Video Content Analysis for Tunnel Surveillance

Reyes Rios-Cabrera, Tinne Tuytelaars and Luc Van Gool

KATHOLIEKE UNIVERSITEIT LEUVEN

K.U.Leuven ESAT-PSI/VISICS

Kasteelpark Arenberg 10,B-3001 Heverlee, BELGIUM

reyes@esat.kuleuven.be

VISICS

Short Description

Problem

As initial part of my PhD research, I have been working 4 months in tunnel surveillance. The project consist of detecting vehicles entering a tunnel and tracking them throughout the tunnel by the means of video cameras. The problem can split up into 2 main problems:

• The detection, classification (cars, trucks, buses, etc.) and identification (car 1 in camera 1, car 1 in camera 2…) of vehicles at the entrance and in each camera

• The subsequent tracking of the vehicles throughout the tunnel

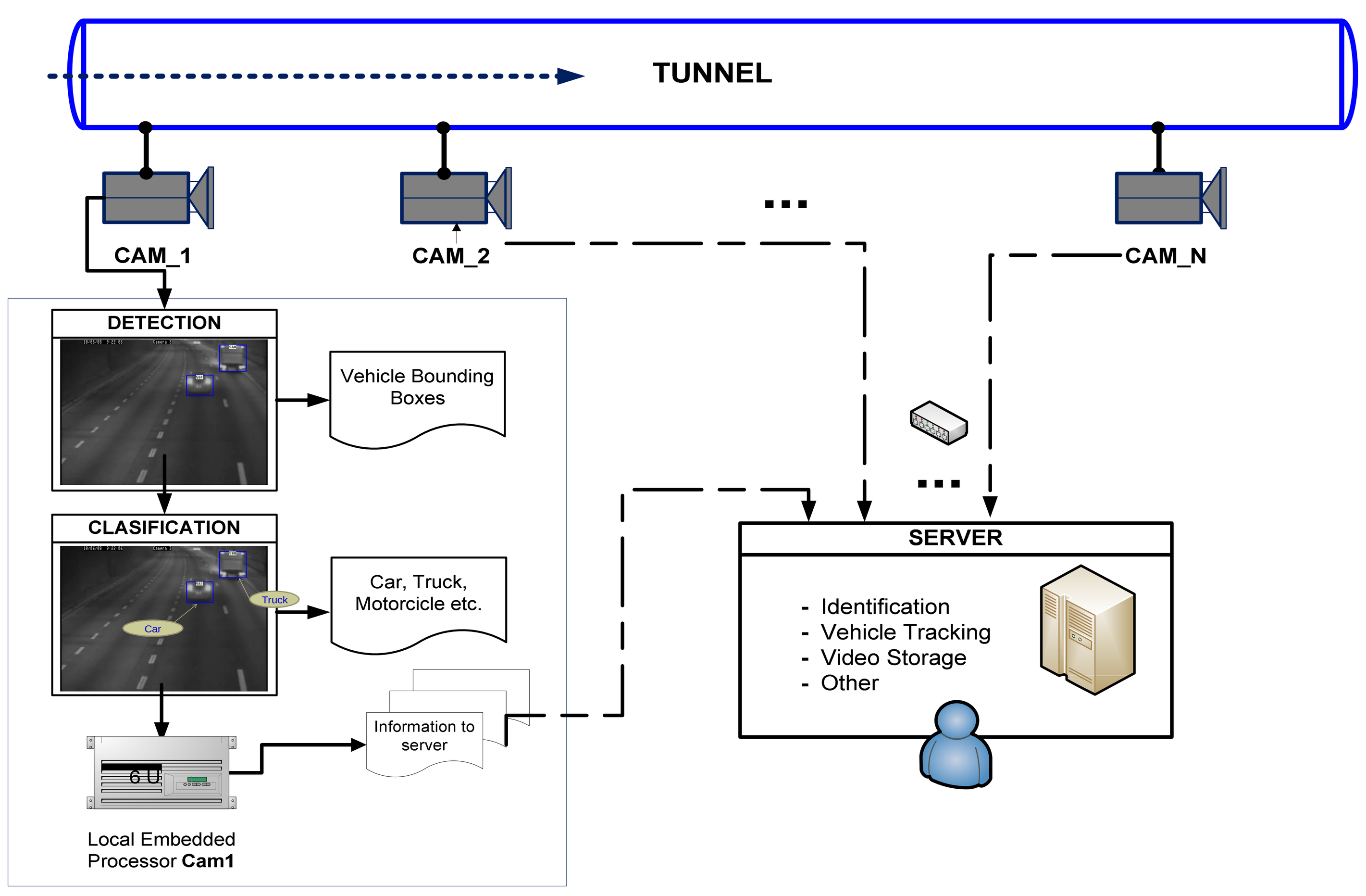
From a research point of view the core challenge consists in the correct detection and tracking of a certain vehicle in a camera image and over the different cameras, where illumination circumstances are far from ideal, dealing with multiple viewpoints, where head lights could appear and very fast processing is necessary to make it useful.

General Architecture

• Local embedded processor will run detectors and classifiers for each camera through the tunnel.

• A server collects the information for identifying and track each vehicle through the tunnel.

• Identification and tracking are carried out in the server.



General Project Contributions

Technically:

• Tracking vehicles through a tunnel by the means of video cameras is new.

• Further integrating this methodology onto embedded hardware and into a traffic management software (TMS) so the user knows at every moment where a certain vehicle is also new. Results will be implemented for end-users applications by a company.

From a research point:

There are several problems that are being tackled and that have not been widely described in the scientific literature:

• Tracking vehicles over several cameras in a tunnel environment is new.

• Detecting, segmenting vehicles and identifying them through the tunnel by the means of object recognition methods is also new.

Partial Results

Adaboost cascade is being used for detection, where the output selection was modified, using a non-expensive method found through experimentation, reducing false positive and increasing bounding box quality selection. For identification we have implemented haar features vectors. We have tested our first proposal on 3 challenging videos

Detection/Identification of Vehicles

Main algorithms

For detection, we are using Cascade Scheme as Viola-Jones proposal

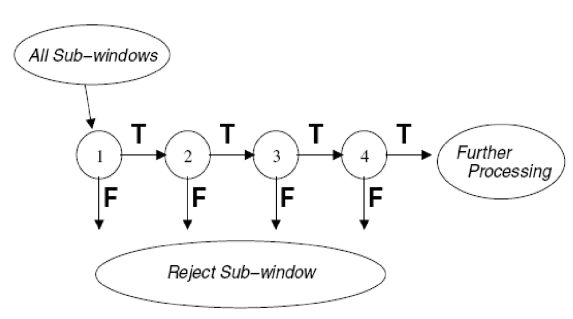
We used Multiboost (C++ AdaBoost implementation)

http://multiboost.sourceforge.net/

We have modified it to implement the cascade scheme and optimized it to train faster.

After running different cascades (dif config),

We decided to used only: 2H, 2V, and 3H, which gave best



Training

3 videos of different cameras of a tunnel were annotated. Containing about 600 vehicles each one. For training/offline-testing the cascade we used the data of the first 2 videos, having a set of about 1200 vehicles. For online testing we used the 3rd video. The data:

- Size of images 50x50. Integral image= 2,500 elements, were considered.
- Using Annotated data we generated more positives, using mirror images, and surrounding bounding boxes. For the negative data, 60 images of different tunnels were found for training and the following data was generated:

POSITIVE DATA	
size Train	5480
sizeEvaluate	4000
sizeTest	2000
total	**11480**
NEGATIVE DATA	
size Train (pool)	40138
sizeEvaluate	4000
sizeTest	2000
total	**46138**

Accuracy of recognition was set to 99.9%,

False positive was set to 50%. The training time: 13.46hrs (whole cascade). We use random homogeneous features distribution for speeding up.

The results were: Detection Accuracy: 98.15%, False Positive: 1.7%, Cascade:15 stages with a total of 1500 classifiers

Testing of Cascade on Video

Using a sliding window for the videos, we changed the standard winners output selection of the cascade to improve accuracy reducing False Positive.

We do not consider all the output of the cascade as winner for post processing, but we calculate the rejection value of each stage as well, and take it into consideration at the end. In this way, only sub windows that had generally a high confidence in most stages, are selected as output.

Winners only If $T < 0$, where T= total output value of the "as background classified element".

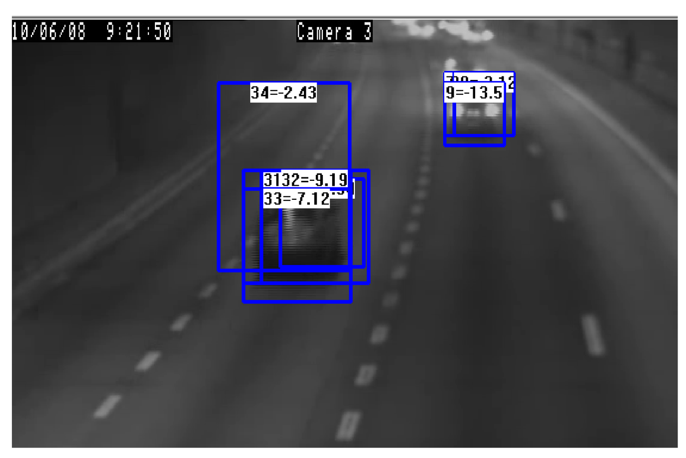
$$T = \sum_{i}^{N} t_i$$

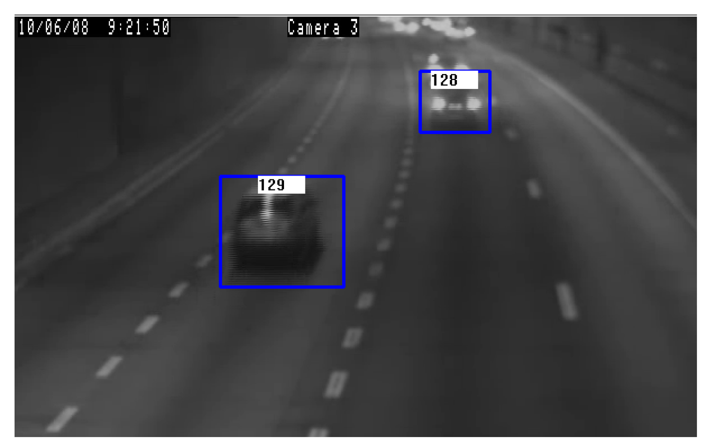
t=cascade stage output value of the "as background classified element"

N= stages of the cascade

To select the best detection a minimum overlapping of 2 was set, and best T average was used when different sizes where found.

After post-processing, false positive is about **1.02 each 100 frames** or about **1 FP every 100 frames**





The above processing has no movement detection, nor background substraction, which, will help to improve accuracy.

For Identification we are using the last stage classifiers of the cascade. Using Euclidean distance and tunnel information, we are able to identify properly 88% of vehicles.

Current and Future Work

Currently we are experimenting with:

- Motion detection
- Background substraction
- Classification of the vehicles using shared features.