Video Super-Resolution Using Codebooks Derived From Key-Frames

Edson Mintsu Hung, Student Member, IEEE, Ricardo L. de Queiroz, Senior Member, IEEE, Fernanda Brandi, Karen França de Oliveira, and Debargha Mukherjee, Senior Member, IEEE

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-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 a video at a lower resolution and also takes periodic snapshots at a higher resolution. Results indicate substantial gains over interpolation and fixed-codebook SR, and significant gains over previous works as well.

Index Terms—Example-based super-resolution (SR), 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 neighboring 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 tries to infer what the high-frequency information might have been. In this 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 $P(X_h|X_l)/P(X_l)$ instead, since quantities can then be estimated by training. Of course, dealing with whole images at a time is not tractable, and the different 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 backprojection [6]–[8], can efficiently minimize the reconstruction error. Other iterative SR algorithms use projection onto convex sets [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 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 nonlocal means, amounting to an 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] and [21]. In recent works [22]–[26], the authors addressed the SR problem in the context of a maximum a posteriori framework, using multichannel image priors, achieving significant improvements for both
Section II explains our approach to example-based SR, while Section III describes our method for video SR using direct examples. The frameworks wherein the described methodology can be applied are described and tested in Section IV. Finally, conclusions are presented in Section 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 that one wants to increase the resolution of a block $X$ by a factor of $L$ so that each super-resolved block $\hat{X}$ would have $LN \times LN$ pixels and is found by adding some high-frequency information $X_h$ to the upsampled version $X_u$, as $\hat{X} = X_h + X_u$. Let $M = LN$. We construct a database of $B$ “example” blocks $\{X_i\}$ 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 $X_i$ is low-pass filtered yielding $Y_{l,i} = F_1(X_i)$ and its respective high-pass version $Y_{h,i} = X_i - Y_{l,i}$. Let the filter $F_1$ be the decimation-interpolation operation by a factor of $L$, i.e., prefiltering, downsampling by $L$, upsampling by $L$, and postfiltering. The SR process works as follows. Block $X$ is interpolated to form $X_u$ so that $X_u$ is compared to each $Y_{l,i}$ under some distance metric $D$, and we pick $i = \min_i D(X_u, Y_{l,i})$, i.e., $Y_{l,i}$ is picked. The high-frequency information associated with $Y_{l,i}$ is $Y_{h,i}$ so that we make $X_h = Y_{h,i}$ and the super-resolved block is $\hat{X} = X_u + Y_{h,i}$.
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 information, when the match is not very good. In our approach, we can significantly reduce noise by using multiple approaches to fuse these predictions.

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

\[
\min_{\{Y(n)\}} D\left( X, \sum_{k=1}^{K} \alpha_{k} X_{k}(n) \right)
\]

so that

\[
X_{k} = \sum_{n=1}^{N} \alpha_{n} X_{k}(n) \quad \text{for} \quad k = 1, 2, \ldots, K.
\]

In order to calculate \( \alpha_{n} \), let \( Y_{k} \) be the enhancement (block with missing high-frequency information) of a block estimated from the fusion of multiple information and let \( Y_{n} \) be an enhancement block prediction at the \( nth \) reference forward or backward codebook. Also, let \( Y_{n} \) be the ideal enhancement of the block and \( \epsilon_{n} \) be spatial noise from the \( nth \) reference key-frame-based codebook. The predicted enhancement block can be modeled as

\[
Y_{n} = Y_{k} + \epsilon_{n} \quad \epsilon_{n} \sim N(0, \sigma_{n}^{2})
\]

assuming that the noise signals \( \{\epsilon_{n}\} \) are i.i.d.

Let \( Y_{k} \) be a set of predicted enhancement blocks, i.e., \( Y_{k} = [Y_{k}(1), \ldots, Y_{k}(N_c)] \). We assume that the probability density function (PDF) of \( Y_{k} \) is modeled by a Gaussian distribution with local mean \( \mu_{k} \) and local variance \( \sigma_{k}^{2} \). The PDF of the predicted enhancement block, conditioned to the ideal enhancement block, is normal with mean \( \mu_{k} \) and covariance \( C = \Gamma_{k}^{-1} + \frac{1}{\sigma_{k}^{2}} \).

The PDF on \( Y_{k} \), given the predicted data \( Y_{k} \), is also normal with mean \( \Gamma_{k}^{-1} Y_{k} + \frac{1}{\sigma_{k}^{2}} \) and covariance \( \Gamma_{k}^{-1} + \frac{1}{\sigma_{k}^{2}} \). One way to fuse these predictions based on the maximum a posteriori (MAP) criterion is

\[
Y_{k} = \arg \max \left( \ln \left( p(Y_{k}|Y_{k}) \right) \right).
\]
The MAP-predicted enhancement block fusion estimate is simply the a posteriori mean
\[
\hat{Y}_h = \left( \Gamma^{-1} + \frac{1}{\sigma^2} \right)^{-1} \left[ \Gamma^{-1} \hat{Y}_k + \frac{\mu}{\sigma^2} \right].
\] (5)

For \(K\) prediction blocks, we get in scalar notation
\[
\hat{Y}_h^k = \left( \sum_{n=1}^{K} \frac{Y^k_n}{\sigma^2_n} \right) \left( \sum_{n=1}^{K} \frac{1}{\sigma_n^2} \right)^{-1}.
\] (6)

The ML fusion estimate can be recovered from (6) by assuming a flat prior, i.e., \(\sigma_n^2 \to \infty\), so the final form is
\[
\hat{Y}_h^k = \left( \sum_{n=1}^{K} \frac{Y^k_n}{\sigma^2_n} \right) \sum_{n=1}^{K} \frac{1}{\sigma_n^2}.
\] (7)

Observe that in (7), the variances \(\sigma_n^2\) are related to the confidence of a predicted high-frequency block information. However, this information is not measurable. Here, we propose a sum of squared differences-based distortion (\(D_n\)) in order to measure the distance between blocks at the non-key-frames (\(Y^k\)) and key-frames (\(X_k\)). We then use \(D_n\) as a replacement for \(\sigma_n^2\) and rewrite (7) as follows:
\[
\hat{Y}_h = \left( \sum_{n=1}^{K} \frac{Y^k_n}{D_n} \right) \left( \sum_{n=1}^{K} \frac{1}{D_n} \right)^{-1}.
\] (8)

Finally, we calculate \(\alpha_n\) as follows:
\[
\alpha_n = \frac{1}{D_n} \left( \sum_{n=1}^{K} \frac{1}{D_n} \right)^{-1}.
\] (9)

In (9), we calculate the weights for the predicted high-frequency information block (\(Y^k_n\)) from a set of \(K\) codebooks. The term \(1/D_n\) implies that the weight of \(Y^k_n\), is inversely proportional to the distortion \(D_n (X_k, Y^k_n)\), normalized by \(\sum_{n=1}^{K} 1/D_n\). In the search over the codebooks, if \(D (X_k, Y^k_n) \gg D (X_k, Y^k_m)\), we may expect \(\alpha_n \gg \alpha_m\). 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 \(16 \times 16\) pixel blocks yield better overall results than its partitions. Unlike 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 \(16 \times 16\) pixels macroblocks is a subset of that using partitioned blocks of \(8 \times 8\) 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 first 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 peak signal-to-noise ratio (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 \(1.3 < p_F < 2.2\).

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

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

In order to increase the resolution of one block:
1) input $X$ and interpolate it by a factor of $L$ to make $X_u$;
2) for each codebook $n$ find $v(n) = \min_k D(X_u, Y_k(n))$;
3) solve (9); 
4) compute $Y_h^v$ as in (8);
5) the super-resolved block is $\hat{X} = X_u + \hat{Y}_h^v$.

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 contents. Fortunately, in some video coding applications, there are cases where high-resolution frames of the same sequence are available. These frames may have content that would be very similar to the frame to be super-resolved. The images that are similar to the frame to be resolved are used as examples rather than a prechosen or offline-trained codebook.

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 nonkey (low-resolution) frames can be super-resolved using the high-resolution key-frames as a 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 nonkey ones. The key-frames are downsampled and upsampled with a Lanczos prefilter and postfilter 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 nonkey 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 ($N_c N_r N_p$ block comparisons). Even with full search, there are $w^2$ block comparisons. Window sizes in motion estimation are typically in the order of $w = N/8$ or $w = N/8$, which makes the speedup in the order of $64 N_c$.

With fast motion estimation techniques [42]–[45], this speedup may largely increase. We use a variable block size ($16 \times 16$,
Fig. 11. Illustration of the performance of weighted SR using multiple frames. The first and 31st frames are key-frames. The SR of the 16th frame of the Foreman sequence using a 32 × 32 search window using (a) first frame, (b) 31st frame, and (c) both first and 31st frames as codebooks.

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

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. Unlike 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-sample 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 SR frame is obtained by adding the high-frequency layer to the interpolated low-resolution frame.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>PSNR Gains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreman</td>
<td>2.47 dB</td>
</tr>
<tr>
<td>Mobile</td>
<td>2.28 dB</td>
</tr>
<tr>
<td>Mother and Daughter</td>
<td>1.23 dB</td>
</tr>
<tr>
<td>Shields</td>
<td>1.30 dB</td>
</tr>
<tr>
<td>Parkrun</td>
<td>1.66 dB</td>
</tr>
<tr>
<td>Medieval</td>
<td>1.73 dB</td>
</tr>
</tbody>
</table>

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

1326 IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 22, NO. 9, SEPTEMBER 2012
**TABLE II**

PSNR Comparison [49] Between Interpolated Video With Lanczos Filter and the SR Using the Snapshots as Codebooks

<table>
<thead>
<tr>
<th>Sequence</th>
<th>PSNR Gains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreman</td>
<td>1.97 dB</td>
</tr>
<tr>
<td>Shields</td>
<td>1.89 dB</td>
</tr>
<tr>
<td>Parkrun</td>
<td>0.93 dB</td>
</tr>
<tr>
<td>Mobcal</td>
<td>3.21 dB</td>
</tr>
<tr>
<td>Stockholm</td>
<td>0.71 dB</td>
</tr>
</tbody>
</table>

**C. Other Application Scenarios**

Another application would be to compress the video at a lower resolution and then, offline, to search databases for high-resolution pictures of similar scenes, as illustrated in Fig. 9. The video and the pictures may contain different illumination. 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, and so on). 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].

**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 a 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-snapsshots 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 first and 31st frames of sequence Foreman are key-frames, while we try to super-resolve the 16th frame. Using only the first 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 in the motion estimation process occur in the SR. In Fig. 11(b), the SR result using a codebook based only on the 31st frame is shown, for which we achieve 35.80 dB. If we use both codebooks and simply choose the block with a 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 first 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 Fig. 12(a) and (b), respectively. Fig. 12(c) shows the difference of the SR results.

As described in Section III, the SR method could be applied to many application scenarios. The tests were performed with 300 frames of the video sequences: Foreman, Mobile, Hall Monitor, Mother and 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 in order to
compare rate-distortion curves [49]. In 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 a 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. Fig. 13 shows the SR result using GOP length of 2. Here, we can achieve up to 4 dB 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.

In order to test the low-resolution video plus the 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 with plain interpolation. The plots for the interpolation-based framework were shown for cases with and without snapshots in the rate and distortion computation. Other objective results are shown in Table II.

In Table III, we compare our results with the frameworks described in [11], [13], and [36]. The tests were performed without compression. We super-resolved the 16th frame using the first 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 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 P9600 at 2.4 GHz with 4 GB of RAM to train the dictionary and then to super-resolve a 720p resolution video took about 12 min using [13] and a few hours with [11]. Our SR took less than a couple of minutes to perform the proposed SR. Note that none of the implementations involved was optimized for speed in any sense. Nevertheless, we want to highlight the potential speedup that the reduced search can provide.

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

V. CONCLUSION

In this paper, we proposed a few scenarios that allowed for the use of correlated and dynamically populated codebooks for example-based SR techniques. We proposed a method to use, discard, or mix the high-frequency information from a set of codebooks, obtaining significant objective and subjective gains. An improved performance occurred when we applied the OBMC, which also contributed to objective gains and blocking-effect reduction. The PSNR showed an improvement of up to 3 dB over the interpolated video 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 was 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 showed 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.
Fernanda Brandi was born in Brazil in 1981. She received the Bachelor’s degree in electrical engineering and the Master’s degree from the Universidade de Brasília, Brasília, Brazil, in 2006 and 2009, respectively. She is currently pursuing the Ph.D. degree with the Institute of Media Technology, Technische Universität München, Munich, Germany. Her current research interests include video and haptic communications with a focus on optimizing the signal compression in the presence of communication unreliabilities.

Karen França de Oliveira was born in Brazil in 1982. She received the Bachelor’s degree in electrical engineering and the Master’s degree from the Universidade de Brasília, Brasília, Brazil, in 2006 and 2009, respectively. She is currently with the Brazilian Court of Audit.

Debargha Mukherjee (M'99–SM'09) was born in Kolkata, India. He received the B.Tech degree in electronics and electrical communication engineering from the Indian Institute of Technology Kharagpur, Kharagpur, India, in 1993, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California, Santa Barbara, in 1995 and 1999, respectively. He was with Hewlett-Packard Laboratories, Palo Alto, CA, from 1999 to 2010, as a Scientist conducting research on image and video compression and communications. He has been with Google, Inc., since 2010, where he worked on backend video transcoding and 3-D processing pipelines for YouTube, and is currently with the Chrome Team. He has been actively involved in the standardization of MPEG-21 from 2002 to 2004. His current research interests include video and image coding, processing, communications, signal processing, and information theory. He has published more than 70 papers in leading international conferences and journals, and holds 15 U.S. patents in these areas.

Dr. Mukherjee served as the Technical Co-Chair of the Second International Conference on Immersive Communication in 2009. He was the recipient of the IEEE Student Paper Award at IEEE ICIP, Chicago, IL, in 1998, and was the co-recipient of the Top 10% Paper Award at the IEEE MMSP Workshop, Brazil, in 2010.