DENSE OPTICAL FLOW FIELD ESTIMATION USING RECURSIVE LMS FILTERING

Mejdi Trimeche [(a)], *Marius Tico* [(a)] *and Moncef Gabbouj* [(b)]

[(a)] Multimedia Technologies Laboratory,
Nokia Research Center
Visiokatu 1, 33720, Tampere, Finland

[(b)] Institure of Signal Processing,
Tampere University of Technology
FIN-33720, Tampere, Finland

email: mejdi.trimeche@nokia.com

ABSTRACT

In this paper, we present a novel recursive method for pixel-based motion estimation. Assuming small displacements, we use an adaptive LMS filtering scheme to match the intensity values and calculate the displacement between two adjacent video frames. Using a sliding window from the template image, the proposed algorithm employs a simple 2-D LMS filter to adapt the corresponding set of coefficients in order to match the pixel value in the reference frame. The peak value in the resulted coefficient distribution points to the displacement between the frames at each pixel position. The experiments demonstrate good results because the filtering takes advantage of the localized correlation of image data in adjacent frames, and produces refined estimates of the displacements at sub-pixel accuracy. One particular advantage is that the proposed method is flexible and well suited for the estimation of small displacements within video frames. The proposed method can be applied in several applications such as super-resolution, video stabilization and denoising of video sequences.

1. INTRODUCTION

Motion estimation establishes the correspondences between the pixel positions from a target frame with respect to a reference frame. This operation is fundamental to many image and video processing tasks. Although the understanding of the issues involved in the computation of motion has significantly increased, we are still far from generic, robust and real-time algorithms [11]. In this paper, we are mostly concerned with image based motion estimation techniques that can be used for video filtering applications such as video denoising, video stabilization [12] and super-resolution [13].

Two different approaches for motion estimation have been separately developed; i. e. image based discrete motion estimation (block matching), and gradient based techniques, or optical flow estimation. The discrete motion estimation establishes the correspondences by measuring similarity using blocks or masks. In general, the advantages of block-matching are simplicity and reliability for discrete large motion. However this approach has been mainly developed to improve compression performance in video coding applications. As a result, the motion vectors do not necessarily reflect real motion, and suffer from discontinuities along the block boundaries, additionally, they generally fail to register detailed motion. Hence, the direct application of block-based motion estimation into filtering applications is not a viable option. On the other hand, optical flow estimation methods aim to obtain a velocity field using the computation of spatial and temporal image derivatives. There are several variations in the literature based on this approach targeting different computer vision applications, but here we refer to those methods that derive a dense flow field by computing the gradient images and possibly solving for local parametric motion at each pixel position. In general the techniques based on this approach produce accurate estimates of the elastic displacement between frames and handle the piecewise motion adequately. However, the computational complexity associated with optical flow estimation techniques is usually prohibitively expensive (see table 1 in [4]). Based on the arguments mentioned above, one may conclude that the traditional approaches for motion estimation are not suitable for motion-compensated filtering applications, mainly because they have been designed for different applications. In this paper, based on the analogy from audio echo-cancelation for matching of delayed components, we utilize 2-D spatially adaptive LMS filtering in order to derive a fast motion estimation algorithm. We claim that since our algorithm is simple enough, it can easily be adapted and used in different video filtering applications.

1-D LMS filtering has been extended earlier to 2-D case and has been applied in various image processing tasks such as image enhancement [10]. In [3], a 2-D block diagonal LMS algorithm was developed for image processing applications. It was claimed that it is possible to preserve the local correlation information of the pixels in both directions when utilizing a 2-D diagonal scanning pattern. In [9], a two-dimensional recursive least squares (LS) filtering scheme was introduced. The filter was tuned to remove the mismatching effects in a stereo image pair, and the weights of the filter were computed using a block-based LS method. In this paper, we utilize a 2-D LMS filter in order to match the input images, and we utilize the corresponding coefficients distribution in order to extract sub-pixel motion information. In the context of block motion estimation, it was suggested in [7] that the estimation of motion vectors based on the spatio-temporal neighborhood information is an effective solution to reduce the effects of uneven error surface. In [6], an adaptive matching scan was employed to reduce the amount of computations needed to perform the full-search block-matching algorithm. In the proposed algorithm, we do not use block matching, but direct pixel based motion estimation. Further, we exploit the LMS filters to obtain the smoothness and realize the adaptive matching scan.

LMS filters do not make a priori assumptions about the statistics of the image data, this enables robust performance against various types of noise and outlier areas that may be present in the image pair. In this paper, we exploit this property to develop a novel motion estimation algorithm that can be used in video filtering applications. The algorithm is based on 2-D LMS matching algorithm which adapts a window of coefficient values to match the central pixel value. The resulted coefficient distribution reveals the localized displacement that happens between two successive frames. In the following Section, we present the observation model which assumes small displacement between two successive frames. In Section 3, we introduce the LMS matching filter, and then the procedure we used to extract the motion from the adapted coefficient distribution. In Section 4, we discuss the effect of the scanning from one direction, and we propose a method to scan from different directions and to combine the final result. We present experimental results in Section 5 using synthetic test set to illustrate the capability of the proposed algorithm to detect motion as well as motion outliers. Finally, in Section 6, we discuss the advantages of the proposed method in the context of video filtering and we present our conclusions.

2. OBSERVATION MODEL

Consider two successive frames of a video sequence, a reference image I, and a template image T that we would like to register with respect to I. Both images have the same size (X,Y). The images are ordered lexicographically into vectors, such that $I(k)$ and $T(k)$ denote the intensity values on the grid position k $(1 \leq k \leq XY)$. We want to estimate the displacement field $D(k) = [u(k), v(k)]'$, which establishes the correspondence between $I(k)$ and $T(k)$. We assume that the relative displacement $D(k)$ is constrained, such that

$$\begin{cases} -s \leq u(k) \leq s \\ -s \leq v(k) \leq s \end{cases} \tag{1}$$

In order to solve for the pixel-based motion estimation problem, the following cost function may be considered

$$j(k) = [T(k) - \tilde{I}(k + D(k))]^2 \tag{2}$$

$\tilde{I}$ denotes the estimated intensity value of the reference image after performing the motion compensation. Note that the displacement $D(k)$ need not be integer valued. In Eq. 2, we chose the simple quadratic functional of the registration error for tractability of the formulation, especially in case of Gaussian additive noise.

The main hypothesis in our formulation is that the pixel value $I(k)$ in the reference image can be expressed as an estimate using a linear filter combination of the window around the central pixel location $T(k)$ in the template image. That is:

$$I(k) = w(k)' * T_w(k) + \eta(k) \tag{3}$$

where $T_w(k)$ is a matrix of windowed pixel values from the template image with size $S = (2s+1)^2$ and centered around the pixel position k. $w(k)$ corresponds to the modulating coefficient matrix, $\eta(k)$ is an additive noise term. For notation convenience, the matrices $T_w(k)$ and $w(k)$ are ordered lexicographically into column vectors, and $'$ denotes the transpose operation.

The model in Eq. 3 tells that each pixel value in the reference image can be estimated with a linear model of a window

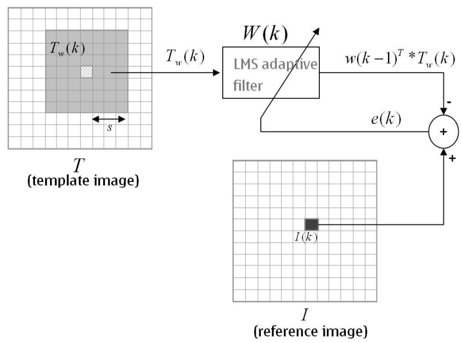

Figure 1: Illustration of the filtering process that is used to adapt the coefficients that are used to matching the frames.

that contains the possible shifted pixels in the template image. In this setting, the motion estimation problem can be mapped into the simpler problem of linear system identification, i. e., we have the desired signal $I(k)$, the input data $T_w(k)$, and and we would like to estimate $w(k)$ according to the formation model in Eq. 3. The goal is to minimize the cost function in Eq. 2 by limiting the motion search within the bounds expressed in Eq. 1.

3. 2-D LMS ADAPTIVE PIXEL MATCHING

The 2-D LMS filter is essentially an extension of its 1-D counterpart. In our solution, it takes the two dimensional window $T_w(k)$ as input data and the desired response to be matched is the intensity value in the reference image $I(k)$. In order to solve for the weight array $w(k)$, we apply the standard LMS recursion [5]. The recursion is applied along a pre-determined scanning path of the image grid (indexed by n), as follows

$$\begin{cases} w(n) = w(n-1) + \mu(n) T_w(k) e(n) \\ e(n) = I(k) - w(n-1)' T_w(k) \end{cases} \tag{4}$$

where $\mu(n)$ is a positive step size parameter, $e(n)$ is the output estimation error, n refers to the iteration number, and k denotes the current pixel position that we are filtering. Note that if the indexing of the pixels k is the same as the indexing of the scanning path, then n and k are identical. $w(n-1)$ refers to the coefficient values that were estimated in the previous pixel position following the employed scanning direction (see the following section for discussion). Fig. 1 shows an illustration of this basic filtering process.

Like its 1-D counterpart, the 2-D adaptive filter does not assume any knowledge of the cross correlation functions [10]. The filter approximates their values by using instantaneous estimates at each pixel position according to the step size μ. For LMS filters, there is a well-studied trade-off between stability and speed of convergence, i. e. a small enough step size $\mu(n)$ will result in slow convergence; whereas a large step size may result in unstable solutions. Alternatively, there are several modifications of the standard LMS algorithm that offer simpler stability requirements, for example, the normalized LMS (NLMS). The NLMS algorithm is obtained by substituting in Eq. (4) the following step size:

$$\mu(n) = \frac{\mu}{\varepsilon + \|T_w(k)\|^2} \tag{5}$$

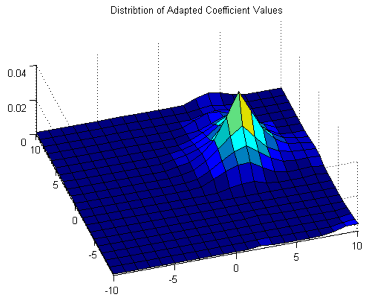

Figure 2: Example distribution of adapted coefficient values. The peak value points to the displacement that happened between the two frames at pixel location k.

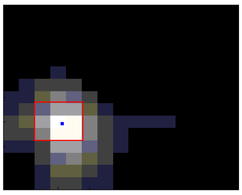

Figure 3: Obtaining the displacement from the coefficient distribution.

where ε is a small positive constant. In this form, the filter is also called $\varepsilon - NLMS$ [1], and the stability condition is given by:

$$\mu < \frac{2}{3} \tag{6}$$

The choice of the step size parameter is critical in tuning the proper performance of the overall algorithm. In general, the motion can be assumed locally stationary, so we would like to tune our algorithm to use a small step size μ in order to favor smooth and slowly varying motion field, rather than a spiky and fast changing motion field.

3.1 Determining the Motion from the Adapted Filter Coefficients

The function of the 2-D LMS filter is to match the pixels in a search window on the template image to the central pixel in the reference image. This matching is done through the smooth modulation of the filter coefficient matrix. Fig. 2 shows a plot of the coefficient values, which peak at the position of the displacement between the corresponding images.

In order to obtain the displacement vector $D(k)$ from the adapted coefficient distribution $w(k)$, we apply a simple filtering operation, which first finds the cluster of neighboring coefficients that contains the global maximum coefficient value (Fig. 3). Then, the center of mass of this cluster is calculated over the support window. The result in the x and y directions make up the horizontal and vertical components of $D(k)$ at sub-pixel accuracy. We inserted a simple intermediate check to assert whether we can confirm motion from the coefficient distribution. Below, we describe the filtering operation in more detail:

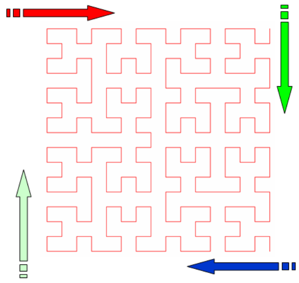

Figure 4: Possibility to employ more elaborate scanning patterns, in this example we propose to use mirrored Hilbert scanning patterns and traverse the image plane from 4 different directions.

1. Find the 3×3 window support, over which the sum of neighboring coefficients is maximum.
2. Check that the sum is larger than a pre-determined threshold (confidence in estimation process). If not, assert an empty pointer.
3. Calculate the center of mass over the obtained 3×3 support window. The vector from the origin to the resulting position is the estimated motion vector.

4. SCANNING DIRECTION

The proposed filtering method is based on recursive scanning of the 2D image grid. As a consequence, the employed scanning pattern impacts the coefficient adaptation, especially if we favor stable adaptation by using a small step size μ. This means that the overall estimation process is spatially causal with respect to the employed scan method. In case the motion is global stationary and constrained, which may be for instance due to camera shaking with respect to a fixed scene, even the simplest of scanning patterns, e.g. raster scan, is sufficient to correctly estimate the stationary displacement.

4.1 Multiple Scanning Directions

On the other hand, if we want to detect arbitrary and localized motion, it may not be possible to estimate the corresponding motion field by utilizing a single scanning direction. Instead, we can perform the scanning in four different directions and obtain the displacement field independently for each scanning direction, the final motion field can be obtained by combining the resulted motion fields. The combining of the displacement vectors can be performed by selecting the vector that minimizes the corresponding error value at each pixel location (error images due to LMS adaptation are stored temporarily in the memory). Another elegant method is to apply a component-wise scalar median filter (or vector median [2]) for the obtained displacement vectors; this allows to obtain the consolidated motion through a voting process and enhance the performance of the estimation process against outliers.

4.2 Enhanced Scanning Patterns

Additionally, instead of the basic raster scan, space-filling curves [8] can be used to traverse the image plane while adapting the LMS coefficients. The typical space filling patterns (e.g. Peano and Hilbert curves [8]) are defined over grid areas that are powers of 2. Fig. 4 shows an example

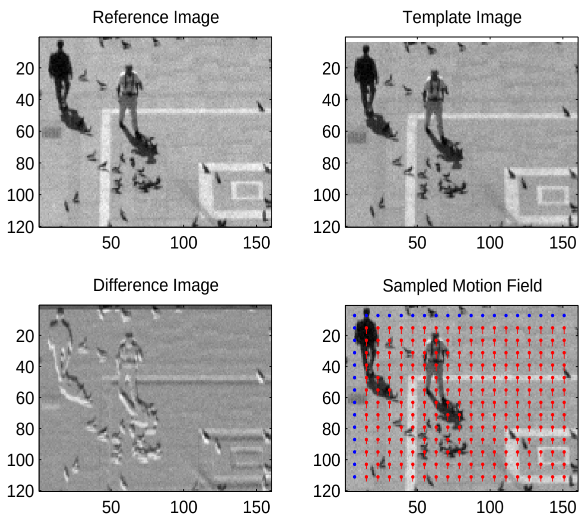

Figure 5: Example of the estimated motion field that is obtained by scanning in a single direction (horizontal). Global translation and noisy input images (Gaussian noise, $\sigma^2 = 40$). Blue points stand for points where algorithm cannot resolve any motion with certainty.

of the Hilbert scanning pattern for a rectangular window of 16x16. This mode of scanning through the pixels, though more complicated, has the important advantage of staying localized within areas of stationary shifts before moving to another area. This scanning mode typically results in superior performance of the overall estimation process, especially in the presence of localized motion or other random outliers. The pattern in Fig. 4 can be easily mirrored and traversed from four directions as discussed previously.

5. EXPERIMENTAL RESULTS

In this section, we briefly show the performance of the proposed approach for motion estimation. we present 3 different experiments to simulate practical situations that may arise in video filtering applications.

In the first experiment illustrated in Fig. 5, we generated a template image from an original reference image by simulating a global translation $(D = [0, 3.5])$. We added Gaussian noise $(\sigma^2 = 40)$ to the template and the reference image. We tested our algorithm with the following parameters $(s = 7, \mu = 0.02)$[1]. We used a simple raster scan to adapt the LMS coefficients. The sampled motion field is displayed in Fig. 5; the red arrows display the estimated displacement vectors, whereas the blue points show areas where the algorithm cannot resolve constrained motion with certainty, these areas generally contain little image details, which confuses the LMS adaptation. The algorithm was successful in determining the global translational motion, for instance, the motion vector that was estimated in the middle of the image was $D_{est} = [0.048, 3.513]$. In fact, since the step size μ was small, the overall performance was robust against noise, meanwhile the coefficient adaptation was capable to follow the stationary delay that is consistently confirmed in the areas of the image that contain contrasted details.

In the second experiment, we generated a template image by warping the reference image with an affine transformation. The test image is meant to simulate the homography effect that may happen due to rotating camera motion with respect to a parallel plane. Fig. 6 displays the estimated motion field that was obtained with the following algorithm parameters $(s = 15, \mu = 0.02)$. The filtering was performed using a single scanning direction (raster scan). In the border area, the algorithm did not detect motion due to the absence of any image details that fall inside the search window. However, in the center of the image, the algorithm was capable to track the smoothly varying motion field.

In the third experiment, we generated a template image by translating the reference image and by inserting an outlier area in the middle of the image. This experiment is designed to simulate the performance of the algorithm in the presence of combined motion. We used the following algorithm parameters $(s = 10, \mu = 0.02)$. In this setting, we used the block Hilbert scanning to traverse the image plane from 4 different directions, and we finally fused the obtained displacement vector components using a median filter. Fig. 7 displays the estimated motion field. The blue points show areas where the algorithm cannot resolve for motion with certainty, which corresponds well to the outlier area that was in the template image. This reveals that the use of multiple scanning from different directions and the subsequent voting process through the median selector adds robustness to the motion estimation. This approach is potentially very useful in video filtering applications, since the detected outlier points can be left out from the filtering process to avoid unwanted artifacts.

In all experiments, the obtained dense motion field is smooth and spatially correlated, which reflects better the real motion that happened in the video frames. The complexity of the algorithm is $O((2s+1)^2 N)$. Comparing to the complexity of optical flow methods (table 1 in [4]), the proposed method is much simpler and faster, thus enabling real-time implementations of motion compensated filtering.

6. CONCLUSIONS

In this paper, we presented a novel recursive method for pixel-based motion estimation. The proposed algorithm employs 2-D LMS filter to adapt a window of coefficients so that we can match the pixel value in the reference frame. The peak value in the resulted coefficient distribution points to the displacement between the frames at each pixel position. The recursive LMS filtering along the scanning direction enables to track the stationary shifts that happen between the reference and template frames, and inherently produces smooth estimates of the displacements, directly at sub-pixel accuracy. We also proposed variations of the initial algorithm such as the use of multiple scanning directions and patterns in order to track complicated motion in the scene.

Experimental results have demonstrated that the overall performance was robust against Gaussian noise. Also, the algorithm was capable to accurately track smooth affine motion, even when using a single scanning direction. When using multiple scanning directions, it was possible to single out outlier regions which correspond for example to moving or

[1] The search extend (s) can be set according to the video type and the target filtering application. When we know the motion is constrained, smaller search windows ensure faster operation and improved precision.

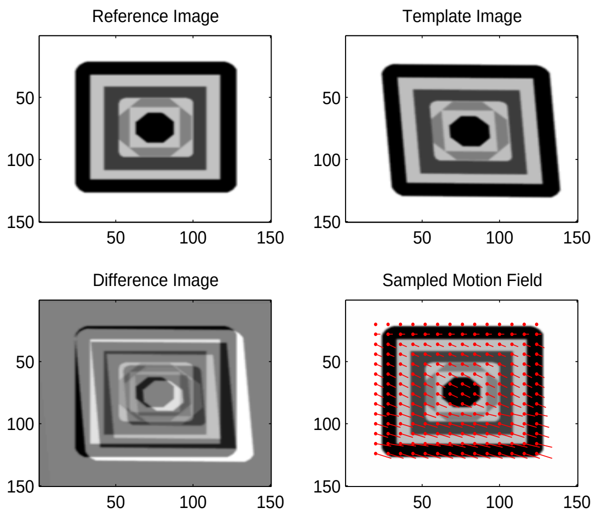

Figure 6: Example of the estimated motion field that is obtained by scanning in a single direction (horizontal). The template image was obtained from the reference image by an affine geometric transformation.

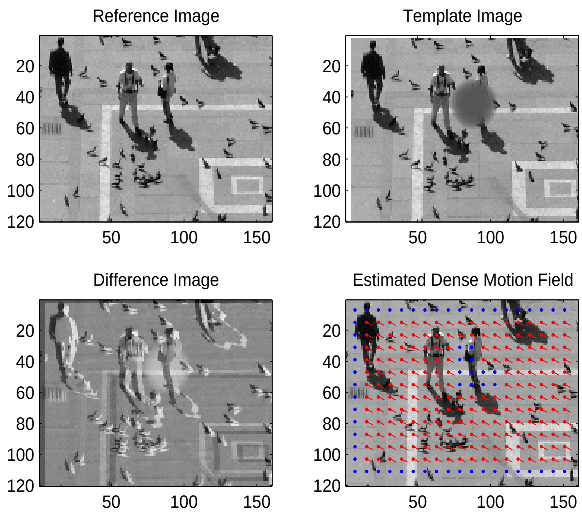

Figure 7: Example of the the estimated motion flow field in the presence of motion outliers. Blue points stand for points where algorithm cannot resolve any motion with certainty. The result was obtained by combining (using median operator) the motion estimates from 4 directions. The algorithm detected the outlier region (blue points in the center) and isolate it from the smooth motion field.

disappearing objects in the scene. One important advantage of the proposed method is its simplicity and relative low computational complexity.

The initial results demonstrate the usability of the algorithm, especially when targeting video filtering applications that are based on motion-compensated filtering such as video denoising, video stabilization and super-resolution processing. In future work, we may investigate different variations of the algorithm in order to enhance the basic motion tracking performance, and to derive simple rules for LMS step size adaptation. Also of interest is the research of different extensions of the algorithm in order to cover complex motion patterns.

REFERENCES

[1] T. Al-Naffouri, A. Sayed and T. Kailath, “On the selection of optimal nonlinearities for stochastic gradientadaptive algorithms,, IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP), vol. 1, pp. 464–467, September 2000.

[2] J. Astola, P. Haavisto and Y. Neuvo, “Vector Median Filters, Proceedings of the IEEE, vol. 78, no. 4, pp. 678–689, April 1990.

[3] M. Azimi-Sadjadi and P. Hongye “Two-dimensional block diagonal LMS adaptive filtering, IEEE Transactions on Signal Processing, vol. 42, no. 9, pp. 2420–2429, September 1994.

[4] S. Baker and I. Matthews, “Lukas-Kanade 20 years on: A unifying framework, International Journal of Computer Vision, vol. 56, no. 3, pp. 221–255, March 2004.

[5] S. Haykin, *Adaptive Filter Theory*, 3rd edition, Prentice Hall, 1996.

[6] J. Kim and T. Choi, “A Fast Full-Search Motion-Estimation Algorithm Using Representative Pixels and Adaptive Matching Scan, IEEE Transactions on Circuits and Systems for Video Technology, vol. 10, no. 7, pp. 1040–1048, October 2000.

[7] K. R. Namuduri, “Motion Estimation Using Spatio-Temporal Contextual Information, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No. 8, pp. 1111–1115, August 2004.

[8] N. Max “Visualizing Hilbert Curves, Proceedings of IEEE Visualization, pp. 447–450, October 1998.

[9] S. H. Seo and M. Azimi-Sadjadi and B. Tia, “A Least-Squares-Based 2-D Filtering Scheme for Stereo Image Compression, IEEE Transactions on Image Processing, vol. 9, no. 11, pp. 1967–1972, November 2000.

[10] T. Soni, B. D. Rao, J. R. Zeidler, and W. H. Ku, “Enhancement of images using the 2-D LMS adaptive algorithm, in Proc. IEEE Conference on Acoustics, Speech, Signal Processing, May 1991, pp. 3029-3032.

[11] C. Stiller and J. Konrad, “Estimating motion in image sequences, IEEE Signal Processing Magazine, vol. 16, no. 4, pp. 70–91, July 1999.

[12] M. Tico and M. Vehvilainen, “Constraint Motion Filtering for Video Stabilization, IEEE International Conference on Image Processing (ICIP), vol. 3, pp. 569–572, September 2005.

[13] M. Trimeche and J. Yrjanainen, “A method for Simultaneous Outlier Rejection in Image Super-Resolution Springer LNCS, vol. 2849, Visual Content Processing and Representation, pp. 188–195, 8th International Workshop on very low bitrate video coding (VLBV), September 2003.