Drone infrared video coding using virtual view generated from iteratively constructed aerial map and historical data

1st Evgeny Belyaev
International Laboratory "Computer Technologies"
ITMO University
Saint-Petersburg, Russia

2nd Søren Forchhammer
Department of Photonics Engineering
Technical University of Denmark
Kgs. Lyngby, Denmark

***Abstract*—This paper is dedicated to drone infrared video coding. First, we assume that a set of drone infrared video sequences corresponding to the same area (historical data) are collected during previous flights. Applying a stitching algorithm to the historical data we build a map of the area (historical map) and store it in the drone memory. Second, during the drone flight we compress input frames and build a current map of the area via stitching of the decoded frames. Finally, we utilize a Multi-view H.265/HEVC encoder, where the virtual view generated from both the aerial maps is used for inter-view prediction of the input video, which is considered as a second view. Experimental results obtained for real-life drone infrared videos show that comparing to the H.265/HEVC the proposed algorithm provides from 8.15 to 30.81% bit rate savings.**

***Index Terms*—drone video coding, infrared video**

I. Introduction

In recent years more and more video data are acquired by infrared cameras mounted on unmanned airborne vehicles such as quadcopters, hexacopters or light fixed wing planes called *drones*. In many applications, such as leakage detection in district heating systems [1], or damages detection of solar panels fields [2], an infrared video should be transmitted to a ground control device in real-time over a wireless channel in order to provide visual input for drone navigation and have an opportunity to immediately change the drone trajectory when any interesting event or object is detected. Herewith, due to the channel capacity limitations, the infrared drone video should be efficiently compressed before the transmission.

As a basic solution, the latest standard H.265/HEVC [3] can be used for an infrared video coding. However, usually drone video footage exhibits camera rotation which cannot be well estimated by block-based motion estimation used in HEVC. Moreover, a drone can periodically fly above the same area, but the motion prediction scheme in HEVC does not exploit similarity between frames having significant difference in time of capturing. Finally, we can also assume that a set of video sequences recorded during two or more flights above the same area contain similar frames, i.e., a set of previously recorded video sequences can be used for efficient coding of a new video sequence. However, this is also out of the scope of the standard. In order to address these issues, in [4] a set of previously encoded video sequences (called *historical data*) captured by a dashboard camera mounted on a single vehicle is used for compression of a current video. First, the historical data is used to generate the most similar frame for each frame of the current video. Then the generated frames are considered as a base view of the 3D-HEVC encoder, while the current video is considered as a second view and encoded utilizing interview prediction. As a result, in average 30% bit rate savings is achieved. The main drawback of this approach is high computational complexity caused by the generation of similar frames, i.e., it can be used only in off-line applications. In [5], [6] it was proposed to use Google Earth data as historical data for satellite video coding. Here, the most similar frames extracted from the historical data are used for prediction of I frames only. As a result, overall bit rate savings from 10 to 22% are reported. Additional drawback of this approach is that the inter-prediction used for I frames is not compatible with HEVC standard.

In this paper we first build an historical aerial map utilizing available historical data and store it in both encoder and decoder memories. Then we use global motion estimation (GME) to extract the most similar frame from the map. The extracted frames are considered as a base view of Multi-view H.265/HEVC encoder (MV-HEVC), while the current video is considered as a second view. Since, the HEVC multilayer extensions support the base layer being coded by other codecs, the extracted frames generator can be considered as a "other codec", i.e., the base layer should not be encoded by MV-HEVC, while the second view containing the current video should be compressed by a MV-HEVC compatible scheme. Finally, we also assume that the historical aerial map will not always cover the drone trajectory. In such case, similar to our previous work [7] we use a simple stitching algorithm to build a current aerial map from already decoded frames, and use it as well in order to take into account camera rotation as well as similarity between frames captured at different moments of

This research was supported in part by the Government of the Russian Federation through the ITMO Fellowship and Professorship Program and in part by the Danish energy technological development and demonstration program (EUDP), EUDP 15-I, 64015-0072.

time. Experimental results obtained for real-life drone infrared videos show that comparing to the H.265/HEVC the proposed algorithm provides from 8.15 to 30.81% bit rate savings.

The rest of the paper is organized as follows. Section II introduces the proposed compression algorithm. Section III analyzes rate-distortion performance of the proposed approach and Section IV concludes the presented results.

II. Proposed coding scheme

In this paper, we assume that a drone on-board video platform includes two video encoders compressing the same video input: the first encoder is needed for storage of input infrared video sequence at near-lossless quality into the drone memory, while the second one is needed for real-time video streaming from the drone to a ground control device. We also assume that a geographic coordinates system is mounted on the drone, so that the drone altitude h_i, latitude φ_i and longitude λ_i at the moment of capturing of frame F_i is embedded into a video bit stream.

After each flight, near-lossless video from the drone memory is stored on a ground computer for further processing. Let us call a set of collected near-lossless infrared video sequences as *historical data*. This historical data is used to build a *historical aerial map* using any known video frame stitching algorithm. Moreover, since the stitching is performed in off-line mode by the ground computer, we assume that even high complex stitching algorithms can be applied to build the historical aerial map. In contrast, as it was mentioned in Introduction, a *current map of the area* is constructed using stitching of the decoded frames in real-time on drone, i.e., only low-complexity stitching algorithms can be used. In this paper we use the following stitching approach presented in Algorithm 1. First, we apply GME between current frame F_i and previous frame F_{i-1} and obtain relative displacements $\Delta x, \Delta y$ and rotation angle $\Delta\alpha$ (line 5). Then, we apply GME between frame F_i and aerial map A utilizing $\hat{x}_i = x_{i-1} + \Delta x$, $\hat{y}_i = y_{i-1} + \Delta y$, $\hat{\alpha}_i = \alpha_{i-1} + \Delta\alpha$ as an initial estimates (line 7). As a result, rotation angle α_i and coordinates x_i, y_i for frame F_i within map A are determined. Finally, we update map A utilizing mask-based image blending from [9] (lines 9–11): each pixel in the map A is computed as a weighted sum of all corresponding pixels having the same coordinates in the map. The weight matrix W contains a weight of each pixel depending on its position within a frame: pixels which are closer to a frame center have higher weights. Here, operator $\mathbb{M}(F, x, y, \alpha)$ creates a zero matrix of the same size as A, rotates frame F by angle α and inserts it with coordinates x, y into the created matrix. In line 11, the division is performed in an element by element way, and δ is a small value preventing division by zero.

Figure 1 shows the proposed drone infrared video encoding scheme using virtual view generated from iteratively constructed aerial map and historical data. In this scheme we use MV-HEVC encoder, where view 1 is an input infrared video sequence and view 0 is a virtual view generated utilizing both the historical area map and the aerial map iteratively

Algorithm 1 : Aerial map construction

1: $Q \leftarrow \mathbf{0}, U \leftarrow \mathbf{0}$
2: $x_1 \leftarrow 0, y_1 \leftarrow 0, \alpha_1 \leftarrow 0$
3: **for** $i = 1, ..., M$ **do**
4: **if** $i > 1$ **then**
5: $\{\Delta x, \Delta y, \Delta\alpha\} \leftarrow \text{GME}(F_i, F_{i-1})$
6: $\hat{x}_i \leftarrow x_{i-1} + \Delta x$, $\hat{y}_i \leftarrow y_{i-1} + \Delta y$, $\hat{\alpha}_i \leftarrow \alpha_{i-1} + \Delta\alpha$
7: $\{x_i, y_i, \alpha_i\} \leftarrow \text{GME}(F_i, A, \hat{x}_i, \hat{y}_i, \hat{\alpha}_i)$
8: **end if**
9: $Q \leftarrow Q + \mathbb{M}(F_i, x_i, y_i, \alpha_i) \circ \mathbb{M}(W, x_i, y_i, \alpha_i)$
10: $U \leftarrow U + \mathbb{M}(W, x_i, y_i, \alpha_i)$
11: $A \leftarrow \dfrac{Q}{U + \delta}$
12: **end for**

constructed from the decoded frames of view 1. The encoding process includes the following stages:

1) **Frame extraction from the historical area map.** We assume that the drone memory contains a historical area map and a historical table containing position $\{x_i, y_i\}$ of each historical frame H_i within the map, as well as its geographic coordinates $\{h_i, \varphi_i, \lambda_i\}$. Then, utilizing the geographic coordinates, for each input frame F_j we are searching for the nearest frame H_k in the historical table, scale the historical area map with factor $s = h_k/h_j$, and apply GME between frame F_j and the scaled historical aerial map using initial estimates $\hat{x}_j^h = s \cdot x_k$, $\hat{y}_j^h = s \cdot y_k$, $\hat{\alpha}_j^h = 0$. As a result, estimated coordinates $\{x_j^h, y_j^h, \alpha_j^h\}$ are used to extract frame P_j^h from the map.
2) **Frame extraction from the constructed area map.** At this stage we estimate coordinates $\{x_j, y_j, \alpha_j\}$ of frame F_j in the constructed aerial map (lines 5–7 of Algorithm 1) and use them to extract frame P_j^c from the map.
3) **Fusion and virtual view generation.** A pixel with coordinates (q, w) of virtual frame P_j^v is fused as

$$P_j^v(q,w) = \begin{cases} P_j^h(q,w), \text{ if } P_j^h(q,w) > 0, \\ P_j^c(q,w), \text{ otherwise.} \end{cases} \quad (1)$$

Since the historical area map are made from near-lossless frames which do not have coding artifacts, in (1) we assign higher priority to frame P_j^h. The computed frame P_j^v is added as a new frame for the view 0. The fusion process is illustrated on Figure 2.
4) **The constructed aerial map update.** Decoded frame $\hat{F}_j$ of view 1 is used to update the constructed aerial map via lines 9–11 of Algorithm 1.

Finally, we send a compressed bit stream corresponding to view 1 and side information including coordinates $\{x_j^h, y_j^h, \alpha_j^h\}$ and $\{x_j, y_j, \alpha_j\}$ to the receiver.

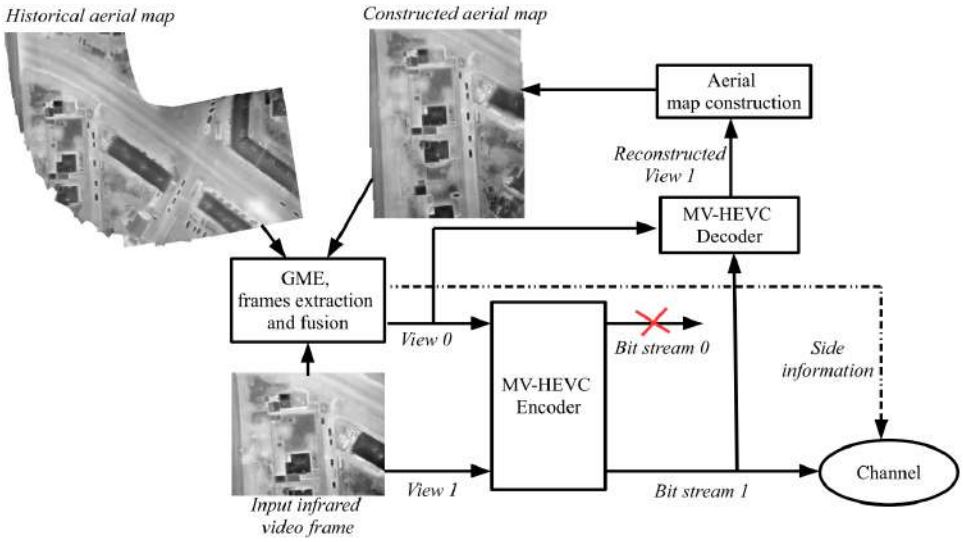


Fig. 1: Proposed encoding scheme based on aerial map prediction

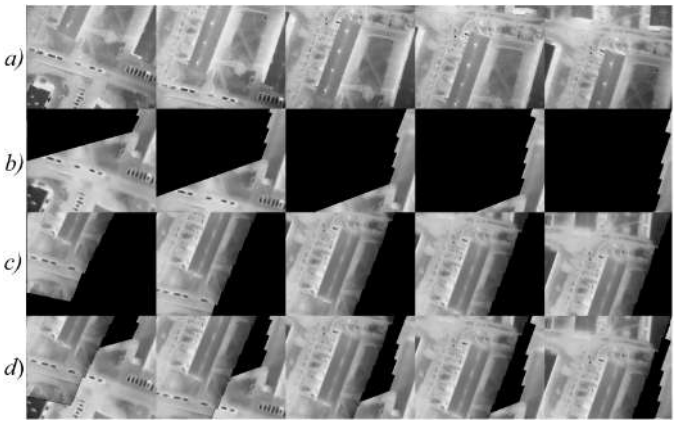


Fig. 2: *a)* Input frames, *b)* Virtual view generated from constructed aerial map, *c)* Virtual view generated from historical data, *d)* Virtual view generated by fusion of *b)* and *c)*

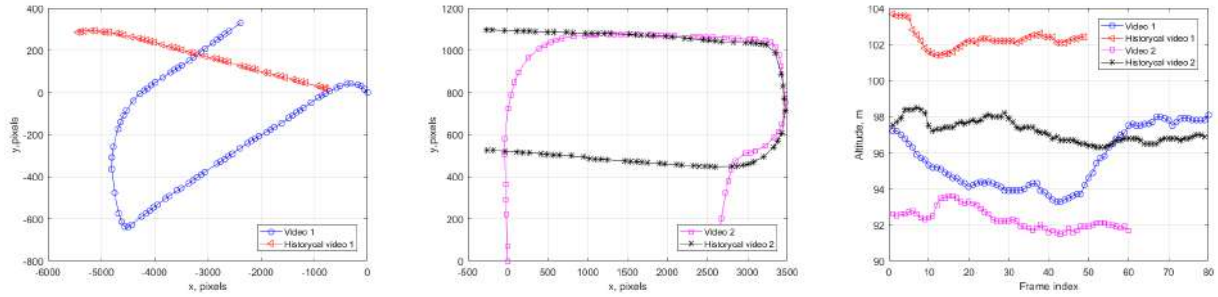
III. Performance evaluation

Experimental results were obtained for four test video sequences captured by Drone Systems ApS[1] by a Flir Tau2 infrared camera with frame resolution 640×512, frame rate 9 Hz. Herewith, two videos (*Video 1* and *Video 2*) were used to obtain rate-distortion results, while the remaining ones (*Historical video 1* and *Historical video 2*) were used as corresponding historical data. Figure 4 *a)*, *b)* shows planar coordinates corresponding to each frame. These coordinates were received from the Global Positioning System (GPS) installed on the drone and, for clarity, were mapped from latitude and longitude values to relative coordinates measured in pixels. Coordinates of the first frame of Video 1 (Figure 3 *a)*) and Video 2 (Figure 3 *b)*) were set to the origin, i.e., (0,0). From Figure 3 *a)*, *b)* it can be observed that approximately 50%[2] of frames in *Video 1* have an overlap with frames from *Historical video 1*, while almost all frames of *Video 2* have an overlapping with frames from *Historical video 2*. Figure 3 *c)* shows altitude values for each frame. One can see that the altitude is not stable and varies between 92 and 104 meters. For simplification, we use Algorithm 1 to build historical aerial maps as well. All the maps are illustrated in Figures 4 and 5.

Table I shows rate-distortion performance for different coding schemes implemented utilizing HTM-16.3 [10] which is a reference software of Multi-view extension of H.265/HEVC [3]. The software was used with GOP size 8. Each GOP was encoded separately. In order to avoid intra-frame coding, the last reconstructed frame of each GOP was used as a first frame of next GOP for both views. Peak Signal-to-Noise Ratio (PSNR) was selected as an objective quality metric. The quantization parameter (QP) for view 0 was set to zero. Hereby, only bit rate and PSNR obtained for view 1 were compared. Here, *HEVC* means each virtual frame P_j^v is zero-frame, *MV-HEVC+CAM* means each virtual frame is extracted from a constructed aerial map, i.e., $P_j^v = P_j^c$, *MV-HEVC+HAM* means each virtual frame is extracted from a historical aerial map, i.e., $P_j^v = P_j^h$ and *MV-HEVC+FAM* means each virtual frame is computed according to (1). Experimental results show that exploiting additional redundancy of a drone infrared video by the scheme *MV-HEVC+CAM* provides from 1.15 to 1.87% bit rate savings. Encoding scheme *MV-HEVC+HAM* provides relatively high bit rate savings, especially when a historical data is available for almost all input frames. For example, almost for all frames in Video 2 there exist corresponding frame extracted from the historical aerial map. As a result, 30.69% bit rate savings is achieved. When the historical data is not always available (*Video 1*), scheme *MV-HEVC+FAM* provides additional 1.21% bit rate savings comparing to *MV-HEVC+HAM*.

[1]Drone Systems ApS, http://dronesystems.dk/

[2]The overlap is relatively high, since the frame resolution is 640×512



(a) Coordinates for Video 1 (b) Coordinates for Video 2 (c) Altitude values for all videos

Fig. 3: Relative coordinates and altitude corresponding to each frame

		Video 1				**Video 2**			
Coding scheme	QP	25	30	35	40	25	30	35	40
HEVC	Bit rate, kbps	339.2	159.0	83.7	47.0	386.5	181.4	95.3	52.5
	PSNR, dB	37.22	35.43	33.37	31.14	36.90	35.00	32.87	30.59
MV-HEVC+CAM	Bit rate, kbps	337.1	157.2	82.5	45.9	380.9	178.4	93.5	51.0
	PSNR, dB	37.21	35.43	33.36	31.12	36.89	35.00	32.87	30.60
	BD-Rate [11]	**-1.15**				**-1.87**			
MV-HEVC+HAM	Bit rate, kbps	330.9	152.1	78.5	43.4	339.2	143.4	69.5	37.2
	PSNR, dB	37.21	35.44	33.45	31.35	36.90	35.15	33.34	31.53
	BD-Rate [11]	**-6.94**				**-30.69**			
MV-HEVC+FAM	Bit rate, kbps	329.4	151.1	77.3	43.0	338.1	143.0	69.4	37.1
	PSNR, dB	37.21	35.45	33.46	31.36	36.89	35.15	33.34	31.53
	BD-Rate [11]	**-8.15**				**-30.81**			

TABLE I: Rate-distortion performance comparison

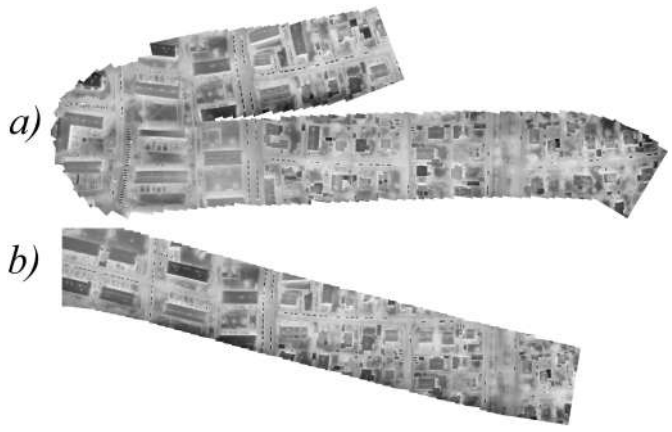


Fig. 4: Aerial map for *a)* Video 1, *b)* Historical video 1

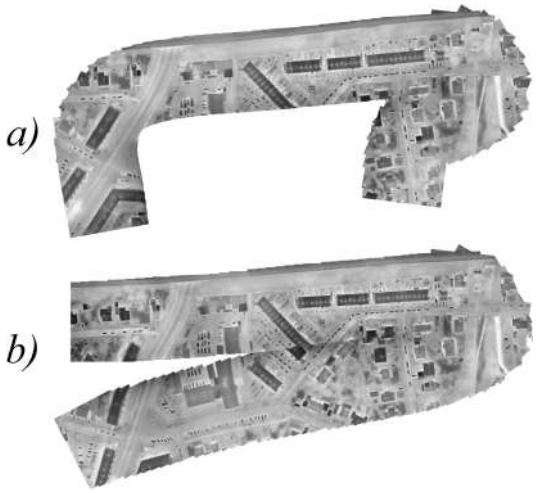


Fig. 5: Aerial map for *a)* Video 2, *b)* Historical video 2

At the encoder side the proposed *MV-HEVC+FAM* requires three GME, four frame rotations and one scale operation per input frame. These operations are complex, but could be performed in real-time on existing video processing platforms taking into account that typical infrared camera frame resolution is relatively low, i.e., from 320×240 to 640×512[3].

IV. Conclusion

In this paper we presented a novel efficient algorithm for drone infrared video coding exploiting additional redundancy of such videos as well as similarity with other infrared videos collected above the same area. The proposed algorithm provides high bit rate savings for real-life drone infrared videos with a price of higher complexity and can be used for real-time infrared video streaming from drones to a other devices, when the channel capacity minimization is more important then the minimization of the video encoder complexity.

[3]For example, see cameras Fluke Ti32, Flir Boson, Flir A655SC, Flir A65 and Flir Tau 320

References

[1] O.Friman, P.Follo et al., "Methods for Large-Scale Monitoring of District Heating Systems Using Airborne Thermography", *IEEE Trans-*

actions on Geoscience and Remote Sensing, Vol.52, Iss.8, 2014.

[2] SungWon Lee, Kwang Eun An et al.,"Detecting faulty solar panels based on thermal image processing," *IEEE International Conference on Consumer Electronics (ICCE)*, 2018.

[3] High Efficiency Video Coding, *document ITU-T Rec. H.265 and ISO/IEC 23008-2*, 2014.

[4] Ma Biao and A. Reibman, "DashCam Video Compression using Historical Data", *Picture Coding Symposium (PCS)*, 2016.

[5] Xu Wang, Jing Xiao et al.,"Cruise UAV Video Compression Based on Long-Term Wide-Range Background", *Data Compression Conference (DCC)*, page 466, 2017.

[6] X.Wang, R.Hu, Z.Wang, J.Xiao, Virtual Background Reference Frame Based Satellite Video Coding, *IEEE Signal Processing Letters*, Vol.25, Iss.10, 2018.

[7] E.Belyaev and S.Forchhammer, "An efficient storage of infrared video of drone inspections via iterative aerial map construction", *IEEE Signal Processing Letters*, 2019. DOI:10.1109/LSP.2019.2921250

[8] F.Dufaux, J.Konrad, Efficient, robust, and fast global motion estimation for video coding, *IEEE Transactions on Image Processing*, Vol.9, Iss.3, pp.497–501, 2000.

[9] Y. Xiong, K. Pulli, "Mask-based image blending and its applications on mobile devices", MIPPR 2009: Remote Sensing and GIS Data Processing and Other Applications, 2009.

[10] MV-HEVC Software [Online], https://hevc.hhi.fraunhofer.de/3dhevc/

[11] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves", VCEG-M33, Thirteenth Meeting of the Video Coding Experts Group (VCEG): Austin, Texas, USA, 2-4 April, 2001.