Local object-based super-resolution mosaicing from low-resolution video

Petra Krämer, Jenny Benois-Pineau, Jean-Philippe Domenger

Local object-based super-resolution mosaicing from low-resolution video

Petra Krämer[a,*], Jenny Benois-Pineau[b], Jean-Philippe Domenger[b]

[a] *L3i, University of La Rochelle, La Rochelle, France*
[b] *LaBRI, University of Bordeaux 1, Talence, France*

Abstract

Many efficient super-resolution methods have been presented in the past for improving resolution of images degraded by global blurs. Unfortunately, in video, more complex situations can be observed where local blurs appear in each frame which are mainly caused by object motions. To address this problem, we propose in this article a local super-resolution method which allows the restoration of such local blurs. Moreover, the motion of objects in video sequences may be very complex and particularly in very low-resolution sequences it is difficult to estimate their motion exactly enough to superimpose them for super-resolution. To this end, we present a generic method: An interpolation method is proposed to improve the resolution of moving objects and we derive from this a super-resolution method for the scene background.

Keywords: Super-resolution, mosaicing, local blur estimation/restoration
2000 MSC: 68U10, 94A08

*Corresponding author. Fax: + 33 5 46 45 82 42
Email addresses: pkraemer@univ-lr.fr (Petra Krämer), benois-p@labri.fr (Jenny Benois-Pineau), domenger@labri.fr (Jean-Philippe Domenger)

1. Introduction

Super-resolution (SR) is the process of combining a sequence of low-resolution (LR) images in order to produce a higher resolution image or sequence, the so-called SR image or sequence. SR is still a very active research area due to ever expanding application domains. For instance, SR methods have been adopted for still construction from video [38], video format conversion [30], demosaicking [26], improving the video resolution in a camera [4], reduction of compression artifacts [16], and deinterlacing [37]. A good overview of existing SR approaches is given in [36, 14] for raw video and in [41] for compressed video.

Even though powerful SR methods have been presented, only few of them address the problem of estimating and restoring spatially variant blurs. These blurs are mainly caused by object motions or atmospheric turbulence. One of the first methods for object-based SR was presented in [23] where an object with dominant motion is tracked over several frames assuming 2D parametric motion to improve its resolution. However, the method does not account for local blurs. The general approach of [3] is similar to [23], but incorporates motion blurring. The authors of [12] propose a SR method accounting for motion blur by extending the method [38] to the case of multiple moving objects in the scene. In [13], the method [12] is reused in the context of mesh-based object representation. For both, results are only shown for rigid objects. Other methods use optical flow to describe object motion [30] or atmospheric distortions [51]. A gradient-based shift estimator is used in [48] for object motion. Moreover, in [48, 31] the problem of super-resolving very small objects is addressed. The authors of [48] state that for small moving

objects the amount of information inside the moving object is too small for appropriate registration. In order to overcome this problem, they propose a polygon-based SR approach [49] in order to super-resolve very small objects. Nevertheless, they still assume rigid object motion. [31] proposes to super-resolve small objects by non-uniform interpolation followed by restoration. Objects are assumed to undergo affine motion, but the problem of local blur is neglected, the shape of the object is approximated by a rectangle and needs to be initialized by the user. A different approach to handle object motions are block-based methods such as [4, 2]. Recently, a new class of local SR methods have been derived from noise filters such as moving least squares [6], kernel regression [45], non-local means [39], or restoration filters like the Wiener filter [19]. [32] proposes a SR method based on locally adaptive bilateral total variation in order to keep edges sharp and flat regions smooth. As this kind of approaches are at the beginning of investigation, few of them account for local motion [39, 45] and none for local blur.

The motion of objects in real video sequences can be very complex. For example consider the sequence shown in Fig. A.4(a) with a walking person undergoing local motions. Particularly in the case of very LR images the objects are typically represented by only few pixels. For that reason, it is very difficult to estimate the motion of moving objects and to superimpose them accurately enough for SR, as also stated in [48]. Thus, the computation of a parametric motion model [23, 31] or optical flow [30] is not appropriate in that case. Furthermore, in contrast to [48, 49] where rigid object motion is assumed, our aim is to propose a generic method which can handle small objects with complex motions. Therefore, we present in this paper an inter-

polation method to improve the the resolution of moving objects and derive from this a SR method to super-resolve the scene background.

Recently, blind SR has become in the focus of interest [44, 35, 20, 21]. That means that the blur function and its parameters used in the SR process are estimated from the image data itself. Typically, the proposed SR methods rely on blind deconvolution methods. Although several methods for blind deconvolution have been proposed [28, 29, 7], at present only few methods have been presented for the case of spatially variant blurs as well. In [46], the image scene is segmented into regions where the blur is approximated as locally spatial invariant. The authors of [34] assume piece-wise constant blur functions which are stitched together by interpolation to obtain a continuous blur function. The method presented in [47] allows the blur to vary at each pixel location whereas the blur size is derived from the estimated displacement vector. In [1], a framework is presented which jointly handles motion estimation, moving object detection, and motion blur deconvolution. A motion blur model is incorporated which is consistent at motion singularities (caused by the moving object occluding and uncovering the background), but the method can only handle one single moving object. The method of [8] estimates motion blur kernels, segmentation, and motion simultaneously allowing multiple moving objects. A different approach is presented in [43] where blur variations due to different depths are considered. The method [25] computes the motion blur from a transparency map. In [22, 18], a block-based method for local blur restoration is presented. The authors of [10] present a method to track motion-blurred targets in video. To this end, local motion blurs are identified by a learning-based scheme.

In our recent work [27], we presented an efficient SR method for the blind restoration of global blurs due to the camera motion. The main drawback of this method is that restoration is performed in frequency domain which does not allow for processing local blurs. Here, we address these shortcomings. The SR method we present in this article performs restoration in spatial domain and thus enables the restoration of local blurs induced by the motion of objects. Based on a segmentation of moving objects, our SR method is able to estimate and locally restore blur. To this end, we extend our blur estimation method [27] for global motion blur due to the camera motion to the case of local blurs. This enhanced method can handle both, motion blur of the camera occurring in the static scene background and motion blurs of various moving objects.

We apply our SR method to the construction of super-resolved mosaics (SR mosaicing) from compressed video. The construction of mosaics from compressed video has become important as still new application domains emerge. First of all, a mosaic image can be efficiently used for visually summarizing compressed video content which is our case. Furthermore, mosaicing, specifically for compressed content, is a powerful tool for the reconstruction of a global scene view from compressed LR videos captured with mobile phones, e.g. document images [17]. In both applications SR mosaicing is a must for the efficient use of LR video data.

This article is organized as follows: In Sect. 2, we present the estimation of local blurs, namely on moving objects and the scene background, and the SR method which allows their restoration. Some results are shown and discussed in Sect. 3. Conclusions and perspectives are given in Sect. 4. Moreover,

Appendix A presents the derivation of our SR method.

2. Local Object-based Super-resolution

As stated above, it is very difficult to superimpose moving objects in very LR images. To this end, we propose to super-resolve separately moving objects and the scene background. This requires preliminary motion estimation in order to determine the geometric transformations of the background between the images of the sequence and the reference image, as well as segmentation and tracking of moving objects. Here, we suppose that this is already accomplished. We used [27] for registration and [33] for moving object segmentation and tracking.

2.1. Local Super-resolution in Spatial Domain

Basic iterative restoration techniques can be easily adapted to the case of local blurs. For this reason, we chose the basic iterative image restoration method [9, 24] as a basis for our SR algorithm. In order to derive from [9, 24] (see also Appendix A) a SR method to restore local blurs, we formulate a degradation model for an arbitrarily shaped region of interest (ROI) considering local motion and downsampling. Indeed, the image formation model [9, 24] was formulated for an ideal unknown image and its observed blurred version of the same resolution. In our case, when zooming on the mosaic, the ideal image is of high resolution, namely SR, and its observed blurred version is of LR. Hence, the ROI in the LR image G can be modelled as:

$$G = S^{-1}.B * [T.F] + V \tag{1}$$

where F is the ROI in the SR image, B the point spread function (PSF) defining the blur of the ROI, T the geometric transformation describing the local motion of the ROI, S^{-1} the downsampling operator by the factor z, and V the noise.

Based on this degradation model, we derive an iterative algorithm to restore and increase the resolution of a ROI:

$$F^i = F^{i-1} + T^{-1}.S\left(G - S^{-1}\left[B * \left(T.F^{i-1}\right)\right]\right) \tag{2}$$

where F^i is the ROI in the SR image at the ith iteration and S the upsampling operator. The proof can be found in Appendix A.1.

Considering a sequence of K LR images where each frame represents the same ROI $G(k), 1 \leq k \leq K$, the following iterative SR algorithm can be derived from (2) for a SR mosaic (see Appendix A.2):

$$M^i = M^{i-1} + \mu(K)\sum_{k=1}^{K} T^{-1}(k).S\Big(G(k) - S^{-1}\left[B(k) * \left(T(k).M^{i-1}\right)\right]\Big) \tag{3}$$

where M^i is the ROI in the SR mosaic at the ith iteration and $\mu(\mathbf{p}, K) = \frac{1}{|\mathbf{p}|}$ with $|\mathbf{p}|$ as the number of available pixels at position $\mathbf{p}$. We can notice that this equation is quite similar to the SR method we presented in [27], but the restoration operator is absent. The advantage here is, if we can identify the blurring operator, the restoration is straight forward and does not require the synthesis of a restoration filter as e.g. in the case of Wiener filtering.

A problem in (2) and (3) is that B is defined for the SR image which is unknown. Our objective is to directly use the LR PSF instead of making an

assumption on it for the SR image as we did in [27]. It can be shown that for small blurs (see Appendix A.3):

$$(S^{-1}.B) * (S^{-1}.F) \approx \frac{1}{z} S^{-1}(B * F) \tag{4}$$

This relation means that the convolution of LR image with the LR PSF can be approximated for small blurs by $1/z$ times the downsampled blurred SR image. Thus, $1/z$ corresponds to a normalization factor. Since we use a normalized convolution mask it is omitted in our computations.

Taking into account (4), we can reformulate the Eqs. (2) and (3). Furthermore, the additional problem of strong aliasing occurs in very LR images. This means that images can not be exactly superimposed on edges and textures in the SR method and a strong motion compensation error $G(k) - [(S^{-1}.B(k)) * (S^{-1}.T(k).M^{i-1})]$ results in such regions. This error amplifies along the iterations and causes spurious artefacts in the SR mosaic. To mitigate these artefacts, we incorporate the regularization operator A [27] in the SR method (3) which penalizes the contribution of edges and textures. The Eqs. (2) and (3) become:

$$F^i = F^{i-1} + T^{-1}.S\left(G - \left[(S^{-1}.B) * \left(S^{-1}.T.F^{i-1}\right)\right]\right) \tag{5}$$

$$M^i = M^{i-1} + \mu(K) \sum_{k=1}^{K} T^{-1}(k).S.A(k)\Big(G(k) - \left[\left(S^{-1}.B(k)\right) * \left(S^{-1}.T(k).M^{i-1}\right)\right]\Big) \tag{6}$$

Note, that T in (5) is not used for motion compensation, but for the estimation of the local blur B as we will describe below. Moreover, there is no need for A in (5) because its computation is based on motion compensation.

In fact, the Eqs. (5) and (6) resemble the steepest descent algorithm [11].

If we solve Eq. (1) in the least-square sense and apply the steepest descent algorithm, there will be a convolution with the transposed PSF B^{T} before S in the Eqs. (5) and (6). Thus, our method is less costly than the steepest descent algorithm as one convolution at high resolution is omitted. Similar to [24] and the steepest descent algorithm, we introduced a gain factor β_1 and β_2 in the Eqs. (5) and (6) respectively.

2.2. Local Blur Estimation

Thanks to moving object segmentation we already know the regions of local blurs contrary to [10]. We assume like in [46] that the blur in the segmented regions can be locally approximated by a spatially invariant blur model. Thus, the shape of the blur operator is unknown beforehand and its parameters can vary in each image. Blur can be due to different reasons such as motions due to the camera or objects, out-of-focus, the imaging system, or compression. Thus, the overall blur corresponds to the sum of the different blurs. Therefore, we consider three different PSFs.

Isotropic Gaussian. It is modelled by a 2D Gaussian function:

$$B_{\mathrm{Gauss2D}}(x, y, \sigma_\alpha) = \frac{1}{2\pi\sigma_\alpha^2} \exp^{-\frac{1}{2}\left(\frac{x^2+y^2}{\sigma_\alpha^2}\right)} \tag{7}$$

where σ_α is the standard deviation in motion direction.

Anisotropic Gaussian in motion direction. The anistropic Gaussian in motion direction can be expressed as [15]:

$$B_{\mathrm{Gauss}}(x, y, \sigma_\alpha, \sigma_{\alpha\perp}, \alpha) = \frac{1}{2\pi\sigma_\alpha\sigma_{\alpha\perp}} \exp^{-\frac{1}{2}\left(\frac{(x\cos\alpha+y\sin\alpha)^2}{\sigma_\alpha^2}+\frac{(-x\sin\alpha+y\cos\alpha)^2}{\sigma_{\alpha\perp}^2}\right)} \tag{8}$$

where $\sigma_{\alpha\perp}$ is the standard deviation in orthogonal motion direction. B_{Gauss} is anisotropic when $\sigma_\alpha \neq \sigma_{\alpha\perp}$.

Linear motion blur. Let v be the velocity of the motion according to the direction $\mathbf{d}$. Then, the PSF in motion direction for linear motion is:

$$B_{\text{box}}(\nu) = \int_{-E/2}^{E/2} \delta(\nu - vt)\,\mathrm{d}t \tag{9}$$

where E is the exposure time and ν is the directional variable.

Let us consider a displacement vector $\mathbf{d}$ in motion direction developed as $\mathbf{d} = \vec{d_x} + \vec{d_y}$ where $\vec{d_x}$ and $\vec{d_x}$ are the components in x-y orthogonal basis. We suppose that this motion is translational. Let us denote by α the angle with the x-axis (see Fig. 1(b)), then the 1D motion blur in $\mathbf{d}$ direction can be represented as a composition of two motion blurs in horizontal and vertical direction:

$$B_{\text{box}}(x, y) = \int_{-E/2}^{E/2} \delta\left(x - v\cos(\alpha)t\right)\mathrm{d}t \cdot \int_{-E/2}^{E/2} \delta\left(y - v\sin(\alpha)t\right)\mathrm{d}t \tag{10}$$

Here $v\cos(\alpha)$ and $v\sin(\alpha)$ respectively represent the horizontal and vertical components, v_x and v_y, of the velocity v (orthogonal projection). Hence:

$$B_{\text{box}}(x, y, b_x, b_y) = \int_{-E/2}^{E/2} \delta\left(x - b_x\right)\mathrm{d}t \cdot \int_{-E/2}^{E/2} \delta\left(y - b_y\right)\mathrm{d}t \tag{11}$$

with b as the size of the blur in motion direction with b_x, b_y as its horizontal and vertical components, and $b_x = v_x t$ and $b_y = v_y t$.

The problem now consists in estimating the parameters $\sigma_\alpha, \sigma_{\alpha\perp}, b_x, b_y$ of Eqs. (7), (8), and (11). Assuming that $b = 3\sigma$ (3σ-property), all blur

models are defined by the blur sizes b, b_x, b_y, and $b_\perp$ which is the blur size in orthogonal motion direction. To estimate these blur sizes, we extend our global blur estimation method [27] which is based on the computation of the edge response in motion direction only on edges orthogonal to the motion direction. Similarly to [47], the motion vector dictates the direction of the blur. For local blur estimation, we distinguish two cases: (i) the blur estimation for the background and (ii) the blur estimation for a moving object. Thus, we propose an adaptation of our blur estimation method [27] for each case.

In order to estimate the blur parameters of the image background, we only take into account the pixels belonging to the scene background obtained from moving object segmentation. We first locate edges by computing the first derivative ∇G of the LR image G using the Sobel operator. Non-maxima suppression and a threshold are applied to the gradient magnitude in order to extract significant edges $\mathcal{E}_\nabla$. To determine the edges which are orthogonal to the motion direction, $\mathcal{E}_{\nabla,\perp}$, we compute the angle between the gradient and the motion vector $\mathbf{d}$ and retain only the pixels for which the angle is smaller than a threshold. The motion vector $\mathbf{d}$ is known from the geometric transformation resulting from motion estimation T and thus no additional computation is needed to determine the direction of the blur.

For each point of $\mathcal{E}_{\nabla,\perp}$ a local estimation of the edge response is accomplished. The edges in a blurred image have the form shown in Figure 1(a). They are limited by a local minimum and a local maximum determining the rise of the edge. Then, the width of the edge response e is defined by the 10% to 90%-distance [42]. Let $\mathbf{m}$ denote the position in the LR image G which

indicates the center of the edge, situated between the local minimum and the local maximum. A discrete straight line segment on a pixel grid according to the direction $\mathbf{d}$ is considered, centered on $\mathbf{m}$. The local maximum and minimum are searched along the line segment in the LR image G. Then, the 10% and 90%-limits can be determined using the same method.

The local edge response in motion direction $e(\mathbf{m})$, $\mathbf{m} \in \mathcal{E}_{\nabla,\perp}$, is computed as the Euclidean distance between the 10% and 90%-limits. Now, its horizontal and vertical components, $e_x(\mathbf{m})$ and $e_y(\mathbf{m})$, can be determined using trigonometric triangle rules. Finally, b, b_x, b_y are, respectively, computed as the average of local values $e(\mathbf{m})$, $e_x(\mathbf{m})$, $e_y(\mathbf{m})$ for all $\mathbf{m} \in \mathcal{E}_{\nabla,\perp}$. We compute $b_\perp$ in the same way than to b, but we use the orthogonal vector of $\mathbf{d}$ in the computations in order to compute the edge response of in orthogonal motion direction on edges in motion direction. The result is a locally constant PSF of the background.

In the case of very LR, objects are typically represented by only few pixels. Therefore, an edge detection inside the object is not reasonable and we directly use the boundary of the segmented region as significant edges $\mathcal{E}_\nabla$. To determine the motion vector $\mathbf{d}$ at each pixel of the object boundary, a rough guess of the motion is sufficient which can be obtained from object tracking. Furthermore, the computations of the edge response and the blur sizes are the same as described for the background.

2.3. Convolution of the ROI

Instead of convolving the ROI with a 1D convolution kernel in motion direction which is complex and costly, e.g. in [47] bilinear interpolation is used to compute the convolution in motion direction, we propose here to compute

a 2D convolution kernel using the horizontal and vertical components of the blur size, b_x and b_y, and compute then a traditional convolution.

We denote $\mathbf{K}$ as the 2D convolution kernel of the size $K_x \times K_y$. In the case of the Gaussian blur model, we determine the kernel size with respect to the 3σ-property:

$$K_x = \lceil 3\sigma_\alpha \rceil + 1 = \lceil b_x \rceil + 1 \qquad K_y = \lceil 3\sigma_{\alpha\perp} \rceil + 1 = \lceil b_y \rceil + 1 \tag{12}$$

where $\sigma_\alpha = \sigma_{\alpha\perp} = \sigma_\alpha$ for the isotropic Gaussian, and $\lceil \cdot \rceil$ is the ceil operator. K_x does not necessarily equal K_y in case of the anisotropic model. Nevertheless, we fix 3×3 as the minimum size of the kernel in both cases. We prefer a slightly larger convolution kernel, therefore also the ceil operator, to avoid a hard cut-off of the convolution kernel. Then, having determined the size of the kernel $\mathbf{K}$, its values $\mathbf{K}(x, y)$ are computed by Eq. (7) for isotropic Gaussian and by Eq. (8) for the anisotropic Gaussian. $\mathbf{K}$ is normalized afterwards.

In case of linear motion blur the kernel is computed as:

$$K_x = \text{round}(b_x) + 1 \qquad K_y = \text{round}(b_y) + 1 \tag{13}$$

$$\mathbf{K}(x, y) = \begin{cases} 0.5 & \text{if round}(b_x) \text{ even and } x = 0 \text{ or } x = K_x - 1 \\ 0.5 & \text{if round}(b_y) \text{ even and } y = 0 \text{ or } y = K_y - 1 \\ 1 & \text{otherwise} \end{cases} \tag{14}$$

Then, $\mathbf{K}$ is normalized. We choose 3×1 or 1×3 as the minimum kernel size depending on whether $b_x > b_y$ or not. It happens that $b_x < 1$ and $b_y < 1$ in our computations since b_x and b_y are, respectively, computed as the average of several estimated values. Thus, fixing the minimal the kernel size to 3×1

or 1×3 for the linear motion blur means that we make the critical assumption that there is a minimal blur of one pixel in horizontal or vertical direction in the ROI. Using this blur model we can not treat blur sizes smaller than one pixel due to the discretisation as the linear motion blur is a constant function. This is different for the Gaussian blur model. As the Gaussian is a continuous declining function, we can handle blur sizes smaller than 1. This only results in a value near 1 in the center of the convolution kernel and in small values near 0 at the borders.

The segmented objects can be of arbitrary form with irregular boundaries. For the convolution on object boundaries, we chose MPEG-4-like padding for boundary macroblocks [40] to extrapolate the object in the region of undetermined pixels underlying the convolution mask.

3. Results

In this section, we show some results obtained using the presented SR method for mosaic construction of compressed video. We evaluate the convergence of our algorithms using quadratic error measures and analyze the results in terms of visual quality, spectrum widening and computational times.

3.1. Mosaic Construction

We used DC images of MPEG-2 compressed streams as LR image sequence. They are a good example of very LR images as they are 8-fold smaller than the original frames. They are strongly aliased, and contain blur due to camera or object motions, and block averaging during compression. For registration, moving object segmentation and tracking, we refer the readers to [27] and [33] respectively. The SR background mosaic is constructed

using (6), some of the foreground objects are restored using (5) and inserted into the SR background mosaic. We used a zoom factor $z = 2$ for all experiments. Then, the resulting mosaic gives an appropriate scene overview for video summarization.

In some cases, additional postprocessing is applied. First, holes can appear in the mosaic due to the exclusion of objects during the blending. As they are usually of small size, we interpolate the lacking pixels from the neighborhood. Furthermore, to remove visual artefacts in the vicinity of background borders, we apply a simple median filtering which removes high frequency noise while preserving edges. Finally, the insertion of an object causes typically seams at the borders of the object. For realistic object insertion, we apply a 3×3 mean filter on the object borders.

3.2. Evaluation of the Proposed Method

The quadratic error measure for the SR background mosaic (6) is:

$$\bar{\epsilon}^i = \frac{1}{K} \sum_{k=1}^{K} \left(\frac{1}{N(k)} \sum_{\mathbf{m}} \left\| G(\mathbf{m}, k) - \left[\left(S^{-1}.B(k) \right) * \left(S^{-1}.T(k).M^{i-1} \right) \right] (\mathbf{m}) \right\|^2 \right) \qquad (15)$$

where $N(k)$ is the number of pixels $\mathbf{m}$ in $G(k)$. We consider color images $G(k)$ and compute here the squared norm of vector difference. We use the error measure (15) additionally as stopping criterion for (6). The maximum number of iterations is achieved when $\bar{\epsilon}$ converges.

Similarly, we define an error criterion for moving objects (see Eq. (5)):

$$\epsilon^i = \frac{1}{N} \sum_{\mathbf{m}} \left\| G(\mathbf{m}) - \left[(S^{-1}.B) * \left(S^{-1}.T.F^{i-1} \right) \right] (\mathbf{m}) \right\|^2 \qquad (16)$$

where N is the number of pixels $\mathbf{m}$ in G.

Since the type of blur is unknown in our LR image sequence and to test the performance of the blur models in our SR method, we compare the PSFs presented in Sect. 2.2. Fig. 2(a) shows the graphs of error measures for the three different PSFs for the restoration of the background mosaic of the sequence "Comportements". One image of the LR sequence is illustrated in (a). The anisotropic Gaussian PSF B_{Gauss} gives the best results according to error measure $\bar{\epsilon}$. The isotropic Gaussian PSF B_{Gauss2D} is close to the B_{Gauss} as we deal only with small blur in this sequence. The result obtained with the linear motion blur PSF B_{box} is not satisfying. This is due to the discretization of the convolution kernel. We made the assumption that at least one pixel blur appears in horizontal or vertical direction. As there is only small blur in this sequence ($b < 1$), the blur model is not appropriate and ringing artefacts appear in the mosaic. For all blur models convergence is achieved after few iterations. The error measure (16) of the object restoration is shown in Fig. 2(b). We observe similar characteristics for the three PSFs.

Fig. A.4 shows some results of mosaic construction for the sequence "Comportements". The corresponding computational times[1] are shown in Tab. 1. The initial mosaic (initial background mosaic combined with the bilinearly interpolated object) before applying the SR method is shown in (b). Its result after applying our method is shown in (c) which is much less blurred (see also the difference image (c)). If we compare the spectrum (b) and (c), we can see that the spectrum has widened after SR. There is only an increase of 8s for the iterations of the SR algorithm. This means that the concatenation of

[1]They were obtained on a 3.2 GHz Intel Core 2 Duo processor using a non optimized C++ code and the VXL image library [50].

motion models in the geometric transform and motion compensation vector field computation is quite expensive. The motion estimation method we used computes a motion model for each I- and P-Frame. However, DC images are extracted at I-Frame basis, so that for each pair of DC images 5 motion models have to be concatenated. The use of another motion estimation method might improve those computational times.

Method	Iterations	Time
(b) Initial mosaic	0	13.34s
(c) Proposed method	19 (+9)	16.36s
(d) Accelerate gradient descent	18 (+19)	18.74s
(e) Upsampled mosaic + deblurring	0	3.84s
(f) Downsampled frames	0	13.98s

Tab. 1: Computational times for the SR mosaics of Fig. A.4.

To improve the computational times of the SR iterations we implemented the gradient descent algorithm for the L2-norm of Eq. (2) for the objects and for the background mosaic similar to [11]. In [5], it was shown that the conjugated gradient method is much slower than the steepest descent method, but the accelerate gradient descent was shown to be two times faster than the steepest descent method. Therefore, we compare our method with an accelerate gradient scheme similar to that one of [5]. The gradient is computed at the initial point and we keep the same gradient while the error functional decreases. The result is shown in (d). Visually there is no difference with respect to our method, Moreover, the computational time is the higher because on the one hand its needs more iterations until convergence and on the other hand it is more complex due to an additional convolution with the transposed PSF and convolution at high resolution.

Additionally, we show in (e) the result for the interpolated mosaic constructed at LR which was deblurred afterwards by a pseudo-inverse filter. We used the isotropic Gaussian PSF and the parameters were estimated from the interpolated mosaic itself [27]. The result is not satisfying as strong ringing artefacts appear on the image border as well on strong edges (also visible in the difference image). This is due to the restoration which causes an amplification of lower and middle frequencies, but a cutoff of high frequencies. The computational time is low as the computation of motion vector fields is achieved at LR, and the computation of regularization operator and iterations of the SR algorithm are omitted.

Furthermore, we show the mosaic constructed from downsampled full-resolution frames in (e). Downsampling was achieved by a Gaussian pyramid. If we compare the resulting mosaic with that one obtained by the proposed method, more less middle and high frequencies are present in the spectrum which lets the mosaic appear less sharper. In the difference image we notice an elevated error of the blue channel in the background which might be due to video compression.

Fig. A.3 shows the result for the sequence "Hiragasy" which contains two moving objects. One image of the LR sequence is illustrated in (a). We tested on this sequences as well the three blur models with similar results than for the previous example. The SR mosaic after 9 iterations for the background and 3 iterations for the objects of our method using the anisotropic Gaussian PSF is shown in (b). Both objects contain more high frequency details than in the LR image.

We used $\beta_1 = 2.5$ and $\beta_2 = 1.5$ for the computation of the results shown

in Figs. A.3 and A.4. This choice seems to be a good compromise between precision and convergence for these images. If β_1, β_2 are chosen to small, convergence is very slow. If they are chosen to high, a high error of (15) and (16) results, and the SR mosaic is blurred.

4. Conclusion and Perspectives

We presented in this article a generic super-resolution method which performs blind restoration of local blurs in spatial domain. Thus, this method allows taking into account moving objects. Based on a segmentation of moving objects, background and moving objects are processed separately. As it is often impossible in very low-resolution image sequence to superimpose moving objects accurately enough for super-resolution, we proposed an interpolation method to improve the resolution of moving objects and a super-resolution method for the scene background. Consequently, we proposed a blur estimation method to estimate local blurs in motion direction.

We tested several blur functions in our restoration scheme. Best results were obtained for the anisotropic Gaussian blur whereas the results for the isotropic Gaussian blur were quite close. In case of small blurs, the linear motion blur does not seem an appropriate blur model due to our discretization of the convolution kernel.

Visual results are quite satisfying. High frequencies could be restored in our experiments for the background as well as moving objects. Computational times are fast, but can still be improved by optimizing the source code. Our super-resolution method assumes that the type of the blur function is known. Thus, future work will focus on the computation of the blur function

from the low-resolution image itself.

Appendix A. Proofs

In this section, we present the mathematical derivation of the spatial domain restoration methods presented in Sect. 2. Both methods are derived from the deconvolution method [9, 24]. They model the blurred image as:

$$\tilde{F}(\gamma) = \int_{-\infty}^{+\infty} F(x)B(\gamma - x)\,\mathrm{d}x \tag{A.1}$$

where $\tilde{F}$ is the blurred image, F the unknown optimal image and B the PSF.

Then, by approximating successively the desired optimal image as:

$$F(\gamma) = \tilde{F}(\gamma) + \Delta(\gamma) \tag{A.2}$$

The following iterative scheme results:

$$F^i(\gamma) = F^{i-1}(\gamma) + \left[\tilde{F}(\gamma) - \int_{-\infty}^{+\infty} F^{i-1}(x)B(\gamma - x)\,\mathrm{d}x\right] \tag{A.3}$$

In the following, we first derive the interpolation method (2) from the deconvolution method [9, 24]. Then, we derive the SR method (3) from the latter. Finally, we derive a relationship between the convolution of a LR image with a LR PSF and the downsampled blurred SR image.

Appendix A.1. Image Interpolation

Here, we demonstrate the derivation of the spatial domain restoration method (2) allowing to increase resolution from the successive approximations (A.2). Therefore, we consider an extended image formation model

relating a SR image with a LR image by incorporating motion and downsampling: SR image $\rightarrow$ motion $\rightarrow$ blur $\rightarrow$ downsampling $\rightarrow$ LR image

Hence, we can rewrite Eqs. (A.1) and (A.2) as:

$$G(y) = S^{-1} \int_{-\infty}^{+\infty} (T.F(x))\, B(\gamma - x)\, \mathrm{d}x \tag{A.4}$$

$$F(\gamma) = T^{-1}\, (S.G(y) + \Delta(\gamma)) \tag{A.5}$$

where y is the position in the observed LR image G, γ is the corresponding position in the SR image F, T is the geometrical transformation from F to the G, T^{-1} is the inverse geometric transformation, S is the upsampling operator, S^{-1} is the downsampling operator, and B the PSF.

Inserting (A.5) in (A.4):

$$G(y) = S^{-1} \int_{-\infty}^{+\infty} (S.G(y))\, B(\gamma - x)\, \mathrm{d}x + S^{-1} \int_{-\infty}^{+\infty} \Delta(x) B(\gamma - x)\, \mathrm{d}x \tag{A.6}$$

Denoting:

$$G^1(y) = S^{-1} \int_{-\infty}^{+\infty} (S.G(y))\, B(\gamma - x)\, \mathrm{d}x \qquad \Delta(\gamma) = \int_{-\infty}^{+\infty} \Delta(x) B(\gamma - x)\, \mathrm{d}x \tag{A.7}$$

Then, (A.6) becomes:

$$G(y) = G^1(y) + S^{-1}.\Delta(\gamma) \quad \Leftrightarrow \quad \Delta(\gamma) = S\left(G(y) - G^1(y)\right) \tag{A.8}$$

Inserting (A.8) in (A.5):

$$F(\gamma) = T^{-1}.S.G(y) + T^{-1}.S\left(G(y) - G^1(y)\right) \tag{A.9}$$

Assuming that:

$$F^0(\gamma) = T^{-1}.S.G(y) \mid T \quad \Leftrightarrow \quad T.F^0(\gamma) = S.G(y) \tag{A.10}$$

Inserting in (A.7):

$$G^1(y) = S^{-1} \int_{-\infty}^{+\infty} \left(T.F^0(x)\right) B(\gamma - x)\,\mathrm{d}x \tag{A.11}$$

Finally, we derive from (A.9):

$$F^i(\gamma) = F^{i-1}(\gamma) + T^{-1}.S \left(G(y) - S^{-1} \int_{-\infty}^{+\infty} \left(T.F^{i-1}(x)\right) B(\gamma - x)\,\mathrm{d}x\right) \tag{A.12}$$

Appendix A.2. Super-resolution

Here, we demonstrate the derivation of the SR method (3) from the restoration method (A.12). Thus, we consider now a sequence of LR images $G_k, 1 \leq k \leq K$ and rewrite (A.12) as:

$$F_k^i(\gamma) = F_k^{i-1}(\gamma) + T_k^{-1}.S \left(G_k(y) - S^{-1} \int_{-\infty}^{+\infty} \left(T_k.F_k^{i-1}(x)\right) B_k(\gamma - x)\,\mathrm{d}x\right) \tag{A.13}$$

where F_k is the kth SR image, G_k is the kth LR image, T_k is the geometrical transformation from F_k to G_k, T_k^{-1} is the inverse geometric transformation, S is the upsampling operator, S^{-1} is the downsampling operator, and B_k the PSF of kth LR image.

For the construction of the mosaic M, we assume:

$$M = \frac{1}{K} \sum_{k=1}^{K} F_k \tag{A.14}$$

Thus (A.13) becomes:

$$\frac{1}{K}\sum_{k=1}^{K} F_j^i(\gamma) = \frac{1}{K}\sum_{j=1}^{K}\left(F_k^{i-1}(\gamma) + T_k^{-1}.S\left(G_k(y) - S^{-1}\int_{-\infty}^{+\infty}\left(T_k.F_k^{i-1}(x)\right)B_k(\gamma - x)\,\mathrm{d}x\right)\right)$$

$$\Leftrightarrow M^i(\gamma) = M^{i-1}(\gamma) + \frac{1}{K}\sum_{j=1}^{K} T_k^{-1}.S\left(G_k(y) - S^{-1}\int_{-\infty}^{+\infty}\left(T_k.F_k^{i-1}(x)\right)B_k(\gamma - x)\,\mathrm{d}x\right) \quad \text{(A.15)}$$

Assuming that F_k is a cut-out of M, then $T_k F_k = T_k M$:

$$M^i(\gamma) = M^{i-1}(\gamma) + \frac{1}{K}\sum_{k=1}^{K} T_k^{-1}.S\left(G_k(y) - S^{-1}\int_{-\infty}^{+\infty}\left(T_k.M^{i-1}(x)\right)B_k(\gamma - x)\,\mathrm{d}x\right) \quad \text{(A.16)}$$

Appendix A.3. Relationship between Convolution at Low and High Resolution

Our objective is to establish the relationship between $S^{-1}(B * F)$ and $(S^{-1}.B) * (S^{-1}.F)$ where $*$ is the convolution operator and S^{-1} is the down-sampling operator by the factor z. Considering the PSF B and the SR image F, then the blurred SR image $\tilde{F}$ is:

$$\tilde{F}(\gamma) = \int_{-\infty}^{+\infty} F(x)B(\gamma - x)\,\mathrm{d}x \quad \text{(A.17)}$$

Denoting B_z and F_z as the subsamples of $B(\gamma)$ and $F(\gamma)$ by the factor z. If we neglect the aliasing effect, then:

$$B_z(\gamma) = B(z\gamma) \qquad F_z(\gamma) = F(z\gamma) \quad \text{(A.18)}$$

Denoting $\tilde{F}_z(\gamma)$ as the result of the convolution $(S^{-1}B) * (S^{-1}F)$:

$$\begin{aligned}\tilde{F}_z(\gamma) &= \int_{-\infty}^{+\infty} F_z(x)B_z(\gamma - x)\,\mathrm{d}x = \int_{-\infty}^{+\infty} F(zx)B(z(\gamma - x))\,\mathrm{d}x \\ &= \int_{-\infty}^{+\infty} F(zx)B(z\gamma - zx)\,\mathrm{d}x = \frac{1}{z}\int_{-\infty}^{+\infty} F(zx)B(z\gamma - zx)\,\mathrm{d}zx \quad \text{(A.19)}\end{aligned}$$

Replacing $y = zx$:

$$\tilde{F}_z(\gamma) = \frac{1}{z}\int_{-\infty}^{+\infty} F(y)B(z\gamma - y))\,\mathrm{d}y = \frac{1}{z}\tilde{F}(z\gamma)$$
$$\Leftrightarrow (S^{-1}.B) * (S^{-1}.F) = \frac{1}{z}S^{-1}(B * F) \tag{A.20}$$

The constant $1/z$ corresponds to a normalization factor and in case of a normalized convolution mask it can be neglected. In reality, Eq. (A.20) is an approximation which only holds for small blurs as we did not consider the spectrum folding in Eq. (A.18).

[1] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In *Proc IEEE Int Conf Comput Vis*, pages 1–8, 2007.

[2] D. Barreto, L. Alvarez, R. Molina, A. Katsaggelos, and G. Callicó. Region-based super-resolution for compression. *Multidimensional Syst Signal Process*, 18(2-3):59–81, 2007.

[3] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *European Conf Computer Vision*, volume 2, pages 573–582, 1996.

[4] M. Ben-Ezra, A. Zomet, and S. Nayar. Video super-resolution using controlled subpixel detector shifts. *IEEE Trans Pattern Anal Mach Intell*, 27(6):977–987, 2005.

[5] J. Benois-Pineau, J. Braquelaire, and A. Ali-Mhammad. Interactive fine object-based segmentation of generic video scenes fro object-based indexing. In *Proc Int Workshop Image Analysis for Multimedia Interactive Services*, pages 200–203, 2003.

[6] N. Bose and N. Ahuja. Superresolution and noise filtering using moving least squares. *IEEE Trans Image Process*, 15(8):2239–2248, 2006.

[7] M. Bronstein, A. Bronstein, M. Zibulevsky, and Y. Zeevi. Blind deconvolution of images using optimal sparse representations. *IEEE Trans Image Process*, 14(6):726–736, 2005.

[8] S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *Proc IEEE Int Conf Comput Vis*, pages 1–8, 2007.

[9] P. V. Cittert. Zum Einfluß der Spaltbreite auf die Intensitätsverteilung in Spektrallinien. II. *Zeitschrift für Physik*, 69:298–308, 1931.

[10] S. Dai, M. Yang, Y. Wu, and A. Katsaggelos. Tracking motion-blurred targets in video. In *Proc IEEE Int Conf Image Proc*, pages 2389–2392, 2006.

[11] M. Elad and Y. Hel-Or. A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur. *IEEE Trans Image Process*, 10:1187–1193, 2001.

[12] P. Eren, M. Sezan, and A. Tekalp. Robust, object-based high resolution image reconstruction from low-resolution video. *IEEE Trans Image Process*, 6(10):1446–1451, 1997.

[13] P. Eren and A. Tekalp. Bi-directional 2-d mesh representation for video object rendering, editing and superresolution in the presence of occlusion. *Signal Processing: Image Communication*, 18(5):321–336, 2003.

[14] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Advances and challenges in super-resolution. *Int J Imaging Syst Technol*, 14(2):47–57, 2004.

[15] J. M. Geusebroek, A. W. M. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. *IEEE Trans Image Process*, 12(8):938–943, 2003.

[16] B. Gunturk, Y. Altunbasak, and R. Mersereau. Multiframe resolution-enhancement methods for compressed video. *IEEE Signal Process Lett*, 9(6):170–174, 2002.

[17] J. Hannuksela, P. Sangi, J. Heikkilä, X. Liu, and D. Doermann. Document image mosaicing with mobile phones. In *Proc Int Conf Image Analysis and Processing*, pages 575–580, 2007.

[18] S. Har-Noy and T. Nguyen. Lcd motion blur reduction: A signal processing approach. *IEEE Trans Image Process*, 17(2):117–125, 2008.

[19] R. Hardie. A fast image super-resolution algorithm using an adaptive wiener filter. *IEEE Trans Image Process*, 16(12):2953–2964, 2007.

[20] H. He and L. Kondi. A regularization framework for joint blur estimation and super-resolution of video sequences. In *Proc IEEE Int Conf Image Proc*, volume III, pages 329–332, 2005.

[21] Y. He, K. Yap, L. Chen, and L. Chau. Blind super-resolution image reconstruction using a maximum a posteriori estimation. In *Proc IEEE Int Conf Image Proc*, pages 1729–1732, 2006.

[22] H. Hu and G. de Haan. Adaptive image restoration based on local robust blur estimation. In *Advanced Concepts for Intelligent Vision Systems*, pages 461–472, 2007.

[23] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *J Vis Commun Image Represent*, 4:324–335, 1993.

[24] P. Jansson. Method for determining the response function of a high-resolution intrared spectrometer. *J Opt Soc Am*, 60(2):184–191, 1970.

[25] J. Jia. Single image motion deblurring using transparency. In *Proc IEEE Int Conf Computer Vision and Pattern Recognotion*, pages 1–8, 2007.

[26] T. Komatsu and T. Saito. Super-resolution sharpening-demosaicking method for removing image blurs caused by an optical low-pass filter. In *Proc IEEE Int Conf Image Proc*, volume I, pages 845–848, 2005.

[27] P. Krämer, O. Hadar, J. Benois-Pineau, and J. Domenger. Super-resolution mosaicing from MPEG compressed video. *Signal Processing: Image Communication*, 22(10):845–865, 2007.

[28] D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE Signal Process Mag*, 13(3):43–64, 1996.

[29] D. Kundur and D. Hatzinakos. Blind image deconvolution revisited. *IEEE Signal Process Mag*, 13(6):61–63, 1996.

[30] M. Kunter and T. Sikora. Super-resolution mosaicing for object based video format conversion. In *Proc Int Workshop Image Analysis for Multimedia Interactive Services*, 2006.

[31] A. Létienne, F. Champagnat, C. Kulcsár, G. Le Besnerais, and P. Viaris De Lesegno. Fast super-resolution on moving objects in video sequences. In *European Signal Processing Conf*, 2008.

[32] X. Li, Y. Hu, X. Gao, D. Tao, and B. Ning. A multi-frame image super-resolution method. *Signal Processing*, 90(2):405–414, 2010.

[33] F. Manerba, J. Benois-Pineau, R. Leonardi, and B. Mansencal. Multiple moving object detection for fast video content description in compressed domain. *EURASIP J Adv Signal Process*, 2008(1):1–13, 2008.

[34] J. Nagy and D. O'Leary. Restoring images degraded by spatially variant blur. *SIAM J Sci Comput*, 19(4):1063–1082, 1998.

[35] N. Nguyen, P. Milanfar, and G. Golub. A computationally efficient image superresolution algorithm. *IEEE Trans Image Process*, 10(4):573–583, 2001.

[36] S. Park, M. Park, and M. Kang. Super-resolution image reconstruction: a technical overview. *IEEE Signal Process Mag*, 20(3):21–36, 2003.

[37] A. Patti, M. Sezan, and A. Tekalp. Robust methods for high quality stills from interlaced video in the presence of dominant motion. *IEEE Trans on Circuits and Systems for Video Technology*, 7(2):328–342, 1997.

[38] A. Patti, M. Sezan, and A. Tekalp. Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. *IEEE Trans Image Process*, 6(8):1064–1078, 1997.

[39] M. Protter, M. Elad, H. Takeda, and P. Milanfar. Generalizing the non-local-means to super-resolution reconstruction. *IEEE Trans Image Process*, 18(1):36–51, 2009.

[40] I. Richardson. *H.264 and MPEG-4 Video Compression*. Wiley, 2003.

[41] C. Segall, R. Molina, and A. Katsaggelos. High-resolution images from low-resolution compressed video. *IEEE Signal Process Mag*, 20(3):37–48, 2003.

[42] S. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.

[43] M. Sorel and J. Flusser. Space-variant restoration of images degraded by camera motion blur. *IEEE Trans Image Process*, 17(2):105–116, 2008.

[44] F. Sroubek, G. Cristóbal, and J. Flusser. A unified approach to superresolution and multichannel blind deconvolution. *IEEE Trans Image Process*, 16(9):2322–2332, 2007.

[45] H. Takeda, P. Milanfar, M. Protter, and M. Elad. Super-resolution without explicit subpixel motion estimation. *IEEE Trans Image Process*, 18(9):1958–1975, 2009.

[46] H. Trussell and S. Fogel. Identification and restoration of spatially variant motion blurs in sequential images. *IEEE Trans Image Process*, 1(1):123–126, 1992.

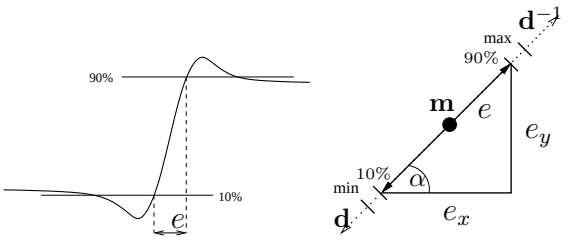
[47] D. Tull and A. Katsaggelos. Iterative restoration of fast-moving objects in dynamic image sequences. *Opt Eng*, 35(12):3460–3469, 1996.

[48] A. van Eekeren, K. Schutte, J. Dijk, D. de Lange, and L. van Vliet. Super-resolution on moving objects and background. In *Proc IEEE Int Conf Image Proc*, pages 2709–2712, 2006.

[49] A. van Eekeren, K. Schutte, and L. van Vliet. Multiframe superresolution reconstruction of small moving objects. *IEEE Trans Image Process*, 19(11), 2010.

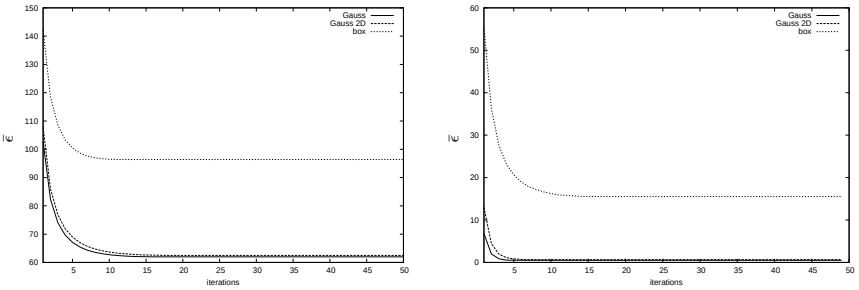
[50] VXL. C++ Libraries for Computer Vision Research and Implementation. `http://vxl.sourceforge.net/`.

[51] L. Yaroslavsky, B. Fishbain, G. Shabat, and I. Ideses. Superresolution in turbulent videos: making profit from damage. *Opt Lett*, 32(20):3038–3040, 2007.



(a) (b)

Fig. A.1: (a) Edge response in motion direction of a blurred edge, (b) Computation of the edge response in horizontal and vertical direction.



(a) (b)

Fig. A.2: The error measure versus the number of iterations for sequence "Comportements": (a) for the background mosaic with $\beta_1 = 1$, (b) for the moving object with $\beta_2 = 1$.

(a) (b)

Fig. A.3: Mosaicing results for the sequence "Hiragasy" (5 LR images): (a) one image of the LR sequence, (b) the SR mosaic. © CERIMES-SFRS

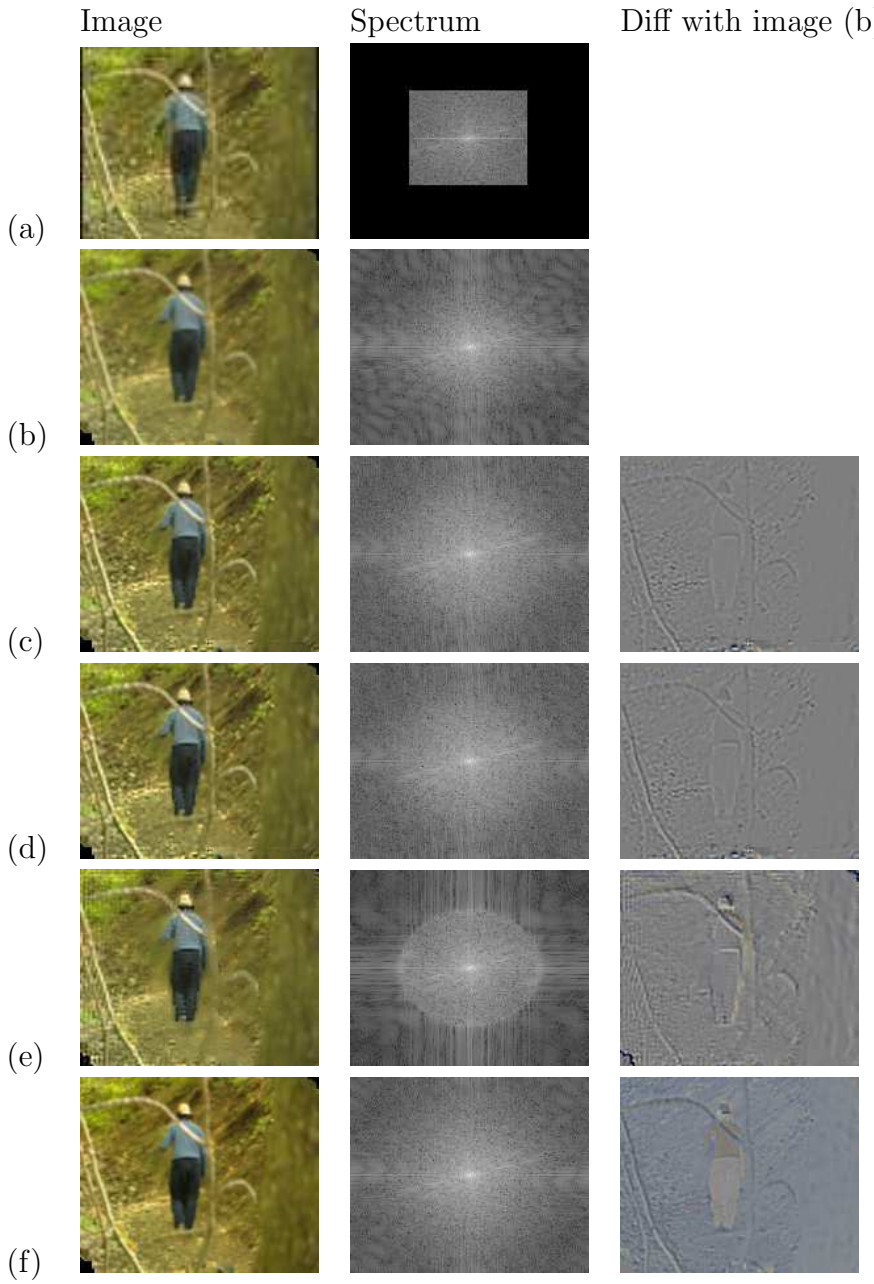


Fig. A.4: Mosaicing results for the sequence "Comportements" (10 LR images): (a) one image of the LR sequence, (b) the initial SR mosaic, (c) the SR mosaic for the proposed method, (d) the SR mosaic for the accelerate gradient descent, (e) the upsampled LR mosaic after deblurring, (f) the mosaic constructed from downsampled full-resolution frames. © CERIMES-SFRS