VIRTUAL OBSERVATORY: Plate Content Digitization, Archive Mining & Image Sequence Processing
edited by M. Tsvetkov, V. Golev, F. Murtagh, and R. Molina, Heron Press, Sofia, 2005

Motion Estimation Techniques in Super-Resolution Image Reconstruction. A Performance Evaluation

D. Barreto[1,2], **L. D. Alvarez**[2], **J. Abad**[2]

[1]Research Institute for Applied Microelectronics (IUMA),
University of Las Palmas de Gran Canaria, Spain

[2]Department of Computer Science and Artificial Intelligence,
University of Granada, Spain

Abstract.

For a number of problems involving image sequences, it is essential to have a good motion estimate. In particular, when obtaining a high resolution image or video sequence from a set of low resolution images [1], the quality of the estimated motion plays a critical role in the performance of the high resolution algorithm. In this paper, we evaluate two different non-parametric motion estimation techniques, one based on block-matching [2] and the other based on the optical flow equation [3], to obtain sub-pixel displacements between frames. We will also consider the local quality of the displacement map estimates to improve the high resolution estimations [4]. The methods are tested on real images.

1 Introduction

Visual communications play a very important role nowadays. Our society is provided with the most appealing technology to capture, transmit, store, reproduce, analyze and visualize images. The quality of an image is mainly determined by its spatial resolution, that is, the number of pixels per unit area. The more resolution an image has the more pleasant the picture is perceived. This is due to the additional details a high resolution (HR) image offers. High precision optics and sensors can produce HR images at the expense of a prohibitively expensive cost and serious limitations like shot noise. Therefore, the use of signal processing techniques is a very promising way to improve the spatial image resolution. Super-Resolution (SR) is the term used to describe techniques that attempt to reconstruct a higher resolution still image or video sequence from a sequence of

low resolution images. SR underlying idea relies on adding new information to a target frame by detecting the sub-pixel displacements between such frame and its adjacent ones. These sub-pixel shifts are produced by the camera and/or by the motion of objects in the scene.

The importance of a good motion estimation (ME) in Super-Resolution naturally arises. Independently of the SR method used (frequency domain, non-uniform interpolation or statistical methods), the underlying motion estimation technique must be as accurate as possible [5]. While literature can be found on SR image reconstruction (see [1] and [15] for an overview), little attention has been paid to the performance of motion estimation techniques when used in SR problems. Schultz et al. [6] compared several sub-pixel motion estimation techniques into the Bayesian multiframe enhancement algorithm: eight-parameter projective motion model, block-matching and Horn-Schunck optical flow estimation. They concluded that each technique has its own advantages and disadvantages depending on the kind of motion and the image characteristics. Quantitative comparisons between the enhanced image and the original one were made using the Improved Signal-to-Noise Ratio (ISNR). They detected inaccurate motion vectors by applying the displaced frame difference (DFD) between the up-sampled and compensated frames. Our research tries to extend this previous work.

In this paper, we also address the study of the effects introduced by different motion estimation techniques when they are applied to the SR problem. Two motion estimation techniques have been chosen for being commonly used. These are block-matching [7] and optical flow [12]. The accuracy of the motion estimates will also be studied to determine when the motion estimate should be used. The obtained motion fields will be used by two different SR methods, one based on non-uniform interpolation.

Several real image sequences, subject to different kinds of motion (local and global) and motion velocities, have been selected for our experiments. From these real HR image sequences, the low resolution (LR) image sequences will be obtained by under-sampling. Then, using the SR and ME methods already described, we will obtain reconstructed HR frames (or sequences applying a sliding-window) for each image sequence under study. Finally, we will compare these reconstructed frames with their corresponding ones in the original sequences. In this way, we will assess the performance of the considered motion estimators in SR.

This paper is organized as follows. Section II explains the motion estimation techniques under consideration: block-matching and optical flow. Section III introduces the observability criteria used to asses the quality of the estimated motion vectors. Section IV describes the SR algorithms developed and the role the motion estimation plays on them. In section V we present the results obtained for the different image sequences and the visual and numerical comparisons between SR methods. Finally, section VI provides conclusions and areas of future work.

2 Notation

Let us denote by $g_l(x, y)$ a pixel in the l-th LR image where x and y represent the spatial coordinates in the image plane coordinate system. The size of the LR image is $M \times N$. If a LR image is divided into blocks of size $m \times n$, the total number of blocks in the image will be $B = \frac{M}{m} \times \frac{N}{n}$. If bm and bn are the number of blocks in the horizontal and vertical axes respectively ($bm = \frac{M}{m}$ and $bn = \frac{N}{n}$), we denote a generic block in the LR image as represented by $g_l(bx, by)$ where $bx = 1, ..., bm$ and $by = 1, ..., bn$. The notation for an entire image is $\mathbf{g}_l$, representing the l-th image in the LR sequence. For this paper, the complete LR sequence that we utilize is denoted by $(\mathbf{g}_{k-2}, \mathbf{g}_{k-1}, \mathbf{g}_k, \mathbf{g}_{k+1}, \mathbf{g}_{k+2})$, where g_k denotes the image we want to improve its resolution.

A HR image pixel is denoted by $f_l(x, y)$. If P is the magnification factor ($P > 1$), the size of one HR image is $PM \times PN$. A HR image block is denoted as $f_l(bx, by)$ and the size of each block in which we can divide our HR image is $Pm \times Pn$. The HR image we seek to estimate is $\mathbf{f}_k$, being the complete HR sequence $(\mathbf{f}_{k-2}, \mathbf{f}_{k-1}, \mathbf{f}_k, \mathbf{f}_{k+1}, \mathbf{f}_{k+2})$.

3 Problem Formulation

The horizontal and vertical components of the displacement of a pixel (x, y) from the k-th LR image into the l-th image are represented by $d_{l,k}^{x,lr}$ and $d_{l,k}^{y,lr}$ respectively, thus $d_{l,k}^{lr}(x, y) = (d_{l,k}^{x,lr}, d_{l,k}^{y,lr})$ and we have $g_k(x, y) = g_l(x + d_{l,k}^{x,lr}, y + d_{l,k}^{y,lr})$. The motion field between frames $\mathbf{g}_k$ and $\mathbf{g}_l$ is denoted by $\mathbf{d}_{l,k}^{lr}$.

For HR images, motion vectors will be denoted by $d_{l,k}^{hr}(x, y) = (d_{l,k}^{x,hr}, d_{l,k}^{y,hr})$, and $\mathbf{d}_{l,k}^{hr}$ is the motion field between frames $\mathbf{f}_k$ and $\mathbf{f}_l$. In other words,

$$f_k(x, y) = f_l\left(x + d_{l,k}^{x,hr}, y + d_{l,k}^{y,hr}\right). \tag{1}$$

To obtain a LR image $\mathbf{g}_l$, from the corresponding HR image $\mathbf{f}_l$, we will decimate the HR image by taking one pixel each P in rows and columns. This process is depicted in Figure 1, mathematically we have

$$g_l(x, y) = f_l\left(Px, Py\right)\ l = k - 2, ..., k + 2 \tag{2}$$

and our goal is to estimate $\mathbf{f}_k$ from $(\mathbf{g}_{k-2}, ..., \mathbf{g}_{k+2})$

4 Motion Estimation Techniques

The large regions overlap that usually exists between successive frames of the same sequence and the multiple sampling of this regions in several frames, yield the conclusion that it is possible to combine this information to achieve higher spatial resolution images. Motion estimation techniques are used to find this overlapping areas from frame to frame. The resulting motion vectors must be

sub-pixel precision to provide useful information for SR. Therefore, two sub-pixel ME techniques have been applied: quarter-pixel block-matching and optical flow.

4.1 Block-Matching Motion Estimation

Block-matching algorithms represent a very popular approach for estimating the motion between frames in an image sequence. They have a performance good enough to be used in video coding for transmission or compression purposes. Another important reason for this wide use is the low computational cost it involves. Block-Matching algorithms simply divide one frame into blocks of size $m \times n$ (usually $m = n$). These algorithms search for each block in certain area ($a = (2p + n)^2$ if $m = n$) of another frame in the sequence. They are then assuming that every pixel in a block has the same movement. Choosing a large block size implies less motion vectors to calculate, but the probabilities for each block to contain objects with different movements are higher. On the other hand, if the block size is too small, it increases the computation but it is also more sensitive to the noise present in the image. For our purposes, a block size of 4×4 is considered to be a good trade-off between computational cost and object motion detection. To define the search area, the parameter p takes the value of 7. These concepts are graphically illustrated in Figure 2.

There are several similarity criteria block-matching can use to determine the best match between the current and the displaced images. We use the Sum of Absolute Differences (SAD) given as

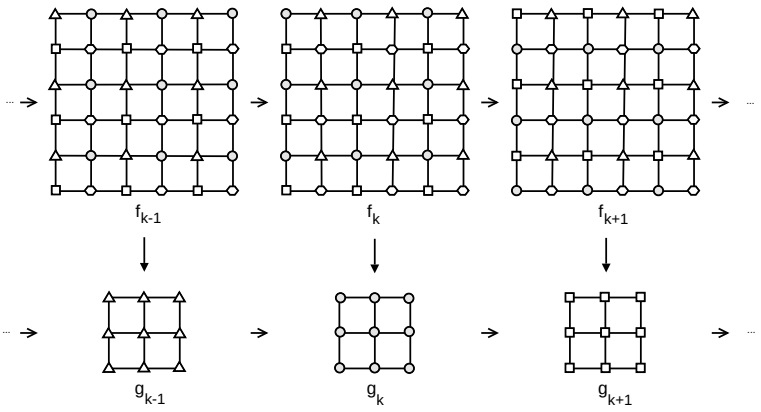

Figure 1. Procedure to obtain a LR image sequence from a HR image sequence by downsampling.

$$SAD_{d_{l,k}^{x,lr},d_{l,k}^{y,lr}} = \sum_{i=1}^{m}\sum_{j=1}^{n}\left|g_k(i,j) - g_l\left(i + d_{l,k}^{x,lr}, j + d_{l,k}^{y,lr}\right)\right|, \qquad (3)$$

where we are considering low resolution data g for the motion estimation. Low values of this measure show high correlation between the current block and the displaced block according to a particular motion vector $d_{l,k}^{lr}(bx, by)$. The block that turns out to be the most similar to the current block (bx, by) in the reference frame, is given by the candidate vector with the minimum SAD:

$$SAD_{bx,by} = min\left(SAD_{d_{l,k}^{x,lr},d_{l,k}^{y,lr}}\right), \qquad (4)$$

for all motion vectors in the search area a.

Searching for the best block in every position of the search window is called full search block-matching. Faster algorithms exist that decrease the computational load (2-D logarithmic search, three-step search, etc), but they usually degrade the quality of the estimations. Full search ME is implemented here, thus the global optimal displacement vector is obtained. We now used the estimated LR motion vectors as initial values of the HR motion vector we want to estimate. In order to achieve quarter sub-pixel precision motion estimation, interpolated image data is needed. Our procedure is divided into the following steps, which are commonly used [14]:

Step 1 Calculate the best motion vector at pixel precision.

Step 2 Search for the best half-pixel precision motion vector at eight half-pixel positions around the point of minimum given in step 1. These new positions contain interpolated values.

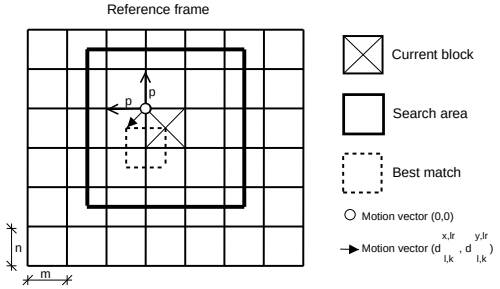

Figure 2. Block-matching frame division in blocks of size $m \times n$. The reference block that is the most similar to the current block is found within the search area defined by the parameter p.

Step 3 Compare the current block with eight new interpolated blocks around the best half-pixel position. These blocks contain values that are interpolated from pixel precision image data and half-pixel precision interpolated data.

4.2 Optical Flow Motion Estimation

Since the Lucas-Kanade algorithm was proposed in 1981 for image alignment it has been widely used in computer vision for multiple tasks, see [12] for review. One of them has been to compute the motion field between a pair of images within an image sequence or video. This technique calculates motion vector estimates $\left(d_{l,k}^{x,lr}, d_{l,k}^{y,lr}\right)$ between LR images l-th and k-th, which satisfy the optical flow equation with the minimum pixel to pixel variation in the velocity field. There are multiple variations in the literature of this method, depending on the purpose, we will implement the algorithm following [13].

The motion vector for each pixel (x, y) is defined by $d_{l,k}^{lr}(x, y) = \left(d_{l,k}^{x,lr}, d_{l,k}^{y,lr}\right)$. We will center a window of size $m \times n$ on each pixel, the error to be minimized in the optical flow equation will be as follows

$$LK\left(d_{l,k}^{x,lr}, d_{l,k}^{y,lr}\right) = \sum_{i=x-m/2}^{x+m/2} \sum_{j=x-n/2}^{x+n/2} \Big(g_k(i,j) - \\ -g_l(i + d_{l,k}^{x,lr}, j + d_{l,k}^{y,lr})\Big)^2. \quad (5)$$

Using a second-order Taylor expansion for the second term of Equation 5 we have

$$LK\left(d_{l,k}^{x,lr}, d_{l,k}^{y,lr}\right) = \sum_{i=x-m/2}^{x+m/2} \sum_{j=x-n/2}^{x+n/2} \Big(g_k(i,j) - \\ -g_l(i,j) - d_{l,k}^{x,lr} g'(i,j) - d_{l,k}^{y,lr} g'(i,j)\Big)^2, \quad (6)$$

and we have

$$\hat{d}_{l,k}^{lr}(x, y) = argmin_d LK\left(d_{l,k}^{lr}\right). \quad (7)$$

To obtain more accurate motion vectors, two processes has been added: motion vectors refinement and a hierarchical approximation. The refinement is based on a iterative estimation of the motion vectors (using 5 iterations). Once the initial motion vectors are calculated the image is compensated via this motion field and Equation 6 is again applied for each pixel. The hierarchical improvement is achieved using a pyramid of three levels to calculate the motion vectors in a coarse to fine approximation. The algorithm, including these improvements, can be summarized in the following steps:

Step 1 Downsample the images by a factor of 2 and 4 to build the hierarchical pyramid.

Step 2 Compute the motion field for the lowest level in the pyramid by minimizing the optical flow equation for each pixel as shown in Equation 6.

Step 3 Iterate over the motion vectors to refine.

Step 4 For each higher level in the pyramid

Step 4.1 Propagate the motion vectors to the higher level.

Step 4.2 Iterate to refine.

Note that because of the way the Lucas-Kanade motion estimation method works, the estimated motion vectors are real numbers (not only integers) and so they can be of sub-pixel size at low resolution levels.

5 Where to Apply SR Methods?

5.1 Observability Map

As we have seen before, accurate motion estimation plays a very important role in SR problems. Pixels associated with poor and/or conflicting motion estimates have a bad effect on the quality of the reconstruction and therefore should not be taken into account. This is an important point, since the SR algorithm will assume that we can obtain a frame by motion compensating other frames. This is not true in some cases due to occlusions and poor motion estimation results.

The absolute value of the Displaced Frame Difference (DFD), given by

$$\mathrm{DFD}\left(\mathbf{g}_l, \mathbf{d}_{l,k}^{lr}, \mathbf{g}_k\right)(x,y) = \left| g_k(x,y) - g_l\left(x + d_{l,k}^{x,lr}, y + d_{l,k}^{y,lr}\right)\right| \qquad (8)$$

will allow us to identify pixels in $\mathbf{g}_k$ that are not predictable by $\mathbf{d}_{l,k}^{lr}$ and $\mathbf{g}_l$. Large values of the DFD will be used to identify non predictable/observable pixels in $\mathbf{g}_k$.

5.2 Homogeneity Map

We have also observed that highly textured regions usually give good motion estimates, while the results obtained from smooth areas do not offer any improvement in the quality of the compensated frame. In this sense, we will consider the possibility of not using the motion estimates coming from flat regions in our SR algorithm.

In order to develop this criterion we will define an homogeneity map, characterizing pixels that come from very flat areas in order to avoid them in the SR algorithm.

Smooth areas will be identified by using the Intra Sum of Absolute Differences (Intra_SAD), defined for every block in the image $\mathbf{g}_k$ as

$$\text{Intra\_SAD} = \sum_{i=1}^{m}\sum_{j=1}^{n} |g_k(i,j) - \mu| \,, \tag{9}$$

where $m \times n$ is the block size and μ is the mean of the block. We can see that low values of the Intra_SAD will identify poorly textured blocks/areas.

If necessary, we will define Intra_SAD for each pixel in $\mathbf{g}_k$ with regard to the mean of its neighborhood.

6 Super-Resolution Algorithms

Interpolation is the most intuitive approach for SR image reconstruction. After estimating the relative motion between frames by the techniques described before, it is possible to project on the desired HR grid the samples where the motion vectors point to. Ur and Gross [8] used the generalized multichannel sampling theorem of Papoulis and Brown to perform a non-uniform interpolation. Alam et al. [9] utilized a weighted nearest neighbor interpolation method. Nguyen and Milanfar [10] proposed a wavelet interpolation for interlaced two-dimensional data. Callico et al. [11] modified an existing hybrid video encoder platform to add SR based on a non-uniform interpolation method. They defined a HR grid where the pixel values of the current image are first placed, while the rest of HR positions remain empty (zero values). A new frame, with sub-pixel shifts with respect to the current image, involves new data to be placed onto the HR grid.

The main advantage of non-uniform interpolation SR is that the computational complexity is low, making real-time applications possible. The SR algorithm implemented here is based on [11]. A description of the steps the SR algorithm follows, which are also summarized in Figure 3, is presented next:

Step 1 Using Equation 2 for $l = k$, place the LR image pixels in their corresponding positions of the HR image $\mathbf{f}_k$

Step 2 New pixels from frames $l \neq k$ will be placed in empty positions of the HR grid if there is sub-pixel motion.

Step 3 After filling all possible positions, empty HR pixel values will be interpolated to achieve the final SR image. Two interpolation methods will be used.

For each empty position in the HR grid, the first method simply takes the information that has been placed at $\frac{1}{P}$ pixel positions, discarding pixel values at a larger precision according to the magnification factor P established. That is, if the magnification factor is 2 we will not used quarter pixel precision. The second

method takes its eight non-empty neighbor values for every $\frac{1}{P}$ pixel position to perform a bilinear interpolation, taking then into account higher precision pixel information. Both methods fill remaining empty HR positions with values from the up-sampled version of the current image using bilinear interpolation.

7 Results

For testing, we use in this paper two image sequences. The first one, *Coastguard*, is a sequence with small object motion (local motion) and background small motion (global motion). The second one, *Mobile*, is a sequence containing fast local motion and moving background with small motion. Both sequences are in CIF format (352×288 pixels). A HR image $\mathbf{f}_k$ is computed from b LR images backwards and f LR images forward. Both test image sequences are first down-sampled by a factor of 2 in both vertical and horizontal directions.

Figure 4 shows the original HR frames we want to estimate in both sequences. The observed LR image sequences are shown in Figures 5 and 6. We want to improve the quality of the images in the middle of the shown LR sequences.

Three regions have been chosen within each image sequence in order to isolate different motion and image characteristics. Figure 7 shows the selected regions for the *Mobile* and *Coastguard* sequences. The regions inside the square with dash-dotted line correspond to textured areas subjected to translational motion (*region 1*). (*Region 2*) is shown within a square with dashed line and corresponds to textured areas with two objects with independent motions. Finally, the square with solid line (*region 3*) emphasizes a flat area (for the *Mobile* image sequence) and a semi-flat area (for the *Coastguard* sequence) with translational motion.

The Peak Signal to Noise Ratio (PSNR) is used as a quantitative measure

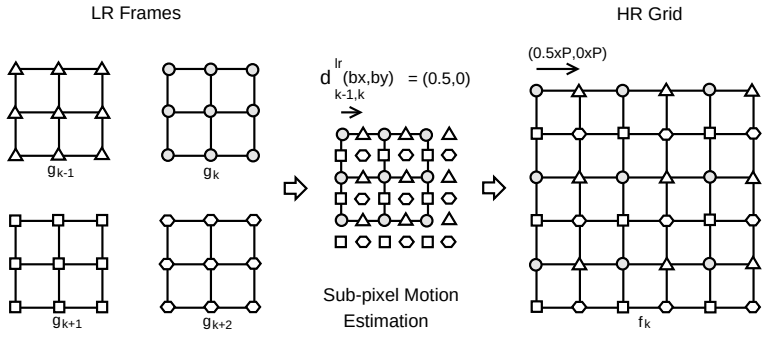

Figure 3. Non-uniform interpolation Super-Resolution. Projection of LR samples onto a high resolution grid according to the estimated motion vectors.

(a) (b)

Figure 4. (a) *Coastguard* k-th original HR image, (b) *Mobile* k-th original HR image, being $k = 9$.

of comparison between the original HR images, the bilinear interpolated LR images and the reconstructed SR images.

The procedure followed to test our SR methods is the following:

1. Using $b = 2$ and $f = 2$, a down-sampling by a factor of 2 is performed on each sequence.

2. Block-matching and Lucas-Kanade ME between LR frame $\mathbf{g}_k$ and LR frames $\mathbf{g}_{k-2}$, $\mathbf{g}_{k-1}$, $\mathbf{g}_{k+1}$ and $\mathbf{g}_{k+2}$, are then applied. These motion vectors will be used in combination with the observability criteria defined in section 5.

3. The estimated motion fields are used by the non-uniform interpolation SR algorithm, in combination with the two different criteria described at the end of section 6, to produce the HR reconstruction.

Figure 5. *Coastguard* LR sequence from frame 7 to frame 11.

Figure 6. *Mobile* LR sequence from frame 7 to frame 11.

(a) (b)

Figure 7. Three different regions have been selected to isolate motion and image characteristics for both image sequences *Coastguard* (a) and *Mobile* (b). The region with dash-dotted line (*region 1*) corresponds to a textured area with translational motion. The square with dashed line (*region 2*) shows a textured area with independent object motion. The last region (*region 3*), with a solid line, represents a flat area with translational motion.

4. PSNR values between the original HR frame $\mathbf{f}_k$ and the resulting SR images are calculated.

As it has already been mentioned, the spatial resolution will be increased by a factor of 2. Thus, it is very interesting to study if quarter pixel motion information, which will be clearly useful for a magnification factor of 4, really improves the PSNR in our case study. *Criterion 1* discards quarter pixel motion vectors when filling the HR grid, the use of these motion vectors is represented in the figures as *Criterion 2*.

For every HR grid interpolation criterion we use four thresholds to define the observability map. Table 1 shows these thresholds for both, the Intra_SAD and DFD observability functions. The thresholds have been estimated using two statistical measurements, the mean and the standard deviation of the distribution given by the Intra_SAD or DFD values. To the mean it has been added or subtracted zero or $\frac{1}{2}$, $\frac{1}{4}$ a part of the standard deviation. It has been empirically proved that those are the best choices for our purposes.

For each region of interest, given one of the two interpolation criteria and one of the four possible thresholds for the observability function, we have seven PSNR: the corresponding to bilinear interpolation of the LR image, and six defined by combinations between the motion algorithms and the observability functions. The following abbreviations are used, *interpolation* means that bilinear interpolation has been used to upsample the image, *me* denotes motion estimation method which can be Block-matching (*BM*), Lukas-Kanade method (*LK*). *om* is used to denote observability map and it can take the values Intra_SAD (*SI*), displacement field difference (*DFD*) or none.

Quantitative comparisons between reconstructions are in Figure 8 for the

Coastguard sequence and in Figure 9 for the *Mobile* sequence. Each plot corresponds to one of the selected regions in the images.

Figures 8 (a) and 9 (a) show the PSNRs of the reconstructions of (*region 1*) in both images. It can be concluded that it is important to take quarter pixel motion estimation into account in such areas, since *Criterion 2* interpolation method provides the best PSNR results. Therefore, quarter pixel motion estimation performs well when translational motion exists in textured areas. This conclusion is valid for both Lucas-Kanade (LK) and Block-Matching (BM) motion estimation techniques, which is specially evident in Figure 8 (a) for the *Coastguard* sequence. It can be seen that the best results are achieved when no observability map is used due to the elimination it involves. This figure is also useful to conclude that BM is very efficient when there is much texture in the region. Figure 9 (a), which corresponds to an area where there is texture but also some very flat sub-regions, the performance of BM decreases and LK achieves the best results. In this case, BM improves its PSNR values when used in combination with Intra_SAD, which is useful to detect flat areas.

The next studied regions are those containing objects with motions. Several ideas have been concluded from the PSNR results shown in Figures 8 (b) and 9 (b) for both image sequences (*region 2*). The first and most remarkable conclusion refers to the fact that the lowest PSNR results are obtained when using quarter pixel motion information. This shows that, when the motion is independent from the camera motion, increasing the accuracy of the motion estimation can add errors to the global SR process. This kind of regions requires, for both motion estimation algorithms, the DFD observability map to improve the final PSNR results. The improvement achieved by the use of the use of DFD is noticeable in Figure 9 (b), where the complexity of the motion is higher (rotation and translation).

Finally, flat regions with translational motion are studied from the PSNR results given in (*region 3*) depicted in figures 8 (c) and 9 (c). For both motion estimation techniques, the best PSNR results are obtained when the Intra_SAD observability map is applied, because it detects flat regions. As for previous kind of regions, quarter pixel motion vectors are not useful for SR in this case.

Table 1. Thresholds for both, Intra_SAD and DFD, observability criteria. μ represents the mean value for each observability criterion in the current image. σ represents the standard deviation for each observability criterion in the current image.

Thresholds	Intra_SAD	DFD
thr 1	$\mu_{IS} - \sigma_{IS}/2$	$\mu_{DFD} + \sigma_{DFD}/2$
thr 2	$\mu_{IS} - \sigma_{IS}/4$	$\mu_{DFD} + \sigma_{DFD}/4$
thr 3	μ_{IS}	μ_{DFD}
thr 4	$\mu_{IS} + \sigma_{IS}/4$	$\mu_{DFD} - \sigma_{DFD}/4$

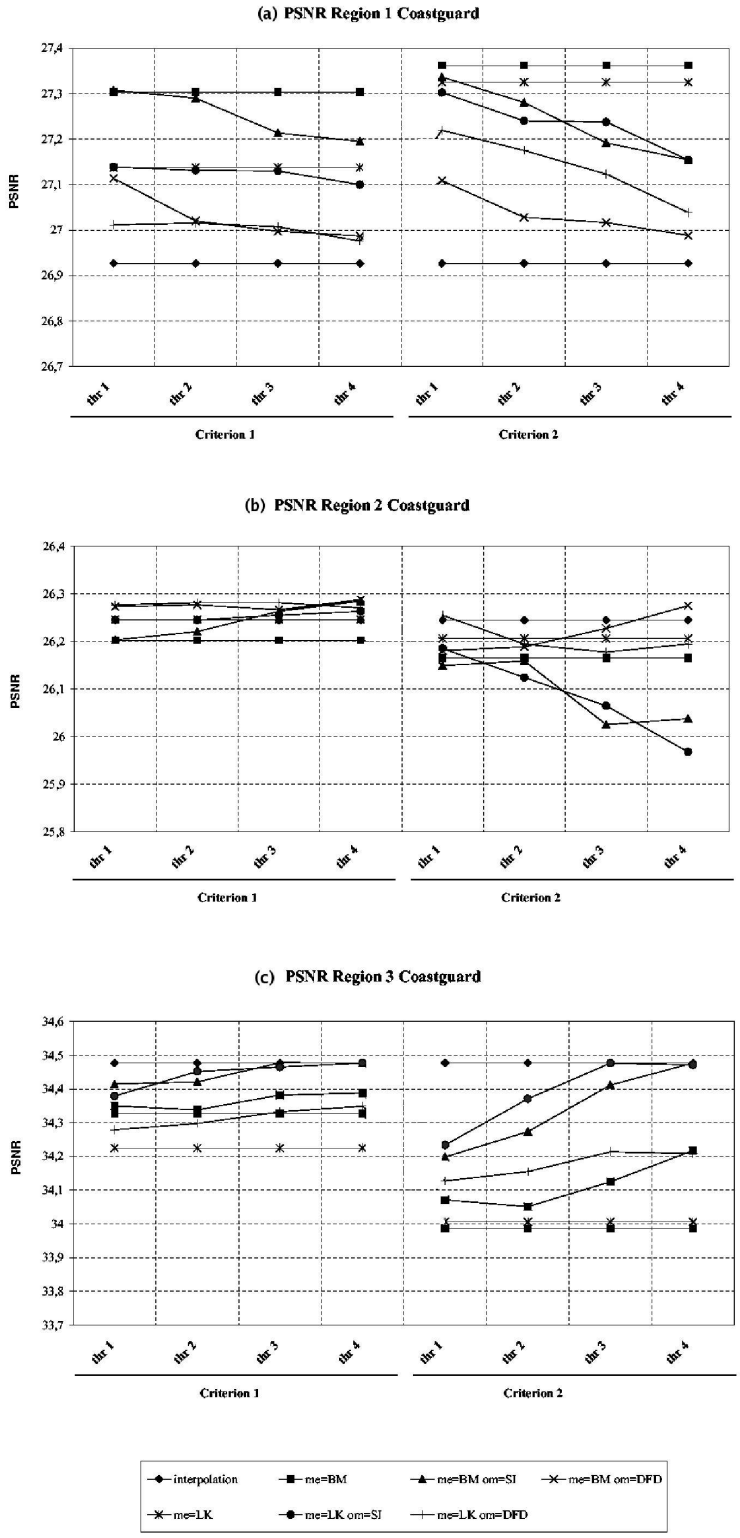

Figure 8. (a) PSNR values for region 1 in *Coastguard* k-th frame. (b) PSNR values for region 2 in *Coastguard* k-th frame. (c) PSNR values for region 3 in *Coastguard* k-th frame.

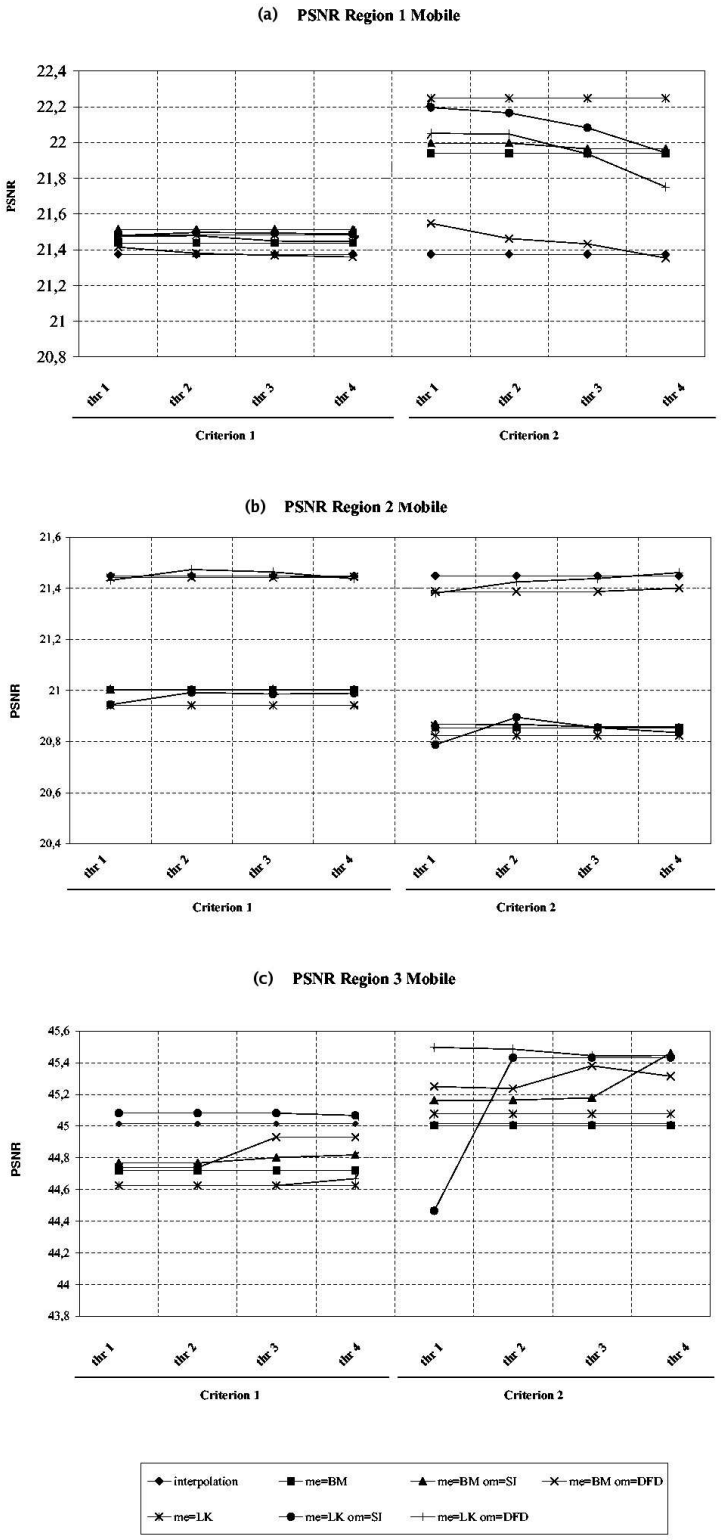

Figure 9. (a) PSNR values for region 1 in *Mobile* k-th frame. (b) PSNR values for region 2 in *Mobile* k-th frame. (c) PSNR values for region 3 in *Mobile* k-th frame.

8 Conclusions

Two sub-pixel motion estimation techniques, Block-Matching and Lucas-Kanade, have been applied within a Non-Uniform Interpolation Super-Resolution environment. Two observability criteria have been used in combination with the displacement maps in order to select the pixels that contribute to the SR process. Quantitative comparisons based on PSNR measurements have been obtained for every motion estimation technique with and without the observability maps.

The main conclusion of this work is that the global SR process is region-dependent. It has been shown that motion estimation does not perform well in flat regions. Therefore, these regions cannot be improved by the global SR process and their approximation by bilinear interpolation is the best tradeoff between computational complexity and image visual quality. Motion estimation obtained in textured areas affected by translational motion is accurate, and it has a positive repercussion on the SR process. The higher the precision of the motion vectors the better PSNR results can be reached. Finally, the improvement of areas containing independent object motion is very sensitive to the correctness of the motion field and thus needs to be used in combination with the observability maps, discarding then wrong motion estimations.

A segmentation process previous to the SR could be very beneficial to achieve, not only an improvement in the image quality by exploiting the characteristics of each region but also a reduction of the computational load.

Acknowledgments

The authors wish to thank Professor R. Molina for his constructive comments and his thoughtful reviews.

This work has been supported by the "Comisión Nacional de Ciencia y Tecnología" under contract TIC2003-00880.

References

[1] S. C. Park, M. K. Park, and M. G. Kang (May 2003) *Super-resolution Image Reconstruction: a Technical Overview*, IEEE Signal Processing Magazine, Vol. 20 **3** 21-36.

[2] R. R. Schultz and R. L. Stevenson (June 1996) *Extraction of High Resolution Frames from Video Sequences*, IEEE Transactions on Image Processing, Vol. 5 **6** 996-1011.

[3] C. A. Segall, R. Molina, A. K. Katsaggelos, and J. Mateos (July 2004) *Bayesian Resolution Enhancement of Compressed Video*, IEEE Transactions on Image Processing, Vol. 13 **7** 898-911.

[4] L.D. Alvarez, J. Mateos, R. Molina, and A.K. Katsaggelos (April 2004) *Observability and Predictibility in Super-Resolution*, Astronomical Data Analysis III, Sorrento, Naples, Italy.

[5] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar (2004) *Advances and Challenges in Super-Resolution*, International Journal of Imaging Systems and Technology, Special Issue on High Resolution Image Reconstruction, Vol. 14 **2** 47-57.

[6] R. R. Schultz, L. Meng, and R. L. Stevenson (March 1998) *Subpixel Motion Estimation for Super-Resolution Image Sequence Enhancement*, J. Visual Communication and Image Representation, Vol. 9 **1** 38-50.

[7] P. Kuhn(1999) *Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion-Estimation*, Kluwer Academics Publisher, Chap. 2 17-60.

[8] H. Ur and D. Gross (March 1992) *Improved Resolution from Sub-pixel Shifted Pictures*, CVGIP: Graphical Models and Image Processing, Vol. 54 181-186.

[9] M. S. Alam, J. G. Bognar, R. C. Hardie, and B. J. Yasuda (October 2000) *Infrared Image Registration and High-Resolution Reconstruction using Multiple Translationally Shifted Aliased Video Frames*, IEEE Trans. Instrum. Meas., Vol. 49 915-923.

[10] N. Nguyen and P. Milanfar (2000) *An Efficient Wavelet-Based Algorithm for Image Superresolution*, Proc. Int. Conf. Image Processing, Vol. 2 351-354.

[11] G. M. Callico, A. Nuñez, R. P. Llopis, R. Sethuraman, and M. O. de Beeck (November 2002) *A Low-Cost Implementation of Super-Resolution Based on a Video Encoder*, IEEE 28th Annual Conf. of the Industrial Elec. Society, Vol. 2 1439-1444.

[12] S. Baker, and I. Matthews (2004) *Lucas-Kanade 20 Years On: A Unifiying Framework*, International Journal of Computer Vision, Vol. 56 **3** 221-255.

[13] B. Lucas, and T. Kanade (1981) *An Iterative Image Registration Technique, with an Application to Stereo Vision*, International Journal Conference in Artificial Intelligence, 121-130.

[14] P. Kuhn (1999) *Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion-Estimation*, Kluwer Academics Publisher, The Netherlands.

[15] L. D. Alvarez, R. Molina, and A.K. Katsaggelos (2004) *High Resolution Images from a Sequence of Low Resolution Observations*, Digital Image Sequence Processing, Compression and Analysis, T. R. Reed (ed.), CRC Press, Chap. 9 233-259.