Improving the quality of building space depth maps using multi-area active-pulse television measurement systems in dynamic scenes

S.A. Zabuga [1], V.V. Kapustin [1], I.D. Musihin [1]
[1] Federal State Autonomous Institution of Higher Education
"Tomsk State University of Control Systems and Radioelectronics" 634050, Russia, Tomsk, Lenin Ave., 40

Abstract

The purpose of this work is software implementation of the temporal frame interpolation, the formation of selection criteria and the choice of a suitable neural network model based on the obtained practical data. And also, evaluation of its efficiency for eliminating the interframe shift effect of dynamic objects on the depth maps of multi-area active-pulse television measuring systems in order to improve the accuracy of map building. As initial data for the experiments, static frames were recorded while moving the test rig along the X and Z axes. The static frames are images of the test rig, averaged 100 times, at a distance of 13 meters, which moved along an automated linear guide with a step of 1 mm. As a result of the work, an assessment of the interframe shift effect influence on space depth maps of multi-area active-pulse television measuring systems containing dynamic objects was made. The implementation and testing of the temporal frame interpolation algorithm for suppressing the interframe shift effect of dynamic objects on depth maps was also performed. The algorithm was implemented using Python and the PyCharm IDE with SciPy, NumPy, OpenCV, PyTorch, Threading and other libraries. Numerical values of the RMSE, PSNR, and SSIM metrics were obtained before and after eliminating the effect of interframe shift of dynamic objects on depth maps. The use of the temporal frame interpolation algorithm allows more accurate measurement of distance to moving object in the field of view of multi-area active-pulse television measuring systems.

Keywords: multi-area active-pulse television measurement system, depth map, Python, neural network, video frame prediction.

Citation: Zabuga SA, Kapustin VV, Musikhin ID. Improving the quality of building space depths maps using multi-area active-pulse television measuring systems in dynamic scenes. Computer Optics 2025; 49(4): 647-653. DOI: 10.18287/2412-6179-CO-1590.

Introduction

In the modern world, distance measurement is a very important and complex task. Many devices have been created in order to measure distance to objects simultaneously at each point of the image (building a depth map of space). An example of this type of device is the active-pulse television measuring system (AP TMS) [1]. Due to the design features and operating principles, AP TMS has the ability to build depth maps even in conditions of turbidity of the optical radiation propagation environment (fog, snowfall, rain and other meteorological phenomena) [2]. These features impose some limitations on the operation of multi-area range measurement methods [3] in the presence of dynamic (moving) objects in the field of view of the AP TMS. Dynamic objects create interframe differences, which negatively affect the quality of the space depth maps building process.

There are many studies on active-pulse television measuring systems aimed at improving the technical characteristics and methods of measuring distances to objects. It is worth noting that there are no studies in the open access on the topic of improving the quality of building depth maps of space by multi-area active-pulse television measuring systems in dynamic scenes.

The aim of this paper is to perform software implementation, formation of selection criteria and selection of a suitable neural network model based on the obtained practical data. And also, to evaluate the efficiency of the algorithm of temporal frame interpolation to eliminate the effects of inter-frame shifts of dynamic objects on depth maps of multi-area action-pulse television measurement systems in order to improve the accuracy of their building process.

1. Assessing the impact of interframe shifts on depth maps of multi-area AP TMS containing dynamic objects

To assess the influence of the contouring effect of dynamic objects, static frames were captured using AP TMS while moving the test rig along the X and Z axes. The static frames are images of the test rig, averaged 100 times, at a distance of 13 meters, which moved along an automated linear guide with a step of 1 mm. 170 instances were captured (maximum shift 170 mm), which is equal to 340 frames in total, since one frame for the summary area and one frame for the first area falls on one test rig movement step. An example of a static frame of the summary area is shown in Fig. 1.

To evaluate the effect of contouring, an artificial delay was created between the frames of the first and summary areas, the step of this delay (t) is equal to 1, 3, 5, 7, 9, 11 frames. This step was chosen due to the possibility of subsequently comparing the results with the frame interpolation. In Fig. 2, a part of the time scale of the selected frames is presented, empty cells mean deleted frames, and dark and light ones correspond to the frames of the summary area and the first area, respectively. For example, when accelerating by two times, the Depth map (Dm) will be built between frames (Fn) at time $n=0$ and at time $n=2$, in the general case it will look like $Dm=f(Fn, F(n+1)+t)$.

Fig. 1. Example of a static frame of the summary area

The impact of the dynamic objects contouring effect was assessed using the following metrics:

1. RMSE (root mean square error) is the most common indicator of the dispersion of values of a random variable relative to its mathematical expectation.
2. PSNR (peak signal noise ratio) – it is a technical term for the ratio between the maximum possible signal power and the power of distorting noise that affects the accuracy of its representation.
3. SSIM (structure similarity index method) – is one of the methods for measuring the similarity between two images. SSIM is a full matching method, in other words, it measures the quality based on the original image without compression or distortion. The method takes into account perceptual losses by taking into account the structural change of information.

An example of the contouring of objects on a depth map obtained with an artificial delay between the frames of the first and total areas (t) equal to 11 frames is shown in Fig. 2.

Tab. 1. Part of the frame timeline

Moment of time, n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Initial frames	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**
t = 1	1		3		5		7		9		11		13		15		17		19		21
			3		**5**		**7**		**9**		**11**		**13**		**15**		**17**		**19**		**21**
t = 3	1				5				9				13				17				21
					5				**9**				**13**				**17**				**21**
t = 5	1						7						13						19		
							7						**13**						**19**		
t = 7	1								9								17				
									9								**17**				
t = 9	1										11										21
											11										**21**
t = 11	1													13							
														13							

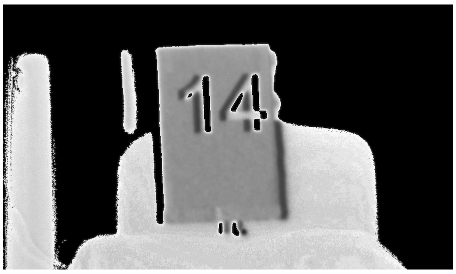

Fig. 2. Depth map when moving the test rig along the X axis with a delay of 11 frames

As a result of the measurements, average metric values were obtained when moving the test rig (with banner) along the X and Z axes. The obtained data are presented in Table 2 and Table 3.

Based on the results of this experiment, RMSE, PSNR and SSIM graphs were obtained (Fig. 3 – 5), as well as tables of the influence of interframe differences on the depth map of dynamic objects at different speeds of movement of the test rig.

A weak influence of the interframe difference on the depth map was also revealed when the test rig moved along the Z axis. This influence is due to the different shift in pixels, which occurs due to the perspective when the test rig moves along the Z axis.

Tab. 2. Average values of RMSE, PSNR, SSIM metrics when moving along the X axis

Metric	Frame lag, *t*					
	1	3	5	7	9	11
Average val. RMSE	14,62	27,43	38,99	46,99	51,78	56,06
Average val. PSNR, dB	24,89	19,48	16,42	14,79	13,68	13,23
Average val. SSIM	0,88	0,76	0,71	0,68	0,65	0,64

Tab. 3. Average values of RMSE, PSNR, SSIM metrics when moving along the Z axis

Metric	Frame lag, *t*					
	1	3	5	7	9	11
Average val. RMSE	12,49	16,41	17,78	18,68	19,57	21,82
Average val. PSNR, dB	26,41	23,97	23,29	22,88	22,41	21,54
Average val. SSIM	0,72	0,56	0,54	0,53	0,52	0,50

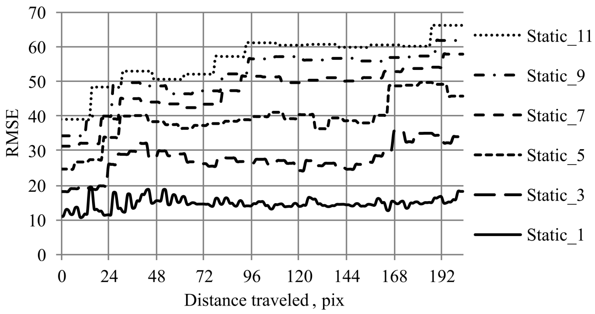

Fig. 3. Ratio of RMSE of all delayed depth map frame sequences to distance traveled in pixels, for movement along the X axis

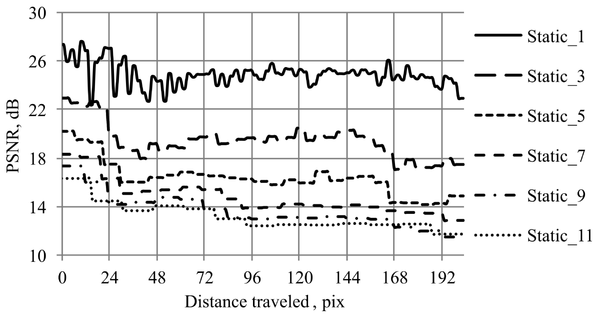

Fig. 4. The ratio of PSNR of all delayed depth map frame sequences to the distance traveled in pixels, for movement along the X axis

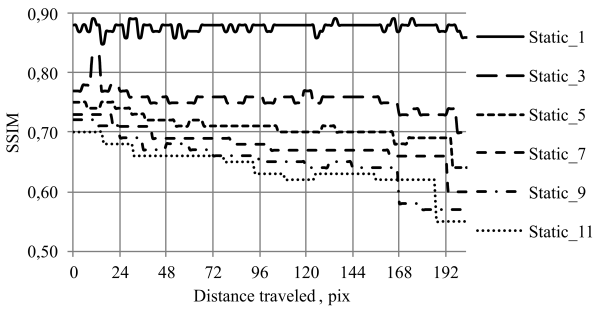

Fig. 5. Ratio of SSIM of all delayed depth map frame sequences to distance traveled in pixels, for movement along the X axis

2. Temporal frame interpolation

The classical approach to solving the video frame prediction problem is to calculate the optical flow. Optical flow is an image of visible motion that represents the shift of each point between two images. It represents a velocity field, because the shift is proportionally equivalent to the instantaneous velocity, with accurate scaling. The main purpose of optical flow is to find, for each point in a frame, a corresponding shift that that the original point in the first frame matches a point in the second frame. Obviously, intensity conservation fails, if the illumination or the angle of incidence of light changes. The classical approaches are the Lucas-Kanade and Farneback algorithms. To implement the algorithm, a window around the point is selected, and the total error is minimized by using Gaussian-distributed weights. Approximated values of intensity change in the neighborhood are used, represented by a quadratic form, respectively. The problem with these algorithms is that even a small shift of 2 pixels can result in a significant error, which grows exponentially with larger shifts of 3 pixels or more. When using Taylor series, we can approximate the change in the value of the function within a finite neighborhood of a point, whereas the derivative provides insight into the behavior of function is an infinitesimal small neighborhood around that point [29 – 30]. Such errors in predicting the frames that contribute to the creation of the depth map are unacceptable, so the article investigates the novel methods based on convolutional neural networks (CNN).

Video frame interpolation (VFI) is a technique that aims to synthesize intermediate frames between two consecutive frames. VFI is applied in various tasks such as slow-motion generation, video stream compression [4], and video frame prediction [5]. VFI is a complex task due to nonlinear motions and illumination changes in real frames. Recently, optical flow-based VFI algorithms have proposed methods to address these issues and achieved significant results [6]. Common approaches to these methods include two stages:

1. «Wrapping» input frames according to approximated optical flows.
2. Combining wrapped frames using convolutional neural networks (CNN).

Optical flow models cannot be directly used in VFI. Given input frames $I0$, $I1$, optical flow-based methods require approximating intermediate optical flows $Ft \to 0$, $Ft \to 1$ in terms of the frame It to be synthesized. The problem is that the frame It is not available before synthesis and its computation is a complex task [7, 8]. Most methods first compute bidirectional flows based on optical flow models, then reverse and refine them to generate intermediate optical flows $Ft \to 0$, $Ft \to 1$. Such flows may have motion boundary defects because the object position changes from frame to frame ("object shift problem"), as shown in Fig. 6.

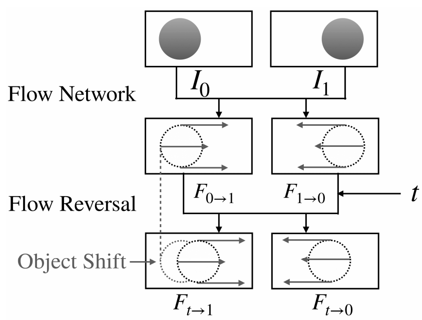

Fig. 6. Illustration of the "object displacement problem"

Appearance Flow [9, 10], a pioneering work on flow synthesis, proposes to estimate the flow starting from the target representation using CNN. Deep voxel flow (DVF) [5, 11 – 13] in turn computes the voxel flow of dynamic scenes to jointly model the intermediate flow and a blend mask to estimate them end-to-end. Adaptive collaboration of flows (AdaCoF) [7, 14] extends intermediate flows to adaptive collaborative flows. Bilateral motion estimation with bilateral cost (BMBC) [15, 16] uses a bilateral cost operator to obtain more accurate intermediate flows (bilateral motion).

To solve the problem of eliminating the contouring of dynamic objects on AP TMS depth maps, it is necessary to use such a feature of VFI as video frame prediction. To calculate intermediate video frames in real time, the model must have the following criteria:

1.No need for additional components such as models for obtaining a depth map from an image, models for refining the optical flow, which are needed to compensate for defects in the intermediate optical flow.

2.End-to-end trainable motion estimation, instead of introducing motion modeling, it is better to make a convolutional neural network (CNN) learn the end-to-end intermediate stream.

3.Providing direct supervision of approximated intermediate optical flows: Most VFI models are trained only with the final reconstruction loss. Propagating per-pixel loss gradients through a warping operator is inefficient for flow estimation [17 – 20]. The lack of supervision specifically designed for flow estimation degrades the performance of VFI models.

Based on the criteria above, the RIFE (Real-time Intermediate Flow Estimation) model was selected. RIFE directly estimates the intermediate flow from neighboring frames and input time data. And, using a coarse-to-fine strategy, it iteratively updates the intermediate optical flows and a soft fusion mask to merge pixels from two input frames with gradually increasing resolution. RIFE uses intermediate supervision during training, which uses a teacher-student model, where the teacher has access to intermediate frames to help the student learn. Figure 7 shows the performance results of RIFE compared to peers on the Vimeo90K validation dataset [5, 16].

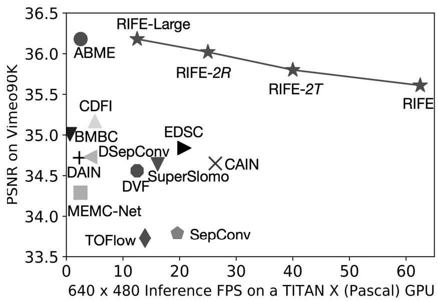

Fig. 7. RIFE Performance

3. Software implementation of the algorithm for temporal frame interpolation

The computer program was written using the Python programming language in the PyCharm IDE. The development of the program utilized by several libraries such as SciPy, NumPy, OpenCV, PyTorch, Threading and other libraries as referenced in [21 – 25].

Due to the peculiarities of AP TMS operations, as well as the requirement for streaming interpolation, the temporal interpolation algorithm [29 – 30] was modified to adapt to accommodate AP TMS. The depth map construction utilizes a multi-zone approach, i.e., adjacent frames having variation in intensity. Conversely, the interpolation algorithms need frames with the identical intensities but with distinct time instants. To bypass this limitation in the algorithm's work, a delay of three frames is artificially created, after which the algorithm interpolates an intermediate frame between the frames of one zone and combines them with the frames of another zone using a neural network. The output is a video sequence of pairs of frames that should differ only in intensity, but not in time moments. The block diagram of the algorithm for suppressing interframe differences on AP TMS depth maps is shown in Fig. 8.

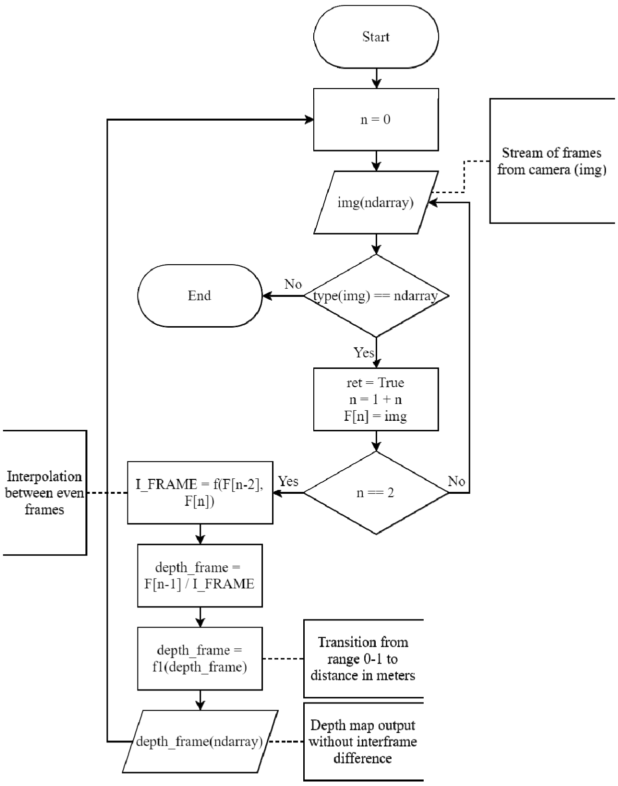

Fig. 8. Block diagram of the algorithm

For real-time frame interpolation, a video card is a prerequisite, since the calculations require multiple simultaneous operations. The parameters of the computer on which the measurements were made are given in Table 4.

Tab. 4. Computer specification

CPU	AMD Ryzen 3700x, 4,3 GHz
GPU	NVIDIA RTX 4070Ti
SSD M.2	Samsung 980
RAM	HyperX DDR4 3333MHz, 32 GB

Fig. 9 shows the interpolation time of each frame in milliseconds, Table 5 shows the average value of the interpolation time.

Based on the maximum frame rate of the AP TMS (50 frames per second or 20 milliseconds per frame), it can be concluded that the RIFE-3 and RIFE-2T neural network models provide the required performance and allow real-time operations. Due to the performance and quality of frame interpolation [8], the RIFE-3 model was selected for further use.

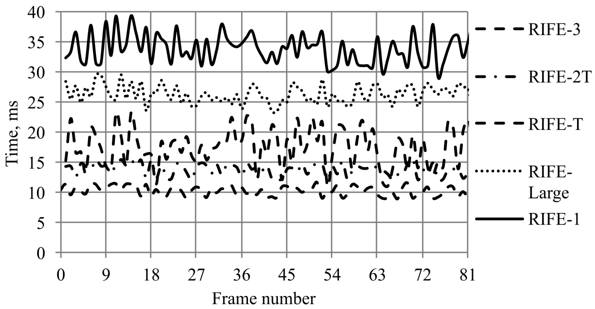

Fig. 9. Frame interpolation time

Tab. 5. Average values of interpolation time of models

Model	RIFE-3	RIFE-2T	RIFE-Large	RIFE-T	RIFE-1
Time spent, ms	10,31	13,76	17,42	26,06	34,10

4. Evaluation of the efficiency of eliminating interframe shifts of dynamic objects on depth maps using the algorithm of temporal frame interpolation

The efficiency of dynamic object contouring elimination on AP TMS depth maps using VFI technology was assessed. Interpolated intermediate frames of the summary area were combined with similar static frames at the same time to create AP TMS depth maps. Fig. 10 shows an example of an interpolated frame of the summary area.

Fig. 10. Interpolated frame of the summary area

The frame delay step 1, 3, 5, 7, 9, 11 was taken in order to be able to use the already existing static frame of the first area for comparison with the interpolated frame of the summary area. The reason for using the already existing static frame of the first area is that in real conditions there is no reason to interpolate both areas, since the frames are sequential. For the interpolated frames, the metrics were calculated, which are given in Table 6 and Table 7.

Tab. 6. Average values of RMSE, PSNR, SSIM metrics when moving the banner along the X axis

Metric	Frame lag, *t*					
	1	3	5	7	9	11
Average val. RMSE	16,64	17,12	17,42	17,89	17,78	18,60
Average val. PSNR, dB	23,79	23,52	23,39	23,13	23,12	22,75
Average val. SSIM	0,84	0,83	0,83	0,82	0,82	0,82

Based on the obtained data, we can conclude that the impact of frame delay (i.e. displacement of objects within the frame) on the depth map, after finding an intermediate frame using interpolation is very negligible, which indicates a good quality of video frame prediction. Typically, an increase in delay, corresponds to an increase in the effect of delineation. With the help of temporal interpolation this effect is leveled out. This inference is drawn from the comparative analysis of depth maps generated with frame delay and those generated without frame delay.

Tab. 7. Average values of RMSE, PSNR, SSIM metrics when moving the banner along the Z axis

Metric	Frame lag, *t*					
	1	3	5	7	9	11
Average val. RMSE	17,11	17,06	17,11	17,87	17,42	18,36
Average val. PSNR, dB	23,51	23,56	23,55	23,19	23,37	22,98
Average val. SSIM	0,64	0,64	0,65	0,64	0,65	0,65

The Depth map obtained after interpolation of the original frames with a delay between the first and the total area (t) equal to 11 frames is shown in Fig. 11.

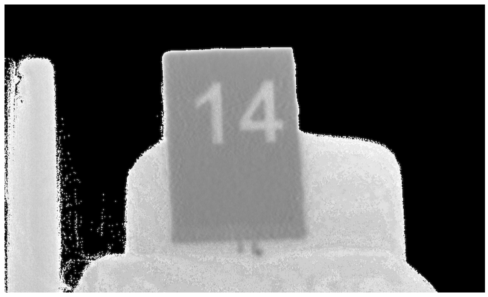

Fig. 11. Depth map when moving the test rig along the X axis after interpolation of the original frames with a delay of 11 frames

Conclusion

In this paper, methods based on two-way optical flow computation and full-circle neural networks are considered. A comprehensive analysis of method's advantages and disadvantages is presented, yielding selection criteria such elimination of additional components, end-to-end trainable motion estimation, instead of introducing some inaccurate motion modeling and providing direct control of approximated intermediate optical flows.

Using the described experimental setup and evaluating with the RMSE, PSNR, and SSIM metrics, plots and tables illustrating the impact of inter-frame difference on the depth map of dynamic objects at various bench speeds were generated. The presented data suggest that even a delay of two frames, which is equivalent to an average shift of 1.5 pixels, increase $\Delta RMSE = 13$ and decrease $\Delta PSNR$ and $\Delta SSIM$ by 5 dB and 0.2, respectively, for X-axis movement and increase $\Delta RMSE = 4$ units and decrease $\Delta PSNR$ and $\Delta SSIM$ by 2 dB and 0.15, for Z-axis movement.

Based on the maximum frame rate of AP TMS, which is 50 frames per second, which in turn equals 20 milliseconds per frame, we can conclude that the neural network models RIFE-3 and RIFE-2T, with processing times of 10.31 ms and 13.76 ms provide the required performance and allows real-time operations. The newer

RIFE-3 was chosen to provide more time for other mathematical operations.

From the data obtained, it can be concluded that there is very little effect of frame delay on the depth map after finding an intermediate frame using interpolation. So, the described method improves the quality of the depth map with dynamic objects, at the maximum delay equal to 11, which corresponds to the movement of the object by 15 pixels per frame RMSE value is reduced by 37.5, and PSNR and SSIM are increased by 9 dB and 0.18, respectively, for movement on the X-axis. For Z-axis movement, the RMSE value is decreased by 4 and increased by 1.5 dB and 0.15, for PSNR and SSIM metrics.

It was observed that at delay $t = 1$, the application of interpolation degrades the depth map. This is attributed to the fact that when moving an object in depth, its perspective changes, causing it to appears almost static withing the frame even at maximum frame delay. This leads to conclude that interpolation has a negative effect on static frames or frames with very little dynamics, because it introduces some distortion. For X-axis movement, a similar explanation applies: extremely small displacement do not generate significant distortion, but interpolation introduces notable changes.

Acknowledgements

The study was carried out with the support of the Russian Science Foundation grant No. 21-79-10200 at TUSUR.

References

[1] Kapustin VV, Zahlebin AS, Movchan AK, Kuryachiy MI, Krutikov MV. Experimental assessment of the distance measurement accuracy using the active-pulse television measuring system and a digital terrain model. Computer Optics 2022; 46(6): 948-954. DOI: 10.18287/2412-6179-CO-1114.

[2] Musikhin ID, Kapustin VV, Tislenko AA, Movchan A, Zabuga SA. Building depth maps using an active-pulse television measuring system in real time domain. Scientific Visualization 2024; 16(1): 38-51. DOI: 10.26583/sv.16.1.04.

[3] Kapustin VV, Movchan AK, Tislenko AA. Experimental evaluation of the accuracy of range measurement with multiarea methods using an active-pulse television measuring system. Optoelectron Instrument Proc 2024; 60: 145-155. DOI: 10.3103/S8756699024700134.

[4] Fleet D, Weiss Y. Optical flow estimation. In Book: Paragios N, ed. Handbook of mathematical models in computer vision. Ch 15. Boston: Springer US; 2006: 237-257.

[5] Liu Z, Yeh RA, Tang X, Liu Y, Agarwala A. Video frame synthesis using deep voxel flow. Proc IEEE Int Conf on Computer Vision 2017: 4463-4471. DOI: 10.1109/ICCV.2017.478.

[6] Jiang H, Sun D, Jampani V, Yang MH, Learned-Miller E, Kautz J. Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. 2018 IEEE/CVF Conf on Computer Vision and Pattern Recognition 2018: 9000-9008. DOI: 10.1109/CVPR.2018.00938.

[7] Lee H, Kim T, Chung TY, Pak D, Ban Y, Lee S. AdaCoF: Adaptive collaboration of flows for video frame interpolation. 2020 IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR) 2020: 5316-5325. DOI: 10.1109/CVPR42600.2020.00536.

[8] Huang Z, Zhang T, Heng W, Shi B, Zhou S. Real-time intermediate flow estimation for video frame interpolation. In Book: Avidan S, Brostow G, Cissé M, Farinella GM, Hassner T, eds. Computer vision – ECCV 2022. 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XIV. Cham, Switzerland: Springer Nature Switzerland AG; 2022: 624-642. DOI: 10.1007/978-3-031-19781-9_36.

[9] Zhou T, Tulsiani S, Sun W, Malik J, Efros AA. View synthesis by appearance flow. In Book: Leibe B, Matas J, Sebe N, Welling M, eds. Computer vision – ECCV 2016. 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV. Cham, Switzerland: Springer International Publishing AG; 2016: 286-301. DOI: 10.1007/978-3-319-46493-0_18.

[10] Siyao L, Zhao S, Yu W, Sun W, Metaxas D, Loy CC, Liu Z. Deep animation video interpolation in the wild. 2021 IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR) 2021: 6587-6595. DOI: 10.1109/CVPR46437.2021.00652.

[11] Yuan S, Stenger B, Kim TK. RGB-based 3D hand pose estimation via privileged learning with depth images. arXiv Preprint. 2018. Source: <https://arxiv.org/abs/1811.07376>. DOI: 10.48550/arXiv.1811.07376.

[12] Ma N, Zhang X, Zheng HT, Sun J. ShuffleNet v2: Practical guidelines for efficient CNN architecture design. In Book: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, eds. Computer vision – ECCV 2018. 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part XIV. Cham, Switzerland: Springer Nature Switzerland AG; 2018: 116-131. DOI: 10.1007/978-3-030-01264-9_8.

[13] Cheng X, Chen Z. Video frame interpolation via deformable separable convolution. Proceedings of the AAAI Conference on Artificial Intelligence 2020; 34(7): 10607-10614. DOI: 10.1609/aaai.v34i07.6634.

[14] Lopez-Paz D, Bottou L, Schölkopf B, Vapnik V. Unifying distillation and privileged information. arXiv Preprint. 2016. Source: <https://arxiv.org/abs/1511.03643>. DOI: 10.48550/arXiv.1511.03643.

[15] Anil R, Pereyra G, Passos A, Ormandi R, Dahl GE, Hinton GE. Large scale distributed neural network training through online distillation. Int Conf on Learning Representations 2018:10-20.

[16] Ranjan A, Black MJ. Optical flow estimation using a spatial pyramid network. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017: 4161-4170. DOI: 10.1109/CVPR.2017.291.

[17] Chen X, Zhang Y, Wang Y, Shu H, Xu C, Xu C. Optical flow distillation: Towards efficient and stable video style transfer. In Book: Vedaldi A, Bischof H, Brox T, Frahm J-M, eds. Computer vision – ECCV 2020. 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI. Cham, Switzerland: Springer Nature Switzerland AG; 2020: 614-630. DOI: 10.1007/978-3-030-58539-6_37.

[18] Meister S, Hur J, Roth S. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. Proceedings of the AAAI Conference on Artificial Intelligence 2018; 32(1): 7251-7259. DOI: 10.1609/aaai.v32i1.12276.

[19] Luo K, Wang C, Liu S, Fan H, Wang J, Sun J. UPFlow: Upsampling pyramid for unsupervised optical flow learn-

ing. 2021 IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR) 2021: 1045-1054. DOI: 10.1109/CVPR46437.2021.00110.
[20] Lu G, Ouyang W, Xu D, Zhang X, Cai C, Gao Z. Dvc: An end-to-end deep video compression framework. 2019 IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR) 2019: 11006-11015. DOI: 10.1109/CVPR.2019.01126.
[21] Threading – Thread-based parallelism. 2025. Source: <https://docs.python.org/3.10/library/threading.html>.
[22] NumPy. NumPy Documentation. 2025. Source: <https://numpy.org/doc/>.
[23] OpenCV. 2025. Source: <https://opencv.org/>.
[24] PyTorch – Your new deep learning framework [In Russian]. 2025. Source: <https://habr.com/ru/post/334380/>.
[25] SciPy. 2025. Source: <https://scipy.org/>.
[26] Ding C, Lin M, Zhang H, Liu J, Yu L. Video frame interpolation with stereo event and intensity cameras. IEEE Transactions on Multimedia 2024:, 26: 9187-9202. DOI: 10.1109/TMM.2024.3387690.
[27] Wu Y, Wen Q, Chen Q. Optimizing video prediction via video frame interpolation. IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR) 2022: 17793-17802. DOI: 10.1109/CVPR52688.2022.01729.
[28] Yang B, Jiang G, Fang Y, Li W. Incorporating event information for high-quality video interpolation. Int Conf on Networking, Sensing and Control (ICNSC) 2023: 1-6. DOI: 10.1109/ICNSC58704.2023.10319006.
[29] Mahalingam V, Bhattacharya K, Ranganathan N, Chakravarthula H, Murphy RR, Pratt KS. A VLSI architecture and algorithm for Lucas-Kanade-based optical flow computation. IEEE Trans Very Large Scale Integr VLSI Syst 2010; 18(1): 29-38. DOI: 10.1109/TVLSI.2008.2006900.
[30] Farneback G. Two-frame motion estimation based on polynomial expansion. In Book: Bigun J, Gustavsson T, eds. Image analysis. 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29 – July 2, 2003 Proceedings. Berlin, Heidelberg: Springer-Verlag; 2003: 363-370. DOI: 10.1007/3-540-45103-X_50.

Author's information

Sergey Alexandrovich Zabuga, born in 2000, graduated from the Master's program in 2024 in the direction of "Radio Engineering" at Tomsk State University of Control Systems and Radioelectronics. Works as a technician in the laboratory of television automation at the Department of Television and Control. Research interests: digital image processing, pattern recognition, neural networks and machine learning. E-mail: *sergeizabuga@gmail.com* ORCID: 0009-0004-2402-2885.

Viacheslav Valerievich Kapustin, born in 1985, graduated from Tomsk State University of Control Systems and Radioelectronics in 2010, specializing in "Household Radioelectronic Equipment". Candidate of Technical Sciences, Associate Professor of the Department of Television and Control at TUSUR, Head of the Laboratory of Television Automation at TUSUR. Research interests: television systems, digital image processing, computer vision.
E-mail: *viacheslav.v.kapustin@tusur.ru* ORCID: 0000-0002-2293-0511.

Ilya Denisovich Musikhin, born in 2000, graduated from the Master's program in 2024 in the direction of "Radio Engineering" at Tomsk State University of Control Systems and Radioelectronics. Works as a technician in the laboratory of television automation at the Department of Television and Control. Research interests: digital image processing, recognition and tracking of objects, machine vision systems. E-mail: *ilia.d.musikhin@tusur.ru* ORCID: 0009-0001-7776-1698.

Received July 25, 2024. The final version – September 16, 2024.