EmerT – Supporting traffic parameter estimation from low cost and low resolution uncalibrated web cameras

Louis Calvin Touko Tcheumadjeu[1*], Sten Ruppe[1], Musharaf Ali[2], Laura Bieker[1]

German Aerospace Center (DLR)

1. Institute of Transportation Systems

2. Institute of Robotics and Mechatronics

Rutherfordstr. 2, 12489 Berlin, Germany, Tel: +49 30 67055 284

{Louis.ToukoTcheumadjeu, Sten.Ruppe, Musharaf.Ali, Laura.Bieker}@DLR.de

* Corresponding author

Abstract

This paper describes the EmerT portal (Emergency mobility of rescue forces and regular Traffic), an extended version of the DELPHI web tool. EmerT is a web-based decision support application for real time traffic situation, prognosis and traffic simulation appropriate for exceptional situation like disasters, major events, incidents or emergencies. One of the main features in this system is to provide traffic state estimation and forecasting with the use of the mesoscopic traffic simulation. Current traffic data are needed for accurate simulation results. Typical inputs data are from ground-based sensor like induction loops. One further possibility represents the use of traffic web cameras, in which images are most freely available. Web cameras have mostly a low resolution and a low frame rate. These underlying limitations as well as the integration in the simulation are the big challenges for an efficient use as data input for traffic simulation. This paper aims to present the process for extraction of traffic data from traffic web cameras and the integration of this approach in the EmerT portal.

Keywords:

TRAFFIC STATE ESTIMATION, DECISION SUPPORT, TRAFFIC WEB CAMERAS, INCIDENT MANAGEMENT

Motivation

Mobility and transport systems are crucial for modern cities. But large events and incidents (like extreme weather conditions, earthquakes, festivals and traffic accidents) might cause problems to the transport systems as a result traffic jams and congestions are occurring. In some cases the effected population needs to be evacuated and traffic management actions need to be taken by public authorities to prevent the worst case. Currently there are no automated traffic information and traffic decision support systems which support these logistic and traffic management decisions. The decisions have to be made by manual estimations, inexplicit experience and available knowledge of traffic scene. The aim of the EmerT portal is to provide a decision support system for traffic management especially for events and natural or man-made disasters. Basis to provide a decision support is a mesoscopic traffic simulation

Mesoscopic Traffic Simulation

The EmerT portal displays the current traffic state of the whole traffic scene in a specified area. Another feature of the system is to estimate a traffic forecast. To reach this goal a traffic simulation called SUMO (Simulation of Urban Mobility) was integrated in the portal. SUMO is developed by the German Areospace Center (DLR) since 2001 [1]. To simulate the traffic of a large region like the area around Munich SUMO was extended by a mesoscopic traffic model [2]. The mesoscopic traffic simulation allows computing a traffic forecast for the next 30 minutes from a traffic section in every 10 minutes. Traffic online data like detector values, FCD (Floating Car Data) and information from traffic cams are used to feed the simulation with appropriate traffic data and to calibrate the simulation. By using several data sources the EmerT portal is able to combine this information to display the traffic situation of a whole region. Especially, the pictures from traffic cams give the opportunity to the user to reassure that the simulated traffic state is valid.

Requirement for EmerT

Current traffic data are needed for accurate simulation results. Typical input data for SUMO are the road-networks, traffic demand and the current traffic data. The latter are traffic flows and local speed from ground-based sensor like induction loops. The use of traffic cameras as an alternative data source seems to be a promising way [3]. In the VABENE project [4], an approach is investigated, which can estimate traffic data from low cost and low resolution uncalibrated web cameras. One advantage is that the images are mostly freely available. Next section described our approach to estimate the traffic information from the traffic web cameras.

Algorithmic concept of the traffic information extraction from the web cam

The main focus of this work is to investigate the feasibility of real time traffic monitoring system using low cost, low resolution uncalibrated web cameras. The general overview of the system is shown in Figure 1 (left).

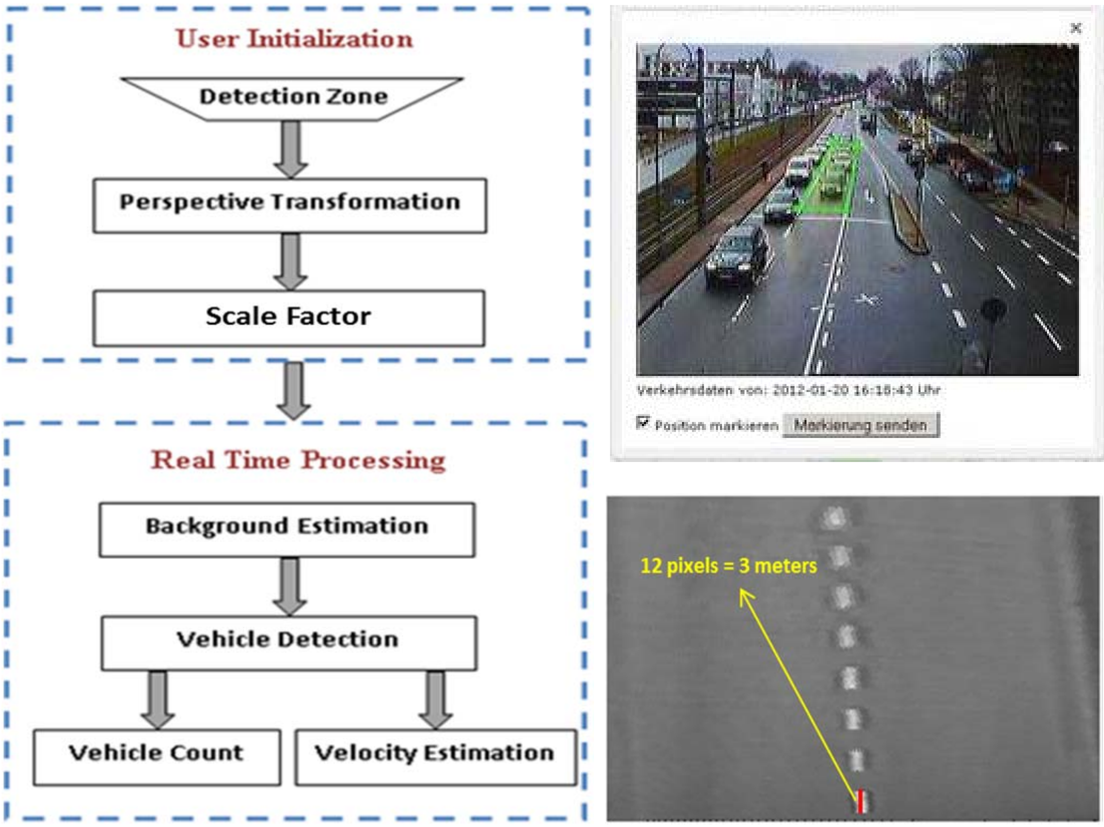


Figure 1: General Block diagram of the traffic camera processing workflow

- In user initialization, first step is the selection of the desired detection zone from the traffic scene as shown in Figure 1 (top right). After the selection of detection zone, second step is the perspective correction in which original image of detection zone is transformed into top-view image as shown in Figure1 (bottom right) as described in [5]. Through perspective correction, all the pixels in the scene become on same scale for further processing and measurements. For traffic parameters estimation, exact camera calibration parameters are necessary. But due to the unavailability of web camera parameters, we can calculate the scale factor in the image by using geometric relationships or markers that are inherently available in the traffic scene such as lane markers in the middle of the road section etc.
- We proposed a novel background estimation algorithm called FIFO buffer based background estimation using Mutual Information which returns a stable background image even under heavy traffic flow and jam conditions. If the mutual information between current image and the previous image is less than the specified threshold then this implies that scene has no traffic or has free flowing traffic, thus the current image is added into the

buffer and the buffer is updated according to the First-in-First-out method. If road traffic condition is heavy or jams, then the mutual information between the current image and the previous image will be higher than the specified threshold and thus it will not be added into the buffer. Median of the current filled buffer generates stable background image which is invariant of changing weather, illumination and traffic conditions.

- After background estimation fifth step is the vehicle detection which is the simple background subtraction procedure in which current image is subtracted from current background image followed by simple morphological operations and connected component analysis.
- In velocity estimation step; for incoming traffic scene, vehicle closest to the camera field of view in the current image is only investigated for finding its correspondence with detected vehicles in the previous image. By using the centroids information of detected vehicles in the rectified images with some ground truth constraints, velocity of the vehicles can be measured by Euclidean distance reliably. Two examples of velocity estimation algorithm are shown in Figure 2.

$$Distance = min\left(\sqrt{(x_t^1 - x_{t-1}^j)^2 + (y_t^1 - y_{t-1}^j)^2}\right)$$

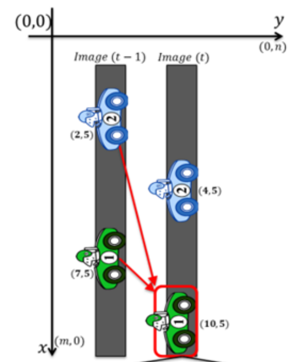


Image (t)	Image(t − 1)	$x_t^1 > x_{t-1}^j$	Euclidean Distance	Correspondence
V_t^1	V_{t-1}^1	true	2	☑
V_t^1	V_{t-1}^2	true	8	

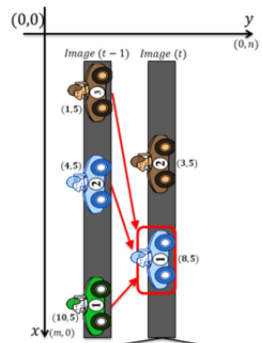


Image (t)	Image(t − 1)	$x_t^1 > x_{t-1}^j$	Euclidean Distance	Correspondence
V_t^1	V_{t-1}^1	false	2	
V_t^1	V_{t-1}^2	true	4	☑
V_t^1	V_{t-1}^3	true	7	

Figure 2: Two different traffic scenarios for velocity estimation Algorithm

- In vehicle counting step; for incoming traffic scene, vehicle closest to the camera in the current *image (t)* and the vehicle closest to the camera in the previous *image (t-1)* are only under investigation for vehicle counting. For incoming traffic scenario, vehicle counting algorithm comprises of two conditions as follows.

  Condition1: If vehicle detected in current *image (t)* and no vehicle was detected in the previous *image(t-1)*. Then increment the counter as new vehicle appeared in the current image such that.

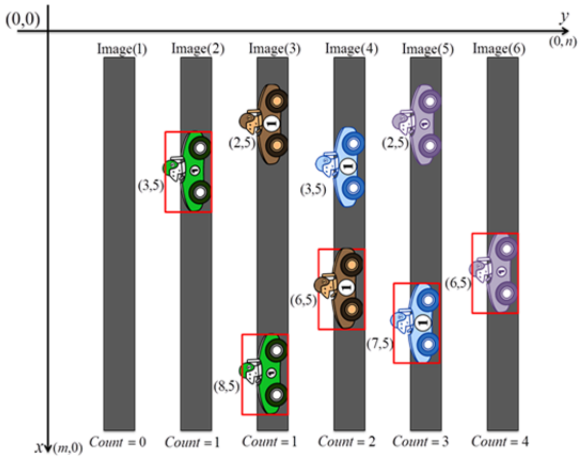
$$if \;\; v_t^i \;\; \Rightarrow \;\; v_{t-1}^j \rightarrow false$$
$$count = count + 1$$
$$end$$

Condition 2: If the vehicle V_t^1 detected in current $image(t)$ closest to the camera with the centroids location (x_t^1, y_t^1), where x_t^1 and y_t^1 are the horizontal and vertical coordinates respectively and vehicle V_{t-1}^1 detected in previous $image(t-1)$ closest to the camera with centroids (x_{t-1}^1, y_{t-1}^1), then the condition 2 is given

$$if \; (\, x_{t-1}^1 \; + \; \alpha \; \geq \; x_t^1) \; \rightarrow \; true$$
$$count \;\; = count + 1$$
$$end$$

Condition 1 and condition 2 are joined to represent the vehicle counting algorithm in a single propositional logic condition, which is as follows

$$if \quad \neg\left(\left(v_t^i \; \Rightarrow \; v_{t-1}^j\right) \; \otimes \; (x_{t-1}^1 + \alpha \geq \; x_t^1)\right) \rightarrow true$$
$$count = count + 1$$
$$end$$



t	x_{t-1}^1	x_t^1	$\neg\left(\left(v_t^i \Rightarrow v_{t-1}^j\right) \otimes \left(x_{t-1}^1 + \alpha \geq x_t^1\right)\right)$	count
2	0	3	true	1
3	3	8	false	1
4	8	6	true	2
5	6	7	true	3
6	7	6	true	4

Figure 3: Example of vehicle counting algorithm

Integration into EmerT

Figure 4 gives an overview of the realized traffic camera processing system in the context of overall process chain -- traffic cam data source (*data provider*) → traffic cam data import and management (*Traffic Data Platform*[6,7]) → traffic cam data processing (*traffic cam processing module)* → real time visualization of the processed traffic cam data (*EmerT-portal*)

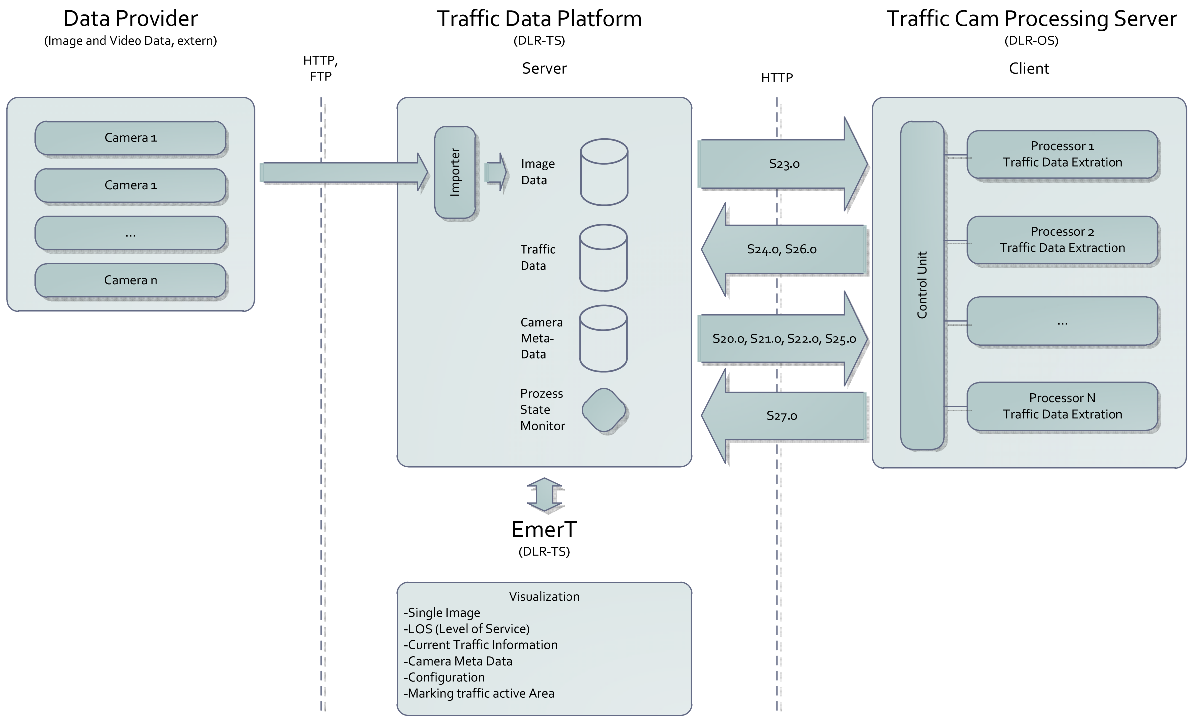


Figure 4: System architecture of the traffic cam processing system

The specification of the system interfaces **S21.0** until **S27.0** is described in the Table 1 in the appendix.

Visualization of the traffic information obtained from the web cameras

The Emert-portal as web based decision support tools for traffic management has been extended to support the configuration and calibration of traffic cameras as well as the visualization of the traffic information obtained from the traffic camera images processing. As shown in the Figure 1 following traffic data has been estimated from the traffic camera images: vehicle counting aggregated to obtain the traffic flow, the velocity as travel speed, the quality factor of the frames of web camera related to the noise level, the information if the traffic camera is moved or not. All these estimated traffic information are visualized in real time in the EmerT-Portal. Additionally the level of service (LOS) traffic information calculated from the estimated velocity as well as the single image of traffic camera is also visualized in the EmerT-Portal (see Figure 5).

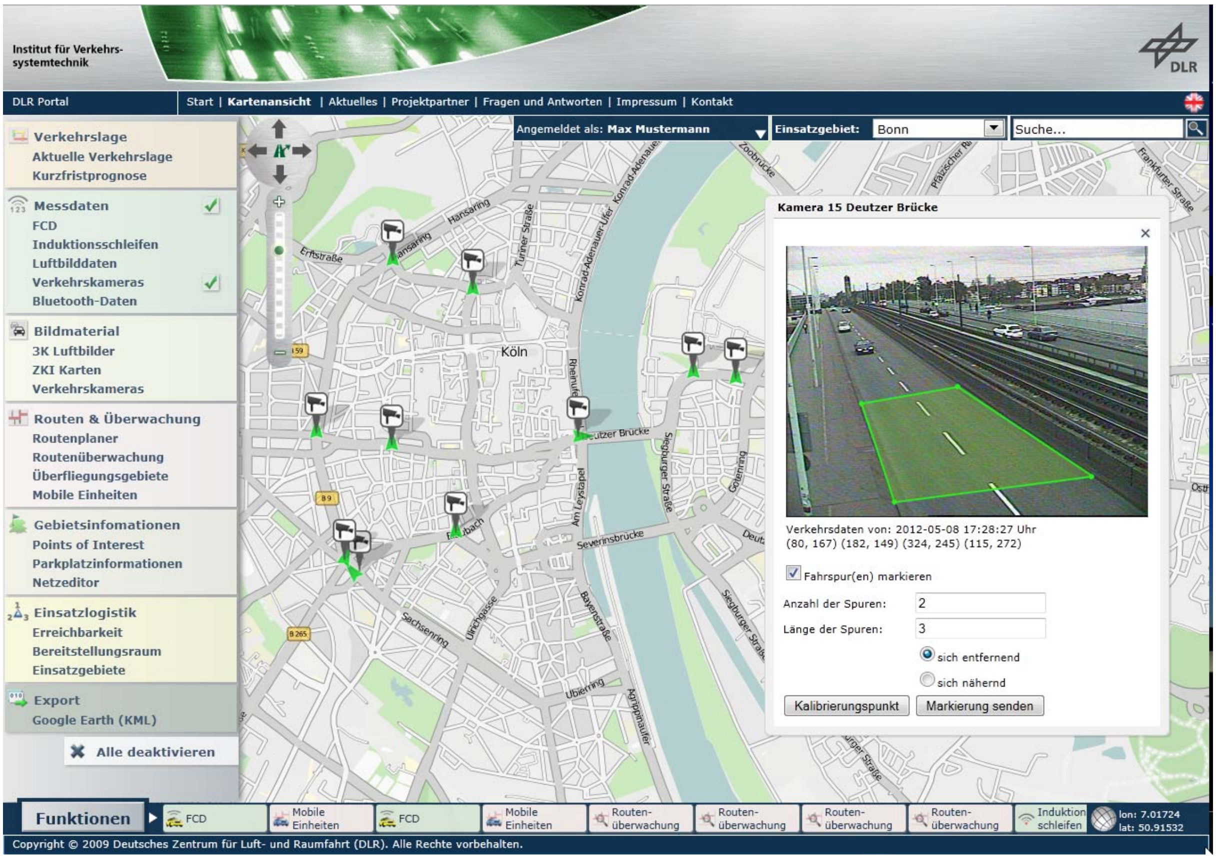


Figure 5: Overview of EmerT portal

Exemplary results of real time traffic monitoring system

The real time traffic monitoring system using low cost, low resolution uncalibrated web cameras has been implemented and exemplary demonstrated for the regions Cologne in Germany. The road side based web traffic cameras for this region have been used as shown in the Figure 4a

We tested the algorithms on two entirely different datasets from the web cameras in the Cologne city. The first data set is the Deutzer Brücke Street with incoming traffic scene in day time and the second dataset is the Köln Straße with outgoing traffic scene in night time. The image of each dataset is shown in Figure 6(b), (c). Both data sets have a pixel resolution of 384 x 288 with a frame rate of 1FPS (Frames per second).

(a) (b) (c)

Figure 6: (a) location of web cameras in Cologne (b) “Deutzer Brücke” street image with incoming traffic in day time, (c) “Köln Straße” image with outgoing traffic in night time.

For several lanes in one direction, each lane is selected as a separate detection zone and processed independently. However during real-time processing, the lanes are processed in parallel. For an evaluation of the vehicle-counting algorithm, we examined 20 minutes of video sequence (1200 frames) from each dataset. Vehicles are also counted manually for reference. Table 1 shows that our system is able to track and count vehicles with high accuracy.

Table 1: The vehicle-counting algorithm evaluation result

Data Set	Frames	Video Sequence in Minutes	Time of Day	Traffic Direction	Lane1			Lane2			Execution Time
					Estimated Count	Manual Count	% Accuracy	Estimated Count	Manual Count	% Accuracy	
Deutzer Brücke	1200	20 mins	Day	Incoming	216	221	97.73%	138	128	92.18%	0.41 Frames/sec
Köln Straße	1200	20 mins	Night	Outgoing	65	68	87.93%	59	50	82%	0.37 Frames/sec

Conclusions and outlook

The main focus of this work is to investigate the feasibility of real time traffic monitoring system using low cost, low resolution uncalibrated web cameras. Main challenges for traffic parameters estimations are vehicle detection, velocity estimation and vehicle counting. Moreover the Emert-portal as web based decision support tools for traffic management has been extended to support the visualization of the traffic information mentioned above. Based on our test results evaluation, implementation and exemplary demonstration for the region Cologne in Germany, we found that for Deutzer Brücke video sequence, the vehicle-counting algorithm returns 98% and 92% accurate vehicle counting for lanes 1 and 2, respectively. The second dataset is Köln Straße in night time with outgoing traffic scene. It has a very high grain noise effect (standard deviation > 12) which is an inherent feature of every digital web camera during night time. In the presence of this high noise factor for night time video sequence, our algorithm achieved high accuracy for vehicle counting, which is 88% and 82% for lanes 1 and 2, respectively. In the next step, we will evaluate the results of our camera based algorithm with the results available from the loop detectors and comparison between them for finding the results accuracy.

References

1. Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D. (2011). SUMO – Simulation of Urban MObility: An Overview. In Proceedings: *SIMUL 2011, The Third International Conference on Advances in System Simulation.* SIMUL 2011, 23.-28. Oct. 2011, Barcelona.
2. Behrisch, M., Bonert, M., Hinkeldein, D., Krajzewicz, D., Kuhns, G., Wang, Y.-P. (2009). DELPHI - a joint web decision support application for real time traffic situation analysis and prognosis, information exchange and cooperation. ITS World Congress 2009, 21.-25. September 2009, Stockholm, Sweden.
3. Guolin Wang, Deyun Xiao and Jason Gu (2008). Review on vehicle detection based on video for traffic surveillance. Proceedings of the IEEE International Conference on Automation and Logistics Qingdao, China September. Pages. 2961-2966.
4. http://vabene.dlr.de/ , VABENE Project, German Aerospace Center (DLR), 23.05.2012
5. Y. Malinovskiy, Y. Wu and Y. Wang, “Video-Based Vehicle Detection and Tracking Using Spatiotemporal Maps,” Transportation Research Record: Journal of the Transportation Research Board, vol. 2121, pp. 81-89, 2009.
6. Touko Tcheumadjeu, Louis Calvin and Brockfeld, Elmar and Ruppe, Sten. (2010). Traffic Data Platform as ITS Infrastructure for Intelligent Traffic Data Management. 17th ITS World Congress 2010, 25.-29.October 2010, Busan, Korea.
7. Touko Tcheumadjeu, Louis Calvin and Ruppe, Sten and Brockfeld, Elmar. (2010). Traffic Data Platform based on the Service Oriented Architecture (SOA). 12th World Conference on Transport Research (WCTR), 11.-15. July 2010, Lisbon, Portugal.

Appendix

Table 1: Specification of interfaces between the system components in Figure 4

Number	Interface	Description
S20.0	**GetActiveTrafficCams**	Interface to provide the list of available camera for a region
S21.0	**GetLatestTrafficCamPictures**	Interface to provide the list of latest camera images
S22.0	**GetLatestTrafficCamPictureMetaData**	Interface to provide the list of camera meta data
S23.0	**GetTrafficCamPicture**	Interface to provide the camera image of a selected web camera
S24.0	**SetProcessingData**	Interface to provide the processed traffic data obtained from traffic camera
S25.0	**GetTrafficCamConfiguration**	Interface to provide the configuration data of a selected web camera
S26.0	**SetTrafficCamPictureFrame**	Interface to provide the camera image of a selected camera as

		frame
S27.0	**SetProcessState**	interface to Process-State-Monitor