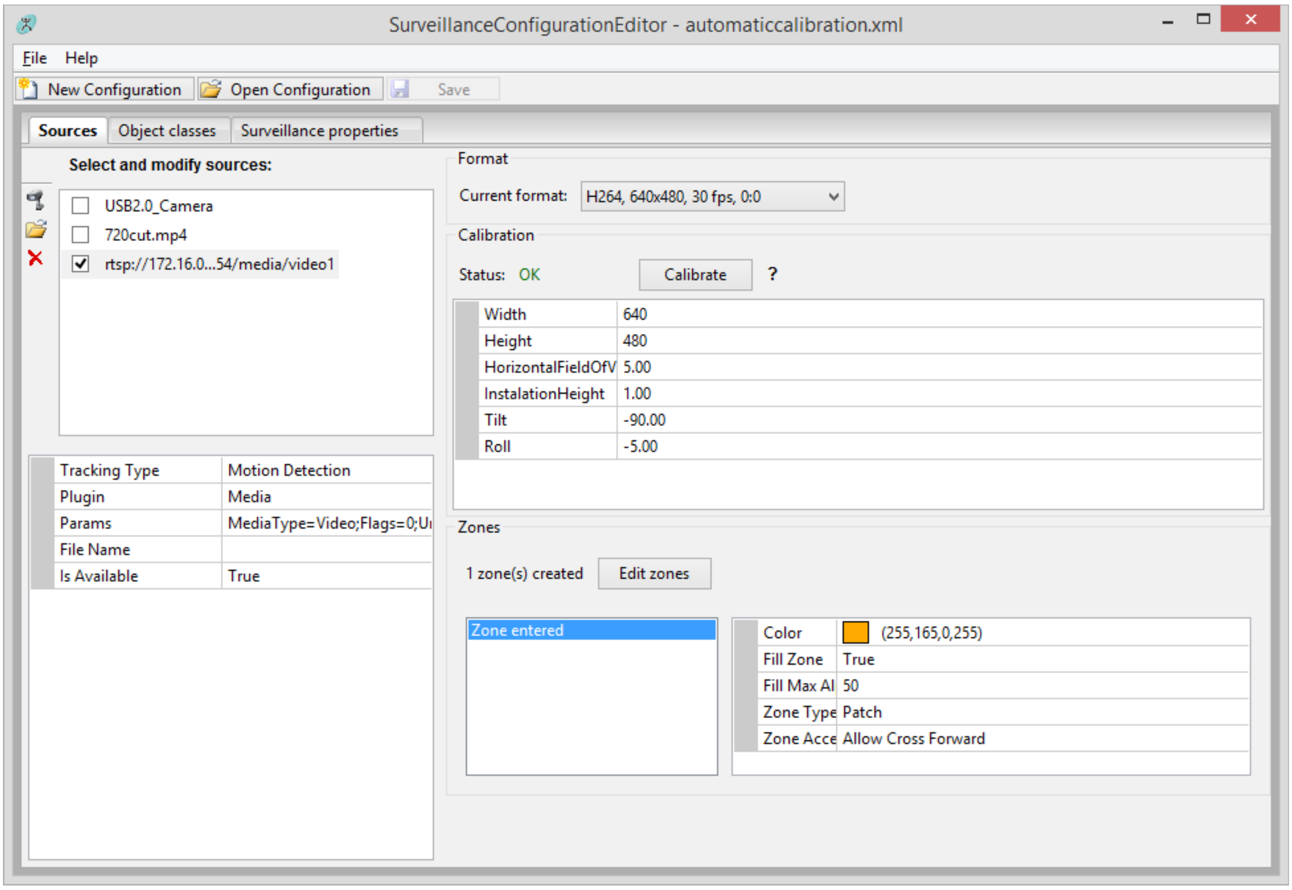
4 zone access types can be selected:

- *Any one* - any one can enter the zone. Tracked subjects will be displayed with a green rectangle.
- *No one* - no one can enter the zone. All tracked subjects in the zone will be displayed with a red rectangle.
- *Allow cross forward* - used for tripwire type zones and is used to display moving objects crossing zones in a wrong direction.
- *Allow cross backwards* - used for tripwire type zones and is used to display moving objects crossing zones in a wrong direction.

When all parameters are set, user should *Save* the zone and press OK button.

Saving configuration

After camera was calibrated, object classes and zones created, you should see the window like this:



Select and modify sources:

□ USB2.0_Camera
□ 720cut.mp4
☑ rtsp://172.16.0...54/media/video1

Tracking Type	Motion Detection
Plugin	Media
Params	MediaType=Video;Flags=0;U
File Name	
Is Available	True

Format

Current format: H264, 640x480, 30 fps, 0:0

Calibration

Status: OK

Width	640
Height	480
HorizontalFieldOfV	5.00
InstalationHeight	1.00
Tilt	-90.00
Roll	-5.00

Zones

1 zone(s) created

Zone entered

Color	🟧 (255,165,0,255)
Fill Zone	True
Fill Max Al	50
Zone Type	Patch
Zone Acce	Allow Cross Forward

It displays all the settings. Also, on the right pane you can change media settings and tracking type for a selected video source. These settings are saved in an *xml* file which can be added manually or loaded to the Configuration editor afterwards.

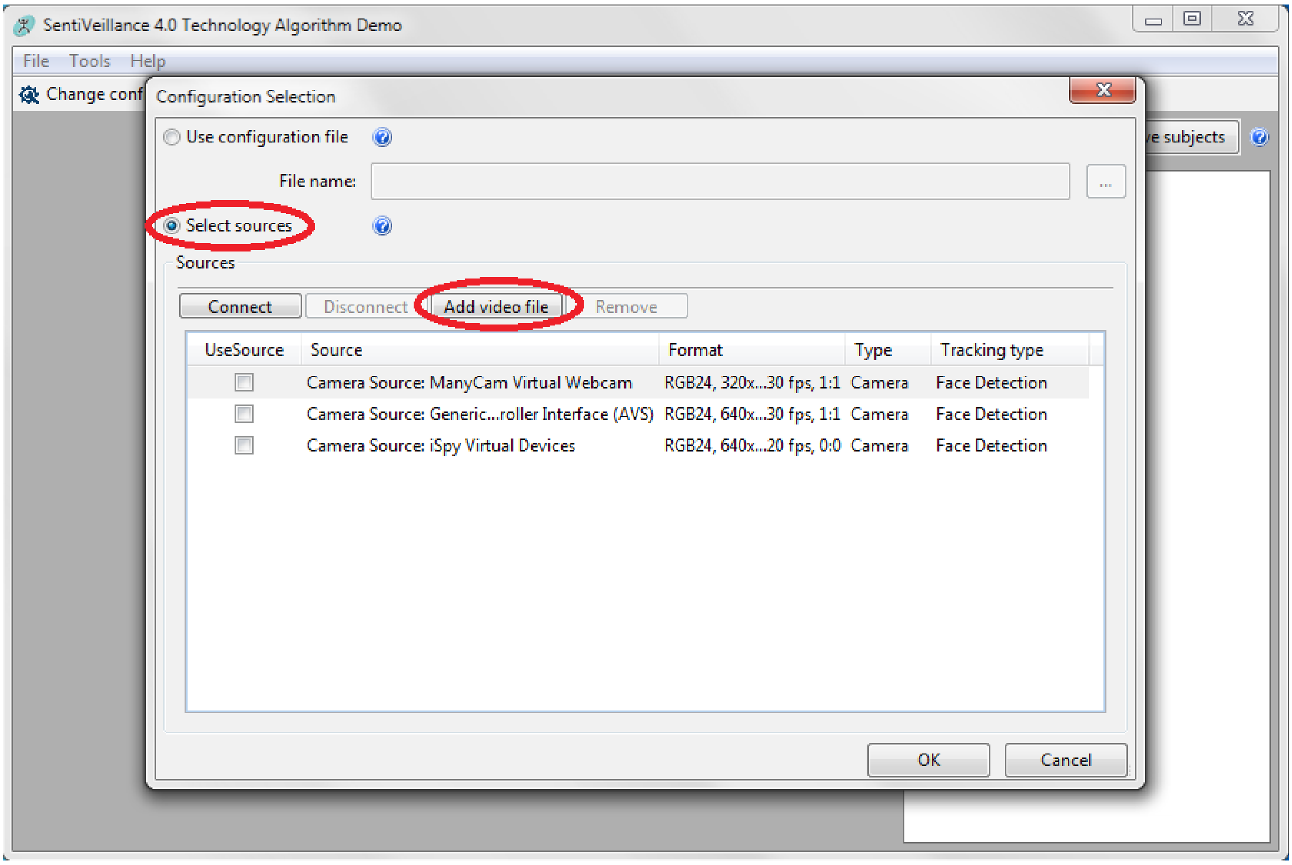
> *"Documentation" folder also includes step-by-step guide for Configuration editor (VideoConfiguration_StepByStep.pdf file).*

Demo Application

SentiVeillance demo application is used for object recognition and tracking, face identification, pedestrian detection using live video streams (cameras or video files). Using this application subjects/objects can be added to internal database and after that tracked.

Adding Video Files

When *SentiVeillanceAlgorithmDemo.exe* is launched the Configuration Selection will appear. To add a video file first make sure that *Select sources* is checked. Then click *Add video file*. After that *Choose video file* dialog will appear. Click *Browse...* and navigate to the video file that you want to add.



○ Use configuration file

File name:

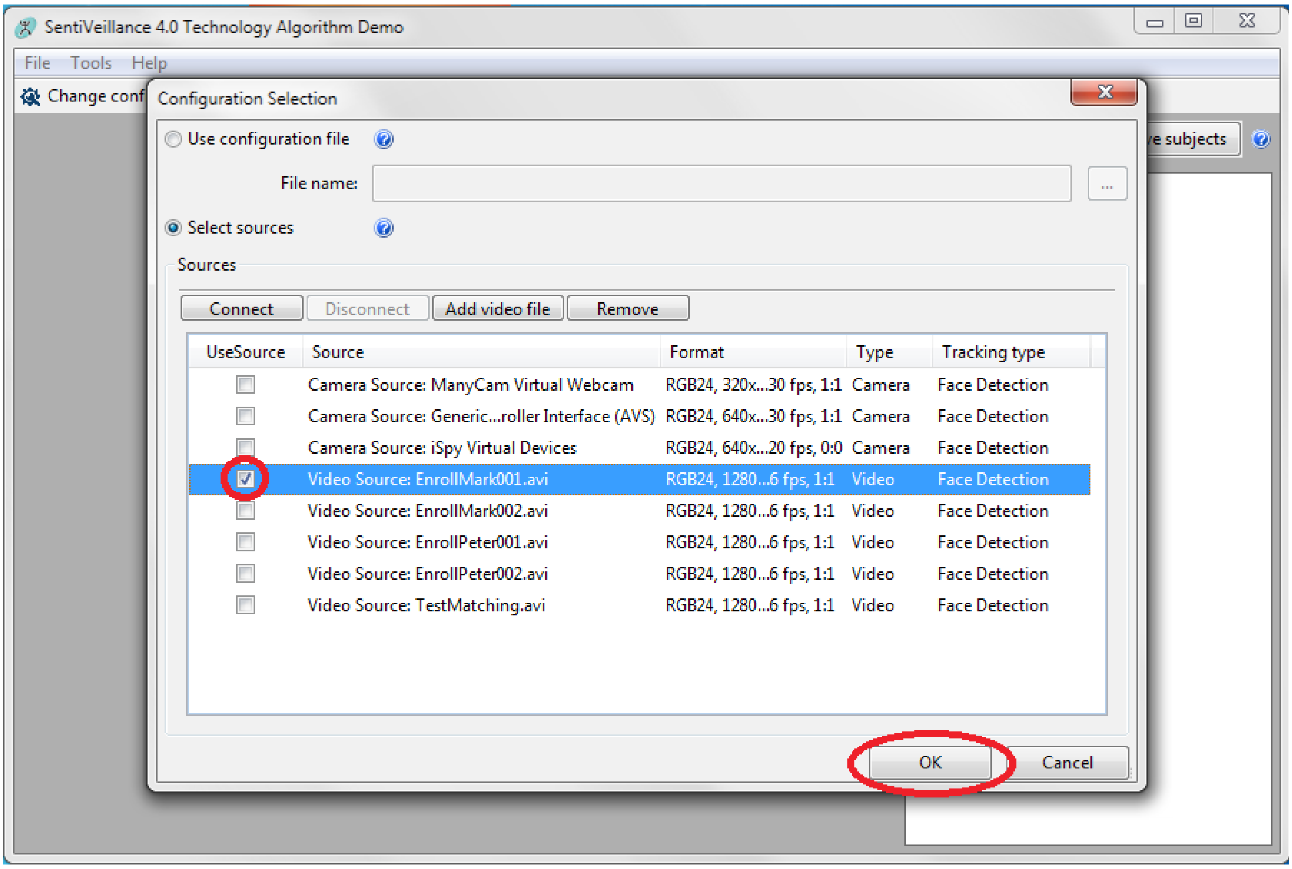
● Select sources

Sources

UseSource	Source	Format	Type	Tracking type
□	Camera Source: ManyCam Virtual Webcam	RGB24, 320x...30 fps, 1:1	Camera	Face Detection
□	Camera Source: Generic...roller Interface (AVS)	RGB24, 640x...30 fps, 1:1	Camera	Face Detection
□	Camera Source: iSpy Virtual Devices	RGB24, 640x...20 fps, 0:0	Camera	Face Detection

Tracking Faces and Adding Subjects to the Watch List

To begin tracking faces, select a video source from list of sources and click OK.



○ Use configuration file

File name:

● Select sources

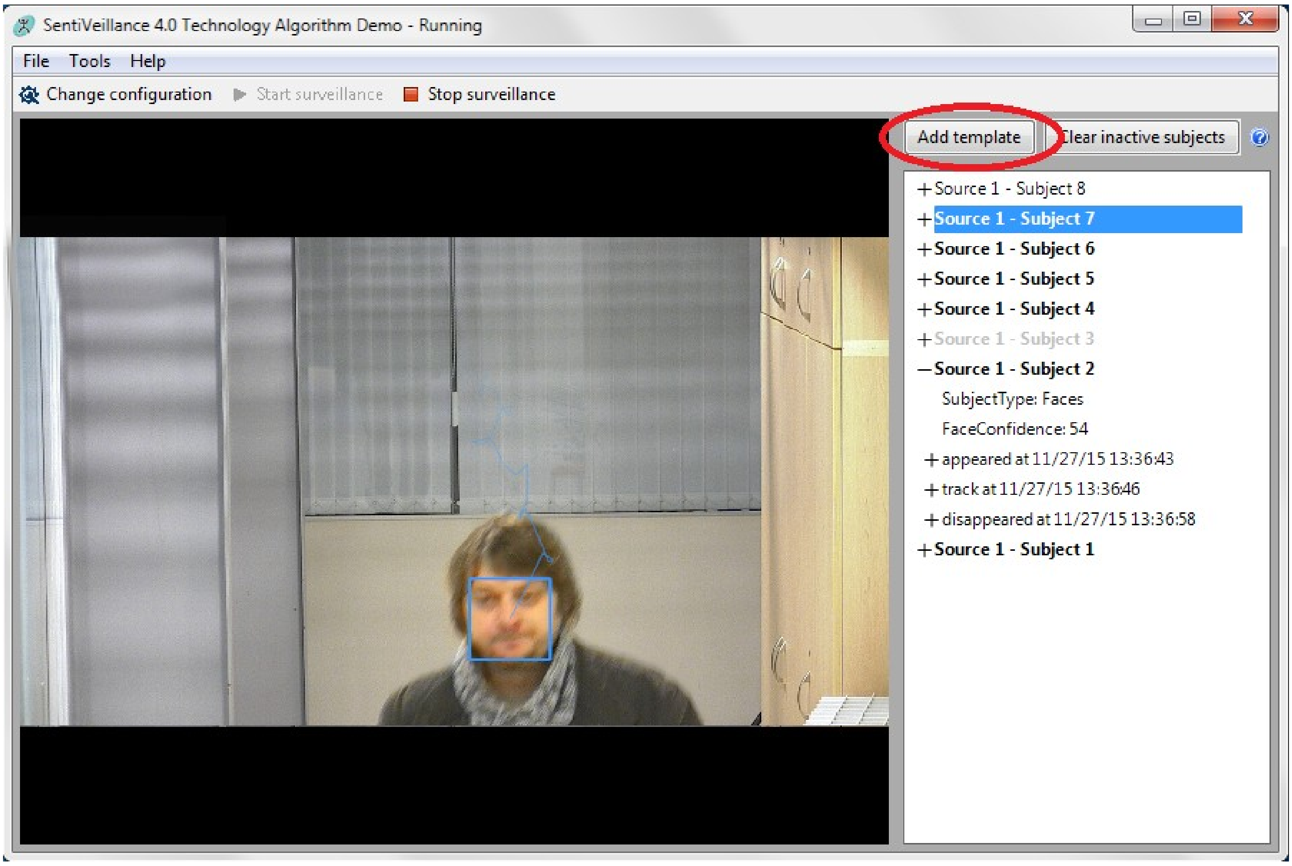
Sources

UseSource	Source	Format	Type	Tracking type
□	Camera Source: ManyCam Virtual Webcam	RGB24, 320x...30 fps, 1:1	Camera	Face Detection
□	Camera Source: Generic...roller Interface (AVS)	RGB24, 640x...30 fps, 1:1	Camera	Face Detection
□	Camera Source: iSpy Virtual Devices	RGB24, 640x...20 fps, 0:0	Camera	Face Detection
☑	Video Source: EnrollMark001.avi	RGB24, 1280...6 fps, 1:1	Video	Face Detection
□	Video Source: EnrollMark002.avi	RGB24, 1280...6 fps, 1:1	Video	Face Detection
□	Video Source: EnrollPeter001.avi	RGB24, 1280...6 fps, 1:1	Video	Face Detection
□	Video Source: EnrollPeter002.avi	RGB24, 1280...6 fps, 1:1	Video	Face Detection
□	Video Source: TestMatching.avi	RGB24, 1280...6 fps, 1:1	Video	Face Detection

Video playback should start immediately, otherwise click *Start Surveillance* button. Also you can stop video playback any time by pressing *Stop Surveillance*.

During video playback the demo program will start tracking faces of subjects that appear in the video. Events will appear on the right side of the demo program. Faces being tracked are represented as simple black text (e.g. Source1 - Subject9). Demo program supports up to 4 different video sources that can be processed at the same time, so *Source1* indicates that *Subject9* was tracked in the first video source. Tracked subjects are numbered in a sequence starting from 1.

The SentiVeillance algorithm will generate a template of the subject's face by selecting best frames from the entire tracking sequence after subject disappeared.

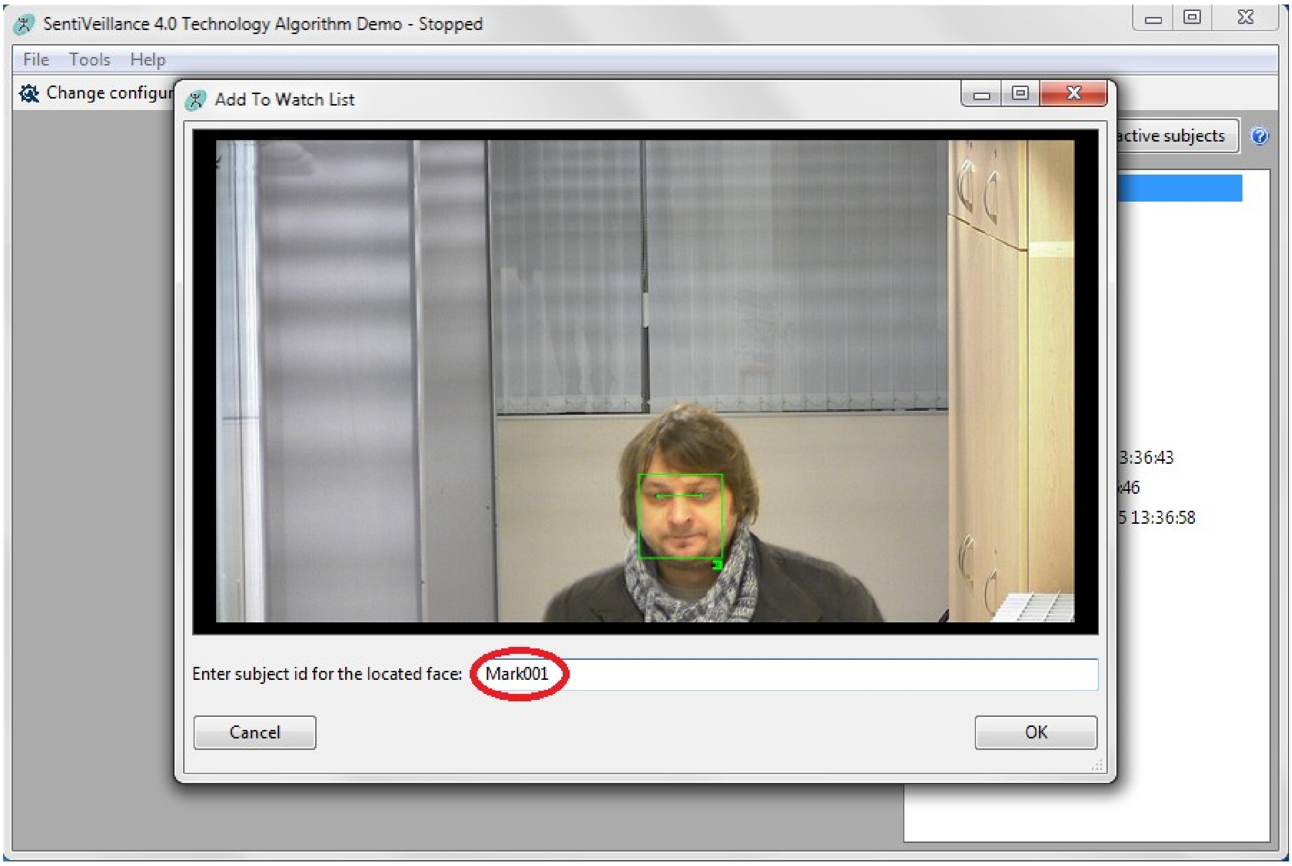


+Source 1 - Subject 8
+Source 1 - Subject 7
+Source 1 - Subject 6
+Source 1 - Subject 5
+Source 1 - Subject 4
+Source 1 - Subject 3
–Source 1 - Subject 2
SubjectType: Faces
FaceConfidence: 54
+appeared at 11/27/15 13:36:43
+track at 11/27/15 13:36:46
+disappeared at 11/27/15 13:36:58
+Source 1 - Subject 1

If a template was generated successfully it will be marked in bold black text (e.g. **Source1 - Subject7**). Such templates can be added to the *Watch List*. To do so, select a template that you want to add (e.g. Source1 - Subject8) and click *Add template* button. *Add to Watch List* dialog window will appear where you must specify the name of the template and then click *OK*. The template with the given name will be added to the *Watch List*.

Warning. *Templates can have the same names. Giving the same name to a new template will not overwrite an existing one - it will add a new template to existing one (it is done by the demo program, SDK users can change this behavior according to their own needs). But giving different names to templates of the same subject (e.g. Mark001, Mark002 ...) will allow you to see the specific template that matched against the tracked face.*

If the algorithm, for some reasons, was not able to generate a template, it will appear as a bold gray text (e.g. Source1 - Subject5).



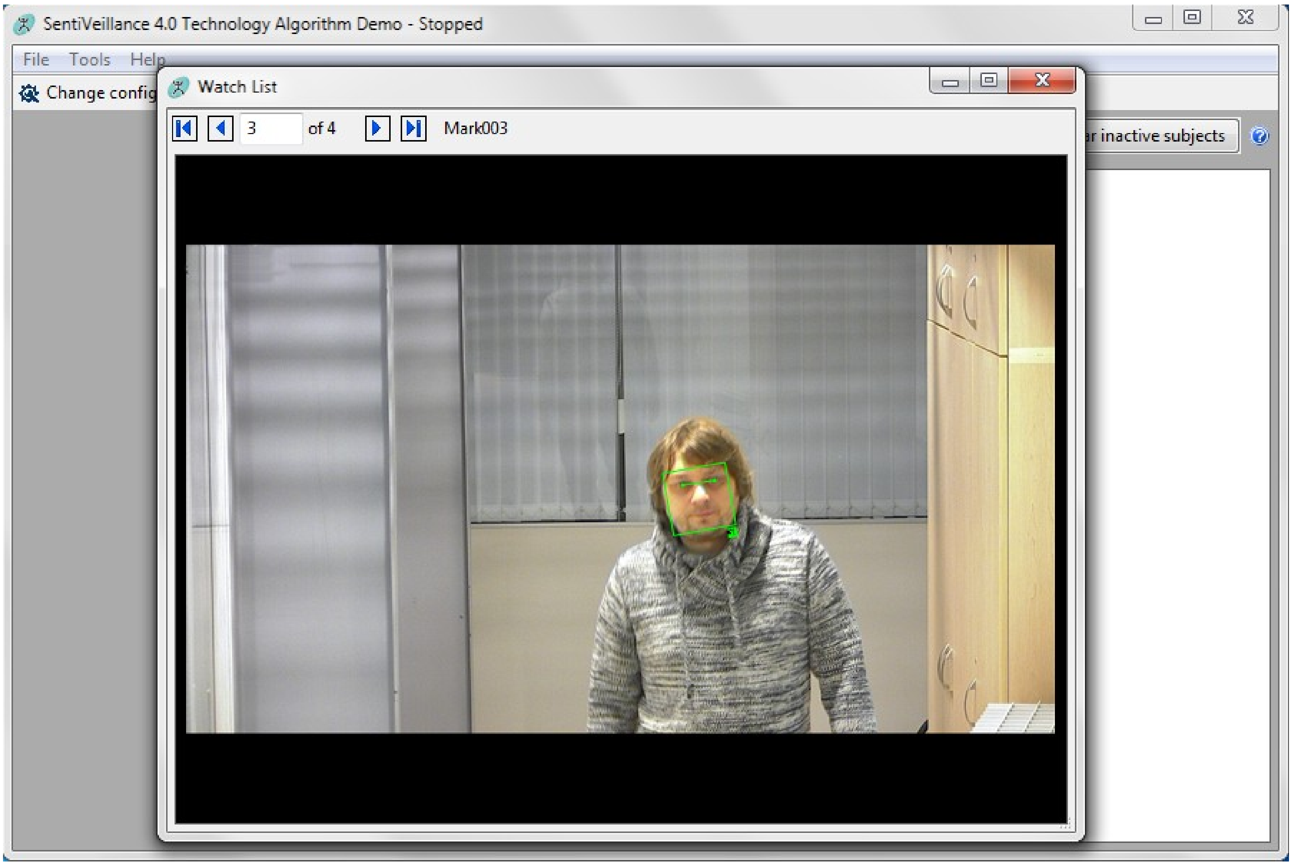
Enter subject id for the located face: Mark001

Subjects can be enrolled using images as well. To enroll from images go to *File* and select *Enroll from image(s)*. In the dialog window you will be able to select single or multiple images for enrollment. Enrollment from the directory can be done in the same way by selecting *Enroll from directory*.

Warning. *During enrollment from images or a directory, templates will be generated for each image and automatically added to the Watch List. The template naming is done automatically according to the name of the image file.*

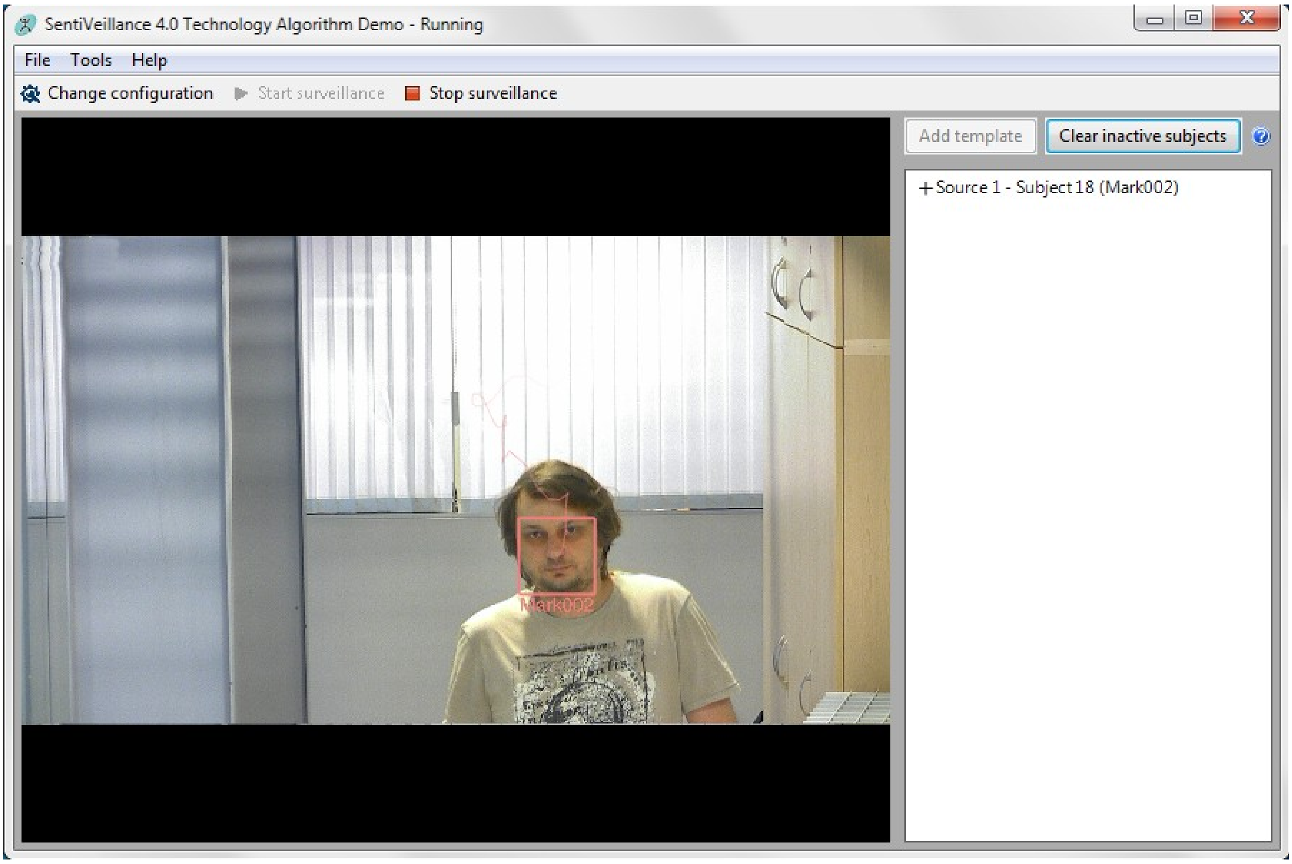
You can always look at your Watch List by navigating to *Tools* and clicking Watch list button. The watch list shows the number of templates in it and names of the templates. The displayed image is associated with a template and does not mean that it is saved in the template. This image is just for representative purposes. The real template uses information of one or more images obtained from face tracking results. Clearing Watch List is done by selecting *Clear watch list* from *Tools* menu.

Warning. *Clearing Watch List will clear the whole watch list. You cannot clear one particular record from the list (this is done by the demo program, SDK users can manage this according to their own needs).*



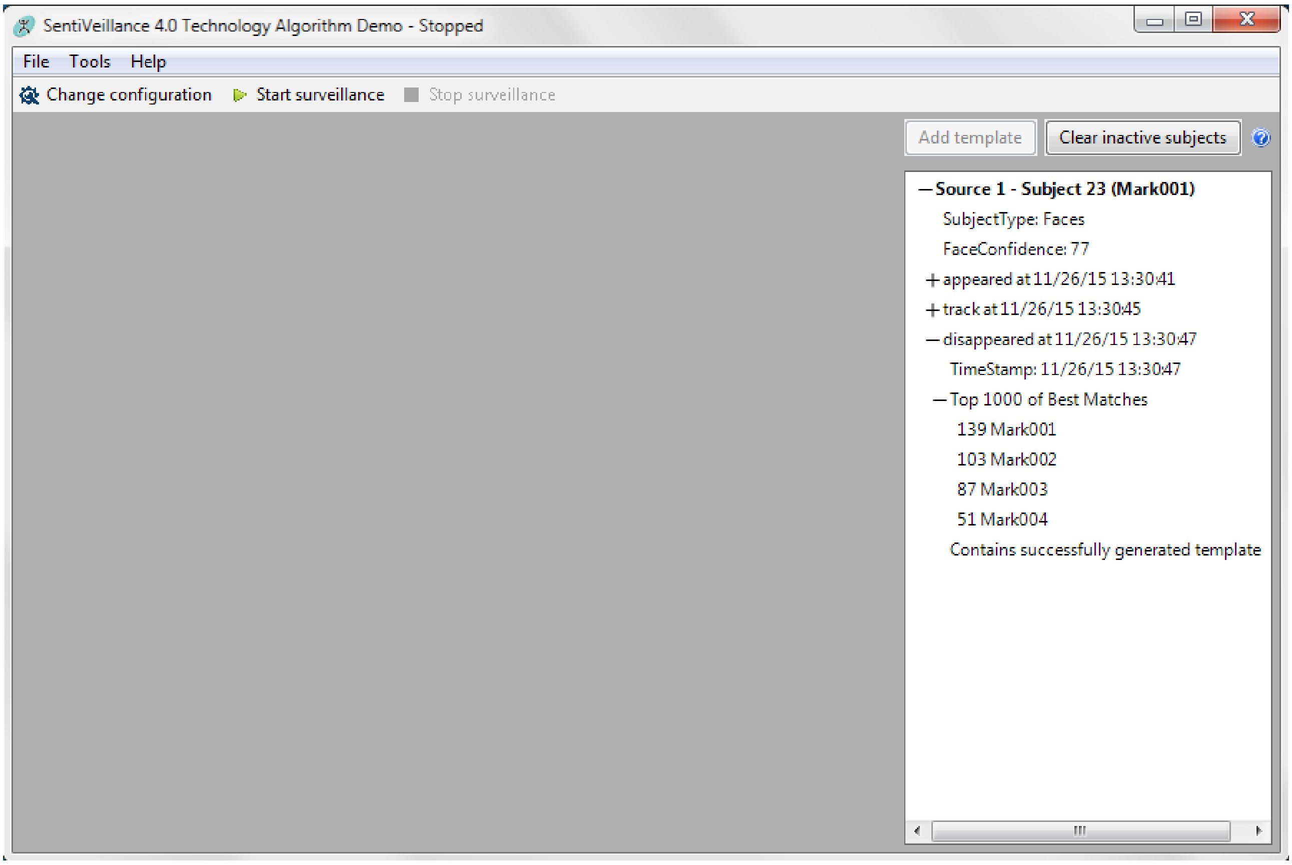
Matching Subjects in the Watch List

Matching is done automatically as soon as *Watch List* has at least one record. After adding subjects to the *Watch List* you can select new video source and start surveillance. Algorithm will be detecting faces in video stream and tracking them, simultaneously performing matching of tracked faces in the *Watch List*.



All tracking results are displayed in the event list window on the right side of the demo. You can see some detailed tracking results by pressing + sign near the event. This drop down list shows useful statistics: subject's appearance and disappearance time, how long it was tracked, etc. Expanding the list near disappeared event will show top best matches, if the tracked face was identified. In this particular example we see that our tracked subject was identified as *Mark001*. The highest matching result for the template *Mark001* was achieved with a score of 139. Lowest result was 51.

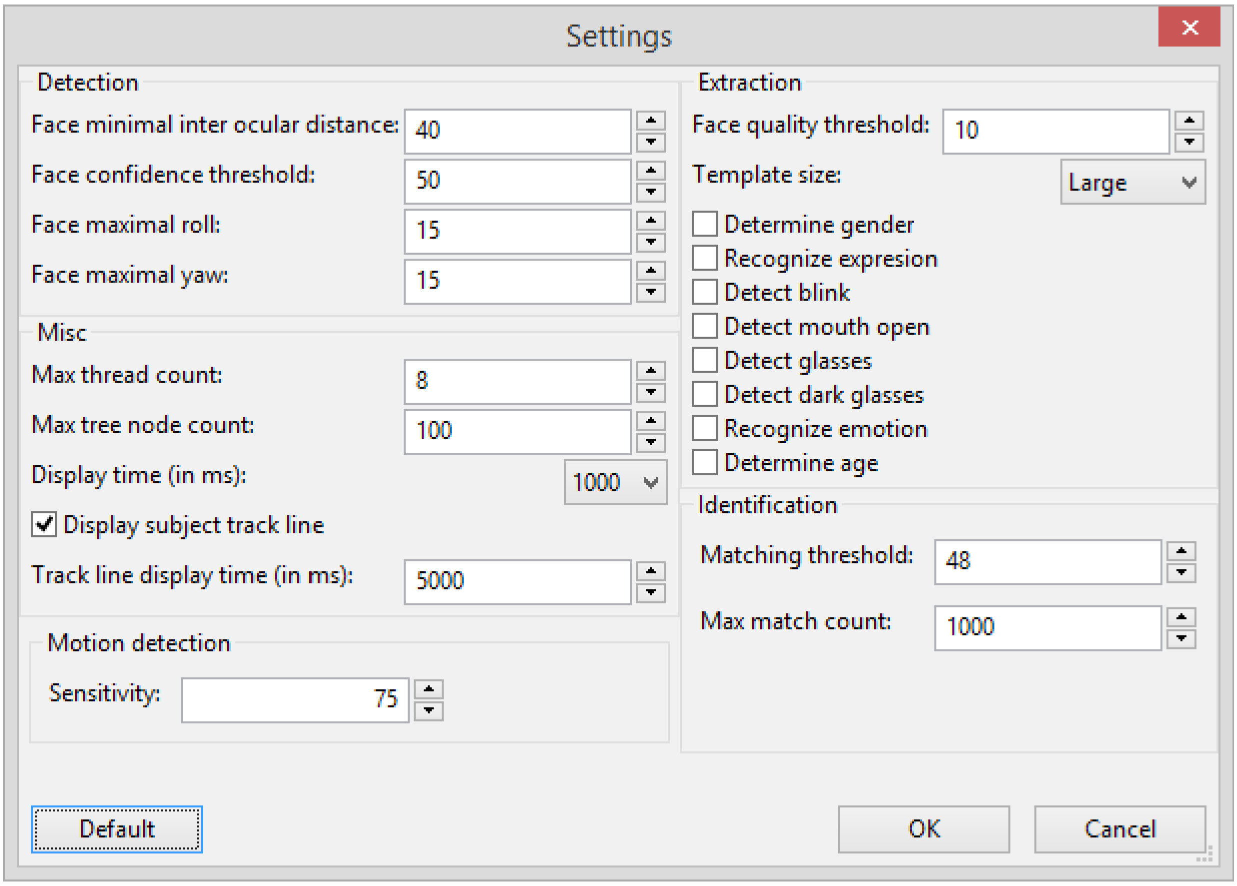
Theoretically, matching score can go up to infinity. But in the real life situations, depending on surveillance system setup and environmental conditions, the score of 70 and higher can be considered as a good matching result.



—Source 1 - Subject 23 (Mark001)
SubjectType: Faces
FaceConfidence: 77
+appeared at 11/26/15 13:30:41
+track at 11/26/15 13:30:45
—disappeared at 11/26/15 13:30:47
TimeStamp: 11/26/15 13:30:47
—Top 1000 of Best Matches
139 Mark001
103 Mark002
87 Mark003
51 Mark004
Contains successfully generated template

Application Settings

Application settings can be accessed by choosing menu item *Tools->Settings*. This allows to change Detection, Identification, Extraction and other application settings:



Detection

Face minimal inter ocular distance: 40

Face confidence threshold: 50

Face maximal roll: 15

Face maximal yaw: 15

Misc

Max thread count: 8

Max tree node count: 100

Display time (in ms): 1000

☑ Display subject track line

Track line display time (in ms): 5000

Motion detection

Sensitivity: 75

Extraction

Face quality threshold: 10

Template size: Large

☐ Determine gender

☐ Recognize expresion

☐ Detect blink

☐ Detect mouth open

☐ Detect glasses

☐ Detect dark glasses

☐ Recognize emotion

☐ Determine age

Identification

Matching threshold: 48

Max match count: 1000

Detection - this set of parameters is related to face detection

- *Face minimal inter ocular distance* - this parameter sets the minimal distance between eyes for face detector. Value of 40 means that faces with less than 40 pixel between eyes will not be detected. The selection of this parameter depends on the camera setup. For low resolution cameras or cameras that capture faces from distance this parameter can be lower. Minimum value of this parameter is 8.
- *Face confidence threshold* - specifies the threshold which is considered when looking for faces in an image. For each face candidate confidence parameter is calculated. With higher threshold value faces are selected more strictly by the face detection routines. Must be in range [0..100]. The default value is 50.
- *Face maximal roll/yaw* - theses parameters specify orientation (pose) angle of the face for the detector. The values are expressed in degrees. For example setting yaw angle to 10 will detect near frontal faces and setting to 45 will try to detect frontal and half profile faces. Must be in range [0..180].

Extraction - this set of parameters is related to face template extraction

- *Face Quality Threshold* - this parameter sets the quality of the face for template extractor. The values of this parameter may vary from 1 to 100. Low value means that template will be generated from low quality facial images like face images affected by motion blur or low dynamic range. The higher value will use sharper images. In practice it is recommended to use higher value during enrolment session and lower value during identification session for example 70 and 50 respectively.

- *Template size* - size of face image templates. Can be used Large, Medium or Small template. It is recommended to use medium template size. This parameter is used when generating a template from tracked subject.
- *Determine gender* - If checked, the application will detect gender (for this and below parameters see NleDetectionDetails)
- *Recognize expression* - recognize face expression (smile).
- *Detect blink* -detects blink (eyes closed).
- *Detect mouth open* - detects if mouth is open.
- *Detect glasses* - detects if face is with glasses.
- *Detect dark glasses* - detects if face is with dark glasses.
- *Recognize emotion* - if checked recognizes face emotion.
- *Determine age* - if checked, subject's age will be determined.

Identification - this set of parameters is related to identification of tracked faces

- *Matching threshold* - the values may vary from 1 to infinity (theoretically). Lower values will give more false matches. The default value is set to 60, but in practice value of 70 or more would be a good choice.
- *Max match count* - maximum length of the list which is returned by each identification. The default value is 10.

Misc

- *Max thread count* - specifies the number of concurrent surveillance threads. Recommended value is "number of logical processors on CPU" plus one. Must be in range [3..8]. The default value is 8.
- *Max tree node count* - maximum number of items in tree node. Defines how many subjects should be displayed in the right panel. Default value is 10.
- *Display Time (in ms)* - specifies the display time.
- *Display subject track line* - if checked, subject tracking line is shown on a screen.
- *Track line display time* - the maximum time for tracking line display.

Motion detection

- *Sensitivity* - motion detection sensitivity.

When camera returns insufficient number of frames per second to sample application, tracking rectangle is not drawn. In this case display time parameter can be used to set time for displaying tracking rectangle.