Fast Vision-Based Road Tunnel Detection

Massimo Bertozzi, Alberto Broggi, Gionata Boccalini, and Luca Mazzei

VisLab - Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Parma, Italy
{bertozzi,broggi,mazzei}@vislab.it,
gionata.boccalini@studenti.unipr.it
http://www.vislab.it

Abstract. When a vehicle equipped with an artificial vision system enters or exits a tunnel, the camera may temporarly suffer from reduced visibility, or even get completely blind due to quick changes in enviromental illumination.

This paper presents a vision-based system that detects approaching tunnels entrances or exits. The proposed system allows other ADAS (*Advanced Driver Assistance Systems*) to act on camera parameters to effectively avoid the tunnel blindness effect. Information regarding approaching tunnel entrance can be helpful for other sensors as well and for sensor fusion systems. In terms of path planning, this system can also inform GNSS-based systems (*Global Navigation Satellite System*), which usually do not receive any signal in tunnels, and trigger dead reckoning techniques.

The proposed system is noticeably fast and therefore well fit to be used as a background process to support other ADAS applications.

Keywords: Intelligent vehicle, autonomous driving, ADAS, tunnel detection.

1 Introduction

This article presents a system for road tunnels detection in automotive scenarios. The recognition of road tunnels may bring important benefits to both automatic driving vehicles and driver assistance systems (ADAS).

Concerning autonomous driving, tunnel detection is primarily useful in two situations:

1. to adapt the camera dynamics in advance. In fact, cameras are generally suffering from abrupt changes in brightness, so the gain control system could be improved by the prediction of the next change in lightining conditions. In [5] a contrast invariant obstacle detection method is shown, which highlights the problem mentioned above.
2. To inform the path planner about the imminent lack of reception of satellite signals. These systems have major problems while the vehicle drives into tunnels [4], therefore the approach presented in this paper can fed the path planner with information regarding decisions about the use of dead reckoning techniques.

G. Maino and G.L. Foresti (Eds.): ICIAP 2011, Part II, LNCS 6979, pp. 424–433, 2011.

Concerning ADAS, the knowledge of an approaching tunnel entrance or exit location may be useful to enable certain vehicle behaviors without the driver intervention: turning lights on or off [3], or turning on or off the wipers, or interacting with other sub-systems related to driver comfort as instrument panel lighting, and more.

The developed system, based on the taxonomy presented in [1], may be placed in the category of assisted ADAS when it informs the driver, or semi automated ADAS when it delivers decisions to vehicle instrumentation.

This paper is arranged as follows: section 2 shows the algorithm architecture and the procedure to detect a tunnel entrance and exit. Results are presented in section 3 with some critical cases, and finally the conclusions will be drawn in section 4.

2 Algorithm Description

The algorithm receives as input a monocular grayscale image, then it performs a downsample procedure to reduce the computational load of the following stages.

The system can be represented as a finite state machine (FSM) with four states; for each image only one state is possible.

Figure 1 shows the finite state diagram architecture of the algorithm and its possible transitions. Each state of the FSM identifies the environment or situation met by the vehicle:

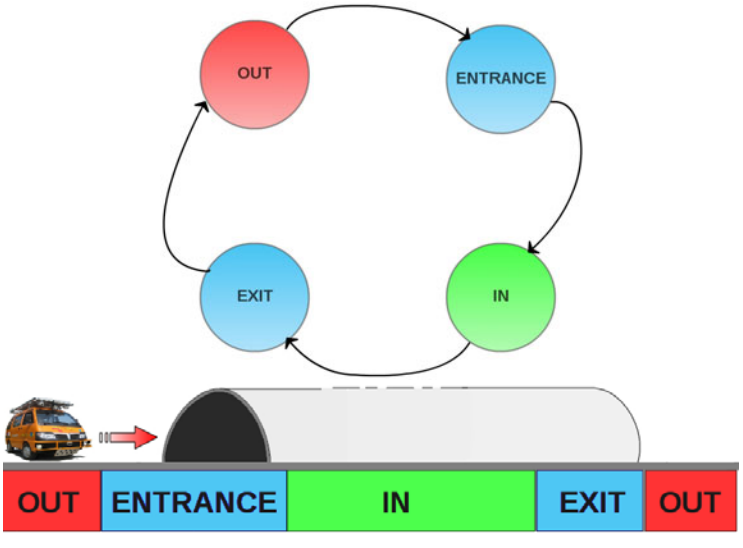


Fig. 1. Finite state machine diagram of the algorithm and representation of the various states

Fig. 2. Images related to FSM states with informative signals: out, entrance, in, exit

- *OUT* state: the vehicle is not inside a tunnel;
- *ENTRANCE* state: the vehicle is close to the entrance of a tunnel. In this state any procedure for entering in the tunnel can be activated, i.e. sending information about the imminent change of light conditions to the camera controller;
- *IN* state: the vehicle is inside the tunnel; the transition from outside to inside has ended.
- *EXIT* state: this is the opposite to the entrance state, from here the vehicle is approaching the tunnel exit and the system can inform the camera controller of the imminent change of light conditions.

Figure 2 schematically shows the four states and where they are activated during the passage in a tunnel.

2.1 Main Execution Flow

The main execution flow of the system mimics the FSM and, when the vehicle is travelling through a tunnel, leads the transitions amongst all states (see fig 3). Each frame is processed by a validation procedure that classifies the frame state and matches it against the current state.

Starting with the *OUT* state, the next one is *ENTRANCE*, in this case each frame is subject to a procedure to validate the entrance.

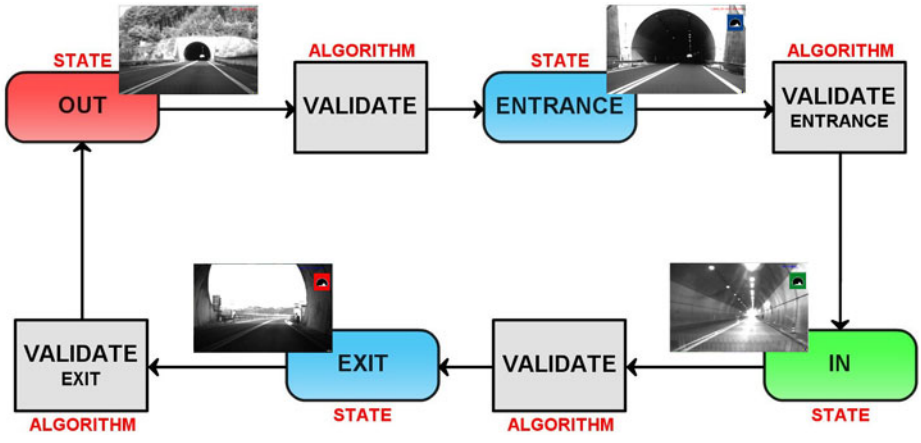


Fig. 3. State flow diagram of the system

If the system finds a positive entrance match, a new state is triggered and therefore the FSM enters the *ENTRANCE* state; the next state to be validated is the *IN* one, and so on.

This section describes algotithms and procedures used for the classification of each frame as shown in figure 3; the validate blocks are made up of a number of steps.

The validation procedures that are used for updating the FSM can be divided in two main classes:

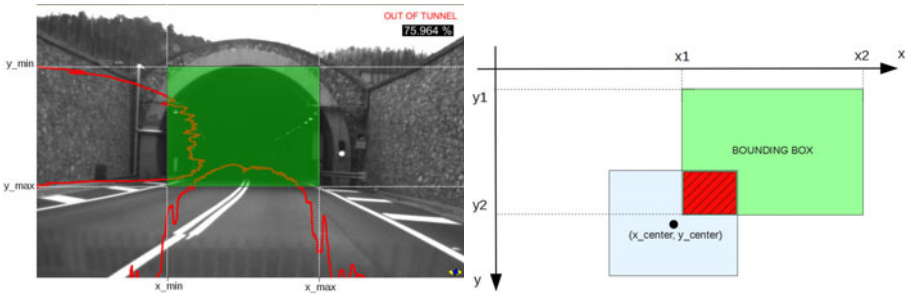
- *IN* and *OUT* validation: as explained in section 2.2, initially the system computes the horizontal and vertical histograms of dark pixels for the *OUT* state, or bright pixels for the *IN* state. Then the system validates the bounding boxes obtained by a histogram properties analysis.
- *ENTRANCE* and *EXIT* validation: in both transitions, the variance of histograms of black or white pixels is computed in an area of interest obtained from the previous state. For *EXIT* validation only, additional stages that involve image cropping and borders analysis are used. These two procedures are detailed in section 2.3 and 2.4.

2.2 Validation Algorithm for OUT and IN States

This section explains the algorithm performed in the *IN* and *OUT* states related to the *VALIDATE ALGORITHM* blocks in figure 3.

This procedure can be divided in three main steps:

- histograms analysis;
- bounding box construction;
- bounding box classification.



(a) (b)

Fig. 4. Bounding box construction and analysis. (a) Image showing the bounding box obtained from the analysis of the histogram of dark pixels. (b) Relationship between bounding box and its centered model.

Histogram analysis. In this step the algorithm builds the horizontal and the vertical histogram of the dark pixels, for the OUT state, and of the bright pixels for the IN state. These histograms give information about the numbers of the dark pixels, considering OUT state as example, for both rows and columns of the image. A dynamic threshold is used to select pixel values to be used for the histogram computation. This dynamic threshold is computed as follows:

$$Th = \begin{cases} th_{max} & if\ th \geq th_{max} \\ th_{min} & if\ th \leq th_{min} \\ value_{min} + P \cdot pixelValueSum & otherwise \end{cases} \quad (1)$$

where th_{max} and th_{min} are saturation values, $value_{min}$ is the minimum value that can have a pixel and the $pixelValueSum$ is an experimental computed value that is weighted using a P weight computed as follows:

$$P = \frac{|brightness_{avg} - value_{min}|}{255} \qquad P \in [0, 1] \quad (2)$$

The P value is proportional to the difference between the darkest pixel ($value_{min}$) and the brightness average ($brightness_{avg}$). The value of th_{max}, th_{min}, $value_{min}$, $pixelValueSum$ are provided as input.

Bounding box construction. The interesting portion of both horizontal and vertical histograms are calculated from the maximum peak with a sliding window according to a predetermined threshold. This analysis returns a bounding box that identifies the darkest area of the image. Figure 4.a shows a bounding box built using the dark pixels histogram values analysis.

Bounding box classification. In this step information obtained from the previous step are analyzed in order to validate the bounding box as a tunnel entrance or exit.

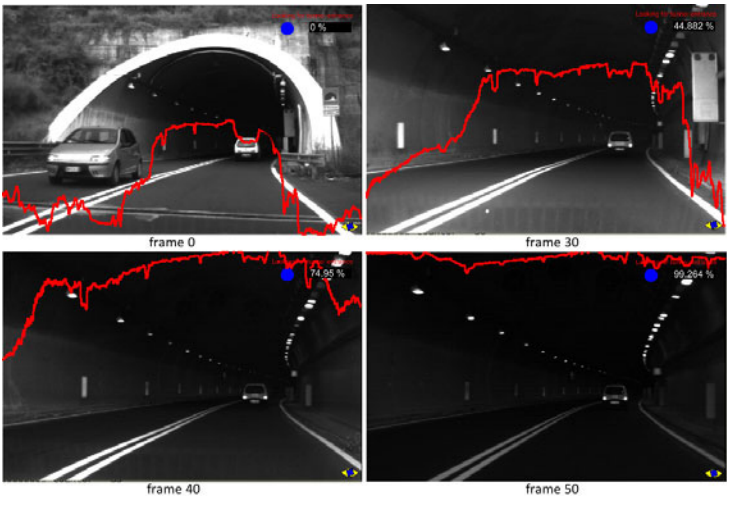


Fig. 5. Histogram of the dark pixels values in *ENTRANCE* most significant frames

Classification is made in according to the following formula, that returns a validation percentage $P_{validate}$:

$$P_{Validate} = \frac{W_1 \cdot P_{dimension} + W_2 \cdot P_{center} + W_3 \cdot P_{filter}}{W_1 + W_2 + W_3} \tag{3}$$

Where $P_{dimension}$ is the ratio percentage between box and current image, P_{center} is the percentage of overlapping with a centered bounding box model as shown in figure 4.b, that provides a measure of how the bounding box in the current frame is centered. This value should increase as vehicle approaches the extremities of a tunnel. The weights W_x have been empirically computed. P_{filter} is related to box filtering functions that concern the aspect ratio of the tunnel extremities, the size of the bounding box, and a fixed percentage added to the box classification if the current frame has a size greater than a given percentage of the size of the box found in the previous frame.

As an example, the $P_{validate}$ computed on the image in figure 4.a is shown in the upper right corner. To fire the state transition from *OUT* or *IN* states to the *ENTRANCE* or *EXIT* states, the average value of the $P_{validate}$ values computed for the last n frames is used.

2.3 Validation Algorithm for ENTRANCE State

This section explains the *VALIDATE ENTRANCE ALGORITHM* block, that is used to fire the transition from *ENTRANCE* to *IN* states. As shown in figure 5, the dark pixels histogram saturates when the vehicle enters a

(a) (b)

Fig. 6. Exit from a tunnel. (a) Filtered and binarized bounding box before the tunnel exit, (b) filtered and binarized bounding box after the tunnel exit.

tunnel, this effect can be exploited to identify the transition. The variance increases when the vehicle is approaching a tunnel entrance and then decreases until it reaches a local minimum when the vehicle has just entered the tunnel. A parameter P that encodes the variance decrement is used to fire the transition. More precisely, P is computed as follows:

$$P = 1.0 - \frac{var}{varMax} \tag{4}$$

The value of P proportionally increases with respect to the decrement of the variance. P is compared against a threshold to fire the transition. Since the *ENTRANCE* has been activated, the variance value is computed, when it reaches the maximum, P start to increase.

Figure 7 shows the evolution of the variance in the entrance of a tunnel, the frame 0 indicates the activation of the *ENTRANCE* state; while the frame 40 represents the instant of the change of the state in which the vehicle is completely into the tunnel.

2.4 Validation Algorithm for EXIT State

In this step the procedure used to validate the exit from a tunnel is explained. This procedure is formed by two key point, an edge filtering procedure and the analysis of a histogram of the edges.

The bounding box obtained by the *IN* state validation block is filtered using a Sobel filter in order to extract the edges. In the transition from inside to outside the tunnel the presence of edges increases due to the change of camera controller parameters as shown in figure 6. The filtered box is binarized and a histogram of the edge pixels is computed for each box column. This histogram is averaged on the last n frames computing therefore the average of the number of edge pixels. The average is compared with a threshold to determine the state transition. Figure 8 shows the temporal histogram average in the *EXIT* state.

3 Results

The system was tested on a computer with the following features:

- Cpu: Intel(R) Core(TM)2 Quad CPU Q9550 @ 2.83GHz;
- Ram memory: 8 GiB DIMM DDR2 Synchronous 800 MHz (1.2 ns);
- Video Card: GeForce 9800 GTX/9800 GTX+ 512 MB memory.

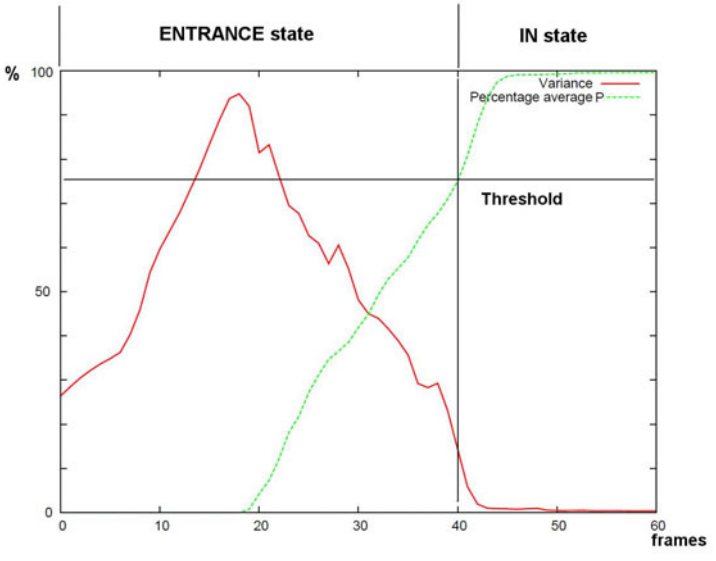


Fig. 7. Black pixels histogram variance and percentage average on frames in the *ENTRANCE* state

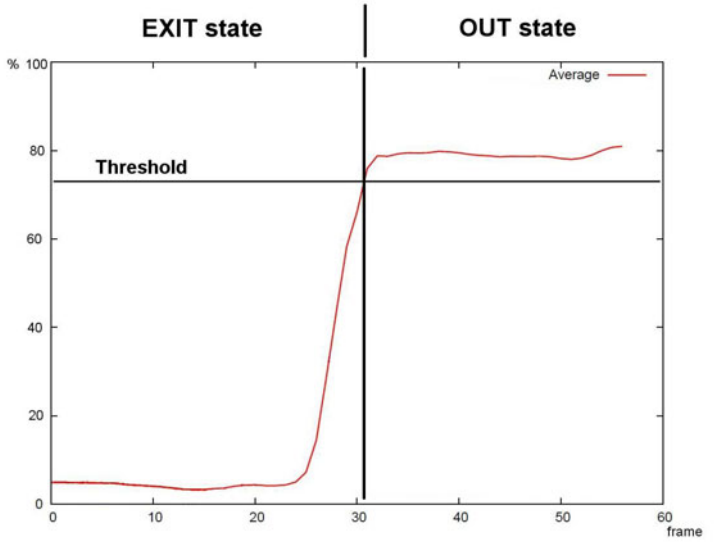


Fig. 8. Edge count average evolution from *EXIT* to *OUT* states

Table 1. Results of the algorithm

Sequence	% Correct Detection	Transitions	False Positives [frames]
first	100.0	80	13
second	83.3	48	0
third	100.0	32	350
total	94.4	160	363

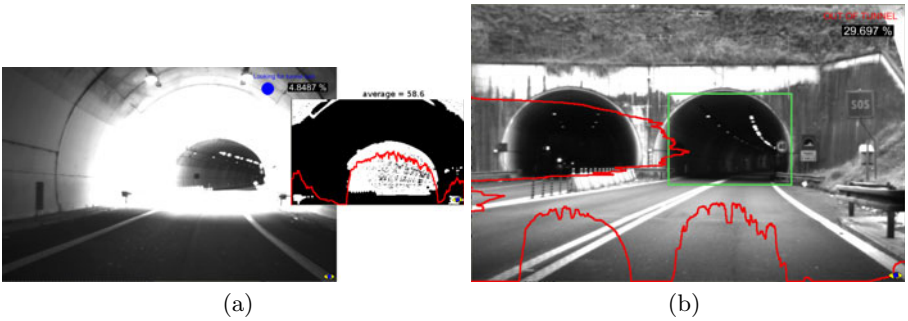
Three different sequences were used in the test, with 54686 frames in total at a frequency of 21 FPS. The image sequences were acquired during the day with frequent tunnels of varying length, with several entrances and exits, and normal traffic conditions. Tunnels have different shapes, with rectangular or arched entry and they are in urban, rural streets, and in highways. The images were acquired by a non calibrated camera with variable gain with a sensor in the visible spectrum with a NIR tail.

The system was proven to be able to detect the correct state with a total percentage of correct detection of 94.4% (see table 1). Missdetections appear in the case of urban settings. In this situation, the tunnel is just after a curve between the buildings of the city center, so the developed algorithm can not locate the entrance due to abrupt scenario variations.

In image sequences with the presence of trees along the road inducing sudden brightness changes, the system presented a limited number of false positives.

While it is actually not a missdetection, in 16.2% of the cases the tunnel exits are triggered in advance with respect to the ground truth due to strong light reflections produced by tunnel walls.

A further key parameter to judge the system performance is execution time. In fact, the system is meant to be run as a background process in conjunction with other ADAS functions. The application is able to process one frame at an average execution time of 750 μseconds, namely in less than 1 ms.



(a) (b)

Fig. 9. Problematic cases. (a) Exit from a tunnel: histogram mean are high because of the presence of another tunnel. (b) Twin galleries. The histogram have two relevant portions.

3.1 Problematic Cases

In this section two problematic cases are presented; this situation concerns consecutive road tunnels and twin frontal tunnels. In both situations system the behaves correctly and perform a correct tunnel detection.

Consecutive road tunnel. The presence of a tunnel entrance close after the exit of the current tunnel weakens the exit footprint in the image. This effect could lead to delayed detection in some cases. The above situation is shown in figure 9.a.

Twin tunnels. A frequent situation that can be especially encountered on motorways, is shown in figure 9.b. Two tunnels, one for each direction, create problems in identifying the histogram peak to detect the entrance bounding box. This problem has been solved giving priority to the central part of the histogram during the search for the largest portion, assuming that the tunnel of interest is in a central position and therefore in the central part of the image. The use of lane markings is also being considered.

4 Conclusions

The system for the recognition of road tunnels presented in this article has proven to work with a sufficiently high robustness.

The system was tested on different sequences including different tunnel situations: rural roads, highway tunnels, downtown with surrounding buildings, twin tunnels, consecutive tunnels, tunnels with several types of lighting or no lighting at all, tunnels with differently shaped entrances.

The application demonstrated to be able to recognize these tunnels and is not affected by traffic conditions at the tunnel entrance or exit.

The system can run in 750 μseconds and therefore is suitable to be used in conjunction with other ADAS systems.

References

1. Broggi, A., Mazzei, L., Porta, P.P.: Car-driver cooperation in future vehicles. In: Procs. Intl. Conf. on Models and Technologies for Intelligent Transportation Systems, Rome, Italy (June 2009)
2. Daytime Running Lights Deliverable 3: Final Report (October 2003)
3. D.M. 5 giugno,"Sicurezza nelle gallerie stradali". Pubblicato nella Gazzetta Ufficiale (217) (Settembre 18, 2001)
4. Ziemer, R.E., Peterson, R.W.: Introduction To Digital Communication, 2nd edn. Prentice-Hall, Englewood Cliffs (2000)
5. Cabani, I., Toulminet, G., Bensrhair, A.: Contrast-invariant Obstacle Detection System using Color Stereo Vision. In: 11th International IEEE Conference on Intelligent Transportation Systems, ITSC 2008, Beijing (October 2008)