Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Watchdog System

Madsen, Tanja Kidholm Osmann; Bahnsen, Chris Holmberg; Jensen, Morten Bornø; Lahrmann, Harry Spaabæk; Moeslund, Thomas B.

Publication date:
2016

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Madsen, T. K. O., Bahnsen, C. H., Jensen, M. B., Lahrmann, H. S., & Moeslund, T. B. (2016). *Watchdog System*. European Commission * Office for Official Publications of the European Union. http://www.indev-project.eu/InDeV/EN/Documents/pdf/watchdog-system.pdf?__blob=publicationFile&v=7

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: April 17, 2024

Project No. 635895 — InDeV

InDeV: In-Depth understanding of accident causation for Vulnerable road users

HORIZON 2020 - the Framework Programme for Research and Innovation

Deliverable 4.1

Watchdog System

Due date of deliverable: (30.04.2016)

Start date of project: 01.05.2015
Duration: 36 months

Organisation name of lead contractor for this deliverable: Aalborg University, Denmark

Revision 1.0 (April 2016)

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document information

Authors

Tanja Kidholm Osmann Madsen, Department of Civil Engineering, Aalborg University, Denmark
Chris Bahnsen, Department of Architecture, Design and Media Technology, Aalborg University, Denmark
Morten Bornø Jensen, Department of Architecture, Design and Media Technology, Aalborg University, Denmark
Harry Lahrmann, Department of Civil Engineering, Aalborg University, Denmark, Aalborg University, Denmark
Thomas Moeslund, Department of Architecture, Design and Media Technology, Aalborg University, Denmark

Project Coordinator

Aliaksei Laureshyn
Department of Technology and Society
Lund University
Box 118
221 00 Lund, Sweden

Phone: +46 46 222 91 31
Email: aliaksei.laureshyn@tft.lth.se

Coordinator of WP4

Thomas B. Moeslund
Department of Architecture, Design and Media Technology
Aalborg University
P.O. Box 159
9100 Aalborg, Denmark

Project funding

Horizon 2020
Grant agreement No. 635895

Revision and history chart

Version	Date	Comment

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 635895

This publication reflects only the authors' view. Responsibility for the information and views expressed therein lies entirely with the authors. The European Commission is not responsible for any use that may be made of the information it contains.

Contents

1 Introduction 5
2 Starting the software 6
3 User interface 7
4 The algorithms of the software 9
4.1 Detectors 9
4.2 Detector modules 11
4.2.1 Single 12
4.2.2 Double 12
4.2.3 Special modules (developed for: “Intersection Solutions for Cyclists”) 14
5 Analysis in RUBA 16
5.1 Import of videos 16
5.2 Creation of modules for the analysis 18
5.3 Calibration of parameters 21
5.4 Run the analysis 26
References 28

1 Introduction

This deliverable is part of WP4. Overall WP4 is motivated by the need for automatic systems that can ease the task of annotating massive amounts of traffic data. Concretely this deliverable is related to WP4.2 - the watchdog system. The idea with the watchdog is to develop a system that can remove huge chunks of video data where no events/interactions of interest are occurring and hence let a user focus on manually annotation of only the interesting stuff.
We have named the watchdog system *RUBA* (Road User Behavior Analysis) and this name it used throughout this report. Since this report has a practical focus, we not only describe the algorithms behind the system, but also focus on how *RUBA* is used.

In short, *RUBA* is a software program for computer-based analysis of traffic videos. The program can be used on Windows computers.

RUBA allows the user to draw fields (detectors) on the video image by using a simple click-based drawing tool. The sensitivity of the detector, regarding movement in the image, is adjusted by different parameters in the program. Figure 1 shows an example of how traffic analysis can be conducted in *RUBA*.

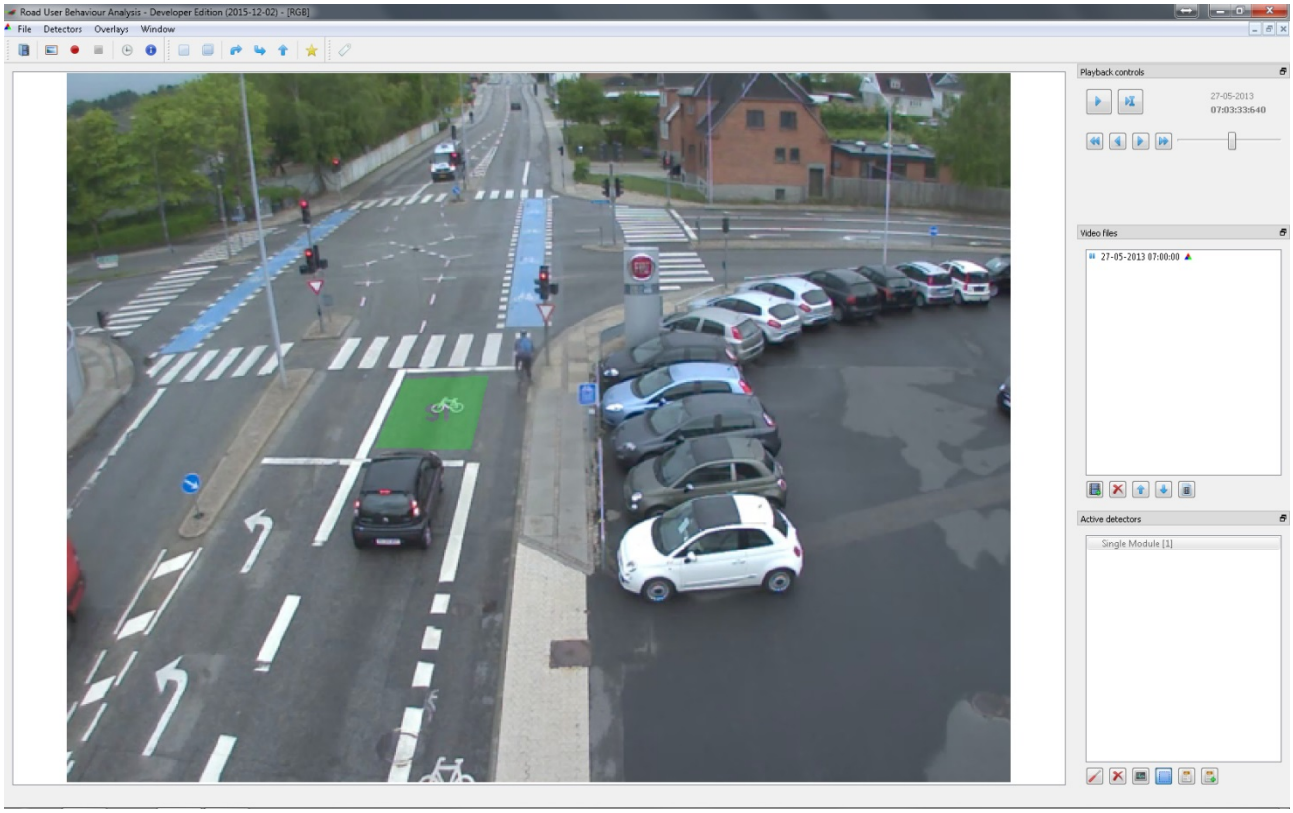


Figure 1 | Analysis in *RUBA*

2 Starting the software

RUBA is a portable application which means the software will run without being installed. The program is opened by clicking the file “RUBA.exe”.

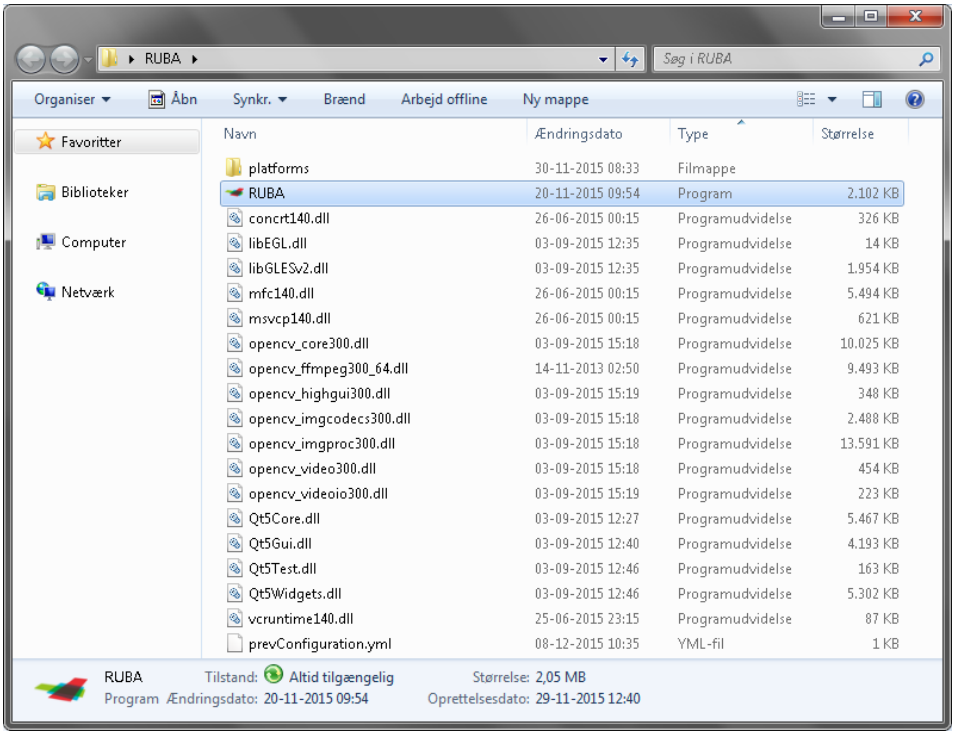


Figure 2 | *RUBA* opens directly without installing anything

Upon opening *RUBA* the following user interface will be shown (Figure 3).

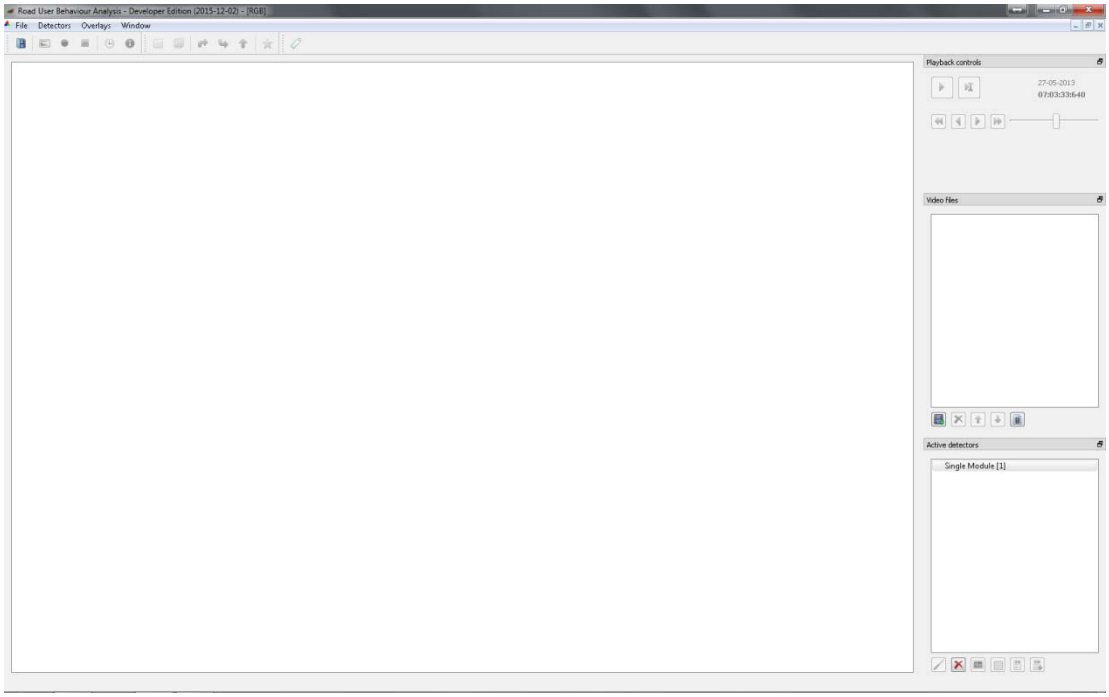


Figure 3 | User interface shown when opening *RUBA*

3 User interface

Figure 4 illustrates the main window of *RUBA*, after a video has been imported. Until then, most buttons are inactive. In the following, the function of each button is described. Keyboard shortcuts are defined in square brackets.

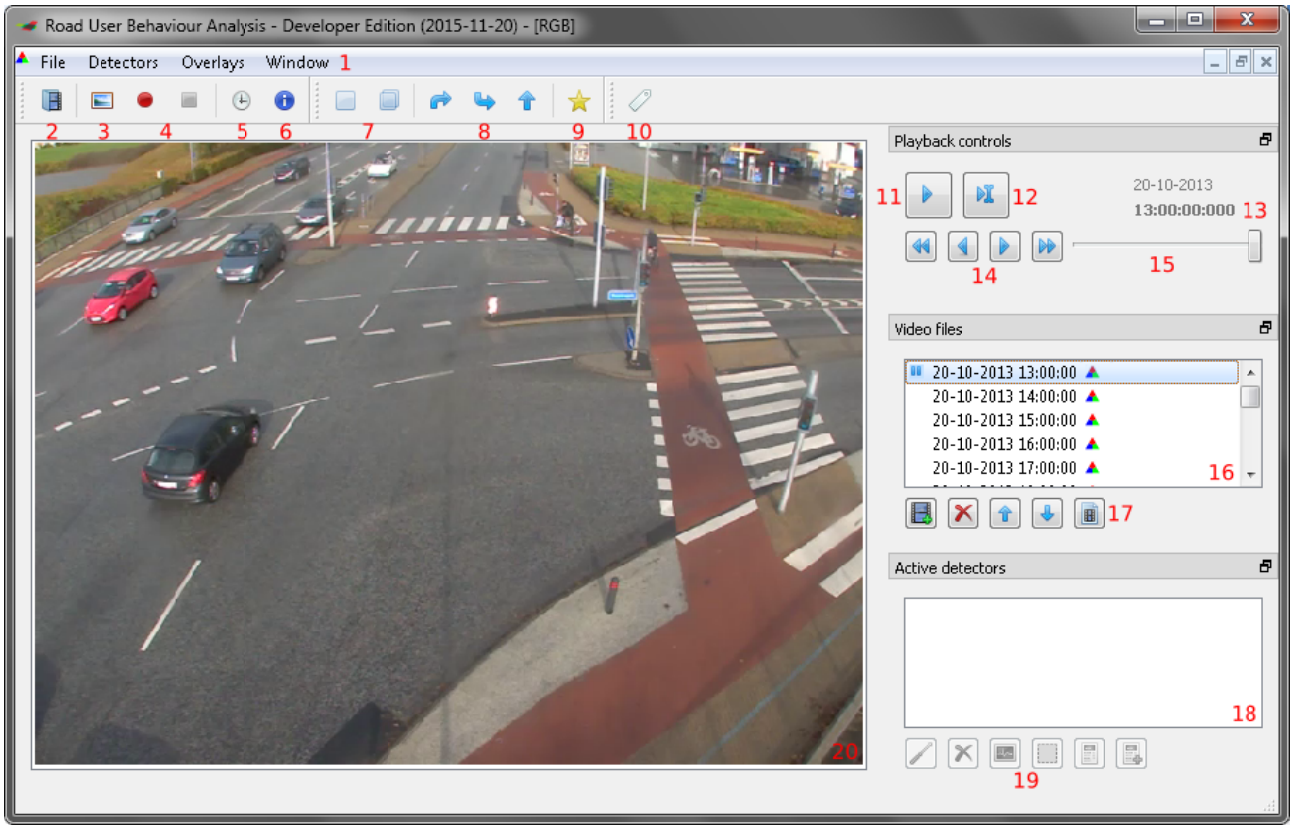


Figure 4 | User interface shown when one or several videos have been imported

1. Main menu. In the tabs, the same functions that are accessible via buttons in the main window can be found, as well as a number of program settings to use before conducting the analysis.
2. Import video(s) [Ctrl + O].
3. Take a screenshot of the video pane (20).
4. Record a video of the video pane. Clicking the red dot starts the recording. It is possible to pause the recording by clicking the button again. The recording can be resumed by clicking the red button again. When clicking the square button the recording is finished. All overlays (detectors, etc.) which are shown in the video pane (20) will appear in the recording.
5. Add/remove overlay which shows a timestamp (13) in the video [F3].
6. Add/remove the information panel. This panel contains several information (time stamp, detector triggered or not) about the activity that is registered in the detectors [F4].
7. Flexible analysis tool (detector) consisting of a single module (left) [Ctrl+1] or double module (right) [Ctrl+2].
8. Analysis tools to detect right turning [Ctrl+3] and left turning [Ctrl+4] vehicles and straight going (Ctrl+5) cyclists. The modules have been developed specifically for the project “Intersection Solutions for Cyclists”. (Madsen *et al.*, 2014)

9. Interaction modules. Combines the analysis modules (7) into interactions by means of time gaps between detections in the modules. Can be used to detect interactions between right turning vehicles vs. straight going cyclists as well as left turning vehicles vs. straight going cyclists [Ctrl+6]. Developed specifically for the project “Intersection Solutions for Cyclists”. (Madsen *et al.*, 2014)
10. Annotate Ground Truth. Opens a “ground truth annotator”-panel which can be used to manually detect activity. These detections can be used to calibrate detectors.
11. Start/pause analysis [space].
12. Jump to a specific frame in the video.
13. Date and time for the video.
14. Navigate through the video. Use these four buttons to respectively jump five frames previous [A], one frame previous [S], one frame forward [D] and five frames forward [F].
15. Adjust video speed. When the slider is placed to the far left the video is paused. When the slider is placed to the far right the video is sped to the maximum.
16. Imported videos. The video that is currently played is marked with a “pause” symbol.
17. Respectively add videos [ctrl+insert], delete imported videos [ctrl+del], change the order of the videos [ctrl+arrow up] and [ctrl+arrow down]. The button to the far right contains properties for the video (start/end time, frame rate, file name and resolution).
18. Inserted/created detectors.
19. Support tools for creating and calibrating of detectors. Consist – among others – of histograms for the detectors, overlay fields to mark the position of the detectors, and information about activity in the detectors.
20. Video pane.

4 The algorithms of the software

4.1 Detectors

The algorithm behind the software consists of three detector types (edge, flow and stationary) with different attributes.

The *edge detector* checks if there is motion in a specific area of the video by means of *Canny edge background subtraction* (Canny, 1986). With this method objects (vehicles, road users) that are not a part of the background (the road, the surroundings) are extracted. This is done by converting the image to a grey scale image and finding edges (continuous lines), where large variations in the contrast appear. In this way the algorithm finds road markings, changes in the pavement, and road users. This is done for all the frames of the video. For two consecutive frames, vectors between the lines are created and summed up. In this way we get a measure for the activity which is based partly on the size of the object, partly on the speed of the object moving across the area. In order to take noise in the image into consideration; i.e. from small changes in the contract, birds, movement of leaves, or shadows, the sensitivity of the edge detector is controlled by some parameters. The background is updated regularly to extract elements that are consistent in the image for a long time, or elements that occur due to changes in the light conditions and the creation of shadows.

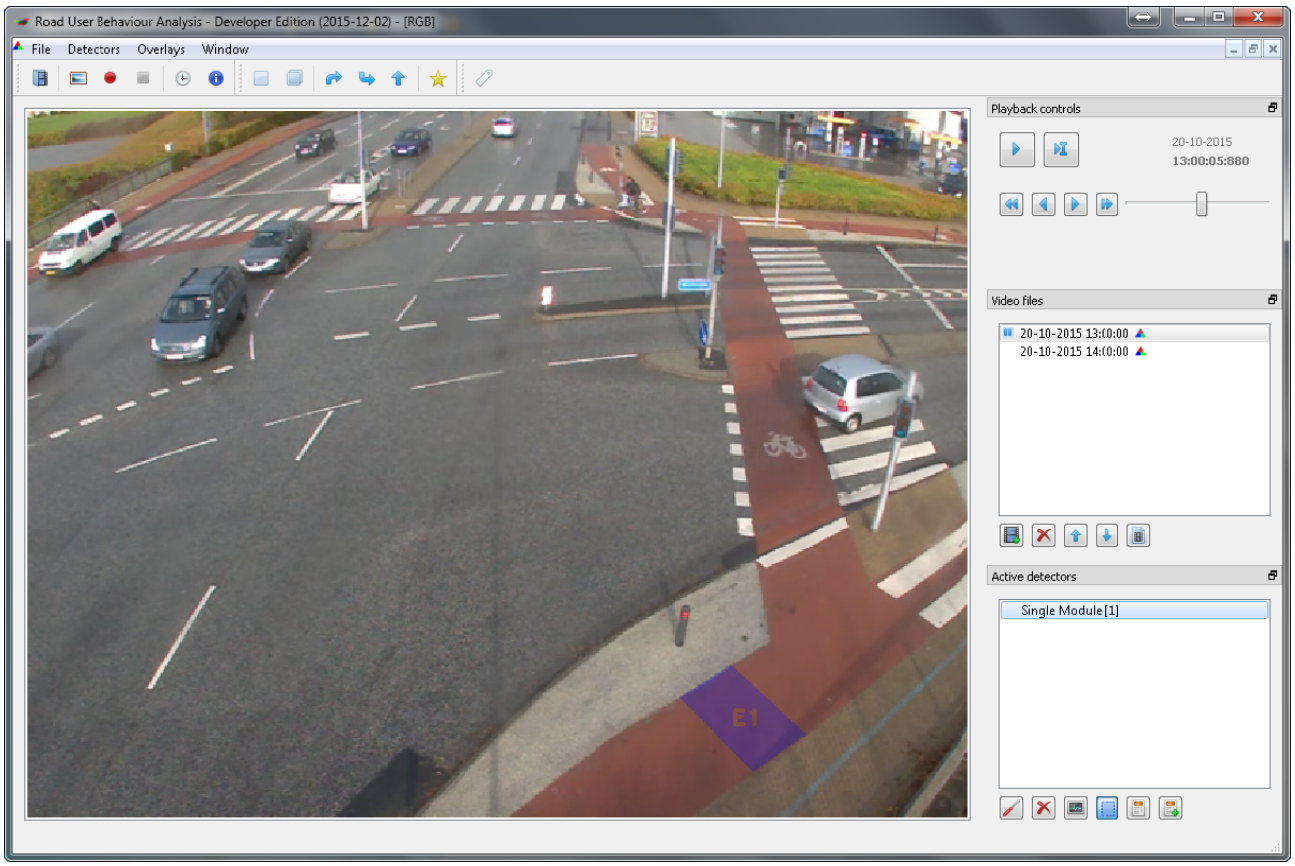


Figure 5 | The edge detector (blue). The edge detector detects activity (movement) through the detector

The *flow detector* checks if there is activity in a specific direction in a certain area of the video by means of the *Farnebäck dense optical flow estimation* (Farnebäck, 2003). In the flow detector points in two consecutive frames are identified and matched. Objects moving across the detector will result in vectors that are dependent on the direction and speed of the object. Higher speeds will result in larger vectors. The amount of vectors per direction (in terms of degrees) is summed

up for those vectors that are higher than a predefined value of the speed of the object. Vectors below this value are omitted.

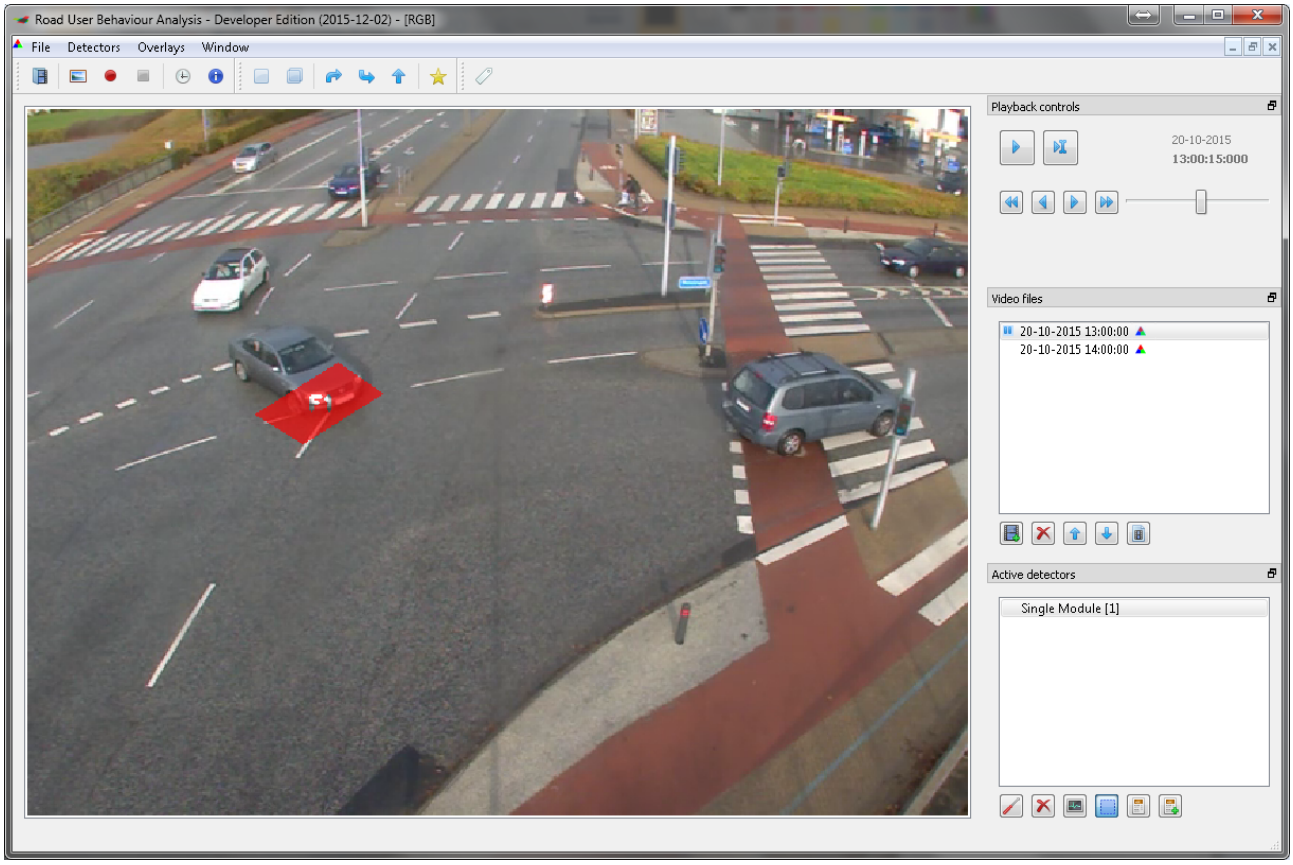


Figure 6 | The flow detector (red). The flow detector detects activity (movement) in a specific direction through the detector

The *stationary detector* detects if an object is idling or moves very slowly through a specific area of the video by means of a combination of the edge and flow detectors. To detect an object (a road user) the edge detector must be triggered, while the flow detector must not be triggered. This indicates that an object is present in the area but is moving very slowly or not moving at all.

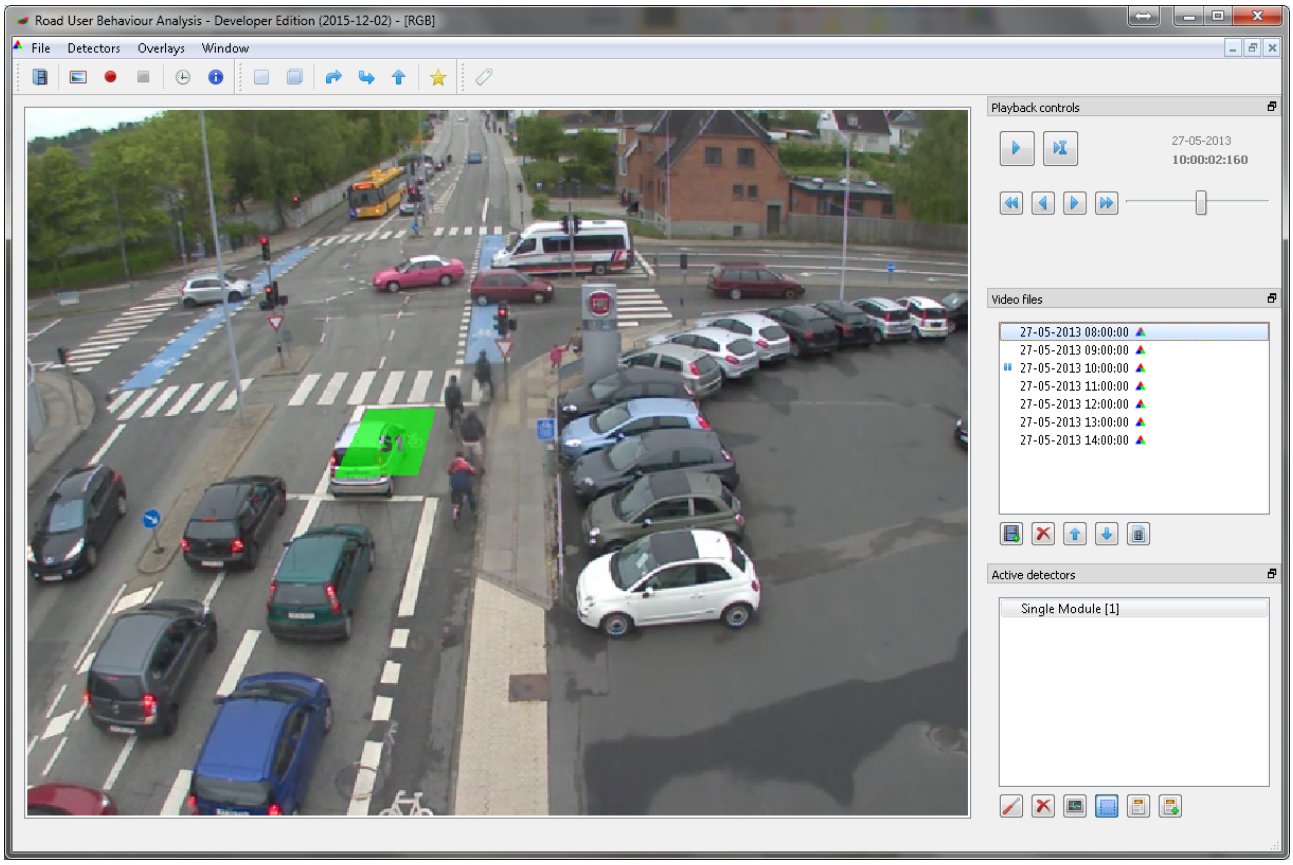


Figure 7 | The stationary detector (green). The stationary detector detects if something is idling or moves slowly through the area covered by the detector

Examples of the application of the three detector types are shown in Table 1.

Table 1 | Applications of the detectors

Edge	Flow	Stationary
Simple analysis and traffic counts. Traffic counts in sections. NB! The highest accuracy is obtained if the traffic streams are separated.	Analysis and traffic counts in areas shared by road users from different directions (e.g. in intersections). Road users driving in the opposite direction of travel.	Analysis of road users that do not move. Detection of parked cars.

4.2 Detector modules

The three detectors can be combined in detector modules. The modules manage ”logic” between two or more detectors including combinations and mutual dependence between the detectors. Furthermore, the modules define when a detector takes on one of three states:

1) *Activated*
   When a detector is activated, movement in the field can be registered.
2) *Triggered*
   The detector has registered activity of the right type (e.g. the right direction) and in an

extent that indicates that the movement comes from a road user (and not just noise in the image).

3) *Flagged (results in the detection of an event)*
   When the detector has been triggered for a number of consecutive frames, an event is registered and saved in a log file.

RUBA consists of a single module (one detector) and a double module (two detectors). Furthermore, specific modules for left and right turning vehicles and straight going cyclists has been developed.

4.2.1 Single

Consists of one optional detector (edge/flow/stationary), see Figure 8. An event is saved in a log file when the criteria are met according to the specifications of the detector in question.

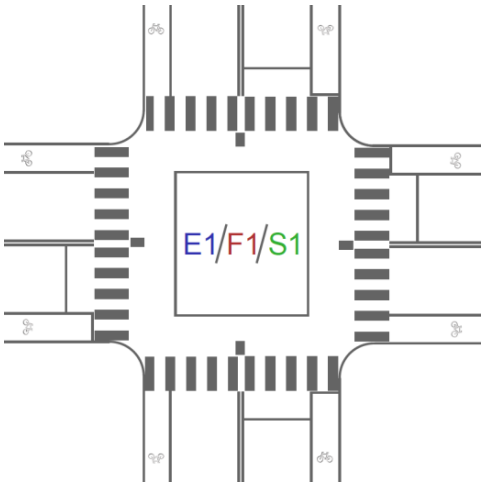


Figure 8 |The single module lets the user create one detector of one's own choice per module

4.2.2 Double

Consists of two optional detectors (edge/flow/stationary), see Figure 9. An event is detected and saved in the log file when both detectors have been triggered within a specific time distance defined by the user.

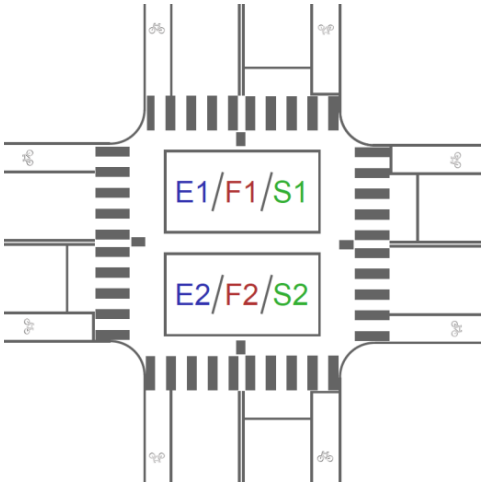


Figure 9 | The double module lets the user create two detectors of one's own choice per module

Figure 10 | Example of a single module

Figure 11 | Example of a double module

4.2.3 Special modules (developed for: “Intersection Solutions for Cyclists”)

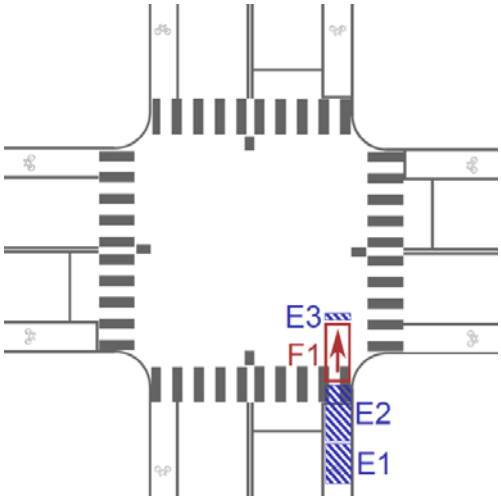


Figure 12 | Detection of straight going cyclists. To detect a cyclist the detectors must be triggered in the order E1, E2, F1 and E3. Each detector will be activated (i.e. can be triggered) for a limited number of seconds after the previous detector has registered activity. A cyclist is recorded by the time for entering and leaving the field E3. In the flow detector movement in the wrong direction – i.e. from left to right – to sort out right turning cyclists

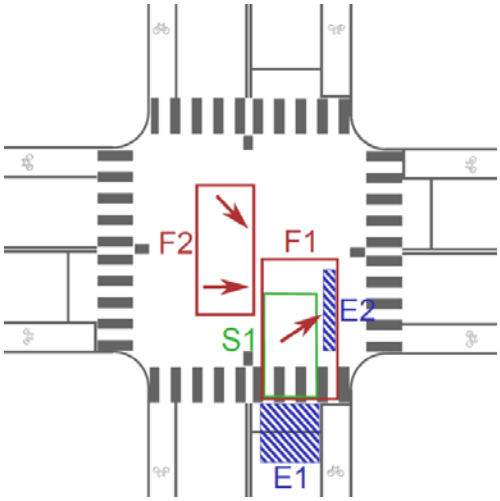


Figure 13 | Detection of right turning vehicles. The vehicle is registered if the detectors has been activated and triggered in the order E1, F1 and E2. If a right turning vehicle stands in the middle of the intersection before the turn is completed it is registered by S1, which will extend the time where F1 is activated. The field F2 sorts out activity from road users from left to right as well as left turning vehicles from the opposite direction. This is done to avoid false positive detections in situations where E1 has been activated by vehicles standing in the area

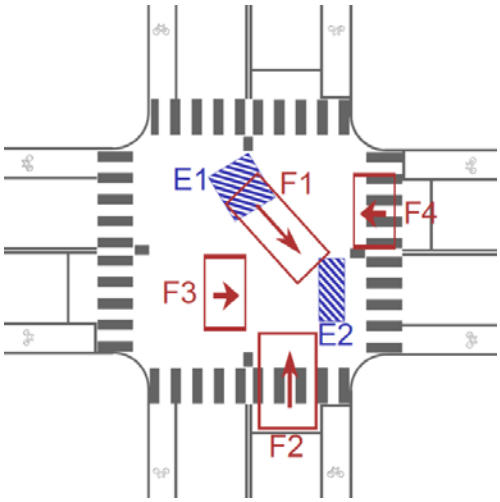


Figure 14 | Detection of left turning vehicles. The vehicle is registered og the detectors are activated and triggered in the order E1, F1 and E2. The fields F2-4 sort out activity from other vehicles to reduce the risk of false positive detections

Figure 15 | Example of the design of detectors in the module for straight going cyclists. The module consists of three edge detectors (blue) and one flow detector (red)

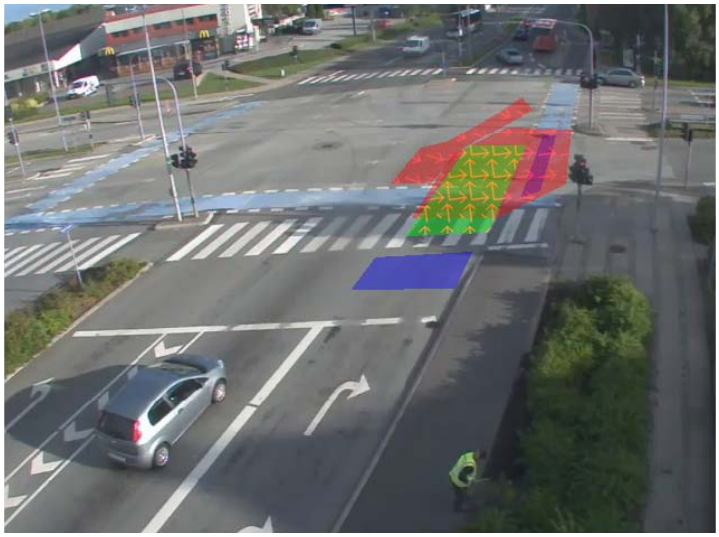


Figure 16 | Example of the design of detectors in the module for right turning vehicles. The module consists of two edge detectors (blue), two flow detectors (red) and one stationary detector (green)

Figure 17 | Example of the design of detectors in the module for left turning vehicles. The module consists of two edge detectors (blue) and four flow detectors (red)

5 Analysis in RUBA

The procedure when conducting an analysis in *RUBA* is as follows:

1. Import video(s)
2. Create module(s) for the analysis
3. Calibrate parameters to ensure that the right movements/road users are registered
4. Run the analysis

5.1 Import of videos

RUBA works with videos independent of the resolution and frame rate. *RUBA* remembers the settings from the last time it was used and therefore these settings must be checked every time videos with different names or frame rates are used.

Setting up the video can be done by clicking the tab “File” and selecting “Settings”. In the “Settings”-window choose “Video”:

1. Playback speed: Used to decide the speed of which the video will be analysed. The speed can be altered later on.
2. Frame rate: Used to define the frame rate. The value should match the frame rate of which the video is recorded. It is possible to let *RUBA* auto-detect the frame rate.
3. Date and time: Used to define the date and time of which the video is recorded. This can be set in a way that fits the file name, so that the information can be imported automatically. The format is chosen as either “MM-dd-HH” (month-day-hour), “yyyyMMdd-HH” (year month day-hour), “yyyy-MMdd-HH” (year month day-hour), “yyyyMMdd-HH-mm-ss” (year month day-hour-minute-second) and “user defined”, in which the date and time is specified manually every time the video is played from the beginning. If the first format is chosen (MM-dd-HH) the year must be specified manually.
4. Resolution: Used to create a warning if the imported video is recorded at a low resolution. The width and height of when the warning is created can be set manually.

Videos are imported, see Figure 19. Multiple videos can be imported at the same time.

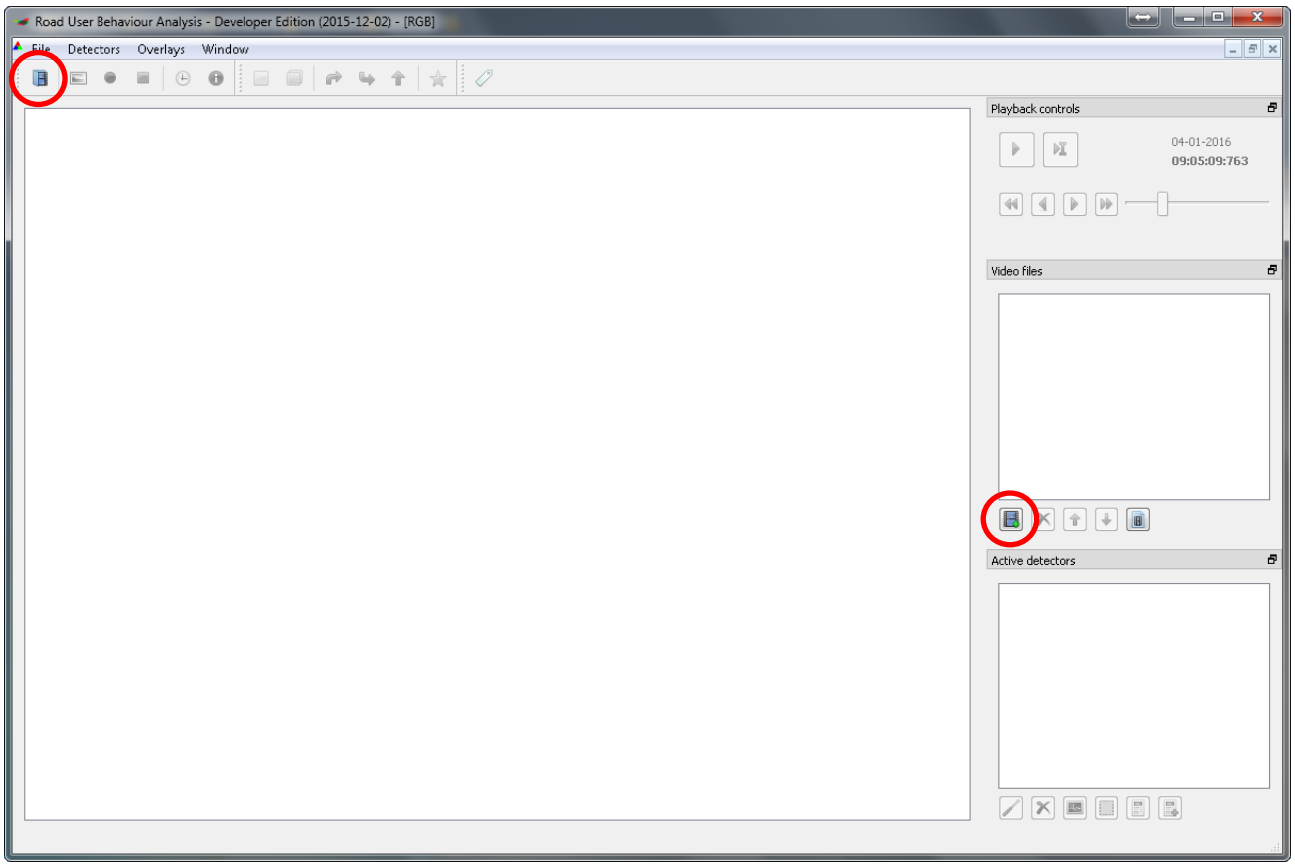


Figure 18 | Import of videos is done via either of the two buttons

5.2 Creation of modules for the analysis

After the videos have been imported the first video is shown in the window pane. Firstly navigate to a spot in the video where no road users are present in the area of interest. This is done to ensure that the program can create a background image without traffic to compare consecutive images with in order to detect the road users.

A module for the analysis is created by pressing the button for the desired module (Figure 19).

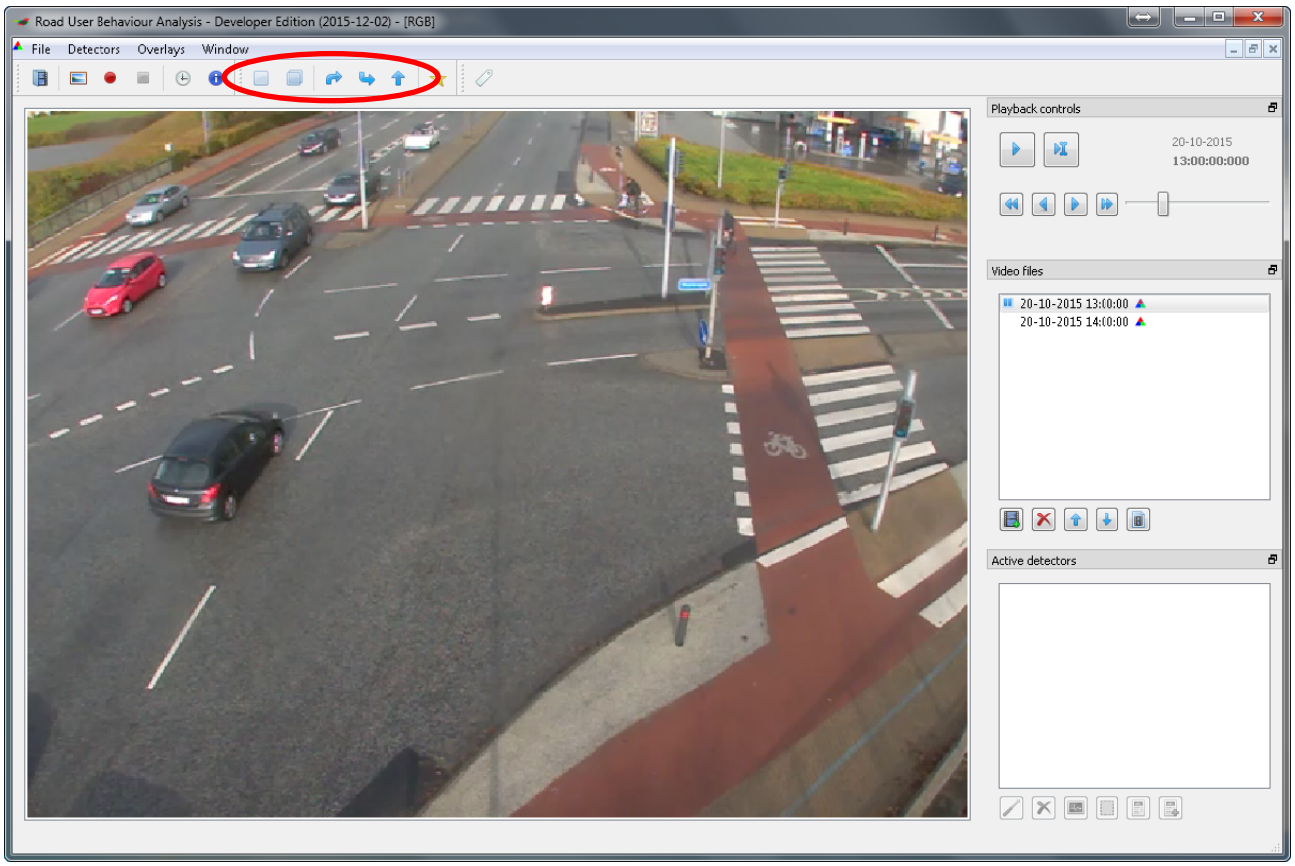


Figure 19 | Creation of modules

Choose the desired detector type. Then press OK (Figure 20)

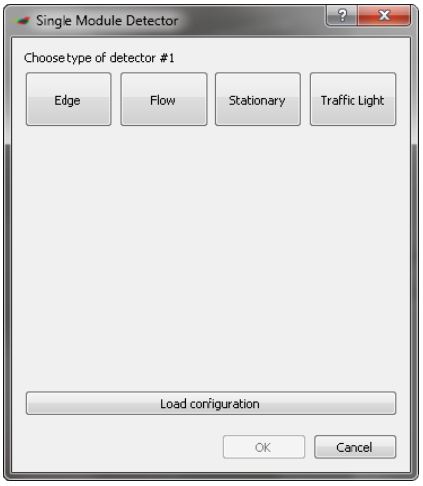


Figure 20 | Choice of detector type in single module

After the desired detector have been chosen a new window opens. This window contains the settings of the detector and lets the user draw the detector (Figure 21).Via ”Configure detectors” the detector is chosen, after which drawing tools to create the detector and a number of detector settings appears (Figure 22). The settings depend on the chosen detector type.

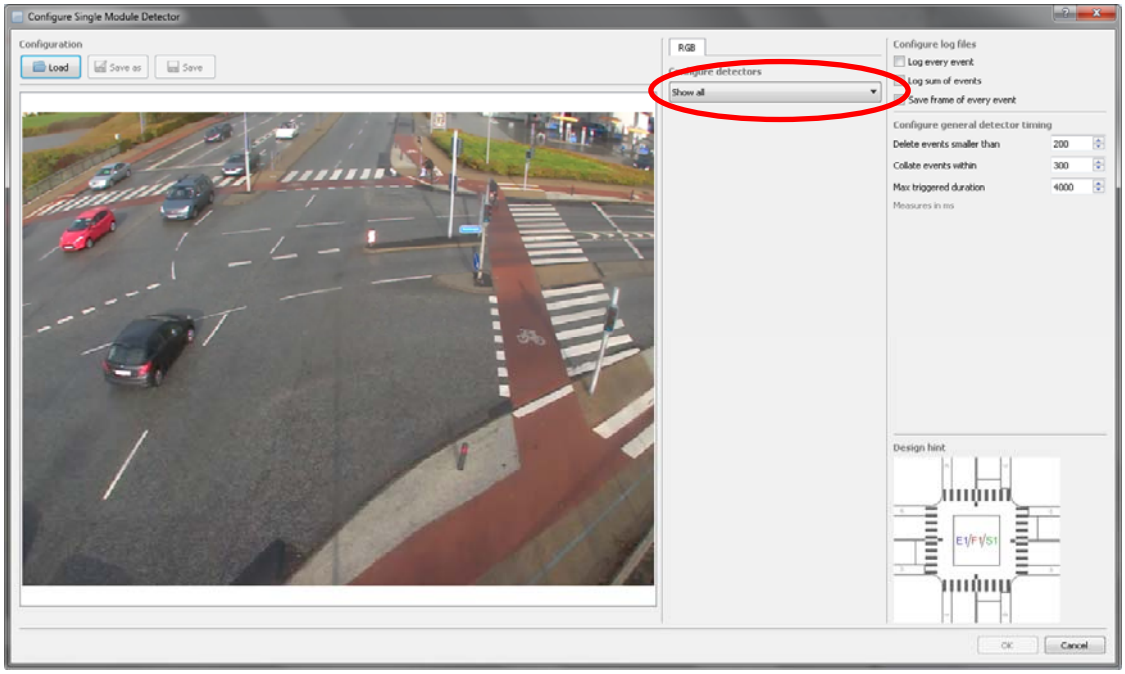


Figure 21 | Creation of detectors

To draw the outline of the detector, click on the pencil. The detector is drawn by clicking in the image. Straight lines are created between the points. The latest point can be deleted by right clicking.

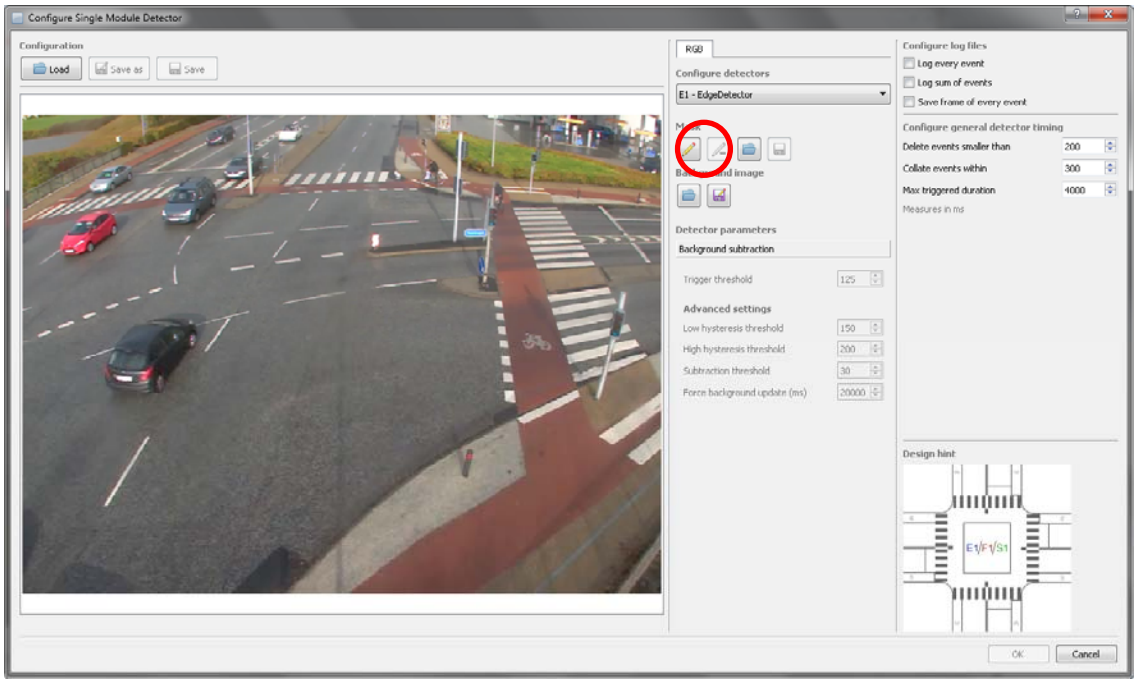


Figure 22 | Drawing tools and settings for an edge detector

Once the detector has been drawn (i.e. it only needs to be closed), double click or press the green tick mark (Figure 24). After this, the parameters can be adjusted, and it can be chosen if logs should be created (”Log every event” and ”Log sum of events”)

The detector is saved via the ”Save”-button before the window is closed via the “OK”-button (Figure 23).
Configuration files that have previously been saved can similarly be imported in this window.

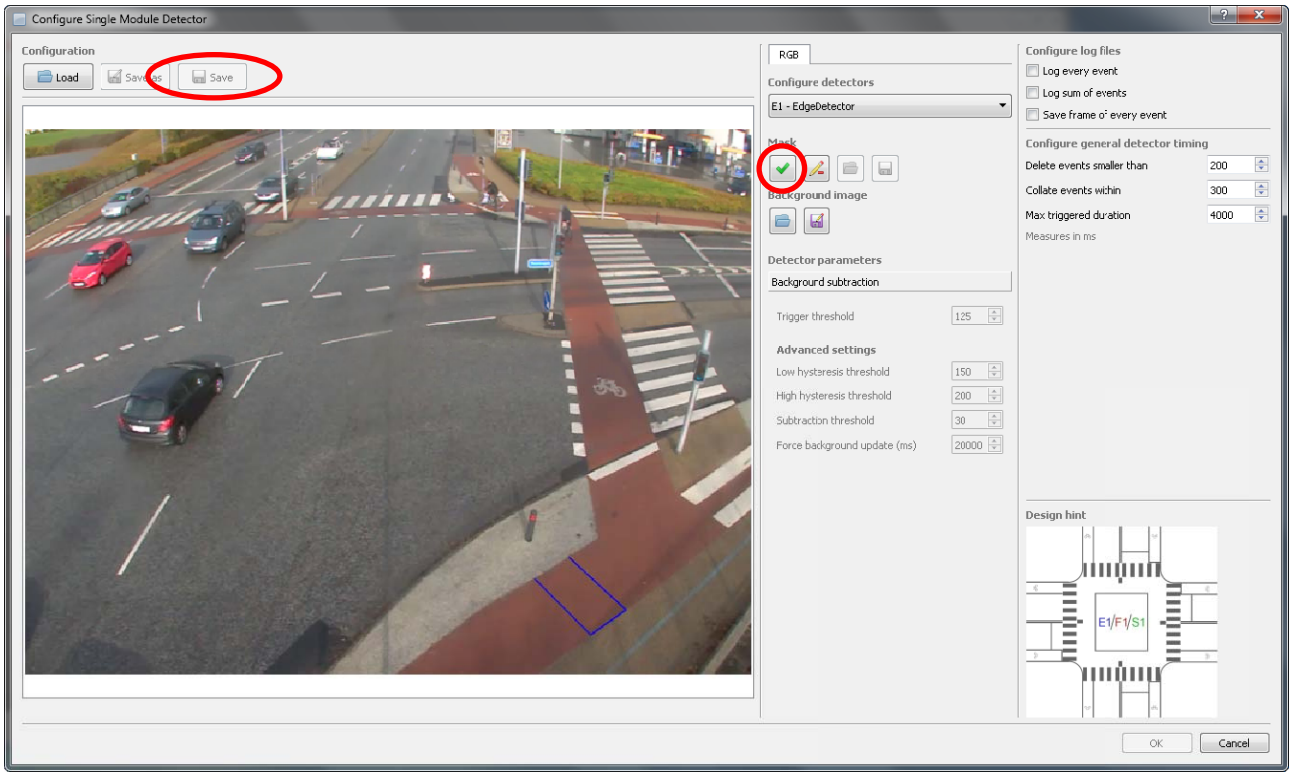


Figure 23 | Creation of detector using the drawing tool

5.3 Calibration of parameters

To ensure that the right objects are detected the parameters must be calibrated. This is done via a number of tools which let the user gain insight into what is detected by the algorithms (Figure 24).

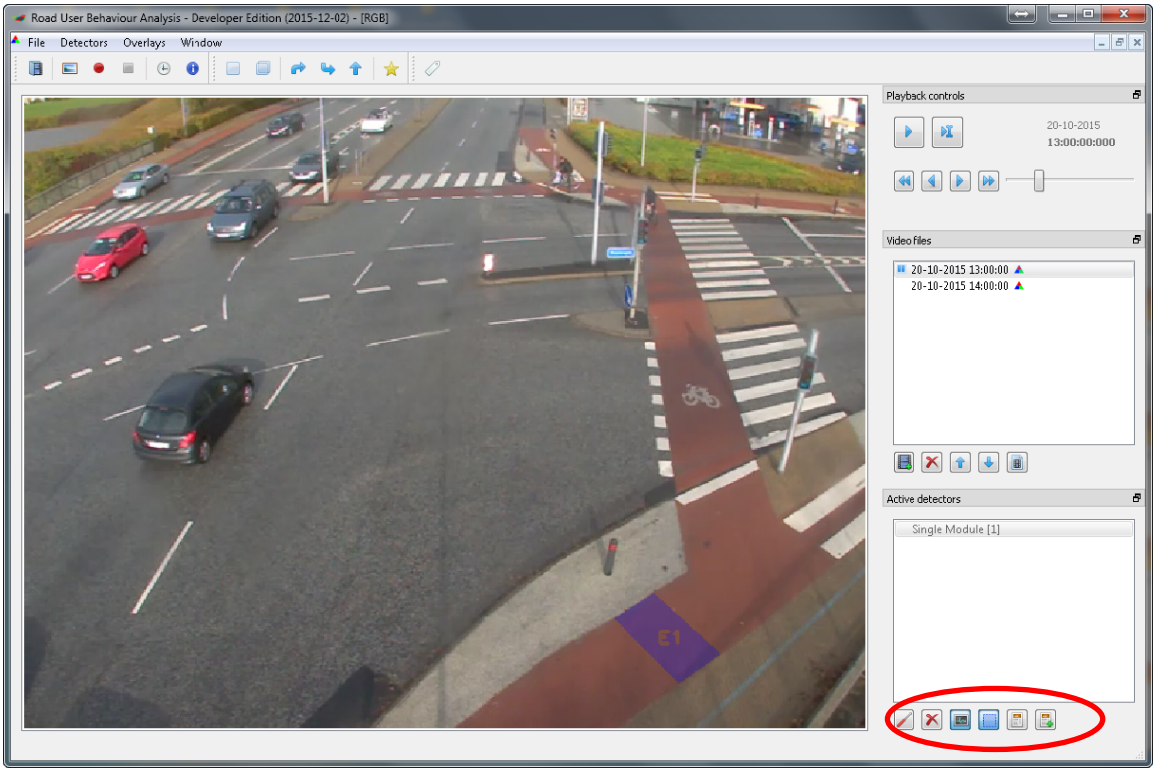
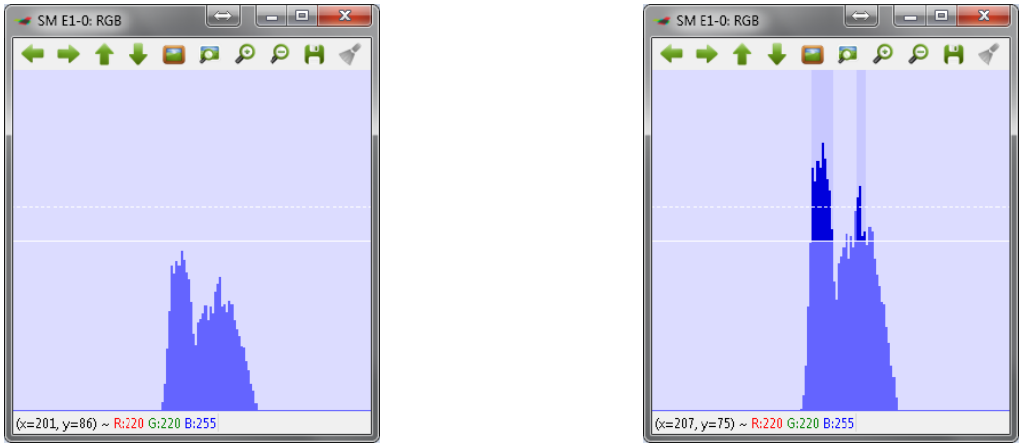


Figure 24 | Tools for calibration of the detectors. From left: 1) edit the detector. Editing can also be done by right clicking on the detector in the "Active detectors" window. 2) delete the marked detector. If the detector has not been saved, it is deleted completely and cannot be imported. 3) histograms. 4) Overlay of the detector in the image. 5) information about activity in the detector. 6) extended information about the detector

The most important tool for the calibration is the histograms of the activity in each detector (Figure 26).



Before calibration

After calibration

Figure 25 | Histograms are used to adjust the detector. If the amount of activity is sufficiently high, so that the software recognizes it as a road user, the activity is marked with a bright colour (in this case blue). If the parameters of the detector are not adjusted correctly, then the road user will either be missed or only partly detected (left). After the adjustments of the parameters the road user will be clearly detected (right)

The histogram of a stationary detector differs from the histograms of the edge and flow detectors, see Figure 26. When the histogram has a bright colour, relevant activity has been registered.

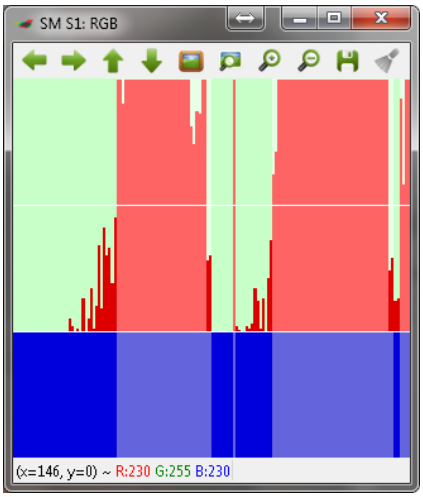


Figure 26 | Histogram of a stationary detector. The detector consists of a combination of the edge and flow detectors. When an object is present in the area of the detector (shown with a bright blue colour), which are moving sufficiently slow (shown with a bright red colour), the object is registered. Bright colours are only shown if both criteria are met. Otherwise, the activity is shown with pastels and no objects (i.e. road users) will be logged

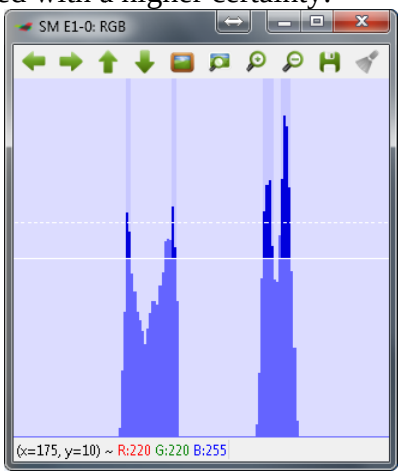
The software offers different parameters to adjust the sensitivity of the detectors. Table 2 gives an overview of the parameters that can be used to adjust each detector. In the following the function of each parameter is described.

As a main rule, only the *trigger threshold* should be adjusted – including *flow range* and *minimum speed* for the flow detector, and *minimum speed* and *max vector count* for the stationary detector – but other advanced parameters may be adjusted as well if necessary.

Table 2 | Adjustable parameters in the detector modules. E = edge detector, F = flow detector, S = stationary detector

	Single			Double			Right turn	Left turn	Straight going	Inter-action
	Edge	Flow	Stationary	Edge	Flow	Stationary				
Trigger threshold	x	x	x	x	x	x	E, F, S	E, F	E, F	
Low hysteresis threshold	x		x	x		x	E, S	E	E	
High hysteresis threshold	x		x	x		x	E, S	E	E	
Subtraction threshold	x		x	x		x	E, S	E	E	
Force background update (ms)	x		x	x		x	E, S	E	E	
Active duration							E, F, S	E, F	E, F	
When active, disable other detectors for							F	F		
Minimum speed		x	x		x	x	F, S	F	F	
Flow range		x			x		F	F	F	
Max vector count			x			x	S			
Disable all detectors after							x	x	x	
Delete events smaller than	x	x	x	x	x	x	x	x	x	
Collate events within	x	x	x	x	x	x	x	x	x	
Max triggered duration	x	x	x	x	x	x			x	
Minimum delay				x	x	x				
Maximum delay				x	x	x				
Car before cyclist time gap										x
Cyclist before car time gap										x

Trigger threshold
Limit for when an activity will be registered. The parameter is used to sort out noise in the video. In the histograms the trigger threshold is shown as a horizontal line, which has been split into two. The lower solid line marks the level of the trigger threshold, while the upper dotted line denotes that the activity is detected with a higher certainty.



Low hysteresis threshold/High hysteresis threshold
Denotes which edges the edge detector will find. Every time the edge detector finds an edge with a contrast that is higher than the high hysteresis threshold it begins searching for the rest of the edge. To take into account parts of the edge that may have a lower contrast than the high hysteresis, a value for the low hysteresis threshold is applied. The low hysteresis threshold is the lowest acceptable contrast difference of the edge. The lower value, the lower variation in the contrast is necessary to be detected as an edge. Thus, if using low values, more edges will be found. However, low values also make the detector more prone to noise in the image.

Subtraction threshold
Denotes what is considered to be an edge. If using low values for the subtraction threshold, the probability of detecting objects belonging to the foreground (i.e. is assumed to be road users) increases. Though, low values also increase the risk of detecting objects that belongs to the background (trees, traffic signs, road marking) if the light conditions changes.

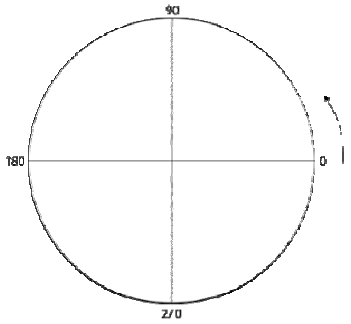
Force background update
Forces the background to update after XX milliseconds. This parameter is used to update the background regularly so that changes in the background (e.g. light conditions) will not be detected as road users. The parameter should be set as high as possible to ensure that road users arriving to the red light in an intersection will not be registered as a part of the background before the signal changes to green.

Active duration
The time interval in which a detector will be active (i.e. able to register movement) after it has been activated. If the detector is not activated or triggered anew, it will be deactivated after XX milliseconds.

When active, disable other detectors for
Discards irrelevant activity in the area of interest, e.g. movement in another direction. When the detector in question is triggered, it makes all other detectors inactive for XX milliseconds.

Minimum speed
Measure for how fast an object must move to be registered in a flow detector. The higher value, the faster it has to move.

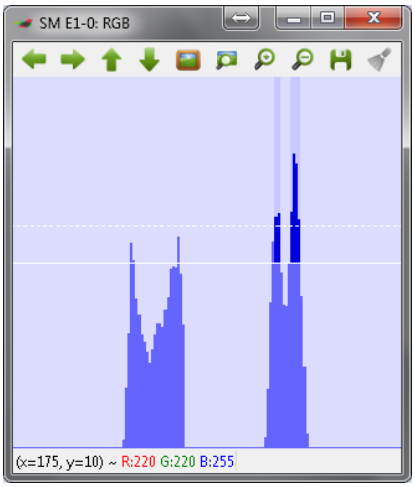
Flow range
Defines in which direction the vectors must go if the activity should be registered as an event. The range can be chosen on a circle (0-360 degrees). The range is chosen by either inserting the range in the fields or by dragging in the dots on the circles.



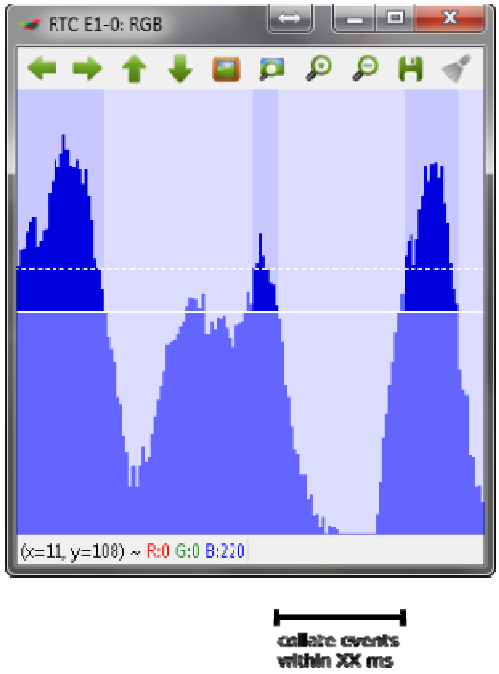
Max vector count
The maximum amount of vectors that is allowed to be above the defined minimum speed to result in an event.

Disable all detectors after
Used in modules with multiple detectors. Except for the first detector in the order, all detectors are made inactive if there has been no activity for XX milliseconds.

Delete events smaller than
Deletes events that are only detected briefly, which often indicates noise in the image. Events with duration less than XX milliseconds will be omitted from the log.



Collate events within
Combines separate events into one event if the time gap between them is less than XX milliseconds. This protects against multiple detections of the same object. If chosen too high, multiple road users driving close to each other will be registered as one road user.



Max triggered duration

Defines the maximum allowed duration (in milliseconds) of an event. If an event is longer than the specified maximum duration, it will be cut off after the max triggered duration has gone, and a new event will be created immediately thereafter.

Minimum delay / Maximum delay

The minimum and maximum acceptable time gap (in milliseconds) from the detection of activity in one detector to the detection of activity in another detector. *Minimum delay* is set to 0 if activity may occur at the same time in the two detectors.

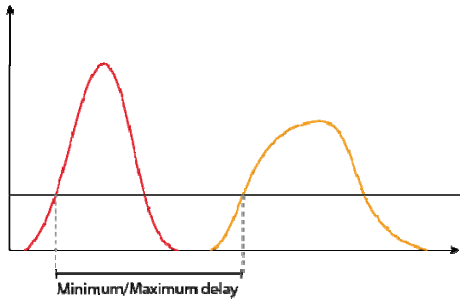


Figure 27 | The minimum and maximum delay are defined as the time from the beginning of the detection in the first detector (red) to the beginning of the detection in the second detector (orange)

Car before cyclist time gap

The maximum PET (Post-Encroachment Time) – given in seconds – that is accepted to register it as an interaction in which the vehicle leaves the intersection point first.

Cyclist before car time gap

The maximum PET (Post-Encroachment Time) – given in seconds – that is accepted to register it as an interaction in which the cyclist leaves the intersection point first.

5.4 Run the analysis

When the detectors have been calibrated the analysis can be performed (Figure 28). If it has not yet been specified which log files should be created during the analysis, this is done by double clicking the detector in ”Active detectors”. Table 3 gives an overview of the information in the log files.

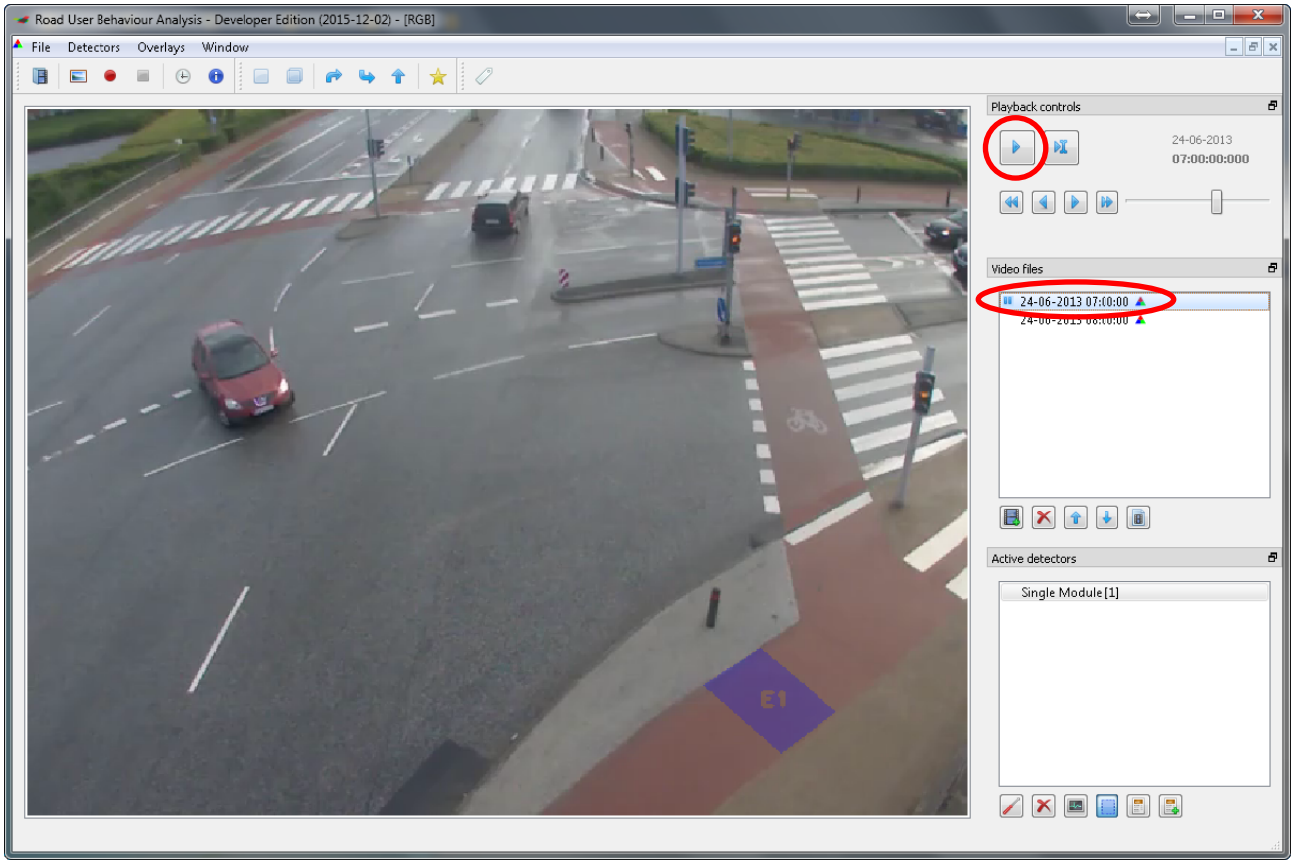


Figure 28 | Running a traffic analysis. Double click on the first video to scroll back to the beginning of the video, and then press the play button to run the analysis. The analysis is complete when the time stops in the end of the last video. Please note, that the time and date must be specified in the beginning of each video if the user defined time format is used

Table 3 | Contents in the log files. The files ”Analytics” are created when marking ”Log every event” in the settings/drawing window of the detector, while ”Counts” will be created when marking ”Log sum of events”. In the latter, the desired time interval can be specified (in minutes), e.g. 15 minutes

	Analytics				Counts	Description
	Special modules	**Interactions**	**SingleModule**	**DoubleModule**	**All**	
File	x	x	x	x	x	File name of the video
Date	x	x	x	x	x	Date for video recording
IsTriggeredTime	x					Time stamp for triggering the detector (activity above threshold have been registered)
Entering (CZ / - / Detector1 / Detector2)	x		x	x		Time stamp for arrival to the detector (i.e. there is activity in the detector)
Leaving (CZ / - / Detector1 / Detector2)	x		x	x		Time stamp for when the detector has been left (i.e. is empty again)
Timegap 1-2				x		Time difference between an object has triggered detector 1 and an object has triggered detector 2
Timegap DetectorX (1/2)				x		Time from one object arrives to detector X (trigger the detector) to the road user has left the detector
FirstTriggered				x		Which of the two detectors was triggered first?
Timestamp		x				Time stamp for the first road user’s arrival to the conflict area
TimeGapInMsec		x				Time gap from the first road user has left the conflict area until the second road user arrives to the conflict area
WhoIsFirst		x				Which road user is passing the intersection point first? (Car or Cyclist)
InteractionType		x				Type of interaction (312 = right turning car involved, 410 = left turning car involved)
TimeStart					x	Time for the beginning of the time interval
TimeEnd					x	Time for the end of the time interval
Object					x	Number of objects that have been detected within a specific time interval

References

Canny, John (1986): *A computational approach to edge detection.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 6, pp. 679-698.

Farnebäck, Gunnar (2003): "Two-Frame Motion Estimation Based on Polynomial Expansion" i *Image Analysis*, eds. Josef Bigun & Tomas Gustavsson, Springer Berlin Heidelberg, pp. 363-370.

Madsen, Tanja K. O., Bahnsen, Chris, Jørgensen, Anders, Moeslund, Thomas B., Olesen, Anne V. & Lahrmann, Harry (2014): *Krydsløsninger for cyklister - Anvendelse af konfliktteknik til vurdering af forskellige løsningers sikkerhed.* Trafikdage på Aalborg Universitet.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 635895

This publication reflects only the author's view. The European Commission is not responsible for any use that may be made of the information it contains.