Video Super-Resolution Using Codebooks Derived From Key Frames

Edson M. Hung, Ricardo L. de Queiroz, Fernanda Brandi, Karen F. Oliveira and Debargha Mukherjee

***Abstract*—Example-based super-resolution (SR) is an attractive option to Bayesian approaches to enhance image resolution. We use a multiresolution approach to example-based SR and discuss codebook construction for video sequences. We match a block to be super-resolved to a low-resolution version of the reference high-resolution image blocks. Once the match is found, we carefully apply the high-frequency contents of the chosen reference block to the one to be super-resolved. In essence, the method relies on "betting" that if the low-frequency contents of two blocks are very similar, their high-frequency contents also might match. In particular, we are interested in scenarios where examples can be picked up from readily available high-resolution images that are strongly related to the frame to be super-resolved. Hence, they constitute an excellent source of material to construct a dynamic codebook. Here, we propose a method to super-resolve a video using multiple overlapped variable-block-size codebooks. We implemented a mixed-resolution video coding scenario, where some frames are encoded at a higher resolution and can be used to enhance the other lower-resolution ones. In another scenario, we consider the framework where the camera captures video at a lower resolution and also takes periodic snapshots at a higher resolution. Results indicate substantial gains over interpolation and over fixed-codebook SR and significant gains over previous works as well.**

***Index Terms*—Example-based super-resolution, video processing.**

I. Introduction

IMAGE super-resolution (SR) is the process of increasing the image resolution using information from other images [1]–[3]. Those other images can be different shots of the same scene, different frames of the same video, or they might simply compose a reference database. SR fundamentally differs from image interpolation as the latter generally uses information from neighbor pixels to estimate the missing ones. In interpolation, the information is local and local structures dictate how the missing information is filled, so that interpolation methods rarely introduce any new high frequency information. In SR, however, one looks at different images of the same object or similar contents and try to infer what the high frequency information might have been. In a sense, SR is much more aggressive than interpolation, being capable of recovering some of the missing high-frequency information, while risking introducing spurious artifacts.

Bayesian methods are widely used in SR [4], [5] as the problem of finding a high-resolution image X_h based on a lower resolution image X_l, i.e. finding X_h that maximizes $P(X_h|X_l)$, is ill-posed. As in a typical Bayesian approach, one tries to maximize $P(X_l|X_h)P(X_h)/P(X_l)$ instead, since quantities can then be estimated by training. Of course, dealing with whole images at a time is not tractable, and all the many works on Bayesian approaches to SR have to do with how one breaks the image, what features or parts of X_l and X_h are considered for training or processing, and so on.

Iterative SR algorithms such as those using back-projection [6]–[8] can efficiently minimize the reconstruction error. Other iterative SR algorithms use projection onto convex sets (POCS) [9], [10]. In those, the super-resolved image can be iteratively improved by projecting it onto constrained sets derived from low-resolution observed images. In related works [11]–[14], algorithms are proposed assuming that the super-resolved image is a sparse representation of raw patches, achieving substantial improvements over bicubic interpolation. In that model, each patch of the image that we want to super-resolve can be represented by a linear combination of a few dictionary elements. In [15], an algorithm based on the multichannel sampling theorem was proposed. A hybrid method that combines maximum likelihood with prior information was developed in [16]. A robust variation [17] has also been suggested. In [18], the authors generalized a denoising method, called non-local-means, amounting to a SR method without explicit motion estimation. Such a work was extended in [19]. Approaches using maximum *a posteriori* formulation to solve SR problems can be found in [20], [21]. In recent works [22]–[26], the authors address the SR problem in the context of a maximum *a posteriori* framework, using multichannel image priors, achieving significant improvements for both compressed and uncompressed data. A set of non-stationary hierarchical priors and observation models were also proposed.

Frequency-domain approaches can be found in [8], [27]–[31]. The work in [8] proposes a technique to estimate the homography between multiple frames in a sequence and a reconstruction algorithm based on wavelets. The result is a robust SR method without significantly sacrificing efficiency.

Example-based SR [32] is a simplification of all the previously described processes. A database of reference images $\{X_h^i\}$ is assembled along with their associated low-resolution

E. Hung is with Departamento de Engenharia Elétrica, Universidade de Brasilia, e-mail mintsu@image.unb.br; R. de Queiroz is with the Departamento de Ciência da Computação, Universidade de Brasilia, Brazil, e-mail queiroz@ieee.org; F. Brandi was with Universidade de Brasilia, she is now with TU Muenchen, Munich, Germany, e-mail fernanda.brandi@tum.de; Karen F. Oliveira was with Universidade Brasilia, she is now with the Brazilian Court of Audit, Brazil, e-mail karen@image.unb.br; D. Mukherjee is with Google Inc. USA.

Manuscript received August 20, 2011; revised November 14, 2011.

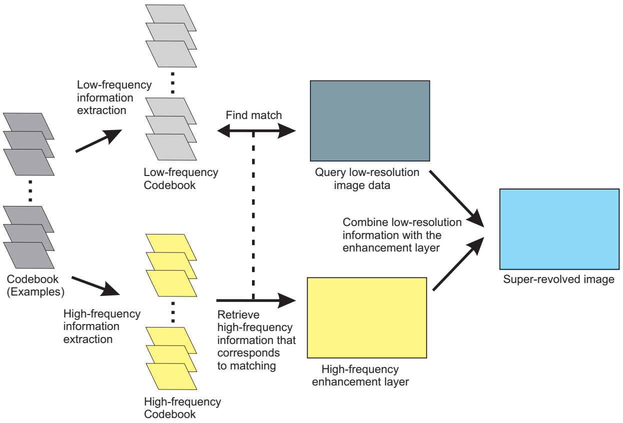

Low-frequency information extraction
Find match
Low-frequency Codebook
Query low-resolution image data
Combine low-resolution information with the enhancement layer
Super-revolved image
Codebook (Examples)
High-frequency information extraction
Retrieve high-frequency information that corresponds to matching
High-frequency enhancement layer
High-frequency Codebook

Fig. 1. General diagram of our example-based super-resolution approach.

versions $\{X_r^i\}$. For a given image (or portion thereof) X_l, a match is sought over $\{X_r^i\}$ and when a given match X_r^n is found, the associated high-frequency information (contained in X_h^n) is applied to X_l.

Different from the traditional SR problem, where higher-resolution images are reconstructed from multiple low-resolution samples, in this work, we use sparsely distributed high-resolution frames to super-resolve the low-resolution ones. In this way, the training is replaced by a search over a database (codebook). Of course, this is an overly simplified view of the process for the sake of the explanation.

Related works can be found in [33]–[36] where the example-based SR is applied to mixed-resolution video, i.e., video with different resolutions along the time. In [36], the authors propose a hybrid SR technique that combines motion compensation and an on-the-fly training dictionary.

In the case of video frames, SR approaches are basically divided into three classes: (i) applying image SR techniques to each frame independently; (ii) using motion information and multiple views of the same object along frames to provide the SR information; and (iii) using high-frequency information from key frames in mixed-resolution-video approaches. This last approach and application will be explained in detail in a later Section.

Section II explains our approach to example-based SR, while Sec. III describes our method for video SR using direct examples. The frameworks wherein the described methodology can be applied are described and tested in Sec. IV. Finally, the conclusions of this work are presented in Sec. V.

II. SR USING MULTIRESOLUTION EXAMPLES

In this section, we present our flavor of an example-based SR algorithm [33]–[35] based on Freeman *et al.* [32]. In this paper, we extend the example-based SR by including multiple-example overlapped patches and the combination of multiple high-frequency information. The proposed SR is tested in different application scenarios.

The general approach is depicted in Fig. 1. There is an image to be super-resolved, which is divided into blocks of $N \times N$ pixels. Assume one wants to increase the resolution of a block X by a factor of L, so that each super-resolved block $\widehat{X}$ would have $LN \times LN$ pixels and is found by adding some high-frequency information X_h to the upsampled version of X, X_u, as $\widehat{X} = X_h + X_u$. Let $M = LN$. We construct a database of B "example" blocks $\{Y_i\}$ of $M \times M$ pixels, compiled over many reference images. B can be very large, in the order of hundreds of thousands or even millions. Each example block Y_k is low-pass filtered yielding $Y_k^l = F_1(Y_k)$ and its respective high-pass version $Y_k^h = Y_k - Y_k^l$. It is preferred the filter F_1 be the decimation-interpolation operation by a factor of L, i.e. pre-filtering, down-sampling by L, upsampling by L, and post-filtering. The SR process works as follows. Block X is interpolated to form X_u, so that X_u is compared to each Y_k^l under some distance metric D, and we pick $\nu = \min_k D(X_u, Y_k^l)$, i.e. Y_ν^l is picked. The high-frequency information associated with Y_ν^l is Y_ν^h so that we make $X_h = Y_\nu^h$ and the super-resolved block is $\widehat{X} = X_u + Y_\nu^h$.

The method is simple, yet efficient. Nevertheless, in such a basic form it is left with many challenges. Most importantly, it may incorporate noise along with plausible high-frequency information, when the match is not very good. In our approach, we can significantly reduce noise by using multiple codebooks.

All the examples in the database form a codebook of example blocks. In essence, we have a codebook of high-frequency patterns from which to choose one to incorporate into the block to be super-resolved. If we populate the database with N_i images of $N_r \times N_c$ pixels, the examples can be all overlapping blocks in those images, so that $B \approx N_i N_r N_c$. As in any vector quantization process, the larger the codebook, the better the chances of good results, but the slower the implementation. Thus, populating the codebook with meaningful blocks is crucial to the algorithm performance. The match may improve if we use a combination of example blocks.

Let we compose K codebooks, each perhaps derived from different sources or images with different characteristics. Let the n-th codebook contain blocks $\{Y_i(n)\}$, with their respective low- and high-pass versions $\{Y_i^l(n)\}$ and $\{Y_i^h(n)\}$. Let also $\nu(n)$ be the index of the best match for the n-th codebook. We search for

$$\min_{\{\alpha_n\}} D\left(X_u, \sum_{n=1}^{K} \alpha_n Y_{\nu(n)}^l(n)\right), \tag{1}$$

so that

$$X_h = \sum_{n=1}^{K} \alpha_n Y_{\nu(n)}^h(n). \tag{2}$$

In order to calculate α_n, let $\widehat{Y}_\nu^h$ be the enhancement (block with missing high-frequency information) of a block estimated from the fusion of multiple information and let Y_ν^h be an enhancement block prediction at the n-th reference (forward or backward) codebook. Also, let $\overline{Y^h}$ be the ideal enhancement of the block and ϵ_n be spatial noise from the n-th reference key-frame-based codebook. The predicted enhancement block can be modeled as

$$Y_{\nu(n)}^h = \overline{Y^h} + \epsilon_n, \quad \epsilon_n \sim N(0, \sigma_n^2), \tag{3}$$

assuming that the noise signals (ϵ_n) are i.i.d..

Let $\mathbf{Y_K^h}$ be a set of predicted enhancement blocks, that is, $\mathbf{Y_K^h} = [Y_{\nu(1)}^h, ..., Y_{\nu(K)}^h]$. We assume that the probability density function (PDF) of $\overline{Y^h}$ is modeled by a Gaussian distribution with local mean μ_0 and local variance σ_0^2. The PDF of the predicted enhancement block, conditioned to the ideal enhancement block, $p(\mathbf{Y^h_K}|\overline{Y^h})$, is normal with mean μ_0 and covariance $\mathbf{\Gamma_K} = diag[\sigma_1^2, \sigma_2^2, ..., \sigma_K^2]$. Hence, using the Gaussian function formula and the Bayes' theorem, the PDF $p(\mathbf{Y^h_K})$ is normal with mean μ_0 and covariance $\mathbf{C} = \mathbf{\Gamma_K} + \sigma_0^2$. The *a posteriori* PDF on $\overline{Y^h}$, given the predicted data $\mathbf{Y^h_K}$, i.e. $p(\overline{Y^h}|\mathbf{Y^h_K})$ is also normal with mean $\left(\mathbf{\Gamma_K}^{-1} + \frac{1}{\sigma_0^2}\right)^{-1} \left(\mathbf{\Gamma_K}^{-1}\mathbf{Y^h_K} + \frac{\mu_0}{\sigma_0^2}\right)$ and covariance $\left(\mathbf{\Gamma_K}^{-1} + \frac{1}{\sigma_0^2}\right)^{-1}$. One way to fuse these predictions based on the maximum *a posteriori* (MAP) criterion is

$$\widehat{Y}_\nu^h = \arg_{\overline{Y^h}} \max \left(\ln \left(p(\overline{Y^h}|\mathbf{Y^h_K}) \right) \right). \tag{4}$$

The MAP-predicted enhancement block fusion estimate is simply the *a posteriori* mean

$$\widehat{Y}_\nu^h = \left(\mathbf{\Gamma_K}^{-1} + \frac{1}{\sigma_0^2}\right)^{-1} \left(\mathbf{\Gamma_K}^{-1}\mathbf{Y^h_K} + \frac{\mu_0}{\sigma_0^2}\right). \tag{5}$$

For K prediction blocks, we get, in scalar notation:

$$\widehat{Y}_\nu^h = \left(\sum_{n=1}^{K} \frac{Y^h_{\nu(n)}}{\sigma_n^2} + \frac{\mu_0}{\sigma_0^2} \right) \left(\sum_{n=1}^{K} \frac{1}{\sigma_n^2} + \frac{1}{\sigma_0^2} \right)^{-1}. \tag{6}$$

The ML fusion estimate can be recovered from (6) by assuming a flat prior, i.e. $\sigma_0^2 \to \infty$, so the final form is:

$$\widehat{Y}_\nu^h = \left(\sum_{n=1}^{K} \frac{Y^h_{\nu(n)}}{\sigma_n^2} \right) \left(\sum_{n=1}^{K} \frac{1}{\sigma_n^2} \right)^{-1}. \tag{7}$$

Observe that in (7) the variances σ_n^2 are related to the confidence of a predicted high-frequency block information. However, this information is not measurable. Here, we propose an SSD-based distortion (D_n) in order to measure the distance between blocks at the non-key frames (Y^l) and key-frames (X_u). We then use D_n as a replacement for σ_n^2 and rewrite (7) as:

$$\widehat{Y}_\nu^h = \left(\sum_{n=1}^{K} \frac{Y^h_{\nu(n)}}{D_n} \right) \left(\sum_{n=1}^{K} \frac{1}{D_n} \right)^{-1}. \tag{8}$$

Finally, we calculate α_n as:

$$\alpha_n = \left(\frac{1}{D_n} \right) \left(\sum_{n=1}^{K} \frac{1}{D_n} \right)^{-1}. \tag{9}$$

In Eq. (9) we calculate the weights for the predicted high-frequency information block $\left(Y^h_{\nu(n)}\right)$ from a set of K codebooks. The term $1/D_n$ implies that the weight of $Y^h_{\nu(n)}$ is inversely proportional to the distortion $D_n\left(X_u, Y^l_{\nu(n)}\right)$, normalized by $\left(\sum_{n=1}^{K} 1/D_n\right)^{-1}$. In the search over the codebooks if, $D\left(X_u, Y^l_{\nu(n)}\right) \gg D\left(X_u, Y^l_{\nu(m)}\right)$, we may expect $\alpha_m \gg \alpha_n$. If that happens for all blocks, then the n-th codebook is completely dominated by the m-th one and it becomes irrelevant.

In modern video coding [37], [38], block partitions in motion estimation are found after a rate-distortion analysis. In our block SR case, we only have distortion available and it has been shown [34] that the 16 × 16-pixel blocks yield better overall results than its partitions. Differently from the coding case, we are not only interested in the minimization of the prediction error, but also in the detection of scene objects to be super-resolved. Thus, with larger block sizes, the object structures are more easily identified than in partitioned blocks. With partitioned blocks we can also look for smaller content details to be super-resolved. Hence, using variable block sizes we can take advantage of both characteristics. The problem is that the motion estimation using 16×16-pixels macroblocks is a subset of that using partitioned blocks of 8×8-pixels. Thus, we suggest a penalty factor (p_F) to multiply the partitioned-block prediction error.

In a search for the best penalty factor, Fig. 2 depicts the system performance as we change p_F. In it, the 1st and 30th frames of a sequence are used as key-frames (codebooks) while we super-resolve the 28 non-key-frames in between them. For each frame, we varied p_F and observed the PSNR of the super-resolved frame. In Fig. 2, we normalized the PSNR values to their maximum. We can see that better performance is reached around $1.3 < p_F < 2.2$.

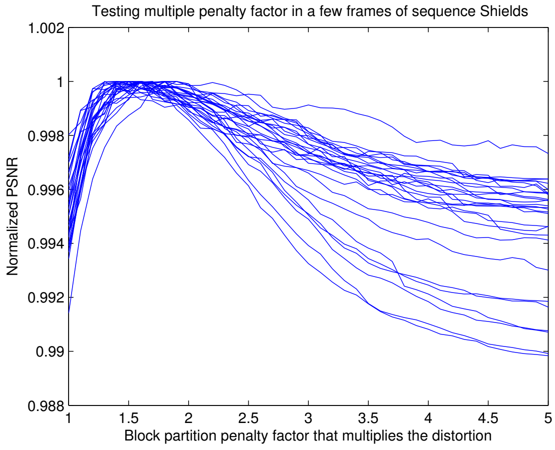


Fig. 2. SR performance as we vary the partition penalty factor applied to the non-key-frames. The 1st and 30th frames are key-frames, while the other frames in between them are the non-key ones. PSNR was normalized to their maximum values.

In order to effectively explore the temporal image correlation, we use variable-block-size motion estimation and overlapped-block motion compensation (OBMC) [39]–[41]. A virtual re-partition [41] of the blocks allows for different block sizes in OBMC. In this case, the blocks are partitioned until the smallest size permitted to the quadtree partition [37], [38] is achieved. That enables an equivalent fixed-block-size scheme as illustrated in Fig. 3.

Fig. 4 illustrates overlapped blocks in OBMC. We use only 2-pixels-wide overlap to minimize the blocking effect, while keeping the most of the high-frequency information of the

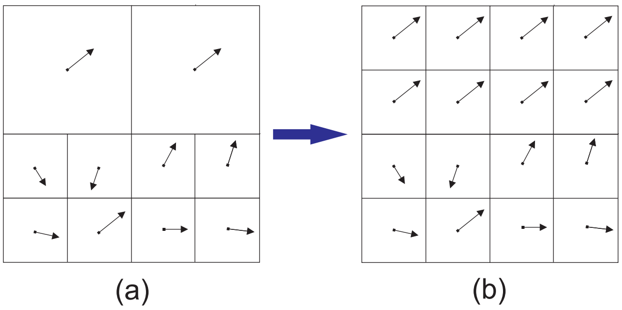


Fig. 3. The 16 × 16 and 8 × 8 blocks in (a) were "virtually re-partitioned" in (b) i. e., they are partitioned and the new blocks inherit the motion vectors.

block interior. The proposed OBMC scheme is also compatible with fast motion estimation algorithms [42]–[45].

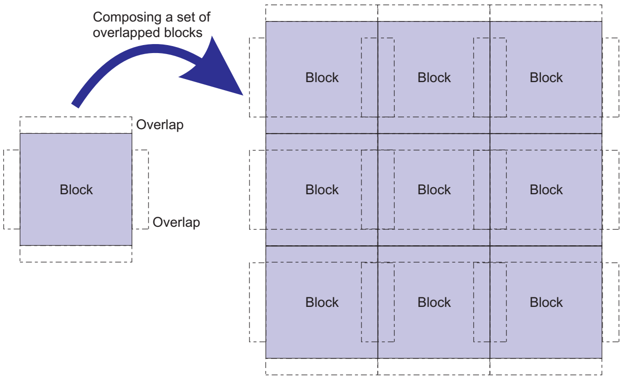


Fig. 4. Overlapped block window in OBMC.

The proposed algorithm is as follows. First, in order to make the codebooks, select blocks $\{Y_i(n)\}$, $1 \leq i \leq B$, $1 \leq n \leq K$ and for each one compute $Y_i^l(n) = F_2(F_1(Y_i(n)))$ and $Y_i^h(n) = Y_i(n) - Y_i^l(n)$, where F_1 and F_2 are the downsizing and upsizing filters.

In order to increase the resolution of one block:

- Input X and interpolate it by a factor of L to make X_u.
- For each codebook n find $\nu(n) = \min_k D(X_u, Y_k^l(n))$
- Solve $\{\alpha_n\}$ as in (9).
- Compute $\widehat{Y}_\nu^h$ as in (8).
- The super-resolved block is $\widehat{X} = X_u + \widehat{Y}_\nu^h$.

The described algorithm is performed for each block of a frame in order to super-resolve the whole image. In this paper, we apply these techniques in different application scenarios described in Section III.

III. Video SR using key frames

Getting good examples for the images to be super-resolved is crucial to achieve good performance. The examples in SR are the codebook entries. Good examples lead to good matches, thus, good results. By applying the proposed distortion-based codebook weights, we can find dominant codebooks. That can be used to keep good codebook examples and discard the unsimilar ones. In image SR, one might look for other images at higher resolution with similar content. Fortunately, in some video coding applications, there are cases where high-resolution frames of the same sequence are available. These frames may have contents that would be very similar to the frame to be super-resolved. The images which are similar to the frame to be resolved are used as examples rather than a pre-chosen or offline-trained codebook.

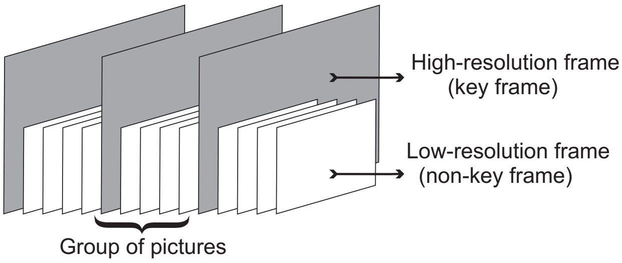


Fig. 5. A mixed resolution video format.

A. A mixed-resolution framework

In the mixed-resolution coding approach, there are key frames at high resolution and non-key frames at a lower resolution in order to save bit-rate and to reduce encoding complexity [46], [47]. The approach is depicted in Fig. 5, where key frames are interspersed periodically among the non-key frames. The non-key (low-resolution) frames can be super-resolved using the high-resolution key frames as codebook source. If the period of key-frames or group of pictures (GOP) is g frames, then for every non-key frame to be super-resolved, there is a key-frame at most $g/2$ frames away. Typically, the closest-key and non-key frames will be very similar.

In Fig. 6, we show the diagram that describes the process of super-resolving the video sequence using the information of the key-frames. In it, the first step is to distinguish the key-frames from the non-key ones. The key-frames are downsampled and upsampled with a Lanczos pre- and pos-filter generating an interpolated version of the key-frame. We perform bidirectional motion estimation [35] between the interpolated version of the key-frames and the interpolated non-key frame, which yields better performance then if motion estimation were carried between the key-frame and the interpolated non-key one. With motion estimation, we dynamically populate the codebook with the contents of the key-frame that may correspond to the block being processed. The process of searching the codebook would be equivalent to block-match motion estimation over a $w \times w$-pixel search window in the key frame. This would reduce the codebook size and avoid searching over the whole image and over many images ($N_i N_r N_c$ block comparisons). Even with full search, there are w^2 block comparisons. Window sizes in motion estimation are typically in the order of $w = N_c/8$ or $w = N_r/8$, which makes the speed up in the order of $64N_i$. With fast motion estimation techniques [42]–[45], this speed up may largely increase. We use a variable block size (16 × 16- and 8 × 8-pixels) OBMC in order to improve temporal prediction. The high-frequency layer is the registered high-frequency information that is extracted from the key frame. The super-resolved frame is obtained by adding the high-frequency layer into the interpolated low-resolution frame.

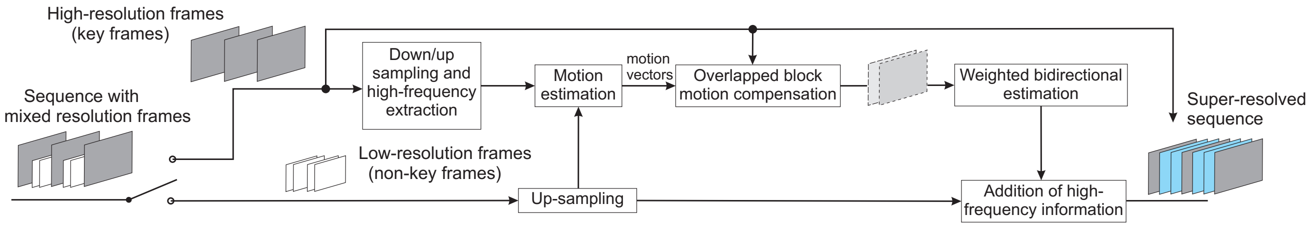


Fig. 6. General diagram of our example-based super-resolution approach for the mixed resolution scheme.

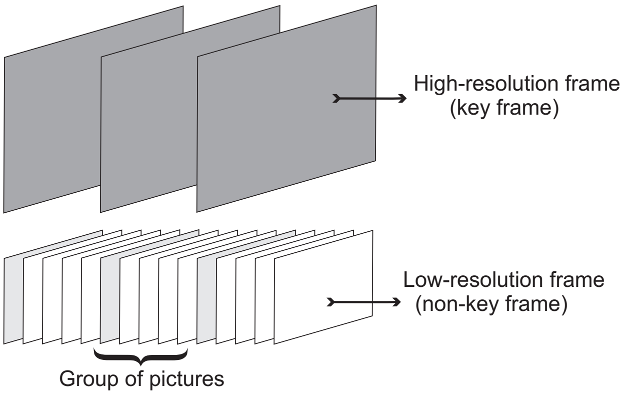


Fig. 7. The multi-recording (video plus photos) video format.

B. Video with redundant snapshots

In another application, the camera captures and compresses the video at lower resolution, but takes periodic snapshots at a higher resolution, e.g. one JPEG per second, as illustrated in Fig. 7. The high-resolution pictures are used to increase the resolution of the video sequence. This high-resolution image can be used to populate the codebook and serves just like the key-frames in the mixed-resolution approach. In other words, we can use motion estimation techniques to explore temporal redundancies and to reduce the codebook size as well. Differently from the previously described application scenario, we have redundant key-frame and non-key-frame in the same temporal instance, which simplifies the extraction of the high-frequency information of a frame. In this case, we are also using different coding standards: one is a video encoder and the other is an image encoder.

In Fig. 8, we illustrate the process of super-resolving the video sequence using snapshots. We associate the simultaneously captured key-frames (snapshots) and non-key frames. In order to extract the high-frequency information we calculate the difference between the snapshot and the interpolated non-key frame that was captured at the same instance. We also down- and up-sample the snapshot to create an interpolated version of the key-frame as input to the motion estimation. We adaptively populate the codebook with the contents of the key-frame that may correspond to the block being super-resolved through motion estimation. The high-frequency information is compensated, using OBMC, to fit the low-resolution frame. The high-frequency layer is the registered high-frequency information that is extracted from the key frame. The super-resolution frame is obtained by adding the high-frequency layer to the interpolated low-resolution frame.

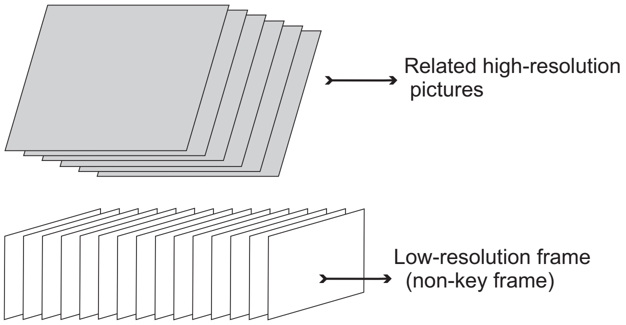


Fig. 9. Video and correlated photographs.

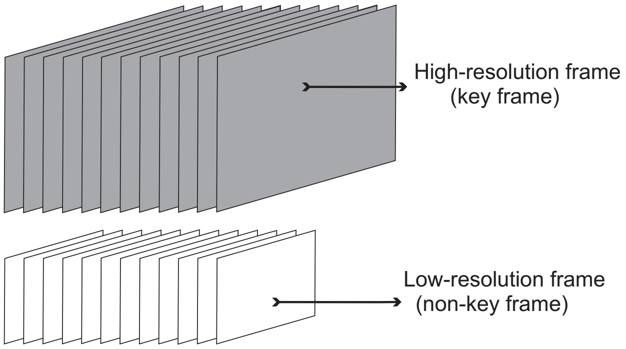
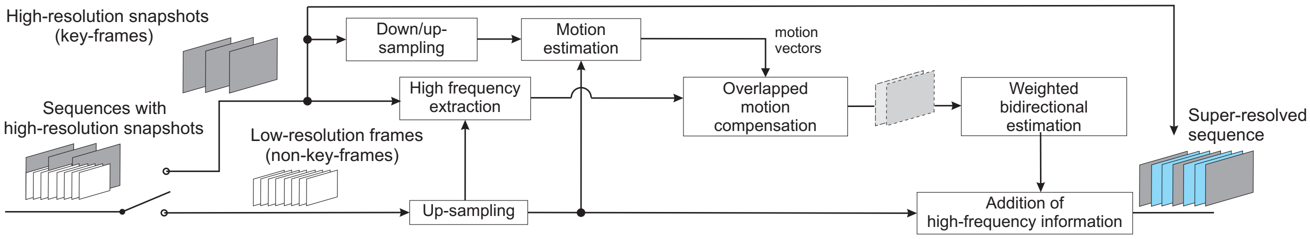


Fig. 10. Video and redundant low-resolution frames for error concealment.

C. Other application scenarios

Another application would be to compress the video at a lower resolution and then, off-line, to search databases for high-resolution pictures of similar scenes, as illustrates Fig. 9. There may contain different illumination among the video and the pictures. This is a variation of example-based super resolution, applied frame-by-frame [32]. The pictures can be used to populate the codebook without any criteria to define a GOP, as in previously described frameworks. The related picture must be well selected (we could use photos with the same geotagging position, similar compass direction and a few criteria based on the picture energy, histogram, etc.). The problem is that we can add errors to the video when we apply mismatched high-frequency information.

The last application example, illustrated in Fig. 10, is error concealment, where a video sent through a channel is compressed at a high resolution, while low-resolution frames (thumbnails) are also sent as redundant information using another reliable channel. The low-resolution information is used when an error at the high resolution occurs [48].



High-resolution snapshots (key-frames)
Sequences with high-resolution snapshots
Low-resolution frames (non-key-frames)
Down/up-sampling
Motion estimation
motion vectors
High frequency extraction
Overlapped motion compensation
Weighted bidirectional estimation
Up-sampling
Addition of high-frequency information
Super-resolved sequence

Fig. 8. Architecture of our example-based super-resolution approach applied to a sequence with snapshots.

IV. Experimental Results

The performance of the SR method is determined by the correlation between the low-quality video with the undecimated frame. For example, we made a test with the Foreman sequence originally in CIF format and downsized it into QCIF (176 × 144 pixels). We interpolated the low-resolution sequence using a bilinear algorithm and obtained a given reconstructed frame yielding a PSNR of 28.97 dB. When we populate a codebook with the image Lena (512 × 512 pixels) and apply the SR we obtain a PSNR of 29.01 dB. However, our results show that, if we populate a codebook with highly correlated information, we can achieve much better results.

We first compare the performance of the proposed usage of multiple codebooks and also the variable block size OBMC with $p_F = 2$. Then, we perform the SR in the mixed resolution and the video-plus-snapshots scenarios. Finally, we compare the proposed SR method with some previous works [11], [13], [36].

In Fig. 11, we illustrate the subjective performance of the weighted enhancement fusion in (8). In the experiment, the 1st and 31st frames of sequence Foreman are key-frames, while we try to super-resolve the 16th frame. Using only the 1st frame in the codebook, we obtain a PSNR of 34.89 dB in the resulting frame in Fig. 11(a). Observe that a few mismatches at the motion estimation process occur in the SR. In Fig. 11(b), it is shown the SR result using a codebook based only on the 31st frame, for which we achieve 35.80 dB. If we use both codebooks and simply choose the block with smaller error we obtain a super-resolved image yielding 36.39 dB. However, in Fig. 11(c), we show the result fusing the best information of both codebooks, yielding a PSNR of 37.03 dB.

In order to compare the regular motion compensation with OBMC, we performed SR at the 16th frame of sequence News, using both 1st and 31st frames as key ones. The PSNR of the SR using OBMC is 38.81 dB, while the regular case achieves 38.50 dB. Both frames can be observed in Figs. 12(a) and 12(b), respectively. Figure 12(c) shows the difference of the SR results.

As described in Section III, the SR method could be applied in many applications scenarios. The tests were performed with 300 frames of the video sequences: *Foreman*, *Mobile*, *Hall Monitor*, *Mother & Daughter* and *News* at CIF (352 × 288 pixels) and *Shields*, *Mobcal* and *Parkrun* at 720p (1280 × 720 pixels) formats.

The videos were encoded using H.264 (JM 15.1) and the set of $\{22, 27, 32, 37\}$ quantization parameters (QP) in order to compare rate-distortion curves [49]. At the SR process, a motion estimation window of 32 × 32 pixels is used for low-resolution frames and a 64 × 64-pixel window is used for high-definition video. The tests were performed to simulate a mixed-resolution framework using QCIF (176 × 144 pixels) and CIF frame sizes. We also mixed 360p (640 × 360 pixels) and 720p resolutions as well. Figure 13 shows the SR result using GOP lenght of 2. Here, we can achieve up to 4dB gains over the interpolated case. In Table I, we can observe significant objective gains of the proposed SR method in comparison to the interpolated case.

TABLE I
PSNR comparison [49] between interpolated video with Lanczos filter and the SR using the proposed codebooks.

Sequence	PSNR gains
Foreman	2.47 dB
Mobile	2.28 dB
Mother and Daughter	1.23 dB
Shields	1.30 dB
Parkrun	1.66 dB
Mobcal	1.73 dB

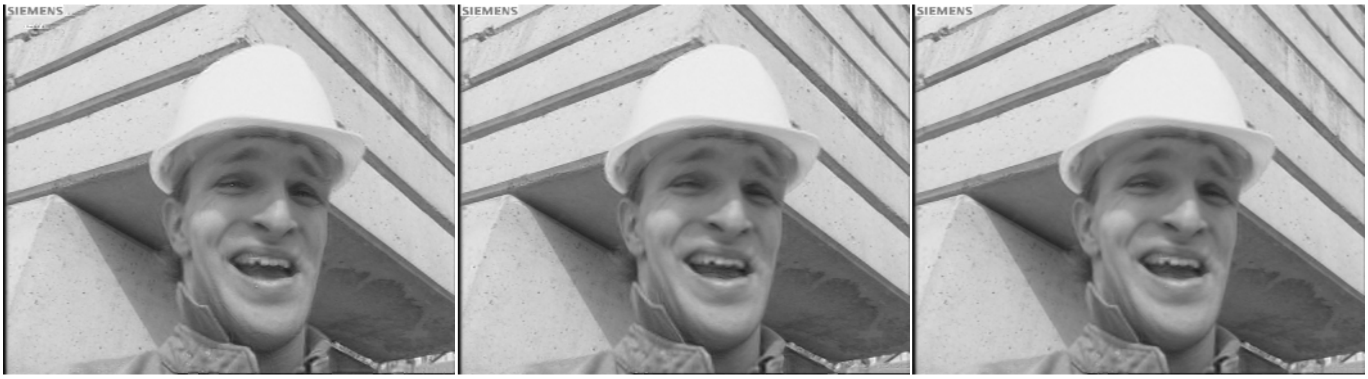
In order to test the low-resolution-video plus snapshots scenario, we used a video sequence in quarter-resolution encoded with H.264 and picked one redundant full-resolution frame per second, resulting in a GOP length of 30. The snapshot was encoded with JPEG using uniform quantization matrices. The rate-distortion curves are presented in Fig. 14, comparing the proposed SR technique against plain interpolation. Plots for the interpolation-based framework were shown for the cases including or not the snapshots in the rate and distortion computation. Other objective results are shown in Table II.

TABLE II
PSNR comparison [49] between interpolated video with Lanczos filter and the SR using the snapshots as codebooks.

Sequence	PSNR gains
Foreman	1.97 dB
Shields	1.89 dB
Parkrun	0.93 dB
Mobcal	3.21 dB
Stockholm	0.71 dB

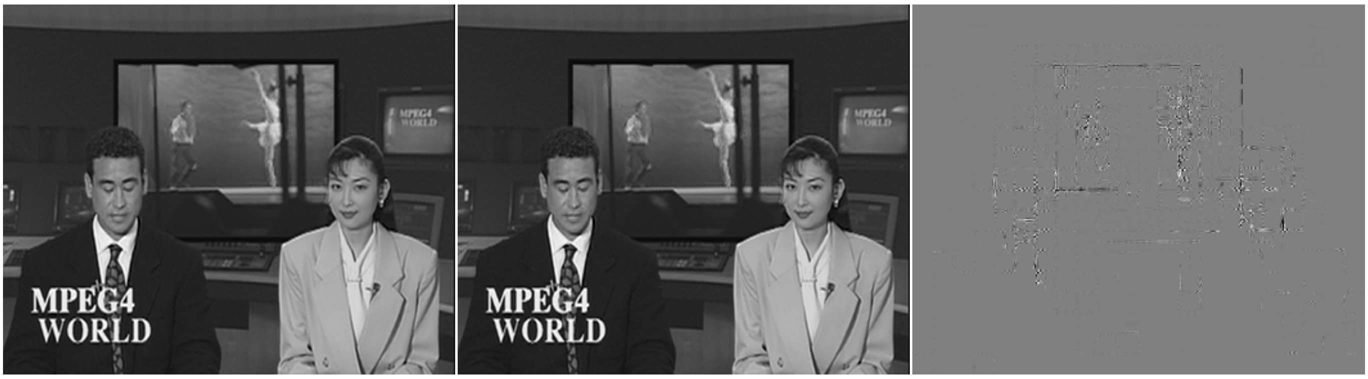
In Table III we compare our results to the frameworks described in [11], [13], [36]. The tests were performed without compression. We super-resolved the 16th frame using the 1st and 31st frames as key ones. We directly used the results reported in [36].

In order to test the SR in [11] and [13], we used the key-frames as training sets. Each training image is downsized with a bicubic filter by a factor of two and the feature extraction



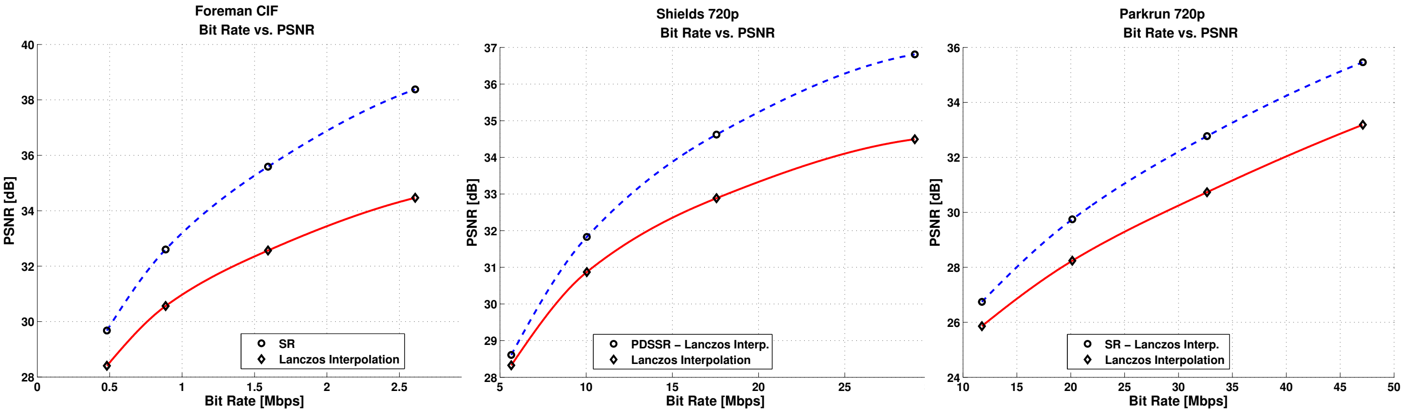
(a) (b) (c)

Fig. 11. Illustration of the performance of weighted SR using multiple frames. The 1st and 31st frames are key-frames. The SR of the 16th frame of Foreman sequence using a 32 × 32 search window: (a) using the 1st frame, (b) using the 31st frame and (c) using both 1st and 31st frames as codebooks.



(a) (b) (c)

Fig. 12. SR results of the 16th frame of the News sequence: (a) using OBMC and (b) regular block-based motion compensation. (c) Difference between (a) and (b).

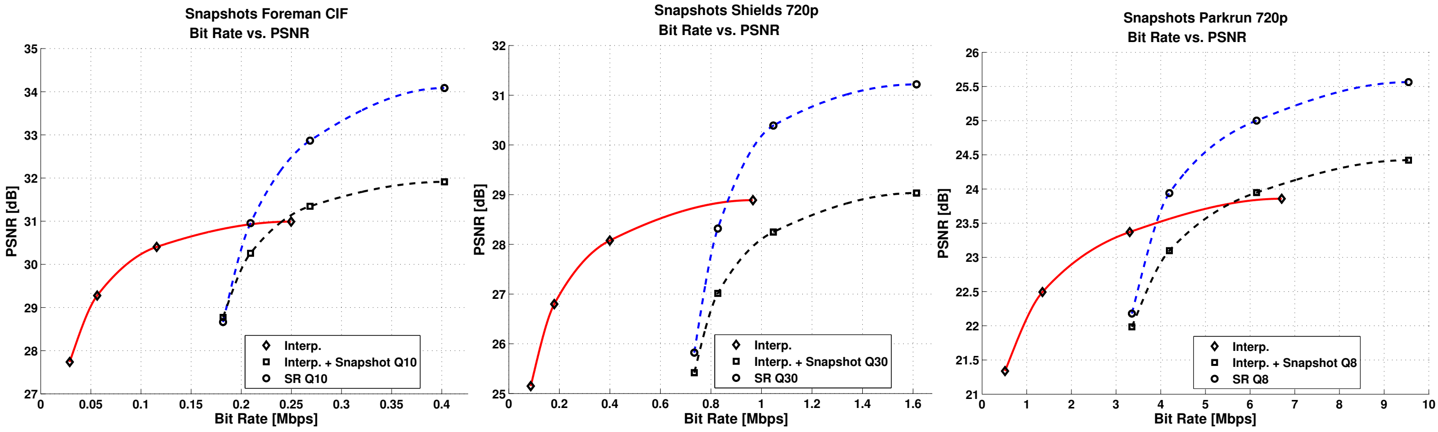


(a) (b) (c)

Fig. 13. Comparison among the interpolated video and super resolution method applied to sequences: (a) Foreman, (b) Shields and (c) Parkrun, in the mixed resolution scenario.

is performed by using gradient and Laplacian filters. Here, we used 1000 patch-pairs to compose the dictionary that was used to super-resolve the frame. For instance, using an Intel Core 2 Duo P8600 at 2.4 GHz with 4GB of RAM to train the dictionary and then super-resolve a 720p resolution video took about 12 minutes using [13] and a few hours with [11]. Our SR took less then a couple of minutes to perform the proposed SR. Note that none of the implementations involved were optimized for speed in any sense. Nevertheless, we just want to highlight the potential speed-up the reduced search can provide.

Figure 15 shows further examples of the interpolated and the super-resolved frames. We can use the original image in Fig. 15(a) to subjectively compare the quality enhancement.



(a) (b) (c)

Fig. 14. Rate-distortion-based comparison between our proposed SR and plain interpolation for the video-plus-snapshots case. Low-resolution H.264-coded video is super-resolved with the aid of periodic JPEG-compressed pictures. The values QN imply that a uniform quantizer with all step values of N are used for JPEG compressing the snapshots. For the interpolation case, plots may or may not include the snapshots in the rate or distortion computations. (a) Foreman, (b) Shields and (c) Parkrun.

TABLE III
PSNR [DB] COMPARISON AMONG INTERPOLATION AND SR METHODS.

Sequence	Bicubic [36]	Lanczos	SR in [11]	SR in [13]	MSR [36]	HSR [36]	Our SR
Container	27.9	27.4	23.6	30.7	31.9	33.2	**36.0**
Hall	29.1	28.2	24.2	32.6	37.4	38.0	**41.1**
Mobile	22.9	22.8	20.4	25.5	24.5	25.5	**27.1**
News	29.4	30.1	24.6	34.1	31.9	36.1	**38.8**
Mobcal	27.7	27.8	24.2	29.8	30.9	31.0	**35.0**
Shields	31.1	33.1	27.4	34.9	31.4	32.7	**36.0**

We also compare the subjective performance of different interpolation kernels: the bicubic shown in Figure 15(b) was used in [11], [13] and the Lanczos was used in our SR can be found in Figure 15(c). The SR proposed in [11], [13] and our algorithm are shown respectively in Figures 15(d), 15(e) and 15(f).

V. Conclusions

In this paper, we propose a few scenarios that allows for the use of correlated and dynamically populated codebooks for example-based SR techniques. We propose a method to use, discard or mixture the high-frequency information from a set of codebooks, obtaining significant objective and subjecive gains. An improved performance occurs when we apply the OBMC, which also contributes to objective gains and blocking-effect reduction. The PSNR improvement over the interpolated video is up to 3 dB for both mixed-resolution framework and video-plus-snapshot architectures. In the first scenario, we can achieve encoding complexity reduction by decreasing the efforts of the motion estimation process (that are performed at low-resolution frames).

In the mixed resolution approach, the proposed SR method has shown to provide better objective and subjective performance compared to previous works. In the other example application, where pictures (snapshots) are taken while the video recording is performed, the proposed SR has shown superior objective and subjective performance. The proposed method can effectively improve the video resolution by extracting the high-frequency information from the snapshots in order to super-resolve the video sequence. As future work, we plan to study the reduction of information due to the down-sampling process. That may enable an estimation of the amount of high-frequency information to be added within the SR process.

References

[1] S. Chaudhuri, *Super-Resolution Imaging*, Kluwer, 2001.

[2] S.C. Park, M.K. Park, and M.G. Kang, Super-resolution image reconstruction: a technical overview, *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 2136, May 2003.

[3] A. K. Katsaggelos, R. Molina and J. Mateos, "Super Resolution of Images and Video". *Synthesis Lectures on Image, Video, and Multimedia Processing*. Morgan and Claypool Publishers, 2007.

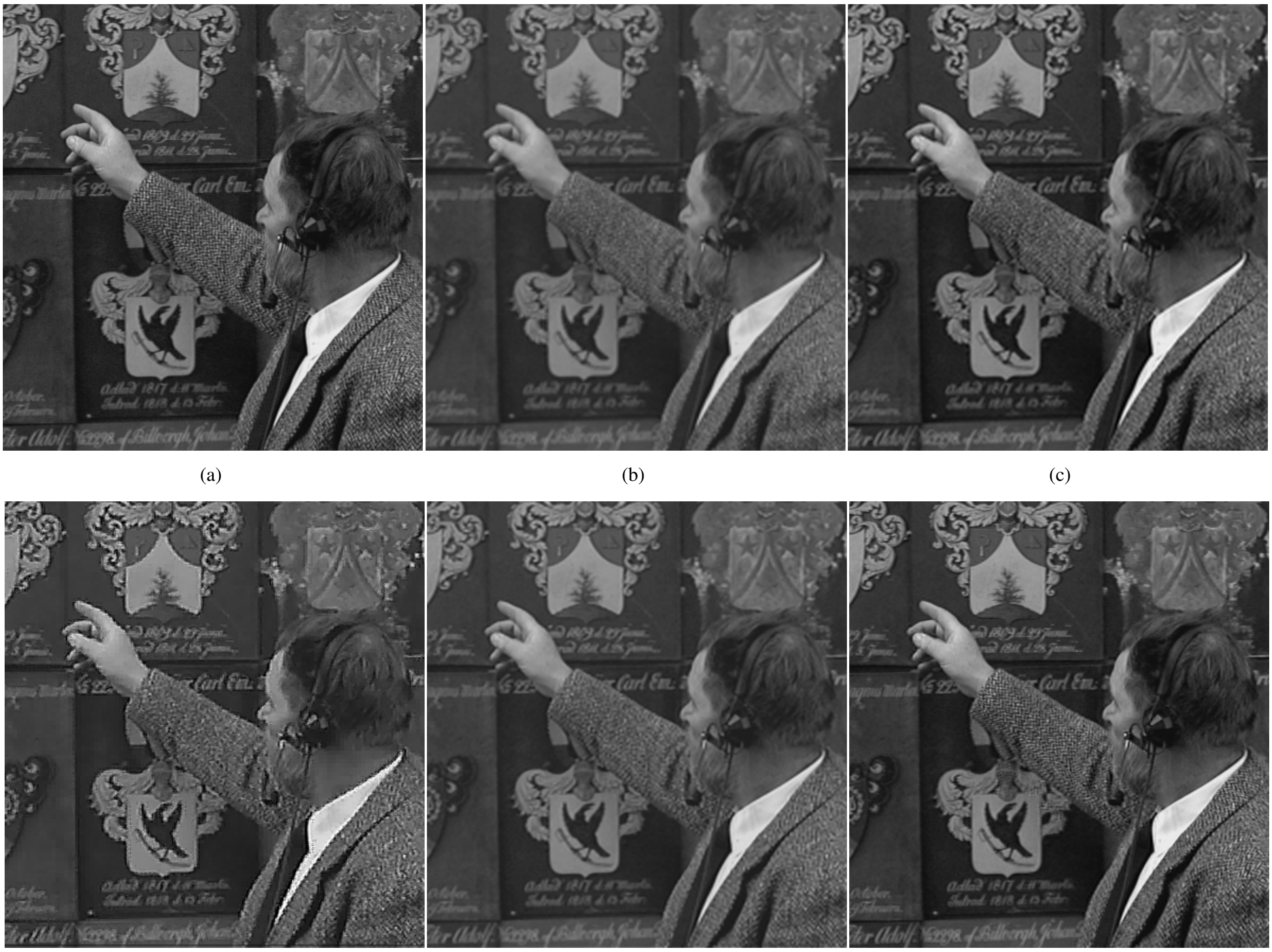
[4] C. A. Segall, A. K. Katsaggelos, R. Molina and J. Mateos, "Bayesian resolution enhancement of compressed video". *IEEE Transactions on Image Processing*, vol. 13, no. 7, 2004.

[5] M. E. Tipping and C. M. Bishop. "Bayesian image super-resolution". *Advances in Neural Information Processing Systems*, Volume 15, pp. 13031310, 2002.

[6] M. Irani and S. Peleg, "Motion analysis for image enhancement: resolution, occlusion and transparency", *Journal of Visual Communication and Image Representation*, Vol. 4, No. 4, Pages 324-335, December 1993.

[7] S. Dai, M. Han, Y. Wu, and Y. Gong, "Bilateral back-projection for single image super resolution," *IEEE International Conference on Multimedia and Expo*, Beijing, China, July 2-5, 2007, pp. 1039-1042.

[8] H. Ji and C. Fermuller, "Robust wavelet-based super-resolution reconstruction: theory and algorithm", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31 (4), April 2009.



(a) (b) (c)

(d) (e) (f)

Fig. 15. A region of the 16th frame of sequence Shields: (a)original, (b) interpolated with Bicubic filter, (c) interpolated with Lanczos filter, (d) super-resolved with [11] (e) super-resolved with [13] and (f) super-resolved with the proposed methods.

[9] H. Stark and P. Oskoui, “High-resolution image recovery from image-plane arrays using convex projections,” *Journal of the Optical Society of America*, Series A, vol. 6, pp. 1715-1726, Nov., 1989

[10] F. W. Wheeler, R. T. Hoctor, and E. B. Barrett, “Super-resolution image synthesis using projections onto convex sets in the frequency domain”, *IS&T/SPIE Symposium on Electronic Imaging, Conference on Computational Imaging*, Vol. 5674, San Jose, pp. 479-490, January, 2005.

[11] J. Yang, J. Wright, T. S. Huang and Y. Ma, “Image super-resolution as sparse representation of raw image patches,” *IEEE Computer Vision and Pattern Recognition (CVPR)*, June 23-28. 2008, pp. 1-8.

[12] M. Protter and Michael Elad, “Image sequence denoising via sparse and redundant representations”, *IEEE Trans. on Image Processing*, Vol. 18, No. 1, Pages 27-36, January 2009.

[13] R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” *Curves & Surfaces, Avignon-France*, June 24-30, 2010.

[14] J. Yang, J. Wright, T. S. Huang and Y. Ma, “Image super-resolution via sparse representation,” in *IEEE Transactions on Image Processing,* vol. 19, issue 11, pp. 2861-2873, Nov. 2010.

[15] H. Ur and D. Gross, “Improved resolution from sub-pixel shifted pictures,” *CVGIP:Graph. Models Image Processing*, vol. 54, no. 181186, Mar. 1992.

[16] M. Elad and A. Feuer, “Restoration of single super-resolution image from several blurred, noisy and down-sampled measured images,” *IEEE Trans. Image Processing*, vol. 6, no. 12, pp. 16461658, Dec. 1997.

[17] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, “Fast and robust multiframe super resolution,” *IEEE Trans. Image Process.,* vol. 13, no. 10, pp. 13271344, Oct. 2004.

[18] M. Protter, M. Elad, H. Takeda, and P. Milanfar, “Generalizing the non-local-means to super-resolution reconstruction”, *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 36-51 , Jan. 2009.

[19] M. Protter and M. Elad, “Super-resolution with probabilistic motion estimation”, *IEEE Transactions on Image Processing*, Vol. 18, No. 8, Pages 1899-1904, August 2009.

[20] R. R. Schultz and R. L. Stevenson, “Extraction of high-resolution frames from video sequences,” *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 9961011, Jun. 1996.

[21] H. Shen, L. Zhang, B. Huang, and P. Li, “A MAP approach for joint motion estimation, segmentation, and super resolution,” *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 479490, Feb. 2007.

[22] S. P. Belekos, N. P. Galatsanos, and A. K. Katsaggelos, “Maximum a posteriori video super-resolution with a new multichannel

image prior," in *Proc. EUSIPCO 2008*, Lausanne, Switzerland, August 25-29. 2008, pp. 25-29.

[23] S. Belekos, N. Galatsanos, S. D. Babacan, and A. K. Katsaggelos, "Maximum a posteriori super-resolution of compressed video using a new multichannel image prior," *IEEE International Conference on Image Processing*, Cairo, Egypt, November 2009, pp. 2797-2800.

[24] S. P. Belekos, N. P. Galatsanos, and A. K. Katsaggelos, "Maximum a posteriori video super-resolution using a new multichannel image prior," *IEEE Trans. on Image Processing*, vol. 19, issue 6, pp.1451-1464, June 2010.

[25] S. P. Belekos, N. P. Galatsanos, and A. K. Katsaggelos, "Maximum a posteriori super-resolution of compressed video with a novel multichannel image prior and a new observation model," *European Signal Processing Conference (EUSIPCO)*, Barcelona, Spain, August 2011.

[26] S. P. Belekos, J. Jeon, J. Lee, J. Paik, and A. K. Katsaggelos, "Region-based super-resolution reconstruction using parallel programming," *International Technical Conference on Circuits/Systems, Computers and communications (ITC-CSCC)*, Gyeongju, Korea, June 2011.

[27] R. Y. Tsai and T. S. Huang, "Multi-frame image restoration and registration," *Adv. Comput. Vis. Image Process.*, vol. 1, no. 1, pp. 317339, 1984.

[28] R. Chan, T. Chan, L. Shen, and Z. Shen. "Wavelet deblurring algorithms for spatially varying blur from high-resolution image reconstruction." *Linear Algebra and its Applications*, pp. 139155, 2003.

[29] R. Chan, S. Riemenschneider, L. Shen, and Z. Shen. "High-resolution image reconstruction with displacement errors: a framelet approach." *International Journal of Imaging System and Technology*, 14:91104, 2004.

[30] R. Chan, S. Riemenschneider, L. Shen, and Z. Shen. "Tight frame: An efficient way for high-resolution image reconstruction." *Applied and Computational Harmonic Analysis*, 17:91115, 2004.

[31] P. Vandewalle, S. E. Ssstrunk, and M. Vetterli, "A frequency domain approach to registration of aliased images with application to superresolution," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 114, 2006.

[32] W.T. Freeman, T.R. Jones, and E.C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, Vol. 22, pp. 56-65, 2002.

[33] F. Brandi, R. de Queiroz, D. Mukherjee, "Super resolution of video using key frames,", *Proc. IEEE Intl. Symp. on Circuits and Systems*, Seattle, USA, May 2008.

[34] F. Brandi, R. L. de Queiroz, and D. Mukherjee, "Super-resolution of video using key-frames and motion estimation," *Proc. IEEE Intl. Conf. on Image Processing*, ICIP, San Diego, CA, USA, Oct. 2008.

[35] K. F. Oliveira, F. Brandi, E. M. Hung, R. L. de Queiroz and D. Mukherjee, "Bipredictive video super–resolution using key–frames," *Proc. IS&T/SPIE Symp. on Electronic Imaging, Visual Information Processing and Communication*, San Jose, CA, USA, SPIE Vol. 7543, Jan. 2010.

[36] B. C. Song, S. C. Jeong and Y. Choi, "Video super-resolution algorithm using bi-directional overlapped block motion compensation and on-the-fly dictionary training", *IEEE Trans. On Circuits & Systems for Video Technology*, vol. 12, No. 3, March 2011.

[37] T. Wiegand, G. Sullivan, G. Bjoontegaard and A. Luthra, "Overview of the H.264 video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, v. 13, pp. 560–576, Jul 2003.

[38] I. E. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*. Wiley, 2003.

[39] E. M. Hung and R. L. de Queiroz, "Blocking-effect reduction in a reversed-complexity video codec based on a mixed-quality framework," *Intl. Telec. Symp.*, ITS, Manaus, Brazil, Sep. 2010.

[40] S. Nogaki and M. Ohta, "An overlapped block motion compensation for high quality motion picture coding", *Proc. IEEE Int. Symp. Circuits Systems*, v. 1, pp. 184-187, 1992.

[41] J. Zhang, M. O. Ahmad and M. N. S. Swamy, "Overlapped variable size block motion compensation," *Proc. IEEE Intl. Conf. on Image Processing*, ICIP, Santa Barbara, CA, USA, Oct 1997.

[42] R. Li, B. Zeng, and M.L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438-42, Aug. 1994.

[43] S. Zhu and K.K. Ma, "A new diamond search algorithm for fast block matching motion estimation," *Proc. of Int. Conf. Information, Communications and Signal Processing*, vol.1, pp.292-6, 1997.

[44] J.Y. Tham, S. Ranganath, M. Ranganath, and A.A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. On Circuits & Systems for Video Technology*, vol.8, pp.369-77, Aug. 1998.

[45] A. Tourapis, O. C. Au, and M. L. Liou, Highly efficient predictive zonal algorithm for fast block-matching motion estimation, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, pp. 934-947, Oct. 2002.

[46] D. Mukherjee, "A robust reversed complexity Wyner-Ziv video codec introducing sign-modulated codes," *HP Labs Technical Report*, HPL-2006-80, May 2006.

[47] D. Mukherjee, B. Macchiavello and R. L. de Queiroz, "A simple reversed complexity Wyner-Ziv video coding mode based on a spatial reduction framework," *Proc. SPIE Visual Communications and Image Processing*, VCIP, Jan 2007.

[48] C. Yeo, W. T. Tan, D. Mukherjee, "Receiver error concealment using acknowledge preview (RECAP) - An approach to resilient video streaming," *Proc. Int. Conference on Acoustics, Speech and Signal Processing, Taiwan*, April 2009.

[49] G. Bjontegaard, "Calculation of Average PSNR Differences between RD curves", *ITU-T SC16/Q6*, 13th VCEG Meeting, Austin, Texas, USA, April 2001, Doc. VCEG-M33.