

Obstacle Detection for Start-Inhibit and Low Speed Driving

M. Bertozzi, A. Broggi, P. Medici, P. P. Porta and R. Vitulli

*Dipartimento di Ingegneria dell'Informazione

Università di Parma
Parma, I-43100, Italy

Abstract—The work described in this paper has been developed in the framework of the Integrated Project APALACI - PReVENT, a research activity funded by the European Commission to contribute to road safety by developing and demonstrating preventive safety technologies and applications.

The goal of the system presented in this work is the development of a vision system for detecting potential obstacles in front of a slowly moving or still vehicle.

When the vehicle is still, a background subtraction approach is used assuming that the background keeps stationary for a limited amount of time; thus, a reference background is computed and used to detect changes into the scene.

A different approach is used when the vehicle is moving. The system, by means of inertial sensors, can detect ego-motion and correct background information accordingly. A temporal stereo match technique, able to detect obstacles in moving situations, completes the system.

According to experimental results, the proposed algorithm can be useful in different automotive applications, requiring real-time segmentation without assumptions on background motion.

I. INTRODUCTION

The second largest source of automotive related injuries or fatalities is due to accidents that involve pedestrians. Systems that mitigate pedestrian accidents, i.e. external airbags or start inhibit, are being considered by all car manufactures; they require an accurate knowledge about pedestrian presence for a correct activation.

In addition, since around two thirds of all accidents happen in the frontal region of the vehicle, a large number of systems is based on the automatic detection of obstacles in front of the vehicle.

A typical method for detecting obstacles when the camera is not moving, is segmentation based on background subtraction [1], a simple but effective method for detecting moving objects by comparing the intensity of the background and observed images. It requires computing a reference image (background), subtract each new frame from this image, and threshold the result. The final result is a binary segmentation of the image, in which non-stationary objects are highlighted.

The simplest way to obtain the reference image is to perform an average amongst a number of subsequent images. Unfortunately, this approach can be effective only when the background is not varying amongst frames, namely a static background. This constraint strongly limits the effectiveness of such method for vision systems installed on a moving vehicle.

Algorithms for stationary applications (security surveillance, human-machine interface, low-bandwidth telecommunications and others) use several techniques to model the background. Existing methods for background modelling may be classified as predictive, probabilistic, or other approaches. Probabilistic algorithms model every background pixel (or image block) as an independent statistical process, neglecting the order of the input observations. In some cases [2], models of background changes are built working on image blocks and drawing from each block information about covariance, generating eigenspace and eigenvector. In other cases [1], [3], [4], the distribution of the intensity value is fitted with a Gaussian distribution (for instance evaluating mean and variance); a Gaussian Mixture Model is used to model the background, while the foreground is modelled by additional Gaussian models. Predictive class algorithms try to track some feature of the image (by means of Kalman filter [5], [6]) to detect the changes in background for every pixels using edge features or intensity values. The estimation of the pixels motion permits to avoid problems due to small changes in background but can be heavily affected by noise. More recent methods are based on more sophisticated models using historical data to predict future evolution as autoregressive models.

In this article, a real-time system for detecting obstacles using a monocular camera is described. The system can access information about the vehicle's state, like speed or yaw-rate. Using this information it is possible to detect motion or stationary scenarios and adapt the system behavior.

When the vehicle is still, a Background Subtraction approach is used assuming that the background is stationary for a limited amount of time; thus, a reference background image is computed and used to detect changes into the scene. When the vehicle is moving, the system exploits a Temporal Inverse Perspective Mapping (IPM) technique: using calibration data and inertial information, the previous frame is warped onto the current one: still or moving obstacles are *miswarped* thus allowing the detection [7].

This paper is organized as follows: section II presents an overview of the system, section III discusses distortion removal, section IV introduces all parts of the algorithm and presents the results obtained, while section V summarizes remaining problems to be solved and concludes the paper.

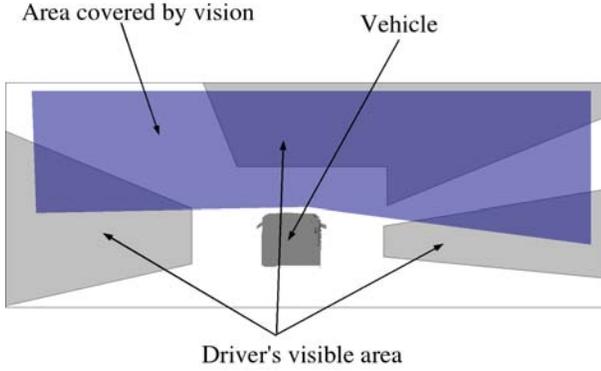


Fig. 1. Driver's blind spots and area covered by the camera.

II. SYSTEM OVERVIEW

The goal of the APALACI project is to develop pre-crash and collision mitigation automotive applications, including the development of innovative sensor fusion techniques. More specifically, it addresses the area close to the vehicle. This includes sensors for pedestrian recognition and methods for distinguishing among different obstacles i.e. vulnerable road users, vehicles, barriers, or other objects.

Amongst the others, two main topics are addressed in this work:

- a Start Inhibit function should prevent truck drivers from accelerating from stationary, when pedestrians or other obstacles are present in the blind spot area (see figure 1), in particular in front of the truck.
- The recognition of obstacles in slow speed situations, by monitoring the blind spot ahead of the vehicle with the aim of informing the driver about pedestrians close to the vehicle or preparing the vehicle to an unavoidable collision (i.e. reducing speed, engage external airbag, retracting the seat belt, and so on).

Two different sensors are used in the system:

- a single color CMOS camera linked to a video framegrabber on a PC;
- an off-the-shelf sensor capable to feed information about speed and yaw rate through the vehicle's CAN bus.

The vision system is installed on a truck, framing the area in front of the vehicle. Since the system needs to cover a quite large area, a camera with a sufficiently large field of view (about 90 degrees) is required. This implies a short focal length and a heavy distortion caused by a fish-eye lens.

III. DISTORTION REMOVAL

The first phase of the processing is the removal of lens distortion.

Infinite series [8] are needed to fully model non-linear lens distortion. The approach used in the system does not try to discover the lens model, but rather generates a lookup table capable of directly transforming a distorted image to an undistorted one. In this way the distortion can be removed

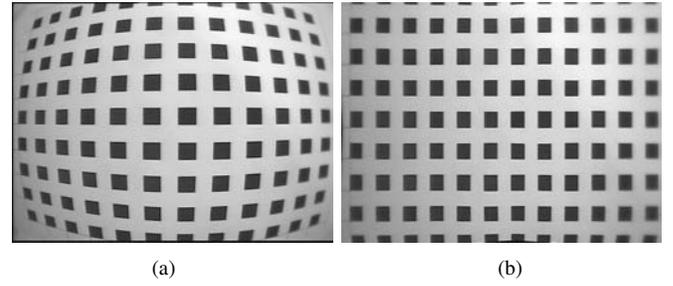


Fig. 2. Distortion removal: (a) acquired image and (b) image without distortion.

without using any model, and therefore without any limitation that may be introduced when high level approximations are neglected. Furthermore, it also favorably affects the execution time.

The lookup table is used to map undistorted coordinate (x^u, y^u) to distorted ones (x^d, y^d) :

$$(x^d, y^d) = f_{LUT}(x^u, y^u) \quad (1)$$

Each pixel of the undistorted image (the resulting image) needs to have a correspondent in the distorted image (the input image acquired from the camera).

To generate this function an off line processing is used. The image of a calibration grid (see figure 2.a) is acquired using the camera-lens system that needs to be profiled. The grid is composed by equally spaced squares that, thus in the undistorted image the squares have to appear equally spaced as well (see figure 2.b). In the first phase of the processing, feature points P_i of the distorted image are computed through a flood-fill algorithm. A sparse lookup table is therefore preliminarily computed in order to map the positions of pixels P_i towards their new position in the undistorted image:

$$(x_i^d, y_i^d) \leftrightarrow (x_i^u, y_i^u)$$

In this way, only few values of the lookup table are computed. The values for pixels other than P_i are obtained interpolating with third order polynomials splines.

IV. ALGORITHM

The general flow of the system is shown in figure 3; a *decisor* examining the video stream selects which algorithm flow should be followed according to the vehicle state.

When the vehicle is not moving, a background subtraction approach is used assuming that the background is stationary for a limited amount of time; a reference background image is computed and used to detect changes into the scene.

Conversely, in motion situations, the system exploits a technique that we named Temporal IPM: using calibration data and inertial information, it is possible to warp the previous frame onto the current one. The positions in which the warped image and the current one differ localize the presence of obstacles.

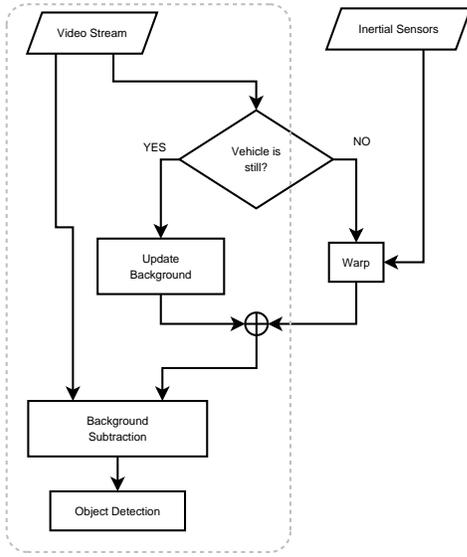


Fig. 3. Overall system flow.

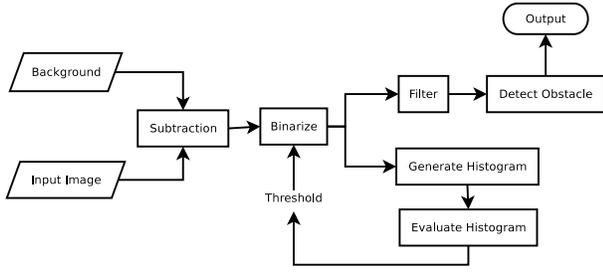


Fig. 4. Background subtraction flowchart.

A. Background Subtraction

The background suppression technique is based on the subtraction of a reference background image from the input image. According to diagram shown in figure 4, the proposed method is divided into four main stages: in the first stage a reference background is obtained. In the second phase, the input image and the background are compared, while in the third step a dynamic threshold used for the detection is computed. In the last step obstacles are detected.

1) *Reference Image Generation:* The background reference images are generated using the most recent input images, through the procedure indicated in figure 5.

Let B_n be the background image at time n and I_n the input image at time n . $B_n(i)$ and $I_n(i)$ are the values for generic pixel P_i in the background scene and image respectively at time n . w is the parameter used to weight how the new image influences background model. Each pixel i of the background image at frame n is updated, so that the older the frame the smaller its weight in the reference background image:

$$B_{n+1}(i) = B_n(i) \times (1 - w) + I_n(i) \times w \quad (2)$$

This digital low-pass filter reject artifacts introduced by vehicle vibration and video signal noise, and requires very small

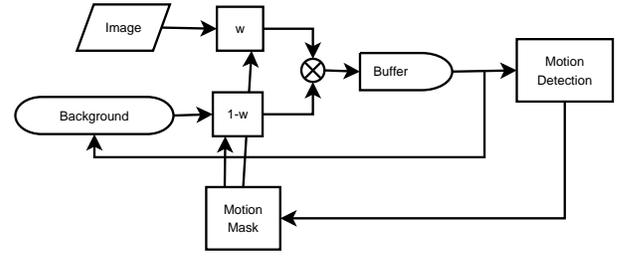


Fig. 5. Background construction flowchart. Input image at frame n is mixed with background for the same frame using weight w . Background for frame $n + 1$ is generated. The weights for every pixels are provided using the output of algorithm itself.

amount of memory and computational resource

Another approach[?] for generating a background model can be based on an average amongst the latest N frames; this method requires more memory to store every frame and does not allow the use of the foreground rejection technique described in the following paragraphs.

The bandwidth of the low pass filter can be changed acting on the value of w in equation (2). A low value of w permits to avoid most noise but requires time to create the background. Conversely, with a high value of w , background can be quickly adapted to abrupt changes but moving objects can erroneously be incorporated within the background model.

The most critical problem of this approach is, in fact, the injection of non background objects in the reference image during the update. This problem can be avoided using, as a feedback, the result of algorithm itself: moving objects are excluded from the reference image update. As mentioned above, the larger the weight w , the higher the influence of the correspondent pixel $I_{n-1}(i)$. Thus a way to implement rejection of moving objects is to use different values for w for different image pixels according to output results. To achieve this, w should become a matrix whose elements describe the influence of every pixel of I_n in the updating of background.

Experimental results showed that two values, w_f and w_b , suffice for updating the reference image:

$$w(i) = \begin{cases} w_f & i \in foreground \\ w_b & i \in background \end{cases} \quad (3)$$

Since information about the foreground and background is not available, the output supplied by the algorithm itself is used. In such way, $w(i)$ has the meaning of *background probability*. A low value in $w(i)$ corresponds to a low probability that pixel i belongs to the background while a high value of $w(i)$ means a high probability that i belongs to background and therefore it should be taken into account for background update.

Thus, equation 2 can be written, more generally, as

$$B_{n+1}(i) = B_n(i) \times [1 - w(i)] + I_n(i) \times w(i) \quad (4)$$

Figure 6 shows the results of the two different methods for background image generation. In the background image obtained using a constant w the trace of a foreground object

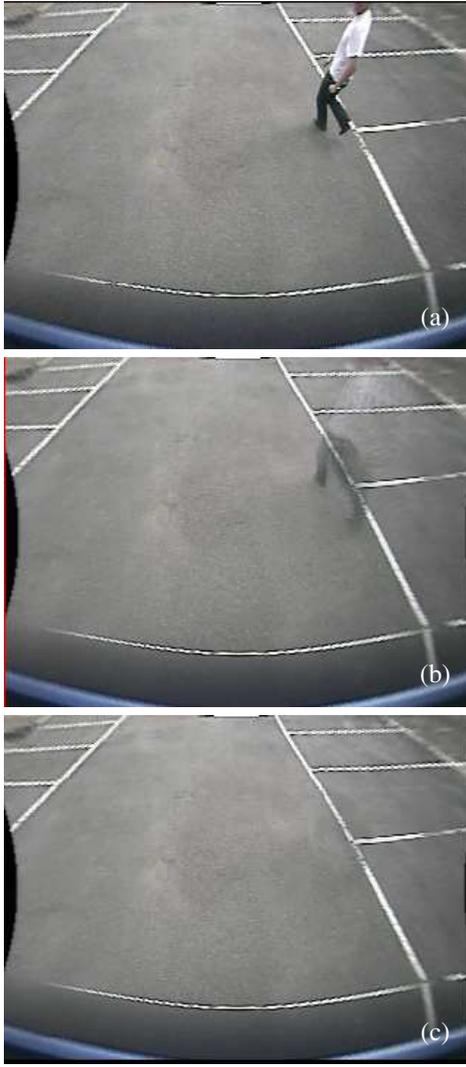


Fig. 6. Comparison of different strategies for reference image generation: (a) acquired image after distortion removal, (b) background obtained using a constant w or (c) using a varying w .

is visible (figure 6.b); conversely, the same objects is barely observable in the reference image obtained feeding the update with algorithm results, namely using a variable w (figure 6.c).

2) *Background Removal*: The obtained background reference image is matched against the most recently acquired image for detecting obstacles; namely, the difference image δ between pixels values of both images is computed.

Since the acquired images are defined in the RGB color space, their pixels can be represented by a vector. It is straightforward the use of the Euclidean distance operator for computing the difference δ between the colors of background pixels ($B(i)$) and foreground ones ($I(i)$):

$$\delta(i) = \|\vec{I}_n(i) - \vec{B}_n(i)\| \quad (5)$$

namely:

$$\delta(i) = \sqrt{(r-R)^2 + (g-G)^2 + (b-B)^2} \quad (6)$$

using the naming convention $\vec{I}_n(i) = \{r, g, b\}^T$ and $\vec{B}_n(i) = \{R, G, B\}^T$.

Unfortunately, preliminary results showed that this approach is too much affected by the luminance of pixels, therefore the Vector Angle [9] has been used for computing $\delta(i)$:

$$\delta(i) = \frac{\vec{I}(i)^T \times \vec{B}(i)}{\|\vec{I}(i)\| \|\vec{B}(i)\|} \quad (7)$$

In such a case, the difference $\delta(i)$ between two pixels' colors is obtained as the angle formed by the two color vectors. This angle, in fact, does not depend on the module of the two vectors and, thus, it is not affected by the pixels luminance.

A threshold ξ is used to binarize the distance image $\delta(i)$:

$$X_n(i) = \begin{cases} foreground & \delta(i) > \xi \\ background & \delta(i) < \xi \end{cases} \quad (8)$$

X_n is a binary image representing foreground and background. The threshold value ξ is critical for the whole system, since it allows to detect the presence of obstacles; a dynamic tuning of this parameter is used to improve the effectiveness of the system.

3) *Threshold tuning*: Once a first output is provided by the system, it is necessary to adjust the threshold matrix used for the foreground/background discrimination.

A vertical and a horizontal histogram are computed counting foreground pixels for each row and column. A Gaussian low pass filter is applied to the histograms in order to remove noisy components and merging close areas.

These two histograms strongly depend on the number and size of foreground areas in X_n . Namely, they reach high values when a large amount of foreground is detected in X_n , while are null when no foreground is detected.

Threshold ξ is lowered when no foreground is detected and increased when too much foreground is detected: a constant threshold γ is applied to the vertical and horizontal histogram and the percentage \ni of overthreshold values of the two histogram is computed. When \ni is bigger than a given threshold γ , ξ is lowered, conversely, ξ is increased if \ni is smaller than γ .

In such way the threshold ξ is tuned according to system results in order to adapt to changing environments.

4) *Obstacle Detection*: The binary image X_n used to generate histograms, is yet a raw result. This preliminary output is affected by a great amount of noise and need an additional filtering.

Morphological filters are used for image enhancement. A first filter removes isolated pixels, simply discarding every pixel in the image that does not have, in the 3×3 neighborhood, any other overthreshold pixels. A second filter applies a morphological expansion to the image using a 5×5 neighborhood to obtain contiguous clusters of overthreshold pixels.

A further processing detect contiguous areas producing – as final output – bounding boxes that enclose the clusters of overthreshold pixels.

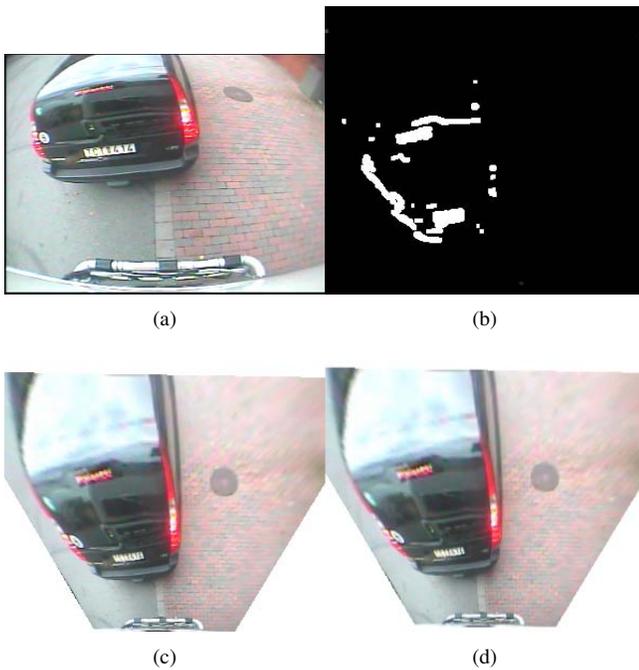


Fig. 8. Temporal IPM: (a) current frame, (b) binarized output image is obtained as difference between (c) remapped current frame and (d) remapped and roto-translated previous frame.

5) *Results:* The discussed system has been tested on several 320×240 pixels color images acquired by a camera installed on a truck and framing the area of the road close to the vehicle.

The test sequences comprise different urban and extra-urban scenarios and contain both moving and still situations.

Figure 7 shows both partial and final results of the background subtraction processing. The first and second rows present a case in which an obstacle (a pedestrian) enters the scene and rapidly moves away; in such case the background image is barely affected by the presence of a foreground object.

Conversely, in the last row a pedestrian stopped and crouched in front of the vision system being incorporated in the background reference image. Anyway, as soon as he starts moving, he is detected as an obstacle and the background reference image slowly updates to his removal.

Working on color images [1], [10] allows the use of Vector Angle technique that proved to be less prone to luminance variation, in particular to shadows.

The system has proven to obtain good results. Anyway, due to the specific camera position, only part of most pedestrians is detected. In addition, vibrations due to the engine of the truck have demonstrated to be a strong source of noise for the detection leading to blinking false positives.

Background subtraction and obstacle detection run at 30 Hz on a 3.2 GHz Pentium 4.

B. Temporal IPM

In motion situations, the background subtraction technique is of no use, since a fast evolving background is present. In such a case a different approach is used: the *Temporal IPM*.

Using the assumption of a flat road in front of the vision system and thanks to the knowledge of camera parameters, the perspective effect can be removed from acquired images (figure 8.a) producing remapped images (figure 8.c). The remapped images contain a bird's eye view of the road [11].

Thanks to the inertial sensors, it is possible to know the speed and yaw of the vehicle. These parameters can be used to bring to correspondence two consecutive remapped frames roto-translating the remapped previous frame with respect to remapped current frame (figures 8.c and 8.d).

The so-remapped images are different only where something other than road surface (namely an obstacle) is framed. Therefore a simple pixel-wise comparison can be used for obstacle detection producing a difference image that highlights the presence of obstacles (figure 8.b).

This approach is currently under further development. Preliminary results show that this method is promising and effective for vehicle's speed up to 40 km/h.

V. CONCLUSIONS

In this paper an approach to obstacle detection for start-inhibit or slow speed situations has been presented.

A single color camera has been installed on a truck monitoring the driver's blind area in front of the truck up to few meters ahead.

When the truck is not moving, a background subtraction approach is used to detect obstacles. The background is assumed as stationary for a limited amount of time, a reference background image is computed and compared against acquired images. A strong difference between acquired and background reference image detects an obstacle. This approach is useful to inhibit the start when obstacles are present in front of the vehicle.

A second approach is used when the truck is moving. The system, by means of inertial sensors, can detect moving situations and compute its ego-motion. A temporal stereo match technique is currently under development to detect obstacles in such case.

Current work is aimed at the development of a system for distinguishing among different obstacles i.e. vulnerable road users, vehicles, other objects/barriers.

REFERENCES

- [1] D. H. A. Woo, "A Background Subtraction for a Vision-based User Interface," in *Procs. IEEE Joint Conf. on Information, Communications and Signal Processing and Pacific Rim Conf. on Multimedia*, vol. 1, Singapore, Dec. 2003, pp. 263–267.
- [2] M. Seki, H. Fujiwara, and K. Sumi, "A Robust Background Subtraction Method for Changing Background," in *Procs. IEEE Intl. Workshop on Applications of Computer Vision*, Palm Springs, CA, USA, Dec. 2000, pp. 207–213.
- [3] E. Hayman and J.-O. Eklundh, "Statistical Background Subtraction for a Mobile Observer," in *Procs. IEEE Intl. Conf. on Computer Vision*, Nice, France, Oct. 2003, pp. 67–74.
- [4] Y. Ren, C.-S. Chua, and Y.-K. Ho, "Motion Detection with non-Stationary Background," in *Procs. IEEE Conf. on Image Analysis and Processing*, Palermo, Italy, Sept. 2001, pp. 73–83.
- [5] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Trans. ASME Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

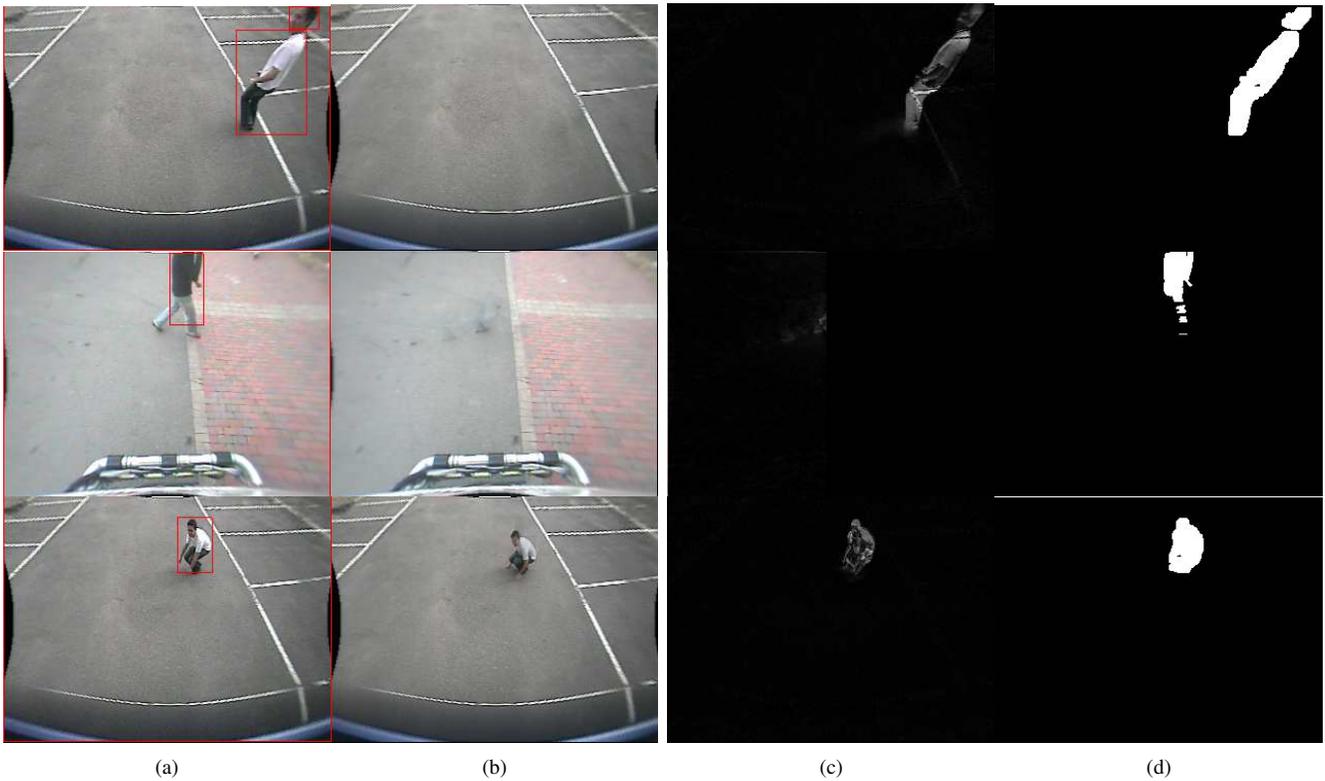


Fig. 7. (a) final result, (b) background model, (c) difference image δ , and (d) binarized difference image after morphological filtering.

- [6] D. Koller, J. Weber, T. Huang, J. Malik, G. G. Ogasawara, B. Rao, and S. Russell, "Towards robust automatic traffic scene analysis in real-time," in *Procs. IAPR Conf. on Computer Vision & Image Processing*, Oct. 1994, pp. 126–131.
- [7] R. Alix, F. L. Coat, and D. Aubert, "Flat world homography for non-flat world on-road obstacle detection," in *Procs. IEEE Intelligent Vehicles Symposium 2003*, Columbus, USA, June 2003, pp. 310–315.
- [8] J. Wend, p. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 65 – 980, Oct. 1992.
- [9] S. Wesolkowski, R. Dony, and M. E. Jernigan, "Global Color Image Segmentation Strategies: Euclidean Distance vs. Vector Angle," in *Procs. IEEE Workshop on Neural Networks for Signal Processing*, Madison, WI, USA, Aug. 1999, pp. 419–428.
- [10] H. Han, Z. Wang, J. Liu, Z. Li, B. Li, and Z. Han, "Adaptive Background Modeling with Shadow Suppression," in *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems 2003*, Shanghai, China, Oct. 2003, pp. 720–724.
- [11] M. Bertozzi, A. Broggi, and A. Fascioli, "Stereo Inverse Perspective Mapping: Theory and Applications," *Image and Vision Computing Journal*, vol. 8, no. 16, pp. 585–590, 1998.