ALGORITHMIZATION OF ANALYTICAL METHODS FOR FINDING MOTION VECTORS WHEN PROCESSING IMAGE SERIES

O. Ja. Kravets(1), I. A. Aksenov(2), P. A. Rahman(3)*

(1) Voronezh state technical university, Voronezh; (2) Vladimir State University named after Alexander and Nikolay Stoletovs, Vladimir; (3) Ufa State Petroleum Technological University, Ufa
Russian Federation

* Corresponding Author, e-mail: csit@bk.ru

Abstract: Modern monitoring and operational decision-making systems based on image series have become widely used for a wide variety of applications. Among such systems, we will single out real-time monitoring, the requirements for an image processing system in which are very strict. At the same time, the traditionally high complexity of images is in contradiction with the need for fast processing. One of the ways to resolve the contradiction is to represent a series of images as a sequence of additions and changes to the previous image in the next. An important component of such a representation is the search for motion vectors and the corresponding algorithms for searching for such vectors. A new algorithm based on an analytical solution is obtained and provides a multiple compression ratio by eliminating the temporary redundancy of data determined by a high correlation between adjacent frames of video sequences.

Key words: motion vectors, image series, fast processing, analytical solution, machine graphics.

1. INTRODUCTION

Modern monitoring and operational decision-making systems based on image series have become widely used for a wide variety of applications. Among such systems, we will single out real-time monitoring, the requirements for an image processing system in which are very strict. At the same time, the traditionally high complexity of images is in contradiction with the need for fast processing. One of the ways to resolve the contradiction is to represent a series of images as a sequence of additions and changes to the previous image in the next.

2. METHODS FOR FINDING MOTION VECTORS

The set of basic analytical solutions for image processing includes the following methods for finding motion vectors [1]:

- optical flow method. The analysis of approaches to calculating the optical flow was carried out in [2];
- the phase correlation method, the description of which is presented in [3];
- the method of global motion estimation, the main idea of which is described in [4];
- block-by-block methods of motion estimation [5, 6] and various combinations of these approaches.

Neighboring frames, as a rule, have a high correlation coefficient. Taking into account this circumstance, in the displacement compensation algorithm, it is necessary to take into account the large redundancy of neighboring frames. As a result, the processing speed of the stream increases, which is important in real-time tasks. The difference between frames is processed much faster than the whole frame.

3. BLOCK-BY-BLOCK METHOD OF MACHINE GRAPHICS FOR MOTION ESTIMATION

We propose to modify the analytics [6], which decomposes each frame into separate components. These components are compared with similar ones in neighboring frames. This comparison is the basis for finding motion vectors. As a result, it becomes possible to implement the synthesis of an intermediate frame.

A compensated or intermediate frame is one of the neighboring frames, in which part of the information is contained in the form of motion vectors of the blocks of the previous frame.

We are also interested in the reference frame, the main frame. It is essentially a data carrier about the original image. In the process of searching for motion vectors, intermediate frames basically use a reference frame. You can adjust the frequency of occurrence of such frames, it is determined by the dynamism of the object [6].

So, we break the frames into blocks. Thus, each frame as a set of intensities A_t is a set of fixed-size $m*k$ blocks $A_t(i, j)$ with coordinates i, j [1].

The values m and k are elements of the algorithm adjustment in terms of its accuracy and computational complexity [5]. The fewer blocks – the fewer motion vectors. It should be understood that block partitioning almost never coincides with the boundaries of mobile objects. Hence the need for significant grinding of blocks. But such grinding entails an increase in the number of motion vectors. Thus, excessive grinding leads to a long processing time and a decrease in quality.

4. ANALYTICAL FOUNDATIONS FOR CREATING AN ALGORITHM

The problem of two-dimensional motion estimation is reduced to finding a similar block $f(\vec{x}, n)$ in the previous image $f(\vec{x}, n-1)$ within a certain search area, where $\vec{x}$ is a two-dimensional position in the image, and n is the number of the image, such a vector field $t\vec{d}(\vec{x}, n)$ that the following expression is executed, represented by formula (1) [6, 7]:

$$f(\vec{x} - \vec{d}(\vec{x}, n), n-1) \approx f(\vec{x}, n) \quad (1)$$

The principle under consideration, on which many algorithms for matching blocks are based, including those implemented within the framework of this study, proceeds from the fact that the motion vector is the same for a certain block of pixels, therefore equation (2) is valid:

$$\vec{d}(\vec{x}, n) = \vec{d}(\vec{x}', n) \quad \forall \vec{x}' \in B(\vec{x}) \quad (2)$$

where $B(\vec{x})$ is a block of pixels at the position $\vec{x}$.

The search area for motion vectors is determined by the formula (3) [1]:

$$O = \{(x, y) \mid x \in [-a_{max}, a_{max}], y \in [-b_{max}, b_{max}] \quad (3)$$

It must be remembered that the maximum number of settlement operations is carried out here. Therefore, the parameters a_{max}, b_{max}>0 are identified to ensure the necessary accuracy [7].

An effective solution to the problem of two-dimensional motion estimation is achieved by the developed block search algorithm described below.

The main basic solutions for finding the similarity measure of parts of images used by the motion compensation algorithm are presented by formulas (4)-(6) [8]:

$$SAD(\text{Sum of absolut differences}) = \sum_x \sum_y |B_1(x, y) - B_2(x, y)| \to \min \quad (4)$$

$$SAD(\text{Sum of squared differences}) = \sum_x \sum_y (B_1(x, y) - B_2(x, y))^2 \to \min \quad (5)$$

$$CC(\text{Cross} - \text{correlation}) = \sum_x \sum_y B_1(x, y) * B_2(x, y) \to \max \quad (6)$$

where x,y are pixel coordinates, B_1 is the block of the previous frame, B_2 is the block of the current frame.

5. IMAGE PROCESSING ALGORITHM FOR DETECTING DYNAMIC OBJECTS

The developed image processing algorithm for detecting dynamic objects is presented below, which includes a set of basic analytical solutions for identifying images, as well as a proposed solution for block-by-block evaluation of movements based on the principle of motion compensation:

1. Entering a sequence of images A_t, t=0…n-1.

2. Entering the parameter of the frequency of formation of key frames p.
3. Entering the block size $m*k$ in pixels.
4. Definition of the search area O, described by the formula (3).
5. Cycle through the images with step 1. If $t \geq n$, then terminate the algorithm.
6. If A_t is the keyframe, then save the frame and go to step 5.
7. To form an intermediate frame according to the developed hierarchical adaptive algorithm of machine graphics based on orthogonal search, presented below.
8. Save the intermediate frame.
9. Analysis of the presence of lost objects by the obtained motion vectors.
10. If there are no objects that have left the observation area, then proceed to step 5.
11. Formation of areas of interest for lost objects.
12. Identification and sending of samples for further analysis, in case of similarity with the template of the object of interest. Go to step 5.

We use the quasi-linear metric SAD. It is based on calculating the difference between blocks as the sum of the multidimensional absolute difference between individual pixels of blocks.

A set of basic rules was formulated for the stages of forming areas and analyzing the presence of lost objects within the framework of the developed algorithm for detecting dynamic objects.

A set of motion vectors V_i reflecting the movement of a dynamic object O_i is assembled according to the following rule:

If for any t for two adjacent blocks I_t^{xy}, I_t^{ab}, *where* $x-1 \leq a \leq x+1$, $y-1 \leq b \leq y+1$, *is executed* $x = a + \alpha$, $y = b + \beta$ *and* α, β *the constants, then* I_t^{xy}, I_t^{ab} *refer to a single dynamic object.*

The analysis of the presence of lost objects is carried out after each newly obtained vector of side movements according to the following rule:

If no new motion vector O_i *was found for the set* V_i *reflecting the movement of a dynamic object, obtained after processing the next frame from the analyzed video sequence, then the object is considered lost.*

6. FORMATION OF AREAS OF INTEREST FOR FURTHER IDENTIFICATION OF THE IMAGE OF A DYNAMIC OBJECT

The formation of areas of interest for further identification of the image of a dynamic object O_i is carried out by a set of blocks, the movement of which describes a set of received vectors V_i.

Figure 1 shows a diagram of block relationships using the example of a video sequence of 6 frames. Here I_t^{xy} is a sequence of frames, where t=0...5 is the sequence number of the frame; x,y=0...5 are the coordinates of the block on the frame I_t.

0

I_0^{00}	I_0^{01}	I_0^{02}	I_0^{03}	I_0^{04}	I_0^{05}
I_0^{10}	I_0^{11}	I_0^{12}	I_0^{13}	I_0^{14}	I_0^{15}
I_0^{20}	I_0^{21}	I_0^{22}	I_0^{23}	I_0^{24}	I_0^{25}
I_0^{30}	I_0^{31}	I_0^{32}	I_0^{33}	I_0^{34}	I_0^{35}
I_0^{40}	I_0^{41}	I_0^{42}	I_0^{43}	I_0^{44}	I_0^{45}
I_0^{50}	I_0^{51}	I_0^{52}	I_0^{53}	I_0^{54}	I_0^{55}

1

I_0^{00}	I_1^{01}	I_1^{02}	I_0^{03}	I_0^{04}	I_0^{05}
I_0^{10}	I_0^{01}	I_0^{02}	I_0^{13}	I_0^{14}	I_1^{15}
I_0^{20}	I_0^{21}	I_0^{22}	I_0^{23}	I_0^{15}	I_1^{25}
I_0^{30}	I_0^{31}	I_0^{32}	I_0^{33}	I_0^{25}	I_1^{35}
I_0^{40}	I_0^{41}	I_0^{42}	I_0^{43}	I_0^{44}	I_0^{45}
I_0^{50}	I_0^{51}	I_0^{52}	I_0^{53}	I_0^{54}	I_0^{55}

2

I_0^{00}	I_2^{01}	I_2^{02}	I_0^{03}	I_0^{04}	I_0^{05}
I_0^{10}	I_1^{01}	I_1^{02}	I_0^{13}	I_0^{14}	I_1^{15}
I_0^{20}	I_0^{01}	I_0^{02}	I_0^{23}	I_0^{24}	I_2^{25}
I_0^{30}	I_0^{31}	I_0^{32}	I_0^{15}	I_1^{25}	I_0^{35}
I_0^{40}	I_0^{41}	I_0^{42}	I_0^{25}	I_1^{35}	I_0^{45}
I_0^{50}	I_0^{51}	I_0^{52}	I_0^{53}	I_0^{54}	I_0^{55}

3

I_0^{00}	I_2^{01}	I_2^{02}	I_0^{03}	I_0^{04}	I_0^{05}
I_0^{10}	I_0^{11}	I_0^{12}	I_0^{13}	I_0^{14}	I_1^{15}
I_0^{20}	I_1^{01}	I_1^{02}	I_0^{23}	I_0^{24}	I_2^{25}
I_0^{30}	I_0^{01}	I_0^{02}	I_0^{33}	I_0^{34}	I_0^{35}
I_0^{40}	I_0^{01}	I_0^{15}	I_1^{25}	I_0^{44}	I_0^{45}
I_0^{50}	I_0^{51}	I_0^{25}	I_1^{35}	I_0^{54}	I_0^{55}

4

I_0^{00}	I_2^{01}	I_2^{02}	I_0^{03}	I_0^{04}	I_0^{05}
I_0^{10}	I_0^{11}	I_0^{12}	I_0^{13}	I_0^{14}	I_1^{15}
I_0^{20}	I_0^{21}	I_0^{22}	I_0^{23}	I_0^{24}	I_2^{25}
I_0^{30}	I_1^{01}	I_1^{02}	I_0^{33}	I_0^{34}	I_0^{35}
I_0^{40}	I_0^{01}	I_0^{02}	I_0^{43}	I_0^{44}	I_0^{45}
I_0^{50}	I_0^{15}	I_1^{25}	I_0^{53}	I_0^{54}	I_0^{55}

5

I_0^{00}	I_2^{01}	I_2^{02}	I_0^{03}	I_0^{04}	I_0^{05}
I_0^{10}	I_0^{11}	I_0^{12}	I_0^{13}	I_0^{14}	I_1^{15}
I_0^{20}	I_0^{21}	I_0^{22}	I_0^{23}	I_0^{24}	I_2^{25}
I_0^{30}	I_0^{31}	I_0^{32}	I_0^{33}	I_0^{34}	I_0^{35}
I_0^{40}	I_1^{01}	I_1^{02}	I_0^{43}	I_0^{44}	I_0^{45}
I_0^{50}	I_0^{01}	I_0^{02}	I_0^{53}	I_0^{54}	I_0^{55}

Fig. 1. Diagram of the relationship between the key and intermediate frames

Description of the key and intermediate frame formation scheme:

- The frame I_0 is the key and contains all the information about the blocks in its original form.
- The frame I_1 and all subsequent ones are intermediate and include only information about the parts of the image that are missing from the previous frames and the block motion vector.

The frame in question includes data about the blocks I_1^{01}, I_1^{02}, I_1^{25}, I_1^{35} and I_1^{15}, as well as clarifying information about the movement I_0^{01}, I_0^{02}, I_0^{15} and I_0^{25}.

- The frame I_2 includes new blocks I_2^{01}, I_2^{02} and clarifying information about the motion vectors of the blocks I_0^{01},I_0^{02},I_0^{15}, I_0^{25}, I_1^{01},I_1^{02}, $I_1^{2,5}$ and $I_1^{3,5}$.

- The resulting list of coordinates of side movements at different points in time is presented in Table 1.

Table 1. The coordinates of the blocks on the frame moment in time t

Block \ t	0	1	2	3	4	5
I_0^{01}	(0,1)	(1,1)	(2,1)	(3,1)	(4,1)	(5,1)
I_0^{02}	(0,2)	(1,2)	(2,2)	(3,2)	(4,2)	(5,2)
I_1^{01}	-	(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
I_1^{02}	-	(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
I_0^{15}	(1,5)	(2,4)	(3,3)	(4,2)	(5,1)	-
I_0^{25}	(2,5)	(3,4)	(4,3)	(5,2)	-	-
I_1^{25}	-	(2,5)	(3,4)	(4,3)	(5,2)	-
I_1^{35}	-	(3,5)	(4,4)	(5,3)	-	-

Based on the tabular values of block movements, it can be concluded that there are two dynamic objects: V_1, including blocks I_0^{01}, I_0^{02}, I_1^{01}, I_1^{02}, and V_2, including blocks I_0^{15}, I_0^{25}, I_1^{25}, I_1^{35}.

7. HIERARCHICAL ADAPTIVE ALGORITHM OF MACHINE GRAPHICS BASED ON ORTHOGONAL SEARCH

There is a large number of image processing algorithms for block-by-block images for block-by-block evaluation of the movements of dynamic objects on video sequences. The highest accuracy is provided by the Full Search algorithm, which evaluates all possible candidate blocks within the search area. However, the estimation of movements based on it requires too much computing and is unacceptable for real-time systems. In this regard, in this study, a search was carried out for alternative algorithms for block estimation of motion and the possibilities of combining them to ensure the greatest efficiency of calculations.

The analysis of the feasibility of using the motion detection algorithm on video sequences in real time was carried out among the basic analytical solutions for calculating block-by-block displacements described in [1, 5]:

- Binary search, in which the search area is divided into 9 parts, in each of which motion estimates are calculated for the initial points, which are usually in the center. A full search is performed in the zone with the best rating.
- Three-step search, considering 8 blocks at an initial distance from the analyzed block, conducting an assessment for each of the initial points, reducing the distance between the centers of the blocks and shifting to a point with minimal distortion. The steps are repeated until the initial step is less than 1. The main problem of the

algorithm is the homogeneous arrangement of points at the first step and, as a result, not efficiency for areas of small movement.

- A four-step search that selects nine blocks in the search area at a distance of 2 from each other, determining the block with the best motion estimate. If this block is the center of the search area, then exit, otherwise offset to the block with the least distortion and repeat the iteration. This algorithm shows orientation to the center of the frame and, as a rule, shows good efficiency for complex movement options.
- Rhombic search, which works the same as a four-step search, but has an unlimited number of steps and uses 2 templates (small and large) of a fixed size of a rhombic shape.
- Logarithmic search, which requires more steps and provides the possibility of achieving greater accuracy. At the first step, 5 blocks are evaluated according to a template in the form of a "+" sign. In the second step, if the central block has less distortion, then the interblock distance is halved, otherwise the search center is shifted to the edge block. Steps 1 and 2 are repeated until the distance between the blocks is equal to 1. Out of 9 evaluated blocks, the block with the best score is selected.
- Orthogonal search, which is a combination of three-step and logarithmic search algorithms. At the first step, the interblock distance is selected equal to half of the maximum possible offset in the search area. In the second step, three blocks are evaluated horizontally and the search center is shifted to the block with the best estimate. Then similar calculations are performed vertically. At the third step, if the interblock distance is greater than 1, then it is halved, otherwise the output is. The search for a similar block using an orthogonal algorithm is shown in Figure 2.
- Spiral search, taking into account that the closer the candidate block is to the analyzed block, the higher the probability that it has the greatest similarity with it. The accuracy of the algorithm decreases from the center to the edges of the search area.
- Template search, taking into account the translational movement of objects and the assumption that the candidate block inside the selected search template has a similar motion vector as the analyzed block.

To conduct a comparative analysis, the peak signal-to-noise ratio calculated by the formula (7) and fairly objectively reflecting the proximity of the reconstructed image from the intermediate frame with the original one is taken as an efficiency parameter [9]:

$$\mathrm{PSNR} = 10\log_{10}\left[\frac{\mathrm{MAX}_I^2}{\mathrm{MSE}}\right] \tag{7}$$

where MSE (Mean Square Error) is the RMS error calculated by the formula (8), MAX_I is the maximum value taken by the pixel of the image I:

$$\mathrm{MSE} = \frac{1}{m*n}\sum_{x=1}^{m}\sum_{y=1}^{n}\left[I_t(x,y) - I_{t-1}(x,y))\right]^2 \tag{8}$$

where $I_{t-1}(x,y)$ and $I_t(x,y)$ is the brightness of the original and restored frame at the point (x,y), $m*n$ is the block size.

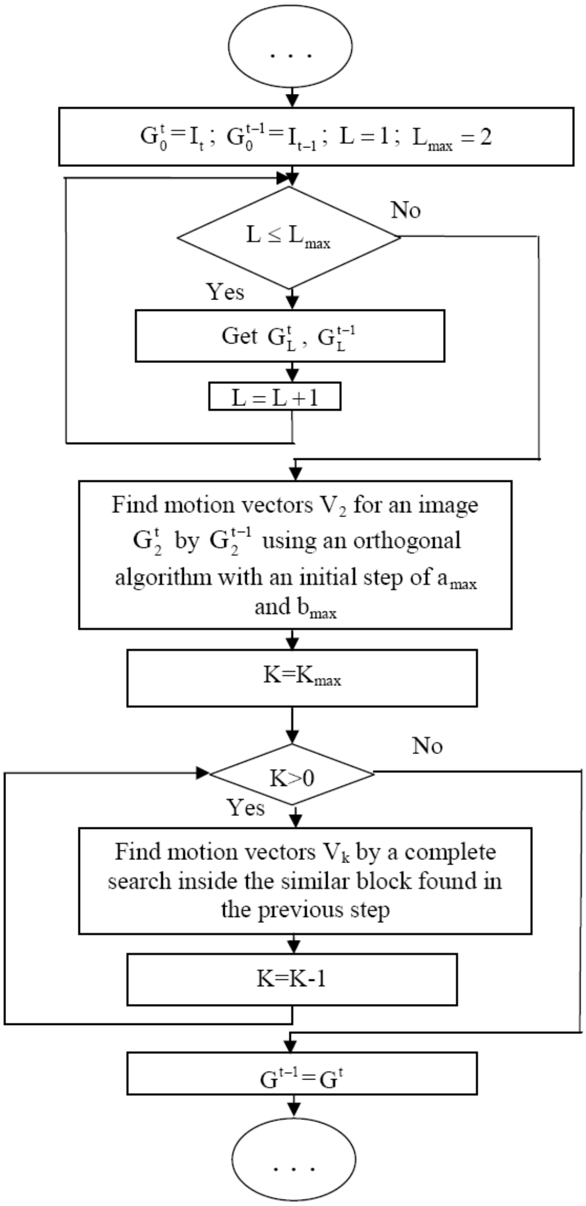

Fig. 2. The developed hierarchical adaptive orthogonal search algorithm

The higher the PSNR value, the closer the restored image is to the original image [10, 11].

The analyzed video sequence consisted of 3025 frames with complex and multi-speed variants of the movement of dynamic objects with a spatial resolution of 320x256 pixels and a time sampling of 25 frames/second. The size of the partition block is 16x16.

8. RESULTS OF COMPARATIVE ANALYSIS OF BLOCK SEARCH METHODS

The results of a analysis of block search methods are contained in Table 2.

Table 2. The average value of the frame-by-frame coefficient PSNR, dB

Algo-rithm	Binary	Three-Step	Four-Step	Rhombic	Loga-rithmic	Ortho-gonal	Spiral	Template
PSNR	27,012	33,231	32,019	33,732	32,099	35,152	34,105	33,728

Based on the experimental results, the orthogonal algorithm is the most effective among the considered analytical solutions for image processing for block-by-block evaluation of movements. However, in order to achieve even greater accuracy in detecting dynamic objects, a hierarchical adaptive search algorithm was developed using a combination of an orthogonal approach at the upper level and a full search method to refine the result inside a small search area at the lower level of the hierarchy.

Hierarchical search as an additional analytical solution for combination with orthogonal search allows for refinement of displacement vectors at each of the motion estimation operations, as well as to ensure resistance to high-frequency noise by performing analysis on reduced copies of images.

Orthogonal search, used to build an advanced image processing algorithm for motion estimation, involves adaptation to the predominantly horizontal or vertical movement of dynamic objects by introducing two parameters to set the value a_{max} of the initial step horizontally and b_{max} vertically.

To adapt the algorithm to the preferred direction of movement within a specific observation scene, we take into account: $a_{max} > b_{max}$ - predominantly horizontal movement of dynamic objects, $a_{max} < b_{max}$ - predominantly vertical movement of dynamic objects.

Configuring the algorithm for a specific observation scene involves reducing the computational complexity of the algorithm and, consequently, increasing the detection speed without significant loss of quality.

Figures 3, 4 show diagrams demonstrating the operation of orthogonal and hierarchical algorithms, respectively.

The progress of the orthogonal machine graphics algorithm for searching for a block $I_t(x, y)$ on a frame I_{t-1}:

1. Set the coordinates x, y of the search center on the frame I_{t-1} and the search area O. In Figure 3, the coordinates of the center are x=5, y=8, and the search area is $a_{max} = 4$ and $b_{max} = 4$.

2. If $a_{max} > 1$, then a block in the center of the search and two blocks in the horizontal direction from it with a step $-a_{max}$ and are considered $+a_{max}$. The search center is shifted to the block with the smallest sum of absolute pixel differences calculated by the formula (4). In Figure 3, the coordinates are x=9, y=8 on the first iteration, x=7, y=4 on the second iteration and x=8, y=2 on the third.

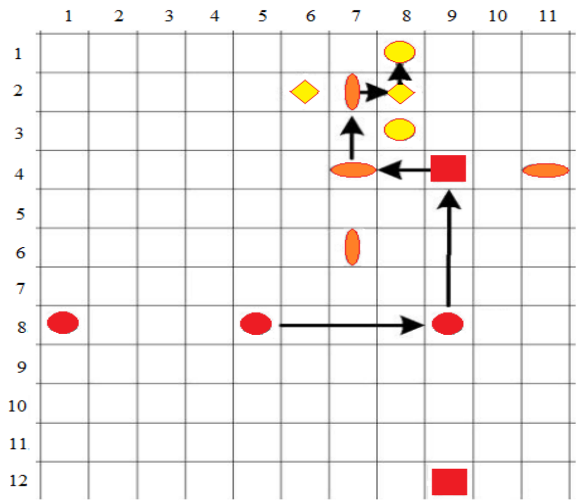

Fig. 3. Orthogonal search scheme

3. If $b_{max} > 1$, then the evaluation of the central block and the other two located in the vertical direction with a step $-b_{max}$ and $+b_{max}$ is carried out . The search center is shifted to the block with the smallest sum of absolute pixel differences. In Figure 3, the coordinates are x=9, y=4 on the first iteration, x=7, y=2 on the second iteration and x=8, y=1 on the third.

4. Halving the step: $a_{max} = a_{max}/2$, $b_{max} = b_{max}/2$. If $a_{max} \leq 1$ and $b_{max} \leq 1$, then the algorithm is completed, otherwise go to step 2.

The images I_t, I_{t-1} are presented in the form of a two-level pyramid. At the zero level of G_0, the frames have the original size. Each next level is constructed according to the formula (9):

$$G_L(x, y) = \frac{1}{4} \sum_{u=0}^{1} \sum_{v=0}^{1} G_{L-1}(2x + u, 2y + v) \qquad (9)$$

where $G_L(x, y)$ is the brightness level of a pixel with coordinates (x,y) at the L level.

One pixel of the upper level corresponds to 4 pixels of the middle level and 16 pixels of the lower level, respectively. At the upper level, a rough estimate is performed using an orthogonal algorithm, and at all subsequent ones, a full search is performed in a small window.

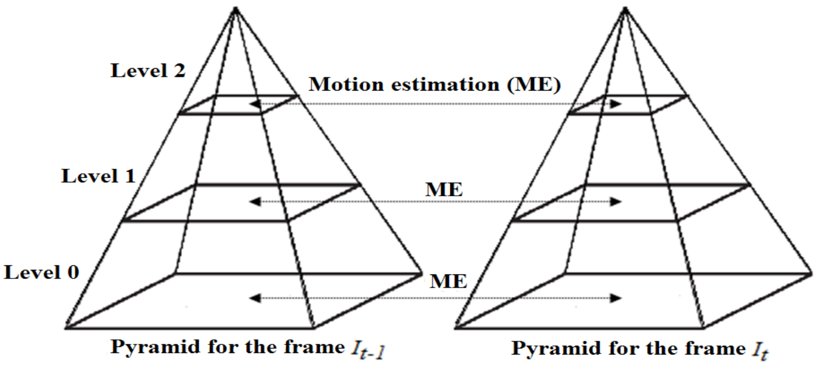

Fig. 4. Hierarchical search scheme

9. CONCLUSION

Based on the experimental results, the orthogonal algorithm is the most effective among the considered analytical solutions for image processing for block-by-block evaluation of movements. However, in order to achieve even greater accuracy in detecting dynamic objects, a hierarchical adaptive search algorithm was developed using a combination of an orthogonal approach at the upper level and a full search method to refine the result inside a small search area at the lower level of the hierarchy.

Hierarchical search as an additional analytical solution for combination with orthogonal search allows for refinement of displacement vectors at each of the motion estimation operations, as well as to ensure resistance to high-frequency noise by performing analysis on reduced copies of images.

The analysis of the hierarchical adaptive orthogonal search algorithm based on the video sequence used to evaluate the performance of the algorithms described above showed an increase in the value of the PSNR parameter to 38.842, which indicates an increase in the proximity of the recovered images to the original ones after processing by the developed image processing algorithm.

REFERENCES

[1] Yang J., Won K., Jeon B. Motion vector coding with selection of an optimal predictive motion vector. *Optical Engineering*, No 1 (vol. 48), 2009, art. No 010501. DOI: 10.1117/1.3070632.

[2] Kumashiro, K., Steinberg, A.M., Yano, M. A functional error analysis of differential optical flow methods. *Experiments in Fluids*, No 8 (vol. 62), 2021, art. No 159. https://doi.org/10.1007/s00348-021-03244-1.

[3] Yang L. J., Hao W., Li C. L. A Modified Phase Correlation Method for Image Registration. *Applied Mechanics and Materials*, vol. 48-49, 2011, pp. 48-51.

[4] Chung Y.-C., He Z. Reliability Analysis for Global Motion Estimation. *Signal Processing Letters, IEEE*, No 11 (vol. 16), 2009, pp. 977-980.

[5] Zhang M., Yang X., Cao Y., Numerical analysis of block-by-block method for a class of fractional relaxation-oscillation equations, *Applied Numerical Mathematics*, vol. 176, 2022, pp. 38-55.

[6] Wang W., Li W. A Multiple Pattern Algorithm for Block Motion Estimation Based on Variable Block-size. *Acta Automatica Sinica*, No 1 (vol. 35), 2009, pp. 34-38.

[7] Wikhe A., Agrawal J., Bankar M., Baviskar P. Analysing Predictive Coding Algorithms for Document Review. *International Journal for Research in Applied Science & Engineering Technology*, No 9 (vol. XI), 2009, pp. 1679-1691.

[8] Zhang L., Wang B. A new expression for sculling motion compensation algorithms. *7th World Congress on Intelligent Control and Automation*, 2008, pp. 7270-7273.

[9] Moreno J., Jaime B., Saucedo S. Towards No-Reference of Peak Signal to Noise Ratio. *International Journal of Advanced Computer Science and Applications*, No 1 (vol. 4), 2012, pp. 123-130.

[10] Metkar S., Talbar S. Fast Motion Estimation Using Modified Orthogonal Search Algorithm for Video Compression. *Signal Image and Video Processing*, No 1 (vol. 4), 2010, pp. 123-128.

[11] Liu S., Bai W., Liu G., Li W., Srivastava H. M. Parallel Fractal Compression Method for Big Video Data. *Complexity*, vol. 2018, 2018, Article ID 2016976. https://doi.org/10.1155/2018/2016976

Information about the authors:

Oleg Jakovlevich Kravets – Dr. Sci (IT), professor of Voronezh state technical university, areas of scientific research – system analysis, optimization, simulation of complex objects

Ilia Antonovich Aksenov – associate professor of Vladimir State University named after Alexander and Nikolay Stoletovs, areas of scientific research – application of information technologies in economic systems

Pavel Azizurovich Rahman, PhD, associate professor of Ufa State Petroleum Technological University, areas of scientific research - information technology, computer networks, reliability modeling and engineering

Manuscript received on 14 March 2022