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

$$z(x,t) = (y_f \otimes \phi)_{\downarrow f}(x,t),$$

(1)
where \( x \in X \subset \mathbb{Z}^2 \) and \( t \in T \subset \mathbb{Z} \) are the 2-D spatial and 1-D temporal coordinates specifying a position in the LR video, \( \otimes \) denotes convolution, \( \downarrow \) denotes a decimation of factor \( f > 1 \), and \( y_f \) is the underlying unknown HR video which is convolved by a blurring kernel \( \phi \). 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 \( N_i \times N_i \) block extracted at the coordinate \((x_i, t_i)\), being \( x_i \) 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 \((x_i, t_i)\), and “\( i, j \)” to denote a pair of coordinates \((x_i, t_i)\) and \((x_j, t_j)\).

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 \( t' = t_R \pm 1 \) within a window of size \( W_{2D} \times W_{2D} \) centered around \( x_R \). As usual, the corresponding motion vector is \( \mathbf{v}_{R,T} = \mathbf{x}_T - \mathbf{x}_R \).

The distance between two blocks is hereby defined as

\[
d_{2D}(B_R, B_T) = \delta_1 \left| \left| B_R - B_T \right| \right|_2 + \delta_2 \left| \left| \mathbf{x}_T - \mathbf{x}_R \right| \right|_2 + \delta_3 \left| \left| \mathbf{v}_{R,T} - \text{median}_{\mathbf{x} \in N_R} \left( \mathbf{v}_{R,i} \right) \right| \right|_2
\]

(2)

where \( \left| \left| \cdot \right| \right|_2 \) denotes a normalized \( \ell_2 \)-norm, \( \mathbf{x}_T \) is the predicted position of the most-similar block estimated from the corresponding motion vector at a lower scale, \( \mathbf{v} \) denotes the direction of the vector, and the weights \( \delta \) define a convex combination (thus (2) always yields a value in \([0, 1]\)). The third term is the direction discrepancy defined as the minimum of the direction of the current \( \mathbf{v}_{R,T} \) and the median direction within a local neighbourhood \( N_R \) of size \( 3 \times 3 \) centered around \( x_R \) 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 \( \tau_{2D} \in [0, 1] \).

### 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 \( (x_R, t_R) \); analogously to the 2-D case, the coordinate \( (x_R, t_R) \) identifies the top-left-front voxel of the 3-D patch. Now we are able to define a patch-distance metric as a normalized \( \ell_2 \)-norm of the difference of corresponding voxels in two different patches as

\[
d_{3D}(P_R, P_T) = \left| \left| P_R - P_T \right| \right|_2
\]

(3)

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

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

\[
S_R = \{ (\mathbf{x}_T, t_T) \mid d_{3D}(P_R, P_T) \leq \tau_{3D} \}
\]

(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 \( W_{3D} \times W_{3D} \times W_{3D} \) centered around the reference coordinate \( (x_R, t_R) \). Fig. 1 illustrates an example of mutually similar patches, note how the targets (in grey) 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 \( \tau_{3D} \), 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 \( \mathbf{\xi}_{x_T} \) and \( \rho_{x_T} \) 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 \( \rho_{x_T} \) and a pixel-precision spatial translation \( \mathbf{\xi}_{x_T} \).

Adjacent reference patches are typically overlapping and, in addition to that, after the registration different patches in
mizes (5) as the distance decreases. The complete aggregation
of the final estimate, but a strategy to aggregate different
overcompleteness is generally helpful in increasing the quality
of the reference) with translations
and spatial (sub-pixel) translations given with respect to the HR grid.

Algorithm 1. Registration algorithm:

\begin{enumerate}
\item $P_T \in S_R$
\item $W_T = \text{FFT}(P_T)$
\item $W_{CC} = W_R \cdot \overline{W_T}$
\item $CC = \text{FFT}_1^{-1}(W_{CC})$
\item $(x_{\text{max}}, t_{\text{max}}) = \arg \max_{(x, t) \in X \times T} CC(x, t)$
\item $\xi_{R,T} = x_{\text{max}} - 1$
\item \textbf{if} $x_{\text{max}} = \text{round}(\frac{[N_1, N_2]}{2})$ \textbf{then}
\item \hspace{0.5cm} $\rho_{R,T} = \xi_{R,T} - [N_1, N_2]$
\item \textbf{end}
\item \textbf{if} $t_{\text{max}} > \text{round}(N_S/2)$ \textbf{then}
\item \hspace{0.5cm} $\rho_{R,T} = \rho_{R,T} - N_S$
\item \textbf{end}
\end{enumerate}

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 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...
learn the linear mapping between 3-D patches of the “ground-truth” LR video and the corresponding HR estimate at an intermediate scale $1 < f_I < f_f$, and then we apply the same mapping at the higher level $f_f$ 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 $\hat{y}_I$ is available (step 1), we construct a ground-truth $y_I$, and HR estimate $\hat{y}_f$, at a chosen intermediate scale $f_I$. Since $f_I > 1$, we cannot access to the ground-truth, thus we estimate $\hat{y}_f$ 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 $\hat{y}_f$ (step 2). Then, $\hat{y}_f$ 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 $\hat{y}_f$ (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 $\hat{y}_I$. The nonlocal search is restricted within a 3-D window as in (4) centered around $\left( f_I^{-1} x_R, f_I^{-1} t_R \right)$, i.e. the position corresponding to $(x_R, t_R)$ at the intermediate scale $f_I$.

**Step 6:** Let us denote as $\hat{P}_R$ the vectorization of a 3-D patch $P_R$ extracted from $\hat{y}_I$, having size $D = N_1 N_2 N_3$, and as $\hat{P}_R \in \mathbb{R}^{D \times K_2}$ the matrix having as columns the $K_2$ vectorized patches. An equivalent $\hat{P}_R$ can be constructed by vectorizing the patches in $\hat{y}_f$. The goal of the error-correction algorithm is to calculate a linear transformation $M_R \in \mathbb{R}^{D \times D}$ that maps the dictionary of mutually similar $K_2$ patches in $\hat{y}_f$ to their corresponding “ground-truth” versions in $\hat{y}_I$. This can be solved by minimizing a constrained Tikhonov regularization problem [13], which admits the closed-form solution

$$ M_R = \hat{P}_R \hat{P}_R^T \left( \hat{P}_R \hat{P}_R^T + \lambda I \right)^{-1}, \quad (7) $$

where the superscript $^T$ denotes transposition, $\lambda \in \mathbb{R}^+$ is a regularization parameter and $I \in \mathbb{R}^{D \times D}$ 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 $f_f$, 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
(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.
patches that also exhibit a sub-pixel translation with respect to the distances (2) and (3). In particular, (3) could favor target sequences and real videos showed promising performances mapping to each 3-D patch. Experimental results on both test registered video estimate by applying an error-correcting linear is finally leveraged to further improve the SR quality of the 3-D Fourier domain. Self-similarity at an intermediate scale register mutually similar patches at sub-pixel precision in patches along the motion trajectories of the video, and then photometric similarity and coherence of neighbouring motion block-wise distance metric which takes into account both to-fine motion-estimation strategy embedding a regularized (SR) algorithm for both grayscale and RGB videos. The have been implemented as a linear operation, this complexity frame (352 600 3-D patches per second. Therefore, depending on the cho- 

C. Real Videos

For these experiments we use the real videos Fan, Flag and Treadmill originally presented in [10]. In this case no decision 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 (352x240 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


TABLE II

<table>
<thead>
<tr>
<th></th>
<th>Bic. int.</th>
<th>Temp. int.</th>
<th>( \tilde{y}_f )</th>
<th>( \tilde{y}_f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus</td>
<td>17.10</td>
<td>0.4373</td>
<td>21.74</td>
<td>0.8193</td>
</tr>
<tr>
<td>City</td>
<td>23.58</td>
<td>0.6254</td>
<td>22.31</td>
<td>0.7716</td>
</tr>
<tr>
<td>Coastg.</td>
<td>25.84</td>
<td>0.8050</td>
<td>23.52</td>
<td>0.8556</td>
</tr>
<tr>
<td>Forem.</td>
<td>29.16</td>
<td>0.9027</td>
<td>28.99</td>
<td>0.9200</td>
</tr>
<tr>
<td>Fl. Gard.</td>
<td>16.11</td>
<td>0.5841</td>
<td>20.96</td>
<td>0.8461</td>
</tr>
<tr>
<td>Miss Am.</td>
<td>37.18</td>
<td>0.9157</td>
<td>36.66</td>
<td>0.9172</td>
</tr>
<tr>
<td>Salesm.</td>
<td>37.46</td>
<td>0.9822</td>
<td>37.59</td>
<td>0.9829</td>
</tr>
<tr>
<td>Tennis</td>
<td>20.93</td>
<td>0.6612</td>
<td>24.96</td>
<td>0.8252</td>
</tr>
</tbody>
</table>

VI. ACKNOWLEDGEMENTS

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