Data-Driven Motion Estimation with Spatial Adaptation

Alessandra Martins Coelho *alessandra.coelho@ifsudestemg.edu.br*
Instituto Federal de Educacao, Ciencia e Tecnologia do Sudeste de Minas Gerais (IF SEMG),Rio Pomba, MG, Brazil

Vania Vieira Estrela *vestrela@id.uff.br*
Departamento de Telecomunicacoes, Universidade Federal Fluminense (UFF), Niteroi, RJ, Brazil

Abstract

Besides being an ill-posed problem, the pel-recursive computation of 2-D optical flow raises a wealth of issues, such as the treatment of outliers, motion discontinuities and occlusion. Our proposed approach deals with these issues within a common framework. It relies on the use of a data-driven technique called Generalized Cross Validation (GCV) to estimate the best regularization scheme for a given moving pixel. In our model, a regularization matrix carries information about different sources of error in its entries and motion vector estimation takes into consideration local image properties following a spatially adaptive. Preliminary experiments indicate that this approach provides robust estimates of the optical flow.

Keywords: Motion Estimation, Generalized Cross Validation, Video Processing, Computer Vision, Regularization.

1. INTRODUCTION

Motion estimation is very important in multimedia video processing applications. For example, in video coding, the estimated motion is used to reduce the transmission bandwidth. The evolution of an image sequence motion field can also help other image processing tasks in multimedia applications such as analysis, recognition, tracking, restoration, collision avoidance and segmentation of objects [6, 7, 10].

In coding applications, a block-based approach [7] is often used for interpolation of lost information between key frames. The fixed rectangular partitioning of the image used by some block-based approaches often separates visually meaningful image features. If the components of an important feature are assigned different motion vectors, then the interpolated image will suffer from annoying artifacts. Pel-recursive schemes [2,3,6] can theoretically overcome some of the limitations associated with blocks by assigning a unique motion vector to each pixel. Intermediate frames are then constructed by resampling the image at locations determined by linear interpolation of the motion vectors. The pel-recursive approach can also manage motion with subpixel accuracy. The update of a motion estimate is based on the minimization of the displaced frame difference (DFD) at a pixel. In the absence of additional assumptions about the pixel motion, this estimation problem becomes “ill-posed” because of the following problems: a) occlusion; b) the solution to the 2-D motion estimation problem is not unique (aperture problem); and c) the solution does not continuously depend on the data due to the fact that motion estimation is highly sensitive to the presence of observation noise in video images.

We propose to solve optical flow (OF) problems by means of a framework that combines the Generalized Cross Validation (GCV) and a regularization matrix $\boldsymbol{\Lambda}$. Such approach accounts better for the statistical properties of the errors present in the scenes than the solution proposed by Biemond [1] where a scalar regularization parameter was used.

We organized this work as follows. Section 2 provides some necessary background on the pel-recursive motion estimation problem. Section 3 introduces our spatially adaptive approach. Section 4 describes the Ordinary Cross Validation. Section 5 deals with the GCV technique. Section 6 defines the metrics used to evaluate our results. Section 7 describes the experiments used to access the performance of our proposed algorithm. Finally, Section 8 presents some conclusions.

2. PEL-RECURSIVE DISPLACEMENT ESTIMATION

2.1. Problem Characterization

The displacement of a picture element (pel) between adjacent frames forms the displacement vector field (DVF) and its estimation can be done using at least two successive frames. The DVF is the 2-D motion resulting from the apparent motion of the image brightness (OF) where a displacement vector (DV) is assigned to each image pixel.

A pixel belongs to a moving area if its intensity has changed between consecutive frames. Hence, our goal is to find the corresponding intensity value $I_k(\boldsymbol{r})$ of the k-th frame at location $\boldsymbol{r} = [x, y]^T$, and $\boldsymbol{d}(\boldsymbol{r}) = [d_x, d_y]^T$ the corresponding (true) DV at the working point $\boldsymbol{r}$ in the current frame. Pel-recursive algorithms minimize the DFD function in a small area containing the working point assuming constant image intensity along the motion trajectory. The DFD is defined by

$$\Delta(\boldsymbol{r};\boldsymbol{d}(\boldsymbol{r})) = I_k(\boldsymbol{r}) - I_{k-1}(\boldsymbol{r}-\boldsymbol{d}(\boldsymbol{r})) \tag{1}$$

and the perfect registration of frames will result in $I_k(\boldsymbol{r}) = I_{k-1}(\boldsymbol{r}-\boldsymbol{d}(\boldsymbol{r}))$. The DFD represents the error due to the nonlinear temporal prediction of the intensity field through the DV. The relationship between the DVF and the intensity field is nonlinear. An estimate of $\boldsymbol{d}(\boldsymbol{r})$, is obtained by directly minimizing $\Delta(\boldsymbol{r},\boldsymbol{d}(\boldsymbol{r}))$ or by determining a linear relationship between these two variables through some model. This is accomplished by using the Taylor series expansion of $I_{k-1}(\boldsymbol{r}-\boldsymbol{d}(\boldsymbol{r}))$ about location $(\boldsymbol{r}-\boldsymbol{d}^{\mathrm{i}}(\boldsymbol{r}))$, where $\boldsymbol{d}^{\mathrm{i}}(\boldsymbol{r})$ represents a prediction of $\boldsymbol{d}(\boldsymbol{r})$ in i-th step. This results in

$$\Delta(\boldsymbol{r},\boldsymbol{r}-\boldsymbol{d}^{\mathrm{i}}(\boldsymbol{r})) = -\boldsymbol{u}^T \nabla I_{k-1}(\boldsymbol{r}-\boldsymbol{d}^{\mathrm{i}}(\boldsymbol{r})) + e(\boldsymbol{r},\boldsymbol{d}(\boldsymbol{r})), \tag{2}$$

where the displacement update vector $\boldsymbol{u} = [u_x, u_y]^T = \boldsymbol{d}(\boldsymbol{r}) - \boldsymbol{d}^{\mathrm{i}}(\boldsymbol{r})$, $e(\boldsymbol{r},\boldsymbol{d}(\boldsymbol{r}))$ represents the error resulting from the truncation of the higher order terms (linearization error) and $\nabla = [\partial/\partial_x, \partial/\partial_y]^T$ represents the spatial gradient operator. Applying Eq. (2) to all points in a neighborhood R containing *N* pixels gives

$$z = \boldsymbol{G}\boldsymbol{u} + \boldsymbol{n}, \tag{3}$$

where the temporal gradients $\Delta(\boldsymbol{r},\boldsymbol{r}-\boldsymbol{d}^{\mathrm{i}}(\boldsymbol{r}))$ have been stacked to form the $N \times 1$ observation vector z containing DFD information on all the pixels in R, the $N \times 2$ matrix $\boldsymbol{G}$ is obtained by stacking the spatial gradient operators at each observation, and the error terms have formed the $N \times 1$ noise vector $\boldsymbol{n}$ which is assumed Gaussian with $\boldsymbol{n} \sim \mathrm{N}(0,\sigma_n^2 \boldsymbol{I})$. Each row of $\boldsymbol{G}$ has entries

$\left[g_{xi}, g_{yi}\right]^T$, with $i = 1, ..., N$. The spatial gradients of I_{k-1} are calculated through a bilinear interpolation scheme [2].

2.2. Regularized Least-Squares Estimation

The pel-recursive estimator for each pixel located at position $\boldsymbol{r}$ of a frame can be written as

$$\boldsymbol{d}^{i+1}(\boldsymbol{r}) = \boldsymbol{d}^{i}(\boldsymbol{r}) + \boldsymbol{u}^{i}(\boldsymbol{r}) \tag{4}$$

where $\boldsymbol{u}^{i}(\boldsymbol{r})$ is the current motion update vector obtained through a motion estimation procedure that attempts to solve Eq. (3), $\boldsymbol{d}^{i}(\boldsymbol{r})$ is the DV at iteration i and $\boldsymbol{d}^{i+1}(\boldsymbol{r})$ is the corrected DV. The regularized minimum norm solution to the previous expression, that is

$$\hat{\boldsymbol{u}}(\boldsymbol{\Lambda}) = \hat{\boldsymbol{u}}_{RLS}(\boldsymbol{\Lambda}) = (\boldsymbol{G}^T\boldsymbol{G} + \boldsymbol{\Lambda})^{-1}\boldsymbol{G}^T z \tag{5}$$

is also known as Regularized Least-Squares (RLS) solution. In order to improve the RLS estimate of the motion update vector, we propose a strategy which takes into consideration the local properties of the image. It is described in the next section.

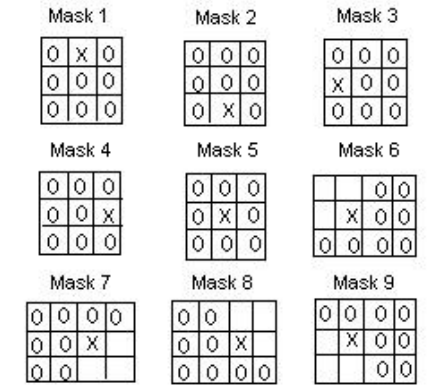

FIGURE 1: Neighborhood geometries.

3. SPATIALLY ADAPTIVE NEIGHBORHOODS

Aiming to improve the estimates given by the pel-recursive algorithm, we introduced an adaptive scheme for determining the optimal shape of the neighborhood of pixels with the same DV used to generate the overdetermined system of equations given by Eq. (3). More specifically, the masks in Fig. 1 show the geometries of the neighborhoods used.

Errors can be caused by the basic underlying assumption of uniform motion inside R (the smoothness constraint), by not grouping pixels adequately, and by the way gradient vectors are estimated, among other things. It is known that in a noiseless image containing pixels in textured areas most errors, when estimating motion occur close to motion boundaries. This information leads to a hypothesis testing (HT) approach to determine the most appropriate neighborhood shape for a given pixel. The best neighborhood from the finite set of templates shown in Fig. 1, according to the smallest $|DFD|$ criterion, in an attempt to adapt the model to local features associated to motion boundaries.

4. ORDINARY CROSS VALIDATION (OCV)

Cross validation has been proven to be a very effective method of estimating the regularization parameters [4,5,8], which in our work are the entries of $\boldsymbol{\Lambda}$, without any prior knowledge on the noise statistics. The degree of smoothing of the solution $\hat{\mathbf{u}}(\boldsymbol{\Lambda})$, in Eq. (5), is dictated by the regularization matrix $\boldsymbol{\Lambda}$.

OCV divides the data into two disjoint subsets obtained from the original observations: an estimation/prediction set and a validation set. In the context of neural networks, the former is called "training set". Let M be the number of observations used for the validation set, where $M \geq 1$, and N be the total number of observations. In our particular problem, N is the mask size and z is the entire observation set, that is an N-dimensional measurement vector. For each set $j = 1, ..., M$, where j is the size of the validation set, the minimum mean-square error (MSE) is calculated using the left out data set, that is, the remaining $(N - j)$ observations, and varying the regularization matrix $\boldsymbol{\Lambda}$. In other words, for each value of j, a corresponding set with $(N - j)$ elements is used to predict the j data points left out.

OCV is defined as the average of all the MSE's evaluated over all possible $\binom{N}{j}$ combinations of validation sets. For the case $j = 1$, that is the validation set has only one element, the OCV or prediction MSE is given by

$$OCV(\boldsymbol{\Lambda}) = \frac{1}{N}\sum_{i=0}^{N-1}\left[z_i - \hat{\mu}_i^{-} \right]^2 ,$$

where z_i is the i-th entry of the observation vector z, $i = 0, ..., (N-1)$ as follows: $z = \begin{bmatrix} z_0 \\ \mathrm{M} \\ z_i \\ \mathrm{M} \\ z_{N-1} \end{bmatrix}$,

z_{-i} the vector obtained after making the i-th entry of z equal to zero, that is $z_{-i} = \begin{bmatrix} z_0 \\ \mathrm{M} \\ z_{i-1} \\ 0 \\ z_{i+1} \\ \mathrm{M} \\ z_{N-1} \end{bmatrix}$, and

$\hat{\mu}_i^{-} = [\boldsymbol{G}(\boldsymbol{G}^T\boldsymbol{G} + \boldsymbol{\Lambda})^{-1}\boldsymbol{G}^T z_{-i}]_i$ is the estimate of point z_i using vector z_{-i}.

The previous OCV equation averages all the MSE's obtained by leaving each of the entries of z out. Therefore, the data division into validation and estimation/prediction sets is done in an alternate fashion. All the data is used for both purposes. This technique is also called predictive sample reuse or leave-one-out principle. The idea behind OCV is to perform a data-driven consistency check that, essentially, measures the adequacy of a parameter set via the model ability to predict some of the observations based on the other ones. Hence, the optimum value of the regularization parameters for N samples is the one that minimizes the mean-square error $OCV(\boldsymbol{\Lambda})$. The previous expression has to be further manipulated in order to express the OCV in terms of $\boldsymbol{\Lambda}$. The optimum $\boldsymbol{\Lambda}$, that is $\hat{\boldsymbol{\Lambda}}$, is the one that minimizes the following function:

$$OCV(\boldsymbol{\Lambda}) = \frac{1}{N}\left\|\boldsymbol{H}(\boldsymbol{\Lambda})\{\boldsymbol{I} - \boldsymbol{A}(\boldsymbol{\Lambda})\}z\right\|^2,$$

where $$\boldsymbol{H}(\boldsymbol{\Lambda}) = diag\left\{\frac{1}{1-\{\boldsymbol{g}_0{}^T\boldsymbol{B}^{-1}\boldsymbol{g}_0\}}, \mathrm{L}\ , \frac{1}{1-\{\boldsymbol{g}_{N-1}{}^T\boldsymbol{B}^{-1}\boldsymbol{g}_{N-1}\}}\right\},$$

$\boldsymbol{A}(\boldsymbol{\Lambda}) = \boldsymbol{G}[\boldsymbol{G}^T\boldsymbol{G} + \boldsymbol{\Lambda}]^{-1}\boldsymbol{G}^T$, and $\boldsymbol{B} = \boldsymbol{B}(\boldsymbol{\Lambda}) = (\boldsymbol{G}^T\boldsymbol{G} + \boldsymbol{\Lambda})$.

The main advantage of the OCV is its systematic way of determining the regularization parameter directly from the observed data. However, it presents the following drawbacks:

(i) It uses a noisy performance measure, Mean Squared Error (MSE). This means that since we are looking at the average value of the MSE over several observation sets re-sampled from the original z, we can only guarantee the OCV estimator of $\boldsymbol{\Lambda}$ is going to be a good predictor when $N >> 1$.

(ii) It treats all data sets equally. In terms of image processing, we expect close neighbors of the current pixel to behave more similarly to it (in most of the cases) than pixels that are more distant from it. Of course, this is not the case with motion boundaries, occlusion and transparency.

5. THE GENERALIZED CROSS VALIDATION (GCV)

The OCV does not provide good estimates of $\boldsymbol{\Lambda}$ [5, 8]. A modified method called GCV function gives more satisfactory results. GCV is a weighted version of the OCV, and it is given by

$$GCV(\boldsymbol{\Lambda}) = \frac{1}{N}\sum_{i=1}^{N}[z_i - \hat{z}_i]^2 w_i(\boldsymbol{\Lambda}), \tag{6}$$

where the weights w_i are defined as follows:

$$w_i(\boldsymbol{\Lambda}) = \left\{\frac{[1 - a_{ii}(\boldsymbol{\Lambda})]}{\left(1 - \frac{1}{N}Tr[\boldsymbol{A}(\boldsymbol{\Lambda})]\right)}\right\}^2, \text{ and} \tag{7a}$$

$$\boldsymbol{A}(\boldsymbol{\Lambda}) = \boldsymbol{G}\left(\boldsymbol{G}^T\boldsymbol{G} + \boldsymbol{\Lambda}\right)^1\boldsymbol{G}^T \tag{7b}$$

with $a_{ii}(\boldsymbol{\Lambda})$ being the diagonal entries of matrix $\boldsymbol{A}(\boldsymbol{\Lambda})$ as defined in Eq. (7b). The main shortcoming of OCV is the fact that OCV is not invariant to orthonormal transformations. In other words, if data $z' = \boldsymbol{\Gamma}z$ is available, where $\boldsymbol{\Gamma}$ is an $N \times N$ orthonormal matrix, and z' is the observation vector corresponding to the linear model given by

$$z' = \boldsymbol{G}'\boldsymbol{u}' + \boldsymbol{n}' = \boldsymbol{\Gamma}\{\boldsymbol{G}\boldsymbol{u} + \boldsymbol{n}\}. \tag{8}$$

Therefore, the *OCV*, and, consequently the regularization matrix $\boldsymbol{\Lambda}$, depends on $\boldsymbol{\Gamma}$. *GCV* on the other hand is independent of $\boldsymbol{\Gamma}$. Thus, the *GCV*($\boldsymbol{\Lambda}$) is a better criterion for estimating the regularization parameters [5, 8]. So, Eq. (6) can also take the form

$$\mathrm{GCV}(\boldsymbol{\Lambda}) = \frac{1}{N} \frac{\left\| [\mathbf{I} - \mathbf{A}(\boldsymbol{\Lambda})]\mathbf{z} \right\|^2}{\left[\frac{1}{N} Tr\{\mathbf{I} - \mathbf{A}(\boldsymbol{\Lambda})\} \right]^2}. \tag{9}$$

5.1 Regularization Matrix Determination

The *GCV* function for the observation model in Eq. (3) is given in closed form by Eq. (9). Let us call $\hat{\boldsymbol{u}}_{GCV}$ the solution for Eq. (3) when an optimum parameter set (the entries of the regularization matrix) $\boldsymbol{\Lambda}_{GCV}$ is found by means of the GCV. Then, Eq. (5) becomes

$$\hat{\boldsymbol{u}}_{\mathrm{GCV}} = \left(\boldsymbol{G}^T \boldsymbol{G} + \boldsymbol{\Lambda}_{GCV} \right)^{-1} \boldsymbol{G}^T \boldsymbol{z} \tag{10}$$

5.2 The GCV-based Estimation Algorithm

For each pixel located at $\boldsymbol{r} = (x, y)$ the GCV-based algorithm is given by the following steps:

1) Initialize the system: $\boldsymbol{d}^{\circ}(\boldsymbol{r})$, $m \leftarrow 0$ (m = mask counter), and $i \leftarrow 0$ (i = iteration counter).
2) If $|DFD| < T$, then stop. T is a threshold for $|DFD|$.
3) Calculate $\boldsymbol{G}^i$ and z^i for the current mask and current initial estimate.
4) Calculate $\boldsymbol{\Lambda}^i$ by minimizing the expression

$$GCV(\boldsymbol{\Lambda}^i) = \frac{1}{N} \frac{\left\| \left[\boldsymbol{I} - \boldsymbol{A}(\boldsymbol{\Lambda}^i) \right] z^i \right\|^2}{\left[\frac{1}{N} Tr\left\{ \boldsymbol{I} - \boldsymbol{A}(\boldsymbol{\Lambda}^i) \right\} \right]^2}, \tag{11}$$

where

$$\boldsymbol{A}(\boldsymbol{\Lambda}^i) = \boldsymbol{G}^i \left[\left(\boldsymbol{G}^i \right)^T \boldsymbol{G}^i + \boldsymbol{\Lambda}^i \right]^{-1} \left(\boldsymbol{G}^i \right)^T. \tag{12}$$

5) Calculate the current update vector:

$$\boldsymbol{u}^i = \left[\left(\boldsymbol{G}^i \right)^T \boldsymbol{G}^i + \boldsymbol{\Lambda}^i \right]^{-1} \left(\boldsymbol{G}^i \right)^T z^i. \tag{13}$$

6) Calculate the new DV:

$$\boldsymbol{d}^{\mathrm{i}+1}(\boldsymbol{r}) = \boldsymbol{d}^{\mathrm{i}}(\boldsymbol{r}) + \boldsymbol{u}^{\mathrm{i}}(\boldsymbol{r}) \tag{14}$$

7) For the current mask m:

   If $\left\| \boldsymbol{d}^{\mathrm{i}+1}(\boldsymbol{r}) - \boldsymbol{d}^{\mathrm{i}}(\boldsymbol{r}) \right\| \leq \varepsilon$ and $|DFD| < T$, then stop.

   If $i < (I-1)$ where I is the maximum number of iterations allowed, then go to step 3 with $i \leftarrow i+1$ and use $\boldsymbol{d}^{\mathrm{i}}(\boldsymbol{r}) \leftarrow \boldsymbol{d}^{\mathrm{i}+1}(\boldsymbol{r})$ as the new initial estimate.

Otherwise, try another mask: $m \leftarrow m+1$. If all masks where used and no DV was found, then set $\boldsymbol{d}^{i+1}(\boldsymbol{r}) = 0$.

(a) (b) (c)

FIGURE 2: Frames from the video sequences used for tests: (a) synthetic frame, (b) Mother and Daughter and (c) Foreman

6. METRICS TO EVALUATE THE EXPERIMENTS

The motion field quality is accessed using the four metrics [2, 3] described below applied to the video sequences shown in Fig. 2.

6.1 Mean Squared Error (MSE)

Since the MSE provides an indication of the degree of correspondence between the estimates and the true value of the motion vectors, we can apply this measure to two consecutive frames of a sequence with known motion. We can evaluate the MSE in the horizontal (MSE_x) and in the vertical (MSE_y) directions as follows

$$MSE_x = \frac{1}{RC}\sum_{\boldsymbol{r}\in \boldsymbol{S}}[d_x(\boldsymbol{r}) - \hat{d}_x(\boldsymbol{r})]^2 \text{, and} \tag{15}$$

$$MSE_y = \frac{1}{RC}\sum_{\boldsymbol{r}\in \boldsymbol{S}}[d_y(\boldsymbol{r}) - \hat{d}_y(\boldsymbol{r})]^2 , \tag{16}$$

where $\boldsymbol{S}$ is the entire frame, $\boldsymbol{r}$ represents the pixel coordinates, $\boldsymbol{R}$ and $\boldsymbol{C}$ are, respectively, the number of rows and columns in a frame, $\boldsymbol{d}(\boldsymbol{r}) = \left(\boldsymbol{d}_x(\boldsymbol{r}), \boldsymbol{d}_y(\boldsymbol{r})\right)$ is the true DV at $\boldsymbol{r}$, and $\hat{\boldsymbol{d}}(\boldsymbol{r}) = \left(\hat{d}_x(\boldsymbol{r}), \hat{d}_y(\boldsymbol{r})\right)$ its estimation.

6.2 Bias

The bias gives an idea of the degree of correspondence between the estimated motion field and the original optical flow. It is defined as the average of the difference between the true DV's and their predictions, for all pixels inside a frame S, and it is defined along the x and y directions as

$$bias_x = \frac{1}{RC}\sum_{\boldsymbol{r}\in \boldsymbol{S}}[d_x(\boldsymbol{r}) - \hat{d}_x(\boldsymbol{r})] \tag{17}$$

and

$$bias_y = \frac{1}{RC}\sum_{\boldsymbol{r}\in \boldsymbol{S}}[d_y(\boldsymbol{r}) - \hat{d}_y(\boldsymbol{r})] . \tag{18}$$

Ean-squared Displaced Frame Difference

This metric evaluates the behavior of the average of the squared displaced frame difference ($\overline{DFD}^2$). It represents an assessment of the evolution of the temporal gradient as the scene evolves by looking at the squared difference between the current intensity $I_k(\boldsymbol{r})$ and its predicted value $I_{k-1}(\boldsymbol{r}-\boldsymbol{d}(\boldsymbol{r}))$. Ideally, the $\overline{DFD}^2$ should be zero, which means that all motion was identified correctly ($I_k(\boldsymbol{r}) = I_{k-1}(\boldsymbol{r}-\boldsymbol{d}(\boldsymbol{r}))$ for all $\boldsymbol{r}$'s). In practice, we want the $\overline{DFD}^2$ to be as low as possible. Its is defined as

$$\overline{DFD}^2 = \frac{\sum_{k=2}^{K}\sum_{\boldsymbol{r}\in S}[I_k(\boldsymbol{r}) - I_{k-1}(\boldsymbol{r}-\boldsymbol{d}(\boldsymbol{r}))]^2}{RC(K-1)}, \tag{19}$$

where K is the length of the image sequence.

6.3 Improvement in Motion Compensation

The average improvement in motion compensation $\overline{IMC}(dB)$ between two consecutive frames is given by

$$\overline{IMC}_k(dB) = 10\log_{10}\left\{\frac{\sum_{\boldsymbol{r}\in S}[I_k(\boldsymbol{r}) - I_{k-1}(\boldsymbol{r})]^2}{\sum_{\boldsymbol{r}\in S}[I_k(\boldsymbol{r}) - I_{k-1}(\boldsymbol{r}-\boldsymbol{d}(\boldsymbol{r}))]^2}\right\}, \tag{20}$$

where S is the frame being currently analyzed. It shows the ratio in decibel (dB) between the mean-squared frame difference ($\overline{FD}^2$) defined by

$$\overline{FD}^2 = \frac{\sum_{\boldsymbol{r}\in S}[I_k(\boldsymbol{r}) - I_{k-1}(\boldsymbol{r})]^2}{RC}, \tag{21}$$

and the $\overline{DFD}^2$ between frames k and $(k-1)$.

As far as the use of the this metric goes, we chose to apply it to a sequence of K frames, resulting in the following equation for the average improvement in motion compensation:

$$\overline{IMC}(dB) = 10\log_{10}\left\{\frac{\sum_{k=2}^{K}\sum_{\boldsymbol{r}\in S}\left[I_k(\boldsymbol{r}) - I_{k-1}(\boldsymbol{r})\right]^2}{\sum_{k=2}^{K}\sum_{\boldsymbol{r}\in S}\left[I_k(\boldsymbol{r}) - I_{k-1}\left(\boldsymbol{r}-\boldsymbol{d}(\boldsymbol{r})\right)\right]^2}\right\}. \tag{22}$$

When it comes to motion estimation, we seek algorithms that have high values of $\overline{IMC}(dB)$. If we could detect motion without any error, then the denominator of the previous expression would be zero (perfect registration of motion) and we would have $\overline{IMC}(dB) = \infty$.

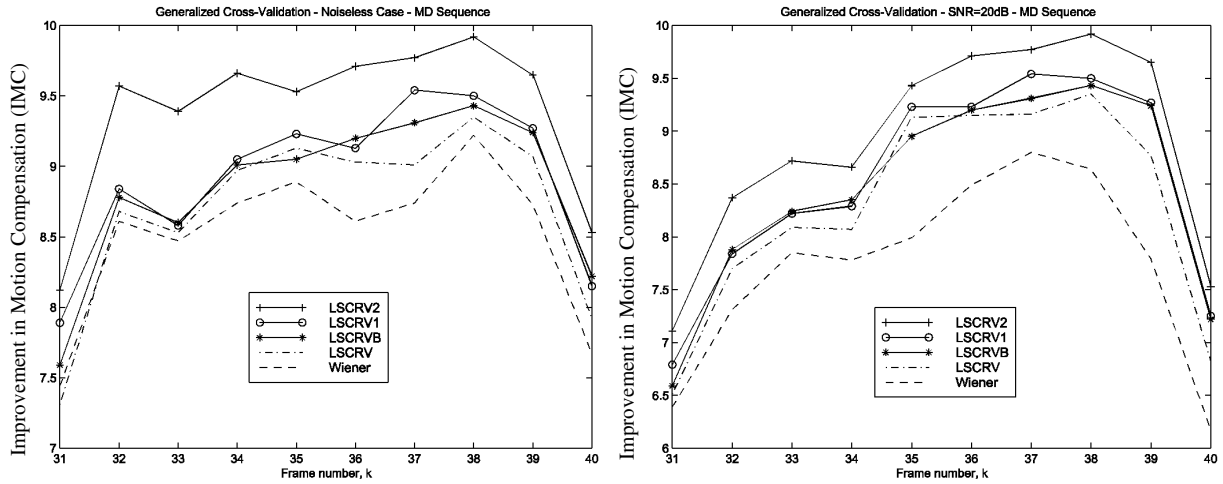

FIGURE 3: $\overline{IMC}(dB)$ for the noiseless (left) and noisy (right) cases for the MD sequence.

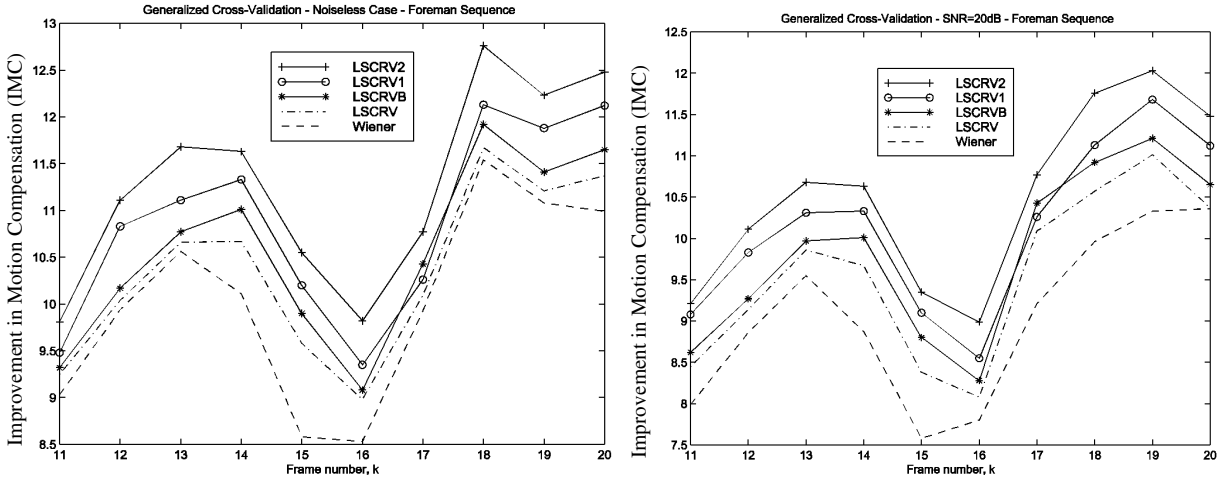

FIGURE 4: $\overline{IMC}(dB)$ for frames 11-20 of the noiseless (left) and noisy with $SNR = 20dB$ (right) for the Foreman sequence.

7. IMPLEMENTATION

In this section, we present several experimental results that illustrate the effectiveness of the GCV approach and compare it with the Wiener filter [2, 3 ,7] similar to the one in [1] given by

$$\hat{\boldsymbol{u}}_{Wiener} = \hat{\boldsymbol{u}}_{LMMSE} = \left(\boldsymbol{G}^T\boldsymbol{G} + \mu\boldsymbol{I}\right)^{-1}\boldsymbol{G}^T\boldsymbol{z}\,, \tag{23}$$

where μ=*50* was chosen for all pixels in an entire frame. All sequences are *144×176*, *8*-bit (QCIF format). The algorithms were applied to three image sequences: one synthetically generated, with known motion; the "Mother and Daughter" (MD) and the "Foreman". For each sequence, two sets of experiments are analyzed: one for the noiseless case and the other for a sequence whose frames are corrupted by a signal-to-noise-ratio (SNR) equal to 20 dB. The SNR is defined as

$$SNR = 10\log_{10}\frac{\sigma^2}{\sigma_c^2}. \quad (24)$$

where σ^2 is the variance of the original image and σ_c^2 is the variance of the noise corrupted image [8].

	Wiener	LSCRV	LSCRVB	LSCRV1	LSCRV2
MSE_x	0.1548	0.1534	0.1511	0.1493	0.1440
MSE_y	0.0740	0.0751	0.0753	0.0754	0.0754
$bias_x$	0.0610	0.0619	0.0599	0.0581	0.0574
$bias_y$	-0.0294	-0.0291	-0.0294	-0.0294	-0.0293
$\overline{IMC}(dB)$	19.46	19.62	19.74	19.89	20.38
$\overline{DFD}^2$	4.16	4.05	3.921	3.76	3.35

TABLE 1: Comparison between GCV implementations and the Wiener filter. $SNR = \infty$.

	Wiener	LSCRV	LSCRVB	LSCRV1	LSCRV2
MSE_x	0.2563	0.2544	0.2446	0.2437	0.2373
MSE_y	0.1273	0.1270	0.1268	0.1257	0.1254
$bias_x$	0.0908	0.0889	0.0883	0.0881	0.0852
$bias_y$	-0.0560	-0.0565	-0.0564	-0.0561	-0.0553
$\overline{IMC}(dB)$	14.74	14.83	14.98	15.15	15.32
$\overline{DFD}^2$	12.24	12.02	11.60	11.16	10.78

TABLE 2: Comparison between GCV implementations and the Wiener filter. $SNR = 20dB$.

7.1 Programs and Experiments

The following GCV-based programs were developed:

a) LSCRV: λ is a scalar; a non-causal *3×3* mask, centered at the pixel being analyzed.
b) LSCRVB: λ is a scalar; we tried all nine masks.
c) LSCRV1: $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}_{GCV} = diag(\lambda_1, \lambda_2)$, where λ_1 and λ_2 are scalars, is a matrix; a non-causal *3×3* mask centered at the pixel being analyzed.
d) LSCRV2: $\boldsymbol{\Lambda}$ is a matrix; we tried all nine masks.

Results from the proposed algorithms are compared to the ones obtained with the Wiener (LMMSE) filter from Eq. (23) in the subsequent experiments.

Experiment 1. In this sequence, there is a moving rectangle immersed in a moving background. In order to create textures for the rectangle and its background (otherwise motion detection would not be possible), the following auto-regressive model was used:

$$I(m,n) = \tfrac{1}{3}\left[I(m,n-1)+I(m-1,n)+I(m-1,n-1)\right] + n_i(m,n), \quad (25)$$

where $i = 1, 2$. For the background $(i = 1)$, n_1 is a Gaussian random variable with mean $\mu_1 = 50$ and variance $\sigma_1^2 = 49$. The rectangle $(i = 2)$ was generated with $\mu_2 = 100$ and variance $\sigma_2^2 = 25$. All pixels from the background move to the right, and the displacement from frame *1* to frame *2* is

$\boldsymbol{d}_b(\boldsymbol{r}) = \left(\boldsymbol{d}_{bx}(\boldsymbol{r}), \boldsymbol{d}_{by}(\boldsymbol{r})\right) = (2,0)$. The rectangle moves in a diagonal fashion from frame *1* to *2* with $\boldsymbol{d}_r(\boldsymbol{r}) = \left(\boldsymbol{d}_{rx}(\boldsymbol{r}), \boldsymbol{d}_{ry}(\boldsymbol{r})\right) = (1,2)$.

Table 1 shows the values for the *MSE*, bias, $\overline{IMC}$ (dB) and $\overline{DFD}^2$ for the estimated optical flow using the Wiener filter and the four programs mentioned previously when no noise is present.
All the algorithms employing the *GCV* show improvement in terms of the metrics used. When we compare LSCRV with the Wiener filter, we see that with a regularization matrix of the form $\boldsymbol{\Lambda} = \lambda \boldsymbol{I}$, whose regularization parameter λ is determined by means of the minimization of the GCV function and using the same 3×3 mask as the Wiener, the improvements are small (we discuss some of our findings about the drawbacks of the GCV at the end of this article). The performance of the GCV implementation increases with the spatially adaptive approach a scalar regularization parameter λ (algorithm LSCRVB) when compared to the OLS. Now, when we compare the performance of the previous algorithms with the case where we have a more complex regularization matrix $\boldsymbol{\Lambda} = diag\{\lambda_1, \lambda_2\}$, that is, the implementation LSCRV1, then get even more improvements, although we have a single mask. Finally, using both the spatially adaptive approach and $\boldsymbol{\Lambda} = diag\{\lambda_1, \lambda_2\}$ we get the best results (the $\overline{IMC}(dB)$ goes up almost *1 dB* on the average).

Table 2 shows the values for the MSE, bias, $\overline{IMC}(dB)$ and $\overline{DFD}^2$ for the estimated optical flow using the Wiener filter and the four programs mentioned previously with two noisy frames ($SNR = 20dB$).

The results for both the noiseless and noisy cases present better values of $\overline{IMC}(dB)$ and $\overline{DFD}^2$ as well as MSE's and biases for all algorithms using the GCV. The best results in terms of metrics and visually speaking are obtained with the LSCRV2 algorithm ($\boldsymbol{\Lambda} = diag\{\lambda_1, \lambda_2\}$ and multi-mask strategy). For the noisy case, it should be pointed out the considerable reduction of the interference of noise when it comes to the motion in the background and inside the object. For this algorithm, even the motion around the borders of the rectangle is clearer than when the LMMSE estimator is used.

Experiment 2. Fig. 4 presents the values of the improvement in motion compensation for frames *31* to *40* of the MD sequence for the noiseless and noisy ($SNR = 20dB$) cases, respectively, for all algorithms investigated. Here we concentrate our analysis on the performance of LSCRV2, which is the algorithm that gave us the best results. The LSCRV2 algorithm provides, on the average, $1.5dB$ higher $\overline{IMC}(dB)$ than the LMMSE algorithm for the noiseless case. The $\overline{IMC}(dB)$ for the noisy case is not as high as in the previous situation. Their qualitative performance can be observed in Fig. 3. By visual inspection, the noiseless case does present dramatic differences between both motion fields. For the noisy case, we were able of capturing the motion relative to the rotation of the mother's head, although incorrect displacement vectors were found in regions were there is no texture at all such as the background, for instance, but there is less noise than when we use the Wiener filter.

Experiment 3. Fig. 4 demonstrates results obtained for frames *11-20* of the "Foreman" sequence. Some frames of this sequence show abrupt motion changes. One can see that all the algorithms based on GCV outperform the LMMSE. This sequence shown very good values for the $\overline{IMC}(dB)$ for both the noiseless and the noisy cases. As one can see by looking at the plots for the errors in the motion compensated frames, the algorithm LSCRV2 performs better than the Wiener filter visually speaking.

8. CONCLUSION AND DISCUSSIONS

This work addresses some issues related to the application of two adaptive pel-recursive techniques to solve the problem of estimating the DVF. We analyzed the issue of robust estimation of the DVF between two consecutive frames, concentrating our attention on the effect of noise on the estimates. The observation z is subjected to independent identically distributed (i.i.d.) zero-mean additive Gaussian noise $\boldsymbol{n}$. This entire work considered $\boldsymbol{n}$ and the update vector $\boldsymbol{u}$ as the only random signals present, as well as z, since it is obtained from a linear combination of $\boldsymbol{u}$ and $\boldsymbol{n}$. Robustness to noise was achieved by means of regularization and by making the regularization parameters dependent on data.

In our case, the regularization matrix is no longer of the form $\lambda \boldsymbol{I}$, where λ is a scalar, but has a more general form $\boldsymbol{\Lambda}$. The entries of $\boldsymbol{\Lambda}$ form a set of regularization parameters and such a formulation allows us more possibilities when it comes to find the best smoothed estimate.

A spatially adaptive approach was introduced and it consists of using a set of masks, each one representing a different neighborhood and yielding a distinct estimate. The final estimate is the one that provided the smallest $|DFD|$. The results from some experiments demonstrated the advantages of employing multiple masks.

A strategy for choosing the regularization parameter without knowledge of the noise statistics was introduced: the GCV. It depends solely on the observations. Two cases were explored: scalar λ and $\boldsymbol{\Lambda} = diag\{\lambda_1, \lambda_2\}$. The best results were obtained using the GCV with $\boldsymbol{\Lambda} = diag\{\lambda_1, \lambda_2\}$. All implementations of the GCV presented in this article performed better than the LMMSE technique. The drawbacks exhibited by the GCV in the context of motion estimation/detection were also analyzed. GCV evaluates the variability of the regression results when some subset of observations is omitted from the original set z. The criterion to select the best $\boldsymbol{\Lambda}$ is the minimum prediction MSE. Since the regularization matrix $\boldsymbol{\Lambda}$ is related to the autocorrelation of the data, there is some implicit stochastic knowledge in our model. The main advantage of our approach is the fact we use the GCV to choose the best set of regularization parameters and, then, we use these values to calculate the update vector estimate $\hat{\boldsymbol{u}}_{GCV}$. This take on GCV brings in an implicitly Bayesian touch because the entries of $\boldsymbol{\Lambda}$ are actually variance ratios. We improved the GCV performance via the introduction of local adaptability (through the use of multiple neighborhoods).

It should be pointed out that our GCV model can handle the motion estimation/detection problem well and, as expected, a more complex $\boldsymbol{\Lambda}$ gives better result than a scalar regularization parameter.

The proposed method provides an automatic, data-based selection of the regularization parameter by means of the minimization of Eq. (14). The GCV is a non-parametric estimation method that does not require any knowledge about the probability density functions of the model variables, although the regularization parameters sought are related to the covariance matrices of $\boldsymbol{u}$ and $\boldsymbol{n}$. It relies solely on the minimization of a function obtained from the weighted sum of squared prediction errors.

However, the technique presents some drawbacks [8]. This technique works very well in most of the cases (approximately 95% of the time), but due to the volume of minimizations done, the GCV failed to produce good estimates at all points because of one of the following situations:

- **a)** *GVC*($\boldsymbol{\Lambda}$) has multiple minima;
- **b)** There is no minimum such that all entries of $\boldsymbol{\Lambda}$ are positive;
- **c)** The minimum is hard to be found (no convergence);
- **d)** The global minimum of the *GCV* results in a undersmoothed solution; a local minimum can be better; or
- **e)** We may have found a saddle point.

Our spatially adaptive scheme indeed improves the behavior of the routines based on GCV around motion borders due to the fact that it seeks the neighborhood which provides the best system of equations according to the smoothness constraint assumption.

An interesting problem we are currently investigating, is a more intelligent way of choosing a neighborhood upon which to build our system of equations. We are also looking at more complex regularization matrices.

REFERENCES

[1] J. Biemond, L. Looijenga, D. E. Boekee, R. H. J. M. Plompen, "A pel-recursive Wiener-based displacement estimation algorithm," Signal Proc., 13, 1987, pp. 399-412.

[2] J. C. Brailean, A. K. Katsaggelos, "Simultaneous recursive displacement estimation and restoration of noisy-blurred image sequences," IEEE Trans. Image Proc., Vol. 4, No. 9, 1995, pp. 1236-1268.

[3] V. V. Estrela, N. P. Galatsanos, "Spatially-adaptive regularized pel-recursive motion estimation based on cross-validation," ICIP 98 Proceedings (2), 1998, pp. 200-203.

[4] N. P. Galatsanos, A. K. Katsaggelos, "Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation," IEEE Trans. Image Proc., Vol. 1, No. 3, 1992, pp. 322-336.

[5] G. H. Golub, M. Heath, G.Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," Technometrics, Vol. 21, No. 2, 1979, pp. 215-223.

[6] A. K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall, New Jersey, 1989.

[7] A. M. Tekalp, Digital Video Processing, Prentice-Hall, New Jersey, 1995.

[8] A. M. Thompson, J. C. Brown, J. W. Kay, D. M. Titterington, "A study of methods for choosing the smoothing parameter in image restoration by regularization," IEEE Trans. P.A.M.I., Vol. 13, No. 4, 1991, pp. 326-339.

[9] G. Wahba, Spline Models for Observational Data, SIAM, Philadelphia, 1990.

[10] J.L. Barron, D.J. Fleet, and S.S. Beauchemin, "Performance of optical flow techniques," International Journal of Computer Vision, 12**,** 1994, pp. 43–77.

[11] MPEG-7 Overview, ISO/IEC JTC1/SC29/WG11 WG11N6828, 2004.

[12] T. Ebrahimi, Y. Abdeljaoued, R. Figueras i Ventura, and O. Divorra Escoda, "MPEG-7 Camera," IEEE Proc. Int. Conference on Image Processing (ICIP), Thessaloniki, Greece, Oct. 2001.

[13] R. Kapela and A. Rybarczyk, "Real-time shape description system based on MPEG-7 descriptors," Journal of Systems Architecture, Volume 53, Issue 9, 2007, pp. 602-618.

[14] R. Kapela, and A. Rybarczyk, "The neighboring pixel representation for efficient binary image processing operations," International Symposium on Parallel Computing in Electrical Engineering (PARELEC'06), 2006, pp. 396-404.

[15] A. Petrovic, O. Divorra Escoda and P. Vandergheynst, "Multiresolution segmentation of natural images: from linear to non-linear scale-space representations," IEEE Transactions on Image Processing, Vol. 13, No 8, 2004, pp. 1104-1114.

[16] A. Bab-Hadiashar, N. Gheissari, and D. Suter, “Robust Model Based Motion Segmentation,” ICPR 2002, Quebec, Canada, 2002, pp. 753-757.

[17] N. Gheissari, and A. Bab-Hadiashar, “Motion analysis: model selection and motion segmentation,” Proceedings of 12th International Conference on Image Analysis and Processing, 2003, pp. 442-448.