

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 210937, 8 pages
doi:10.1155/2010/210937

Research Article

Modeling of Video Sequences by Gaussian Mixture: Application in Motion Estimation by Block Matching Method

Abdenaceur Boudlal,¹ Benayad Nsiri,² and Driss Aboutajdine¹

¹LRIT Laboratory associated with CNRST, Faculty of Science, University Mohammed V-Agdal, Rabat 10080, Morocco

²LPMMAT Faculty of Science Ain Chock, University Hassan II Ain Chock, Casablanca 20100, Morocco

Correspondence should be addressed to Benayad Nsiri, nsiri2000@yahoo.fr

Received 7 October 2009; Revised 13 March 2010; Accepted 26 April 2010

Academic Editor: A. Enis Cetin

Copyright © 2010 Abdenaceur Boudlal et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article investigates a new method of motion estimation based on block matching criterion through the modeling of image blocks by a mixture of two and three Gaussian distributions. Mixture parameters (weights, means vectors, and covariance matrices) are estimated by the Expectation Maximization algorithm (EM) which maximizes the log-likelihood criterion. The similarity between a block in the current image and the more resembling one in a search window on the reference image is measured by the minimization of Extended Mahalanobis distance between the clusters of mixture. Performed experiments on sequences of real images have given good results, and PSNR reached 3 dB.

1. Introduction

Motion estimation is the process which generates the motion vectors that determines how each motion compensated prediction frame is created from the previous frame. It examines the movement of objects in an image sequence to try to obtain vectors representing the estimated motion. Motion compensation uses the knowledge of object motion obtained to achieve data compression.

Motion estimation plays a key role in many video applications, such as frame-rate video conversion, video retrieval, video surveillance, and video compression.

The key issue in these applications is to define appropriate representations that can efficiently support motion estimation with the required accuracy.

In interframe coding, motion estimation and compensation have become powerful techniques to eliminate the temporal redundancy due to high correlation between consecutive frames [1].

In real video scenes, motion can be a complex combination of translation and rotation. Such motion is difficult to estimate and may require large amounts of processing. However, translational motion is easily estimated and has been used successfully for motion compensated coding.

Most of the motion estimation algorithms make the following assumptions [2–4].

- (1) Objects move in translation in a parallel plane to the camera plane, that is, the effects of camera zoom and object rotations are not considered.
- (2) Illumination is spatially and temporally uniform.
- (3) Occlusion of one object by another and uncovered background are neglected.

Several motion estimation approaches have been proposed so far in the open literature such as pel-recursive algorithms, frequency domain techniques, optical flow, and block matching methods.

Pel-recursive Algorithms rely on iterative refining of motion estimation for individual pels by gradient methods that enable to predict recursively the displacement of each pel from its neighbouring pels. These algorithms involve more computational complexity and less regularity and are therefore difficult to realize in hardware [5]. Frequency motion estimation techniques are mainly used for the global motion estimation. The most known frequency technique is the phase correlation method that capitalizes on the well-known Fourier shift theorem which states that shifts in the

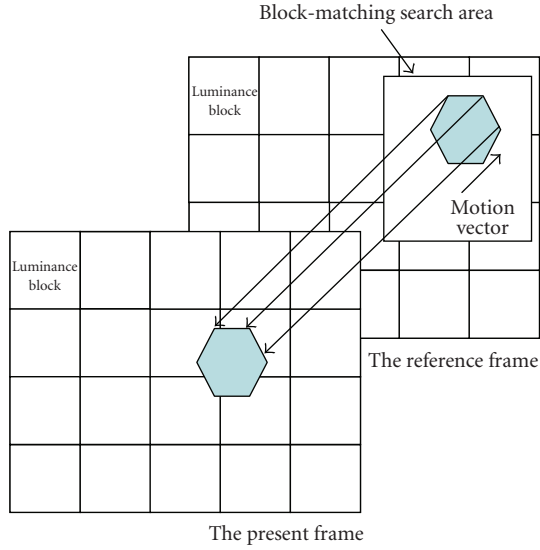


FIGURE 1: Block Matching Process, the two frames used to determine the motion vector of a given block.

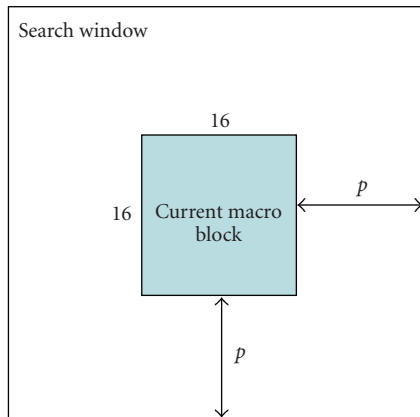


FIGURE 2: Block Matching a macroblock of side 16 pixels and a search parameter p of size 3 pixels.

spatial domain correspond to linear phase changes in the Fourier domain [6–8]. Optical flow estimation ensures high accuracy for scenes with small displacements but fails when the displacements are large. In general, these methods suffer from the aperture problem because each neighbourhood of pixels can have a different motion in the image [9–11]. Block Matching Algorithms estimate motion on the basis of rectangular blocks and produce one motion vector for each block. These algorithms are more suitable for a simple hardware realization because of their regularity and simplicity [12].

Figure 1 illustrates a process of block matching algorithm BMA. In a typical BMA, each frame is divided into blocks, each one of them consists of luminance and chrominance blocks. Usually, for coding efficiency, motion estimation is performed only on the luminance block. Each luminance block in the present frame is matched against candidate blocks in a search area on the reference frame. These

candidate blocks are just the displaced versions of the original block. The best (least distorted, i.e., most matched) candidate block is found, and its displacement (motion vector) is recorded. In a typical interframe coder, the input frame is subtracted from the prediction of the reference frame. Consequently the motion vector and the resulting error can be transmitted instead of the original luminance block; thus, interframe redundancy is removed and data compression is achieved. At the receiver end, the decoder builds the frame difference signal from the received data and adds it to the reconstructed reference frames. The summation gives an exact replica of the current frame. The better the prediction, the smaller the error signal and hence the transmission bit rate [13]. Despite the success of block matching standards method, these techniques are only based on the luminosity, and one can find misleading cases where the hypothesis of constant levels of gray is not verified. Some models take into account a factor of change of brightness in the scene. Thus the change in brightness is supposed to offset during the estimation. Such adjustments can improve the result of global changes in brightness, for which it is possible to estimate the change in a reliable manner, as against it does not take into account local variations, such as shadows. Therefore, we propose a statistical method to estimate the motion based on modeling image blocks by a mixture of Gaussian distributions. These mixtures are robust, relatively easy to use, and have traditionally learned independently, class by class, using a criterion of maximum likelihood [14–16]. The optimization is then performed using the algorithm EM (Expectation-Maximization) based on an iterative optimization of the model parameters (a priori probability, vectors means, and covariances matrices). Then we resort to the Extended Mahalanobis distances to measure the similarity and to search for the best matching between two windows spaces (or blocks) located in consecutive frames.

This paper is organized as follows. In Section 2, the modeling and parameter estimation of Gaussian mixtures is briefly described. The distance measure between Gaussian mixtures models is studied in Section 3. We describe our approach in Section 4. Section 5 presents simulation results under and without influence of noise. Some concluding remarks are given in Section 6.

2. Modeling and Parameter Estimation of Gaussian Mixtures

We consider the following Gaussian mixture model [17]

$$p\left(\frac{\mathbf{x}}{\Theta_k}\right) = \sum_{i=1}^k \alpha_i p\left(\frac{\mathbf{x}}{\theta_i}\right) = \sum_{i=1}^k \alpha_i p\left(\frac{\mathbf{x}}{\mu_i}, \Sigma_i\right), \quad (1)$$

where k is the number of components in the mixture, ($\alpha_i \geq 0$) are the mixing proportions of components satisfying $\sum_{i=1}^k \alpha_i = 1$, and each component density $p(\mathbf{x}/\theta_i)$ is a Gaussian probability density function given by

$$p\left(\frac{\mathbf{x}}{\mu_i}, \Sigma_i\right) = \frac{1}{(2\pi)^{n/2} |\Sigma_i|^{1/2}} e^{-(1/2)(\mathbf{x}-\mu_i)^T \Sigma_i^{-1} (\mathbf{x}-\mu_i)}, \quad (2)$$

where n is the dimensionality of the vector x , μ_i is the mean vector, and Σ_i is the covariance matrix assumed to be positive definite. For clarity, we let Θ_k be the collection of all the parameters in the mixture, that is, $\Theta_k = (\theta_1, \dots, \theta_k, \alpha_1, \dots, \alpha_k)$.

Given a set of N i.i.d. samples, $X = \{x_t\}_{t=1}^N$, the log-likelihood function for the Gaussian mixture model is expressed as follows:

$$\log p\left(\frac{X}{\Theta_k}\right) = \log \prod_{t=1}^N P\left(\frac{x_t}{\Theta_k}\right) = \sum_{t=1}^N \log \sum_{i=1}^k \alpha_i p\left(\frac{x_t}{\theta_i}\right), \quad (3)$$

which can be maximized to get a Maximum Likelihood (ML) [18] estimate of Θ_k via the following EM algorithm:

$$\begin{aligned} \alpha_i^+ &= \frac{1}{n} \sum_{t=1}^N P\left(\frac{i}{x_t}\right), \\ \mu_i^+ &= \frac{\sum_{t=1}^N x_t P(i/x_t)}{\sum_{t=1}^N P(i/x_t)}, \\ \sum_i^+ &= \frac{\left(\sum_{t=1}^N P(i/x_t) (x_t - \mu_j^+) (x_t - \mu_j^+)^T\right)}{\sum_{t=1}^N P(i/x_t)}, \end{aligned} \quad (4)$$

where $P(i/x_t) = \alpha_i p(x_t/\theta_i) / \sum_{i=1}^k \alpha_i p(x_t/\theta_i)$ are the posterior probabilities.

As EM is highly dependent on initialization, the first set of parameters selection is very important for EM algorithm. If the initial parameters are not well selected, the algorithm may converge into local maxima points. The convergence properties of EM algorithm over Gaussian Mixture Model have been extensively studied in [19, 20].

3. Distance Measures between Gaussian Mixtures Models

Popular distance measures from the field of statistics include the Kullback-Leibler divergence and the Extended Mahalanobis distance. The Kullback-Leibler divergence can be seen as a dissimilarity measure between two probability functions. However, it is not symmetric and does not obey the triangle inequality and is thus not a true metric. The Extended Mahalanobis distance metric can be extended to a distance measure between two distributions by combining the covariance matrices of the distributions [21]. We have no prior preference for any of these two distances; they give almost the same results, which is based on the statistical distribution of data and not on data directly. The advantage of the Extended Mahalanobis distance relies on the fact that it is easy to calculate. In practice, in the case of two Gaussian distributions $N_1(\mu_1, \Sigma_1)$ and $N_2(\mu_2, \Sigma_2)$ the measure between two means vectors is defined as follows:

$$D_{extMhn}(N_1, N_2) = \sqrt{(\mu_1 - \mu_2)^T \left(\sum_1 + \sum_2 \right)^{-1} (\mu_1 - \mu_2)}. \quad (5)$$

However, this measure creates a singularity for singular covariance matrices. In practical problems it often appears

in learning such models mixture. The acquired covariance matrix are not always conditioned and their inversion creates a problem. In our implementation, we replace the inverse of singular covariance matrix by its pseudoinverse. Singular value decomposition is used for the calculation of the pseudoinverse. Roundoff errors can lead to a singular value not being exactly zero even if it should be. Tolerance parameter places a threshold when comparing singular values with zero and improves the numerical stability of the method with singular or near-singular matrices.

4. Approach and Conception of the Proposed Method

Considering a video sequence containing moving objects, we estimate the displacement vector of each object in the image plane by the technique of Full Search Block Matching Algorithm. The current frame is divided into a matrix of ‘‘macro-blocks’’ that are then compared with the corresponding block and its adjacent neighbours in the previous frame. This enables to create a vector that stipulates the movement of a macro-block from one location to another in the previous frame. This movement, calculated for all the macro-blocks included in a frame, represents the motion estimated in the current frame. The search area for a good macro-block match is constrained up to p pixels on all four sides of the corresponding macro-block in the previous frame. This ‘‘ p ’’ stands for the search parameter. Larger motions require a larger p ; the larger the search parameter is, the more computationally expensive the process of motion estimation becomes. The idea is depicted in Figure 2. The matching of one macro-block with another is based on the output of a cost function. The macro-block that results in the least cost is the one that closely matches to current block [22, 23].

4.1. The Cost Function. The cost function is defined by Extended Mahalanobis distance weighted by the weight of Gaussian distributions components. This distance is applied between the following components: Gaussian interblocks (reference/Current), the components of strong weights, the components of medium weights, and the components of weak weights (Figure 3).

The distances d_1 , d_2 and d_3 are defined as follows:

$$\begin{aligned} d_1 &= (\alpha_1^1 \mu_1^1 - \alpha_2^1 \mu_2^1)^T \left(\alpha_1^1 \sum_1 + \alpha_2^1 \sum_2 \right)^{-1} (\alpha_1^1 \mu_1^1 - \alpha_2^1 \mu_2^1), \\ d_2 &= (\alpha_1^2 \mu_1^2 - \alpha_2^2 \mu_2^2)^T \left(\alpha_1^2 \sum_1 + \alpha_2^2 \sum_2 \right)^{-1} (\alpha_1^2 \mu_1^2 - \alpha_2^2 \mu_2^2), \\ d_3 &= (\alpha_1^3 \mu_1^3 - \alpha_2^3 \mu_2^3)^T \left(\alpha_1^3 \sum_1 + \alpha_2^3 \sum_2 \right)^{-1} (\alpha_1^3 \mu_1^3 - \alpha_2^3 \mu_2^3). \end{aligned} \quad (6)$$

When modeling by a mixture of two Gaussian distributions, the cost function is defined by the Extended

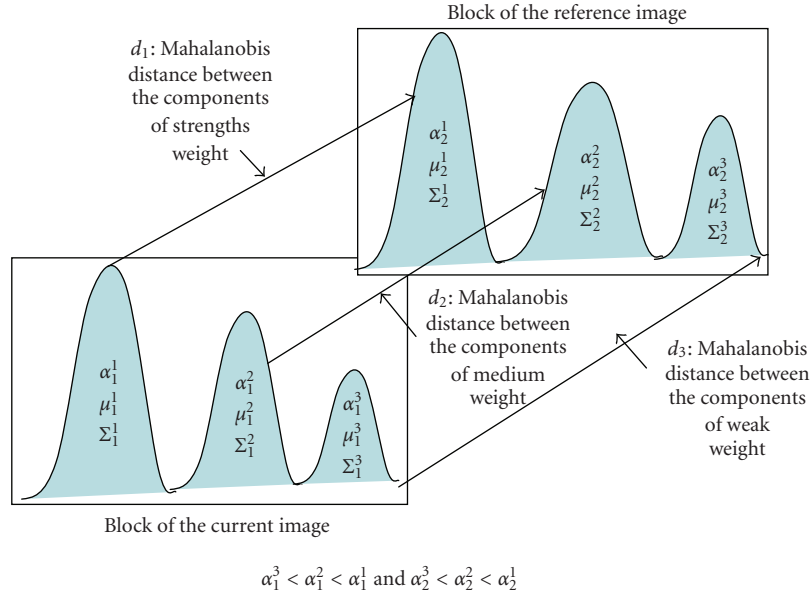


FIGURE 3: Extended Mahalanobis distance between the components of strong weights (d_1), the components of medium weights (d_2), and the components of weak weights (d_3).

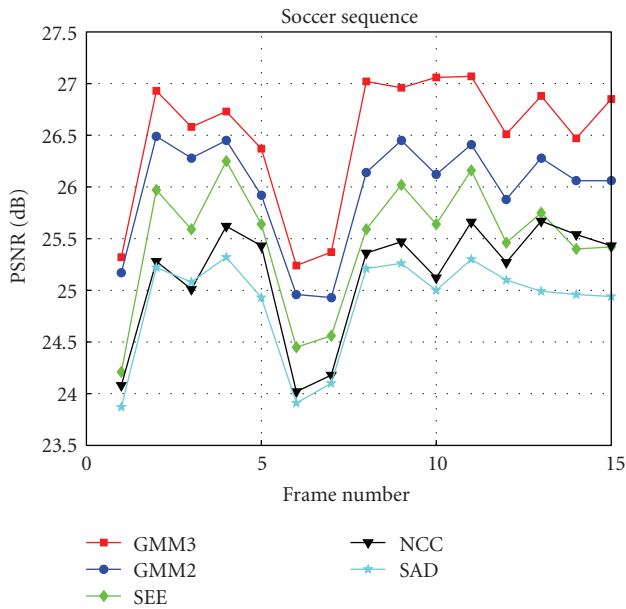


FIGURE 4: PSNR comparisons of GMM3, GMM2, SAD, SSE, and NCC algorithms for the “Soccer” sequence without influence of noise.

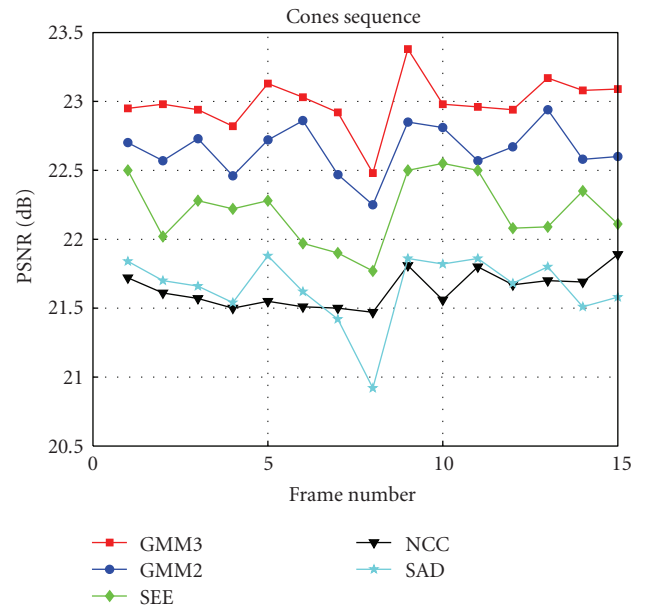


FIGURE 5: PSNR comparisons of GMM3, GMM2, SAD, SSE, and NCC algorithms for the “Cones” sequence without influence of noise.

Mahalanobis distance between the components of strong weights (d_1) and the components of weak weights (d_3).

4.2. Steps of the Proposed Method. The proposed method is based on three steps design.

- (1) Each block in the reference image or the current image is modeled by a mixture of three Gaussian

distributions. This modeling consists in estimating the parameters of the mixture (weight, means vectors, and covariance matrix).

- (2) The Parameters are sorted based on their weights in mixture. This allows the identification of the components of weak weights, the components of medium weights, and the components of strong weights.

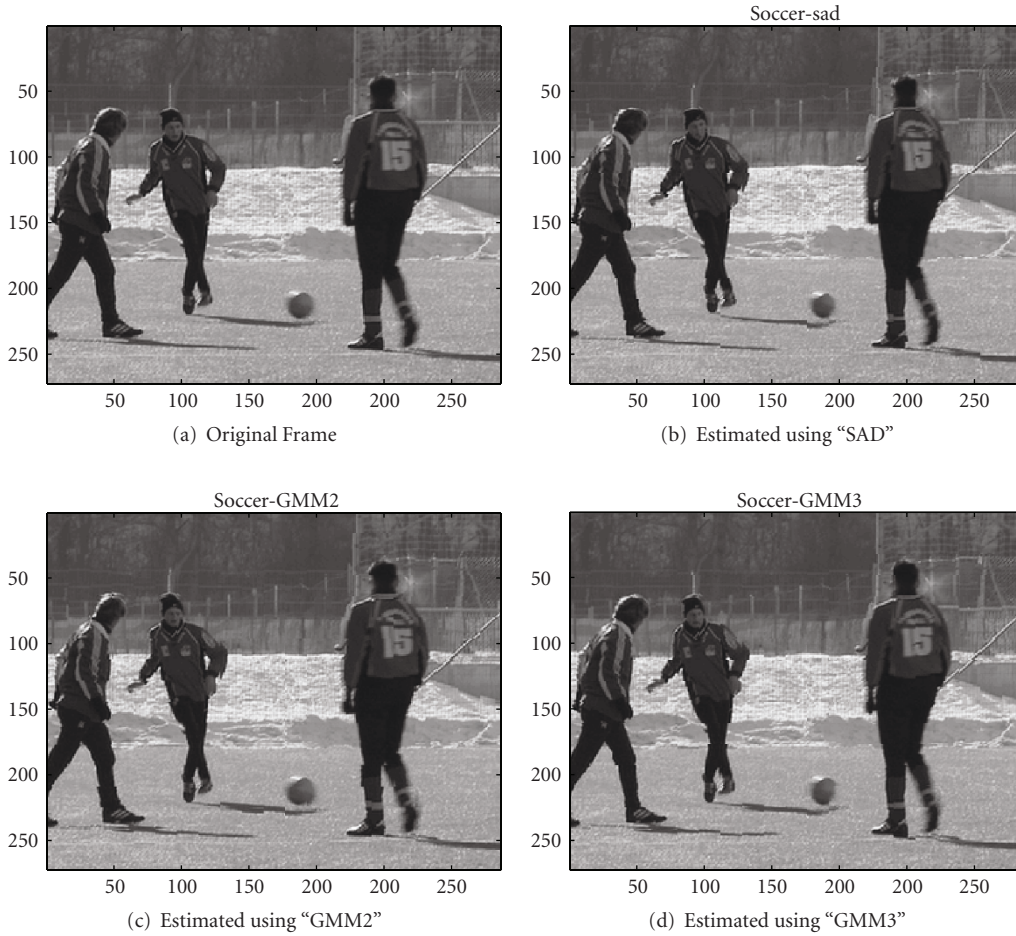


FIGURE 6: Subjective Image Quality.

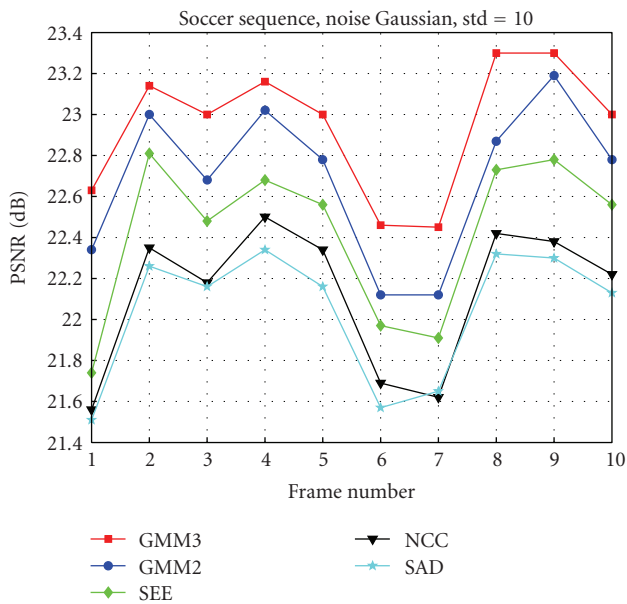


FIGURE 7: PSNR comparisons of SAD, SSE, NCC, GMM2, and GMM3 methods for the “Soccer” sequence with influence of Gaussian noise for standard deviation equal to 10.

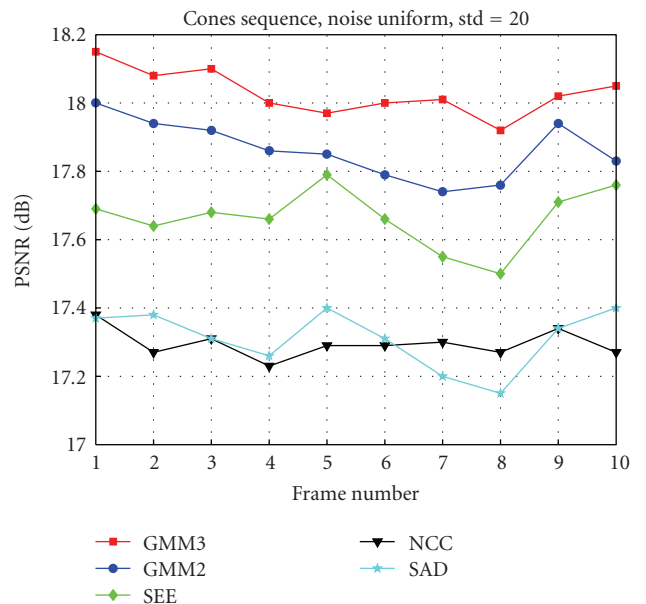


FIGURE 8: PSNR comparisons of SAD, SSE, NCC, GMM2, and GMM3 methods for the “Cones” sequence with influence of Uniform noise for standard deviation equal to 20.

TABLE 1: Matrix M_1 .

68.08	135.85	19.66
19.66	19.66	50.00
193.15	135.85	83.99

TABLE 2: Matrix M_2 .

12.66	0.93	15.90
15.90	15.90	56.04
14.36	70.64	13.80

TABLE 3: Matrix M_3 .

146.65	137.85	13.80
13.80	13.80	130.26
3.60	144.59	15.21

(3) Research of minimal interblocks distance (reference/current).

(a) The Extended Mahalanobis distances between a block of the current image and all blocks in a search window $[-1, +1]$ in the reference image are stored in the matrices M_1 , M_2 , and M_3 .

(M_1) matrix contains the values of Extended Mahalanobis distances between the components of weak weights.

(M_2) matrix contains the values of Extended Mahalanobis distances between the components of medium weights.

(M_3) matrix contains the values of Extended Mahalanobis distances between the components of strong weights.

(b) The value of the minimal distance of the three matrices M_1 , M_2 , and M_3 corresponds to the most similar block in reference image.

4.3. Practical Considerations. Matrices M_1 , M_2 , and M_3 show the Extended Mahalanobis distances between a block of the current image and all blocks in a search window $[-1, +1]$ in the reference image of the Foreman sequence.

Matrix M_1 contains the distances between the components of weak weights.

Matrix M_2 contains the distances between the components of medium weights.

Matrix M_3 contains the distances between the components of strong weights.

The value of the minimum distance of the three matrices M_1 , M_2 , and M_3 is equal to 0.93 corresponding to the first line second column indices in the matrix M_2 . These indices correspond to the most similar blocks in the reference image.

For these types of Foreman sequence, about 80% minimum distances are in the matrix M_1 (distances between the components of weak weights). This percentage mainly depends on statistical characteristics of pixels in the image.

5. Experimental Results

The parameters of motion estimation used for comparing and evaluating the quality of the obtained results from our proposed methods are as follows.

(i) Method: exhaustive block-matching (full search) is the most obvious candidate for a search technique for finding the best possible weight in the search area.

(ii) Classical criterion methods.

(a) Sum of Absolute Differences “SAD”.

(b) Sum of Square Error “SSE”.

(c) Normalized Cross-Correlation “NCC”.

(iii) Method proposed criterion: minimization of Extended Mahalanobis distance between mixture of two and three Gaussian distributions (“GMM2” and “GMM3”).

(iv) Precision: pixel.

(v) Block Size: 16×16 .

(vi) Search area: $[-1, +1]$.

5.1. Simulation Results without Noise Influence

5.1.1. Objective Evaluation. The proposed approach was evaluated by using a standard measure, the average PSNR (Peak Signal to Noise Ratio), given as

$$\text{PSNR}_{\text{avg}} = \frac{1}{F} \sum_{i=1}^k \text{PSNR}_i, \quad (7)$$

where PSNR_i is the measured PSNR for frame i , and F is the total number of frames. We will compare the “GMM3” and “GMM2” methods against the “SAD”, “SSE”, and “NCC” methods. In addition, the PSNR comparison among the five algorithms will be introduced.

As an example, Table 4 shows that six test image sequences have different characteristics. While “Foreman” represents the characteristics of slow motion image sequences, “Hand” and “Soccer” are fast motion image sequences, “coastguard” is global motion image sequence, and Football and Cones contain multiple objects moving. Table 4 shows that the performance of “GMM3” and GMM2 methods are better than “SAD”, “SSE”, and “NCC” methods for the test video sequences.

Another performance comparison is made among the first 15 frames of each sequence. As an example, Figures 4 and 5 show the performance comparison for the first 15 frames of “Soccer” and “Cones” sequences. The PSNR comparison shows that the “GMM3” and “GMM2” usually perform better than the “SAD”, “SSE”, and “NCC” algorithms.

5.1.2. Subjective Evaluation. To demonstrate the performance of our algorithm, we applied the new methods of motion estimation (“GMM2” and “GMM3”) on the “soccer” sequences (Figure 6). The players shadows (Figure 6(c)) and (Figure 6(d)) are better estimated by the methods “GMM2” and “GMM3” compared to the ones estimated by the method “SSE” (Figure 6(b)).

TABLE 4: Average PSNR Values for test images.

Sequences	Algorithm	PSNR [dB]
"Hand" 255 × 255	"CORR"	22.49
	"SAD"	21.73
	"SEE"	22.31
	"GMM2"	24.05
	"GMM3"	24.64
"Foreman" 175 × 256	"CORR"	27.04
	"SAD"	28.60
	"SEE"	28.53
	"GMM2"	29.00
	"GMM3"	29.58
"coastguard" 240 × 240	"CORR"	23.37
	"SAD"	23.21
	"SEE"	23.93
	"GMM2"	24.31
	"GMM3"	24.86
"Soccer" 335 × 270	"CORR"	24.08
	"SAD"	23.87
	"SEE"	24.21
	"GMM2"	24.80
	"GMM3"	25.18
"Football" 240 × 336	"CORR"	24.26
	"SAD"	24.24
	"SEE"	24.88
	"GMM2"	25.22
	"GMM3"	25.27
"Cones" 256 × 256	"CORR"	21.63
	"SAD"	21.64
	"SEE"	22.20
	"GMM2"	22.64
	"GMM3"	23.00

5.2. *Simulation Results under Influence of Noise.* Additive Gaussian noise with standard deviation equal to 10 and Uniform noise with standard deviation equal to 20 degraded the video sequences. We applied the motion estimation algorithms "SAD", "SSE", "NCC", "GMM2", and "GMM3" on Soccer and Cones sequences. The results are summarized in Figures 7 and 8.

6. Conclusion and Perspective

In this paper we have modeled sequence images blocks by a mixture of two and three Gaussian distributions and have used block matching criterion based on Mahalanobis distance minimization between the clusters of mixture to estimate motion. This technique has been compared to other equivalent methods in the literature. The simulation confirms that the proposed technique allows the significant PSNR gains. These gains can be observed in terms of both the perceptual quality and the PSNR of the restored images. However, this technique requires more computations. It might be necessary to further reduce computations to fit

real time requirements. Parameters estimation of Gaussian mixture consists of repetitive operations which could greatly benefit from some existing architectures to perform repetitive tasks efficiently. A forthcoming work will be devoted to improve the speed of execution and further increase performance by modeling the blocks of the image by a Gaussian mixture where the number of clusters varies.

Acknowledgments

The authors would like to thank Mohamed Zyoute, Fatima Boudlal, Fadoua Ataa-Allah Nizar Bennani, and Hatim Chergui for their help.

References

- [1] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the DARPA Image Understanding Workshop*, pp. 121–130, 1981.
- [2] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: a review and a new contribution," *Proceedings of IEEE*, vol. 83, no. 6, pp. 858–876, 1995.
- [3] G. Tziritas and C. Labit, "Motion analysis for image sequence coding," in *Advances in Image Communication*, vol. 4, Elsevier, Amsterdam, The Netherlands, 1994.
- [4] C. Stiller and J. Konrad, "Estimating motion in image sequences, a tutorial on modeling and computation of 2D motion," *IEEE Signal Processing Magazine*, vol. 16, no. 4, pp. 70–91, 1999.
- [5] V. Estrela, L. A. Rivera, P. C. Beggio, and R. T. Lopes, "Regularized pel-recursive motion estimation using generalized cross-validation and spatial adaptation," in *Proceedings of the 16th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI '03)*, pp. 331–338, 2003.
- [6] V. Argyriou and T. Vlachos, "Performance study of gradient correlation for sub-pixel motion estimation in the frequency domain," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 152, no. 1, pp. 107–114, 2005.
- [7] S. Ertürk, "Digital image stabilization with sub-image phase correlation based global motion estimation," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1320–1325, 2003.
- [8] F. Essannouni, R. Oulad Haj Thami, A. Salam, and D. Aboutajdine, "A new fast full search block matching algorithm using frequency domain," in *Proceedings of the 8th International Symposium on Signal Processing and Its Applications (ISSPA '05)*, vol. 2, pp. 559–562, 2005.
- [9] D. J. Heeger, "Optical flow using spatiotemporal filters," in *Proceedings of the 1st International Conference on Computer Vision*, pp. 181–190, 1988.
- [10] L. Alvarez, J. Weickert, and J. Sánchez, "Reliable estimation of dense optical flow fields with large displacements," *International Journal of Computer Vision*, vol. 39, no. 1, pp. 41–56, 2000.
- [11] L. Jacobson and H. Wechsler, "Derivation of optical flow using a spatiotemporal-frequency approach," *Computer Vision, Graphics, & Image Processing*, vol. 38, no. 1, pp. 29–65, 1987.
- [12] C. Zhu, W.-S. Qi, and W. Ser, "A new successive elimination algorithm for fast block matching in motion estimation," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '04)*, vol. 3, pp. 733–736, 2004.

- [13] J.-B. Xu, L.-M. Po, and C.-K. Cheung, "Adaptive motion tracking block matching algorithms for video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 7, pp. 1025–1029, 1999.
- [14] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical, Series B*, vol. 39, pp. 1–38, 1977.
- [15] R. Neal and G. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, M. I. Jordan, Ed., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.
- [16] M. Afify, "Extended baum-welch reestimation of Gaussian mixture models based on reverse jensen inequality," in *Proceedings of the 9th European Conference on Speech Communication and Technology*, pp. 1113–1116, 2005.
- [17] G. J. McLachlan and D. Peel, *Finite Mixture Models*, John Wiley & Sons, New York, NY, USA, 2000.
- [18] S. Sanjay-Gopal and T. J. Hebert, "Bayesian pixel classification using spatially variant finite mixtures and the generalized EM algorithm," *IEEE Transactions on Image Processing*, vol. 7, no. 7, pp. 1014–1028, 1998.
- [19] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, 1984.
- [20] L. Xu and M. I. Jordan, "On convergence properties of the EM algorithm for Gaussian mixtures," *Neural Computation*, vol. 8, no. 1, pp. 129–151, 1996.
- [21] K. Younis, M. Karim, R. Hardie, J. Loomis, S. Rogers, and M. DeSimio, "Cluster merging based on weighted Mahalanobis distance with application in digital mammography," in *Proceedings of the IEEE National Aerospace and Electronics Conference (NAECON '98)*, pp. 525–530, July 1998.
- [22] R. A. Packwood, M. K. Steliaros, and G. R. Martin, "Variable size block matching motion compensation for object-based video coding," in *Proceedings of the 6th IEE International Conference on Image Processing & Its Applications*, pp. 56–60, Dublin, Ireland, July 1997.
- [23] A. Barjatya, "Block matching algorithms for motion estimation," *IEEE Transactions Evolution Computation*, vol. 8, no. 3, pp. 225–239, 2004.