HIERARCHICAL LOCALLY ADAPTIVE MULTIGRID MOTION ESTIMATION FOR SURVEILLANCE APPLICATIONS

J. E. Santos Conde, A. Teuner, and B. J. Hosticka

Fraunhofer Institute of Microelectronic Circuits and Systems,
Finkenstr. 61, D-47057 Duisburg, Germany

ABSTRACT

In this communication we address the problem of detection and tracking of moving objects for surveillance or occupant detection systems. The primary goal in this framework is the motion estimation of the extracted foreground. To overcome the drawbacks characteristic of classical block matching techniques, this contribution presents a new feature based hierarchical locally adaptive multigrid (HLAM) block matching motion estimation technique based on a foreground detection procedure using an adaptive recursive temporal lowpass filter. It leads to a robust and precise motion field estimation, close to the true motion in the scene. The simulation results highlight the superior performance of the proposed method. It yields better performance than the classical exhaustive search (ES) and the modified three-step search (MTSS) technique in terms of the peak signal-to-noise ratio (PSNR).

1. INTRODUCTION

Detection and tracking of moving objects in image sequences is a substantial prerequisite for the analysis of a scene. In this contribution we present a feature based motion estimation technique developed for surveillance and occupant detection applications consisting of two main steps. In the first step we use a novel procedure which is able to detect and track independently moving objects and to indicate their instantaneous positions as well as apparent shapes with high accuracy, even for temporarily stationary objects. The approach is robust against temporal variations of the background illumination. It works also very well on noisy image sequences [1]. In the second step a hierarchical locally adaptive multigrid block matching motion estimation technique is presented, which overcomes the typical drawbacks of classical block matching techniques, namely block artefacts and unreliable motion fields in the scene. It adapts to the spatial content of the scene while estimating the motion only for the detected foreground. The two steps of the procedure will be discussed in Sections 2 and 3. Afterward some simulation results are presented that verify the superior performance of the proposed method.

2. MOVING OBJECT DETECTION AND TRACKING

All algorithms for object detection which will be investigated in this contribution belong to the class of *reference image methods* [1]. These methods are very suitable for surveillance applications with low processing power. It is assumed that the images are taken by a stationary surveillance camera system with fixed focal length. Using this assumption, the changes in the captured images will be assumed to originate from the intrinsic motion of the objects and/or the variations of illumination. A reference image $b(\mathbf{r},t)$, designated as background, will be compared with the actual input image $g(\mathbf{r},t)$ to compute the binary mask image $\nu(\mathbf{r},t)$ which indicates the membership of each pixel in the image to one of the two classes, namely back- or foreground. The indices for horizontal, vertical, and temporal directions will be $\mathbf{r} = (x,y)^T$ and t, respectively. Before introducing the proposed algorithm, we will discuss several algorithms known from the open literature.

The simplest traditional approach to the detection of moving objects is the so-called difference method which is based on computing the absolute difference of consecutive frames by forming and applying a suitable chosen constant threshold λ to this absolute difference in order to generate the binary mask $\nu(\mathbf{r},t)$. This method is thus relying on the assumption that the variation of illumination is normally slow when compared to the intensity variations caused by moving objects and that the fast variations in the spatiotemporal intensity are due to local motions. A major drawback of this approach is that slowly moving or stationary objects cannot be detected as foreground. Hence, the difference method produces an ambiguous mask image. The mask image may contain changes due to the object uncovering background, changes due to the object covering up background, and changes due to the object movement. The mentioned ambiguity can be avoided by estimating the background image using a recursive temporal lowpass (RTL) filter [2]

$$b(\mathbf{r},t) = \alpha g(\mathbf{r},t) + (1-\alpha) b(\mathbf{r},t-1), \quad 0 \le \alpha \le 1, \qquad (1)$$

where α denotes the filter coefficient which controls the speed of the background image adaption to the changes of the input image. As nonstationarities in temporal signals are often due to object motion, a background extraction can be accomplished using a simply temporal lowpass filtering which removes temporal impulses and edges. For $\alpha = 1$ the RTL-filter method yields the difference method. Thus the difference method is just a special case of the RTL-filter method. An advantageous characteristic of this type of filter is that uncorrelated noise is suppressed. The drawbacks of this approach are that objects exhibiting stop-and-go motion are adapted to the background which yields again ambiguities of the mask images, and that the choice of α is critical for determining the performance of the algorithm. In the framework of Kalman-filter theory the equation (1) can be interpreted as a recursive estimation of the background. The input image sequence can be regarded as a background image sequence contaminated by statistical noise and the moving objects. The system is controlled by a Kalman-Filter in order to adapt quickly to the illumination changes in the background, and to perform a slow adaption inside the regions including the moving objects [3].

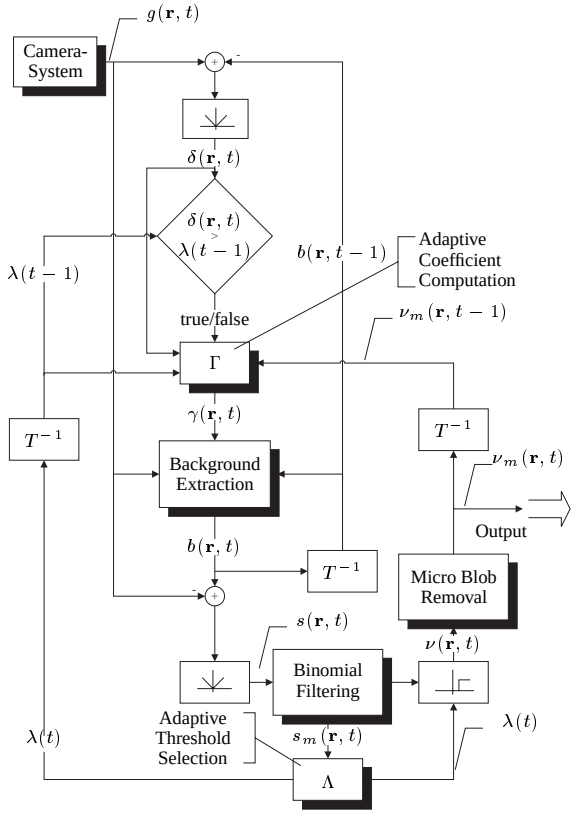

Figure 1: *Flowchart of the ARTL-filter method.*

2.1. The ARTL-Filter Method

The algorithms presented so far have the drawback that the required parameters are not adapted automatically to the observed scene. The parameters have to be set manually, which is not very useful for stand-alone surveillance systems. The performance of the algorithms is, for instance, essentially dependent on the choice of the threshold separating the foreground from the background. In most cases a fixed threshold is used, which yields unsatisfactory results. Furthermore, the filter gains are constant and cannot be adapted to the varying scenes. We propose a novel method which adapts the required parameters automatically to the observed scene. The method is based on the use of a recursive temporal lowpass filter, as discussed above, but it employs an adaptive filter gain $\gamma(\mathbf{r},t)$. This depends on the location $\mathbf{r}$ and the time t

$$b(\mathbf{r},t) = \gamma(\mathbf{r},t)g(\mathbf{r},t) + (1-\gamma(\mathbf{r},t))\,b(\mathbf{r},t-1). \qquad (2)$$

We call the proposed filter *adaptive recursive temporal lowpass filter* abbreviated as ARTL-filter. The flowchart of the method employing the ARTL-filter is depicted in Figure 1. The filter gain is computed as

$$\gamma(\mathbf{r},t) = \begin{cases} g_\beta \nu_m(\mathbf{r},t-1) & \text{if } \delta(\mathbf{r},t) \geq \lambda(t-1), \\ g_\alpha\left(\delta(\mathbf{r},t)\right)\left(1-\nu_m(\mathbf{r},t-1)\right) & \\ \quad + g_\beta \nu_m(\mathbf{r},t-1) & \text{else.} \end{cases} \qquad (3)$$

$\nu_m(\mathbf{r},t)$ is the filtered binary mask, where the elimination of small regions that have not changed within changed regions and vice versa is performed using mathematical morphology. $\delta(\mathbf{r},t)$ is the absolute difference of actual input image $g(\mathbf{r},t)$ and previous estimated background image $b(\mathbf{r},t-1)$. $g_\alpha(\delta(\mathbf{r},t))$ and g_β are the back- and foreground gain, respectively. The background gain depends on the selected threshold, and the absolute difference of the actual input image and the previous computed background image

$$g_\alpha(\delta(\mathbf{r},t)) = \exp\left(\frac{\delta(\mathbf{r},t)}{\lambda(t-1)}\right). \qquad (4)$$

The higher the absolute difference, the lower may be the background gain, assuming that the gray value changing in the background image is slow. The background gain must be low, if the selected threshold is low, to guarantee that the background is not adapted to the foreground. The foreground gain is constant and has to be sufficiently low. It allows the control over the foreground adaption. A too high foreground gain would adapt objects exhibiting stop-and-go motion completely to the background.

When taken the absolute difference $s(\mathbf{r},t)$ the borders between the subregions appear as cracks in the output mask. To avoid these cracks, the absolute values of the difference images are lowpass filtered. Using a binomial operator the smoothing can be performed very efficiently on a multigrid data structure. The lowpass filtering *fills* the cracks and makes the mask homogenous. It also suppresses uncorrelated noise in the difference image.

$\lambda(t)$ is the adaptively computed threshold adjusted to the absolute difference image $s_m(\mathbf{r},t)$. In an ideal case the histogram of the gray levels of $s_m(\mathbf{r},t)$ is bimodal. In this case, a threshold can be chosen as the gray level that corresponds to the valley of the histogram. In our situations the gray level histogram is not bimodal. Therefore, an optimal threshold is computed using the discriminant analysis [4]. The threshold selection method is nonparametric and unsupervised. In this method, the threshold operation is regarded as a separation of the pixels of the image $s_m(\mathbf{r},t)$ into two classes C_f and C_b, namely the fore- and background, at gray level ι. The optimal threshold ι_{opt} is determined by maximizing the following discriminant criterion measure, namely $\eta(\iota) = \sigma_B^2(\iota)/\sigma_T^2$, where σ_B^2 and σ_T^2 are the between-class and the total variance, respectively. The procedure utilizes only the zeroth- and the first-order cumulative moments of the gray level histogram of $s_m(\mathbf{r},t)$. To ensure a stable thresholding we set upper $\iota_{\max}$ and lower ranges $\iota_{\min}$ for the final threshold $\lambda(t)$.

3. FEATURE BASED MOTION ESTIMATION

Using block matching motion estimation, the current frame is divided into nonoverlapped rectangular blocks and the same vector is assigned to all pixels within a block. It is assumed that the image is composed of rigid objects in translational motion, justified by the fact that complex motion can be decomposed as a sum of translational components. The relative position of the closest block in the previous frame defines the motion vector associated with the present block. The displacement vector $\mathbf{p}$ is evaluated by matching the information content of a measurement window $\mathcal{W}$ with that of a corresponding measurement window within a search area $\mathcal{S}$ in the previous frame, and by searching the spatial location minimizing the matching criterion [5]

$$\mathbf{p} = \arg\min_{\mathbf{p}\in\mathcal{S}} \sum_{\mathbf{r}\in\mathcal{W}} \left|g(\mathbf{r},t) - g(\mathbf{r}-\mathbf{p},t-1)\right|. \qquad (5)$$

We have to distinguish between the notions *2D motion field* and *optical flow*. The former is the projection of the 3D motion in the scene on the 2D image plane, and the latter is the field associated with the spatiotemporal variation of intensity. In an ideal case, the optical flow corresponds to the 2D motion field. To bound the maximum displacement an object can move between two frames, the search area is limited to a maximum displacement of $\mathcal{D}$ pixels per frame for both spatial directions, resulting in $(2\mathcal{D}+1)^2$ locations to search for the best match to the present block. This exhaustive search procedure finds the optimal vector in the specified search area, but the amount of required operations are evidently too high for real time applications. Therefore, a variety of fast algorithms were proposed to reduce the computation effort by limiting the number of locations searched, such as the three-step search, one-at-a-time search, orthogonal search, and genetic search. These algorithms rely on the *unimodal error surface assumption*, namely that the distance measure increases monotonically around the location of the optimal vector. In reality the distance measure surface has several minima in which the search can be entrapped instead of the global minimum, due to the aperture problem, the inconsistent block segmentation of moving objects and background, the textured local image structure, and the luminance change between frames.

The classical block matching techniques often produce *inferior* motion vector fields as a result of fixed measurement window sizes. The obtained motion vector field is optimal in the sense of a distance measure, but habitually does not correspond to the true motion in the scene. Therefore, we propose to use a hierarchical search, which uses diverse measurement window sizes at different levels of the hierarchy [6]. The resulting motion vector field is reliable and homogeneous, close to the true motion in the scene, and the computational complexity is reduced drastically. This robust technique is able to cope with large displacements induced by fast moving objects at low computational power, ideally for object tracking in surveillance applications, and produces better performance than the ES and MTSS technique in terms of PSNR. The basic idea of the hierarchical method is to estimate a coarse and robust motion vector field at the first level of the hierarchy containing the basic motion. Then this is used as an initial estimate and further refinement is carried out with reduced measurement windows in the subsequent levels. Hereby, the local minimum problem is diminished based on the successive refinement of motion vector candidates. As small measurement windows are not capable of estimating true motion, especially in the presence of large amounts of motion, and large measurement windows cannot give accurate estimates if the constituent parts within them have different specific motion parameters, we use the *coarse-to-fine* technique. In order to take account the mentioned requirements, the proposed technique starts with a large measurement window size at the first level to estimate the major part of the displacement, decreasing the size from one level to the next level of the hierarchy to refine the resolution of the vector field. At each level of the hierarchy a separate $\log(\mathcal{D}_u+1)$-step search technique is used, where the maximum update displacement $\mathcal{D}_u$ is decreased from one level to the next level of the hierarchy (MTSS technique). The number of required steps is $N=[\log_2(\mathcal{D}_u+1)]$, where $[x]$ denotes the smallest integer larger or equal to x. The stepsize for the nth step is given by $\kappa(n)=2^{N-n}$. For $\mathcal{D}_u=7$ the $\log(\mathcal{D}_u+1)$-step search technique yields the TSS technique. To accelerate the search window procedure we use a dynamic stepsize adjustment. The stepsize convergence ratio, which is defined as $R_s=\kappa(n+1)/\kappa(n)$, is fixed at $1/4$ for the standard $\log(\mathcal{D}_u+1)$-step search technique. Instead we use a dynamical stepsize convergence ratio that can vary between two modes, namely the fast $(R_s=1/2)$ and normal mode $(R_s=1/4)$. The switching is controlled by following criterion

$$R_s=\begin{cases}1/2 & \text{if } \vartheta < T_a,\\ 1/4 & \text{else,}\end{cases} \tag{6}$$

where ϑ is the ratio between the smallest and second smallest distance measure obtained from the set of search positions in the present step. This ratio is compared to the a priori selected threshold T_a for discriminating the convergence modes $(0\le\vartheta\le 1)$. If ϑ is close to zero, a fast covergence is desired, because the search direction is probably accurate.

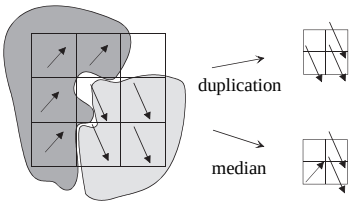

Figure 2: *Down conversion of the central block using duplication and median of the neighborhood.*

In order to reduce the block artifacts, due to the assumption that all pixels within a block have the same motion vector and to speed up the estimation, we introduce a multigrid procedure. In the first level of the hierarchy we split the image in nonoverlapping rectangular blocks of size $\mathcal{B}_i$, where i indicates the level of hierarchy. These initial blocks are labeled according to their content, namely appertaining to the fore- or background, where no motion estimation is performed on blocks belonging to the background, and their corresponding motion vectors are all zero. The membership of each block is determined using following criterion

$$\mathcal{L}_i(j)=\begin{cases}0 & \text{if } \chi_i(j)<\varepsilon,\\ 1 & \text{else,}\end{cases} \tag{7}$$

where $\mathcal{L}_i(j)$ denotes the label of the block j in level i, $\chi_i(j)$ is defined as the ratio between the number of pixels labeled as foreground and number of total pixels in block j, and ε is a threshold controlling the labeling $(0\le\varepsilon\le 1)$. $\mathcal{L}_i(j)=0$ means that the block j at level i belongs to the background. For each level a motion estimation is performed for the central pixel of the blocks appertaining to the foreground, and the same motion vector is assigned to all pixels within their block. In the next level the blocks labeled as foreground are splitted according to the quadtree manner, and the membership of each splitted block is determined according to the segmentation decision rule (7). Blocks previously labeled as background are not processed. The corresponding motion vectors are downconverted to the finer grid and refined using the median of actual and neighboring blocks, where blocks labeled as background are not considered to incorporate a spatial consistency of the motion field, and to avoid the propagation of wrong motion vector estimates throughout the levels (see Figure 2). The downconverted motion vector field serves as initial estimate for the further refinement. The procedure iterates until the final level or the minimum block is reached.

Table 1: *Parameters of the HLAM motion estimation technique* ($T_a = 0.5,\ \varepsilon = 10\%,\ \mathcal{B}_1 = 16 \times 16$). *The resulting maximum displacement is* ± 25 *pixels per frame.*

Level i	$\mathcal{W}_i$	$\mathcal{D}_{u,i}$	o_i	ρ_i
1	64×64	15	4	4
2	32×32	7	4	2
3	16×16	3	2	1

In order to reduce the computational effort resulting from large measurement windows, we introduce a subsampling in the measurement window. Hence, the search procedure is still performed on the original grid to avoid unreliable estimates. The subsampling rate ρ is adapted to the measurement window size. By applying a binomial smoothing filter of binomial order o to the image, aliasing effects are avoided and the reliability of the estimated motion vector field is improved. Herewith the risk of being trapped at a local minimum of the distance measure is reduced.

4. SIMULATION RESULTS

To evaluate the performance of the presented algorithm, we have used a test sequence for human tracking consisting of 50 frames each exhibiting a size of 256×256 pixels. In the scene a person walks in the front of a wardrobe from the right to the left. The results in Figure 3 clearly indicate the superior performance of the ARTL-filter algorithm proposed in in this contribution. The shape of the moving person is homogenous, nearly independent of slow illumination changes, and does not vary significantly if the object speed is raised, while the shadow which originates from the person is not detected as foreground. In contrast to the RTL-filter and the Kalman method, the ARTL-filter does not adapt objects moving at stop-and-go to the background, and avoids the troublesome ambiguity of the generated masks. These are necessary requirements for a surveillance system [1].

To verify the usefulness of the HLAM motion estimation technique, we compare it with the ES and MTSS technique. In our contribution, we have used three hierarchy levels, each level consisting of a separate $\log(\mathcal{D}_u + 1)$-step search technique. The parameters of the technique are tabulated in the Table 1. The ES and MTSS technique were applied on a block size of 16×16 pixels with a maximum displacement of $\mathcal{D} = 15$ pixels per frame. The performance comparison among the different techniques is based on PSNR. Figure 3 shows the difference in PSNR with respect to the ES technique. The average PSNR differences show that the presented technique is slightly superior than the ES and MTSS technique. The obtained motion vector field is more homogenous and close to the true vector field, while reducing the typical block artifacts.

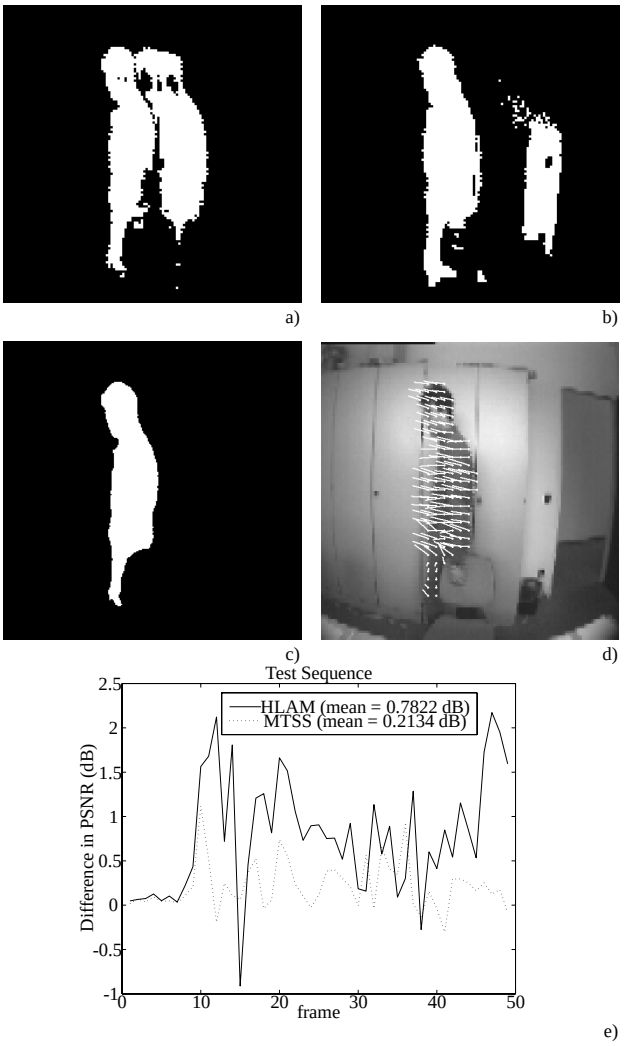

Figure 3: *Simulation results (frame #19): Extracted foreground using a) the RTL-filter method* ($\alpha = 0.5,\ \lambda = 20$), *b) the Kalman method* ($\lambda = 20$), *and c) the ARTL-filter method* ($g_\beta = 0.001,\ \iota_{min} = 10,\ \iota_{max} = 40,\ o = 6$). *d) Original frame with superimposed motion vector field. e) Performance comparison in terms of the difference in PSNR with respect to the ES technique.*

5. REFERENCES

[1] J. E. Santos Conde, A. Teuner, S.-B. Park, and B. J. Hosticka, "Surveillance system based on detection and tracking of moving objects using cmos imagers," in *International Conference on Vision Systems*, (Las Palmas de Gran Canaria, Canary Islands, Spain), Springer-Verlag, January 1999.

[2] G. W. Donohoe, D. R. Hush, and N. Ahmed, "Change detection for target detection and classification in video sequences," *Proc. ICASSP 1988*, vol. 2, pp. 1084–1087, 1988.

[3] K.-P. Karmann and A. von Brandt, "Detection and tracking of moving objects by adaptive background extraction," *Proceedings of the 6th Scand. Conf. on Image Analysis*, pp. 1051–1058, 1988.

[4] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Systems Man Cybernet.*, vol. SMC-9, pp. 62–66, January 1979.

[5] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital tv: A review and a new contribution," *Proceedings of the IEEE*, vol. 83, pp. 858–876, June 1995.

[6] M. Bierling, "Displacement estimation by hierarchical blockmatching," *SPIE Proc. Visual Communications and Image Processing'88*, vol. 1001, pp. 942–951, November 1988.