

Accurate and Efficient Background Subtraction by Monotonic Second-Degree Polynomial Fitting

Alessandro Lanza Federico Tombari Luigi Di Stefano
DEIS, University of Bologna
Viale Risorgimento 2, 40136 Bologna, Italy

Abstract

We present a background subtraction approach aimed at efficiency and accuracy also in presence of common sources of disturbance such as illumination changes, camera gain and exposure variations, noise. The novelty of the proposal relies on a-priori modeling the local effect of disturbs on small neighborhoods of pixel intensities as a monotonic, homogeneous, second-degree polynomial transformation plus additive Gaussian noise. This allows for classifying pixels as changed or unchanged by an efficient inequality-constrained least-squares fitting procedure. Experiments prove that the approach is state-of-the-art in terms of efficiency-accuracy tradeoff on challenging sequences characterized by disturbs yielding sudden and strong variations of the background appearance.

1. Introduction and previous work

Many video analysis applications, such as e.g. intelligent video surveillance and automated traffic monitoring, rely on background subtraction as a critical pre-processing step aimed at detecting where the attention of higher level processing modules should be focused on. The main difficulty with background subtraction consists in detecting such areas robustly with respect to sources of disturbance that may modify the appearance of the reference scene, such as noise, gradual or sudden illumination changes (e.g. due to time of the day or a light switch), dynamic adjustments of camera parameters (e.g. auto-exposure, auto-gain). To carry out background subtraction robustly is a challenging research issue, as vouched by the richness of proposals found in literature (see [5] for a recent survey). The other constraint that should be taken into account when designing a background subtraction algorithm concerns efficiency. In fact, the lower is the algorithm complexity, the greater are the computational resources available for higher level reasoning tasks, such as temporal tracking and classification of moving targets.

Background subtraction algorithms can be roughly divided into two main classes. A first class relies on learning off-line, i.e. at initialization time, a per-pixel model of the background appearance and on continuously updating it to incorporate the possible effects of disturbs. Then, a pixel from a new frame is marked as foreground if its intensity “does not fit” the current background appearance model. Several statistical models, ranging from a single Gaussian [13] to mixtures of Gaussians [12] and kernel-based non-parametric models [4] have been proposed. Yet, the need for on-line updating the model inherently implies a tradeoff associated with the chosen adaptation rate: prompt model updating can result in foreground elements being mistakenly integrated into the background model whilst slow updating makes this first class of algorithms inadequate to cope with fast background appearance variations, such as those due to sudden illumination changes and camera gain and exposure adjustments.

A second class of algorithms relies on estimating off-line the background model and a-priori modeling the background appearance changes yieldable by the sources of disturbance within small image patches. Accordingly, a pixel from a new frame is classified as foreground if the intensity variations with respect to the background observed within a surrounding neighborhood “does not fit” the a-priori model of changes. Since changes due to disturbs are a-priori modeled, these algorithms can handle seamlessly slow and sudden photometric variations and the background model does not need to be updated. A critical issue with these algorithms concerns the a-priori model of changes, which is generally linear [3, 11] or order-preserving [14, 9, 6]. In principle, the more restrictive such a model the higher is the ability to detect foreground elements (sensitivity) but the lower is robustness to sources of disturbance (specificity). As discussed in [14], many non-linearities may arise in the image formation process, so that a more liberal model than the linear is often required to achieve adequate robustness in practical applications.

In this paper we propose a novel background subtraction algorithm belonging to the second class which assumes a

monotonic, homogeneous, second-degree polynomial transformation as a-priori model for the local intensity variations due to disturbs. That is, we assume a model that is more general than the linear one and less general than the order-preserving one. In this manner, our approach holds the potential for achieving a better tradeoff between robustness and sensitivity. In more practical terms, this translates into the potential to achieve good accuracy over a wider spectrum of scenarios characterized by different types of intensity changes due to nuisance factors. Moreover, the selected parametric model allows for a particularly efficient fitting procedure, so that, in particular, the complexity turns out to be $O(1)$ with respect to the neighborhood size.

2. The proposed algorithm

Let us consider a generic pixel in the two images under comparison, i.e. background and current frame, and let us denote, respectively, as

$$\mathbf{x}^t = [x_1, \dots, x_N] \quad \text{and} \quad \mathbf{y}^t = [y_1, \dots, y_N] \quad (1)$$

the N pixel intensities measured in a surrounding neighborhood. We aim at detecting scene changes occurring in the central pixel by evaluating the local intensity information contained in \mathbf{x} and \mathbf{y} .

We assume that the background model is computed by means of a statistical estimation over an initialization sequence (e.g. temporal averaging of tens of frames) so that noise affecting the inferred background intensities can be neglected. Hence, \mathbf{x} can be thought of as a vector of noiseless intensities that are constant over time, with noise affecting the elements of \mathbf{y} only. In particular, we assume that noise is additive, zero-mean, i.i.d. Gaussian. In addition to noise, we assume that the main photometric distortions are due to illumination changes in the scene and variations of camera parameters (such as exposure and gain).

The main idea behind the novel change detection algorithm proposed in this paper is that the effects of photometric distortions occurring in the scene can be well modeled locally, i.e. within a small neighborhood of pixels, by means of a monotonic non-decreasing, homogeneous, second-degree polynomial transformation of background intensities. In formulas:

$$y_i = \phi(x_i; \boldsymbol{\theta}) + \xi_i = \theta_1 x_i^2 + \theta_2 x_i + \xi_i \quad \forall i = 1, \dots, N \quad (2)$$

with the constraint:

$$\frac{d\phi}{dx}(x; \boldsymbol{\theta}) = 2\theta_1 x + \theta_2 \geq 0 \quad \forall x \in [0, G] \quad (3)$$

where ξ_i denotes the additive Gaussian noise affecting the current frame intensity of the i -th pixel in the neighborhood and G is the highest measurable intensity (for 8-bit images,

$G = 255$). Due to linearity of the first derivative of $\phi(\cdot)$, the monotonicity constraint (3) over the entire x -domain is equivalent to imposing monotonicity at the domain extremes:

$$\frac{d\phi}{dx}(0; \boldsymbol{\theta}) \geq 0 \quad \wedge \quad \frac{d\phi}{dx}(G; \boldsymbol{\theta}) \geq 0 \quad (4)$$

which, in turn, can be expressed as:

$$\boldsymbol{\theta} \in \Theta = \{ \boldsymbol{\theta} \in \mathbb{R}^2 : 2G\theta_1 + \theta_2 \geq 0 \quad \wedge \quad \theta_2 \geq 0 \} \quad (5)$$

It is important to note that the proposed model (2), (5) for photometric distortions is a super-set of the linear model assumed in [3, 11], for which any non-linear mapping function between the background and current frame intensities is considered as the effect of scene changes. At the same time, our model is a sub-set of the order-preserving one deployed in [14, 9, 6], for which any monotonic non-decreasing intensity transformation is considered as an allowable effect of nuisance factors.

Given the model (2), (5) for photometric distortions yieldable by disturbance factors over a neighborhood of pixel intensities, scene changes are detected by computing a measure of the distance between the sensed neighborhood intensities \mathbf{x} and \mathbf{y} and the model. In other words, if the sensed intensities are not well-fitted by the model, the central pixel is marked as changed.

The minimum-distance intensity transformation among those belonging to the space defined by the model is computed by a maximum-likelihood estimation of the parameters vector:

$$\boldsymbol{\theta}_{ML} = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmax}} p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}) \quad (6)$$

By making the additive, zero-mean, i.i.d. Gaussian noise model explicit and transforming likelihood maximization into minus log-likelihood minimization, we obtain:

$$\boldsymbol{\theta}_{ML} = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmin}} E(\boldsymbol{\theta}) \quad (7)$$

where the cost function

$$E(\boldsymbol{\theta}) = \sum_{i=1}^N (y_i - \phi(x_i))^2 = \sum_{i=1}^N (y_i - \theta_1 x_i^2 - \theta_2 x_i)^2 \quad (8)$$

represents the least-squares regression error, i.e. the sum over all the pixels in the neighborhood of the square difference between the frame intensity and the background intensity transformed by the model. To the purpose of classifying the pixel as change or unchanged, actually we are not interested in estimating the intensity transformation that best explains the data, i.e. $\boldsymbol{\theta}_{ML}$, but indeed the distance of such transformation from the data. Hence, we write more

compactly the estimation problem (7), (8) as the following quadratic programming problem:

$$E^{(min)} = E(\boldsymbol{\theta}_{ML}) = \min_{A\boldsymbol{\theta} \geq 0} \frac{1}{2} \boldsymbol{\theta}^t H \boldsymbol{\theta} - \mathbf{b}^t \boldsymbol{\theta} \quad (9)$$

with the Hessian matrix H and the coefficient vector \mathbf{b} of the linear part of the quadratic cost function given by:

$$H = \begin{bmatrix} Sx^4 & Sx^3 \\ Sx^3 & Sx^2 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} Sx^2y \\ Sxy \end{bmatrix} \quad (10)$$

and, for simplicity of notation:

$$\begin{aligned} Sx^2 &= \sum_{i=1}^N x_i^2 & Sx^3 &= \sum_{i=1}^N x_i^3 & Sx^4 &= \sum_{i=1}^N x_i^4 \\ Sx^2y &= \sum_{i=1}^N x_i^2 y_i & Sxy &= \sum_{i=1}^N x_i y_i & Sy^2 &= \sum_{i=1}^N y_i^2 \end{aligned} \quad (11)$$

As for the linear constraints matrix A defining the feasible set Θ of the quadratic programming problem:

$$A = \begin{bmatrix} 2G & 1 \\ 0 & 1 \end{bmatrix} \quad (12)$$

Due to the positive-definiteness of the Hessian matrix H , the quadratic cost function $E(\boldsymbol{\theta})$ is strictly convex. Hence, the cost function admits a unique point $\boldsymbol{\theta}^{(u)}$ of unconstrained global minimum that can be easily found by searching for the unique zero-gradient point, i.e. by solving the linear system of normal equations:

$$\boldsymbol{\theta}^{(u)} = \boldsymbol{\theta} \in \mathbb{R}^2 : H \boldsymbol{\theta} = \mathbf{b} \quad (13)$$

The above system can be solved in closed form by Cramer's rule:

$$\theta_1^{(u)} = \frac{|H_1|}{|H|} \quad \theta_2^{(u)} = \frac{|H_2|}{|H|} \quad (14)$$

where $|H|$ denotes the determinant of the system coefficient matrix and $|H_1|$, $|H_2|$ the determinants of the matrices obtained by replacing, respectively, the first and second column of the matrix H with the right-hand side vector \mathbf{b} . The unconstrained global minimum of the cost function, $E^{(u)}$, can thus be obtained by replacing $\boldsymbol{\theta}^{(u)}$ computed by (14) in the expression of the cost function (8). After some algebra:

$$\begin{aligned} E^{(u)} &= E(\boldsymbol{\theta}^{(u)}) = Sy^2 + \\ &- \frac{Sx^2(Sx^2y)^2 + Sx^4(Sxy)^2 - 2Sx^3SxySx^2y}{Sx^2Sx^4 - (Sx^3)^2} \end{aligned} \quad (15)$$

It can be easily demonstrated that the point of unconstrained global minimum $\boldsymbol{\theta}^{(u)}$ belongs to the problem feasible set Θ , i.e. satisfies the constraint $A\boldsymbol{\theta} \geq 0$, if and only if:

$$\frac{2G Sx^3 - Sx^4}{2G Sx^2 - Sx^3} \leq \frac{Sx^2y}{Sxy} \leq \frac{Sx^4}{Sx^3} \quad (16)$$

Hence, if condition (16) is satisfied, the solution of the quadratic programming problem (9) is:

$$E^{(min)} = E^{(u)} \quad (17)$$

On the contrary, if the point $\boldsymbol{\theta}^{(u)}$ falls outside the feasible set Θ , the solution of the problem must lie on the boundary of the feasible set $\partial\Theta$, that is on one of the two half-lines:

$$\begin{aligned} \partial\Theta_1 &: 2G\theta_1 + \theta_2 = 0 \wedge \theta_1 \leq 0 \\ \partial\Theta_2 &: \theta_2 = 0 \wedge \theta_1 \geq 0 \end{aligned} \quad (18)$$

The minimum of the cost function on each of the two half-lines can be determined by replacing the line equation into the expression of the cost function and then searching for the minimum of the obtained one-variable quadratic function. After simple analysis, we obtain:

$$\begin{aligned} E^{(1)} &= Sy^2 - \frac{(Sx^2y)^2 + G^2(Sxy)^2 - 2G SxySx^2y}{Sx^4 + G^2Sx^2 - 2G Sx^3} \\ E^{(2)} &= Sy^2 - \frac{(Sx^2y)^2}{Sx^4} \end{aligned} \quad (19)$$

Once the two minima have been calculated, the solution of the problem is clearly:

$$E^{(min)} = \min \{ E^{(1)}, E^{(2)} \} \quad (20)$$

Finally, the considered pixel is classified as changed if the computed cost $E^{(min)}$, i.e. the distance between the measured intensities and the model assumed for photometric distortions due to nuisance factors, is higher than a fixed threshold.

By means of the proposed solution, we have no need to resort to any iterative approach. This renders our method particularly efficient. In addition, it is worth pointing out that all terms involved in the calculations, i.e. those in the Hessian matrix H and in the vector \mathbf{b} , can be computed either off-line (i.e. those involving only background intensities) or by means of very fast incremental techniques such as Summed Area Table [2] (those involving also frame intensities). Overall, this allows our method to exhibit a computational complexity of $O(1)$ with respect to the neighborhood size N .

3. Experimental results

In this section we propose an experimental comparison between our approach and other state-of-the-art techniques. In particular, we have selected four techniques as terms of comparison, all of them belonging to the class of methods that models a-priori the effects of disturbance factors on a local neighborhood of intensities. Between the selected techniques, three are invariant to order-preserving intensity transformations, i.e. Local Binary Pattern by Heikkila and

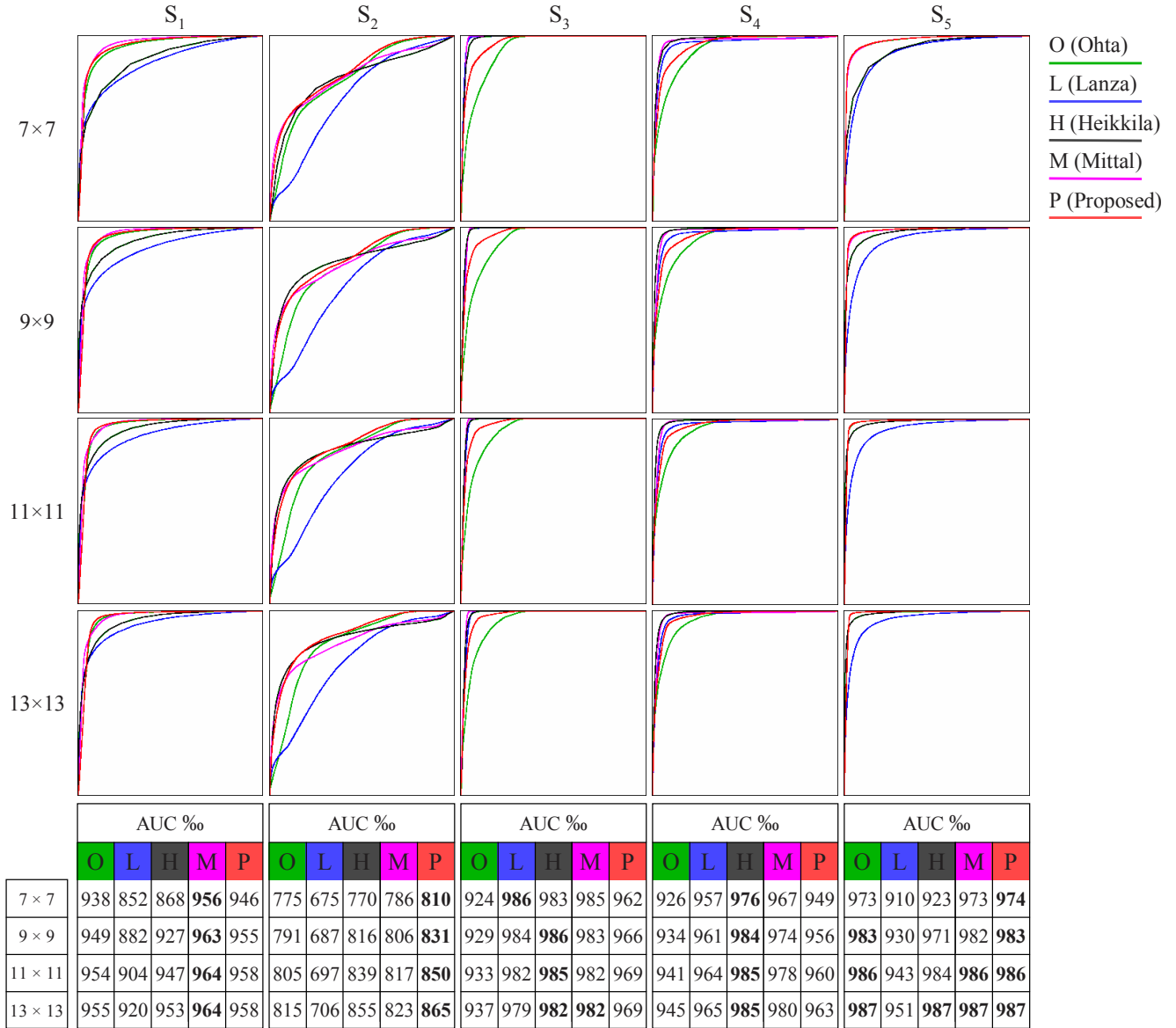


Figure 1. ROC curves (top) and AUC values (multiplied by 10^3 , bottom) obtained by the algorithms on the five test sequences with four different neighborhood sizes.

Pietikainen [6], the visual correspondence measure by Mittal and Ramesh [9] and the algorithm proposed by Lanza and Di Stefano [7]. Hereinafter, these techniques will be referred to, respectively, as *Heikkila*, *Mittal* and *Lanza*. As for the fourth technique, we have considered a well-known representative of methods invariant to linear intensity transformations, i.e. the algorithm described by Ohta in [11]. All algorithms were implemented in C using the same optimizations (e.g. using incremental techniques [2] wherever possible) in order to allow also for a comparison in terms of efficiency among the selected techniques. As for the implementation of *Heikkila*, since the other methods only model the effect of disturbance factors spatially, for a fair compar-

ison we do not apply the temporal updating procedure.

We have compared the techniques on five test sequences S_1, \dots, S_5 characterized by sudden and notable photometric changes yielding both linear and non-linear intensity transformations between the frames and the inferred background. For each sequence, the background and one sample frame are shown, respectively, in the first and second row of Figure 2. In particular, S_1, S_2 are real indoor sequences we acquired by means of, respectively, a high-quality camera and a webcam. S_1 shows a person moving around a room, and sudden illumination changes are caused by varying the rheostat of the artificial light illuminating the environment. Instead, S_2 concerns an object on a cluttered

	7x7		9x9		11x11		13x13		AVG.		RANK		MEAN
	AUC	FPS	AUC	FPS	AUC	FPS	AUC	FPS	AUC	FPS	AUC	FPS	RANK
O	907.2	103.2	917.2	105.1	923.8	103.5	927.8	103.4	919.0	103.8	4	1	2.5
L	876	27.6	888.8	17.1	898	12.3	904.2	8	891.8	16.3	5	3	4.0
H	904	3.2	936.8	3.1	948	2.9	952.4	2.5	935.3	2.9	3	4	3.5
M	933.4	0.4	941.6	0.1	946.6	0.06	947.2	0.03	942.2	0.1	1	5	3.0
P	928.2	37.8	938.2	37.6	944.6	37.8	948.4	37.5	939.9	37.7	2	2	2.0

Table 1. Average values and ranks of AUC and FPS over the five test sequences and for the four neighborhood sizes.

background, with photometric distortions induced by turning on and off an artificial light and varying the camera gain and exposure by both manual and automatic adjustments. S_3, S_4 are two real outdoor sequences acquired with a standard surveillance camera and dealing with people and vehicles moving over a parking area. In addition to illumination changes, photometric variations are induced by manually adjusting the camera gain and exposure. Finally, S_5 is a synthetic benchmark sequence available on the web [10] in which moving people as well as photometric changes have been artificially superimposed on an outdoor background. Some compression artifacts are also present, particularly in correspondence of the textured parts of the scene.

We first show quantitative results aimed at evaluating the algorithms in terms of both accuracy and computational efficiency. As for the former, we have obtained the ground truth change masks for sequences S_1, S_2, S_3, S_4 by manual labeling of a subset of frames, while the ground truth for S_5 is available from [10]. This allows for comparing the algorithms by means of true positive rate (TPR) versus false positive rate (FPR) receiver operating characteristic (ROC) curves. It is also worth pointing out that it is possible to obtain indications concerning the type of spurious changes occurring in each sequence by computing the *comparagrams* [8], i.e. the 2-dimensional joint histograms of background versus frame intensities. To better represent the effects of disturbance factors in the considered frames, comparagrams have been computed by taking into account only background pixels, as labeled in the corresponding ground truth masks. The comparagram images, shown in the fourth row of Figure 2 with brighter pixels corresponding to higher frequencies, indicate with strong evidence linear underlying transformations of pixel intensities in S_1, S_5 while in S_2, S_3, S_4 they suggest with more evidence non-linear transformations. The background model of each sequence has been built by averaging an initial sequence of frames free of moving objects. We report the performance of the algorithms for four different neighborhood sizes ($7 \times 7, 9 \times 9, 11 \times 11, 13 \times 13$). We start from 7×7 since this is the minimum allowable size for algorithm *Heikkila*.

Figure 1 (top) shows the ROC curves obtained for each considered algorithm, sequence and neighborhood size. From each curve we have also computed a well-known

scalar measure of performance, the area under the curve (AUC), which represents the probability for the algorithm to assign a randomly chosen changed pixel a higher change score than a randomly chosen unchanged pixel [1]. AUC values are reported in the Table included in Figure 1 (bottom), with best performance denoted in boldface.

In addition, Table 1 summarizes the results of Figure 1 by showing AUC values averaged over the 5 sequences for the different neighborhood sizes. The Table reports also the average frame-per-second (FPS) measured for each algorithm at the different neighborhood sizes. The average AUC and FPS values over all sequences and neighborhood sizes are then shown in column 9 and 10 of Table 1. According to the methodology employed by a well-known benchmarking website for computer vision algorithms¹, we also show, in the third-last and second-last columns of the Table, the rankings of the algorithms in terms of, respectively, AUC and FPS. Finally, the rightmost column reports the mean ranking obtained by averaging together the AUC rank and the FPS rank.

The ROC curves and the AUC values of Figure 1 shows that the proposed algorithm is the most accurate on sequence S_2 while *Mittal* and *Heikkila* are the most accurate on S_1 and S_4 respectively. On sequences S_3 and S_5 more than one algorithm achieves substantially the best performance. Overall the proposed algorithm, *Mittal* and *Heikkila* seem significantly more accurate than the other two, as vouched also by the average AUC values reported in Table 1. Another aspect that can be noted from the AUC values shown in the Table is that *Heikkila* does not perform well with small neighborhood sizes. It is worth pointing out that, as shown by Table 1, the proposed algorithm yields the second best average AUC over all the different sequences and neighborhood sizes, outperformed only - though slightly - by *Mittal*. For what concern efficiency, though, *Mittal* pays off the high accuracy with a high computational cost, resulting in the slowest FPS between the considered methods. In particular, as shown by the Table, *Mittal* runs two orders of magnitude slower than the proposed approach, with frame-rates that are far from supporting video-rate processing. As for *Heikkila*, it turns out

¹<http://vision.middlebury.edu/stereo>

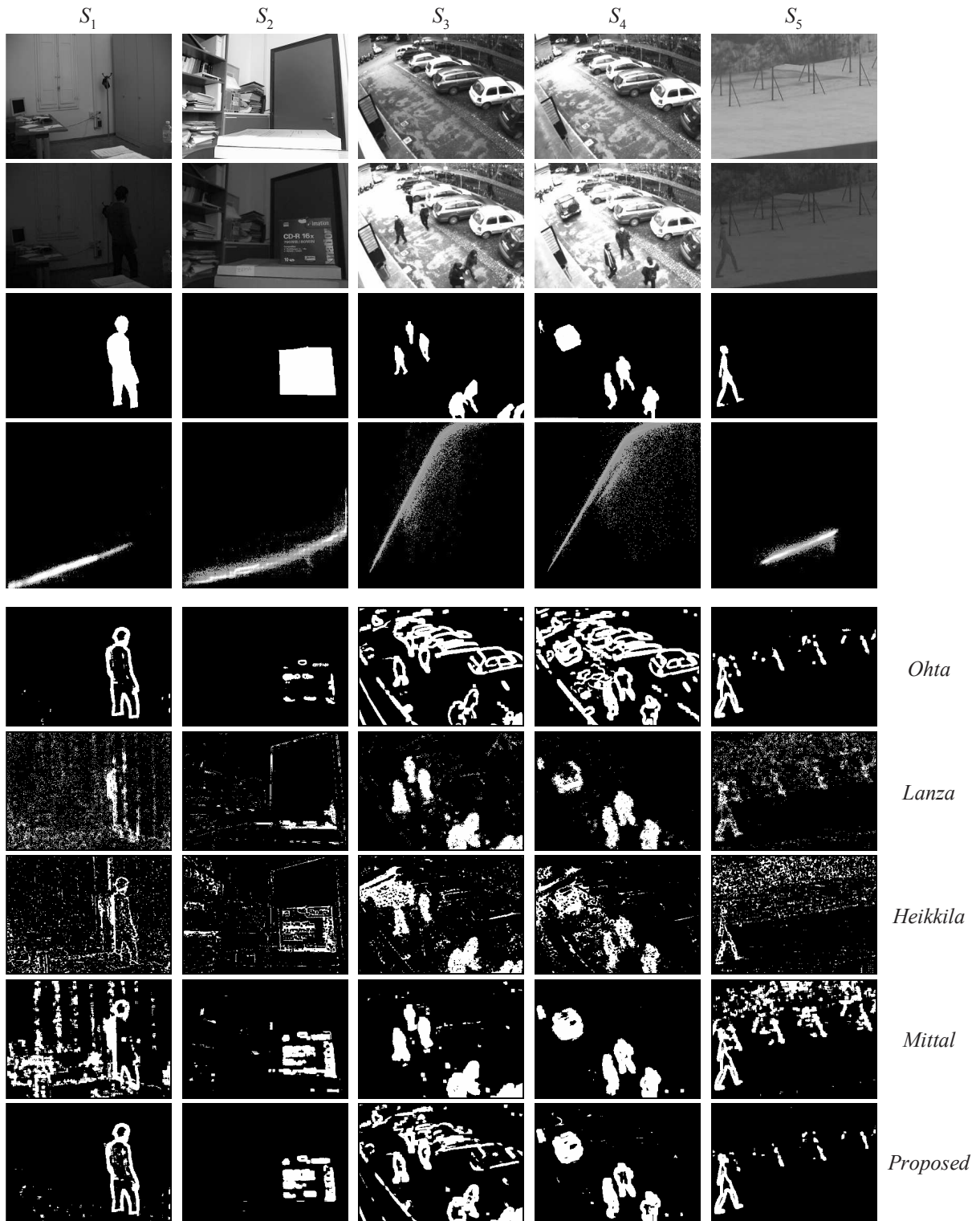


Figure 2. Qualitative results yielded by the evaluated algorithms on the five test sequences. For each sequence, from top to bottom: the inferred background, one sample frame, the corresponding ground truth mask, the associated comparagram and the binary change masks yielded by the algorithms.

on average slightly less accurate but significantly less efficient (i.e. one order of magnitude slower) than the proposed method. Overall, only the proposed algorithm, *Ohta* and *Lanza* can support video-rate processing, the proposed algorithm being the second most efficient. Therefore, our experiments allow us to conclude that the algorithm proposed in this paper exhibits the best accuracy-efficiency tradeoff, being substantially as accurate as *Mittal* and *Heikkila* but dramatically more efficient. This is also highlighted by the rightmost column of Table 1, with the proposed method exhibiting the highest mean rank between the overall accuracy and efficiency ranks. Finally, it is worth pointing out that, as it is also the case of *Ohta*, since the complexity of the proposed algorithm is $O(1)$, its efficiency does not decrease with increasing neighborhood sizes, as demonstrated by the FPS values reported in the Table. This is important since, as indicated by the Table, the accuracy of the algorithms tends to improve as the neighborhood size increases, but this does not imply any loss of computational efficiency when using the proposed algorithm.

In addition to the quantitative analysis conducted so far, we also show some qualitative results in Figure 2. In particular, for each sequence we show, from top to bottom, the inferred background, one sample frame, the corresponding ground truth mask, the associated comparagram and the binary change masks yielded, respectively, by *Ohta*, *Lanza*, *Heikkila*, *Mittal* and the proposed algorithm. The masks have been obtained by using a 7×7 neighborhood size and by choosing, for each algorithm and in each sequence, the threshold value that yields a FPR equal to 5% on the whole sequence. Since *Ohta* relies on a linear model for intensity transformations due to disturbs, it performs well on S_1 , S_5 , as shown by the change masks in the first and last column of Figure 2. On the other hand, the masks in the second and, particularly, the third and fourth columns of Figure 2 demonstrate the inability of such a linear model to capture the non-linear effects of disturbs affecting S_2 , S_3 , S_4 . On the contrary, the algorithms that rely on a non-parametric order-preserving model, i.e. *Lanza*, *Heikkila* and *Mittal*, exhibit high robustness to such non-linear effects, as it is evident by looking in Figure 2 at the change masks obtained by these algorithms on S_2 , S_3 , S_4 . Nevertheless, whenever the effects of disturbs are well-modeled by a more stringent linear model, as it is case in S_1 , S_5 , these algorithms inherently suffer more than *Ohta* from the *foreground aperture* problem. That is, they are intrinsically more prone to classify erroneously as unchanged foreground pixels undergoing the same transformation as that assumed to model disturbance factors. This explains the worst quality of the change masks obtained by these algorithms with respect to *Ohta* in the first and last column of Figure 2. Thanks to the assumed non-linear order-preserving but parametric model, lying in-between the linear model of *Ohta* and the

non-parametric order-preserving model of *Lanza*, *Heikkila*, *Mittal*, the proposed algorithm achieves a better tradeoff between robustness against disturbance factors and ability in detecting foreground elements, as shown by the change masks in the last row of Figure 2. In fact, it suffers less than *Ohta* from false positives due to non-linear effects of disturbs and less than *Lanza*, *Heikkila*, *Mittal* from false negatives due to foreground aperture.

4. Conclusions

In this paper we have proposed a novel background subtraction approach relying on least-squares monotonic second-degree polynomial fitting. The ability of the assumed polynomial transformation in flexibly modeling the local effect of disturbs, on one hand, and the computational efficiency of the developed fitting procedure, on the other hand, allow the proposed approach to outperform in terms of accuracy-speed tradeoff state-of-the-art methods based on either linear or order-preserving assumptions, as vouched by experiments.

References

- [1] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, July 1997. 5
- [2] F. Crow. Summed-area tables for texture mapping. *Computer Graphics*, 18(3):207–212, 1984. 3, 4
- [3] E. Durucan and T. Ebrahimi. Change detection and background extraction by linear algebra. *Proc. IEEE*, 89(10):1368–1381, Oct. 2001. 1, 2
- [4] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *Proc. IEEE Int'l Conf. Computer Vision*, Sept. 1999. 1
- [5] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed. Moving object detection in spatial domain using background removal techniques - state-of-art. *Recent Patents on Computer Sciences*, 1:32–54, 2008. 1
- [6] M. Heikkila and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006. 1, 2, 4
- [7] A. Lanza and L. D. Stefano. Detecting changes in grey level sequences by ML isotonic regression. In *Proc. Int'l Conf. Advanced Video and Signal-based Surveillance*, pages 1–4, Nov. 2006. 4
- [8] S. Mann. Comparametric equations with practical applications in quantigraphic image processing. *IEEE Trans. Image Process.*, 9(8):1389–1406, Aug. 2000. 5
- [9] A. Mittal and V. Ramesh. An intensity-augmented ordinal measure for visual correspondence. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, volume 1, pages 849–856, June 2006. 1, 2, 4
- [10] MUSCLE Network of Excellence. Motion detection video sequences. 5
- [11] N. Ohta. A statistical approach to background subtraction for surveillance systems. In *Proc. IEEE Int'l Conf. Computer Vision*, volume 2, pages 481–486, July 2001. 1, 2, 4

- [12] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, volume 2, pages 246–252, June 1999. [1](#)
- [13] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: Real-time tracking of the human body. *19(7):780–785*, July 1997. [1](#)
- [14] B. Xie, V. Ramesh, and T. Boulton. Sudden illumination change detection using order consistency. *Image and Vision Computing*, *22(2):117–125*, Feb. 2004. [1](#), [2](#)