JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125VOL 7, ISSUE 15, 2020

PROCESSING OF VIDEO INFORMATION IN UNMANNED AERIAL VEHICLES

E.B. Tashmanov, B.J. Saidboyev, M.A. Kamilov

Military Technical Institute of the National Guard of the Republic of Uzbekistan

Received: 16 March 2020 Revised and Accepted: 17 June 2020

ABSTRACT: The article discusses the problems of providing large video compression ratios without a noticeable deterioration in the visual quality of decoded images. To improve the efficiency of real-time coding of images, a brightness conversion method is proposed based on adaptive partitioning of images into blocks and restoration of their values during decoding. Experimental results on evaluating the effectiveness of the use of this method in processing a video stream are presented.

Keywords: unmanned aerial vehicles, video compression ratio, image encoding efficiency, video stream processing,MJPEG, MPEG.

I. Introduction

The effectiveness of the methods of warfare is determined by the quality indicators of weapons, reconnaissance, communications and automated control systems. The lack of modern intelligence and control systems does not fully realize the potential of weapons. The capabilities of the currently existing ground-based radar and optical-electronic reconnaissance systems are limited by the line of sight and do not provide for the detection of enemy targets and objects located behind natural shelters. The use of unmanned aerial vehicles (UAVs) for military purposes has become one of the important directions in the development of modern intelligence and allows automating command and control of troops and reducing the loss of personnel in battle due to operational intelligence information about the current situation.

However, the greatest value of this technology is that it can be used in combination with CCTV cameras as a solution to the complex tasks of ensuring enemy intelligence.

However, when converting an analog television signal into digital form, the output video stream can reach 240 - 800 Mbit / s, which for an hour of transmission is 108 - 360 GB. This requires a communication channel with a bandwidth of 120 - 400 MHz for their transmission and, accordingly, does not allow the transfer of such a huge amount of information in real time [1].

Thus, since the communication channels do not have such a wide bandwidth, special methods for compressing digital video data are used to match the parameters of the signals with the parameters of the communication channels.

Currently, in UAVs, codecs based on the standards of the MPEG family (H.264 / AVC, MPEG-2, etc.) and MJPEG are most often used for compressing TV images. MPEG codecs, which provide high compression efficiency for video data, as using the estimation and motion compensation algorithms take into account the interframe dependence of the video information samples. However, existing algorithms have great computational complexity. Codecs based on the MJPEG standard take into account only the intraframe dependence of video information samples and, therefore, have low computational complexity, but are less effective from the point of view of compression. In the codecs based on the MPEG and MJPEG family of standards, the principle of constant brightness is implemented, which initially degrades the clarity of the color details of the video image when thinning color difference samples (decomposition formats 4: 2: 2, 4: 2: 0, etc.). But the feature of the proposed option is a method that allows you to switch from working with rectangular blocks of a fixed size to blocks of arbitrary shape in segmented areas [2].

II. Main part

In the proposed method for processing TV images, it is carried out by dividing the image into small blocks of 2x2 size and gradually increasing the block size to 32x32.

The image processing algorithm by this method is divided into the following steps [3]:

1. The partition and finding the minimum values of the brightness of the blocks (2x2, 4x4, 8x8, 16x16 or 32x32);

2. In each block, we subtract the minimum values of the brightness of the blocks from each internal value of the block.

The procedure for splitting and finding the minimum values is as follows:

-The image is conditionally divided into 2x2 blocks;

- Inside each block (4 pixels) there is a minimum brightness value (hereinafter the minimum value of a 2x2 block);

ISSN- 2394-5125VOL 7, ISSUE 15, 2020

- The obtained minimum valuesof 2x2 blocks are compared with each other so that 4 values from neighboring blocks, together forming a 4x4 block, are compared with an error of 8. Moreover, if the difference in the values of 2x2 blocks does not exceed 8, then the lowest pixel value found will be the minimum value 4x4 block, i.e. four 2x4 blocks are combined into one 4x4, otherwise the minimum value does not change - the blocks remain separately. Combining and not combining blocks means that a given portion of the image is homogeneous or heterogeneous, i.e. there is a sharp change in the picture, respectively.

-Then, in the same way, the obtained values of the 4x4 block are compared if they exist, forming 8x4 blocks.

-Combining in the same way, first 8x8 blocks, we get 16x16 blocks, and after combining them, we get 32x32 blocks.

The results (minimum block values) are subtracted, respectively, from the internal block values.

If smaller blocks are combined, the pixel values are reduced in size to the minimum value of this block.

The block diagram of this algorithm is presented in Fig. 2.

In block 1 of this block diagram, the initial parameters of the counters of the number of blocks and the counters of the height and width of the blocks necessary for setting the cycles are set. Counters of the number of blocks are set for all types of blocks (i.e., a counter of the number of blocks, size 2x4, a counter of 4x4, etc.).

In block 2 of the flowchart, arrays are declared for the minimum values of minarray_n for each type of block: a separate cycle for the minimum values of 2x4 blocks (minarray_2), a separate cycle for the minimum values of 4x4 blocks (minarray_4), etc. up to 32x32. This algorithm uses blocking with a maximum size of 32x32 because found that a larger block size simply does not increase efficiency. Loops describing the declaration of arrays for each type of block have their own set of input parameters. So, for example, a cycle for specifying an array

ISSN- 2394-5125VOL 7, ISSUE 15, 2020

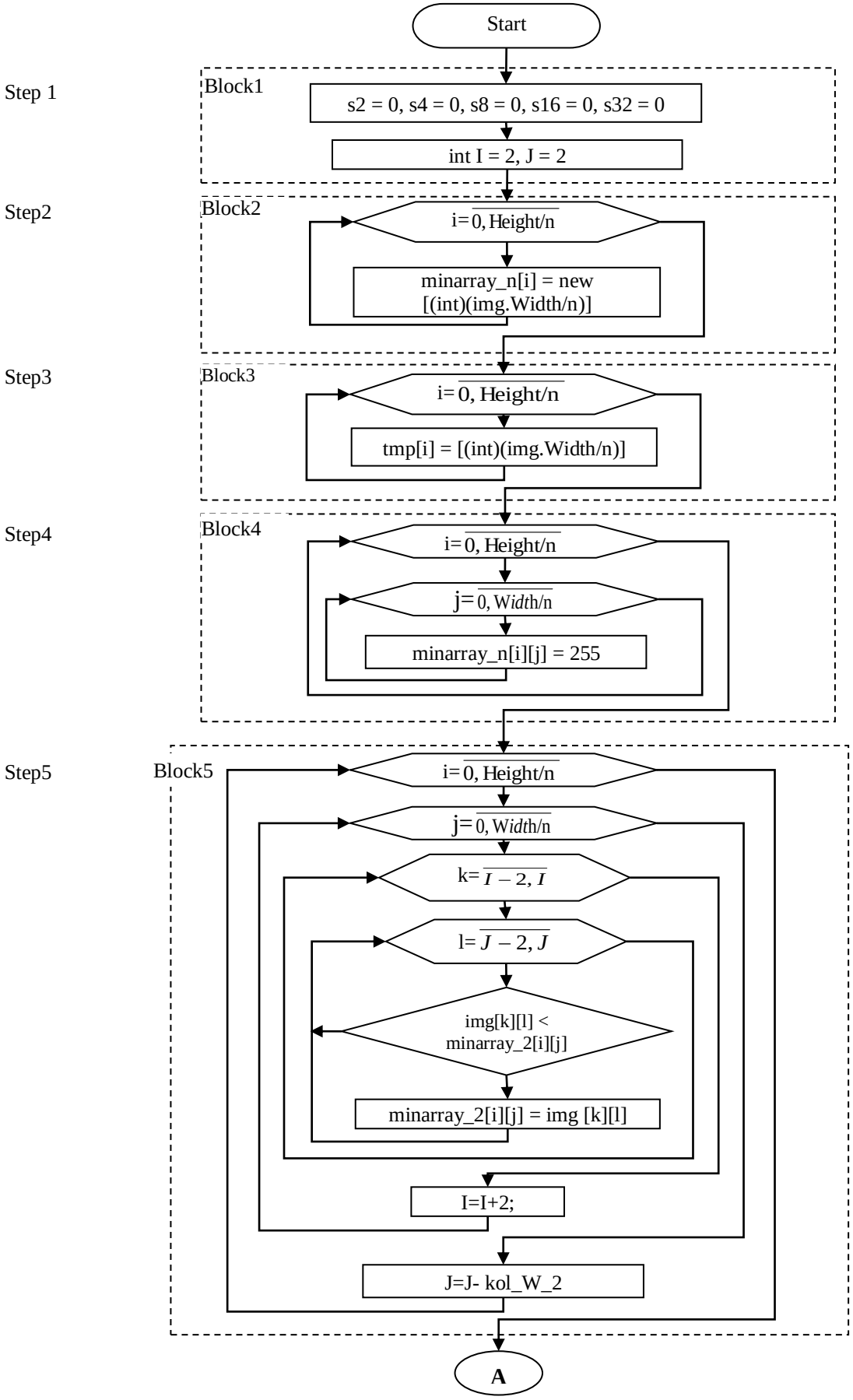

Fig. 2 (a). Block diagram of the luminance conversion algorithm based on adaptive blocking

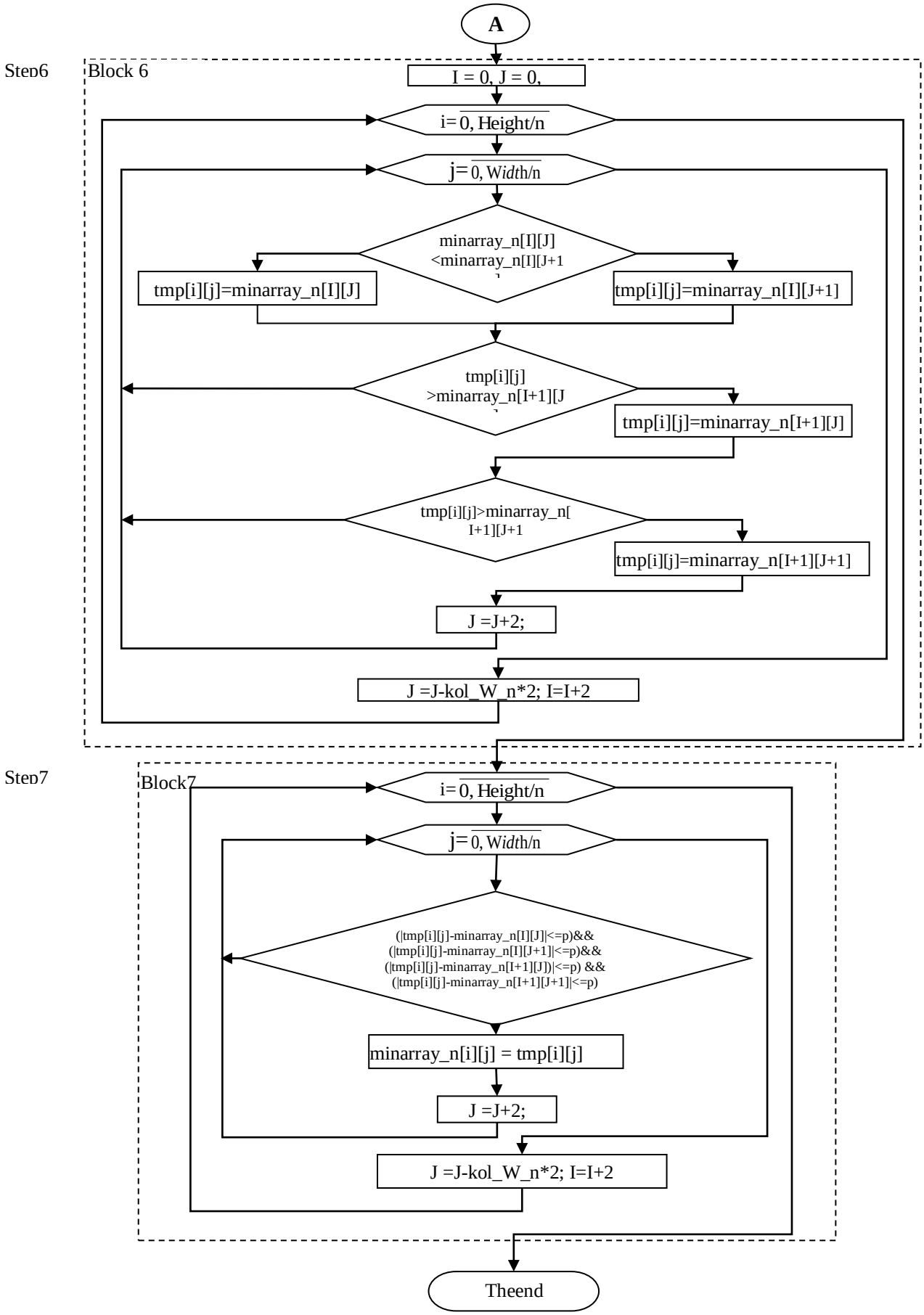

Fig. 2 (b). Block diagram of the luminance conversion algorithm based on adaptive blocking

ISSN- 2394-5125VOL 7, ISSUE 15, 2020

of minimum values of a 2x4 block "spins" to the values of the cycle counters i = Height / 2,, a 4x4 block - to values i = Height / 4, etc. In the general case, in block 2, this value is indicated as Height / n, where n is the value of the block size (2.4, etc.).

In block 3, an array is declared to temporarily store the found minimum values and then compare them with the newly found values.

In block 4 of this block diagram, the minimum pixel brightness values are set to 255, in order to avoid overflowing the array and to search for the minimum values.

The setting of values is also carried out in separate cycles with different parameters of the cycle counters for each type of block. Those.the cycle for setting the minimum valuesfor a 2x2 block has a counter parameter i = Height / 2, etc. For convenience, we denote this common variable as kol_H_x, where x is the block size (kol_H_2 for a 2x4 block, kol_H_4 for a 4x4 block, etc.).

From step 5, iterates over the image pixel by pixel and searches for the minimum value. It begins with the allocation of 2x2 blocks. Thus, in block 5, the minimum values of the brightness of pixels in blocks of 2x2 are determined. This happens as follows. The first block 2x2 pixels in size is conditionally allocated in the image, after which the pixel with the minimum brightness is determined by the enumeration method. This value is written as the minimum in the array of minimum values for 2x2 blocks. After that, in a cycle, a transition to neighboring pixels occurs, and the algorithm proceeds to step 6.

In block 6, the minimum values of the 4x4 blocks in the particular case and the enlarged blocks in the general case are determined. This means that if the minimum values for 2x4 blocks are found, then at this step, the minimum values for 4x4 blocks are searched, if found for 4x4, then the same algorithm searches for 8x8, 16x16 blocks, etc. The search is carried out according to the following principle: the minimum values of 2x4 blocks obtained in the previous step are compared with each other so that 4 values from neighboring blocks are compared, which together form a 4x4 block. Those. first, the minimum value for the first 2x2 block is compared with the minimum value of the block located under the first analyzed block. The value of the two of them, which will be less, is written to the array of temporary storage of minimum values described in block 3. Then, the temporary minimum value is compared with the minimum value of the block on the left and, finally, the block diagonally from the first analyzed block. Thus, in the temporary storage array, we have a minimum brightness value of 4 minimum values of 2x2 blocks, as well as 4 minimum values in an array of minimum values for 2x2 blocks.

In block 7, 4 values are compared from neighboring blocks together forming an enlarged block (in the first case, a 4x4 block, with repeated passes, a block is 8x4, 16x16, etc., respectively). The comparison is performed with an error of 8, because this is the most optimal error value for determining the uniformity of neighboring pixels, however this value can be varied. The comparison is based on the value of the difference module between each of the values of the array of 4 minimum values and the value of the minimum value in the temporary array. If the absolute value of this difference exceeds the specified error, then the selected sections are not sufficiently homogeneous and the partition of this section remains 2x2 blocks in size, and the analysis is transferred to the neighboring region, if the difference is less than the specified error, the section can be called homogeneous and 2x2 pixel blocks are combined into a single 4x4 block and the minimum value found from the temporary array is written for it.

The subsequent gradual increase in block sizes to 32x32 (combining homogeneous blocks) occurs according to the algorithm of step 6 and subsequent comparison with the specified error in step 7 with an increase in the value of the parameters of the cycles if the condition for homogeneity is fulfilled.

After that, the results obtained (minimum block values) are subtracted, respectively, from the internal block values.

To evaluate the effectiveness of this method, computer modeling of image processing was carried out, divided into blocks of non-fixed size, starting with blocks of 2x2 pixels in size, followed by a wavelet transform. At the same time, the coding was carried out in different color models: for the experiment, RGB and YUV 4: 4: 4 models were chosen.

In addition, the compression was carried out with different values of the threshold error: for the experiment, valuesof 100%, 80%, 20%, and 1% were taken. The choice of an error of 1% means that the minimum error is selected at which the minimum value found will be the minimum value of the 2x2 block.

Since it is necessary to save the metadata of block coding parameters to ensure the restoration of the original form of the converted images, metadata files were also compressed by several standards to evaluate the effectiveness of this method: ARJ, ZIP, RAR.

To demonstrate the results obtained by testing this method, the compression results of the Kamaz plot [4] are shown, as well as an assessment of its effectiveness, which are shown in Tables 1-2 and histograms in Fig. 3, 4.

Table 1.
The results of the compression of the plot "Kamaz" (in KB)

	Sourceframe		RGB				YUV 4:4:4			
	Exodus. thesize	Compressed frame(Kb)	Error				Sinfulness			
			100%	80%	20%	1%	100%	80%	20%	1%

ISSN- 2394-5125VOL 7, ISSUE 15, 2020

Lossless	1,1 Mb	317	319	322	334	308	330	330	333	315
20 times		90	90.9	91.8	89.3	74.2	69.3	69.3	67	53.7
50 times		50	50.4	51	47.1	36.6	36.8	36.9	33.4	23.4
Compressmetadatafiles										
ARJ	-	-	1.71	4.24	57.5	223	1.26	1.42	15.7	126
ZIP	-	-	1.86	4.35	57.5	225	1.36	1.51	15.7	128
RAR	-	-	1.53	4.25	50.3	170	1.11	1.26	15.5	100

In fig. 3 it can be seen that the use of coding errors of 100% and 80% gives the greatest efficiency, i.e. providing compression with the least loss. The second graph (Fig. 4) shows that the highest efficiency is provided by compression when encoding an image in the YUV 4: 4: 4 color model. This is due to the fact that this model provides better compression when using DCT, due to the use of different color-difference components. As you know, the human eye is more sensitive to the brightness of the color than to its hue, and this model takes into account just the brightness of the color and the number of red and blue color components. The 4: 4: 4 model is used just when processing images that are saturated with small details and rather sharp transitions in color-difference components and, at the same time, with a slight or even zero deviation in brightness.

Table 2.

Evaluation of the compression efficiency of the Kamaz plot (in KB)

	Average frame size in total with metadata (in KB)							
Model	RGB				YUV 4:4:4			
Error	100%	80%	20%	1%	100%	80%	20%	1%
Lossless	320,53	326,25	384,3	478	331,11	331,26	348,5	415
20 times	92,43	96,05	139,6	244,2	70,41	70,56	82,5	153,7
50 times	51,93	55,25	97,4	206,6	37,91	38,16	48,9	123,4
	Compressionefficiency							
Model	RGB				YUV 4:4:4			
Error	100%	80%	20%	1%	100%	80%	20%	1%
Lossless	0,99	0,97	0,82	0,66	0,96	0,96	0,91	0,76
20 times	0,97	0,94	0,64	0,37	1,28	1,28	1,09	0,59
50 times	0,96	0,9	0,51	0,24	1,32	1,31	1,02	0,41

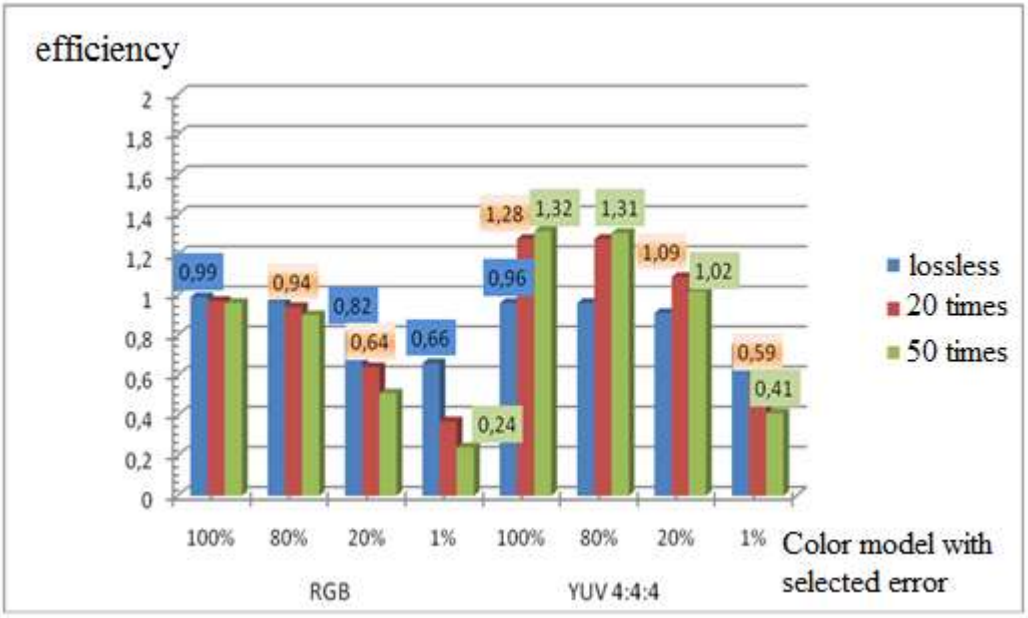

Fig. 3. Evaluation of the effectiveness of image compression during processing in different color models.

ISSN- 2394-5125VOL 7, ISSUE 15, 2020

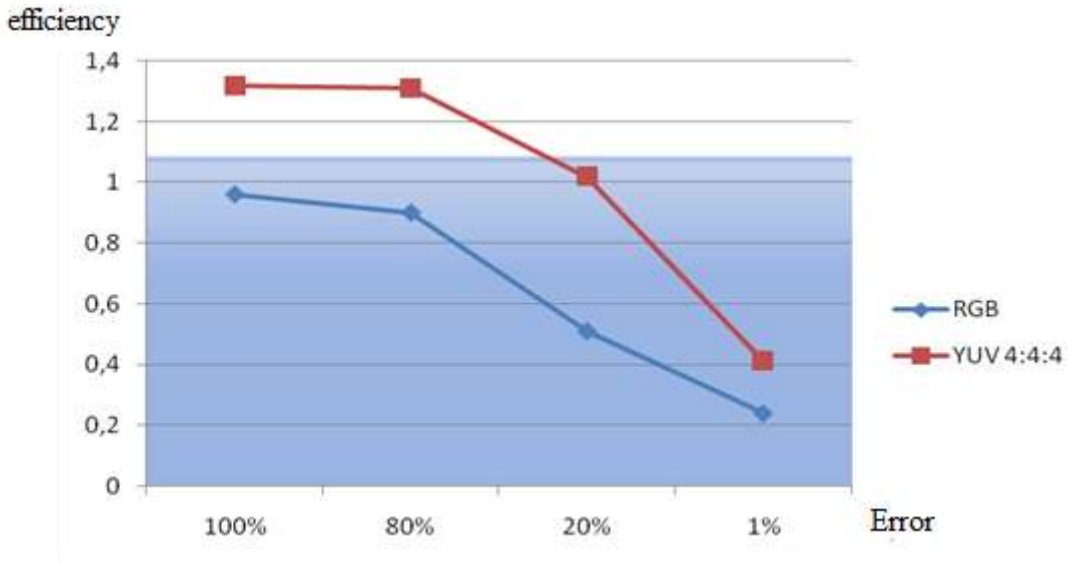

Fig. 4. Evaluation of the image compression efficiency during processing in RGB and YUV 4: 4: 4 color models with a compression parameter of 50 times

However, the use of this model leads to a significant increase in the speed of the digital stream, and in this study, its priority is to reduce it.

Based on all this, it is possible to evaluate the high efficiency of the proposed method for segmenting homogeneous regions into blocks of non-fixed size, however, the efficiency of this algorithm with respect to the RGB model should be increased to obtain a better result compared to the YUV 4: 4: 4 model.

III. Conclusion

As a result of the experiments, it was also found that the highest efficiency is provided by compression during image encoding using the brightness conversion algorithm based on adaptive splitting in the YUV 4: 4: 4 color model. However, the use of this model leads to a significant increase in the speed of the digital stream, and to ensure signal processing in real time, the priority is to reduce it to ≈40 ms.

IV. References

1. Tashmanov E.B. On the task of managing digital image compression TATU messages. Tashkent. 2015. – № 2(34) . – C. 77-81.

2.Tashmanov EB, Puziy AN Analysis of image compression methods based on Wavelet transforms // Vestnik TUIT.-Tashkent,2015.-№1.-стр.21-25

3. Tashmanov E.B., Vinogradov A.S., Glukhov E.V. Compression of images using structural lines // Information Communications: Networks - Technologies - Solutions. Tashkent –2018. № 2(46). – C. 5-12.

4. Tashmanov E.B., Vinogradov A.S. Image compression by wavelet transform in video information systems // Bulletin of the Military Technical Institute of the National Guard of the Republic of Uzbekistan. Tashkent – 2018. №4. –C.127-129.

5. Tashmanov E.B., Vinogradov A.S., Glukhov E.V. Compression of television images in conditions of redundancy of information // Uzbek journal "Problems of Informatics and Energy". Tashkent –2018. №3 – C. 78-82.