Suspicious Behavior Detection in Video Surveillance

AALBORG UNIVERSITY

VGIS8 Group 843

Rita and Atanas

Overview

Introduction
- Motivation
- Anomalous vs Suspicious Behavior

Object Detection
- Background Subtraction
- Pre-processing
- BLOB Fusion

Tracking
- Kalman Filter
- Data Association
- Occlusion Handling

Classification
- Fainting
- Abandoned Object
- Abandoned Object Picked Up
- Loitering
- Running

Results
- Fainting
- Abandoned Object
- Abandoned Object Picked Up
- Loitering
- Running

Introduction

Motivation

- Nowadays, public safety has become a great concern.
- Thus, the need for automated surveillance systems has increased, with the goal of assisting security officers in performing their job more efficiently.

Anomalous vs Suspicious Behavior

- Anomalous behavior can be described as a deviation from the normal behavior. It is a behavior that differs from the usual and expected.
- Suspicious behavior takes into account the context of the environment to classify a behavior as suspicious.
- A behavior that is considered normal in one context can be considered suspicious in a different context.

Object Detection

First step

◎ From RGB frame to gray level and Gaussian Blur.

RGB frame.

Gray-level and Gaussian Blur frame.

Background Subtraction

- The Temporal Median Filter extracts the background image by removing the noise from the image.
- The values assigned to each pixel throughout time are stored and ordered.
- The median value of the list is chosen as the background, this eliminates values that are too high or too low, thus eliminating noise in the image.

Background image.

Background Subtraction

Frame.

-

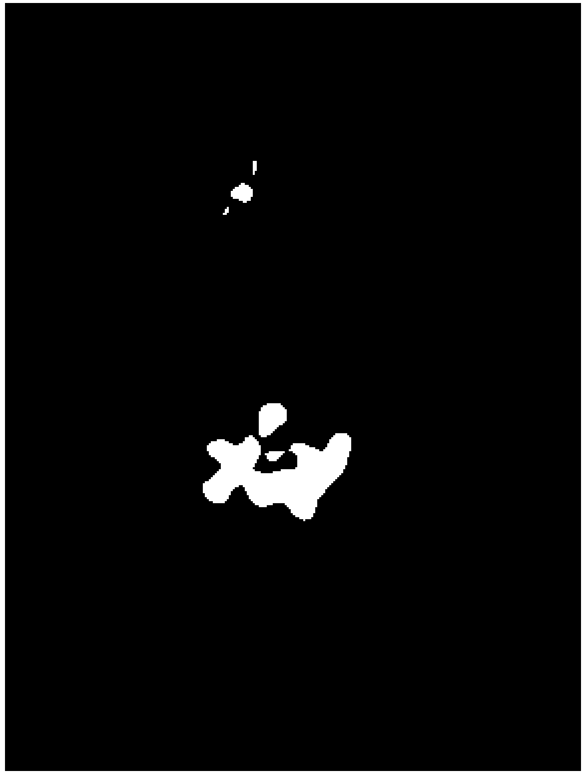
Background image.

=

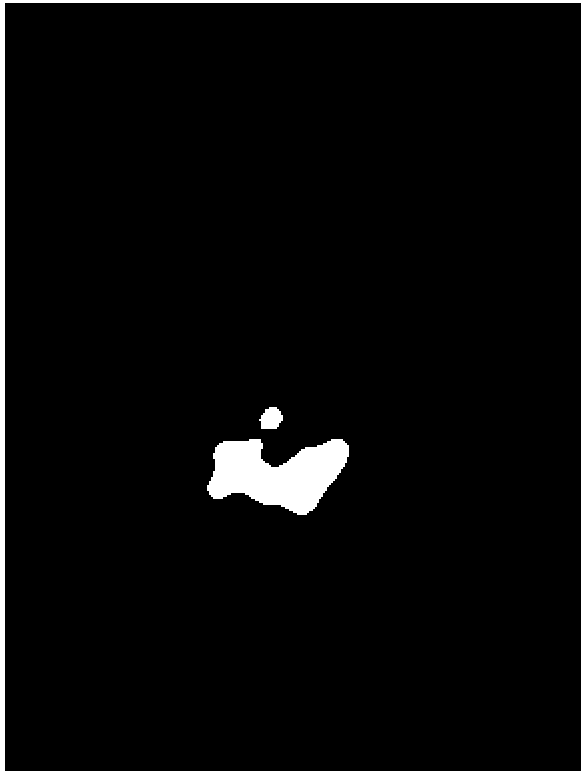
Result of background subtraction.

Pre-processing

- After the background subtraction, the binarization of the image takes place by thresholding the image.
- Noise in the frame is corrected by applying a median filter to the frame.



Binary image.

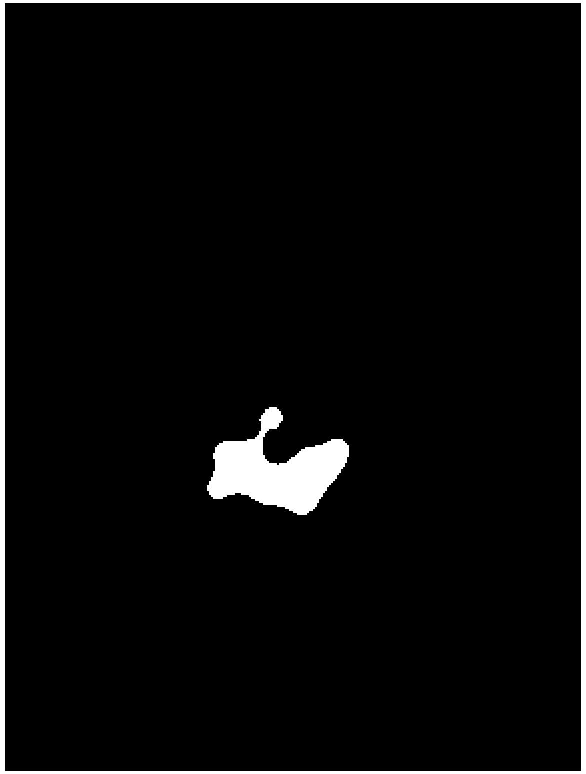


Median filter.

Pre-processing

- Lastly, the morphological operation of Closing is applied to the frame.
- A morphological operator takes as input a binary image and a kernel and combines them using an operation.
- The Closing operator is a combination of Dilation and Erosion.

$$g(x,y) = (f(x,y) \oplus K) \ominus K$$

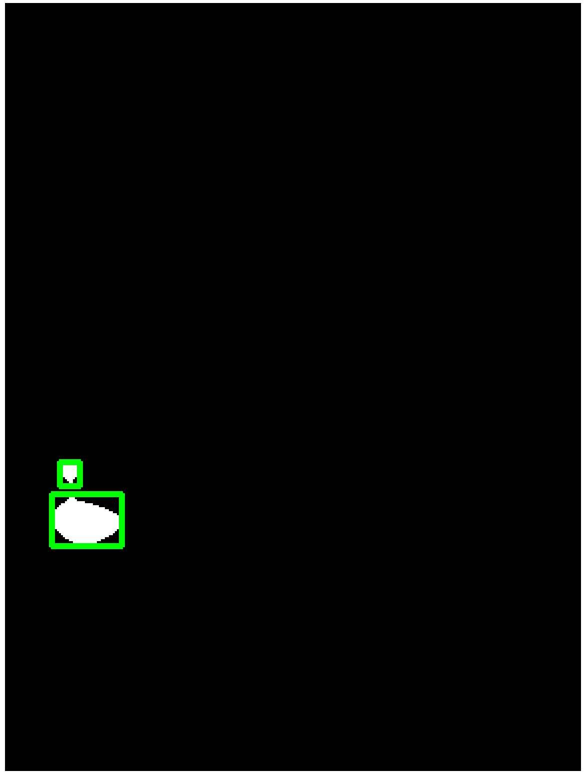


Frame with Closing morphological operation applied.

BLOB Fusion

- To correct cases where the segmentation of a person is split into more than one BLOB, a BLOB fusion algorithm was implemented.

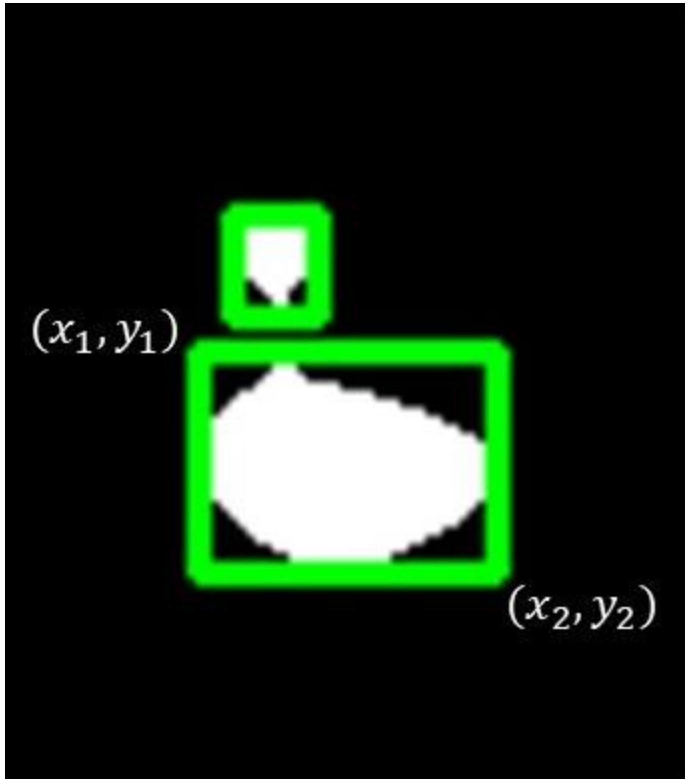
Frame.



Binary image with bounding boxes surrounding detected BLOBs.

BLOB Fusion

- Each detected BLOB in the image is characterized by their upper left and lower right corners coordinates (x_1, y_1, x_2, y_2).
- Two BLOBs are fused together if they pass all of the three conditions imposed.

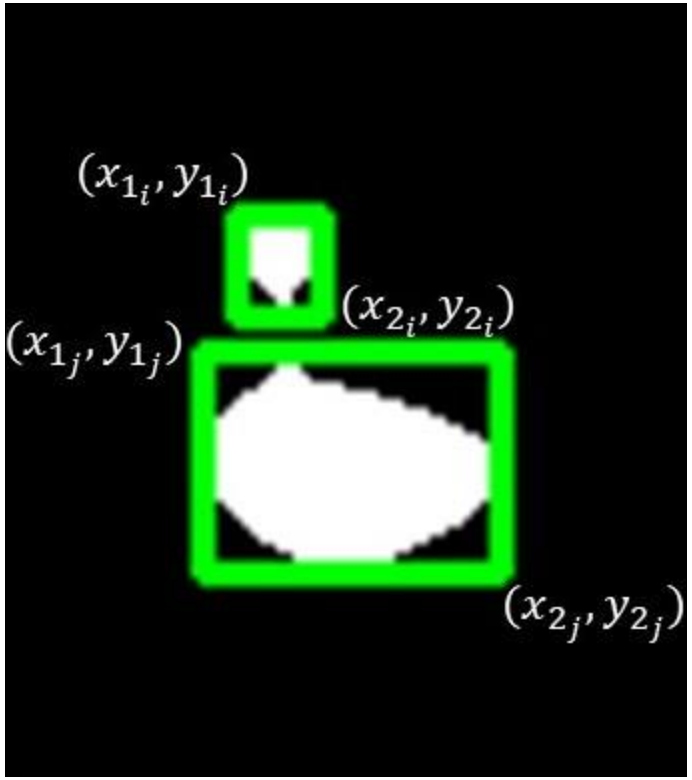


Fusion condition 1

- At least one of the BLOBs must have a bounding box area lower than a given limit l_1. The limit l_1 is an approximated value to the average of the area of all the BLOBs in the frame.

$$l_1 \approx \frac{\sum_{k=1}^{N} (x_{2_k} - x_{1_k}) \times (y_{2_k} - y_{1_k})}{N}$$

$$((x_{2_i} - x_{1_i}) \times (y_{2_i} - y_{1_i}) < l_1) \vee ((x_{2_j} - x_{1_j}) \times (y_{2_j} - y_{1_j}) < l_1)$$

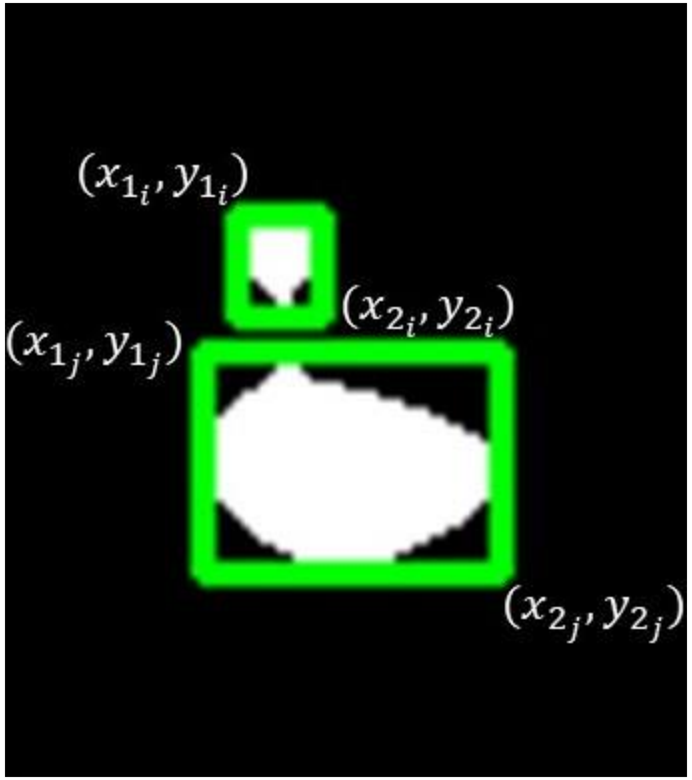


Fusion condition 2

- The second condition checks the vertical distance between the BLOBs. If the absolute distance between the lower corner of one of the BLOBs and the upper corner of another one is less than the set limit l_2, then the BLOBs are still candidates to being fused.
- l_2 is half of the median value of the BLOBs heights, measured throughout the video.

$$l_2 = \frac{Median(\{(y_{2_0} - y_{1_0}), \cdots, (y_{2_M} - y_{1_M})\})}{2}$$

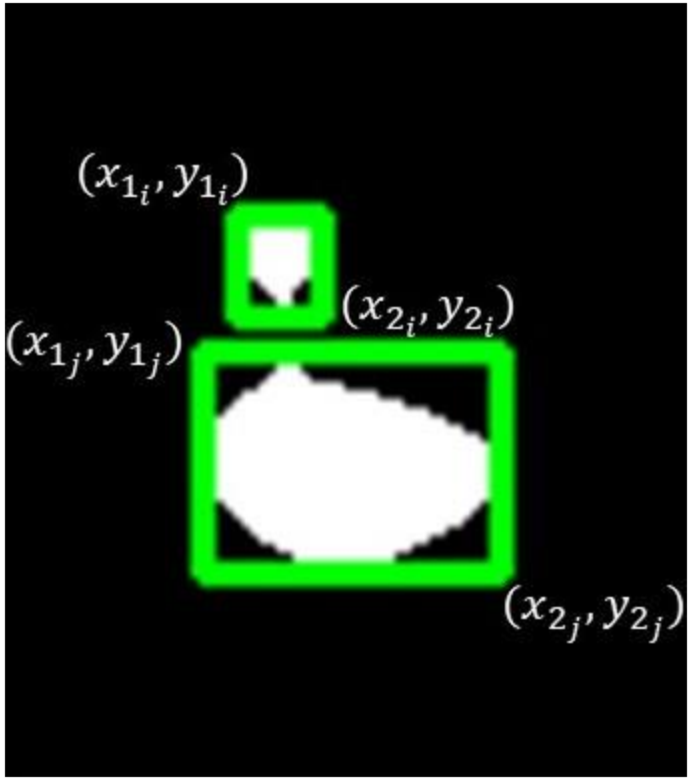
$$(|y_{2_i} - y_{1_j}| < l_2) \vee (|y_{2_j} - y_{1_i}| < l_2)$$



Fusion condition 3

- The last condition makes sure that the BLOBs are aligned horizontally.
- The BLOB with greater x_1 value (with the left corner more to the right) must be at least slightly aligned with the other BLOB by not having its left corner with a value greater than the right corner of the first BLOB. It is allowed for the BLOB to have a small offset of value *o*.

$$(x_{1_i} > x_{1_j} \wedge (x_{1_i} + o) \leqslant x_{2_j}) \vee (x_{1_i} < x_{1_j} \wedge (x_{1_j} + o) \leqslant x_{2_i})$$



Fusion

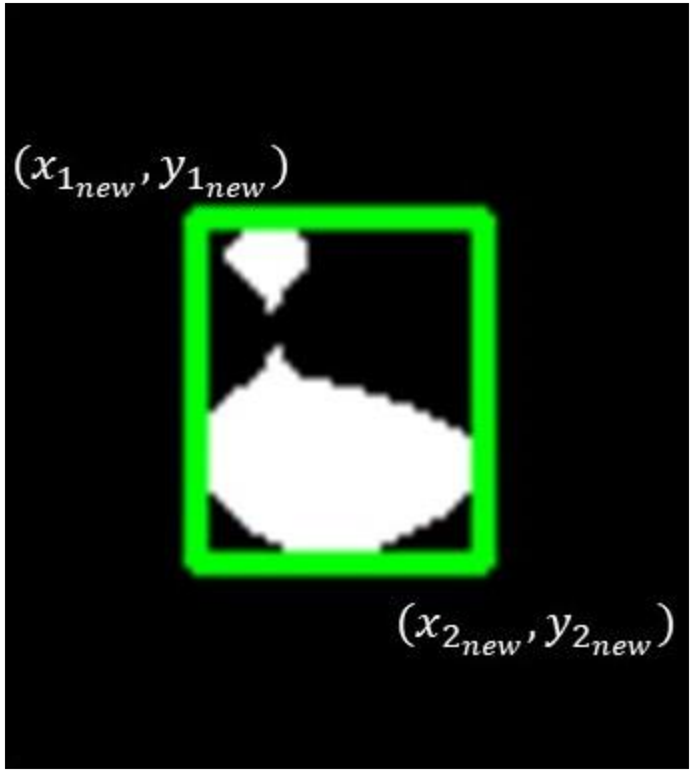
- If the pair of BLOBs passes all conditions they are merged, deleted from the list of candidates and a new candidate is added which is the result of the merger given by:

$$x1_{new} = \min(x1_i, x2_i, x1_j, x2_j)$$

$$y1_{new} = \min(y1_i, y2_i, y1_j, y2_j)$$

$$x2_{new} = \max(x1_i, x2_i, x1_j, x2_j)$$

$$y1_{new} = \max(y1_i, y2_i, y1_j, y2_j)$$



Tracking

Kalman Filter

- Recursive algorithm that predicts the next step from the previous state, filtering noisy measurements, and uses the new measurement to update the estimate.

Kalman Filter

Predict

Project the state ahead:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

Project the error covariance ahead:

$$P_k^- = AP_{k-1}A^T + Q$$

Correct

Compute the Kalman gain:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

Update estimate with measurement z_k:

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

Update the error covariance:

$$P_k = (I - K_k H)P_k^-$$

Gade, R., & Moeslund, T. B. (2014). Thermal Tracking of Sports Players. Sensors, 14(8), 13679-13691. DOI: 10.3390/s140813679

Data Association

- For each detected person a new Kalman filter is created.
- In each frame a Kalman filter is assigned the nearest detection (given a distance threshold).
- For each detection not assigned to a Kalman filter, a new one is created, thus beginning a new track.
- Kalman filters with no assigned detections are still continued based on predictions for a maximum of 5 frames without detections.

Occlusion Handling

- If a person is occluded by another person or if its BLOB is merged with the BLOB of another person in the segmentation stage, the people affected cannot be properly re-identified once the occlusion ends.
- To minimize the occurrence of this issue, an algorithm based on visual appearance was implemented.

Poor tracking due to occlusion.

After the end of the occlusion a person cannot be re-identified.

Occlusion Handling

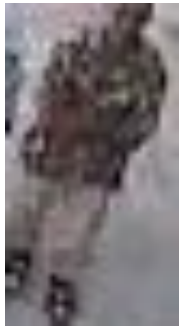
- When a track is lost, the algorithm tries to identify if an occlusion has begun.
- For each of the other tracks, the algorithm calculates the area of intersection between the lost track's bounding box and that track's bounding box.
- If the area of intersection is significant, an occlusion has officially begun and the tracks have merged.

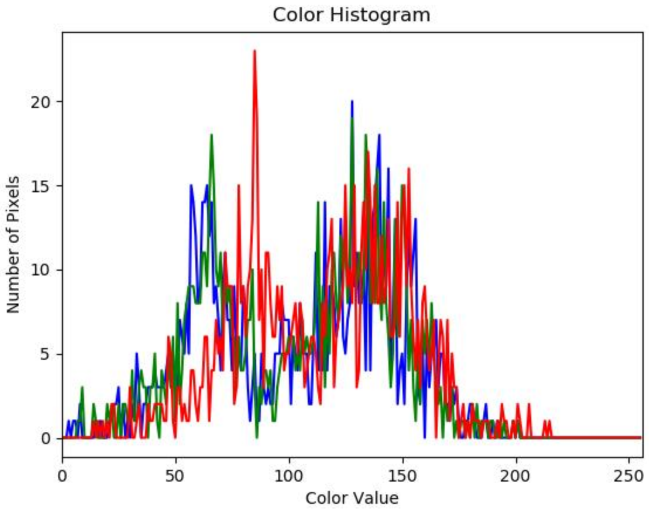
Frame right before occlusion.

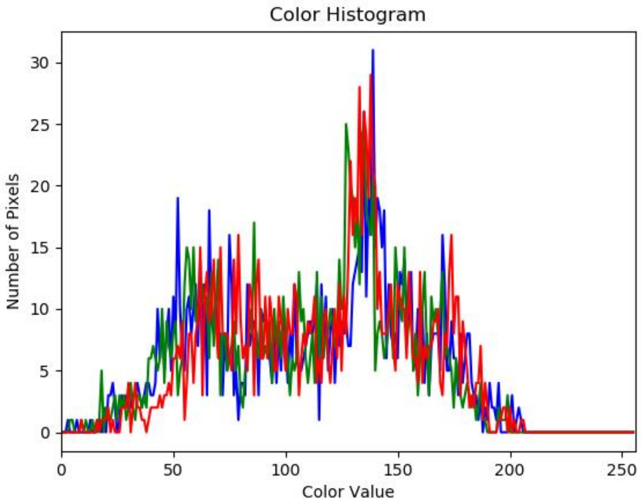
Frame where occlusion begins.

Occlusion Handling

- The color histograms of people involved in the occlusion are calculated and stored.
- The color histogram is a feature that distinguishes people from each other based on the color of their clothes.







Occlusion Handling

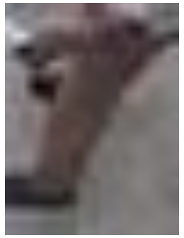
- When a new detection appears in the video it is verified if it might correspond to the end of an occlusion.
- If there is significant intersection between the bounding box of this detection and the bounding boxes of the other detections in the prior frame, that are marked as having more than one person merged in them, this means that an occlusion has ended.

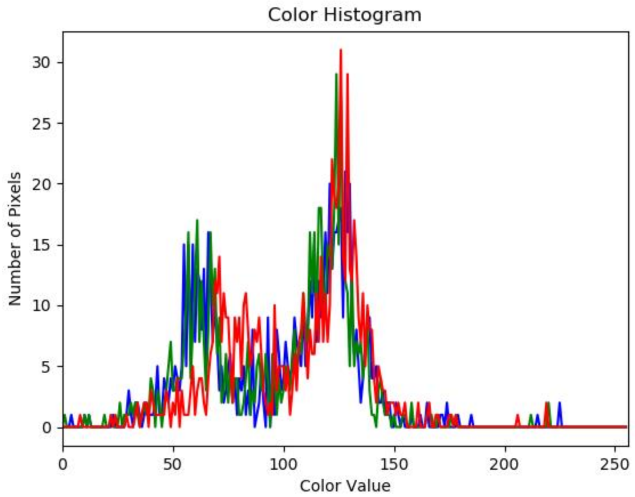
Frame right before occlusion ends.

Frame after occlusion ends.

Occlusion Handling

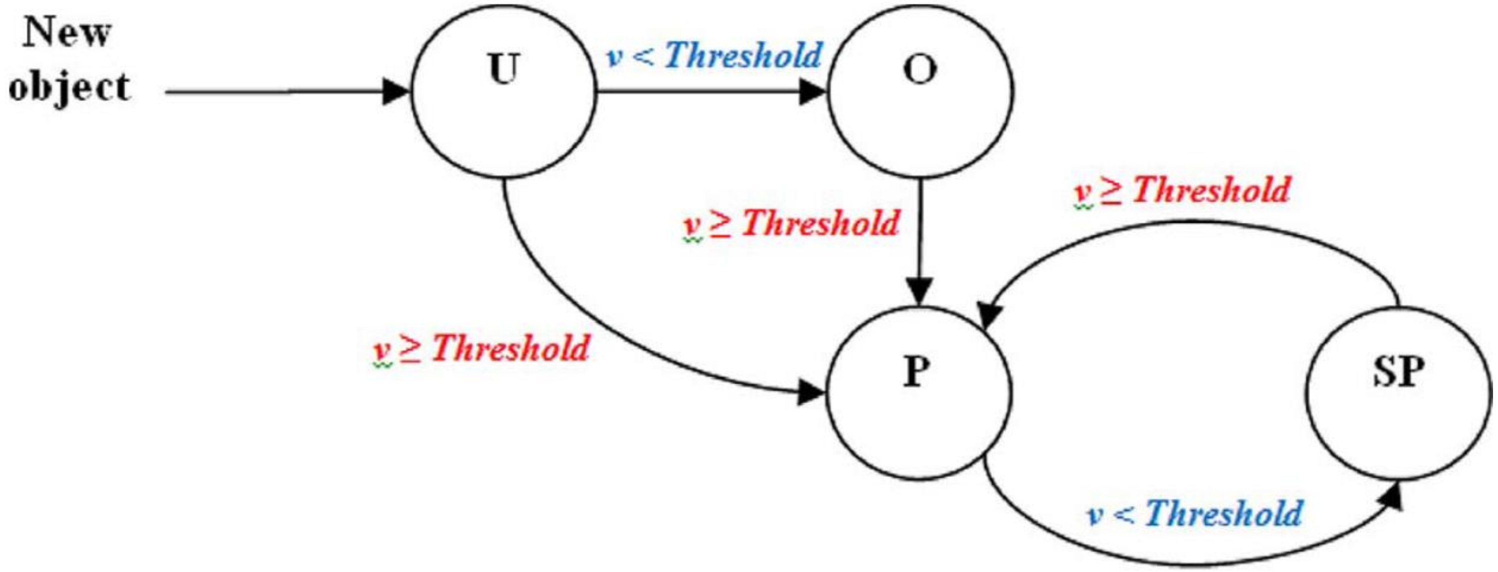
- When the end of an occlusion is verified the color histogram of the new detection is compared to all color histograms stored in the track that contained occluded people.
- The new detection is assigned to the track with the histogram with highest comparison score.





Classification

Classification



U = unknown, P = person, SP = still person, O = inanimate object, v = velocity.

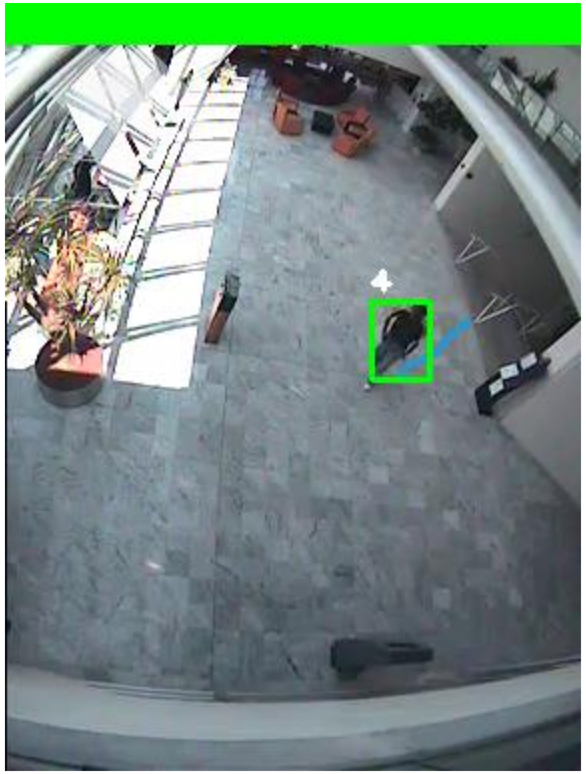
M. Elhamod and M. D. Levine. Automated Real-Time Detection of Potentially Suspicious Behavior in Public Transport Areas.

Fainting

- The aspect ratio of a person's bounding box can help determine whether a person is standing up or lying down.

$$\text{aspect ratio} = \frac{\text{height of bounding box}}{\text{width of bounding box}}$$

$$\mathit{category} = \mathrm{SP} \wedge \mathit{ratio} < \mathit{threshold}$$
$$\wedge\, \mathit{merged} = \mathrm{False} \wedge \mathit{width} \approx \mathit{standard\_height}$$

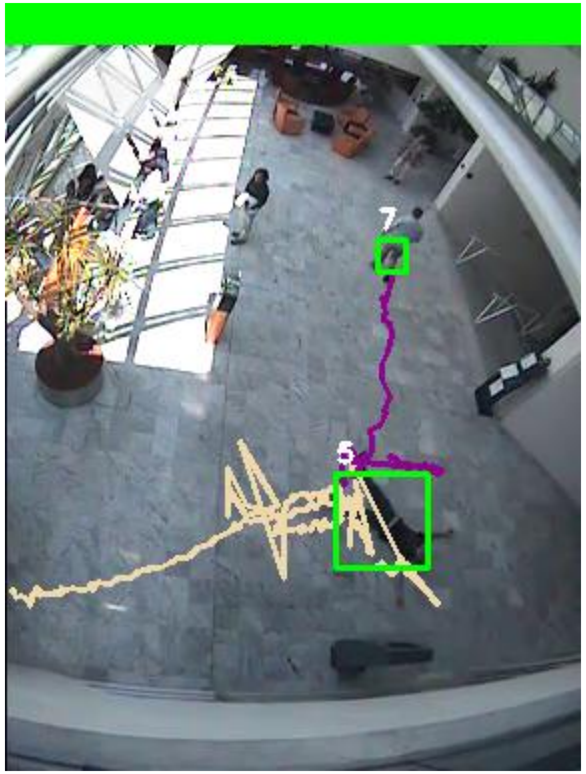


Frame of video with no suspicious behaviors detected.

Frame of video with fainting detected.

Fainting

- The fainting detection faces issues.
- Seen as the video is filmed from above, sometimes the bounding box of a person is not represented as expected which causes the aspect ratio not to be as expected. This, consequently, leads to a poor detection of this behavior.



Poor detection of the fainting behavior.

False detection of the fainting behavior.

Abandoned Object

- If a object is detected the system finds the person that left it unattended by checking if the bounding box of the person, around the time the object appeared, intersects almost completely with the bounding box of the object.
- If so, both, the object and the person, are flagged as suspicious.

Person leaving object on floor.

Person stepping away from object.

Abandoned Object

- The alert is given if the person is at a certain distance from the object and the person has been tracked for longer than the object.

$$
\begin{aligned}
&\text{category}(det1) = \text{O} \wedge \text{lifetime}(det1) > t_{abandoned} \\
&\wedge\ \text{category}(det2) \in \{\text{P}, \text{SP}\} \wedge \text{lifetime}(det2) > \text{lifetime}(det1) \\
&\wedge \text{distance}(det1, det2) > d_{abandoned} \\
&\wedge \text{intersection\_area}(det1, det2) \approx det1\_area
\end{aligned}
$$

Person leaving object unattended, rendering both person and object as suspicious.

Abandoned Object Picked Up

- As soon as a suspicious object stops being detected the system will look for someone that might have picked up the object.
- If the bounding box of where the object used to be, now intersects almost entirely with the bounding box of a person, then this person is flagged as possibly being suspicious.

Another person is approaching the suspicious object.

Person and object merge together.

Abandoned Object Picked Up

- If the person leaves the location of the object and the object is not detected again, it means that the person picked-up the object indeed and it is flagged as suspicious.

$$\text{category}(det1) = \text{O} \wedge \text{classification}(det1) = suspicious$$
$$\wedge\ \text{lost\_track}(det1) = \text{True} \wedge \text{intersection\_area}(det1, det2) \approx det1\_area$$

Person picks up suspicious object.

Loitering

- Loitering is defined as the presence of someone in an area for longer than a defined period of time.

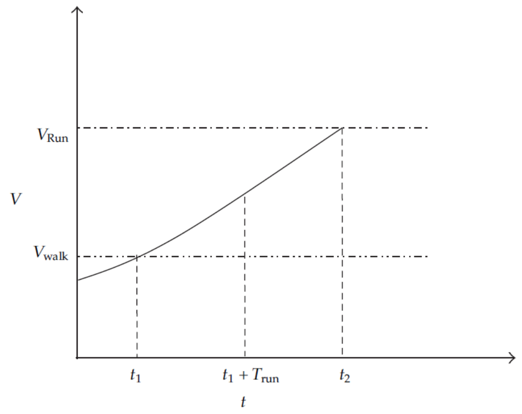
$$category \in \{P, SP\} \wedge lifetime > t_{loitering}$$

Person standing in lobby.

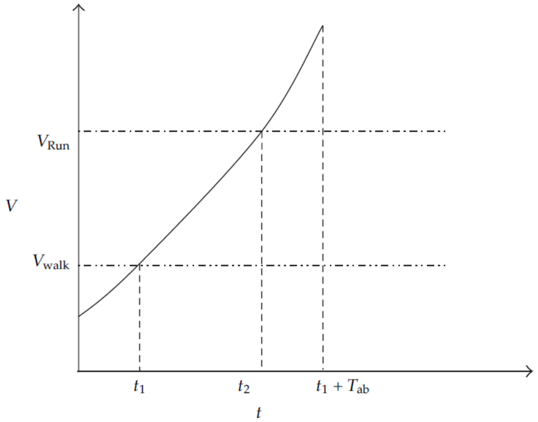
Suspicious behavior flagged as loitering.

Running

- Running behavior can be classified in two categories, normal running and abnormal running.
- Normal Running - A walking or stationary object maintains a constant acceleration over a long period of time before reaching and exceeding a set normal running velocity.
- Abnormal Running - A walking or stationary object suddenly accelerates before reaching and exceeding normal running velocity.



Normal running.



Abnormal running.

Running

- Determining whether an observed running behavior is normal or abnormal is done by comparing the average velocity over the past five frames to a normal velocity threshold. If the average velocity is greater than the threshold the observed behavior is classified as running.
- To classify a running behavior as abnormal the acceleration leading to the running state needs to be rapid. If the current acceleration exceeds a threshold the detected running behavior is marked as abnormal and hence suspicious.

$$velocity > v_{threshold} \land acceleration > a_{threshold}$$

Person running after fight.

Results

Results

- Recall demonstrates the ability of the system to find all the suspicious behaviors within the dataset.
- Precision expresses the proportion of behaviors that the system identified as suspicious that were actually suspicious behaviors.
- False Alarm Rate shows the probability of a false alarm.
- The accuracy demonstrates how close the number of detected suspicious behaviors is to the actual true number.

$$\text{recall} = \frac{TP}{TP + FN}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{false alarm rate} = \frac{FP}{FP + TN}$$

$$\text{accuracy} = \frac{TP + TN}{TP + FN + TN + FP}$$

Results

Fainting

- The system recognizes most of the fainting occurrences but only a few of the behaviors flagged as fainting are actual fainting occurrences.
  - Recall = 66,7%
  - Precision = 22,2%
  - False Alarm Rate = 3,1%
  - Accuracy = 96,5%

Abandoned Object

- A precision of 100% assures that all the objects identified as being abandoned are truly abandoned which relates to a False Alarm Rate of 0% and a very high Accuracy.
  - Recall = 66,7%
  - Precision = 100%
  - False Alarm Rate = 0%
  - Accuracy = 99,5%

Results

Abandoned Object Picked Up

- Only half of the times an abandoned object was picked up were detected by the system, however every detection was precise.
  - Recall = 50%
  - Precision = 100%
  - False Alarm Rate = 0%
  - Accuracy = 99,5%

Loitering

- The system classified correctly and precisely every loitering episode.
  - Recall = 100%
  - Precision = 100%
  - False Alarm Rate = 0%
  - Accuracy = 100%

Results

Running

- The implemented system is capable of recognizing all the running occurrences, however, only approximately a quarter of the flagged running occurrences correspond to the truth, which relates to a relatively high false alarm rate.
  - Recall = 100%
  - Precision = 26%
  - False Alarm Rate = 13%
  - Accuracy = 88%

Existing Similar System

- In the work developed by Elhamod and Levine the authors investigated similar behaviors and calculated the recall and precision for each and for the same dataset as used in this work.
- However, they present the results for specific videos, which means these results relate to singular videos and not to the entire dataset as the results of this work do.

Suspicious Behavior	**Recall**	**Precision**
Fainting	100%	80%
	80%	69%
	100%	67%
Abandoned luggage	89%	77%
	93%	82%
	81%	100%
Theft of luggage	100%	100%
Loitering	98%	100%

Conclusions

Conclusions

- Overall, the developed system has a good performance and is able to identify correctly most of the suspicious behaviors in the videos. Still, some aspects could be further improved in order to achieve better results.
- The occlusion handling algorithm could benefit from using another visual descriptor, besides the color histogram, like, for example, LBP (Local Binary Pattern) or HOG (Histogram of Oriented Gradients) to improve its performance.

Conclusions

- In the detection of the fainting behavior there are quite a few misclassifications. This might have to do with the angle of the camera filming. More research should be done on the most optimal ways of identifying this type of behavior.
- The running behavior relies greatly on defined thresholds which should be better investigated.

Thanks!