COCKPIT VIDEO CODING WITH TEMPORAL PREDICTION

Iulia Mitrica, Attilio Fiandrotti, Marco Cagnazzo, Eric Mercier, Christophe Ruellan

COCKPIT VIDEO CODING WITH TEMPORAL PREDICTION

Iulia Mitrica, Attilio Fiandrotti, Marco Cagnazzo — Télécom ParisTech, Université Paris Saclay Paris, France

Eric Mercier, Christophe Ruellan — Zodiac Data Systems Paris, France

ABSTRACT

Recording an airplane cockpit screen is a challenging task since video codecs hardly preserve text details at the low bitrates required by avionic applications. We recently proposed a scheme for semantic compression of airplane cockpit video that preserves the readability of text while meeting bitrate and encoder complexity constraints. Within each frame, text is segmented from the video and encoded as character strings rather than as pixels. Text in the screen is then inpainted, producing a residual video with few high frequency components easily encodable with standard codecs. The residual video is transmitted with the encoded text as side-information. At the receiver side, characters are synthesized atop the decoded residual video, leaving the text unaffected by compression artefacts. In this work, we evaluate our scheme with multiple video codecs with different prediction schemes, producing novel experimental evidence in terms of attainable rate-distortion performance and highlighting directions for future research.

***Index Terms*—** HEVC, AVC, screen content coding, airplane cockpit video, low bitrate, character recognition, semantic video coding, convolutional neural networks

1. INTRODUCTION

Recording the cockpit screen of an airplane is emerging as a novel application of video compression applied to the avionic domain. In modern airliners, direct access to the plane sensors and devices (GPS navigator, fuel level gauge, etc.) is often impossible owing to multiple reasons. So, the only way to record key flight information, e.g. for post-accident investigations, is capturing and storing the content of the cockpit screen as a video. An airplane cockpit screen typically consists in a number computer generated graphics superimposed either on monochrome background (e.g., virtual gauges) or natural images (e.g., navigation maps). Recording the content of an airplane cockpit as a video poses however a number of technical challenges. First, the on-board storage capability is usually limited at the order of gigabytes, allowing at most of some tens of hours of recording which demands low bitrate encoding. Second, compression artefacts shall not affect the readability of computer-generated contents such as letters and digits due to the security and safety reasons. Third, the video encoder complexity is typically limited by safety norms on maximum power consumption and heath dissipation.

Airplane cockpit screen coding represents a special case of screen coding, for which a number of techniques have been proposed. Some techniques consider each image as a compound of multiple blocks that are compressed using different tools according the block type (natural or computer-generated blocks) [1, 2, 3, 4]. Computer-generated blocks (e.g., text) are encoded lossless, whereas natural blocks are compressed with lossy techniques. Notice that the rate-distortion performance depends not only on the dimensions of those blocks, but also on the used block classification method. The recently approved Screen Content Coding (SCC) extension [5] of H.265/HEVC [6] standard offers ad-hoc tools for coding generic screen content. Nevertheless, our previous research showed that computer graphics are strongly affected by the compression artefacts at low bitrates required by this application, not to mention the complexity of H.265/HEVC.

In our preliminary research [11] we proposed a screen compression scheme where computer-generated graphics (characters, lines) are segmented from the video at the source. The characters are recognized via a Convolutional Neural Network (CNN) [8] and are encoded as text rather than as pixels. Characters are then encoded using a rate-efficient intra-frame predictive scheme and delivered to the receiver. Characters are removed from the video via inpainting [9] and the *residual video* is compressed with H.265/HEVC video codec. At the receiver, the characters are synthesized from scratch and overlaid on the decoded residual video, recovering the original stream. Removing high-frequency components from the video reduces the video bitrate, whereas the characters synthesized at the receiver are not affected by compression artefacts.

This work extends and improves upon the results of [11] in two ways. First, we explore the potential gains enabled by an inter-frame predictive scheme in the context of cockpit video compression. The method in [11] being intra-frame only, the gains were limited by the fact that we disregarded the temporal correlation among neighbouring frames. Second, we explore a lower complexity solution where the residual video is compressed via H.264/AVC instead of H.265/HEVC. Such solution addresses the need for an encoder [7] implementable in FPGA, while it enjoys the benefits of the well established licensing program of the H.264/AVC stan-

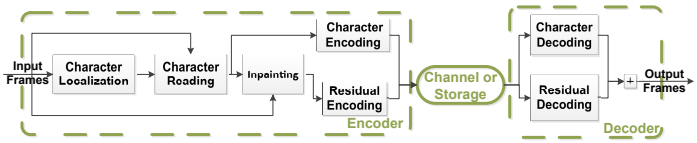

Fig. 1. Simplified representation of our proposed scheme for cockpit video compression. We refer the interested reader to [11] for further details.

dard. Our experiments show that H.264/AVC retrofitted with our semantic coding scheme is competitive with a reference H.265/HEVC encoder in rate-distortion sense, plus they highlight promising directions for further enhancing the performance of our scheme.

2. SEMANTIC COCKPIT VIDEO COMPRESSION

This section overviews the key features of our method for low-bitrate cockpit video semantic compression, illustrated in Fig. 1. We refer the interested reader to [11] for further details.

2.1. Character Localization

As a first step, we localize the characters in the screen exploiting some features of cockpit screens to keep complexity low. Namely, characters aspect in cockpit screens changes depending on the background type (natural image or uniform colour) to improve readability. Characters are in fact outlined when superimposed on a natural images (typically, white text with black background), whereas there is no outline when the background is monochrome.

Therefore, we first coarsely classify each screen either as having natural background (*natural* screen) or monochrome background (*computer-generated* screen) as follows. Each frame is first subdivided in tiles and for each tile we compute the colour histogram: if most histograms are spiky, the screen is labelled as synthetic, otherwise the screen is labelled as natural.

Then, depending on the screen class, pixels corresponding to text are segmented with an appropriate thresholding algorithm. Concerning natural screens, characters are segmented leveraging the knowledge they have an outline. Two thresholding operations are combined, one on the letters colour and another one on the contour colour. Otherwise, the Otsu thresholding [10] algorithm is used for computer-generated screens. Both thresholding algorithms produce a binary *threshold mask* where , e.g. white, pixels correspond to characters.

Further, Connected Components Analysis (CCA) clusters the white pixels in the threshold mask assigning the same label to pixels in the same *neighbourhood*. E.g., the comma and the dot forming a semicolon are assigned the same label because they belong to the same letter.

Next, a rectangular bounding box is casted around each pixel cluster with identical label, representing the location of each candidate character as a rectangular bounding box.

The output of the character localization algorithm is finally a set of bounding boxes where each box represents a candidate letter or digit.

2.2. Character Reading via Convolutional Neural Network

Each potential character detected above is recognized using a neural network. In [11] we explored three different architectures with different performance-complexity trade-offs. We describe here the solution based on the *LeNet5* architecture [8], which showed a character recognition accuracy of 99.84% at reading computer generated characters and with bearable complexity for FPGA implementation. The network is composed by two convolutional layers and three fully connected layers. Each convolutional layer includes 6 and 16 5×5 filters each layer followed by a max-pooling feature map subsampling layer. The first connected layer includes 120 units whereas the second layer includes 84 units. The output layer finally includes 41 units as the network classifies each character image according to C=41 labels (letters, digits, symbols plus one *no character* class). We train the network over character samples we extracted from a set of hand-annotated airplane cockpit videos. Training samples are augmented by randomly shifting each character to achieve robustness to character localization errors. The networks is trained to minimize the classification error across the classes using the SGD method over batches of 128 samples and with a learning rate of 10^{-2}. The network outputs a most likely class for each potential localized character, or it rejects the input in the case of, e.g., background clutter erroneously localized as a character, achieving robustness against noisy backgrounds.

2.3. Predictive Character Encoding

For each video frame, detected characters are encoded together with their coordinates and aspect information using a rate-efficient predictive scheme. In typical cockpit screen videos, we observed that the probability distributions of the characters coordinate differences are very spiky. So, our scheme exploits the regularity of each set of characters by differentially encoding their horizontal and vertical coordinates as follows. First, because characters are aligned in rows, the vertical coordinate of two successive characters are either identical or differ by $\pm$ one pixel. Thus, just 4 different symbols are required to signal the differentially encoded vertical coordinate: 3 of them indicate the same coordinate of the previous character accounting for the localization noise, while a fourth symbol will signal that the coordinate will be explicitly signalled. Second, on the horizontal axis, the characters are often shifted with a constant number of pixels corresponding to the character width, c_w (plus some spacing).

This allows us to differentially encode this coordinate by using three symbols: one for the default width c_w, a second that allows to account for a one-pixel tolerance (*i.e.*, it encodes c_w+1) and a third that indicates explicit signalling will follow. Our previous experiments showed that our scheme allows to reduce the characters coding rate to less than an average of 10 bits per character for different real airplane cockpit test video sequences.

2.4. Residual Video Compression

As a final step, the text is erased from the cockpit video and the resulting *residual video* is compressed. Each pixel in the threshold map is inpainted using a low complexity colour inpainting technique [9]. Inpainting fills each character pixel with the most likely colour according to its neighbourhood. The inpainting result is a *residual video* without computer-generated graphics, and thus with fewer high frequency components to encode. The residual video is then compressed with some standard video codec (see Sect. 3.1) and, for example, stored on a removable support. The encoded characters are stored alongside the compressed video. At the receiver side, the inpainted residual video is first recovered. Then, for each frame, we synthesize the characters that were transmitted as side information. The result is a cockpit video sequence where characters are not affected by compression artefacts, thus preserving their readability.

3. EXPERIMENTAL RESULTS

In this section we experiment with our cockpit video semantic compression scheme over multiple video sequences and using different codecs and video compression schemes.

3.1. Experimental Setup

We experiment over the six cockpit video sequences illustrated in Fig. 2. Those sequences can be classified according to the type of the background. Two of them (Seq. 5 and 6) have complex computer-generated text and graphics overlaid on black background. Four of them (Seq. 1, 2, 3 and 4) include text superimposed on natural background, which can be either captured with surveillance cameras installed outside the airplane operating in the visible light (Seq. 1) and infrared spectrum (Seq. 2), or it can be synthetic represented by H.265/HEVC test sequences *Cactus* (Seq. 3) and *Park* (Seq. 4). The purpose of using the latter two sequences is to stress the resilience of the character detector to background clutter. Table 1 summarizes the characteristics of our test sequences. Each sequence is processed according to our semantic compression scheme [11] summarized in the previous section (*Proposed* scheme, in the following).

Concerning the residual video compression in Sec. 2.4, we recall that we operate in an embedded avionics scenario

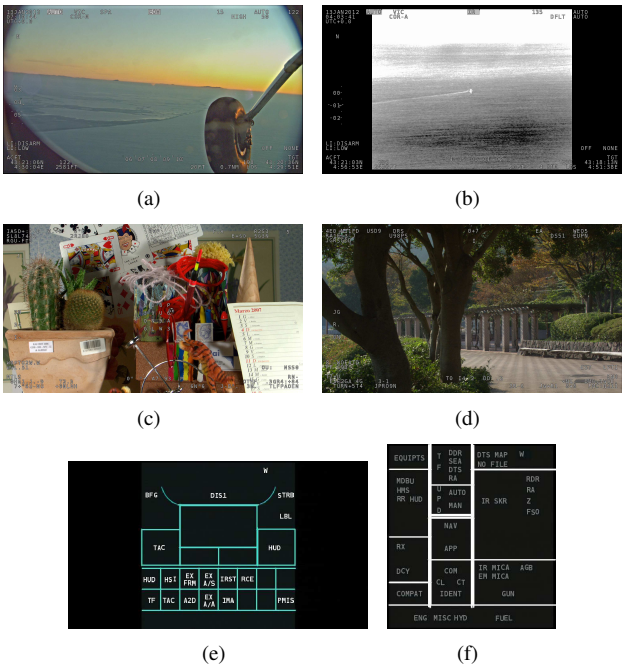

Fig. 2. The six airplane cockpit screens video sequences used in our experiments (see Tab. 1 for the relative characteristics).

Table 1. Characteristics of our six test sequences.

# Seq.	Resolution	Frame Rate	Background
1 (Fig. 2(a))	1920x1080	24 fps	Natural
2 (Fig. 2(b))	1920x1080	24 fps	Natural
3 (Fig. 2(c))	1920x1080	24 fps	Natural
4 (Fig. 2(d))	1920x1080	24 fps	Natural
5 (Fig. 2(e))	720x576	24 fps	Black
6 (Fig. 2(f))	720x576	24 fps	Black

where the overall power consumption must be kept under control. As the codec used to compress the residual video is one of the major computational complexity sources, Tab. 2 estimates the complexity of some H.265/HEVC and H.264/AVC encoders implemented in FPGA technology. The first two rows compare two implementations of H.265/HEVC and H.264/AVC encoders in inter-mode, showing that H.264/AVC power consumption is one third of its H.265/HEVC counterpart. The last two rows detail two all-intra and inter-enabled low-memory implementations of the H.264/AVC codec: in this case the H.264/AVC complexity can be less than one fifth the H.265/HEVC counterpart. Thus, while AVC compression efficiency is lower than H.265/HEVC, its lower complexity makes it an interesting option for compressing the residual video in our power-constrained scenario. For this reason, in the following we experiment both with the HEVC/H.265 codec (HM-16.14) plus its Screen Content Coding extension (SCM-8.3) and with the earlier yet less complex H.264/AVC codec (JM 19.0).

Table 2. FPGA complexity estimate for H.264/AVC and H.265/HEVC encoders (1920x1080, 30 fps).

Codec	Logic [kALM]	RAM [kbits]	Power [W]
AVC-Inter [13]	30-60	5000	< 1
HEVC-Inter [14]	90	10000	< 3
LowMem AVC-Intra [12]	5	58	0.5
LowMem AVC-Inter [12]	8.6	114	0.6

In the following, we refer to our proposed scheme with H.265/HEVC and with H.264/AVC compression of the residual video as *Prop-HEVC* and *Prop-AVC* respectively.
As reference schemes, we consider the case where each sequence is encoded with the standard H.265/HEVC codec and its SCC extension (*HEVC* and *SCC*, respectively) and H.264/AVC codec (*AVC*). We recall that with such schemes characters are encoded as pixels exactly as the rest of the screen, so such schemes do allow us to evaluate the effect of compression artefacts on the characters readability.
We evaluate the rate-distortion (called R-D) performances of each scheme at different quantization (QP) values corresponding to high-bitrates (QP from 20 to 45 with steps of 5) and low-bitrates (QP from 45 to 51 with steps of 1) ranges. We objectively evaluate the quality of the recovered video in terms of PSNR, plus we visually inspect the characters in the recovered video to assess how compression artefact impair their readability.

3.2. Experiments with All-Intra Coding

To start with, we experiment in the case where the video codec operates in Intra-only mode, i.e. no temporal correlation whatsoever is exploited. Despite the reduced compression efficiency, Intra-coding saves on computational complexity by skipping motion search and on memory complexity not having to keep the decoded frames in the reference buffer.

Fig. 4 shows the rate-distortion curve for each video sequence and each compression scheme.
Concerning the three reference schemes, HEVC has better R-D performance than AVC by reason of having better coding tools. In turn, SCC outperforms HEVC due to the coding tools tailored for screen compression. SCC performs particularly well especially with the two sequences in Fig. 4(e) and 4(f) which consists entirely of computer graphics on a black background (and thus the PSNR close to 50 dB). However, Fig. 3 (left, centre) shows that neither HEVC nor SCC preserve the readability of the characters at high QP values. The characters are hardly readable because the coarse quantization of the transform coefficients at high QP values makes impossible to correctly recover high frequency primitives such as edges, generating compression artefacts.
Concerning our two proposed schemes, Prop-AVC shows constantly better R-D performance than its AVC counterpart in Fig. 4. Most interesting, the experiments reveal that Prop-AVC outperforms HEVC in all natural sequences at medium to low coding rate and outperforms even SCC for low bitrate. Our explanation of these unexpected results is as follows. First, in our proposed scheme, only the inpainted residual video is encoded, so there are no high-frequency components to encode, making it more suitable to be coded with AVC. Therefore, the coding rate of the video is significantly reduced due to the fewer high-frequency elements to encode. Second, at the decoder, the characters are synthesized and overlaid on the decoded residual video and the original-decoded PSNR is computed. So, our proposed method achieves far better PSNR in character areas than the reference schemes and thus overall. In the same time, HEVC pays the cost in rate and distortion of preserving the shapes of computer-generated text. Fig. 3 (right) confirms that Prop-AVC preserves the readability of the text, since the synthesized characters are not affected by compression artefacts. That is, these experiments reveal that proposed semantic compression scheme allows the H.264/AVC codec to outperform the more recent and more complex H.265/HEVC codec both in terms of background video quality and characters readability, whereas it performs close to its SCC extension. Finally, the Prop-HEVC curves show that our method allows plain H.265/HEVC to outperforms even its specialized SCC extension, as already verified in our previous research.
Concerning the different classes of video sequences, our proposed method achieves the best results over the sequences 5 and 6 (Prop-AVC largely outperforms even SCC). These sequences contain only computer-generated graphics over black background, thus the residual video to encode is almost all black, explaining the large gains offered by our scheme. Sequences 1 and 2 obtain consistent gains, however these gains are lower (albeit in excess of 5 dB at most) since their natural background with moderate motion is more complex to encode than a black background. Finally, sequences 3 and 4 show the least improvements due to the high amount of motion details in the background video, which makes the savings in high frequency elements encoding less relevant with respect to the cost of encoding the background in Intra mode.
Notice that for the the Prop-AVC and Prop-HEVC schemes, the rate includes also the rate of the encoded characters. Such rate is equal to about 1.7 kbit per frame, i.e. about 41 kbit/s (for natural screens), which represents a negligible fraction of the overall video rate.

3.3. Experiments with Inter-frame Coding

Next, we repeat our experiments allowing the H.264/AVC and the H.265/HEVC video codecs to exploit the temporal correlation among adjacent frames in the (residual) video. In order to keep the encoder computational complexity low, we use a low-delay configuration, with a Group of Pictures (GOPs) of 4 frames. Also, the encoder is allowed to keep just one frame in the decoded picture buffer to keep low the memory com-

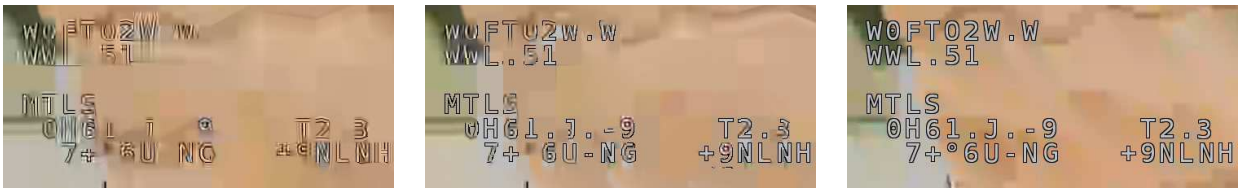

Fig. 3. Reconstruction artefacts at QP=50 for All-Intra coding configuration. Left: HEVC (PSNR 25 dB); Center: SCC (PSNR 25.6 dB); Right: Prop-AVC (PSNR 26.7 dB). Our proposed scheme preserves the characters readability since they are not affected by compression artefacts, resulting in better PSNR.

plexity.

Fig. 5 shows the corresponding rate-distortion curves. First and unsurprising, exploiting temporal correlation largely improves the rate-distortion performance of all schemes when compared with the corresponding graphs in Fig. 4. The better temporal prediction tools of H.265/HEVC clearly enable better R-D performance than H.264/AVC. However, when we compare the Prop-AVC and Prop-HEVC schemes with the corresponding references, we observe different trends with respect to the Intra-only case. In particular, we observe that for sequences 5 and 6 (Fig. 5(e) and 5(f)), Prop-AVC outperforms HEVC only for a subset of the tested QP values. By comparison, in Intra-only mode Prop-AVC outperformed even SCC at any bitrate. Also, Prop-AVC cannot achieve the low bitrates achieved by HEVC any more. In any case, SCC outperforms both Prop-AVC and Prop-HEVC. Looking at sequences 1 and 2, Prop-AVC does not outperform HEVC but for very few QP values and only for the second sequence. We explain these results as follows. The video bitrate, in our scheme, accounts both for the residual video coding rate and for the encoded characters rate. The characters rate is identical to the Intra-only experiments since no temporal correlation is exploited in our character encoding scheme. As the residual video is encoded exploiting temporal correlation, the ratio between characters rate and residual video rate increases. In this scenario, the characters rate is not negligible any more with respect to the residual video rate, explaining the less competitive performance of our scheme when temporal prediction is enabled. These results call for exploiting the temporal redundancy between co-located text in temporally adjacent video frames. Our analysis of real cockpit video sequences verified the intuition that text co-located characters in neighbour frames are strongly correlated. We postulate that by extending our semantic encoding scheme to exploit temporal redundancy, we could drastically reduce the character rate, making it negligible with respect to inter-predicted residual video.

Finally, we also evaluate the performance of our Proposed-SCC scheme with respect to the SCC scheme for the five class-A sequences using the Bjontegaard metrics. Tab. 3 shows that our proposed method achieves consistent R-D gains with respect to the reference when the Inter prediction is enabled. Depending on the considered QP ranges, our gains range from $18\,\%$ for low QPs values to $57\,\%$ for high QP values.

Table 3. Bjontegaard metrics (BD-PSNR and BD-RATE) for class-A sequence, computed for Proposed-SCC vs. SCC codecs using Inter coding configuration.

	High Quality $QP = [40, 35, 30, 25]$		Low Quality $QP = [50, 45, 40, 35]$	
Video	BD−PSNR	BD−Rate	BD−PSNR	BD−Rate
Seq. 1	0.69 dB	-34.3%	3.05 dB	-40.0%
Seq. 2	0.43 dB	-17.9%	1.49 dB	-30.0%
Seq. 3	1.66 dB	-38.2%	3.66 dB	-57.1%
Seq. 4	1.31 dB	-31.2%	2.60 dB	-55.5%

4. CONCLUSIONS AND FURTHER WORK

In this work, we experimentally evaluated our semantic scheme for airplane cockpit video compression with two different video codecs. Availing different coding configurations, several interesting results emerge.

When the video is encoded in Intra-only mode, the H.264/AVC codec retrofitted with our proposed scheme outperforms the more recent H.265/HEVC codec and performs close to its SCC extension. Thus, we can exploit the modular structure of the semantic coding scheme to improve the coding efficiency keeping the complexity suitable for avionic applications.

Contrary, when temporal inter-frame prediction is used, the competitive advantage of our semantic video compression scheme decreases as the rate of the encoded characters is not negligible any more with respect to the rate of the residual video.

5. REFERENCES

[1] R. L. de Queiroz, "Compression of compound documents," in *Image Proc., 1999, ICIP*, vol. 1, pp. 209–213.

[2] M. Cagnazzo, S. Parrilli, G. Poggi, and L. Verdoliva, "Costs and advantages of object-based image coding with shape-adaptive wavelet transform," *EURASIP J. Image Video Proc.*, vol. 2007, doi:10.1155/2007/78323.

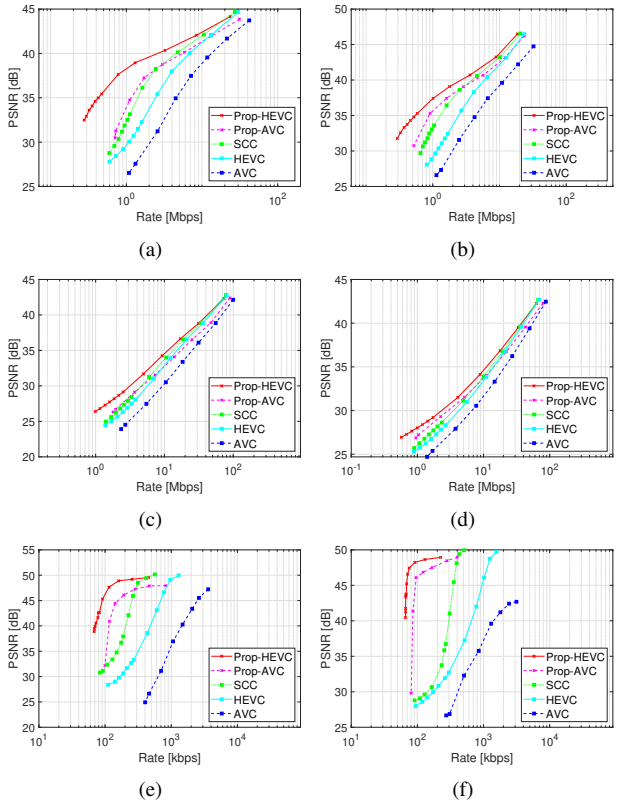

Fig. 4. PSNR vs. video bitrate for All-Intra coding configuration. First row - Left: Seq. 1 (Fig. 2(a)); Right: Seq. 2 (Fig. 2(b)). Second row - Left: Seq. 3 (Fig. 2(c)); Right: Seq. 4 (Fig. 2(d)). Third row - Left: Seq. 5 (Fig. 2(e)); Right: Seq. 6 (Fig. 2(f)).

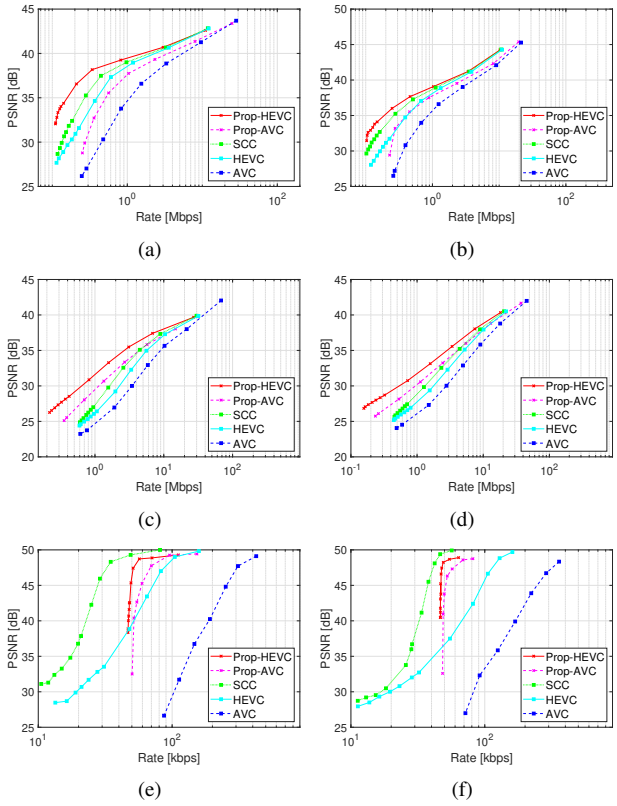

Fig. 5. PSNR vs. video bitrate for Inter coding configuration. First row - Left: Seq. 1 (Fig. 2(a)); Right: Seq. 2 (Fig. 2(b)). Second row - Left: Seq. 3 (Fig. 2(c)); Right: Seq. 4 (Fig. 2(d)). Third row - Left: Seq. 5 (Fig. 2(e)); Right: Seq. 6 (Fig. 2(f)).

[3] T. Lin and P. Hao, "Compound image compression for real-time computer screen image transmission," *IEEE Trans. Image Processing*, vol. 14, pp. 993–1005, 2005.

[4] W. Ding, D. Liu, Y. He, and F. Wu, "Block-based fast compression for compound images," in *Proceed. of IEEE Intern. Conf. on Multim. and Expo*, pp. 809–812.

[5] R. A. C. Jizheng Xu, Rajan Joshi, "Overview of the emerging HEVC screen content coding extension," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, pp. 50–62, Jan. 2016.

[6] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, pp. 1649–1668, 2012.

[7] F. Bossen, B. Bross, K. Suhring, D. Flynn, "HEVC complexity and implementation analysis," *IEEE Trans. Cir. and Sys. for Video Tech.*, vol. 22, pp. 1685–1696, 2012.

[8] Y. Le Cun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] C. Guillemot, O. Le Meur, "Image inpainting: overview and recent advances," *IEEE Signal Proc. Mag.*, 2014.

[10] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man.*, vol. 9, 1979.

[11] doi:10.1109/TMM.2019.2900168

[12] Cast, "H264-E-BPS: Low-Power AVC/H.264 Baseline Profile Encoder," Tech. Rep., Cast, 2019.

[13] SoC Technologies, "H.264 HD Video Encoder IP Core," Tech. Rep., SoC Technologies, 2019.

[14] SoC Technologies, "H.265/HEVC HD Encoder IP Core,"Tech. Rep., SoC Technologies, 2019.