Video Temporal Super-Resolution Using Nonlocal Registration and Self-Similarity

Matteo Maggioni and Pier Luigi Dragotti

Communications and Signal Processing Group
Department of Electrical and Electronic Engineering
Imperial College London, London SW7 2AZ, UK

Email: m.maggioni@imperial.ac.uk, p.dragotti@imperial.ac.uk

***Abstract*—In this paper we present a novel temporal super-resolution method for increasing the frame-rate of single videos. The proposed algorithm is based on motion-compensated 3-D patches, i.e., a sequence of 2-D blocks following a given motion trajectory. The trajectories are computed through a coarse-to-fine motion estimation strategy embedding a regularized block-wise distance metric that takes into account the coherence of neighbouring motion vectors. Our algorithm comprises two stages. In the first stage, a nonlocal search procedure is used to find a set of 3-D patches (targets) similar to a given patch (reference), subsequently all targets are registered at sub-pixel precision with respect to the reference in an upsampled 3-D FFT domain, and finally all registered patches are aggregated at their appropriate locations in the high-resolution video. The second stage is used to further improve the estimation quality by correcting each 3-D patch of the video obtained from the first stage with a linear operator learned from the self-similarity of patches at a lower temporal scale. Our experimental evaluation on color videos shows that the proposed approach achieves high quality super-resolution results from both an objective and subjective point of view.**

I. Introduction

Digital video acquisition devices necessarily have a spatial and temporal resolutions which are limited by the size of the sensor, optics, point-spread function (PSF), and exposure time. Thus, solutions able to increase the resolution of the camera become critical whenever high-quality sensors are either not existent or prohibitively expensive to use, e.g., in forensic or surveillance imaging.

Super-resolution (SR) is a numerically ill-posed inverse problem and thus remains a challenging topic despite the vast literature existing on the subject [1], [2]. Common approaches for video SR first warp and then fuse adjacent low-resolution (LR) frames at sub-pixel precision in high-resolution (HR) space, and finally deconvolve the final fused data [3]–[5]. Other popular strategies embed example-based techniques using external databases [6] or multiple video recordings [7]. Recently, effective single-video SR has been achieved by leveraging self-similarity of small patches at different scales of the LR input video [8]–[10] with multi-scale learning [11]–[13] as well as modern sampling theory [14], [15].

In this paper, we focus on the problem of temporal SR from a single video, i.e., effectively incrementing the frame-rate of the input sequence. The proposed method harnesses the power of self-similarity, which has been proven to be abundant in both space [11] and space-time [10], in combination with the nonlocality principle [16] that mutually similar local features can be found at different location within the data.

Our proposed method comprises two main stages. Each stage begins by estimating all motion vectors in the video using a coarse-to-fine approach with a distance metric embedding a regularization prior on the position and gradient of the motion vectors, designed to enhance both the accuracy and coherence of the motion field. During the first stage, motion-compensated 3-D (reference) patches are first extracted from the video by stacking together 2-D blocks following a motion trajectory (i.e., a concatenation of motion vectors), and then matched against other 3-D (target) patches at different spatio-temporal locations. Subsequently, the targets are registered with respect to the reference at sub-pixel precision in an upsampled 3-D Fourier domain [17], [18]. Finally, the resulting patches are returned and aggregated in the appropriate locations within the HR video. The second stage further improves the HR video by alleviating the registration artifacts with the application of an error-correcting linear operator [13] learned from a pair of videos at an intermediate temporal scale, in a way similar to [15]. The proposed SR algorithm can be used for both grayscale and color videos. Preliminary experimental results show the effectiveness of the proposed method both from a subjective and an objective point of view on standard test sequences as well as real videos.

The remainder of the paper is organized as follows. Section II formally describes all the building blocks of the proposed methods, namely motion estimation (Section II-A), 3-D patch registration (Section II-B), and error correction (Section II-C). Then, Section IV reports our experimental evaluation, and finally Section V presents the final remarks.

II. Temporal Super-Resolution Algorithm

This section formally describes the proposed SR method. Let us denote a LR video as

$$z(\mathbf{x}, t) = \big(y_f \circledast \phi\big)_{\downarrow f}(\mathbf{x}, t), \tag{1}$$

where $\mathbf{x} \in X \subset \mathbb{Z}^2$ and $t \in T \subset \mathbb{Z}$ are the 2-D spatial and 1-D temporal coordinates specifying a position in the LR video, $\circledast$ denotes convolution, $\downarrow f$ denotes a decimation of factor $f > 1$, and y_f is the underlying unknown HR video which is convolved by a blurring kernel ϕ. This kernel in general models both the PSF of the camera and the camera exposure time [10], but, for our purposes, we can assume it to be a 1-D rectangular kernel, with support depending on the exposure time, acting solely along the temporal dimension. The goal is to find an estimate of the temporal HR video y_f from the observed z.

A. Motion Estimation

Let B_i be a 2-D $N_1 \times N_2$ block extracted at the coordinate $(\mathbf{x}_i, t_i)$, being $\mathbf{x}_i$ its top-left corner. For the sake of notation simplicity, in what follows, if not specified otherwise, we will use the subscripts "i" to denote a coordinate $(\mathbf{x}_i, t_i)$, and "i, j" to denote a pair of coordinates $(\mathbf{x}_i, t_i)$ and $(\mathbf{x}_j, t_j)$.

The first step consists in estimating the motion field. We use a coarse-to-fine motion-estimation strategy where the motion vectors are iteratively improved from those obtained at a lower scale. In particular, for a given reference block B_R at a given scale, we look for the position $\mathbf{x}_T$ of most-similar block B_T in frame $t_T = t_R \pm 1$ within a window of size $W_{2\text{D}} \times W_{2\text{D}}$ centered around $\mathbf{x}_R$. As usual, the corresponding motion vector is $\vec{\mathbf{v}}_{R,T} = \mathbf{x}_T - \mathbf{x}_R$.

The distance between two blocks is hereby defined as

$$\begin{aligned} d_{2\text{D}}\big(B_R, B_T\big) =& \delta_1 \big|\big|B_R - B_T\big|\big|_2 + \delta_2 \big|\big|\hat{\mathbf{x}}_T - \mathbf{x}_T\big|\big|_2 + \\ & \delta_3 \big|\big|\angle\vec{\mathbf{v}}_{R,T} - \text{median}_{\mathbf{x}_i \in \mathcal{N}_R}\left(\angle\vec{\mathbf{v}}_{R,i}\right)\big|\big|_2 \end{aligned} \quad (2)$$

where $||\cdot||_2$ denotes a *normalized* ℓ_2-norm, $\hat{\mathbf{x}}_T$ is the predicted position of the most-similar block estimated from the corresponding motion vector at a lower scale, $\angle$ denotes the direction of the vector, and the weights $\delta_\cdot$ define a convex combination (thus (2) always yields a value in $[0, 1]$). The third term is the direction discrepancy between the direction of the current $\vec{\mathbf{v}}_{R,T}$ and the median direction within a local neighbourhood $\mathcal{N}_R$ of size 3×3 centered around $\mathbf{x}_R$ at a lower scale (whenever this is available).

Once all the correspondences between each pair of adjacent frames are computed, it is straightforward to extract a trajectory of arbitrary length starting from any given coordinate by iteratively concatenating motion vectors. Observe that a trajectory can be stopped at any time (i.e., when no match exists in the target frame) if (2) exceeds a predefined threshold $\tau_{2\text{D}} \in [0, 1]$.

B. Nonlocal 3-D Patch Registration

Let P_R be a $N_1 \times N_2 \times N_3$ motion-compensated 3-D patch composed by a sequence of 2-D blocks extracted from the video following a trajectory of length $N_3 \in \mathbb{N}$ originating from the location $(\mathbf{x}_R, t_R)$; analogously to the 2-D case, the coordinate $(\mathbf{x}_R, t_R)$ identifies the top-left-front voxel of the 3-D patch. Now we are able to define a patch-distance metric

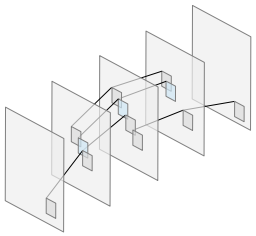

Fig. 1. Nonlocal target patches (gray) similar to a reference patch (blue).

as a *normalized* ℓ_2-norm of the difference of corresponding voxels in two different patches as

$$d_{3\text{D}}\big(P_R, P_T\big) = \big|\big|P_R - P_T\big|\big|_2, \quad (3)$$

and we call two patches similar if their distance is smaller than another predefined threshold $\tau_{3\text{D}} \in [0, 1]$.

An important step of the registration algorithm consists in finding a set of similar patches (targets) within the video, which can be interpreted to be examples of the same feature acquired at different (sub-pixel) spatio-temporal positions. Given a reference 3-D patch P_R, we construct the set

$$S_R = \big\{(\mathbf{x}_T, t_T) \;\big|\; d_{3\text{D}}\big(P_R, P_T\big) \leq \tau_{3\text{D}}\big\}, \quad (4)$$

containing the coordinates of the mutually similar patches within the video. Note that, again for computational constraints, the nonlocal search (4) is restricted within a 3-D search window of size $W_{3\text{D}} \times W_{3\text{D}} \times W_{3\text{D}}$ centered around the reference coordinate $(\mathbf{x}_R, t_R)$. Fig. 1 illustrates an example of mutually similar patches, note how the targets (in gray) have different trajectories, and are located at nonlocal positions in both space and time. We restrict the cardinality of (4) to be at most equal to a predefined $K_1 \in \mathbb{N}$. The actual number of can be smaller than K_1 when not enough targets satisfy the similarity threshold $\tau_{3\text{D}}$, however $K_1 \geq 1$ because (4) always contains the coordinates of the reference patch.

The registration is performed by first placing the reference P_R onto a HR grid, and then aggregating all patches in the corresponding (4) at sub-pixel precision. Let us call $\boldsymbol{\xi}^f_{R,T}$ and $\rho^f_{R,T}$ the spatial and temporal sub-pixel translations obtained from the registration of the target P_T with respect to the reference P_R with a SR factor f. The translations are classically computed by localizing the maximum value of the 3-D patch cross-correlation, which can be implemented as a pixel-wise multiplication in an upsampled 3-D Fourier domain [17].

Algorithm 1 summarizes the main steps of a general 3-D registration process, which is also illustrated in Fig. 2. Note that, as the SR factor increases ($f \gg 2$), the computation of the upsampled 3-D Fourier transform becomes quickly prohibitive, therefore alternative fast algorithms can be used [18]. Additionally, if only the temporal translation is of interest, the upsampling can be performed solely along the third (temporal) dimension, thus yielding a sub-frame precision temporal translation $\rho^f_{R,T}$ and a pixel-precision spatial translation $\boldsymbol{\xi}^f_{R,T}$.

Adjacent reference patches are typically overlapping and, in addition to that, after the registration different patches in

$W_R = \text{FFT}(P_R)$
foreach $P_T \in S_R$
  $W_T = \text{FFT}(P_T)$
  $W_{CC} = W_R \cdot \overline{W_T}$
  $CC = \text{FFT}_f^{-1}(W_{CC})$
  $(\mathbf{x}_{\max}, t_{\max}) = \text{argmax}_{(\mathbf{x}_i, t_i) \in \chi_{CC}} CC(\mathbf{x}_i, t_i)$
  $\boldsymbol{\xi}_{R,T}^f = \mathbf{x}_{\max} - 1$
  if $\mathbf{x}_{\max} > \text{round}([N_1, N_2]/2)$ **then**
    $\boldsymbol{\xi}_{R,T}^f = \boldsymbol{\xi}_{R,T}^f - [N_1, N_2]$
  end
  $\rho_{R,T}^f = t_{\max} - 1$
  if $t_{\max} > \text{round}(N_3/2)$ **then**
    $\rho_{R,T}^f = \rho_{R,T}^f - N_3$
  end
end

Algorithm 1. Registration algorithm: P_R and P_T are 3-D patches of size $N_1 \times N_2 \times N_3$, f is the SR factor, FFT is the fast Fourier transform, FFT_f^{-1} is the upsampled inverse FFT, $\overline{W_T}$ is the complex conjugate of W_T, and CC is the cross-correlation. Finally, $\rho_{R,T}^f \in \mathbb{R}$ and $\boldsymbol{\xi}_{R,T}^f \in \mathbb{R}^2$ are the temporal and spatial (sub-pixel) translations given with respect to the HR grid.

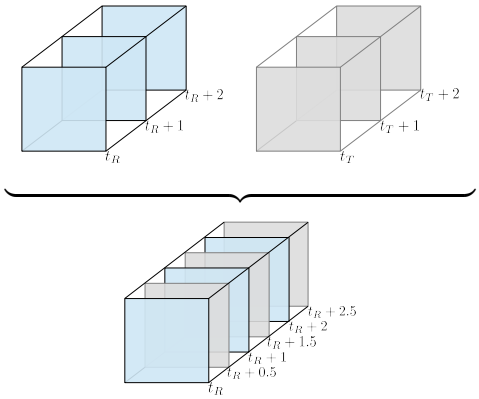

Fig. 2. Registration of two patches with SR factor $f = 2$ (the blue one being the reference) with translations $\boldsymbol{\xi}_{R,T}^f = \mathbf{0}$ and $\rho_{R,T}^f = 1$.

(4) can be registered at overlapping (sub-pixel) positions. This overcompleteness is generally helpful in increasing the quality of the final estimate, but a strategy to aggregate different patches within the overlapping regions is needed. We use a convex combination with weights proportional to the similarity (3) defined as

$$w_{R,T} = e^{\gamma \cdot \left(d_{3\text{D}}\left(P_R, P_T\right)\right)^2}, \tag{5}$$

where $\gamma \in \mathbb{R}^-$ is a negative scaling parameter, which maximizes (5) as the distance decreases. The complete aggregation process can be formalized as

$$\hat{y}_f = \frac{\sum_{(\mathbf{x}_R, t_R) \in X \times T} \sum_{(\mathbf{x}_T, t_T) \in S_R} w_{R,T} \cdot \mathring{P}_T^f}{\sum_{(\mathbf{x}_R, t_R) \in X \times T} \sum_{(\mathbf{x}_T, t_T) \in S_R} w_{R,T} \cdot \chi_{\mathring{P}_T^f}}, \tag{6}$$

where f is the SR factor, $\mathring{P}_T^f$ is the registered LR patch P_T after applying the sub-pixel translations obtained as described in Algorithm 1, and $\chi_{\mathring{P}_T^f}$ is the characteristic function of the support of $\mathring{P}_T^f$ into the HR video $\hat{y}_f$. Intuitively, within each constellation of registered patches, (6) gives more importance to those more similar to the reference one. An additional 3-D Kaiser window can be applies to alleviate blocking artifacts at the patch boundaries.

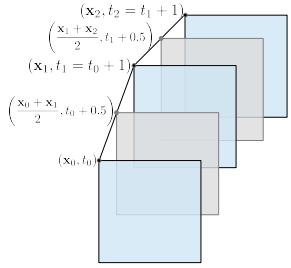

Fig. 3. Linear interpolation of a motion trajectory with SR factor $f = 2$. The blue blocks belongs to the original LR 3-D patch.

At this stage, depending on the content of the LR video z, the registered estimate (6) is likely to be incomplete. Specifically, when all targets in (4) yield null sub-pixel translations there will be gaps in the HR video. In practice, this can happen when the reference and the targets are almost-perfectly identical (e.g., when the patches are extracted in uniform regions in the video), and thus there is no variance allowing the patches to be registered at sub-pixel precision. Thus, we estimate the missing values in (6) through a block-based linear interpolation along the motion field as visualized in Fig. 3. Firstly, we select all the (overlapping) 3-D patches that are (even partially) in contact with any of the missing regions, then we linearly interpolate the 2-D blocks within the patch at the desired sub-pixel precision, and finally we take the average of the (overcomplete) interpolated values. We argue that this is a viable strategy because the uniform nature of the data within each 3-D patch makes the smoothness prior of the interpolating model a reasonable assumption. Note that we also interpolate the trajectory coordinates via linear interpolation to estimate the location of the blocks at sub-frame precision.

C. Error Correction by Self-Similarity

The second stage of the algorithm aims at improving the quality of the estimated video (6), as this inevitably contains errors caused by, e.g., imperfections in the the registration. Thus, we apply an error-correction linear operator to every 3-D patch in the video learned from an appropriately defined (internal) dictionary of mutually similar 3-D patches extracted at a different temporal scale.

Standard SR approaches based on patch self-similarity recursively refine the HR estimate using a coarse-to-fine pyramid composed of different scales of the image [11]. Recently, [13] proposed the use of a double pyramid in which the HR image is recursively estimated by learning linear mapping functions from similar patches at a lower scale (provided that the downscaling is small). Following the same rationale, and inspired by the work on image SR introduced in [14], [15], we use an inverse double-pyramid approach, in which we first

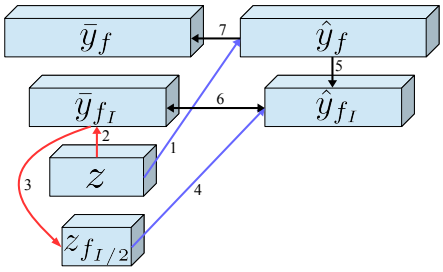

Fig. 4. Schematic visualization of the error correction via self-learned linear mapping. The red arrows denote resampling by temporal interpolation, the blue arrow denote upsampling by registration (Section II-B), and the black arrows denote the linear mapping. See text for details.

learn the linear mapping between 3-D patches of the "ground-truth" LR video and the corresponding HR estimate at an intermediate scale $1 < f_I < f$, and then we apply the same mapping at the higher level f to correct the corresponding data.

In what follows we will focus on case of SR factor $f = 2$, but observe that the same procedure can be iterated to account for larger upsampling factors. Fig. 4 illustrates the complete error-correction process, where each step is numbered according to the explanation detailed below.

Step 1-4: Once the HR estimate $\hat{y}_f$ is available (step 1), we construct a ground-truth y_{f_I} and HR estimate $\hat{y}_{f_I}$ at a chosen intermediate scale f_I. Since $f_I > 1$, we cannot access to the ground-truth, thus we estimate y_{f_I} by upsampling z using the overcomplete block-wise linear interpolation of motion-compensated 3-D patches and then we average all the interpolated results as detailed in Section II-B. Let us call this approximation $\bar{y}_{f_I}$ (step 2). Then, $\bar{y}_{f_I}$ is downsampled by a factor 0.5 with an analogous motion-compensated strategy (step 3) and finally the first-stage registration algorithm is applied with a SR factor 2 on the obtained $z_{f_{I/2}}$ to construct the intermediate estimate $\hat{y}_{f_I}$ (step 4).

Step 5: Once the pair of videos at the intermediate scale is available, for each 3-D patch P_R in $\hat{y}_f$ we search for the $K_2 \geq 1 \in \mathbb{N}$ most similar patches in the intermediate-scale video $\hat{y}_{f_I}$. The nonlocal search is restricted within a 3-D window as in (4) centered around $\left(f_I^{-1}\mathbf{x}_R, f_I^{-1}t_R\right)$, i.e. the position corresponding to $(\mathbf{x}_R, t_R)$ at the intermediate scale f_I.

Step 6: Let us denote as $\hat{\mathbf{p}}_R$ the vectorization of a 3-D patch P_R extracted from $\hat{y}_{f_I}$ having size $D = N_1N_2N_3$, and as $\hat{\mathbf{P}}_R \in \mathbb{R}^{D\times K_2}$ the matrix having as columns the K_2 vectorized patches. An equivalent $\bar{\mathbf{P}}_R$ can be constructed by vectorizing the patches in $\bar{y}_{f_I}$. The goal of the error-correction algorithm is to calculate a linear transformation $\mathbf{M}_R \in \mathbb{R}^{D\times D}$ that maps the dictionary of mutually similar K_2 patches in $\hat{y}_{f_I}$ to their corresponding "ground-truth" versions in $\bar{y}_{f_I}$. This can be solved by minimizing a constrained Tikhonov regularization problem [13], which admits the closed-form solution

$$\mathbf{M}_R = \bar{\mathbf{P}}_R\hat{\mathbf{P}}_R^\top\left(\hat{\mathbf{P}}_R\hat{\mathbf{P}}_R^\top + \lambda\mathbf{I}\right)^{-1}, \qquad (7)$$

TABLE I
SETTINGS OF ALL PARAMETERS IN THE PROPOSED SR ALGORITHM.

Section II-A					Section II-B							Section II-C		
δ_1	δ_2	δ_3	W_{2D}	τ_{2D}	N_1	N_2	N_3	τ_{3D}	K_1	W_{3D}	γ	f_I	K_2	λ
.85	.1	.05	15	.15	7	7	3	.3	8	9	-.5	1.625	8	.1

where the superscript $\top$ denotes transposition, $\lambda \in \mathbb{R}^+$ is a regularization parameter and $\mathbf{I} \in \mathbb{R}^{D\times D}$ is the identity matrix.

Step 7: Finally, the operator (7) learned at the intermediate level f_I can be applied to the corresponding reference patch at the HR level f, and then the corrected patch can be returned to its original location. The overlapping parts are averaged in a fashion similar to (6) –this time using unitary weights– eventually obtaining the final corrected HR video $\bar{y}_f$.

III. Color Processing

The proposed method can be also extended to color (RGB) video processing by first transforming the video from RGB space to a luminance-chrominance (YUV) space, then our SR algorithm is applied to luminance channel whereas the two chrominance channels are upsampled using our motion-compensated temporal interpolation using the motion field calculated within the luminance.

IV. Experiments

To the best of our knowledge, despite the large amount of literature on image and video SR[1], there seems to be a lack of easily available software in the context of temporal SR, thus we only compare the proposed SR algorithm against common interpolation methods. Our test data will be a set of both standard[2] and real[3] video sequences.

A. Algorithm Parameters

The parameters have been set based on a empirical optimization, which resulted in reasonably good SR performances for all the tested sequences. Table I summarizes all the parameters involved in the proposed algorithm grouped using a reference to the corresponding section. Observe that in the following experiments, we will use maximum overlap between adjacent reference patches.

B. Test Videos

For our objective evaluation, we use the eight standard test sequences referred in Table II. We design these experiments by first decimating each sequence along time by a factor 2 (i.e., we remove one every two frames), and then resolving the missing data using different SR methods. Finally, we compute the peak signal-to-noise ratio (PSNR) and the SSIM index [19] of the reconstructed frames.

In Table II we report the objective performances of standard bicubic interpolation (first column in the table), our block-wise linear interpolation along the motion trajectories

[1]http://reproducibleresearch.net/super-resolution/
[2]https://media.xiph.org/video/derf/
[3]http://www.wisdom.weizmann.ac.il/ vision/SingleVideoSR.html

Fig. 5. From top to bottom. Reconstruction of the original frame (left column) in *Bus*, *City*, *Coastguard*, *Foreman*, and *Tennis* using bicubic interpolation (middle column) and proposed method (right column).

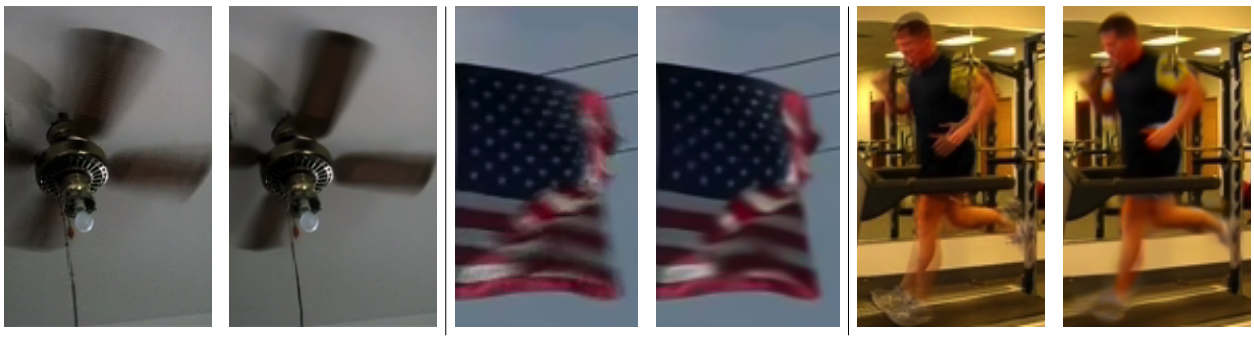

Fig. 6. Super-resolution of *Fan* (left), *Flag* (middle) and *Treadmill* (right) using bicubic interpolation (left in each pair) and proposed method (right).

(second column), the estimate $\hat{y}_f$ obtained after registration (third column), and the estimate $\bar{y}_f$ obtained by the proposed algorithm (fourth column). As one can see, the proposed method almost always achieves the best performances in terms of both PSNR and SSIM. Interestingly, the PSNR favours bicubic interpolation for *Miss America* but, we stress, this is an essentially motion-less video. We also note that the nonlocal registration method is often outperformed by our overcomplete temporal interpolation strategy; we explain this phenomena from the content of the tested sequences which exhibits little to no temporal artifacts (e.g., motion blur) which would negatively affect the temporal interpolation model.

The subjective results shown in Fig. 5 attest the extremely good performances of the proposed SR algorithm. We highlight quality of the fine details in the fence and trees in *Bus*, the rocks in *Coastguard*, the ball and paddle in *Tennis*, the face of *Foreman*, and the buildings in *City*. On the other hand, we sometimes observe excessive smoothing around the moving features in the video, such as in the background around the hands of the player in *Tennis*.

TABLE II
PERFORMANCES IN TERMS OF PSNR (DB, LEFT VALUE IN EACH CELL) AND SSIM [19] (RIGHT VALUE) CALCULATED ON THE RECONSTRUCTED FRAMES USING A SR FACTOR EQUAL TO 2.

	Bic. int.		**Temp. int.**		$\hat{\mathbf{y}}_\mathbf{f}$		$\bar{\mathbf{y}}_\mathbf{f}$	
Bus	17.10	0.4723	21.74	**0.8193**	21.20	0.7716	**21.85**	0.7854
City	23.58	0.6254	27.57	0.8622	27.52	0.8556	**28.93**	**0.8905**
Coastg.	25.84	0.8050	30.57	0.9327	29.44	0.9181	**31.21**	**0.9402**
Forem.	29.16	0.9027	30.71	0.9221	30.55	0.9200	**31.86**	**0.9349**
Fl. Gard.	16.11	0.5841	22.03	0.8844	20.96	0.8461	**23.02**	**0.8990**
Miss Am.	**37.18**	0.9157	36.66	0.9172	36.41	0.9172	36.93	**0.9191**
Salesm.	37.46	0.9822	37.59	0.9829	36.52	0.9794	**38.44**	**0.9846**
Tennis	20.93	0.6612	24.96	0.8252	24.76	0.8019	**25.72**	**0.8441**

C. Real Videos

For these experiments we use the real videos *Fan*, *Flag* and *Treadmill* originally presented in [10]. In this case no decimation is performed, and thus new frames are effectively created in the super-resolved video. As can be seen from Fig. 6, the proposed method is able to reduce the motion blur in the reconstructed frames, such as the ripples in *Flag*, but we note a degradation of performances when the motion blur becomes severe, e.g., around the blades of *Fan* or the feet of *Treadmill*.

D. Computational Complexity

The current single-thread MATLAB/C++ implementation of the proposed algorithm, running on a Intel(R) Core(TM) i7-3770 3.40-GHz with 8GB RAM, processes between 500 and 600 3-D patches per second. Therefore, depending on the chosen overlapping between adjacent patches, one CIF-resolution frame (352×240 pixels) can take between 30 seconds to 5 minutes to be resolved. However, since the Fourier transform has been implemented as a linear operation, this complexity can be greatly reduced by simply using the FFT algorithm.

V. CONCLUSIONS

We have presented an effective temporal super-resolution (SR) algorithm for both grayscale and RGB videos. The foundation of the proposed algorithm is a robust coarse-to-fine motion-estimation strategy embedding a regularized block-wise distance metric which takes into account both photometric similarity and coherence of neighbouring motion vectors. Temporal SR is then achieved by first extracting 3-D patches along the motion trajectories of the video, and then registering mutually similar patches at sub-pixel precision in 3-D Fourier domain. Self-similarity at an intermediate scale is finally leveraged to further improve the SR quality of the registered video estimate by applying an error-correcting linear mapping to each 3-D patch. Experimental results on both test sequences and real videos showed promising performances from an objective (PSNR and SSIM [19]) as well as subjective point of view.

As future works, we target to improve the motion estimation and the registration by including more sophisticated priors in the distances (2) and (3). In particular, (3) could favour target patches that also exhibit a sub-pixel translation with respect to the reference one. Additionally, we argue that extending the proposed method to a pyramidal approach, where the reconstructed estimate is iteratively improved at each scale, would greatly benefit the performances in cases of extreme motion blur. Finally, it is interesting to see how the method generalizes to the problem of joint spatio-temporal super-resolution.

VI. ACKNOWLEDGEMENTS

This work has been supported by European Research Council (ERC) starting investigator award Nr. 277800 (RecoSamp).

REFERENCES

[1] S. Borman and R. L. Stevenson, "Super-resolution from image sequences – A review," in *MWSCAS*, Aug. 1998, pp. 374–378.

[2] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Advances and challenges in super-resolution," *International Journal of Imaging Systems and Technology*, vol. 14, no. 2, pp. 47–57, 2004.

[3] T. S. Huang and R. Y. Tsay, "Multiple frame image restoration and registration," in *Advances in Computer Vision and Image Processing*, Greenwich, 1984, vol. 1, pp. 317–339, JAI.

[4] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP Graphical Models and Image Processing*, vol. 53, no. 3, pp. 231–239, Apr. 1991.

[5] Taehyeun Ha, Seongjoo Lee, and Jaeseok Kim, "Motion compensated frame interpolation by new block-based motion estimation algorithm," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 752–759, May 2004.

[6] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, Mar. 2002.

[7] E. Shechtman, Y. Caspi, and M. Irani, "Space-time super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 531–545, Apr. 2005.

[8] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Transactions on Graphics*, vol. 30, no. 2, pp. 12:1–12:11, Apr. 2011.

[9] M. Shimano, T. Okabe, I. Sato, and Y. Sato, *Computer Vision – ACCV 2010, Revised Selected Papers, Part I*, chapter Video Temporal Super-Resolution Based on Self-similarity, pp. 93–106, Springer, Berlin, Heidelberg, 2011.

[10] O. Shahar, A. Faktor, and M. Irani, "Space-time super-resolution from a single video," in *CVPR*, Jun. 2011, pp. 3353–3360.

[11] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *ICCV*, Sep. 2009, pp. 349–356.

[12] K. Zhang, X. Gao, D. Tao, and X. Li, "Single image super-resolution with multiscale similarity learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 10, pp. 1648–1659, Oct. 2013.

[13] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi Morel, "Single-image super-resolution via linear mapping of interpolated self-examples," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5334–5347, Dec. 2014.

[14] X. Wei and P. L. Dragotti, "Sampling piecewise smooth signals and its application to image up-sampling," in *ICIP*, Sep. 2015, pp. 4293–4297.

[15] X. Wei and P. L. Dragotti, "FRESH – FRI-based single-image super-resolution algorithm," to appear in *IEEE Transactions on Image Processing*, 2016.

[16] J. S. De Bonet, "Noise reduction through detection of signal redundancy," Tech. Rep., Rethinking Artificial Intelligence, MIT AI Lab, 1997.

[17] B. S. Reddy and B. N. Chatterji, "An FFT-based technique for translation, rotation, and scale-invariant image registration," *IEEE Transactions on Image Processing*, vol. 5, no. 8, pp. 1266–1271, Aug. 1996.

[18] M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup, "Efficient subpixel image registration algorithms," *Optical Letters*, vol. 33, no. 2, pp. 156–158, Jan. 2008.

[19] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.