

# A Maximum Likelihood Approach to Speed Estimation of Foreground Objects in Video Signals

Veronica Mattioli, Davide Alinovi and Riccardo Raheli  
*Department of Engineering and Architecture*  
*University of Parma*  
Parma, Italy  
{veronica.mattioli, davide.alinovi, riccardo.raheli}@unipr.it

**Abstract**—Motion and speed estimation play a key role in computer vision and video processing for various application scenarios. Existing algorithms are mainly based on projected and apparent motion models and are currently used in many contexts, such as automotive security and driver assistance, industrial automation and inspection systems, video surveillance, human activity tracking and biomedical solutions, including monitoring of vital signs. In this paper, a general Maximum Likelihood (ML) approach to speed estimation of foreground objects in video streams is proposed. Application examples are presented and the performance of the proposed algorithms is discussed and compared with more conventional solutions.

**Index Terms**—Maximum likelihood, foreground detection, speed estimation, video signals.

## I. INTRODUCTION

Motion and speed estimation can be necessary processes in order to recover specific real-world related information arising from the movement of an object in a three-dimensional (3D) scene captured by an image acquisition system, e.g. a camera. When a video is captured, a 3D real-world scene is projected onto the two-dimensional (2D) camera plane, resulting in a sequence of digital images. Hence, motion in a video stream can be considered as the consequence of the projection of objects moving in the 3D scene [1]. This phenomenon may arise when objects are moving and the camera is still, when the camera is moving and the objects are still or when both are moving.

Various motion estimation algorithms are widely documented in the literature, as reported, e.g., in [1], [2] and [3, Ch. 4]. In particular, the most popular techniques include differential methods and matching methods.

Differential methods aim to estimate the optical flow, defined as the apparent motion perceived from the variations in the pixel intensity patterns in the 2D image, that may be due either to a true motion in the 3D space or to illumination changes [4]. These approaches are based on gradient techniques that exploit the properties of spatial and temporal partial derivatives. The most popular solutions in this category are the Lucas-Kanade [5] and the Horn-Schunck methods [6], that rely on the brightness constancy model and the smoothness constraint of the velocity vector [4], respectively. Due to these specific assumptions, the two methods may fail in properly describing motion in some realistic scenarios, which may represent their main limitation.

Block matching approaches, the most widely employed matching methods, are based on the partition of the considered video frame into several blocks of pixels, that may or may not overlap, where each block is associated with a motion vector [3, Ch. 4]. Block matching criteria, despite being straightforward and basic, are highly sensitive to noise and parameter setting, especially the block size. As a consequence, the estimated motion vector may not coincide with the true motion. In this category, algorithms to estimate the mean speed of a group of vehicles from Motion Pictures Experts Group (MPEG) video streams are presented in [7] and [8], where motion vectors are directly extracted from the considered stream.

To avoid problems related to differential and matching methods, we wish to investigate the application of a more fundamental estimation principle, namely the Maximum Likelihood (ML) approach [9], to estimate the speed of a foreground object in considered video sequences.

The remainder of the paper is organized as follows. In Section II the motion estimation model of a framed object in a video sequence is described and a brief discussion on background removal techniques is presented. The ML estimation approach is proposed in Section III. The performance of the proposed algorithms is discussed in Section IV on the basis of several experimental results. Finally, conclusions are drawn in Section IV.

## II. OBSERVATION MODEL

### A. Preliminary Definitions

A video can be defined as a time sequence of digital images, which are referred to as frames, whose spatial-intensity pattern may vary over time. It is hence a time varying multidimensional signal where two spatial components identify a pixel position within the frame, which can also be thought of as a two-dimensional image (per colour component) formed by projecting a real world 3D scene onto a 2D image plane. The motion and variation of an object in the real world correspond to changes in the pixel intensity values in the 2D image plane. Motion estimation is therefore a process that allows to recover real-world related information by analysing the time evolution of the framed area.

Before presenting the formulation of the motion model and discussing the ML approach to speed estimation of a framed

foreground object, the following simplifying assumptions are introduced:

- only framed areas including a single moving object are considered;
- the capturing camera is still;
- object transformations due to perspectival issues arising from the projection of the 3D real world scene onto the 2D image plane are not accounted for.

The first condition allows to simplify the model and test the effectiveness of the proposed algorithm in the simplest possible scenario. The camera is considered still to avoid superposed motions other than the considered object motion. Finally, the last condition assumes that no perspectival transformation affects the object. These simplifying assumptions allow us to concentrate on the main goal of this paper, which is motion estimation, and will be relaxed by further investigation currently undergoing.

### B. Object Motion Model

A video signal is composed by a set of frames sampled at instants  $nT_s$ , where  $n$  is the discrete time index corresponding to the frame number, and  $T_s$  is the sampling time, with  $f_s = 1/T_s$  being the sampling frequency, or frame rate, of the camera. A digital grayscale video signal can be described as a two-dimensional discrete function  $x[\mathbf{m}, n]$ , where  $\mathbf{m} = (m_1, m_2)^T$  indicates the pixel position within the frame,<sup>1</sup> with  $0 \leq m_1 \leq M_1 - 1$  and  $0 \leq m_2 \leq M_2 - 1$ , whose value corresponds to the pixel intensity. In particular,  $M_1 \times M_2$  is the frame size, being  $M_1$  and  $M_2$  its height and width, respectively. As customary, the dynamic range of the pixel intensity is limited to the interval  $[0, 1]$ . In the case of coloured videos, a fourth dimension can be added to this function in order to specify the colour channel index.

Considering a framed object shifting with a simple translation in the two-dimensional projected camera plane, its motion can be described by a displacement vector  $\boldsymbol{\delta}[n] = (\delta_1[n], \delta_2[n])^T$ , with horizontal ( $\delta_1[n]$ ) and vertical ( $\delta_2[n]$ ) components. Defining  $s[\mathbf{m}]$  as the image of the still object of interest, the pixel intensities of a video stream can be modelled as the summation of a few main elements [10]:

$$x[\mathbf{m}, n] = b[\mathbf{m}] + s[\mathbf{m} - \boldsymbol{\delta}[n]] + v_b[\mathbf{m}, n] + w[\mathbf{m}, n] \quad (1)$$

where  $b[\mathbf{m}]$  is the still background, the object  $s[\mathbf{m} - \boldsymbol{\delta}[n]]$  is shifted with a displacement  $\boldsymbol{\delta}[n]$ ,  $v_b[\mathbf{m}, n]$  is a term which takes into account the occluded/un-occluded parts of the background and  $w[\mathbf{m}, n]$  are samples of independent and identically distributed (i.i.d.) zero-mean Gaussian noise. The background is assumed static, i.e., time-invariant. Equation (1) can be considered as the most general model to describe a single framed moving object in a video stream. This model applies directly to a group of objects subject to the same displacement.

Consider now the special case of constant speed for further simplification. The displacement term can be explicitly written

<sup>1</sup>T denotes the transpose operator.

as  $\boldsymbol{\delta}[n] = \mathbf{v}n$ , where  $\mathbf{v} = (v_1, v_2)^T$  is the uniform speed motion vector in pixel/frame that we wish to estimate. Hence, the observation model in (1) can be rewritten as:

$$x[\mathbf{m}, n] = b[\mathbf{m}] + s[\mathbf{m} - \mathbf{v}n] + v_b[\mathbf{m}, n] + w[\mathbf{m}, n] \quad (2)$$

where the direction and the speed of the object displacement have been considered time-invariant for the sake of simplicity. This model can be easily applied to a slowly time varying scenario, provided  $\mathbf{v}$  can be considered approximately constant over a sufficiently long time window.

Models (1) and (2) are formulated in the time domain. Using a frequency domain representation may result more convenient, as discussed in [10]. The main advantage of operating in the frequency domain relies on the properties of the Fourier Transform (FT). In particular, thanks to the shift theorem, it is possible to describe a displacement in the time domain as a linear phase term in the frequency domain. Hence, the dependence of  $s[\mathbf{m} - \mathbf{v}n]$  on the parameter  $\mathbf{v}$  can be factored out when working in the frequency domain. Defining the Discrete Fourier Transform (DFT) of a generic two-dimensional discrete function of size  $M_1 \times M_2$  as in [11], the observation model in (2) can be expressed as:

$$X[\mathbf{k}, n] = B[\mathbf{k}] + S[\mathbf{k}]e^{-j2\pi\mathbf{u}_k^T\mathbf{v}n} + V_b[\mathbf{k}, n] + W[\mathbf{k}, n] \quad (3)$$

where  $\mathbf{k} = (k_1, k_2)^T$  is the vector collecting the two discrete indices of the two-dimensional DFT, with  $0 \leq k_i \leq M_i - 1$ ,  $i = 1, 2$  and  $\mathbf{u}_k = (k_1/M_1, k_2/M_2)^T$  is the vector of the normalized spatial frequencies.

Equations (2) and (3) describe Gaussian observations that are independent both in discrete time and frequency domains. Focusing on the model in (3), the speed vector  $\mathbf{v}$  represents the only unknown parameter to be estimated.

The ML criterion can be applied to the model in (3) to obtain an expression for the speed vector estimator  $\hat{\mathbf{v}}$ . Before detailing the estimation procedures, we briefly discuss the role of background removal techniques that allow to detect the foreground object  $s[\mathbf{m} - \mathbf{v}n]$  and that may be useful to further simplify the models in (2) and (3).

### C. Background Removal

Background removal techniques aim to separate the background from the foreground moving objects. The basic requirement is that the background is static. Nevertheless, this condition is often violated especially in outdoor scenes that are subject to illumination changes and other variations, e.g., due to wind effects or shadows. For this reason, adaptive algorithms for background removal, such as the Gaussian Mixture Model (GMM) [12], are usually preferred. The concept of background removal relies on the analysis and the comparison of two frames in the considered video stream: the background frame, also referred to as the reference frame where no moving objects are present, and the frame at a chosen instant where moving objects are present. The difference between these two frames represents the foreground. A comprehensive review on background removal approaches is proposed in [13].

In this paper, we exploit the reliable GMM that adaptively models each pixel belonging to the background as a mixture of Gaussian distributions. The interested reader is referred to [12] for an accurate description of the proposed method.

Using the GMM technique, it is possible to remove the background-related terms  $b[\mathbf{m}]$  and  $v_b[\mathbf{m}, n]$  to concentrate on the foreground and simplify the observation model in (2) as:

$$x[\mathbf{m}, n] = s[\mathbf{m} - \mathbf{v}n] + w[\mathbf{m}, n]. \quad (4)$$

Likewise the frequency domain observation model in (3) simplifies as:

$$X[\mathbf{k}, n] = S[\mathbf{k}]e^{-j2\pi\mathbf{u}_k^T\mathbf{v}n} + W[\mathbf{k}, n]. \quad (5)$$

### III. MAXIMUM LIKELIHOOD SPEED ESTIMATION

#### A. Speed Estimation with Included Background

The observation model (2) and its frequency-domain counterpart (3) include the terms  $v_b[\mathbf{m}, n]$  and  $V_b[\mathbf{k}, n]$ , respectively, to account for the background occlusion by the moving foreground object. This masking operation requires the extraction of the foreground and is equivalent to background removal used in the next subsection. As a heuristic approach, the term  $V_b[\mathbf{k}, n]$  in (3) can be simply neglected [14]. The resulting observation model is obtained by setting  $V_b[\mathbf{k}, n] = 0$  in (3).

Following standard methods described in [9], the ML criterion can be applied to obtain an expression for the log-likelihood function to be maximized. A likelihood function of the observed data in (3), with  $V_b[\mathbf{k}, n] = 0$ , can be defined on the basis of a window of  $N$  observed frames as:

$$\begin{aligned} p(X[\mathbf{k}, 0] \cdots X[\mathbf{k}, N-1]; \mathbf{v}) &= \left( \frac{1}{2\pi\sigma^2} \right)^{\frac{M_1 M_2 N}{2}} \\ &\cdot \exp \left\{ -\frac{1}{2\sigma^2} \sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} \sum_{n=0}^{N-1} \left| X[\mathbf{k}, n] \right. \right. \\ &\quad \left. \left. - S[\mathbf{k}]e^{-j2\pi\mathbf{u}_k^T\mathbf{v}n} - B[\mathbf{k}] \right|^2 \right\} \end{aligned} \quad (6)$$

where  $\sigma$  is the standard deviation of the additive Gaussian noise elements. A log-likelihood function can be derived from (6) as:

$$\begin{aligned} \ln(p(X[\mathbf{k}, 0] \cdots X[\mathbf{k}, N-1]; \mathbf{v})) &= -\underbrace{\frac{M_1 M_2 N}{2} \ln(2\pi\sigma^2)}_{(b)} \\ &- \underbrace{\frac{1}{2\sigma^2} \sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} \sum_{n=0}^{N-1} \left| X[\mathbf{k}, n] - S[\mathbf{k}]e^{-j2\pi\mathbf{u}_k^T\mathbf{v}n} - B[\mathbf{k}] \right|^2}_{(a)}. \end{aligned} \quad (7)$$

The operating principle of the ML approach is to find the value of  $\mathbf{v}$  that maximizes the log-likelihood function in (7). Equivalently, the term (a) can be minimized by observing that the term (b) and the multiplicative coefficient  $-\frac{1}{2\sigma^2}$  are

irrelevant and constant, being independent from  $\mathbf{v}$ . The term (a) in (7) can be written explicitly as:

$$\begin{aligned} &\sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} \sum_{n=0}^{N-1} \left\{ |X[\mathbf{k}, n]|^2 + |S[\mathbf{k}]|^2 + |B[\mathbf{k}]|^2 \right. \\ &\quad \left. - 2 \operatorname{Re} \left\{ X[\mathbf{k}, n] S^*[\mathbf{k}] e^{j2\pi\mathbf{u}_k^T\mathbf{v}n} \right\} \right. \\ &\quad \left. - 2 \operatorname{Re} \left\{ X[\mathbf{k}, n] B^*[\mathbf{k}] \right\} \right. \\ &\quad \left. + 2 \operatorname{Re} \left\{ S^*[\mathbf{k}] e^{j2\pi\mathbf{u}_k^T\mathbf{v}n} B[\mathbf{k}] \right\} \right\} \end{aligned} \quad (8)$$

where  $\operatorname{Re}\{\cdot\}$  indicates the real part. The terms  $|X[\mathbf{k}, n]|^2$ ,  $|S[\mathbf{k}]|^2$  and  $|B[\mathbf{k}]|^2$  can be neglected because irrelevant, since independent of  $\mathbf{v}$ . Hence, minimizing (8) corresponds to maximizing the following quantity, where three terms are highlighted:

$$\begin{aligned} &\sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} \sum_{n=0}^{N-1} \left\{ \underbrace{\operatorname{Re} \left\{ X[\mathbf{k}, n] S^*[\mathbf{k}] e^{j2\pi\mathbf{u}_k^T\mathbf{v}n} \right\}}_{(T1)} \right. \\ &\quad \left. + \underbrace{\operatorname{Re} \left\{ X[\mathbf{k}, n] B^*[\mathbf{k}] \right\}}_{(T2)} - \underbrace{\operatorname{Re} \left\{ S^*[\mathbf{k}] e^{j2\pi\mathbf{u}_k^T\mathbf{v}n} B[\mathbf{k}] \right\}}_{(T3)} \right\}. \end{aligned} \quad (9)$$

Using the linearity of  $\operatorname{Re}\{\cdot\}$  and sum operators,  $S^*[\mathbf{k}]e^{j2\pi\mathbf{u}_k^T\mathbf{v}n}$  can be factored out from (T1) and (T3) to obtain:

$$\begin{aligned} &\sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} \left\{ \operatorname{Re} \left\{ \sum_{n=0}^{N-1} (X[\mathbf{k}, n] - B[\mathbf{k}]) S^*[\mathbf{k}] e^{j2\pi\mathbf{u}_k^T\mathbf{v}n} \right. \right. \\ &\quad \left. \left. + X[\mathbf{k}, n] B^*[\mathbf{k}] \right\} \right\}. \end{aligned} \quad (10)$$

The continuous-frequency FT of the temporal sequence  $\{(X[\mathbf{k}, n] - B[\mathbf{k}]) S^*[\mathbf{k}]\}$  can be now defined as:

$$Y[\mathbf{k}, f] = \sum_{n=0}^{N-1} (X[\mathbf{k}, n] - B[\mathbf{k}]) S^*[\mathbf{k}] e^{-j2\pi f T_s n}. \quad (11)$$

Setting  $f = -\frac{\mathbf{u}_k^T\mathbf{v}}{T_s}$ , equation (10) can be expressed as:

$$\sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} \operatorname{Re} \left\{ Y \left[ \mathbf{k}, -\frac{\mathbf{u}_k^T\mathbf{v}}{T_s} \right] + N \bar{X}[\mathbf{k}] B^*[\mathbf{k}] \right\}, \quad (12)$$

where  $\bar{X}[\mathbf{k}]$  is the temporal mean of the sequence  $X[\mathbf{k}, n]$  and is defined as:

$$\bar{X}[\mathbf{k}] = \frac{1}{N} \sum_{n=0}^{N-1} X[\mathbf{k}, n]. \quad (13)$$

Finally, an expression for the estimator of the speed vector  $\hat{\mathbf{v}}$  with included background is:

$$\hat{\mathbf{v}} = \operatorname{argmax}_{\mathbf{v}} \sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} \operatorname{Re} \left\{ Y \left[ \mathbf{k}, -\frac{\mathbf{u}_k^T\mathbf{v}}{T_s} \right] + N \bar{X}[\mathbf{k}] B^*[\mathbf{k}] \right\}. \quad (14)$$

### B. Speed Estimation with Omitted Background

Following the aforementioned procedure, it is possible to derive an estimator of the speed vector in the case of omitted background according to model (5), by setting  $B[\mathbf{k}] = 0$ .

In this case, (11) becomes

$$Y[\mathbf{k}, f] = \sum_{n=0}^{N-1} X[\mathbf{k}, n] S^{*}[\mathbf{k}] e^{-j2\pi f T_s n} \quad (15)$$

and (14) simplifies as:

$$\hat{\mathbf{v}} = \underset{\mathbf{v}}{\operatorname{argmax}} \sum_{k_1=0}^{M_1-1} \sum_{k_2=0}^{M_2-1} \operatorname{Re} \left\{ Y \left[ \mathbf{k}, -\frac{\mathbf{u}_{\mathbf{k}}^T \mathbf{v}}{T_s} \right] \right\}. \quad (16)$$

## IV. NUMERICAL RESULTS

In this section, the performance of the presented ML speed estimation algorithms is discussed. In particular, the results are presented in terms of Root Mean Square Error (RMSE) between  $\hat{\mathbf{v}}$  and the correct speed of the framed object for two different categories of videos. A set of synthetic and software-generated videos is considered to preliminarily test the effectiveness of the derived methods in a controlled environment, then a number of real-world videos, specifically recorded for this purpose, is analysed. Comparisons with the well-known block-matching method [3, Ch.4] are also presented.

### A. Software-Generated Videos

As a first experiment, we apply the proposed speed estimation methods to synthetic videos obtained by inserting real pictures of moving objects in an artificial environment (i.e. a static background) and by adding white Gaussian noise to simulate the behaviour of a real image acquisition system. A sample frame of a grayscale video considered for this purpose is shown in Figure 1(a), where the moving object is the (highlighted) bird, placed in the upper part.<sup>2</sup> The main advantage of synthetic videos relies on the possibility to manually or automatically set a number of parameters, including the speed components, thus allowing a direct assessment of the estimation performance.

In particular, we consider a set of 10 videos in order to test as many values of speed components, measured in pixel/frame. The duration of each video is about 6 s with a frame rate  $f_s = 15$  Hz and a frame size of  $361 \times 616$  pixels. The total number of frames is 85, of which the first 40 are background frames, exploited by the GMM algorithm, and  $N = 45$  in (11) and (15). The initial variance of the Gaussian distributions in the mixture (5 in this case) is set to 0.0025 [12], as a good compromise to account for the variations due to the additive white Gaussian noise (AWGN) and the motion of the foreground object.

Using (14) and (16) the speed components are estimated in the cases of included and omitted background. Results are shown in Figure 1(b), along with a comparison with

<sup>2</sup>The bird has been highlighted to ease visualization in Figure 1(a) only, not in the processed video.

the well-known block-matching approach, for which a block size of  $35 \times 35$  pixels is chosen by trial and error to take into account the foreground size. For each video, the RMSE normalized to the correct values of the speed components for 10 different noise realizations is measured. The obtained normalized RMSE values are then averaged with respect to the various videos and the result is plotted against increasing values of the noise variance  $\sigma^2$  for all the assessed algorithms.

Figure 1(b) shows the robustness of the proposed methods against noise for both the considered cases. The average RMSE is zero for low values of noise variance  $\sigma^2$  because the correct and estimated speed components are quantized to integer values. In particular, the case of omitted background gives lower errors up to a noise variance of about 0.15, while the case of included background is more efficient for higher values of noise variance. On the other hand, the high noise sensitivity of the block-matching criterion is confirmed by observing the rapidly increasing trend of its RMSE curve.

### B. Real-World Videos

As realistic examples, real videos of moving cars are considered. In particular we apply the proposed algorithms to a set of 6 videos with various duration that ranges between about 4 and 10 s and with various numbers of background frames. The videos are recorded with a frame rate  $f_s = 30$  Hz and have a frame size of  $360 \times 640$  pixels. The number of frames  $N$  in (11) and (15) ranges between 7 and 60. Furthermore, the initial variance of the 5 Gaussian distributions in the mixture model is set to 0.81 to track scene modifications due to lighting changes and wind effects in addition to the the AWGN and motion of the foreground object, as in the previous scenario. The car reference speeds are manually measured and can be considered approximately constant over the entire video duration. A sample frame of a video in this category is shown in Figure 2(a).

Following the previous approach, our speed estimation methods are applied with included and omitted background, and compared with the performance of the block-matching algorithm, for which the block size is again set to  $35 \times 35$  pixels.

As in the previous example, for each video the RMSE normalized to the correct speed components is measured for 10 different noise realizations and the obtained values are again averaged with respect to the various videos. In Figure 2(b) the results are plotted against increasing values of the noise variance  $\sigma^2$  for the three algorithms. For very low noise, e.g. for  $\sigma^2 = 0$ , the performance of the block-matching approach provides a low error, nevertheless its high noise sensitivity is again confirmed by the rapidly increasing trend of its RMSE curve. This demonstrates the robustness of the proposed ML estimation techniques, which exhibit lower values of RMSE even at much higher values of noise variance.

In this example, the average RMSE with included background settles about 0.38, but for small values of noise variance, because speed estimation consistently fails in some of the considered videos. However, the performance of the

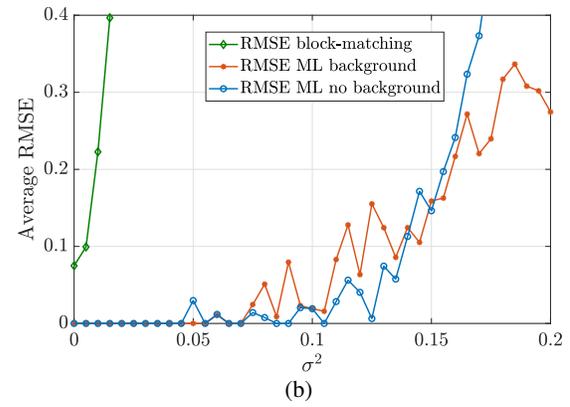
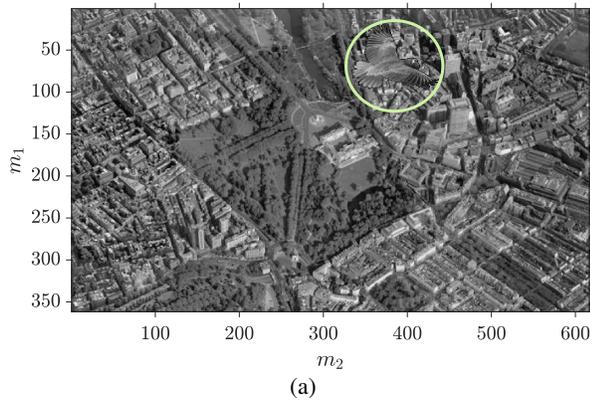


Fig. 1: Examples of applications and results for synthetic software-generated videos: (a) sample frame, (b) average RMSE.

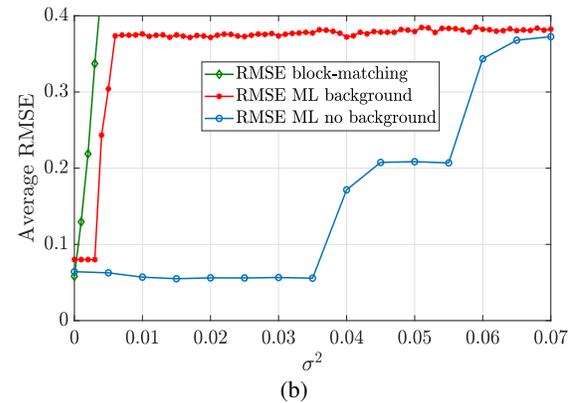
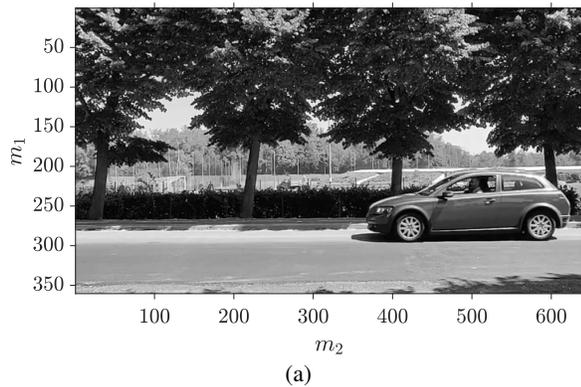


Fig. 2: Examples of applications and results for real videos: (a) sample frame, (b) average RMSE.

algorithm with omitted background is significantly better. We also note that the RMSE values for low noise variance are not exactly zero, unlike the previous case. This is due to the used quantization of the speed component estimates to integer values, unlike the non integer values of the correct speed components. Furthermore, the nature of real videos, including artifacts such as illumination changes and wind effects, may affect the operation of background removal and foreground detection performed by means of the GMM algorithm.

## V. CONCLUSION

In this paper, we presented a novel approach for speed estimation in video signals. We derived an observation model for included and omitted background and we applied the ML criterion to estimate the speed components of a foreground object shifting with a nearly constant speed. By considering synthetic and real videos as examples of application, we demonstrated the effectiveness of the proposed algorithms by comparison with the well-known block-matching algorithm.

## REFERENCES

- [1] B. Pesquet-Popescu, M. Cagnazzo, and F. Dufaux, "Motion Estimation Techniques," Tech. Rep., 2016, [Online] Available: [https://mmnet.wp.imt.fr/files/2016/12/poly\\_me.pdf](https://mmnet.wp.imt.fr/files/2016/12/poly_me.pdf).
- [2] Y.-K. Chen, "True Motion Estimation-Theory, Application, and Implementation," Ph.D. dissertation, Princeton University, 1998.
- [3] A. M. Tekalp, *Digital Video Processing*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2015.
- [4] D. J. Fleet and Y. Weiss, *Handbook of Mathematical Models in Computer Vision*. Boston, MA, USA: Springer, 2006, ch. Optical Flow Estimation, pp. 239–257.
- [5] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artif. Intell.*, San Francisco, CA, USA, Aug. 1981, pp. 674–679.
- [6] B. K. P. Horn and B.G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, no. 1, pp. 185–203, Aug. 1981.
- [7] X. Yu, P. Xue, L. Duan and Q. Tian, "An algorithm to estimate mean vehicle speed from MPEG skycam video," *Multimedia Tools Appl.*, vol. 34, no. 1, p. 85–105, Jul. 2007.
- [8] F. Y. Hu, H. Sahli, X. F. Dong and J. Wang, "A high efficient system for traffic mean speed estimation from MPEG video," in *Proc. IEEE Int. Conf. Artif. Intell. and Comput. Intell.*, vol. 3, Shanghai, China, Nov. 2009, pp. 444–448.
- [9] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall, 1993.
- [10] A. Briassouli and N. Ahuja, "Integration of frequency and space for multiple motion estimation and shape-independent object segmentation," *IEEE Trans. Circuits and Syst. for Video Technol.*, vol. 18, pp. 657 – 669, May 2008.
- [11] C. Solomon and T. Breckon, *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*, 2nd ed. Upper Saddle River, NJ, USA: Wiley-Blackwell, 2011.
- [12] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Int. Conf. Comput. Vision and Pattern Recognit. (CVPR)*, vol. 2, Cambridge, MA, USA, June 1999, pp. 246–252.
- [13] M. Piccardi, "Background subtraction techniques: a review," in *IEEE Int. Conf. Syst., Man and Cybern.*, vol. 4, The Hague, NL, Oct. 2004, pp. 3099–3104.
- [14] A. Briassouli and N. Ahuja, "Extraction and analysis of multiple periodic motions in video sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 7, pp. 1244–1261, July 2007.