10th INTERNATIONAL SCIENTIFIC CONFERENCE
ON DEFENSIVE TECHNOLOGIES
OTEH 2022
Belgrade, Serbia, 13 – 14 October 2022

SOFTWARE FOR VIDEO STREAMS SYNCHRONIZATION IN LONG RANGE SURVEILLANCE SYSTEMS

ALEKSANDAR SIMIĆ
Vlatacom Institute, Belgrade, aleksandar@vlatacom.com and Belgrade Metropolitan University

TAMARA PAROJČIĆ
Vlatacom Institute, Belgrade, tamara.parojcic@vlatacom.com

MIROSLAV PERIĆ
Vlatacom Institute, Belgrade, miroslav.peric@vlatacom.com

DR DRAGAN DOMAZET, EMERITUS
President of Belgrade Metropolitan University, dragan.domazet@metropolitan.ac.rs

Abstract: *Long range surveillance systems are imaging systems used for military or law enforcement applications which include different types of imaging sensors like thermal camera, visible light camera and SWIR camera that are installed on common pan-tilt positioner. Video signals from these sensors propagate to command center through communication network in compressed form (e.g. using H264 video compression) that degrades image quality. In order to perform further signal processing of these stream, like multi-modal target detection using artificial intelligence EDGE processing platforms, these streams should be synchronized in time domain and in observation field of view. In this paper we present new solution for this problem which uses embedding metadata information into video frames on the host side in a manner that this information can be reconstructed on the receiver side. The problem is solved by embedding text and markers at special positions in a frame that keeps information about current azimuth, elevation, field of view of each stream together with frame sequencing marking. The solution also exploits the fact that all streams have in one moment, determined by time synchronization of each stream's frame sequence, the same azimuth and elevation. We present measurement results in urban environment using Vlatacom Institute's multi-sensor imaging systems class VMSIS3.*

Keywords: *Surveillance, multi-spectral imaging systems, video stream, synchronization*

I. INTRODUCTION

In order to achieve its mission in the majority of day/night and meteorological conditions long range surveillance systems – LRSS incorporate multi spectral cameras from various ranges like visible light, thermal in long range infrared – LWIR or medium range infrared MWIR or short range infra-red SWIR. Some early-days solution could be found in [1] and [2], while the more detailed description of one of the world-class solution, denoted as Vlatacom Multisensor Imaging System, third generation, VMSIS3 is presented in [3]. The most important component of LRSS which should be carefully tailored for systems' mission is presented in [4] and [5]. Apart from installed cameras, video signal processing elements are crucial for overall long-range surveillance system performance [6]. One typical application is target tracking [7]. In order to activate target tracker, various modes of target detection system are utilized ranging from manual, via motion detection to artificial intelligence- AI based modules.

Modern surveillance system utilizes several thousands of cameras to cover area of interest. This yields to camera networks. Some basic principles are described in [8] and [9]. Video stream from multiple cameras in LLSS are transmitted to command and control – C2 center usually in digitalized and compressed form. In order to efficiently utilize transmission network bandwidth video stream is compressed. One of the most popular video compression algorithms is H.264 [10]. This algorithm is lossy type, which means that video quality is degraded in comparison to uncompressed raw video. The algorithm is optimized for

visible color image and its performance comparison to other video compression algorithm are investigated in [11] and [12]. Unfortunately, the algorithm is not equally optimized for thermal images, which are intensity only, especially in the cases when image is not focused enough or too noisy. Tailoring algorithm parameters for this use case is usually proprietary by LLSS manufacturers.

Unlike CCTV type of cameras that have fixed position and thus can be easily integrated into advanced video analytics applications like pedestrian monitoring [13] and [14], LLSS video stream has additional parameters like camera position specified by pan (azimuth) and tilt (elevation) angles and zoom (field of view – FOV). This information cannot be accurately embedded into a compressed video stream. Since in a LLSS multiple cameras share the same positioning platform, denoted as a pan-tilt positioner, in this paper we utilize this fact and explore how to embed this information in efficient way in video stream so video analytics software in a C2 center can utilize acquired information more efficiently. The paper is structured as follows: In section II we give system description, in section III we give detailed description of the algorithm in section IV we give measurement results and in section V we give conclusion

II. SYSTEM DESCRIPTION

Overlayed text detection and embedded marker detection was researched/analyzed with the idea of later using it for synchronization of three different cameras- SWIR, Thermal and Low Light. In figure 1 are shown examples of frames from all three cameras.

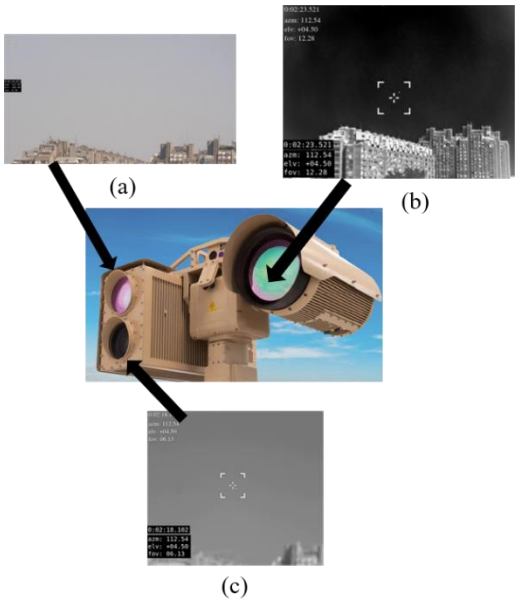


Figure 1. Examples of frames with embedded text from all three cameras: (a) Low Light, (b) Thermal, (c) SWIR

Information about azimuth, time, elevation and field of view is overlayed on videos for testing methods used for character recognition. Also, two types of markers were embedded in videos, blinking and moving marker. Blinking marker is one pixel that appears every 30 frames at the same position for every video, and the idea is to use this marker for camera synchronization. Moving markers are one pixel information moved throughout first row of video, moving its position one pixel per frame. This marker carries analogue information- its position- and this can be used for encrypting any kind of information that user needs. In figure 2 is shown an example of frame which has moving, blinking markers and overlayed text information.

Figure 2. Example of frame with blinking, moving marker, and overlayed text

Analysis was done on 27 videos, nine videos for each channel (SWIR, Thermal, Low Light). Videos were recorded using Vlatacom vMSIS C-825 camera.
The nine videos included recording of three different surroundings/scenes, text written in two different fonts, and text written with or without homogenous background, every video included two types of embedded markers.

III. DESCRIPTION OF THE ALGORITHM

Environment used for data processing is *Python ver. 3.8* (Python Software Foundation, open source) in Spyder (Python Software Foundation, open source). The used libraries were: numpy [15], openCv [16], matplotlib [17], pandas [18], pytesseract [19].

Blinking marker is placed at the same positions in frames from all three cameras thus, the information about marker position was available and its detection was simplified.

Detection of blinking marker was performed by checking if the value of weight function, defined by equation (1) (sum of absolute differences between red, green and blue channels) is greater than threshold value T_{hf}, in the area near/around the previously determined position.

$$f=\sum' |R-G|+|R-B|+|B-G| \quad (1)$$

where

R – pixel value of red channel,

G – pixel value of green channel, and

B – pixel value of blue channel.

Threshold value is determined by comparing values of weight function for every video. We found that value of T_{th}=300 is the optimal value. Moving marker is always placed in the first row so its detection implied finding the y-coordinate (x-coordinate was always 0) of the maximum value of weight function.

For text detection two approaches were used *Tesseract* and template matching method.

We use open-source OCR (Optical Character Recognition) engine *Tesseract* [20]. Tesseract uses trained LSTM (Long short-term memory) models to extract and interpret information from a variety of documents.

Preprocessing for *tesseract* included cropping text parts of images and converting them to grayscale. We have tested OCR algorithms with two types of fonts (*DejaVu Sans Mono and Nimbus Roman No9 L*), with and without homogenous background. *Tesseract* was configured to treat the image as a single word and to read only digits, it takes image and converts it to string. The main downside of using OCR engines is that they are time-consuming, when using *Tesseract* process time per frame was around 0,5 seconds. Beside of being time-consuming *Tesseract* accuracy was low thus other method was proposed.

Template matching is a fast process that moves the template over the entire image and calculates the similarity between the template and the covered window on the image. It is implemented via two-dimensional convolution function from OpenCV framework. Since the font and the position of text were known/predetermined it was possible to use the template matching method. First, it was necessary to make templates of digits, plus and minus signs for two fonts. Template matching was implemented on parts of images that contained information about time, azimuth, elevation and field of view. A threshold value of 0,9 was used to determine if the similarity between the window of the image and the template is high enough.

IV. MEASUREMENT RESULTS

Marker detection was successful for both moving and blinking markers. In figure 3 are shown graphs for y coordinate of moving marker for one video of different types of cameras. On these graphs we can see that y-coordinate has linear trend in range from zero to the maximum width of image. The difference between y-coordinates in adjacent frames can't be easyly observed from these graphs so in figure 4 are shown graphs of differences between neighboring frames.

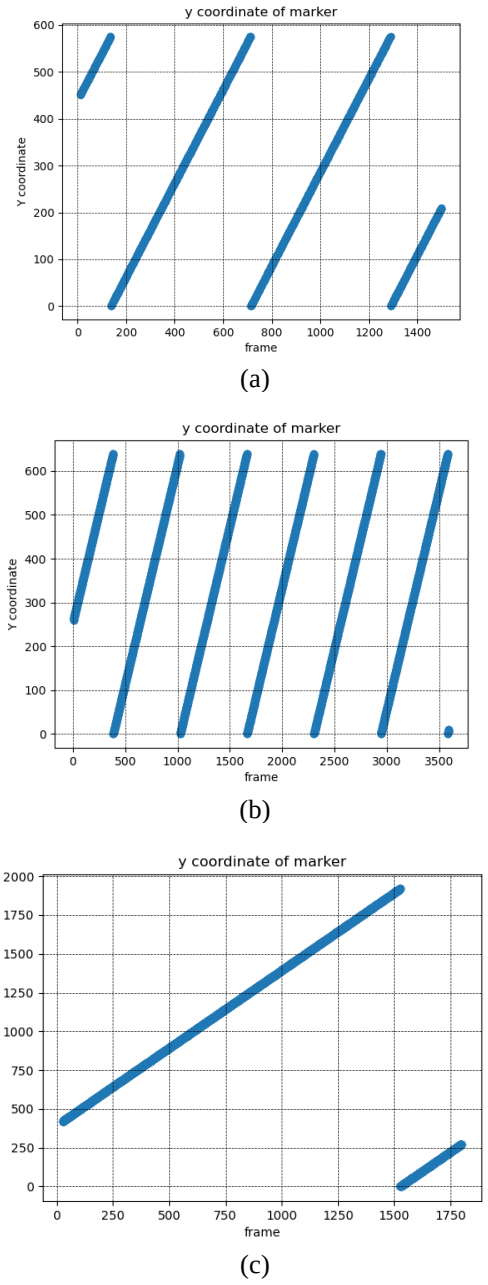
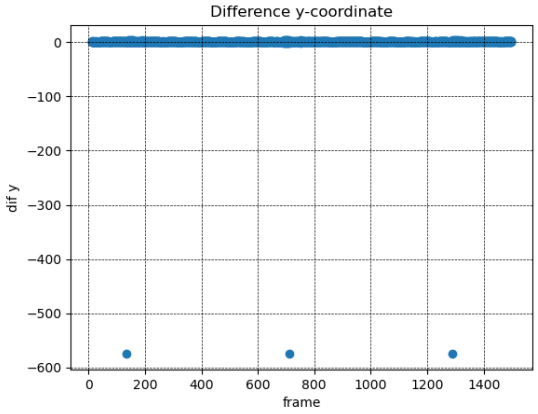
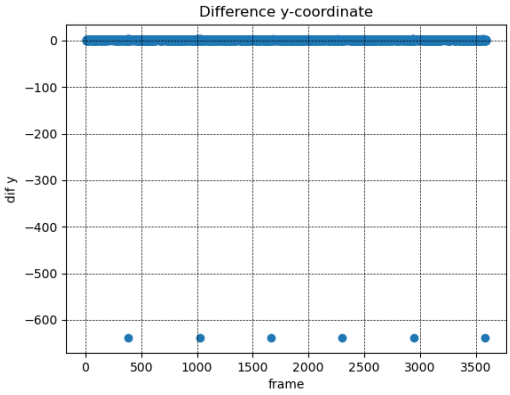


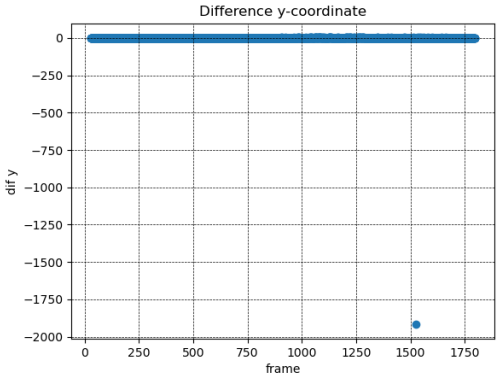
Figure 3. Graphs for y coordinate of moving markers: (a) SWIR, (b) Thermal, (c) Low Light



(a)



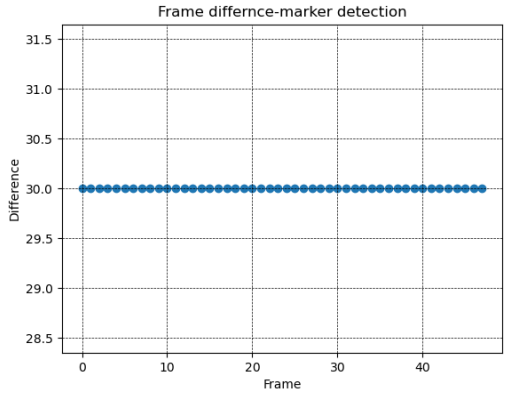
(b)



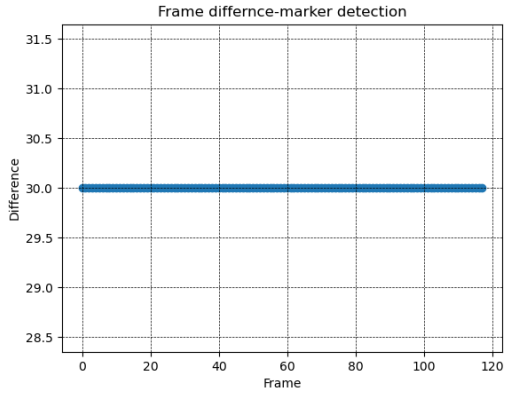
(c)

Figure 4. Difference between y coordinate of moving marker of adjacent frames: (a) SWIR, (b) Thermal, (c) Low Light

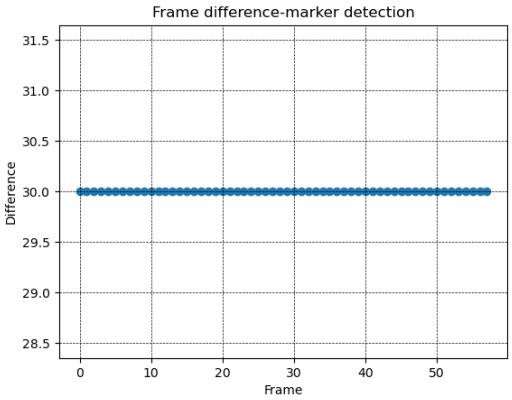
In figure 5 are shown differences between frames in which blinking marker was detected. Since this value is always the same, marker detection was successful.



(a)



(b)



(c)

Figure 5. Difference between frames where blinking marker was detected: (a) SWIR, (b) Thermal, (c) Low Light

Text detection

Tesseract

Using *Tesseract* included lot of fine tuning of crops of image, the results obtained weren't consistent even for the same video where value of azimuth, elevation and field of view is same throughout frames. The execution was slow, per video time of execution was in average around 20 minutes. Resolution of characters, which was 10x14 pixels, represented main issue for Tesseract since it has higher accuracy when using images with larger resolution. However increasing resolution of characters would make

the characters too big for overlaying in videos from the end user perspective.

Template Matching

In order to overcome these issues the Template Matching was subsequently applied. Percentages of frames in which text was wrongly read are given in tables 1-3. Text written using DejaVu Sans Mono font with homogenous black background (in a bottom left corner) was used as a reference, because acquired values were equal to true values..

First column of tables represents ordinal number of video, in videos from 1 to 3 *Nimbus Roman No9 L* font without homogenous background was used for text in upper left corner, in videos from 4 to 6 *Nimbus Roman No9 L* font with homogenous background was used also for text in upper left corner, and for videos from 7 to 9 *DejaVu Sans Mono* font without homogenous background was used for upper left corner. Videos S1, T1, V1, S4, T4, V4 and S7, T7, V7 are recordings of sky in the background; S2, T2, V2, S5, T5, V5 and S8, T8, V8 are recordings of with building in background, while S3, T3, V3, S6, T6, V6 and S9,T9,V9 are recordings of traffic intersection in background. Letter S in video indexing represents videos recorded using SWIR camera, letter T in video indexing represents videos recorded using Thermal camera while letter V in video indexing represents videos recorded using LowLight camera.

Table 1. Percentage of wrongly read text for SWIR camera, using template matching method

Videos	Azimuth (%)	Elevation (%)	Field of view (%)	Time (%)
S1	0.20	2.48	3.78	1.32
S2	0.61	4.56	2.68	2.19
S3	2.96	3.57	2.38	3.03
S4	0	0	0	0.68
S5	0.07	0	0	0
S6	0.14	0	0	0.04
S7	2.80	2.45	1.95	1.04
S8	0.34	0	0	0
S9	0.81	0.27	0.20	0.45

Table 2. Percentage of wrongly read text for Thermal camera, using template matching method

Videos	Azimuth (%)	Elevation (%)	Field of view (%)	Time (%)
T1	0.20	2.48	3.78	2.32
T2	0.61	2.56	2.74	2.19
T3	1.96	4.57	3.38	2.03
T4	0	0	0	0.06
T5	0.07	0	0	0
T6	0.14	0	0	0.24
T7	3.80	2.05	5.95	3.64
T8	0.34	0	0	0
T9	0.81	0.27	0.20	0.56

Table 3. Percentage of wrongly read text for Low Light camera, using template matching method

Videos	Azimuth (%)	Elevation (%)	Field of view (%)	Time (%)
V1	34.12	26.78	19.66	17.76
V2	19.34	22.57	18.36	19.49
V3	5.67	10.12	11.47	16.9
V4	0.17	0	0	0.25
V5	0.06	0	0	0.16
V6	0.11	0	0	0.16
V7	12.56	22.47	18.72	18.44
V8	16.67	55.46	29.49	40.88
V9	13.28	10.05	12.06	41.11

Since the lowest error values were for videos 4, 5 and 6 it can be concluded that text recognition is more accurate when text is written with homogenous background, which was anticipated. Also by comparing the error percentages from the tables it can be observed that text detection had higher error percentage for LowLight videos. By analyzing tables for SWIR and Thermal videos it can be interpreted that text was wrongly read for videos where it wasn‘t overlayed on homogenous background and the scene behind it wasn‘t homogeneous.

The templates that were used had white characters written on black background. They are the reason for lower accuracy for LowLight videos and text written without homogeneous backgrounds. Examples of the templates that were used are shown in Figure 6.

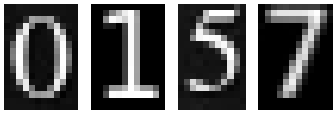


Figure 6. Example of templates used for OCR

In Figure 7 are represented examples of text overlayed without homogenous background and with homogenous background.

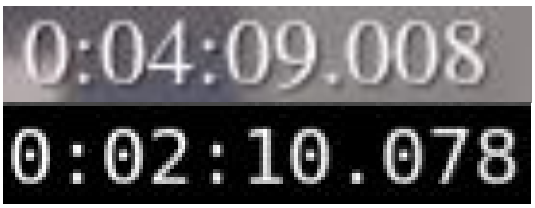


Figure 7. Text overlayed without and with black homogenous background

Impact on the results of text detection and recognition also has H264 compression. In figure 8 is given an example where before mentioned compression had affected text by smearing it in some frames.

Figure 8. Example of smeared time stamp

Synchronization of two rtsp streams using embedded markers

Video signal propagates from camera to client through frame grabber, enters processing module (JetsonTX2) where it is compressed (H264 compression) and after that, using rtsp server, rtsp stream is generated which further propagates through network infrastructure and finally it is received on the client side using rtsp client software. Every element in this path affects latency of video signal, which can be variable. Most impact of latency variability comes from propagation through network. So, in general case, two video streams received on client side are not in synchronization. Signal processing chain is shown on figure 9.

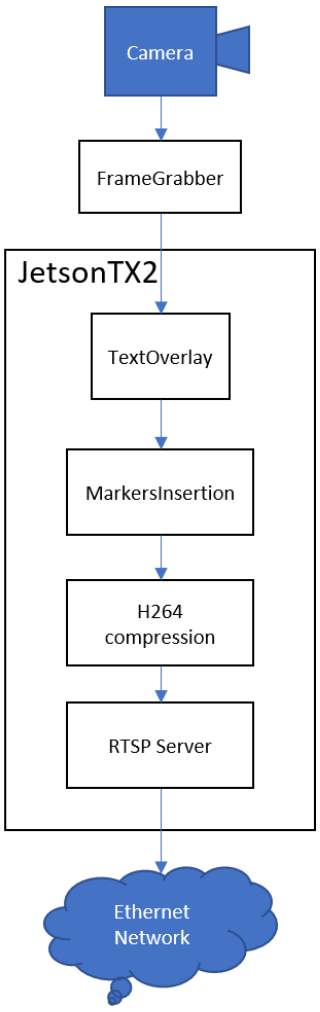
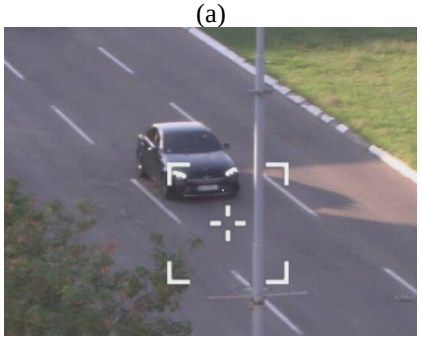


Figure 9. Signal processing chain on single channel

We embedded synchronizaton markers on every 2 seconds on both video streams (Thermal and LowLight). Markers are embedded just before H264 compression. It is considered that latancy of signal through frame grabber is fixed. When we detect these markers on frames received on client side, and offset them with that fixed latancy amount, streams are synchronized. Results are shown on following images.

(a)



(b)

Figure 10. Unsynchronized stream from (a) Thermal and Visible (b) camera

Figure 10 shows two frames received from both cameras without synchronization. Traffic pole is used as reference and we can see that car approaching pole is not on the same place on two images. Measuring time difference between two frames where the object of interest (car) is on the same place we can find out fixed latency between video signals which is 99ms in this case. After that, we simulated network latency of 100ms (with „latency“ parameter of gstreamer pipeline which is used as rtsp client). On Figure 11 we can see larger synchronization issue caused by this latency.

(a)

(b)

Figure 11. Unsychronized stream from (a) Thermal and Visible (b) camera with network latancy of 100 ms

After detection of embedded markers on both streams on Figure 12 is shown that two streams are well synchronized.

(a)

(b)

Figure 12. Synchronized streams from (a) Thermal and Visible (b) camera

Synchronization of azimuth and velocity data with video frames

Meta data relevant for optical systems are propagated through independent RTSP stream (azimuth, velocity, field of view, digital zoom value, etc.). There is a need to synchronize this data with video frames. Overlaying textual information on video stream (before H264 compression) and taking into account fixed amount of latency between actual scene and this data we can synchronize data with appropriate video frame.

Further investigation proposals

Proposed methodology enables data transfer through overlayed text on the image and also using markers which are embedded on first line of the frame. Embedded markers on first line of the image transfer information which has precision determined with image size. In VGA image it is 1/640, HD image 1/1280, FullHD image 1/1920. Markers that we used transfer binary information, but it can be investigated how to achieve larger precision. There can be used, for instance, markers with differente illuminance, or we can use larger number of markers with appropriate coding.

V. CONCLUSION

Proposed methodology using information embedded into compressed video stream and its reconstruction on client side can be used for solving synchronization problem of rtsp streams in multisensors optical systems and also for data synchronization (azimuth, elevation, field of view...) with video frames. Result for text recognition on thermal and SWIR channel give satisfactory results, while for low light channel, methodology can be used only if overlayed text has fixed (e.g. black) background. On the other hand, synchronization by embedded markers proves reliable on all channels, so the idea of using azimuth and elevation info for one, the most reliable channel to all available video channels is feasible.

References

[1] TRZASKAWKA,P., KASTEK,M., ŻYCZKOWSKI,M., DULSKI,R., SZUSTAKOWSKI,M., CIURAPIŃSKI,W., BAREŁA,J.: *System for critical infrastructure security based on multispectral observation-detection module*, Proceedings of the SPIE, Volume 8896, 2013.

[2] SZUSTAKOWSKI,M., ŻYCZKOWSKI,M., KAROL,M., KASTEK,M., DULSKI,R., SZUSTAKOWSKI,A.,M., BAREŁA,J., MARKOWSKI,P., KOWALSKI,M.: *Ultra long range surveillance camera for critical infrastructure protection research range*, SPIE Security + Defence, Dresden, Germany, 2013.

[3] CIZELJ,V.: *Vlatacom Institute of High Technology—Ten Years since the First Accreditation*, Belgrade, 2021; ISBN 978-86-7466-891-7.

[4] PERIĆ,D., LIVADA,B., PERIĆ,M., VUJIĆ,S.: *Thermal Imager Range: Predictions, Expectations, and Reality*, Sensors 2019.

[5] KURODA,T.: *Essential principles of image sensors*, Boca Raton, Florida, CRC Press, 2015.

[6] LATINOVIĆ,N., POPADIĆ,I., TOMIĆ,B., SIMIĆ,A., MILANOVIĆ,P., NIJEMČEVIĆ,S., PERIĆ,M., VEINOVIĆ,M.: *Signal Processing Platform for Long-Range Multi-Spectral Electro-Optical Systems*, Sensors, 2022.

[7] STOJANOVIĆ,M., VLAHOVIĆ,N., STANKOVIĆ,M., STANKOVIĆ,S.: *Object Tracking in Thermal Imaging using Kernelized Correlation Filters*, Proc. of 17th International Symposium INFOTEH, Jahorina, March 2018.

[8] AGHAJAN,H., CAVALLARO,A.: *Multi-Camera Networks: Principles and Application*, Academic Press, 2009.

[9] WANG,X.: *Intelligent multi-camera video surveillance: A review*, Pattern Recognit. Lett. 2013.

[10] ITU-T REC. H.264.: *SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infrastructure of audiovisual services – Coding of moving video, Advanced video coding for generic audiovisual services*, revison 08/2021.

[11] KHAN,I.,U., ANSARI,M.,A., YADAV,A., SAEED,S.,H.: *Performance analysis of H.264 video decoder: Algorithm and applications*, 2015 International Conference on Energy Economics and Environment (ICEEE), 2015.

[12] LAYEK,A.: *Performance analysis of H.264, H.265, VP9 and AV1 video encoders*, 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), 2017.

[13] DIMITRIEVSKI,M., VAN HAMME,D., VEELAERT,P., PHILIPS,W.: *Cooperative Multi-Sensor Tracking of Vulnerable Road Users in the Presence of Missing Detections*, Sensors 2020.

[14] NEFF,C., MENDIETA,M., MOHAN,S., BAHARANI,M., ROGERS,S., TABKHI,H.: *REVAMP2T: Real-Time Edge Video Analytics for Multicamera Privacy-Aware Pedestrian Tracking*, IEEE Internet of Things Journal, vol. 7, no. 4, pp. 2591-2602, April 2020.

[15] HARRIS,C.,R., MILLMAN,K.,J., VAN DER WALT,S.,J.: *Array programming with NumPy*, Nature 585, 357–362, 2020.

[16] BRADSKI,G.: *The OpenCV Library*, Journal of Software Tools, 2000.

[17] HUNTER J.,D.: *Matplotlib: A 2D Graphics Environment*, Computing in Science, Engineering, vol. 9, no. 3, pp. 90-95, 2007.

[18] MCKINNEY,W.: *Data structures for statistical computing in python*, Proceedings of the 9th Python in Science Conference. p. 51–6, 2010.

[19] DOCSUMO: *10 Best OCR software in 2022*, https://docsumo.com/blog/best-ocr-software, 2022.

[20] GITHUB INC: *Tesseract OCR*
https://github.com/tesseract-ocr/tesseract, 2022.