

White Paper

# People Counting with iCAT





## **Copyright**

Copyright © 2003-2018 Netavis Software GmbH. All rights reserved.  
Netavis and Observer are trademarks of Netavis Software GmbH. All other trademarks are trademarks of their respective holders.

Netavis Software GmbH  
Handelskai 388, Top 221  
A-1020 Vienna  
Austria

Tel +43 (1) 503 1722  
Fax +43 (1) 503 1722 360  
[info@netavis.net](mailto:info@netavis.net)  
[www.netavis.net](http://www.netavis.net)

## Contents

<b>Introduction .....</b>	<b>4</b>
<b>1 The camera .....</b>	<b>5</b>
1.1 Choosing the camera .....	5
1.2 Mounting and adjusting the field of view .....	6
1.3 Useful camera settings.....	8
<b>2 Setting up the people counter .....</b>	<b>9</b>
2.1 Object tracking region.....	9
2.2 Event Trigger - Tripwires.....	17
<b>3 Evaluating the performance .....</b>	<b>21</b>
<b>4 iCAT people counter with multiple cameras.....</b>	<b>24</b>
<b>5 Additional resources .....</b>	<b>28</b>

## **Introduction**

Along with many other computer vision solutions, the efficiency of the iCAT people counter strongly depends on the position and field of view of the camera and the appropriate configuration of its parameters. Because of the wide range of environmental conditions it is used in and the complexity of the underlying algorithms, it is often not a trivial task to set up people counting in order to fulfill the customer's requirements (achieving as low a counting error as possible). The aim of this document is to provide a step-by-step guide for this important procedure, making it easily understandable with many practical examples, and revealing the most common pitfalls.

This document assumes basic knowledge of Netavis Observer and discusses the subject of people counting from a practical point of view. If you do not know how to use or find the mentioned UI controls of the Observer client, please refer to the Netavis Observer User Manual. In particular section 15 "Video analytics with iCAT" in the Observer User Manual provides complementary material to the information presented in this guide.

## 1 The camera

Selecting and positioning a camera to be suitable for people counting is a challenging task. It can be restricted by the structure of the building (e.g. the height of the ceiling), the existing wiring, and the budget of the installation. However these decisions have a very strong influence on the efficiency of the counter and a badly positioned camera can even make counting impossible.

### 1.1 Choosing the camera

The first step to take is choosing a camera that will provide the video stream which will be analyzed. All iCAT functionality including people counting is designed for use with stationary cameras. Pan-tilt-zoom (PTZ) cameras can also be used as stationary ones. However this solution is not recommended, except when all possibility to alter the camera view is disabled.

Another selection criterion is the quality of the video that the camera can provide. High quality cameras have a higher Signal to Noise Ratio (SNR) even in poorer lighting conditions. This is particularly important when the camera operates in a place where natural light sources are dominant (e.g. outside a building) and the brightness can vary widely. Figure 1 shows a very noisy image taken outside of a building, where people counting was working with poorer performance once the brightness fell below a certain level.



**Figure 1: Noisy camera image**

Even when the counter is applied in well and uniformly lit indoor scenarios, the choice of a higher quality camera is always a good decision. The reason is that a lower noise level increases the sensitivity of the algorithms inside iCAT.

As the color channels of a video stream are intensively used by the algorithms of the counter, it is highly recommended to use a camera capable of producing color video. Using a grayscale camera (or switching to grayscale mode) does not make counting impossible, but the performance will be lower in almost every case and the potential improvements of the algorithms in the future, which may rely on color features, won't be applicable.

The streaming format used by the camera is not a relevant selection criterion, because the people counter is able to process every video format supported by a Netavis Observer server.

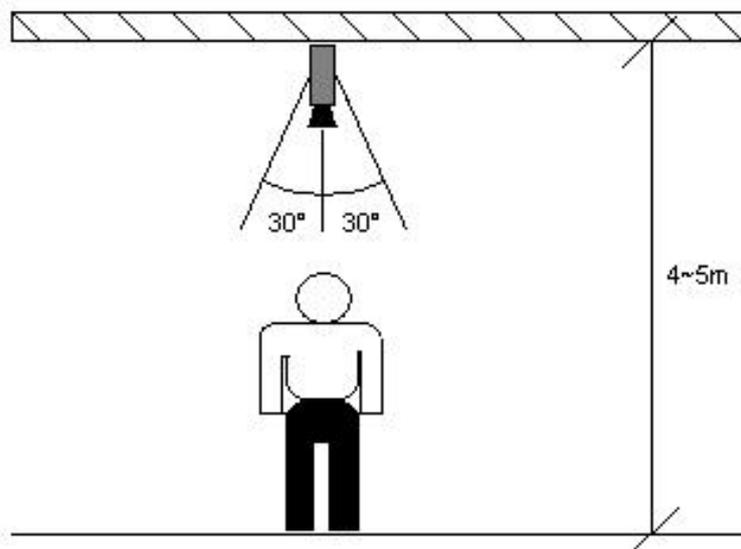
Sometimes cameras mounted outdoor do not have housings, so lenses get dirty easily or blurred by raindrops. This could significantly impact the performance of even the best setup.

## 1.2 Mounting and adjusting the field of view

iCAT people counter is designed for operation on a video stream provided by a downward looking camera. In this case the overlapping of the objects (people) is minimal. People moving close together do not cover each other, so it is easier to track and count them. With a downward looking camera it is also more obvious to find out the direction a person is moving to, making the definition of the counter lines (called „tripwires” in iCAT) easier.

With a heavily tilted camera a person getting closer or further away can only be seen as an object with growing or shrinking size. If one imagines the sequence of the previous and the current positions of the person connected with small line segments, it won't form a definite curve that can cross a tripwire. Hence it is very important to mount the camera as close as possible to a downward looking pose.

There is of course a tolerance range where the algorithm is able to work without difficulties. It is about 30° deviation in each direction from the downward looking pose. The distance of the camera to the persons is also very important, though it is often determined by the height of the ceiling. The ideal height is about 4~5 meters, but it can vary depending on the type of the camera. The main point is to mount the camera so that the people on the images are neither too small (camera too high) nor too big (camera too low). People who are too far away from the camera are difficult to detect and those too close to the camera move across the scene (field of view) quickly, covering a large part of the image, and thereby making the counting less efficient. Figure 2 shows a drawing of the ideal camera position, and Figure 3 shows an example how it looks like in a real scenario.



**Figure 2: Ideal camera position**



**Figure 3: Example from a shop**

The object tracking algorithm in the iCAT people counter requires the objects (people) to be in the camera view long enough and make significant movements there. When positioning the camera, try to maximize the relevant area of the field of view, where people are possibly moving. In practice this means that no walls, elevators and other objects should be in the camera view. Rather the camera should be positioned so that more of the ground/floor, where people may appear, is visible. When there is enough space for the people to move in the scene, their route will be longer and the chance of recapturing a person lost by the counter will be higher.

The counting process is the following: detecting, validating, tracking, and then counting the object if it crosses the tripwire. The detection and validation steps require a few frames. All detected objects have to pass a validation in order to filter out false trackings (e.g. doors opening, moving shadows). If there is not enough space in the scene, this validation cannot always happen before the object passes the tripwire, and so it won't be counted. The left images in Figures 4 and 5 show two bad camera views and the right images show their corrected versions.



**Figure 4: A bad (left) and a good (right) camera view in a shop**



**Figure 5: A bad (left) and a good (right) camera view of an escalator**

In the good views there is more space for people to be detected and tracked. Here the cameras were tilted to be closer to the downward looking pose. Note that in the right image of Figure 5 less of the escalators are visible. Continuously moving parts of the image result in the background model of the tracker becoming insensitive and thereby making these areas useless for tracking.

It is also essential not to have any disturbing objects in the camera view which cover a precious part of the scene where people are moving.



**Figure 6: Object obstructing the camera's view**

Figure 6 shows such a situation, where the drop ceiling (black on the right side) is splitting the view of the camera in two parts. Here, tracked objects were usually lost.

### **1.3 Useful camera settings**

There are three camera settings that may be useful when setting up iCAT people counter.

The first is the exposure control to avoid flickering video. It can have a slight impact on the sensitivity of the background model used by the counter.



**Figure 7: Settings on an Axis camera's Web page**

Secondly, if a PTZ camera (fixed as though it would be stationary camera) is used to count people, it is useful to switch off autofocus. Some cameras adjust their focus continuously when objects enter or leave the scene, which leads to the images being blurred and then sharp again. This results in the video appearing to have a low level noise superimposed on it which decreases the sensitivity.

The third camera option, which is highly relevant for the CPU load caused by iCAT, is multi-streaming. For iCAT to work it must first decode the video stream from the camera. With the H.264 format this decoding process requires significant CPU resources. If a camera supports multi-streaming and can provide MJPEG, it is possible to use an H.264 stream for live viewing and archiving while relying on the MJPEG stream for iCAT. This reduces the overall CPU load needed for operation (at the cost of a slight increase in the required bandwidth) and can therefore increase the number of iCAT-enabled cameras which can be connected to a server.

**Summary:** (1) Choose a stationary color-camera with a high SNR (2) Mount the camera in a downward looking fashion (or with a maximum deviation of 30°), ideally at a height of 4~5m. Maximize the relevant area where people are moving in the camera's field of view. (3) Configure the camera for flicker-free operation, turn off auto-focus, and (if available) configure multi-streaming.

## 2 Setting up the people counter

This part of the setup process has at least the same importance as the selection and mounting of the camera. It consists of three steps and is usually done in an iterative fashion.

First one has to create an **Object tracking region** and set all the parameters according to the scenario (section 2.1). Then at least one **Event Trigger** has to be defined to carry out counting (section 2.2). The last step consists of testing the result (e.g. watching what the counter does in Object Monitor (OM) for a while) and modifying the settings of the previous two steps until no further improvements can be observed.

### 2.1 Object tracking region

#### 2.1.1 Defining a region

To create an *Object tracking region* select a camera, right-click in the **Video Analysis (iCAT)** tab (see Figure 10), and select **Add new definition** or choose **Add new definition** in the **iCAT definition** menu. Then from the drop-down list next to **Type** (in the middle of the tab) select **Object tracking region** and enter an **Identifier** for the region.

After that a geometric definition of the region has to be drawn on the small camera view in the top-right corner. Alternatively the default, which is the whole image, can be used. There are two possibilities to define the region: **Rectangle** or **Polygon** which can be selected by checking the corresponding checkbox. Drawing a polygon is much more flexible than drawing a rectangle, hence it is used more frequently. This allows for as many vertices (a vertex being a connection point of two lines with different angles) and lines to be drawn as needed. To close the polygon, cross an existing edge or double click with the mouse. It is also possible to modify a polygon by dragging the vertices, however if more vertices are needed a new polygon has to be drawn. The main role of this region is to determine the relevant area for tracking and subsequently people counting. All algorithmic processing is carried out only on this region, parts of the image outside the polygon/rectangle are ignored. This makes it possible to mask out irrelevant areas where people can never appear (e.g. the upper part of a wall), as well as problematic areas such as moving door wings or people moving outside the building who are visible through a window. Excluding parts of the image for processing also means a lower computational load. Thus, masking out irrelevant areas has a positive effect on the CPU load.

When trying to save CPU load and excluding walls, one should be aware not to mask out areas where people could possibly move. As mentioned earlier in the camera mounting section, it is crucial to let the objects (people) have enough space to be tracked (and perhaps be recaptured by the tracker). A good practice is to include all areas where a person can appear in the Object tracking region.

This is not always obvious because of the larger viewing angle of some cameras. This means that close to the edges of the image people cannot be seen from an overhead perspective but rather at a certain angle. Hence, it is not always sufficient to define the region to just fit the contour of the floor (assuming that this is the place where people move). Rather the lower part of the walls (or other surrounding objects) might have to be included as well to make the full body of a person moving close to the wall visible for the tracker.

Figure 8 and Figure 9 show two properly set region definitions. In Figure 8 the black stripe with the camera information text is excluded and the edge of the region on the lower side of the image is slightly moved inwards to reduce the disturbing effect of the shadows. Note that the edges cannot be moved further inwards here. Although it would eliminate the shadows entirely, it would mean that the counting lines (tripwires) would also have to be moved towards the center of the image (explanation in the next section). In doing so, the possibility that a person crossing the tripwire is really leaving is reduced.

In Figure 9 the region polygon does not contain the wall in the top-right corner as it is an irrelevant area as well as the shelves on the upper-left, because clothes there are often swinging and thereby cause false detections. Additionally both automatic doors are masked out. One may ask why the platform in the middle is included in the region. This obstacle divides the entrance area into two corridors leading to larger groups of people coming in being split up. Placing the tripwire (red) for incomers in front of these corridors makes it possible to count the members of a larger group separately (more about tripwires in the next section).



**Figure 8: Proper object tracking region #1**



**Figure 9: Proper object tracking region #2**

### **2.1.2 Parameters**

When the region polygon has been drawn, one can go on with adjusting its parameters. Figure 10 shows the entire **Video Analysis (iCAT)** tab with all of the adjustable parameters.

#### **Indoor camera**

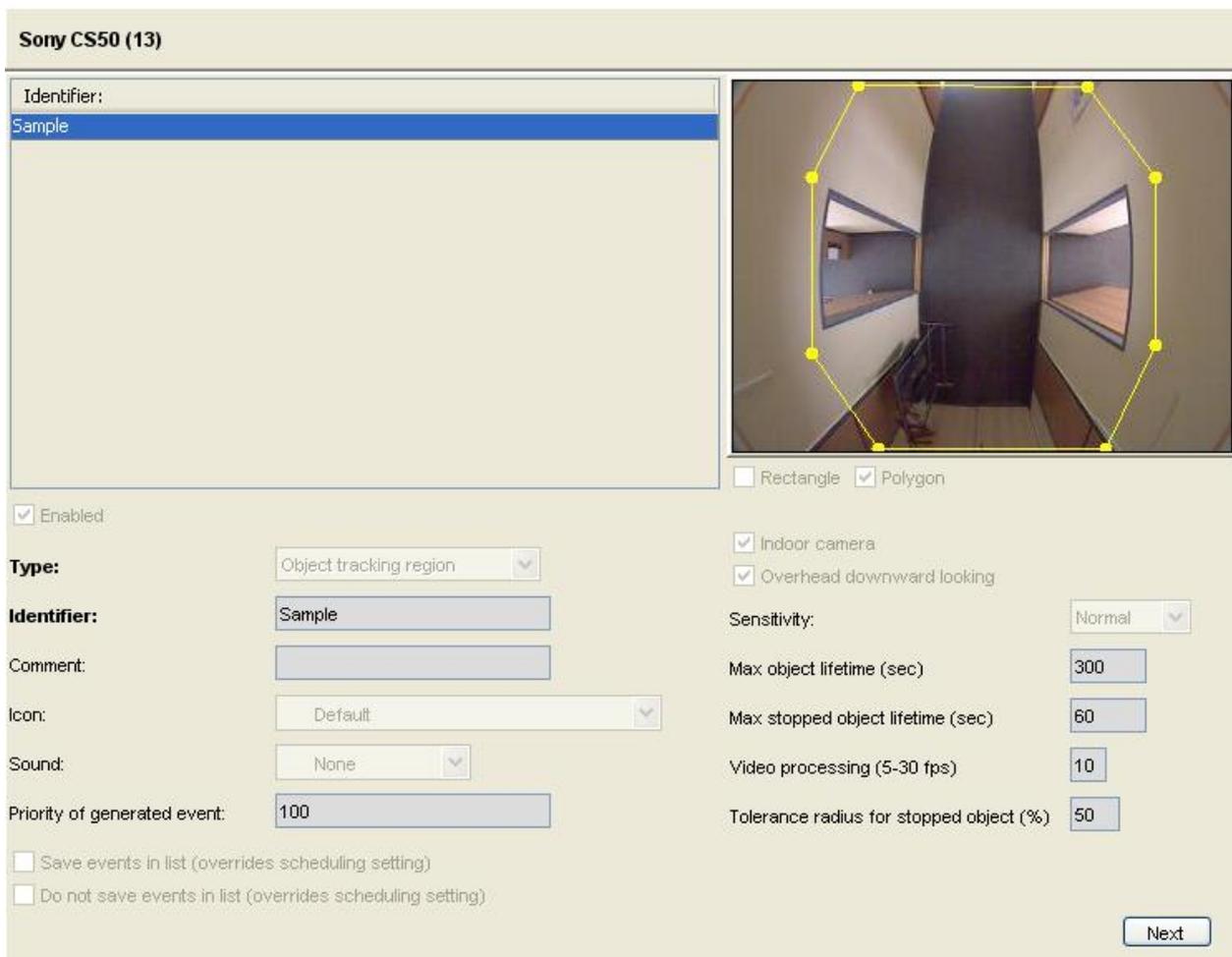
First of all one has to select the tracker's operation mode. The two possible choices are indoor (if the **Indoor camera** box is checked) and outdoor (when it is unchecked). The two modes use different predefined parameter sets which can be further adjusted, for example by changing the **Sensitivity** option.

These parameter sets reflect assumptions about the size of the tracked objects, their motion, and environmental conditions:

- Indoor mode assumes stable lighting conditions and large or medium-sized objects (people being closer to the camera). It also expects a less uniform motion from the objects, as people can suddenly change their speed and direction.
- Outdoor mode assumes smaller objects being further away from the camera. It is also set up to cope with distracting environmental conditions like swaying trees and moving shadows.

In the vast majority of people counter setups, the indoor mode is recommended even when the camera is actually mounted outside a building (for example such as in Figure 1). In such cases indoor mode is a reasonable choice because people are closer to the camera and the environmental conditions are fairly stable (there are no swaying trees, the place is not fully open, etc.).

Outdoor mode only has to be chosen if the people are far away from the camera and move in a real outdoor environment. An example would be the observation of pedestrians on a pavement from a camera installed on the roof of a building.



**Sony CS50 (13)**

Identifier:  
Sample

Enabled

**Type:** Object tracking region

**Identifier:** Sample

Comment:

Icon: Default

Sound: None

Priority of generated event: 100

Save events in list (overrides scheduling setting)  
 Do not save events in list (overrides scheduling setting)

Rectangle  Polygon

Indoor camera  
 Overhead downward looking

Sensitivity: Normal

Max object lifetime (sec): 300

Max stopped object lifetime (sec): 60

Video processing (5-30 fps): 10

Tolerance radius for stopped object (%): 50

Next

**Figure 10: The complete Video Analysis (iCAT) tab**

## Overhead downward looking

Next, it has to be decided whether the camera is downward looking or not. If this is the case, the **Overhead downward looking** box has to be checked. Selecting this option uses the predefined indoor parameter set and enables an estimator module.

This module estimates the average size of a person across the entire image. In general when two or three people move together as a group the tracker tracks and subsequently counts them as one object (they are in one bounding box). However if the average size of one person is known on all areas of the image, it is possible to estimate the number of people in such groups and thereby significantly improve the accuracy of the counting result. So in practice, when the estimator is enabled, groups of people are taken into account with their estimated participant number rather than as only a single person.

A large deviation from the downward looking direction makes the estimator less reliable. With a heavily tilted camera it is possible that a single person will be counted twice or even more often. Hence when a camera has an angle of 30° or more (see Figure 2) it is better not to check **Overhead downward looking**. The exact range in which the estimator module becomes unreliable also depends on the camera lens.

One may experiment with the estimator by enabling it and letting it run for some time (1 hour ~ 1 day, the more people pass by, the shorter this interval has to be). Then the real number of the people moving together can be compared to the number of people calculated by the tracker using the **Object markers** in the **Online Monitor (Online monitor → View control → iCAT → iCAT info display → Object markers → Show always)**. If the counter estimates that a group of people consists of more than one person this is shown in the Object marker as, for example, "(2 objects)".

## Sensitivity

As mentioned above, selecting the indoor operation mode selects a predefined parameter set for the counter. One can modify a given subset of these parameters by setting the **Sensitivity**. There are five sensitivity levels ranging from **Very low** to **Very high**. While in most cases the **Normal** (default) level is more than enough, it could be useful to understand the impacts of modifying sensitivity.

Table 1 shows a comparison table including the five possible levels and the parameters they affect in the two most frequent counting scenario, indoor camera and **indoor camera + overhead downward looking**. Below the table there is also a short explanation of every parameter.

	Common parameters (for both Indoor camera and Indoor camera + Overhead downward looking)			Indoor camera	Indoor camera + overhead downward looking
Sensitivity	Shadow removal <sup>(1)</sup>	Motion uniformity <sup>(2)</sup>	Object validation time <sup>(3)</sup>	Size filter <sup>(4)</sup>	Size filter
Very low	Yes	0.6	very long	1.6	2.7
Low	Yes	0.5	long	1.3	2.2

Normal	Yes	0.4	normal	1.0	1.6
High	Yes	0.3	short	0.6	1.1
Very high	No	0.2	very short	0.3	0.55

**1. Shadow removal:** In most lighting conditions, objects (people) have a shadow moving with them. This could connect two objects when they are moving close to each other. The shadow removal module tries to identify shadow regions and removes them.

**2. Motion uniformity:** New objects (people) have to show a certain smoothness of motion to be validated and tracked in order to filter out noisy objects which usually have an irregular motion pattern. The measure of uniformity can range from 0 to 1 whereby 0 means no validation and towards 1 the validation criterion becomes stricter.

**3. Object validation time:** The time interval during which a new object's validation is carried out. While being validated, the object has to show a smooth motion and must not disappear.

**4. Size filter:** It filters out small noisy objects. Every object below a certain size will be regarded as noise and will be removed. To allow comparisons the size of the indoor camera setting at normal grade was chosen to be 1.0. In practice, this size is much smaller than the expected object size but large enough to efficiently filter out false detections.

Table 1: Overview of the parameters for the Object tracking region

This table may initially look overwhelming, but it is simpler than it seems. The emphasis is on the tendencies and not on the values itself. When the levels are changed from very high to very low in a step-by-step manner, the newly appearing objects have to fit stricter conditions to be tracked and stronger noise filtering is applied.

In practice when several false tracks are experienced (by watching the counter in the Online Monitor) setting sensitivity a grade lower could be a solution. In case people frequently seem to be recognized too late by the tracker (this could have another reason as well, see **Video processing** below) or are lost turning the sensitivity higher could help. Using the highest and the lowest levels is only recommended in extreme situations (very low – in a very noisy environment, very high – if one wants to detect and track everything and possible false detections are not a problem).

### Max object lifetime / Max stopped object lifetime

These two parameters are time limits measured in seconds.

The first parameter sets the maximum time during which a person is followed by the counter. People having a longer lifecycle than that will be deleted and perhaps recaptured again, if they show a definite motion.

The second one determines the time after which objects remaining still are deleted. The goal of it is to filter out false but tracked objects usually caused by large illumination changes, which often remain still somewhere in the image.

### Video processing

Here it is possible to set the input frame rate of the counter and the optimal frame rate is 15 fps.

It is important to know that the counter checks every new person in the scene for at least 5 frames. Suppose that 5 fps means at least 1 second then sometimes this could be too long for counting. To avoid that just set the frame rate higher. Taking more frames per second means that the consecutive positions of the objects (people) are closer to each other and so they are also more correlated. This makes it easier for the tracker to assign the objects together from frame to frame leading to more accurate counting.

### **Tolerance radius for stopped object**

This parameter influences the visual statistics of stopped objects (see **Visual statistics** below). When set to 100 per cent, the people remaining inside the boundaries of their bounding box longer than a predefined duration (2 seconds), will be labeled as stopped and added to the **Object stopping time** and **Stopped object count** statistics. Set it higher than 100 to gather statistics about loitering people and set it to less than 100 to include people who are almost or fully stopped.

### **Visual statistics**

Clicking the **Next** button in the bottom-right corner of the tab (see Figure 10) leads to a slightly different page. Here various types of visual statistics can be switched on or off (see Figure 11). The statistics module collects certain data about the tracked people and accumulates them on a coarser grid over the region. The collected data can be displayed in **Online Monitor** or in **Archive Playback**.

**Sony CS50 (13)**

Identifier:

Enabled

Type:

Identifier:

Comment:

Icon:

Sound:

Priority of generated event:

Save events in list (overrides scheduling setting)

Do not save events in list (overrides scheduling setting)

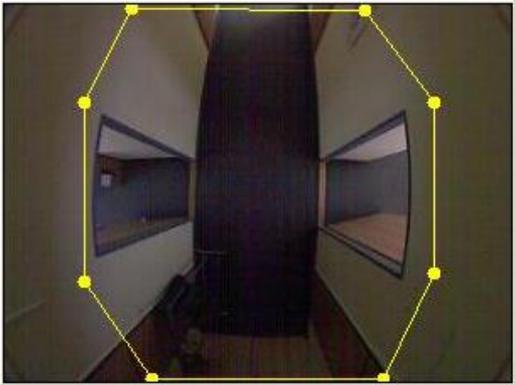
Count statistics

Speed statistics

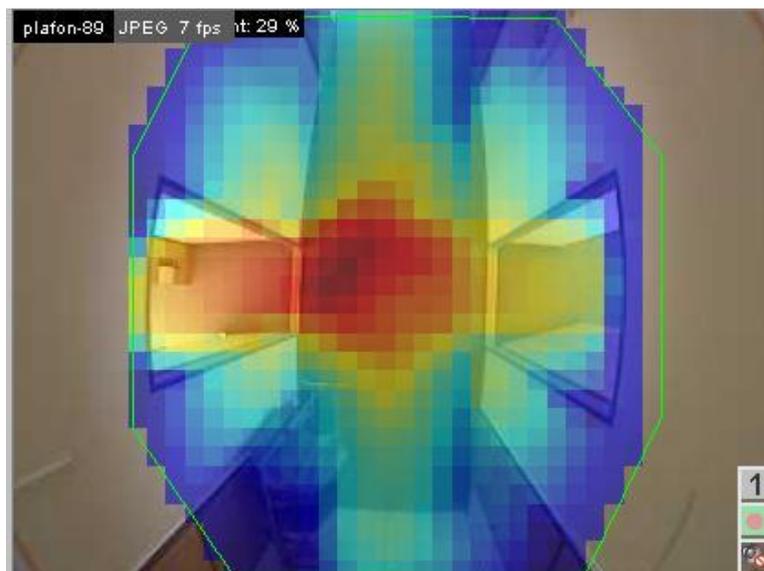
Stopped object count statistics

Object stopping time statistics

Size statistics



**Figure 11: Types of visual statistics**



**Figure 12: Visualization**

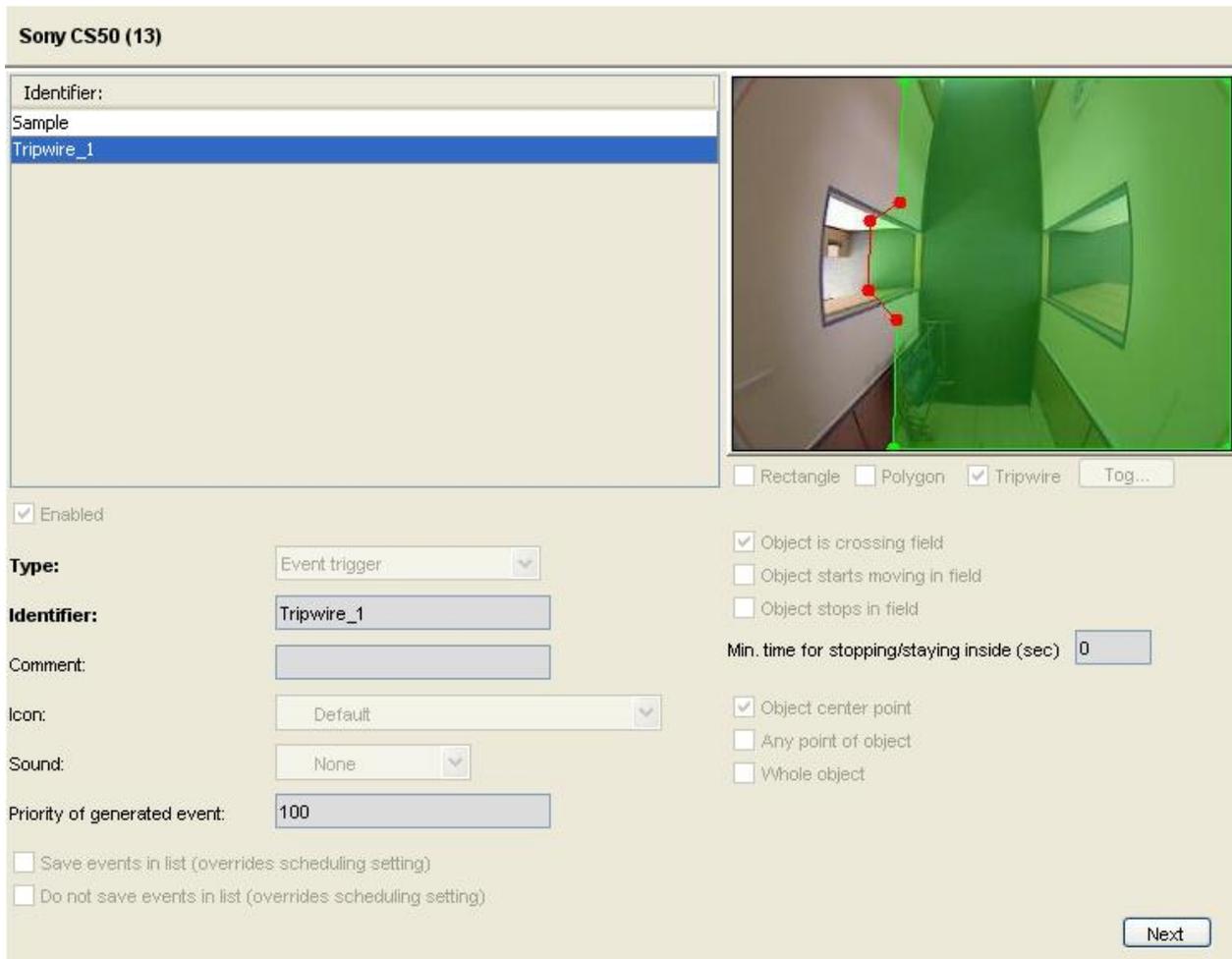
The visualization looks like a grid of small rectangles with different colors. Lower values are closer to blue, higher ones are closer to red (Figure 12). Among other things this color representation is able to reveal the most frequently used paths, the places where people usually spend more time or the average velocity of the objects. Depending on the number of people passing by it can take some hours or possibly even days before enough data is collected in order to produce meaningful statistics.

## **2.2 Event Trigger - Tripwires**

### **2.2.1 Defining tripwires**

Once you have properly defined an **Object tracking region** and its parameter set, you have a tracker that is able to track people in the camera view, but no corresponding counter yet. Adding a special kind of **Event trigger** called the **tripwire** to your existing setup completes the people counter. The tripwire is a polyline that counts people whenever they cross it. It is possible to add as many tripwires as you need and all of them will individually count the people who cross it. Besides being a polyline, a tripwire also determines a field („green field”) and only objects which are inside this field, as well as inside the previously defined **Object tracking region**, will be counted. Any other people detected outside the „green field” will be ignored. The goal of this functionality is to allow a finer filtering of people.

To create an **Event trigger** you can either right-click in the iCAT tab and choose **Add new definition** or select **Add new definition** from the main menu. Then select **Event trigger** from the **Type** dropdown menu and enter an **Identifier** for the trigger.



**Figure 13: A tripwire definition**

Check the **Tripwire** checkbox and draw the tripwire on the small camera view in the top-right corner. The drawing process is similar to the region definition (see section 2.1.1), though the polyline does not have to be closed. Double-click to finalize the tripwire.

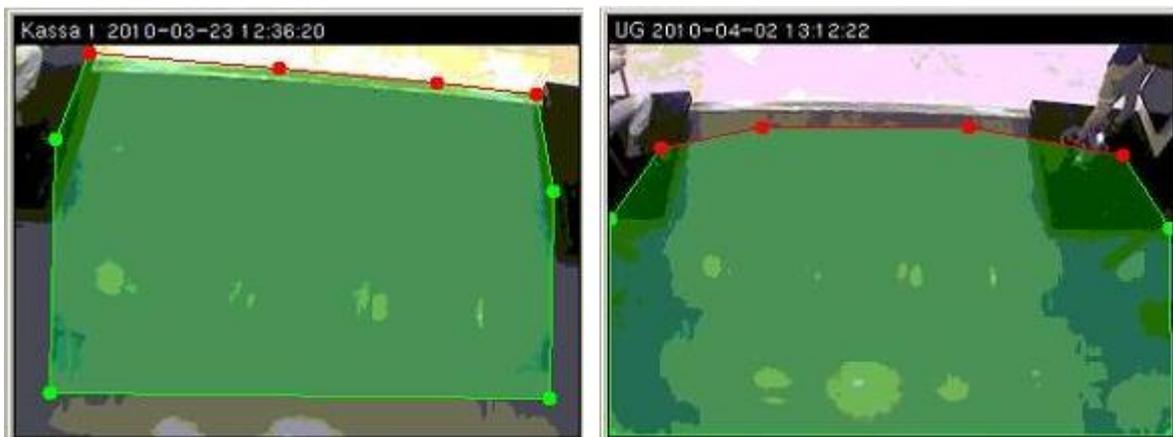
A green field will appear and split the image into two parts. With the **Toggle** button the area which the field covers can be inverted. It is also possible to modify an existing tripwire and its „green field” by dragging their vertices. Below the camera view it is possible to select which part of the object has to cross the line in order to be counted. In almost every people counter scenario, the first option (**Object center point**) is used because of its robustness.

Even this seemingly straightforward process presents its own challenges and a wrongly defined tripwire can make counting impossible. A common mistake is to place the tripwire close to the edges of the region or the edges of the image (if it is coincident with the region border) and to other areas where objects can disappear. In such areas an object’s center point cannot cross the tripwire before it disappears (this is due to the underlying background model however an in-depth explanation is beyond the scope of this guide). As a rule of thumb the tripwire should be drawn at least three quarters of the object’s size away from such areas. For the object’s size one may select either the width or height (the one which is closer to the general moving direction of the people) or simply the larger one of the two.

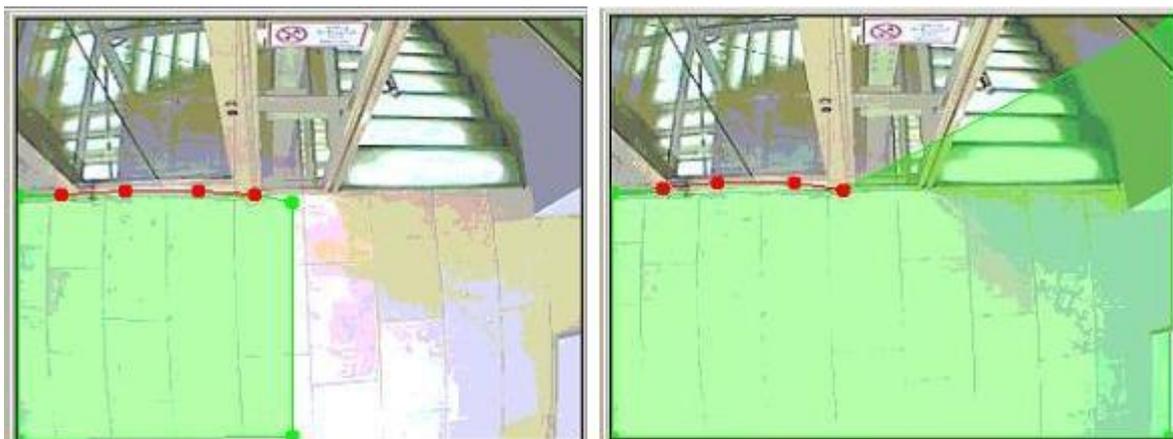
Similarly care needs to be taken when adjusting the green field. Remember every person detected outside of it will be ignored by the tripwire. Imagine the situation where the field is close to the side of the image but leaves a small stripe uncovered. People who enter the scene on this side of the image and are detected in the small uncovered stripe won't be counted even when they cross the tripwire. If only every fifth person is detected on the stripe then the error rate would be 20%. This has nothing to do with weaknesses of the tracking algorithms but is simply due to an improper tripwire definition. Hence it is a good practice to initially set up the field to cover all areas where people who might possibly cross the tripwire can be detected. Later it can be modified when people who should not be included are counted.

It can also happen that the tripwire is not long enough and people can slip through next to it. Such cases can be uncovered by checking the operation of the counter in the Online Monitor.

Figure 14 (left) shows an example of a tripwire which was placed too close to the edge of the image (actually to the camera's info text bar, but the result is the same) and the version on the right is its corrected counterpart. Here the „green field” was also expanded.



**Figure 14: Example of a tripwire that is placed too close to the edge (left) and the corrected version (right)**

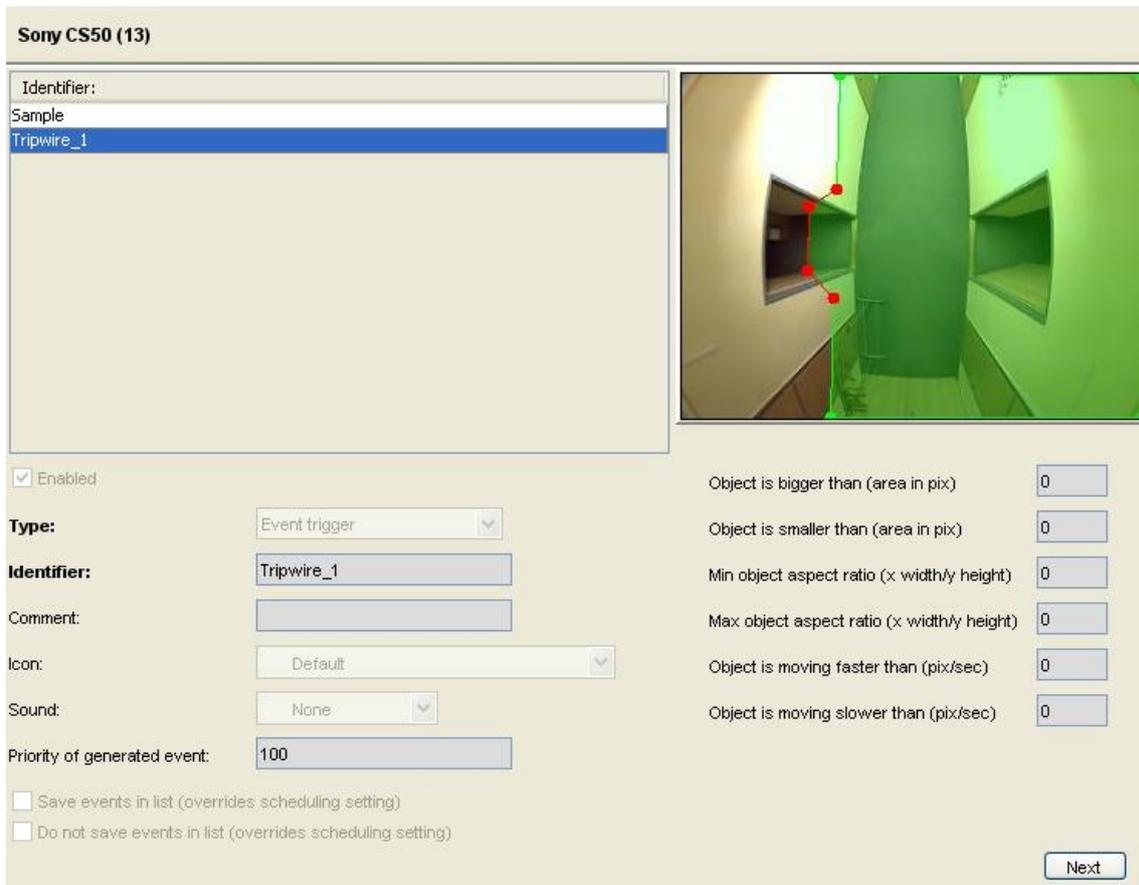


**Figure 15: Example of the green field being too small (left) and the corrected version (right)**

Figure 15 (left) is a typical example of the green field being too small. People entering from the corridor on the right side and getting into the elevator are not counted, because they are detected long before they enter the green field. The image on the right shows the corrected field.

## 2.2.2 Parameters

Filtering parameters make the definition of a tripwire more flexible (Figure 16). People can be filtered by their area in the pictures, their aspect ratio, and their speed.



**Sony CS50 (13)**

Identifier:  
 Sample  
 Tripwire\_1

Enabled

**Type:** Event trigger

**Identifier:** Tripwire\_1

**Comment:**

**Icon:** Default

**Sound:** None

**Priority of generated event:** 100

Save events in list (overrides scheduling setting)  
 Do not save events in list (overrides scheduling setting)

Object is bigger than (area in pix) 0

Object is smaller than (area in pix) 0

Min object aspect ratio (x width/y height) 0

Max object aspect ratio (x width/y height) 0

Object is moving faster than (pix/sec) 0

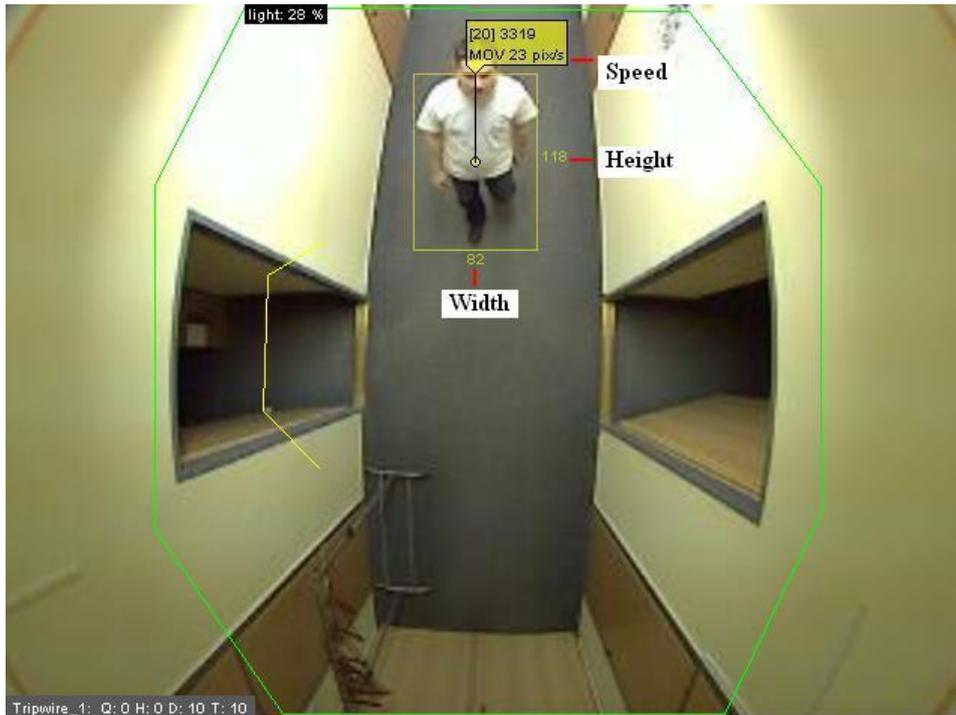
Object is moving slower than (pix/sec) 0

Next

**Figure 16: Tripwire configuration**

The appropriate values to be set for these options can be found in the Online Monitor. Switch on rendering of **Object marker** and **Object bounding boxes** in the online view of your camera (**Right-click in the camera view → iCAT info display → Object markers / Object bounding boxes → Show always**).

Object markers contain the speed of the objects in pixel/seconds when they are moving (MOV). Bounding boxes have their width and height in pixels next to the corresponding sides (Figure 17). The width and height can be used to calculate the aspect ratio. The area of the object is the second value next to object ID (the first number in square bracket) in the first line of the marker.



**Figure 17: Additional tracking information**

**Summary:** (1) Define the Object tracking region that determines the area where people are tracked. Use the “Indoor” and “Overhead download looking” presets to configure the region or manually set the parameters. (2) Define the tripwires and associated “green field” responsible for the counting. Ensure that the tripwire is not too close to the edges of the region or image. Adjust parameters as needed.

### 3 Evaluating the performance

In many cases it is necessary to evaluate the performance of the people counter. One may report this measure to the customer or use it to figure out whether a modified setup is better than a previous one. Hence such a performance measure should be as meaningful and accurate as possible. Depending on the customer’s specific requirements and on the scenario, such a measure can take various forms. Below a generally applicable one is presented whose results can be used as a foundation to form further measures with different meanings.

Every tripwire has its own people count and measures the directional traffic crossing it. Even on the same image the environmental and traffic conditions are slightly different for each tripwire. Hence it is reasonable to have a measure for every tripwire.

Expressing the performance with an error rate (or error percentage) rather than with a hitrate allows for handling both the cases when the tracker counts more or fewer people than actually walk past it. For example, if 15 people are counted even though only 13 crossed the tripwire then the hitrate would be  $15/13 = 1.15$ . However it would be odd to talk about a 115% hitrate. It is better to say there is a 15% error (false positives). Similarly there is a 10% error (false negatives) if 9 out of 10 people who actually crossed

the line are counted (9/10 = 0.9). In practice one can say that the people counter is working with a 10% error in both negative and positive direction.

The definition of error rate:

$$\mathbf{error\ rate} = \left| 1 - \frac{\mathbf{iCATcount}}{\mathbf{realcount}} \right|$$

$$\mathbf{error\ percentage} = \mathbf{error\ rate} * \mathbf{100}$$

iCATcount – number of people counted by iCAT

realcount – number of people who actually walked across the tripwire

In order to be expressive, one should calculate the error rate of all possible situations the counter can face. In a real environment, the lighting, traffic and other environmental conditions vary over time. Usually these factors show a daily periodicity so choosing a day as an interval for evaluation is often a good decision.

Sometimes there is no need to examine a whole day, for example in a shop the opening hours are sufficient. This interval is still too long to check every minute of it, but it is possible to approximate an all-day estimate by selecting short periods all along the day. After computing the error rate of all selected periods they can be combined with a weighted average to obtain the final result. Weighting is done to improve accuracy since an error value based on counting 10 people is not as relevant as one obtained after counting 50 people. The people who actually walked over the tripwire can be used as weights.

The weighted average of error rates:

$$\mathbf{average\ error\ rate} = \frac{(\mathbf{error}_1 * \mathbf{realcount}_1 + \mathbf{error}_2 * \mathbf{realcount}_2 + \dots + \mathbf{error}_n * \mathbf{realcount}_n)}{(\mathbf{realcount}_1 + \mathbf{realcount}_2 + \dots + \mathbf{realcount}_n)}$$

$$\mathbf{average\ error\ percentage} = \mathbf{average\ error\ rate} * \mathbf{100}$$

error\_n – error rate of a period

realcount\_n – the actual number of people in a period

### Example

Let's see how this can be done in practice. First a sufficiently long (~1 day) archive recording is required from which the intervals will be selected. Then the actual number of people can be counted while playing back the intervals in the archive view. Unfortunately this is time consuming however the process can be shortened by accelerating the playback (**Archive view** → **Player tab** → **Playback Acceleration** slider). iCAT people count can be displayed on the images by switching on **Event count fields (Archive view** → **iCAT button** → **iCAT info display** → **Event count fields**).

To obtain the iCAT count for the period, subtract the value at the beginning of the interval from the value at the end of it. Table 2 below shows example counts, error rate and weights measured on 15 minutes periods.

Period	iCAT count	Real count	Error rate	Weight (real count)
10:00 – 10:15	90	95	0.052	95
11:15 – 11:30	102	90	0.133	90
12:30 – 12:45	54	61	0.114	61
13:45 – 13:59	12	11	0.09	11
...	...	...	...	...

**Table 2: Example scenario**

Using the values in the table above, the average weighted error percentage is calculated in the following way:

$$\text{average error rate} = \frac{((95 * 0.052) + (90 * 0.133) + (61 * 0.114) + (11 * 0.09))}{(95 + 90 + 61 + 11)}$$

$$\text{average error rate} = \frac{24.854}{257} = \pm 0.0967$$

$$\text{average error percentage} = 0.0967 * 100 = \pm 9.67\%$$

The method discussed above is an accurate but a time consuming way of evaluation. When it is representative (conditions are stable during the day) one can select fewer intervals as well. To tune the algorithm it is sufficient to use only one, but longer interval. If there is not enough time, checking what the counter does in the Online Monitor and then modifying the counting setup if required is also a possibility. This way no overall performance measure is obtained, but it is possible to quickly tune the setup.

A typical scenario for the counter is when it counts the incoming and outgoing number of people at the entrances of a building. Many assume that the counter is performing well when the incoming and outgoing number is almost identical during a certain timeframe when few people are in the building (typically early in the morning or late in the night). While it sounds logically care must be taken when interpreting such a result. Imagine a case where the counter only counts half of the incoming and similarly half of the outgoing people. The two values will be close to each other at the end of the day but the results will have little to do with the reality! Hence, this comparison cannot be used as evidence of a correct operation. However finding a big difference between incoming and outgoing numbers could be indicative of a serious problem (assuming everybody who enters also leaves through the same entrance).

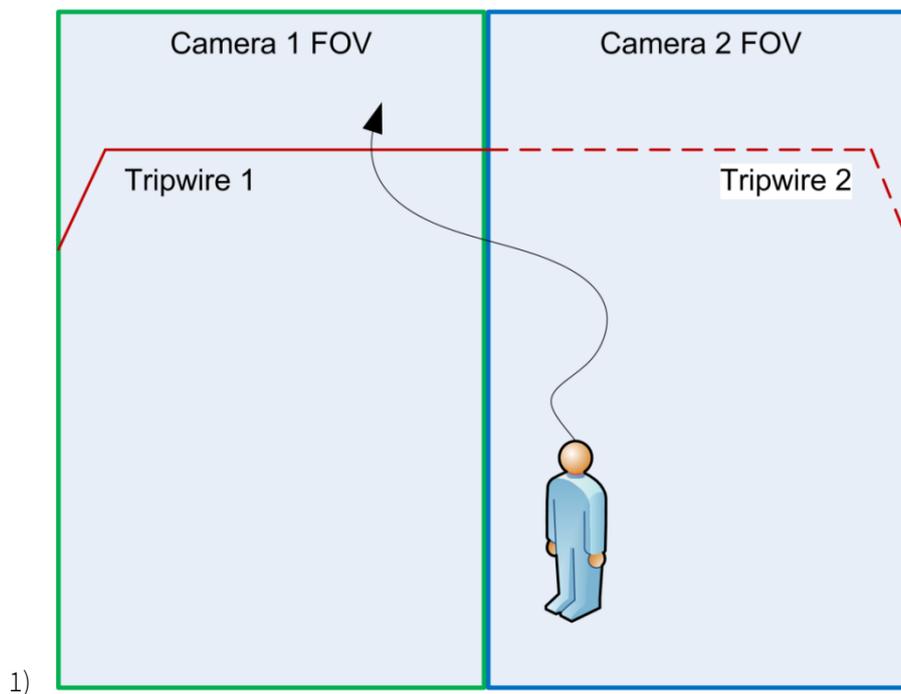
**Summary:** (1) Evaluate the performance of the setup by comparing the results of the people counter with the real number of people in the area. Calculate the error rate, based on as many different situations as possible. (2) Tune the setup and configuration to iteratively minimize the error rate.

#### 4 iCAT people counter with multiple cameras

Typically counting at entrances could only be carried out by using a single camera. However there are situations when the entrance is far larger and multiple cameras have to be used. In this case the counting is running in parallel on multiple cameras and each camera measures a different part of the traffic. When more than one camera cover the area where people come in or go out, people may exit from one camera's view and enter into another camera's view. This kind of motion across camera views requires special care when setting up camera views and defining counting lines.

##### The problems which arise when cross view motion is possible:

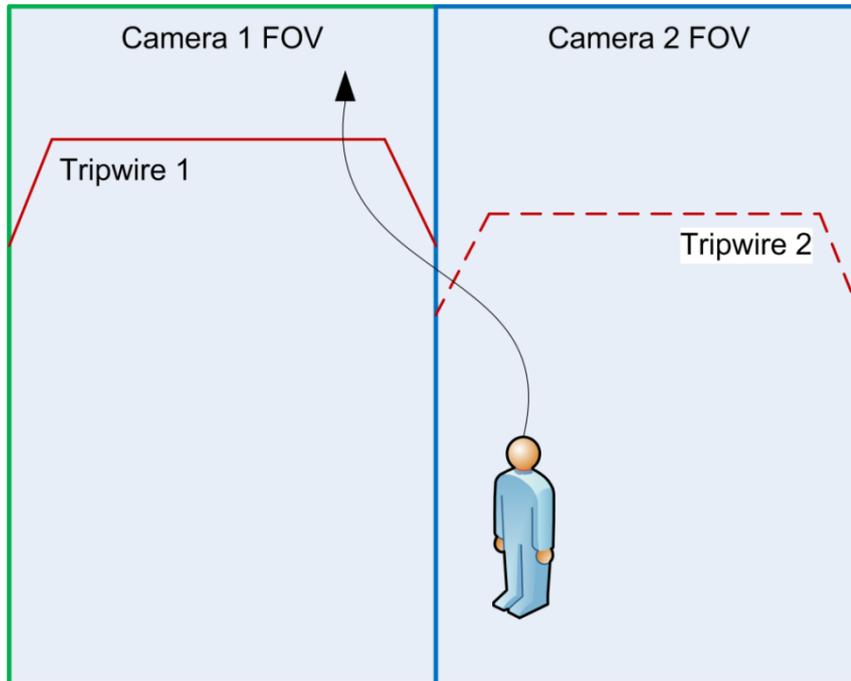
1. Target person enters the view (from another camera's view) just before the counting line:



**Figure 18: Person enters the view just before the counting line**

Figure 18 above shows two adjacent cameras' fields of view (FOVs), and a person who crosses from camera 2's view to camera 1's view. The problem here is that the time the person spends in camera 1's view might not be long enough to detect him reliably before the tripwire, so it won't be counted. If the person could have been detected earlier it would have been also counted.

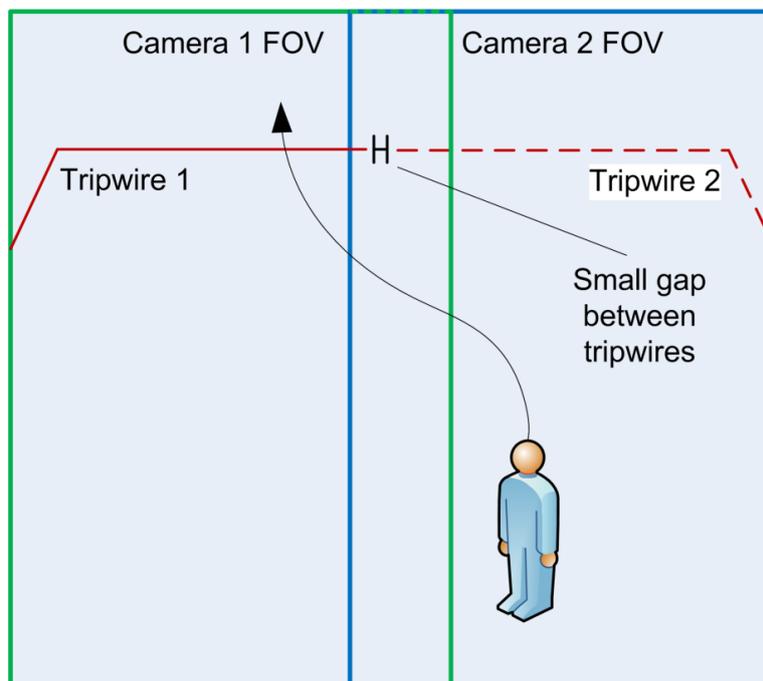
- 2) 2. Target person crosses more than one counting line:
- 3) The problem shown in Figure 19 is that the tripwires are not aligned with each other. This layout runs the risk that the target person can cross both lines, one after another, and will be counted twice.



**Figure 19: Target person crosses more than one counting line**

**The solution:**

The basic idea is to stitch camera views together, slightly overlapping on one side, as if they were one big counting view. The tripwires should also look like they are parts of one big virtual tripwire, though a small gap should be left between the adjacent ends of the lines.



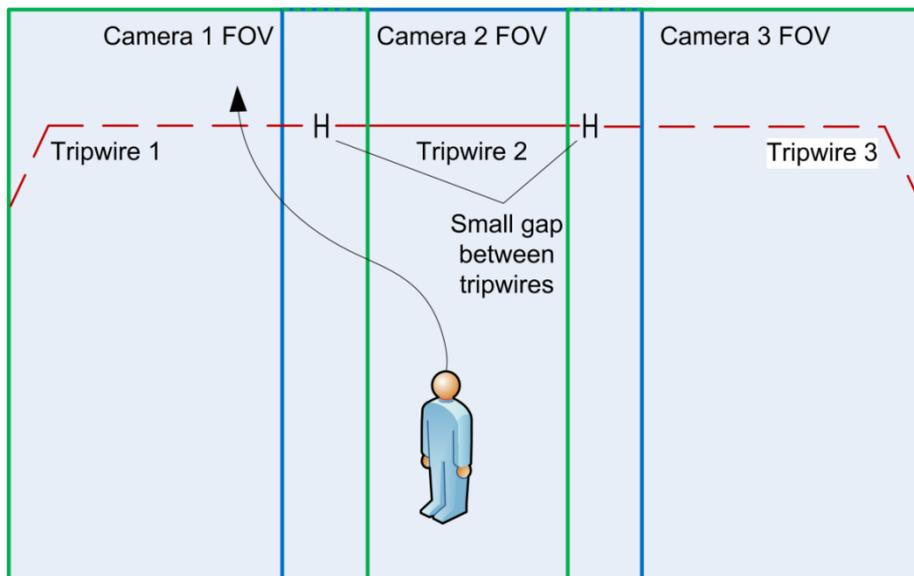
**Figure 20: The solution for problems 1 and 2**

The layout shown above solves both problems 1 and 2.

- Camera views are slightly overlapping. The extent of overlapping should be at least the width of a person, about 60cm on the ground. This ensures enough room in camera 1's view to reliably detect the target before it crosses tripwire 1. This eliminates problem 1.
- The two tripwires (solid for camera 1, dashed for camera 2) are drawn as parts of one big virtual tripwire. There is no route for a person to cross both tripwires. This eliminates problem 2.

Important: „Object tracking regions” are assumed to be the same as the cameras' FOVs. The overlapping parts should be included in the region of every camera!

**For more than two cameras:**



**Figure 21: Setup with more than two cameras**

How to create the layout shown in Figure 20 or Figure 21 in practice:

1. Position the camera FOVs to be slightly overlapping.
2. Draw the tripwire as parts of a big virtual line.

**Note:** Currently every iCAT counter works individually without information exchange between them. Therefore the total result for the traffic flow is the sum of the individual results of all the cameras covering an area.

Initially it was thought that the ends of the tripwires of different camera views should be coincident or overlapping. However experiences show that there should be a small gap between the adjacent ends of lines. The reason is that if the line ends are coincident or overlapping a crossing object might be counted by both tripwire lines. If a small gap is left between the lines (about 10% of an object's size) the risk of multiple counting is lowered. With a gap being small enough the likelihood that an object will slip through the gap and will not be counted is very low.

**Summary:** To avoid counting people twice, or not counting them at all: (1) camera views have to be slightly overlapping as if they were one big counting view and (2) tripwires should look one big virtual tripwire, though small gaps should be left between adjacent tripwires.

## 5 Additional resources

- Netavis Observer User Manual:  
Netavis Observer – Help – User Manual  
<http://netavis.net – Support - Documentation>