Feature-based Video Coding: Designing an RD Efficient and Search Friendly Framework

Renam C. da Silva[1], Fernando Pereira[2], Eduardo A. B. da Silva[1]
[1]Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil
[2]Instituto Superior Técnico – Instituto de Telecomunicações, Lisboa, Portugal
{eduardo, renam.silva}@smt.ufrj.br; fp@lx.it.pt

Abstract—**To provide more powerful video enabled applications, e.g. in video surveillance environments, it is increasingly more critical not only to have access to the decoded video but also to, e.g. efficiently search for similar videos. In this context, this paper proposes a feature-based video coding solution adopting a hybrid approach where both pixels and local visual features are exploited for coding. In this novel solution, part of the frames are coded using a set of key point matches, thus allowing not only to decode the usual frames for visualization but also valuable key point information extracted from uncompressed frames which is instrumental for searching. Experimental coding results for video surveillance like sequences show that the obtained RD performance outperforms the state-of-the-art HEVC standard while additionally facilitating more accurate searching.**

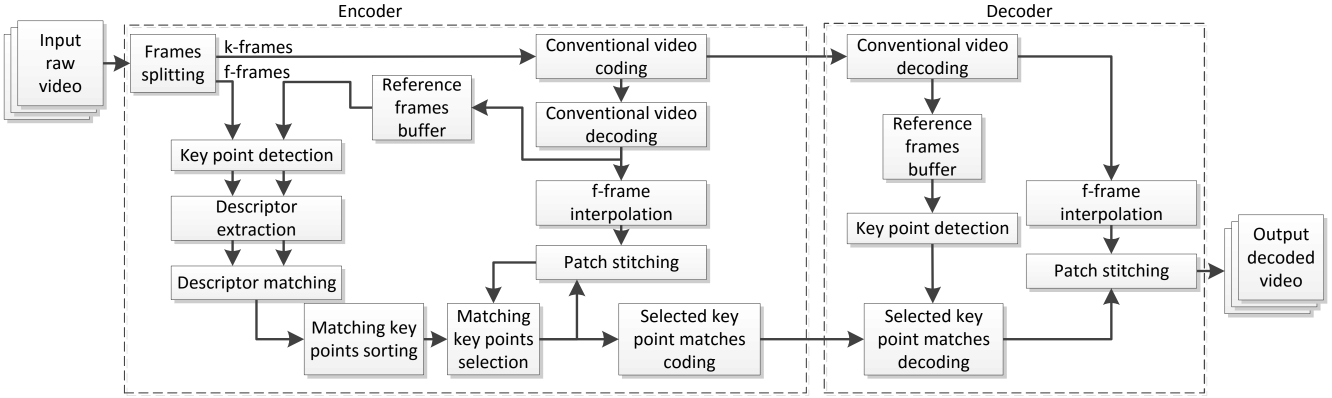
I. INTRODUCTION

Developments in the field of computer vision have led to the emergence of visual information representations which are better suited for visual analysis tasks than just pixels. Local visual features are a powerfull type of such representations which are able to efficiently perform a number of tasks such as image/video search and retrieval [1], object recognition [2, 3] and event detection. Such local features describe image characteristics that are distinctive, representative and informative. Despite their power in analysis tasks, only recently they were considered for image and video coding.

Inspired by the technique reported in [4], Yue *et al.* [5, 6] have proposed an image coding solution based on Scale Invariant Feature Transform (SIFT) descriptors. SIFT descriptors are extracted from the original image and differentially encoded with respect to SIFT descriptors extracted from a poor quality and low rate version of the image, that is first conveyed to the decoder. This downsampled image is used to guide the target quality reconstruction, since it carries enough information about the edges, colors and objects. The decoded descriptors are used to retrieve highly correlated images available in the cloud which provide image patches to reconstruct a higher quality image. A related system is proposed in [7] by Baroffio *et al.* in the context of Visual Sensor Networks (VSN). Such systems aggregate a huge amount of data captured from multiple and distributed visual sensors and perform complex visual analysis with the target to provide services such as augmented reality in sport events, behaviour analysis in security systems and automotive drive assistence. Targeting such scenarios, an hybrid coding paradigm, coined Hybrid-Analyze-Then-Compress (HATC), is proposed in [7] aiming at overcoming the limitations of the two competing paradigms, namely Compress-Then-Analyse (CTA) and Analyse-Then-Compress (ATC) [3, 7]. The HATC paradigm jointly codes the visual content using both pixel and local image features representations. A similar solution is proposed in [8] by Chao and Steinbach, where key point information detected on the uncompressed video frames is encoded along with regularly coded video. It was experimentally demonstrated that key points detected on uncompressed video are effective in reducing the detrimental effects of compression on feature matching performance [8]; *this highlights the importance of using key point data extracted from uncompressed data*. In these previous works, pixel and feature-based representations are essentially used independently from each other, meaning that the feature-level data is not exploited to aid pixel-level coding.

In this paper, a novel feature-based video coding solution is designed and assessed adopting a hybrid video coding paradigm based on both pixels and local visual features. In this coding solution, the input video frames are split into two sets of frames, namely the key-frames and feature-frames, for short *k-frames* and *f-frames*. The periodic k-frames are coded using a standard pixel-based approach and used as reference frames for the coding of the f-frames. Once the k-frames are decoded, frame rate-up conversion is performed to obtain a first coarse estimation of the f-frames. After performing feature detection and descriptor extraction in both frame sets, the f-frames are coded by simply establishing correspondences between the f-frames set of features and the reference frames set of features thus relating specific image regions in the two sets. In this way, the f-frames quality may be gradually improved by reusing appropriate texture patches from the reference frames taking advantage from the fact that video sequences usually exhibit a significant temporal redundancy. Since key point positions are provided for the f-frames extracted from uncompressed video data, visual search performance may be boosted compared with decoder extracted key points based on decoded data.

The remaining of this paper is organized as follows: in section II the novel coding architecture and associated walkthrough are presented; next, Section III details the most original tools in the architecture; finally, Section IV presents the performance assessment, and Section V highlights the main conclusions of this work.



II. Proposed Feature-based Video Coding Solution: Architecture and Walkthrough

This section presents the architecture and walkthrough of the proposed feature-based video coding (FBVC) solution. In this context, let us denote by $\boldsymbol{p}_{n,i}$ the feature vector including the key point position (x, y), scale σ and angle θ of the i-th feature in frame n and by $\boldsymbol{d}_{\mathrm{n},\boldsymbol{i}}$ the associated descriptor vector, e.g. SIFT, implying that each visual feature is fully represented by the pair $\{\boldsymbol{p}_{n,i};\ \boldsymbol{d}_{\mathrm{n},\boldsymbol{i}}\}$. The proposed coding architecture is presented in Fig. 1 and its processing flow explained in the following:

A. Encoder

- ***Frame splitting***: The original frames are split in two sets, namely the *k-frames* and the *f-frames*. While the k-frames are encoded using a conventional video codec, the f-frames are encoded using the proposed feature-based codec. The frames are arranged in a group of pictures (GOP) structure where a k-frame is periodically inserted among the f-frames. A GOP includes a k-frame and the set of f-frames until the next k-frame; for a GOP size 2, the k-frames and f-frames alternate.
- ***Conventional video coding and decoding:*** After frame splitting, the k-frames are Intra coded and decoded using a standard video codec; in this case, the state-of-the-art HEVC codec is used [9]. The decoded k-frames play a central role in the proposed coding solution since they provide references for the coding of the f-frames.
- ***F-frame interpolation***: To restore the original frame rate and effectively estimate regions with smooth spatial and temporal evolutions, the f-frames are interpolated from the neighbor k-frames both at encoder and decoder using a block-based motion compensated frame interpolation algorithm based on the next past and future decoded k-frames [10].
- ***Reference frames buffer***: The reference frames provide texture patches to improve the initially estimated f-frames. There are two types of reference frames: i) decoded k-frames, usually one or more past k-frames and only one future k-frame to limit the delay; ii) original related content frames, e.g. other video surveillance frames, which provide texture innovation since no residual texture coding is finally performed; also an initially acquired original, background frame from the same environment is included.
- ***Key point detection and descriptor extraction***: The original f-frames and the decoded k-frames feed a key point detection module which identifies the most distinctive frame positions for their characterization. For each key point a descriptor is extracted to capture the local image patch. The target is that only a well selected number of key points detected in the original f-frames are conveyed to the decoder to indicate the areas in the interpolated f-frames most needing quality improvement.
- ***Descriptor matching***: To be able to improve the f-frames with patches from the reference frames, the f-frames descriptors are matched to the reference frames descriptors using the Euclidian distance as matching metric. The intuition is that the matching descriptor pairs represent regions with similar visual content in the f-frames and reference frames. If rate-distortion (RD) efficient, some interpolated f-frame regions will be improved with appropriate matching patches from the reference frames by performing patch stitching using Poisson editing techniques [11].
- ***Matching key points sorting:*** The order by which the reference frames patches are added to the interpolated f-frame is critical for the final RD performance. Since it is desirable to encode first those key point matches which bring the most significant quality gain, this module targets to sort the key point matches according to their potential quality gain.
- ***Patch stitching***: To check the RD added value of reusing some reference frames patches to improve the interpolated f-frame, the image patch centered at a certain reference frame key point location is seamlessly stitched at the matching f-frame key point location. Before patch stitching, the reference frame patch is rotated, scaled and translated to better fit the target location using the information provided by the key point extractor.
- ***Matching key points selection****:* Not all key point matches are useful in terms of RD performance even if increasing the interpolated f-frame quality as also the rate must be minimized. Therefore, it is critical to select the limited set of key point matches which are RD efficient, notably using a Lagrangian cost, $J = D + \lambda R$, which is evaluated for

each key point match, after the corresponding reference frame patch is stitched to the f-frame. Here D is the f-frame distortion after patch stitching and R the coding rate for all syntactic elements corresponding to a specific key point match. The selection process is performed following the order defined in the sorting module which means the D is cumulative. The selected matchings are those providing a significant enough reduction in the RD cost, J, with the Lagrangian multiplier λ controlling the RD trade-off.

- ***Selected key point matches coding****:* The RD selected key points are differentially encoded with respect to the reference-frames matching key points. Before entropy coding, the residue for each key point parameter is scalar quantized to reduce its coding rate. An arithmetic encoder is used to entropy code the various syntactic elements.

B. Decoder

Since the decoder is mostly embedded in the encoder, decoding proceeds essentially as already described for the encoder. The exception is the decoding of the matching key point pairs described in detail in Section III. As usual, the most critical coding tool is the 'clever' encoder selection of where to invest the rate, meaning of which key point matches to select, while the decoder just follows the choices made by the encoder. At this stage, it is important to stress that to limit the complexity no texture residual coding is performed for the f-frames which limits its quality improvements to what the reference frames may provide.

III. Proposed Feature-Based Video Coding Solution: Coding Tools

This section targets the detailed presentation of the most novel and critical modules in the proposed architecture.

A. Patch Stitching

The patch stitching process targets to improve the f-frames quality with appropriate patches from the reference frames. In the stitching process, an image patch ${\Omega_{m,j}}^{(k)}$ centered at a key point location $(x_{m,j},\ y_{m,j})^{(k)}$ from a reference frame ${I_m}^{(k)}$ is extracted and seamlessly stitched over a matching key point location $(x_{n,i},\ y_{n,i})^{(f)}$ in the f-frame ${I_n}^{(f)}$, thus generating the stitched f-frame ${I'_n}^{(f)}$. The stitching process aims to keep unchanged the pixel values both over and outside the boundary $\partial\Omega$ of Ω, while inside blending its pixels seamlessly with those from ${I_n}^{(f)}$ and retaining the patch structure. A comprehensive formulation of this problem is given in [4, 11]. In this paper, the patch stitching is performed using the Poisson technique proposed in [11]. Given the matching key points, the source reference frame and the destination f-frame, the stitching proceeds as follows (the index *n* refers to the f-frame, the index m to the reference frame and a hat over a variable indicates quantization):

1. For a circularly-shaped image patch ${\Omega_{m,j}}^{(k)}$ with a support size defined by its diameter $m_s\hat{\sigma}$.
   - Rotate the patch by ${\hat{\theta}_{n,i}}^{(f)} - {\theta_{m,j}}^{(k)}$.
   - Scale the patch by the factor $\frac{{\hat{\sigma}_{n,i}}^{(f)}}{{\sigma_{m,j}}^{(k)}}$.
   - Translate the patch by $({\hat{x}_{n,i}}^{(f)} - {x_{m,j}}^{(k)};\ {\hat{y}_{n,i}}^{(f)} - {y_{m,j}}^{(k)})$.
   - Perform Poisson stitching as described in [11].
2. Return the stitched f-frame ${I'_n}^{(f)}$ with the support size, $m_s\hat{\sigma}$, providing the largest quality gain.

Note that the patch stitching is carried out using the quantized and non-quantized key point parameters of the original f-frame and decoded reference frame, respectively.

B. Matching Key Point Sorting

Since the order by which the key point matches are considered in the next selection process is critical for the final RD performance, it is essential to previously and appropriately sort them. The proposed key points sorting is performed as follows:

1. First, the encoder preliminarily sorts the key point matches by decreasing scale parameter.
2. In order that only RD and quality useful key point matches are coded, the mean-squared error (MSE) distortion reduction is independently estimated for those associated to a RD cost reduction. To avoid using here the complex Poisson stitching process mentioned above, the potential quality gain of each key point match is assessed by simply copying the image patch centered at the key point location in the reference frame over the matching key point location in the f-frame. This is a low complexity stitching process which avoids solving the Poisson equation at the penalty of giving only an estimation of the quality gain. For complexity reasons, this process is performed for every key point match independently, implying that the cumulative effect of the key point matches is not considered.
3. The key point matches which are able to provide RD gain are moved to the top positions of the initially sorted list ordered by the potential MSE reduction, this means quality improvement.

At the end, the list will include in the top positions the key point matches with higher quality gains and only after those without potential quality gains ordered by the scale parameter as they may still bring gains if using the 'real' Poisson stitching. The key point matches sorted list is provided to the selection module.

C. Matching Key Point Selection

A RD-driven optimization is performed to select the best set of key point matches (thus providing a set of reference frame image patches) by minimizing a Lagrangian cost function, J. More precisely, for each f-frame, the set of key point-centered image patches $\mathcal{P}^* = \bigcup {\Omega_{m,j}}^{(k)}$ belonging to the reference frames available in the reference buffer minimizing a Lagrangian cost function as follows is selected:

$$\arg\min_{\mathcal{P}^*} J(\mathcal{P}) = D(I_o, I_{\mathcal{P}}) + \lambda R(\mathcal{P}) \qquad (1)$$

where D is the distortion between the original f-frame, I_o, and the f-frame, $I_{\mathcal{P}}$, after stitching the set of patches$\mathcal{P} = \bigcup {\Omega_{m,j}}^{(k)}$, and $R(\mathcal{P})$ is the rate to code the selected set of key point matches. The mean squared error (MSE) has been used as distortion metric.

To determine the set of patches $\mathcal{P}^*$, an iterative procedure associated to each key point match is adopted as follows:

1. Perform patch stitching to check the benefit of selecting the current key point match and thus associated patch. At the beginning, the cost function minimum is set equal to the distortion between the interpolated f-frame and the original f-frame (equivalent to zero rate and $\mathcal{P} = \{\emptyset\}$). Then, patch stitching is successively carried out for each key point match and the cost function in (1) is evaluated. The cost function takes into account the resulting distortion (after cumulative patch stitching) and the accumulated rate to encode the syntactic elements.
2. If the RD Lagrangian cost is reduced, then the key point match is selected for coding and the Lagrangian cost function minimum is updated; otherwise, the key point match is discarded.

In (1), the rate cost is computed as follows:

$$R_t = R(r_k) + R(r_m) + R(m_s) + R\big(\boldsymbol{p}_{n,i}{}^{(f)} - \boldsymbol{p}_{m,j}{}^{(k)}\big) + R(\mathcal{P}^*) \qquad (2)$$

where $R(r_k)$ is the rate to code the reference frame index, $R(r_m)$ is the rate to code the matching key point index in the reference frame, $R(m_s)$ is the rate to code an appropriate scale multiplicative factor, $R\big(\boldsymbol{p}_{n,i}{}^{(f)} - \boldsymbol{p}_{m,j}{}^{(k)}\big)$ is the rate to lossy code the residuals of the key point match parameters, and $R(\mathcal{P}^*)$ is the rate to code all the previously selected key point matches.

To decrease the computational complexity associated to the RD optimization step, the total rate is estimated by computing the self-information of each syntactic element according to the probability models described in the next subsection.

D. Selected Key Point Matches Coding

In the RD optimization step, a set of image patches, $\mathcal{P}^* = \bigcup \Omega_{m,j}{}^{(k)}$, associated to the matching key point pairs is selected with the aim to improve the quality of the initially interpolated f-frame. To replicate the same patch stitching process at the decoder side, it is necessary to transmit, for each selected key point match, the following syntactic elements: a) index of the reference frame, r_k; b) index of the matching key point in the reference frame considering the order given by the key point detector itself, r_m; c) encoder selected scale multiplicative factor, m_s; d) quantized residue of each key point parameter $\boldsymbol{p}_{n,i}{}^{(f)} - \boldsymbol{p}_{m,j}{}^{(k)}$.

To improve the statistics, the position, scale and angle key point parameters of the f-frames are residually encoded using as reference the matching reference frame key point. To further reduce the rate, the parameter residues are scalar quantized. For instance, the angle parameter residue of the matching key point pair, $c_\theta = \theta_{n,i}{}^{(f)} - \theta_{m,j}{}^{(k)}$, is quantized as:

$$\text{round}\left(\frac{\theta_{n,i}{}^{(f)} - \theta_{m,j}{}^{(k)}}{\Delta}\right)\Delta \qquad (3)$$

where Δ is the quantization step. Thus, the f-frame decoded angle parameter is given by $\hat{\theta}_{n,i}{}^{(f)} = \theta_{m,j}{}^{(k)} + \text{round}\left(\frac{\theta_{n,i}{}^{(f)} - \theta_{m,j}{}^{(k)}}{\Delta}\right)\Delta$ where $\theta_{m,j}{}^{(k)}$ for the reference frame is not quantized as it is extracted at the decoder. The same quantization step size, $\Delta = 0.25$, has been used for the position, scale and angle parameters.

The syntactic elements $\mathrm{r_k}, \mathrm{r_m}$ and $\mathrm{m_s}$ are coded using arithmetic coding with a uniform probability model. On the other hand, the key point parameter residues are coded using adaptive arithmetic coding with an initial statistical model set up for each parameter using a training sequence different from those coded. One can roughly estimate the rate associated to each syntactic element by considering a coding set up using 2 reference frames, a maximum of 256 key points per frame, 16 multiplicative scale factor values, CIF resolution and a maximum scale value of 120. In the worst case, using a uniform probability model for each syntactic element, the coding of each key point match would require 1 bit for $\mathrm{r_k}$, 8 bits for $\mathrm{r_m}$, 4 bits for $\mathrm{m_s}$, 23 bits for the position residue, 12 bits for angle parameter residue and 10 bits for the scale parameter residue, in a total of 58 bits per key point match. With entropy coding using an adaptive probability model, the rate can be significantly lowered. In fact, on average, 11 key point matches are selected for each f-frame at the average cost of 26 bits.

IV. Performance Assessment

This section presents the RD performance for the proposed FBVC solution using meaningful test conditions.

A. Test Conditions

To appropriately assess the proposed codec, the following test conditions have been adopted:

- Two surveillance video sequences: Hall and Container, CIF spatial resolution at 30Hz, 10 seconds long.
- GOP size 2 (i.e. structure k-frame, f-frame, k-frame).
- QP values for k-frame coding: 45, 40, 37, 34, 40 and 25.
- Maximum 256 SIFT features per frame; step size 0.25 for each key point parameter residue quantization.
- The reference frames buffer includes two reference k-frames (one past and one future) and one reference (uncompressed) frame with related content (in this case from Container sequence when coding Hall, and vice-versa) as well as an initial 'background' frame from the coding sequence.
- Coding benchmarks: HEVC Main profile in all Intra and IBI prediction configurations with HM reference software, version 16.3.

B. RD performance

In Fig. 2 and Fig. 3, the RD performance of the proposed FBVC solution is presented for two video surveillance sequences. In the figures, Motion Compensated Frame Interpolation (MCFI) refers to the initially interpolated f-frames using the algorithm in [10]; here the rate corresponds only to the k-frames. For the lower rates, the proposed FBVC solution has a competitive RD performance and even outperforms HEVC IBI. As, for the lower rates, HEVC IBI does not code many transform residues, the fact that the proposed FBVC solution does not code texture residues at all does not have a negative impact. At higher rates, the proposed coding solution performs slightly worse than HEVC IBI although this difference may not be perceptually relevant. This difference is associated to the difficulty of the proposed solution to reach very high qualities as no texture residuals are coded for the f-frames; this means that the image patches inherited from the reference frames are not fully capable of representing the texture innovation in the f-frames but this does not seem to be a major problem considering the high qualities still reached.

Table I shows the Bjontegaard Delta-Rate for the proposed FBVC solution regarding the two benchmarks. The rate reductions are significant, going up to 5.4% and 55% for HEVC IBI and Intra, respectively. It is essential to highlight here that the proposed FBVC solution not only provides a better RD performance regarding the relevant benchmarks but it is also search friendly by providing uncompressed domain key points for the f-frames which not only reduces the search complexity but it also potentiates its efficacy.

Table 1. Bjontegaard Delta-Rate

FBVC (GOP 2) w.r.t	**Container**	**Hall**
HEVC (IBI)	-5.41%	-4.279%
HEVC (I)	-55.3029%	-50.0575%

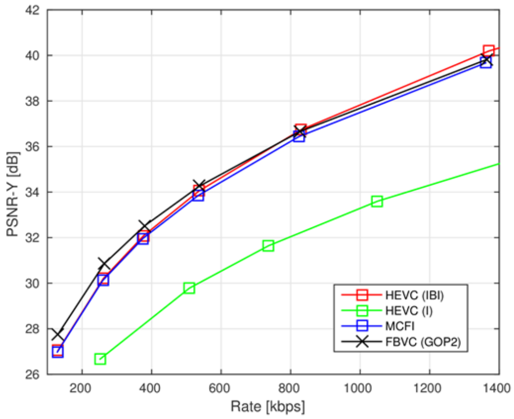


Fig. 2 – RD performance for the *Container* sequence

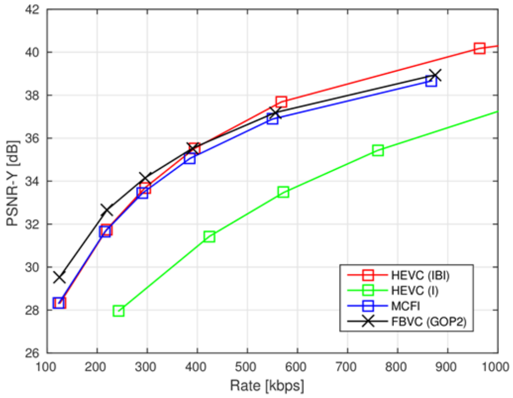


Fig. 3 – RD performance for the *Hall* sequence

V. Conclusions

This paper proposes a novel feature-based video coding solution that provides not only the usual visualization capabilities but it also potentiates simpler and better searching. The experimental results show that the proposed solution outperforms HEVC IBI and Intra for video surveillance like sequences with rate reductions around 5% and 50%, respectively. Future work will consider the scenario where the feature descriptors themselves will be efficiently encoded for the f-frames.

Acknowledgement

The authors would like to thank CNPq and FAPEAM for financial support.

Reference

[1] B. Girod *et al.*, "Mobile visual search," *IEEE Signal Processing Magazine,* vol. 28, no. 4, pp. 61-76, June 2011.

[2] L. David, "Distinctive image features from scale-invariant keypoints," *Intern. Journal of Computer Vision,* vol. 60, no. 2, pp. 91-110, November 2004.

[3] L. Baroffio *et al.*, "Coding visual features extracted from video sequences," *IEEE Trans. on Image Processing,* vol. 23, no. 5, pp. 2262-2276, April 2014.

[4] P. Weinzaepfel, H. Jégou and P. Pérez, "Reconstructing an image from its local descriptors," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, US, June 2011.

[5] H. Yue *et al.*, "SIFT-based image compression," in *IEEE Intern. Conference on Multimedia and Expo*, Melbourne, July 2012.

[6] H. Yue *et al.*, "Cloud-based image coding for mobile devices—toward thousands to one compression," *IEEE Trans. on Multimedia,* vol. 15, no. 4, pp. 845-857, January 2013.

[7] L. Baroffio *et al.*, "Hybrid coding of visual content and local image features," in *IEEE Intern. Conference on Image Processing (ICIP)*, Quebec City, September 2015.

[8] J. Chao and E. Steinbach, "Keypoint encoding for improved feature extraction from compressed video at low bitrates," *IEEE Trans. on Multimedia,* vol. 18, no. 1, pp. 25-39, November 2015.

[9] G. Sullivan *et al.*, "Overview of the high efficiency video coding (HEVC) standard," *IEEE T-CSVT,* vol. 22, no. 12, pp. 1649-1668, September 2012.

[10] J. Ascenso *et al.*, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," in *5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, Smolenice, July 2005.

[11] P. Pérez *et al.*, "Poisson image editing," *ACM Trans. on Graphic,* vol. 22, no. 3, pp. 313-318, July 2003.