HIERARCHICAL MOTION ESTIMATION USING RECURSIVE LMS FILTERS

Mejdi Trimeche [a], *Jani Lainema* [a] *and Moncef Gabbouj* [b]

[a] Interaction Core Technologies Center,
Nokia Research Center, Tampere, Finland

[b] Institute of Signal Processing,
Tampere University of Technology, Finland

email: mejdi.trimeche@nokia.com

ABSTRACT

In this paper, we present a hierarchical motion estimation algorithm that is based on adaptive LMS filters. The algorithm is an extension of an earlier work [1], which uses an adaptive 2-D LMS filter to match the intensity values while passing through the image pixels according to a Hilbert scanning pattern, the algorithm adapts the corresponding set of FIR coefficients. The peak value in the resulted coefficient distribution points to the localized displacement that happens between two consecutive frames. We extend the algorithm to use mirrored scanning and we apply it hierarchically across diadic spatial resolutions. The obtained displacement at each resolution level is mapped to the next level, which reduces the search area and improves the precision of the matching process. The algorithm is particularly useful for tracking slowly varying motion, such as affine or rotational motion even in the presence of noise. We also show an example application for motion compensated sharpening of video frames using the proposed LMS filtering without explicit computation of the displacement vectors.

1. INTRODUCTION

Since the problem of motion estimation can be stated as a system identification problem, adaptive filters constitute an attractive solution for tracking slowly varying and stationary displacements. Further, since adaptive filters do not make a-priori assumptions about the statistics of the signal data, these filters enable robust performance in the presence of various types of noise and outliers signals [2]. In particular, the LMS filter is attractive due to its simplicity and low computational complexity.

In the context of motion estimation, adaptive filters have been employed earlier in [3], [4], [5], [6], [7]. In [3], 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 [4], it was suggested 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 [5], an adaptive matching scan was employed to reduce the amount of computations needed to perform the full-search block-matching algorithm. In [6], correction for translational misregistration is used in a multi-frame restoration process, this was done by simultaneously estimating the output image and the non-symmetric point spread function. In [7], the local image registration is formulated as a two-dimensional (2-D) system identification problem with spatially varying system parameters, and the adapted filter coefficients are used to compensate for the effect of local distortions/displacements without explicitly estimating a displacement field. In [1], we proposed a method which uses the peak value in the resulted coefficient distribution to estimate the displacement field between the frames at each pixel position. The recursive LMS filtering produces smooth estimates of the displacements, directly at sub-pixel accuracy. In the following, we extend our formulation of the motion estimation as a localized tracking problem, and we use hierarchical decomposition to obtain precise displacement vectors at each pixel position.

2. OBSERVATION MODEL

Consider two consecutive video frames, 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, 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 modulated 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.

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

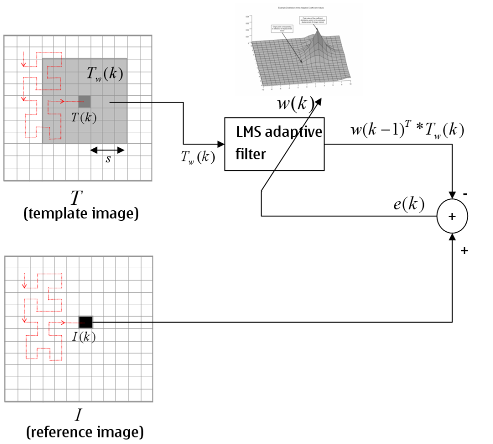

Fig. 1. Illustration of the LMS filtering that is used to calculate the displacement between the frames. The location of the peak value in the resulted coefficient distribution $w(k)$ points to the displacement between the two frames.

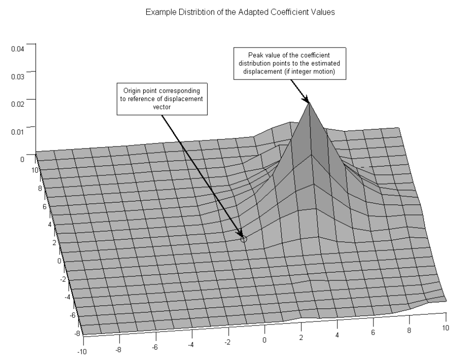

Fig. 2. Example distribution of adapted coefficient values. The peak value points to the displacement that happened between the two frames at pixel location k.

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. PIXEL MATCHING USING 2-D LMS FILTERS

Assuming small displacements within the search range s, we use 2-D LMS filtering to obtain the displacement by tracking the values of the adapted coefficients $w(k)$ as we scan from a pixel position (k) to the neighboring position $(k+1)$. Using a sliding window of pixel values $T_w(k)$ from the template image, the LMS filter adaptively modulates the coefficients of w to match the pixel $I(k)$ in the reference image. The peak value in the resulted coefficient distribution $w(k)$ points to the displacement between the two frames. To solve for $w(k)$, the standard LMS recursion is applied along a pre-determined scanning path (indexed by n) on the image grid, 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 according to the employed scanning pattern (see the following section for discussion). Fig. 1 shows an illustration that explains the basic filtering process, where the 2-D LMS filter is used to match the pixels through smooth modulation of the filter coefficient matrix. Fig. 2 shows a plot of the adapted coefficient values, which peak at the position corresponding to the displacement between the images.

Like its 1-D counterpart, the 2-D adaptive filter does not assume any knowledge of the cross correlation functions [8]. 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 step size $\mu(n)$ will result in slow convergence; whereas a large step size may cause instability. Alternatively, there are few modifications of the standard LMS algorithm that offer simpler stability requirements such as the normalized LMS (NLMS). The NLMS algorithm is obtained by substituting in Eq. (4) the following step size:

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

where ε is a small positive constant. In this form, the filter is also called $\varepsilon - NLMS$ [9], 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. If we use a small step size μ, the adaptation enables to precisely track smooth and slowly varying motion field, this may be suitable if we assume the motion is locally stationary. On the other hand, a larger step size will enable the tracking of fast changing motion but at the expense of reduced stability.

3.1. Space-filling patterns for scanning of the image grids

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 want to 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, even the simplest of scanning patterns, e.g. raster scan is sufficient to correctly estimate the stationary displacement. However if the frames undergo more complicated motion, we can use space-filling curves such as Peano or Hilbert curves [10] to traverse the image plane. The 2-D Hilbert curve sacans each grid point in a given quadrant (square of 2) only once and, continuously from quadrant to quadrant. 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. Usually, this scanning mode results in superior performance of the estimation process, especially in the presence of localized motion or other random outliers.

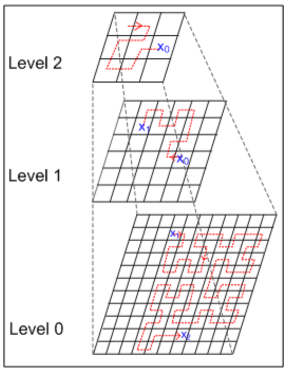

Fig. 3. Hierarchical processing paradigm. The estimated displacement is mapped between consecutive resolution levels to initialize the the starting points.

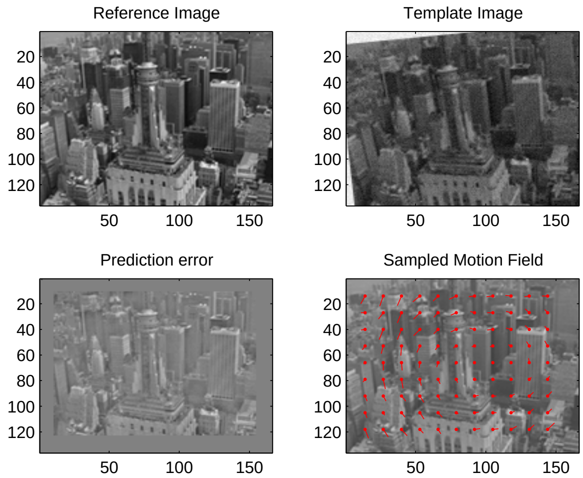

Fig. 4. Example of the estimated motion field. The template image (top right) is obtained by rotating and adding Gaussian noise ($\sigma^2 = 40$) to the original image. The sampled displacement field (bottom right) is obtained using the proposed technique

3.2. Multiple Scanning

By applying Hilbert scanning patterns in a mirrored fashion, it is possible to traverse each pixel from four different directions. This is very useful if we want to detect arbitrary and discontinuous motion, which may not be otherwise possible to detect using a single scanning direction. The scanning from different directions can be performed in separate filtering passes, which requires to temporally store the error images due to LMS adaptation from each direction. The final displacement field can be obtained by selecting the estimate that minimizes the corresponding error value at each pixel position. Another method to select the final displacement is to apply a component-wise scalar median filter, or the vector median [11] for the obtained displacement vectors from different scanning directions; this means that the final result is determined through a voting process, which additionally enhances the performance of the image registration against outliers.

3.3. Determining the displacement vectors

In order to explicitly determine 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 maximum coefficient value. 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. Additionally, we inserted a simple intermediate check to assert if the coefficient distribution can reliably point to a maximum in the coefficient distribution. This operation is described next:

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. HIERARCHICAL PROCESSING

The size of the window used in the adaptive filter $(2s + 1)$ defines the range of motion search between the two images. If we would like to detect larger displacements, we need to increase the window size. However, as we increase the window size, the speed of LMS convergence decreases, which negatively impacts the precision of the estimation process, additionally the computational cost also increases. This situation can be avoided if we use hierarchical decomposition of the images.

Similar to hierarchical block matching techniques, the original images are decomposed into lower resolution images by consecutive subsampling. We apply the proposed filtering from the lowest image resolutions, and the obtained displacement (at integer precision) is mapped to the next resolution level to initialize the position of the data window in the template image. Only at the last resolution, we calculate the displacement at subpixel accuracy. By using this technique, we can ensure that while using a small window for the coefficient adaptation, we can still precisely estimate large displacements. This process is illustrated in Fig. 3.

5. EXPERIMENTAL RESULTS

In this section, we briefly demonstrate the performance of the proposed approach for image registration, We also show an example application for motion compensated filtering.

In the first experiment (Fig. 4), we generated the template image by rotating an image from the video sequence "City". We further added Gaussian noise $(\sigma^2 = 30)$ to the resulting image. The test image is meant to simulate the effect that may be due to camera rotation while shooting a video in noisy conditions. We applied the hierarchical registration algorithm at two consecutive resolution levels. The following algorithm parameters were used $s = 8$ and $\mu = 0.15$. We used Hilbert scanning on blocks of 16×16, and we applied the scanning from multiple directions as described above. Note that the borders of the image were not filtered in this example, only the central part of the image is processed. The final motion vector was selected as a the component-wise median of the displacements obtained from the 4 different directions. The bottom left image in Fig. 4 shows the Prediction error due to the adaptive LMS matching, the image corresponds to $T(k) - \tilde{I}(k + D(k))$ in equation (2). The bottom right image in Fig. 4 shows the sampled motion field between the two images. Although the simulated motion is not trivial to correctly estimate using traditional block based methods, our algorithm was capable of smoothly tracking the rotation. In this example, we have not used any error rejection strategies (as mentioned in section 3.3), hence the displacement vectors were determined solely based on the peak in the coefficient distribution without imposing any other conditions. As it can be obderved from the result, the obtained displacement field is smooth and spatially correlated, which reflects better the real motion that happened between the images. The complexity of the algorithm is $O((2s+1)^2N)$. Comparing to the complexity of optical flow methods (table 1 in [12]), the proposed method is simpler and faster, thus enabling real-time implementations of motion compensated filtering.

The proposed registration technique can be also used in other applications involving motion compensated filtering without explicitly determining the displacement vectors. For example, for video enhancement. In Fig. 5, two consecutive frames from the City test sequence (frames 1 and 2) at QCIF resolution are combined together. The second image in (b) is sharpened by back-projecting the matching error due to adaptation process. Each pixel in the reference image is updated as $I(k) = I(k) + \alpha e(k)$, where $e(k)$ is calculated as in equation 4. For the particular example, we used $\alpha = 0.5$. Further inspection into the images reveals that the result image is sharper and contains more details than the original image. This is because the matching error between the two images reveals ”novelty” information from the template image that is not otherwise predictable from the reference image, especially by using linear filtering between the two images, or in another sense, the error image may be considered as a cross-frame gradient, which when projected back onto the original image creates high frequencies detail information that enhances the overall perception of the image sharpness. Together with improved outlier detection mechanism, this approach using adaptive filters can provide a suitable framework for implementing implicit and integrated motion compensated filtering, which can be very useful for video sharpening, video denoising and super-resolution.

(a) (b)

Fig. 5. Example application of the proposed algorithm for motion compensated filtering (a) Original image (b) Image sharpened by projecting the prediction error due to in the neighboring frame

6. CONCLUSIONS

In this paper, we presented a recursive method for pixel-based image registration. 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. Experiments have shown that applying the LMS filtering across disdic resolutions and along multiple scanning directions enables to smoothly track complicated motion such as rotations between the reference and template frames. The algorithm inherently produces smooth estimates of the displacements, directly at sub-pixel accuracy.

Additionally, the proposed filtering technique can be useful in several video filtering applications that are based on motion-compensated filtering such as video denoising, video sharpening and super-resolution. In this paper we have shown some initial results, which involve the use of the proposed filtering technique without explicit computation of the motion vectors for the sharpening of video frames using consecutive video frames with similar content. This integrated approach for video filtering is promising, and shall be extended in future work for other filtering tasks such as image super-resolution.

7. REFERENCES

[1] M. Trimeche, M. Tico, and M. Gabbouj, “Dense optical flow field estimation using recusrsive LMS filtering,” in *European Signal Processing Conference, Eusipco 2006, Florence*, September 2006.

[2] J. Proakis, C. Rader, F. Ling, M. Moonen, I. Proudler, and C. Nikias, *Algorithms for Statistical Signal Processing*, Prentice-Hall, 2002.

[3] S. H. Seo, 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.

[4] 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.

[5] 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.

[6] F. Sroubek and J. Flusser, “Multichannel blind deconvolution of spatially misaligned images,” *IEEE Transactions on Image Processing*, vol. 14, no. 7, pp. 874–883, July 2005.

[7] G. Caner, M. Tekalp, G. Sharma, and W. Heinzelman, “Local image registration by adaptive filtering,” *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 3053–3065, October 2006.

[8] T. Soni, B. Rao, J. Zeidler, and W. Ku, “Enhancement of images using the 2-d lms adaptive algorithm,” in *IEEE Conference on Acoustics, Speech and Signal Processing*, May 1991, vol. 4.

[9] 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*, vol. 1, pp. 464–467, September 2000.

[10] N. Max, “Visualizing hilbert curves,” in *Proceedings of IEEE Visualization*, October 1998, pp. 447–450.

[11] J. Astola, P. Haavisto, and Y. Neuvo, “Vector median filters,” *Proceedings of the IEEE*, vol. 78, no. 4, pp. 678–689, April 1990.

[12] 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.