Websites:
http://www.sciencepub.net/newyork
http://www.sciencepub.net

Emails:
newyorksci@gmail.com
editor@sciencepub.net

New York Science Journal

MARSLAND PRESS
Multidisciplinary Academic Journal Publisher

Mathematical Model Of Geometric Characteristics Of Images Based On the method Deconvolution Procedure

Saida Safibullayevna Beknazarova

Doctor Of Technical Science, Professor Audiovisual Technologies Of Tashkent University Of Information Technologies Named After Muhammad Al-Khwarizmi, Tashkent, Uzbekistan
saida.beknazarova@gmail.com

Abstract: Finding and tracking moving objects in a video stream is an important task for a different class of tasks that need to allocate moving objects on a complex background. Image segmentation based on mask selection is considered for solving this class of problems due to the fact that changing the parameters of the position, rotation and scale of the image has little effect on the amount of calculations. In addition, the contours completely determine the shape of the image, are slightly dependent on color and brightness and contain the necessary information for further classification of the object. This approach makes it possible not to consider the internal points of the image and thereby significantly reduce the amount of information processed by moving from the analysis of the function of two variables to the function of one variable. The consequence of this is the possibility of ensuring the operation of the processing system on a time scale closer to real. In addition to the well-known traditional applications of deconvolution, there are also various exotic applications of deconvolution. One of the most remarkable applications is the restoration of recordings of famous singers voices on old gramophone records by blind deconvolution.
[Saida Safibullayevna Beknazarova. **Mathematical Model Of Geometric Characteristics Of Images Based On the method Deconvolution Procedure.** *N Y Sci J* 2020;13(3):45-57]. ISSN 1554-0200 (print); ISSN 2375-723X (online). http://www.sciencepub.net/newyork. 4. doi:10.7537/marsnys130320.04.

Keywords: building a mask, objects, video analysis, intellectual video surveillance systems, deconvolution procedure.

Introduction

There are several different types of sensitivity. Light sensitivity characterizes the property of the eye to respond to the smallest possible light flux. However, it should be noted here that the probability of recognizing the smallest possible light flux also depends on other factors, such as the angle of view.

The visual system reacts differently to radiation that is equal in power, but radiated from different ranges of the spectrum. This sensitivity is called spectral.

The ability of the eye to distinguish minimal differences in the brightness of adjacent areas of the image is characterized by contrast sensitivity. Also, the visual system is characterized by different sensitivity to color tone, i.e. to radiation from different parts of the spectrum. The visual system is also characterized by sensitivity to color saturation.

The above types of visual system sensitivity are not constant, but depend on many factors, in particular, lighting conditions. For example, when moving from a dark room to a light room, it takes some time to restore the light sensitivity of the eye. This process is called brightness adaptation of the eye.

Color perception is characterized by three main characteristics: lightness, color tone, and saturation. Color spaces are used to classify colors.

Various color vision models are created based on the properties and characteristics of visual systems. Among them, we should highlight the model of color vision proposed by Frey. The peculiarity of this model is that the visual system is represented by three channels, two of which characterize color, and the third-brightness. This model is most successfully consistent with many properties of color vision.

In fact, the transformations of interest to us surpass in complexity all the color transformations considered so far. This is the fact that with any method of cutting the color range used in practice, each color component of the converted pixel depends on everything n color components of the original pixel.

One of the simplest ways to “divide” a color image is to display all the colors that lie outside the region of interest, into some neutral, non-striking color.

If the colors of interest are enclosed in a cube (or hypercube, when $n > 3$) with rib length W and c center at $(a_1, a_2, \dots, a_n)$ color space, which corresponds to a given prototype color, the set of necessary transformation functions is given by

$$S_i = \begin{cases} 0{,}5, \text{if a } |r_j - a_j| > \frac{W}{2} \text{ for anyone } 1 \le j \le n; \\ r_i, \text{in other cases;} \end{cases} \quad i = 1,2,\dots,n$$

These transformations highlight colors around a given one, replacing all the others with the color of the midpoint of the used color space (a randomly chosen neutral color). In the case of the RGB color space, a suitable point is the middle of a segment of gray colors, i.e. the color (0.5, 0.5, 0.5).

If a sphere is used to specify a color region of interest, then formula (11) takes the form

$$S_i = \begin{cases} 0{,}5, \text{if a } \sum_{j=1}^{n} (r_j - a_j)^2 > R_0^2; \\ r_i, \text{in other cases;} \end{cases} \quad i = 1,2,\dots,n$$

Here R_0 — radius of color bounding sphere (or hyperspheres at $n > 3$), a $(a_1, a_2, \dots, a_n)$ — The coordinates of the center of the sphere in the color space, which determine the color of the prototype. In other useful modifications of formulas (1.1-11) and (1.1-12), several prototype colors are used, and the intensities of colors that lie outside the region of interest decrease — instead of replacing these colors with some neutral color.

Color conversions are feasible on most of the personal computers. In conjunction with digital cameras, flatbed scanners and color inkjet printers, they turn a personal computer into a digital photo lab that allows brightness equalization and color correction (two operations that form the basis of any high-quality color reproduction system) without using traditional chemical processing tools used in conventional photo labs. Although the luminance and color corrections are useful in various areas of image processing, our discussion will focus on their most common application - improving photographs and color reproduction.

When developing and improving these transformations, evaluation is carried out using a monitor; therefore it is necessary to ensure a high degree of color correspondence between the monitors used and possible output devices. In fact, the monitor must accurately reproduce the colors of the original image presented in a digital format, as well as the final colors of the image as they appear on the print. This is best achieved by using a device-independent color model that interconnects the color coverage of monitors, output devices, and other devices used. The success of this approach is defined as the quality of the color profiles used to display each of the devices in the color model, as well as the model itself. The role of such a model in many color management systems is the CIE (CIE) model $L^*a^*b^*$, also called the CIELAB model ([CIE, 1978], [Robertson, 1977]). Color coordinates in the model $L^*a^*b^*$ given by the following expressions:

$$L^* = 116 \cdot h\left(\frac{Y}{Y_W}\right) - 16, \quad (1.1\text{-}13)$$

$$a^* = 500\left[h\left(\frac{X}{X_W}\right) - h\left(\frac{Y}{Y_W}\right)\right], \quad (1.1\text{-}14)$$

$$b^* = 200\left[h\left(\frac{Y}{Y_W}\right) - h\left(\frac{Z}{Z_W}\right)\right], \quad (1.1\text{-}15)$$

where

$$h(q) = \begin{cases} \sqrt[3]{q}, \text{if a } q > 0{,}008856; \\ 7{,}787q + {}^{16}/_{116}, \text{if a } q \le 0{,}008856; \end{cases}$$

and magnitudes X_W, Y_W и Z_W represent the coordinates of the reference white. As such, white light is usually used, reflected by an ideal diffuse surface illuminated by the D65 source of the MKO standard (which is shown in the MKO color chart).

Since images contain a large amount of information, questions about how it is presented begin to play an important role. We show that the geometric characteristics we are interested in can be extracted from projections of binary images. Projections are much easier to store and process. We also consider continuous binary images whose characteristic function is zero or one at each point in the image plane. This simplifies the analysis, but when using a computer, the image must be divided into discrete elements.

Characteristic function $b(x, y)$ defined at each point in the image. This image will be called continuous. Later, we will consider discrete binary images obtained by appropriately splitting the image field into elements.

Simple geometric characteristics

Let's assume again that there is only one object in view. If the characteristic function is known $b(x, y)$, then the area of the object is calculated as follows:

$$A = \iint_I b(x, y)\,dx\,dy$$

where integration is performed over the entire image I. If there is more than one object, this formula makes it possible to determine their total area.

Area and position. How do I determine the position of an object in an image? Since an object usually does not consist of a single point, we must clearly define the meaning of the term "position". Usually, the geometric center of an object is chosen as the characteristic point of the object (Fig. 2).

The geometric center is the center of mass of a uniform shape of the same shape. In turn, the center of mass is determined by the point at which you can concentrate the entire mass of an object without changing its first moment relative to any axis. In the two-dimensional case, the first moment relative to the axis is calculated using the formula

$$\int_I b(x,y)dxdy = \iint_I xb(x,y)dxdy,$$

а относительно оси y — по формуле

$$y\int_I b(x,y)dxdy = \iint_I yb(x,y)dxdy$$

where (x,y) - coordinates of the geometric center. The integrals on the left side of the given relations are nothing more than the area *A*, which was discussed above. To find the value *x* и *y*, it must be assumed that the value of *A* not equal to zero. Note in passing that the value of *A* represents the zero-order moment of the function $b(x,y)$.

Today, the Matlab system, in particular the Image Processing Toolbox, is the most powerful tool for modeling and researching image processing methods. It includes a large number of built-in functions that implement the most common image processing methods. Let's look at the main features of the Image Processing Toolbox package.

Geometric transformations of images

The most common functions of geometric transformations include image cropping (imcrop), resizing (imresize), and image rotation (imrotate).

The essence of cropping is that the imcrop function allows you to interactively cut a part of an image using the mouse and place it in a new viewport.

One of the most important practical methods of deconvolution is the method of blind deconvolution. Note that all methods of processing speckle images can be considered as special cases of blind deconvolution.

In addition to the well-known traditional applications of deconvolution, there are also various exotic applications of deconvolution. One of the most remarkable applications is the restoration of recordings of famous singers voices on old gramophone records by blind deconvolution.

The convolution integral is represented by the expression

$$b(x) = f(x) * h(x) \quad (1)$$

where $h(x)$ is the function that sets the distortion; $f(x)$ is the function that needs to be restored.

According to the Fourier convolution theorem, the image of a quantity (1) is equal to

$$B(u) = F(u)H(u) \quad (2)$$

where $F(u)$ is the function associated with the function f (x) by a two - dimensional Fourier transform; $H(u)$ is the Fourier image of the optical transfer function.

The idealized problem of finite deconvolution is as follows: the functions $b(x)$ and $h(x)$ are given, and the function $f(x)$ must be restored, provided that all three quantities have a finite extension.

It follows from the relation (2) that this problem can be solved as follows

The division operation inside curly brackets in expression (3) is called simple inverse filtering. The term "filtering" is used here by analogy with the classical theory of circuits and the modern theory of signal processing. A classic filter is a device that changes the spectrum of time frequencies of a signal. The spectrum $B(u)$ is a function of the spatial frequency.

$$b(x) = f(x) * h(x) \quad (3)$$

The optical transfer function $H(u)$ changes the spatial frequency spectrum $B(u)$ as a result of applying the above division operation.

Since processed images are usually stored in computer memory as quantized values, image processing techniques typically use digital rather than classical analog filters. A digital filter is defined by a discrete array, generally speaking, of complex numbers, which changes the spectrum of spatial frequencies during some processing operation. Therefore, both functions, h (x) in formula (1) and $H(u)$ in formula (2), can be considered as filters (and in most applications they are implemented digitally). The generally accepted classification of digital filters originated in the theory of signal processing as functions of time, and this classification can be used in the theory of processing one-dimensional images, i.e. signals as functions of a (single) spatial variable. We will transfer the corresponding terminology to the two-dimensional case. The concept of "reference" in the theory of signal processing passes into the concept of "image element" in the theory of image processing. Both samples and image elements must be quantized in amplitude before they are digitally processed. The image to which the filtering operation is to be applied is called the specified image, and it is referred to as consisting of the specified image elements. Elements of a filtered image are called output image elements. In the case of a non-recursive digital filter, each output image element is a weighted sum of the specified image elements. In the case of a recursive digital filter, each image output element is a weighted sum of the specified image elements and the previously calculated image output elements. All practically implemented digital filters, of course, are described by arrays of finite sizes (in the one-dimensional case, the final filter is often called a short one). A digital filter is called a direct filter if it is applied in the image plane, and a spectral filter if it is applied in the frequency plane. The causal filter is one-way in the sense that its response always lags behind the input action (this is somewhat artificial in the two-dimensional case, but of course is very important for one-dimensional filtering operations that are the basis of signal processing as functions of time). Causal filters are almost always implemented as direct filters. A multiplicative digital filter is a spectral filter in which each output sample is obtained as the product of a given input signal element by one element of the filter array.

If all the essential aspects of the practical problems of deconvolution were reduced to the formula (3), then the entire content of this publication could easily be contained in a small article. However, there are many practical difficulties in the deconvolution problem. This is because the data being processed is always distorted in practice.

Before setting the practical problem of deconvolution, we examine some properties of convolution consistency.

In the one-dimensional case, the ratio (2) is represented as

$$B(w) = F(w)H(w) \quad (4)$$

where the real variable u is replaced by the complex variable $B(w)$. If the functions $f(x)$ and $h(x)$ have a finite extension, so that the extension of the function $b(x)$ is also finite, then their spectra are characterized by sets of zeros in the complex w-plane. If a given set z_g is represented as a set of real zeros z_{gr} and zeros that can be complex z_{gc}, then we can write

$$z_b = z_f \cup z_k \quad (5)$$

This means that the one-dimensional deconvolution problem is consistent only if all zeros of the function $H(w)$ are also zeros of the function $B(w)$. Therefore, the values $b(x)$ and $h(x)$ cannot be set independently; it must be known beforehand that they satisfy the relation (1). The same applies to two-dimensional convolutions.

Results. Now let's go back to the periodically extended (overlapping) ideal distorted imb (x) image and Its IMB (x) spectrum. The latter can be written as

$$M_b(x) = \sum_l^{\infty} F_l \, H_{l,m} \delta(L_1 u - l)\delta\,(L_2 \gamma - m) \quad (6)$$

where $\delta\,(*)$ is the Delta function; Fi, m are the Fourier coefficients of the true image $f(x)$, which are also counts of the function $F(u)$which are considered in the counting theorem. These samples are taken at raster points (l/L_1, m/L_2) in the frequency plane. The values Hl, m included in expression (6) are samples of the optical transfer function $H(u)$, at the same points in the raster:

$$H_{l,m} = H(\frac{1}{L_1}, \frac{m}{L_2}) \quad (7)$$

where l and m are arbitrary integers.

Now we can set an idealized problem of periodic deconvolution: given the functions $imb(x)$, and $h(x)$, we need to find the function $f(x)$, [knowing that $f(x)$, and $f(x)$, are functions of finite extension, and $imb(x)$ is a periodic function].

For a given function b (x), you can calculate the function B (u) and immediately find that

$$H_{p,l,m} = B(\frac{1}{L_1}, \frac{m}{L_2}) \quad (8)$$

Similarly, the counts of the optical transfer function Hl, m are calculated. The expression (6) shows that each value of F_l, m is given by a division $\frac{B_{p,l,m}}{H_{p,m}}$ operation, which can always be performed if the values of Hl, m are different from zero. This simple approach is adequate in the case of functions $b(x)$ and $f(x)$, which are selected independently, since the function $imb(u)$ in accordance with expression (6) actually exists only at the above points in the raster. But this approach is not acceptable in an idealized problem in the case of finite convolution, since then $B(u)$, is a continuous function of the variable u.

It is therefore surprising that the only consistency condition for periodic convolutions is the requirement that the values $H_{l,m}$ can be zero only for those values l and m for which $H_{l,m}$=0. This condition is called the consistency condition for periodic convolutions. We emphasize that there is no value. $H_{l,m}$ cannot be exactly zero in the real measurement of the function $h(u)$, or, equivalently, the function $H(u)$, so that periodic convolutions are always consistent in practice (they are, of course, very noisy when a large number of values $H_{l,m}$ are "small" at values l and m that correspond to values significantly different from zero $B_{p,l,m}$).

The practical problem of deconvolution is set as follows: the functions $b(x)$, and $h(x)$ are given, you need to find the function $f(x)$, knowing that it is a truncated version of the function of the recorded image $r(x)$.

One of the "Golden rules" in image reconstruction is to avoid processing data that contains any discontinuities, of which clipping and truncation are the most undesirable, since they almost always produce false details (often called artifacts, especially in medical applications). Thus, as a rule, it is desirable to pre-process the image in order to fully compensate for all existing gaps and other removable defects.

Conclusion. Any type of preprocessing can, of course, contribute noise in addition to the image distortion f (x) already present in the recorded image

$r(x)$. But if the gaps are not fixed, then the corresponding artifacts usually prevail over any additional noise introduced by preprocessing. The" aligned " shape of the image is denoted here by $a(x)$. and will be called a pre-processed recorded image. Although all three values must change as a result of preprocessing, there is rarely any way to estimate how much, and therefore it usually makes no sense to talk about the difference between images $a(x)$ and $r(x)$. Next, we will treat these two images as identical, at least in the frame Ω_a (i.e., in the area of the image plane) where the pre-processed version of the meaning image fits $a(x)$. Therefore, we assume that

$$a(x) = f(x) * h(x) + c(x) \quad (9)$$

This assumption does not affect the generality of reasoning, since noise c (x) includes the effects of arbitrary additive distortion associated with preprocessing.

Now we will introduce the concept of "recoverable true image $\hat{f}(x)$ This is an estimate of the image $f(x)$ that can be obtained from the image $h(x)$.

In any rational approach to solving a practical deconvolution problem, the pre-processed image a (x) is first obtained from the specified image $a(x)$. Then, a suitable deconvolution procedure is selected to obtain $\hat{f}(x)$ based on h (x) and a (x). Some of these procedures can be seen as the process of obtaining $\hat{h}(x)$ a modified point propagation function that is associated with a pre-processed recorded image and a recoverable true image ratio

$$a(x) = \hat{f}(x) * h(x) + c(x) \quad (10)$$

It is convenient to denote the Fourier coefficients $\hat{f}(x)$ of a function by, and to denote the spectra $\widehat{F_{l,m}}$ of functions $a(x)$, $c(x)$, and use $\hat{f}(x)$ $\hat{h}(x)$ the corresponding capital letters with or without a" hat".

If there is a concern that the differences between $\hat{f}(x)$ and $f(x)$ will greatly increase due to the lack of consistency between the functions $a(x)$ and $h(x)$, taken explicitly finite, then you can refer to the formula

$$im_b(x) = \sum_{l,m\to\infty}^{\infty} b(x - lL_1, y - mL_2)$$

for a periodic $imb(x)$ image with b replaced by a. Then the imb_u spectrum of the convolution is given by expression (6), but with the replacement of the values $\widehat{F_{l,m}}$ and $\widehat{H_{l,m}}$ the values F_l , m and H_l , m, respectively. Recall that periodic convolutions are not affected by inconsistency, which, as already mentioned, can distort convolutions of quantities that have finite extensions.

The first step to solving these problems is to select the foreground, i.e. it is necessary to form a background frame and build activity masks from the difference between the background and the current frame with its subsequent viewing, in order to select moving objects. Such classes include the detection of a moving air object observed against a clear or cloudy sky.

With constant movement of the camera itself the only source information can only be adjacent frames- they contain overlapping part of the scene and analyzing it, you can get any results. Let's consider the features on the basis of which segmentation methods are built.

1. The luminance characteristic. Most often histogram methods or an algorithm are used to analyze an image based on a brightness characteristic.

watershed's. Histogram methods set a threshold corresponding to the minimum of the bimodal histogram of the brightness distribution in the image or local window. Pixels whose brightness is below the threshold are considered to belong to the background, higher-to the object. This allows you to highlight objects brighter than the background. In case the objects are darker than the background the decisive rule should be the opposite. Histogram methods require a priori knowledge of whether an object is lighter or darker than the background and allow only objects of uniform brightness to be distinguished [5]. The watershed algorithm works with the preparation of the original image obtained using a multiscale morphological gradient. The prerequisite for segmentation is the minimum brightness gradient vector. This eliminates the need for a priori knowledge of the object brighter or darker than the background, but retains the requirement of the same brightness within the entire image of the object for its correct segmentation [5].

Conditions for observing objects of interest outdoors in natural light suggest the presence of shadows and cloud cover, which contributes to the occurrence of significant segmentation errors when using only the brightness characteristic [5];

2. Texture feature. The basis of segmentation methods is to identify areas with different values of quantitative estimates of textures. According to according to the definition of texture given in [6], it is distinguished by the following: there is a local fragment that repeats regularly within an area larger than it; a drawing of a local fragment;

3. Sign of form. The basis of the methods using the shape feature for segmentation of objects of interest is the comparison of the reference description of the object (contour, skeleton or binary mask) with the preparation obtained from real images [7];

4. Sign of movement. It can be estimated based on motion energy and motion vectors. Motion energy is the temporal changes in the brightness of pixels in adjacent frames of a video sequence. Methods and

algorithms based on motion energy are sensitive to the appearance of false motion energy caused by noise and light changes. Their main disadvantage is the inability to segment objects located in close proximity to each other. This is because the motion energy is a scalar estimate that gives information that changes have occurred in the frame region in question; however, unlike motion vectors, it does not provide information about the velocity and direction of motion. The use of optical flow (motion vectors) allows you to separate a moving object from a complex static background and from other objects, if their direction of motion and speed are different. Information about the speed and acceleration of the object, and, in the case of using a parametric model, also about the angles of its rotation, allows to resolve the situation of occlusion, separation of the object into parts, to identify objects of interest after conducting a prediction [1].

The main problem in creating most object recognition methods is determining which pixels to recognize and which to ignore [2]. To solve this problem, it is necessary to consider a compact representation of the original data. The initial data for segmentation and tracking of an air object, in this work, is a halftone video sequence (image sequence) of some scene S, obtained from a mobile video camera. There is no movement in the first n frames of the video sequence; starting with n + 1 frames, objects of interest may appear in front of the camera. For each frame of the sequence it is required to obtain a binary image of the front moving objects. In the resulting mask: 0 is black, which corresponds to the background, and 1 is white, which corresponds to the foreground. When the camera is moving, a part of the scene may drop out of view from time to time, for example, it may be an observed object. The requirement that the first n frames do not contain any movement is due to the fact that just before the shooting of an aerial object, there is a stage of calibration and adjustment of the video camera [3]. Therefore, at the beginning of the sequence there is a set of frames in which there is almost no movement. However, this requirement cannot be fully implemented, because of the so-called dynamic background. As an example of a dynamic background, we can consider a video sequence in which a cloudy sky with moving clouds, etc.

I. Production model based on the fuzzy sets theory

A. Description of linguistic variables

Let —$\{F^f\}_{f=1}^{L}$ the set of frames of the input video sequence length L. (F^f) are either two-dimensional matrices in size $H\times W$, if the video is black and white, or three-dimensional $H\times W\times 3$, if the video is in color. We will consider the processing of black-and-white video (since most CCTV cameras are black-and-white), then (F^f) (y,x) — the value of the pixel at the intersection y- th row and x- the second frame of the video sequence. We define the task of constructing a mask of foreground objects as obtaining a binary matrix (A^f) size $H\times W$ by frames of the input video sequence F^f

$$A^f(y,x)=\begin{cases}1,\text{if the pixel }(y,x)\text{ belongs to a moving object}\\0,\text{if the pixel }(y,x)\text{ belongs to a fixed background.}\end{cases}\quad(11)$$

Note that in terms of "moving object", "stationary background" there is some ambiguity inherent in all problems relating to the semantic meaning of real world objects [4]. So, an object that has stopped for a while should be considered an object ($A\,(y,x)=1$), however, standing for a long time should be considered a background $A\,(y,x)=0$). However, such ambiguity is not the key problem of this task. Moreover, in one of the most cited works on background models, listing the fundamental principles of their construction, it is argued that the problem should be solved without taking into account semantics [7].

Let's set the mask of objects as binarization of the absolute difference of the background model (B^f)) and the current frame (F^f)) on the threshold (T^f))

$$A^f(y,x)=\left|F^f(y,x)-B^f(y,x)\right|>T^f(y,x)\quad,\quad(12)$$

In this case, the main difficulty of the task is shifted to the exact construction of the background model (B^f). In the above statement (B^f)— it is a matrix of the same size as (F^f). In simplest case T= ($T^f(y,x)$), however, in a number of methods, including those often used, the binarization threshold T may depend on the position of the pixel (x,y) (for example, a method based on a single Gaussian [2]) or from a frame number. In some methods, for example, based on optical flow [4], the background model (B^f) it may not be explicitly evaluated.

In the vast majority of works, the quality of methods for constructing a background model or mask of objects is estimated by the accuracy (precision), equal to the ratio of the number of correctly marked pixels to the number of marked pixels, and by the completeness (recall), equal to the ratio of correctly marked pixels to the number of pixels related to objects in the standard [13, 14, 15]:

$$\text{Precision}=\frac{|\{(y,x):A^f(y,x)=1\wedge G\,(y,x)=1\}|}{\{(y,x):A^f(y,x)=1\}},\quad(13)$$

$$\text{Recall}=\frac{|\{(y,x):A^f(y,x)=1\wedge G\,(y,x)=1\}|}{\{(y,x):G^f(y,x)=1\}},\quad(14)$$

When objects are detected, video sequence frames are input $\{F^f\}_{f=1}^{L}$. Let $P^{(f)}$ — the set of positions of objects of the required class on *f- th* frame:

$P^{(f)}=\left\{P_i^{(f)}\right\}_{i=1}^{C(f)}$ (15), where $P_i^{(f)}=(y,x)$ position *i-* th object *f –th* frame, $y \in \{1,\dots,H\}, x \in \{1,\dots,W\}, C(f)$- number of objects per *f –th* frame. The task of obtaining the expected positions of objects (goals or hypotheses) $\widehat{P^{(f)}}$. Let's consider the object $P_i^{(f)}$ detected if there is at least one target $\hat{P}_j^{(f)}$ at a distance of no more *R* pixels. Note that the above detection condition is not the only possible one. The quality of detection will be measured with the help of accuracy and completeness, but with some changes due to the specifics of the task of the video surveillance system. Namely, the so-called multi objects (multiple objects) and multi goals (multiple trackers) will not be penalized, i.e. when one object corresponds to several goals or one goal corresponds to several objects (Fig. 1). Such situations are not critical in the context of identifying any events, because it is not important for us to know the exact number of objects, it is important to note the locations of one or more objects, and not to mark them where there is no object. For objects, duplication of targets or objects does not lead to system errors, but when a violation is detected, it is important not to write out more than one event for objects. In this case, you can filter targets by limiting the distance between them from below.

Let $N_i^{(f)}$ a set of targets issued by the detector, at a distance of no more than *N* pixels from the object $P_i^{(f)}$ *(neighborhood of the object):*

$$N_i^{(f)} = \left\{\hat{P}_j^{(f)} \middle| distance\ (\hat{P}_j^{(f)}, P_i^{(f)}) \le R\right\}, \quad (15)$$

and $\hat{N}_j^{(f)}$- a set of objects at a distance of no more than *R* from detection $\hat{P}_j^{(f)}$ *(target neighborhood).*

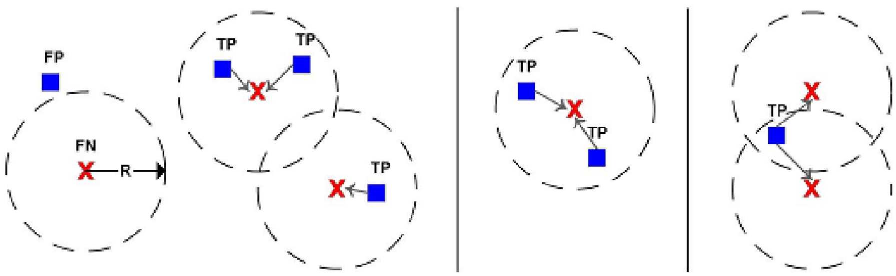

Figure 1: Left: example of matching targets (squares) and objects (crosses). In the center: an example of multipurpose. Right: an example of multi-object and.

II. Testing the production model

Let's define the number of true *TP* (true positives), false *FP* (false positives) detections, and *FN* (false negatives) object misses, as well as the accuracy of the Precision and the completeness of Recall as follows:

$$TP = \left|\left\{\hat{P}_j^{(f)} \middle| \hat{N}_j^{(f)} \ne \emptyset\right\}\right|, \quad (7)$$

$$FP = \left|\left\{\hat{P}_j^{(f)} \middle| \hat{N}_j^{(f)} = \emptyset\right\}\right|, \quad (8)$$

$$FN = \left|\left\{\hat{P}_j^{(f)} \middle| \hat{N}_i^{(f)} = \emptyset\right\}\right|, \quad (9)$$

$$Precision=\frac{TP}{TP+FP}, Recall=\frac{TP}{TP+FN} \quad (10)$$

In addition to estimating the coverage of objects by targets the average distance from objects to their respective targets is also calculated:

$$D_i^{(f)} = \frac{\sum_{j:\hat{P}_j^{(f)} \in N_i^{(f)}} Distance(\hat{P}_j^{(f)}, P_i^{(f)})}{\left|N_i^{(f)}\right|}, \quad (16)$$

$$D^f = \frac{\sum_{i=1}^{cf} D_i^{(f)}}{cf}. \quad (17)$$

The ultimate goal of tracking, as well as the task of object detection, is to find the positions of objects $\left\{P_i^{(f)}\right\}_{j=1}^{cf}$ in the frames of the video sequence $\left\{F^{(f)}\right\}_{f=1}^{L}$, but the tracker, unlike the detector, receives information about the positions of objects in the previous frame, i.e. continues the trajectory of detected objects. Therefore, the result of the tracker is measured not only by coverage metrics (accuracy and completeness), but also by the metric of consistency of goals on different frames: the goal, led by the tracker, must correspond to the same object on different frames: $\hat{E}_{\hat{Q}_j^{(f)}}^{(f-1)} = \hat{E}_j^{(f)}$, (13) where $\hat{E}_j^{(f)}$ the identifier of the corresponding object *j- th* purpose of trakeen *f*-th frame, $\hat{Q}_j^{(f)}$- the target number in the previous frame (i.e., the tracker has updated the position purposes with $\hat{P}_j^{(f-1)}$ (pre frame) on

$\hat{P}_j^{(f)}$ (current frame)), if the target existed in the previous frame, otherwise $\hat{Q}_j^{(f)} = 0$. The ratio of the number of failures of this condition in the course of the tracker to the total number of goals on all frames will be used as a metric of inter-frame consistency of the goals of the tracker, which is called the number of inconsistencies (mismatch errors), *ME*.

$$ME = \frac{\sum_{f=2}^{L} \sum_{j=1}^{\hat{C}f} \left[\hat{Q}_j^{(f)} > 0 \Delta \hat{E}_{\hat{Q}_j^{(f)}}^{(f-1)} \neq \hat{E}_j^{(f)} \right]}{\sum_{f=2}^{L} \hat{C}f}, \qquad (18)$$

For example, for the problem of detecting violations at a pedestrian crossing, the condition (13) is weak, since we are not interested in the exact support of the identifiers of observed objects, but for determining violations it is important to know the direction of movement of objects. Therefore (13) is weakened to the following condition:

$$Dir\left(\hat{E}_{\hat{Q}_j^{(f)}}^{(f)}\right) = Dir\left(\hat{E}_j^{(f)}\right), \qquad (19)$$

where Dir (e) $\in \{-1; +1\}$ the direction of movement of the object with the identifier (either from left to right or in the opposite direction). For simplicity, we assume that pedestrians do not change direction during the crossing of the crossing. In view of the fact that cars overlap each other only partially and the car closest to the transition is always important, the inter-frame inconsistency of the goals of car trackers is much less common, so it does not require special measurement.

Cutting a specific range of colors in an image used to select some objects from their environment. The basic idea is to either (1) reproduce the colors of interest so that they appear on a general background, or (2) use the color-defined areas as a mask for further processing. The simplest approach is to generalize the method of cutting the luminance range. However, since the color pixel value is n-dimensional vector, the conversion functions for cutting the color range are more complex than the similar functions for cutting the luminance range. In fact, the transformations of interest to us surpass in complexity all the color transformations considered so far. This is the fact that with any method of cutting the color range used in practice, each color component of the converted pixel depends on everything n color components of the original pixel.

One of the simplest ways to "divide" a color image is to display all the colors that lie outside the region of interest, into some neutral, non-striking color. If the colors of interest are enclosed in a cube (or hypercube, when $n > 3$) with rib length W and c center at $(a_1, a_2, \ldots, a_n)$ color space, which corresponds to a given prototype color, the set of necessary transformation functions is given by

$$S_i = \begin{cases} 0{,}5, \text{if a } \left|r_j - a_j\right| > \frac{W}{2} \text{ for anyone } 1 \leq j \leq n; \\ r_i, \text{in other cases;} \end{cases} \quad i = 1,2,\ldots,n \qquad (20)$$

These transformations highlight colors around a given one, replacing all the others with the color of the midpoint of the used color space (a randomly chosen neutral color). In the case of the RGB color space, a suitable point is the middle of a segment of gray colors, i.e. the color (0.5, 0.5, 0.5).

If a sphere is used to specify a color region of interest, then formula (11) takes the form

$$S_i = \begin{cases} 0{,}5, \text{if a } \sum_{j=1}^{n} \left(r_j - a_j\right)^2 > R_0^2; \\ r_i, \text{in other cases;} \end{cases} \quad i = 1,2,\ldots,n$$

Here R_0 — radius of color bounding sphere (or hyperspheres at $n > 3$), $(a_1, a_2, \ldots, a_n)$ — The coordinates of the center of the sphere in the color space, which determine the color of the prototype. In other useful modifications of formulas (11) and (12), several prototype colors are used, and the intensities of colors that lie outside the region of interest decrease — instead of replacing these colors with some neutral color.

Color conversions are feasible on most of the personal computers. In conjunction with digital cameras, flatbed scanners and color inkjet printers, they turn a personal computer into a digital photo lab that allows brightness equalization and color correction (two operations that form the basis of any high-quality color reproduction system) without using traditional chemical processing tools used in conventional photo labs. Although the luminance and color corrections are useful in various areas of image processing, our discussion will focus on their most common application - improving photographs and color reproduction.

When developing and improving these transformations, evaluation is carried out using a monitor; therefore it is necessary to ensure a high degree of color correspondence between the monitors used and possible output devices. In fact, the monitor must accurately reproduce the colors of the original image presented in a digital format, as well as the final colors of the image as they appear on the print. This is best achieved by using a device-independent color model that interconnects the color coverage of monitors, output devices, and other devices used. The success of this approach is defined as the quality of the color profiles used to display each of the devices in the color model, as well as the model itself. The role of such a model in many color management systems is the CIE (CIE) model $L^*a^*b^*$, also called the CIELAB model ([CIE, 1978], [Robertson, 1977]). Color coordinates in the model $L^*a^*b^*$ given by the following expressions:

$$L^* = 116 \cdot h\left(\frac{Y}{Y_W}\right) - 16, \quad (18)$$

$$a^* = 500\left[h\left(\frac{X}{X_W}\right) - h\left(\frac{Y}{Y_W}\right)\right], \quad (19)$$

$$b^* = 200\left[h\left(\frac{Y}{Y_W}\right) - h\left(\frac{Z}{Z_W}\right)\right], \quad (20)$$

where

$$h(q) = \begin{cases} \sqrt[3]{q}, \text{if a } q > 0{,}008856; \\ 7{,}787q + {}^{16}/_{116}, \text{if a } q \le 0{,}008856; \end{cases}$$

and magnitudes $X_{W,}$ Y_W и Z_W represent the coordinates of the reference white. As such, white light is usually used, reflected by an ideal diffuse surface illuminated by the D65 source of the MKO standard (which is shown in the MKO color chart).

Match the chromaticity coordinates $x = 0{,}3127$ and $y = 0{,}3290$). Color space $L^*a^*b^*$ is colorimetric (i.e. equally perceived colors have the same color coordinates), equally contrasting (i.e. equal changes in chromaticity coordinates correspond to equal changes in color perception — see the classic work of Mac-Adam [16]) and independent of devices. Although this color space cannot be displayed directly (a transition to another color space is necessary for display on the screen or when printing), its color scope includes the entire visible spectrum and allows you to accurately represent the colors of any monitors, printers and other input-output devices. Like the HSI system, the system $L^*a^*b^*$ perfectly divides the intensity (which is represented by the brightness L^*) and chromaticity (which is represented by two colors: a^* — red minus green and b^* — green minus blue). This property makes the system $L^*a^*b^*$ very convenient for improving images (tonal and color correction), and for their compression [6-16].

The main advantage of calibrated image processing systems is that they allow performing brightness and color corrections in interactive mode independently, i.e. in the form of two successive operations. Initially, the image's brightness range usually corrected, and then the color imbalance, such as insufficient or excessive color saturation, eliminated. The luminance range of a color image related to the overall intensity distribution of its colors. As in the monochrome case, it is often desirable to distribute the intensity values of a color image evenly between the lightest and darkest values. The following examples discuss the various color conversions used for luminance and color correction.

In fig. Figure 2 shows typical transformations used to correct the three main types of luminance imbalance, when the image has low contrast, is too light or too dark. S- the shaped curve in the top row in this figure is ideal for enhancing contrast. The center point of this curve positioned so that the values in the light and dark areas become correspondingly brighter and darker. (The inverse curve with respect to the considered one can used to reduce contrast.)

transformations for each of the three (or four) color components; in the HSI color space, only the intensity component is converted.

Although in reality the transformation functions, like the color components, are discrete, it is customary to depict them and work with them as with continuous functions; moreover, they are usually composed of piecewise linear functions or (for smoother mappings) of higher order polynomials. Note that the type of brightness imbalance for the images in Fig. 2 is visible directly, but it can also be determined from the histograms of the values of the color components of the image.

After the brightness properties of the image correctly adjusted, you can begin to correct the color imbalance. Although objectively, the presence of color imbalance can established by analyzing an image of a certain area of a known color using a colorimeter, but in the case when the image has white areas where the RGB or CMY (K) component values should be equal to each other A fairly accurate visual assessment is also possible. As can be seen in fig. 2, skin tones are also an excellent object for visual assessment of color, since human vision is extremely sensitive to skin color. Bright colors, such as bright red, are not very useful when it comes to visually assessing the quality of color reproduction.

Trainer basic file serving to preserve digitally converted analog data (i.e., what people see and hear in real life). Typically, such a stored text, graphics, audio and video information occupies a large volume, so it is compressed using a variety of audio and video codecs. All utilities for working with these files installed with the operating system.

The Code abbreviation for Coder / Decoder - program that allows you to convert the recorded information so that it takes up less space. In this case, the file extension can't be changed, i.e., the basic structure of the container does not change, change the representation of text, graphics, audio and video, but to play the file, "encrypted" with any codec, it is necessary for it to be installed on the user's computer [1].

TIAV- multimedia system - a class of objects consisting of textual, illustrative, video and audio information streams that form the basis for the development of multimedia content.

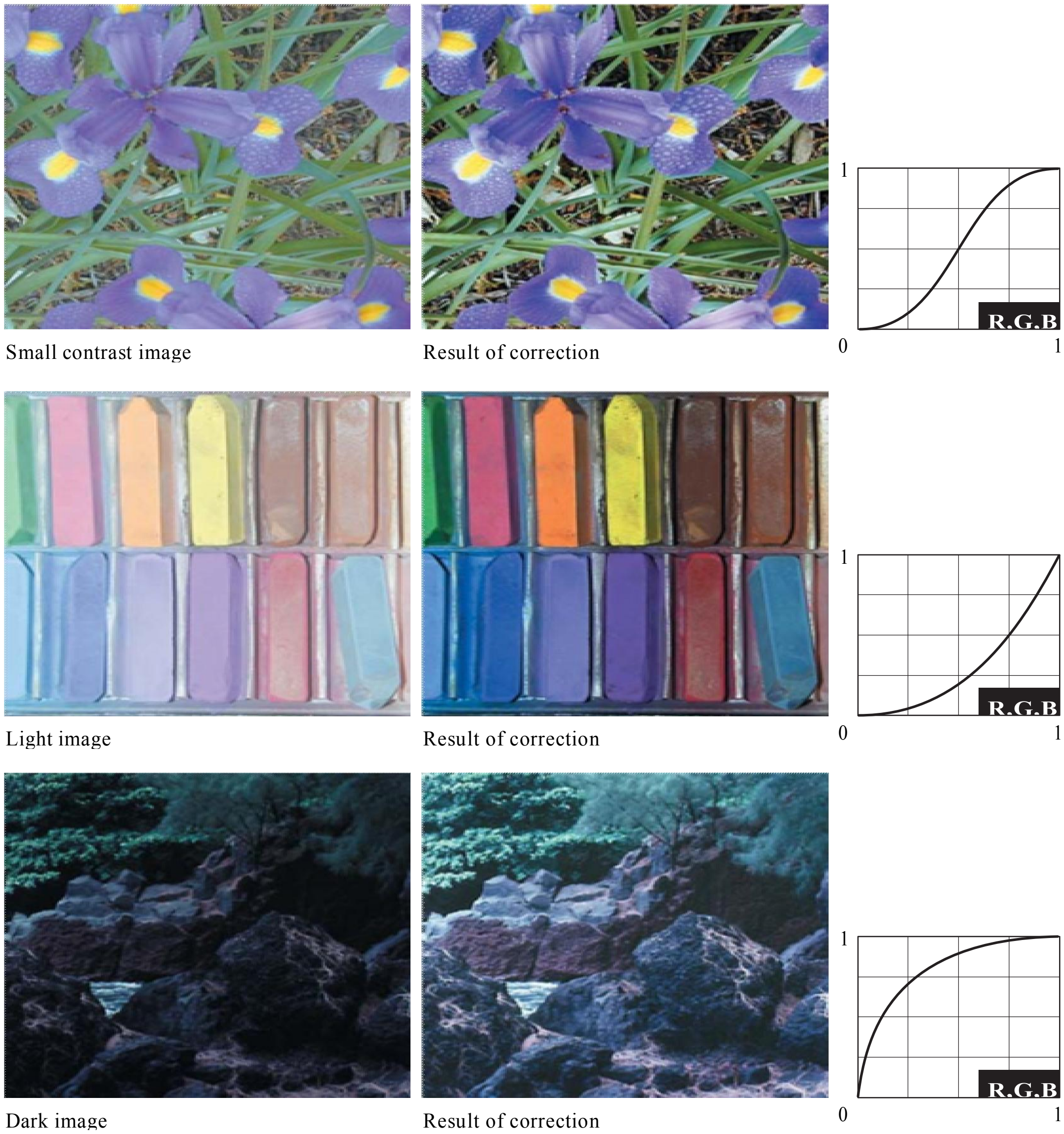

Figure 2. Luminance correction for low-contrast, light and dark color images. With the same change of the red, green and blue components, the color tone of the image does not always noticeably change.

Synthesis algorithm software modules formed task-oriented management system TIAV- discrete-continuous processes multimedia system based on the concept of the formation of individual learning paths multi contents that takes into account the described features and apply the developed models and methods to solve the problem of class formation multi contents for individual training is given in Figure 1.

The modern development of the information society, advancing the information flow of the material is aimed at removing bottlenecks in a multimedia course. Leading information flow in the opposite direction contains, as a rule, the initial information; outpacing the information flow in the forward direction - these are preliminary reports of the upcoming multimedia process [3].

Accompanying when simultaneously with the material flow is information about the quantity and quality of material flow allows you to quickly and correctly identify the material values and send them to their destination.

In today's digital world there are many different ways of presenting multimedia information. Of course, in order to convert analog information into digital form, special programs are needed, create a file (it will be called a container), which contains all the text, graphics, audio and video the information.

Container TIAV (text, image, audio, video), Figure 4 - a project aimed at creating an open flexible cross-platform (including hardware platform) standard of multimedia container formats and a set of tools and libraries for working with data in this format. This project is a development project MCF, but differs significantly in that based on EBML (Extensible Binary Meta Language - Extensible Binary meta language) - binary counterparts language XML. Using EBML allows you to expand the format without losing compatibility with older programs.

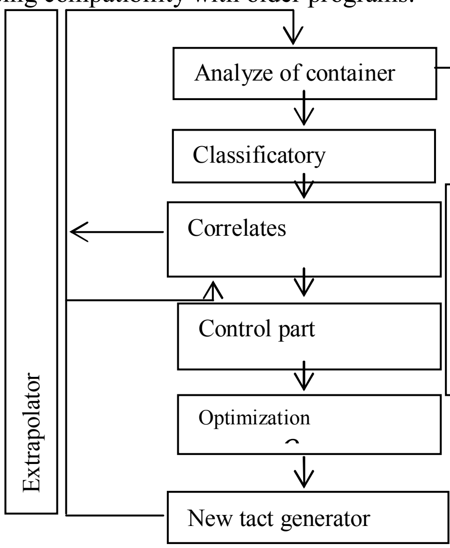

Figure 3. Algorithm synthesis software modules formed task-oriented management system TIAV-discrete-continuous processes of multimedia system

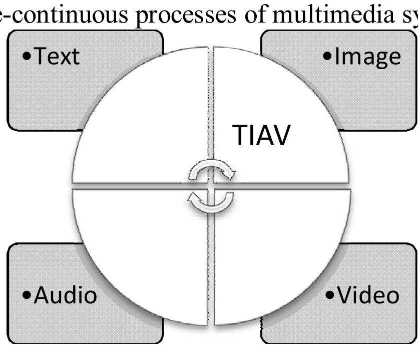

Figure 4. Scheme container TIAV

3. Realization of the concept

TIAV, not the format in the truest sense of the word. A container, a kind of packaging, which can wrap text, image, video and audio content. Container TIAV designed with all modern features and is aimed at future trends in the field of play. It is built on the principle of EBML, which is analogous to XML only for binary data. Scope TIAV very versatile. In the package can be consist the big volume audio, video sequences, subtitles, chapters, posters, fonts, descriptions, comments, photo albums and more. This container is compatible with all modern requirements video container.

Possible formats that are run inside TIAV: broadcast over the Internet (HTTP and RTP); fast file; robustness; screen menus (both on DVD); splitting a file into chapters (Chapters); toggle subtitles on the fly; switchable audio tracks; modular expandability;

It should be noted that the draft text / image / audio / video content does not include a video compression formats and codecs (such as MP3 or JPEG). This pack, which may contain a large number of streams text, image, audio, video, allowing a user to store a single file and view the information using resource TIAV multimedia system.

TIAV container is an open project (open standard). This means that for personal use it is absolutely free, and the technical specification of the format of the bit stream is available to anyone, even to companies wishing to build support into their format. The source code of all libraries created by a group of developers of the project TIAV distributed under the LGPL (Library to play, written in C using integer arithmetic, also distributed under the terms of the license BSD) [1,2].

In TIAV there is support for the adaptation and implementation of libraries for TIAV Open Be OS MEDIAKIT and GSTREAMER (Eng.) (Multimedia environment running GNU / Linux, similar to Microsoft Direct Show for Windows) and a set of DirectShow filters for playback and creation of files in Windows TIAV by TIAV system [1].

Analysis of the quality management multimedia media educational process shows that all solved with the practical problems are multicriteria, ie to select the best alternative to the weighting of all reasonable alternatives, a quality criterion for adequate assessment of their comparison is not enough. At the same time, unfortunately, for the problem of multi-criteria comparison of alternatives are virtually no effective methods of choice. The main difficulty of solving this problem lies in the fact that the results of the choice of the most efficient alternative based on certain criteria do not match. In addition, among the various selection criteria is usually observed contradiction associated with opposite changes their values at the same changes in the control parameters. In this connection, for determining the quality and effectiveness of media educational process is necessary in the formation of a heuristic selection technique, which allows to circumvent these difficulties.

Analysis of the various factors affecting the quality of the container TIAV, showed that the main of them are: providing a container TIAV certain components: media text, images, graphics, audio, video (animation), or if you need a simulation model.

Thus, the rate of assessment quality assurance (K) container TIAV in general terms can be defined according to the following expression:

$$K = F(M_m, M_i, M_{a,v},, M_{an,im}),$$

where M_m, M_i, $M_{a,v}$,, $M_{an,im}$ - key factors to ensure the quality of the container, respectively media - (text, image is a graphic; AV; animation- simulation model).

An analytical representation of this index can be obtained, for example, on the basis of statistical data by regression analysis. In this case, it will have the following form:

$K = a_1 M_m + a_2 M_i, + a_3 M_{a,v}, + a_4 M_{an,im} + a_0,$

where a_i, i = 1, n-linear regression coefficients determined by statistical processing;

a_0- free term, taking into account other factors that affect the provision of quality container.

However, the construction of such a regression model is difficult because of the difficulty of obtaining necessary for this statistic, because the impact of these factors on the quality of the container TIAV is mostly subjective.

Therefore, to assess the level of quality assurance (K *) multimedia process in high school, you can use the following heuristic expression:

$$K^* = \frac{b_1 M_t^* + b_2 M_i^* + b_3 M_{a,v}^* + b_4 M_{an,im}^*}{\sum_{j=1}^{4} b_j K^*},$$

where K*- Means a quantitative expression of the factor K;

b_j - boost factor j-th factor to ensure the quality of the container TIAV, defined by experts.

Thus, to assess the level of quality container TIAV all included in the model of the factors necessary to present a quantitative manner. Use one of the methods to quantify these factors with the following corresponding coefficients.

Presentation of information in visual form (images, animation, video), as well as symbolic (text, charts, tables, diagrams) in a container TIAV help to better understand the material provided, according to the respondents, the choice of the most effective, successful perceived multimedia products (media texts, audio, video, animation), the representation of media information in the form of a media text were equally, more than 2/3 of the respondents appreciated the use of images in the feed of media course, highlighted the value and ease of perception media content saturated videos; animations, audio accompaniment. By analyzing the diagram it can be concluded that it is advisable to use the design of media courses, media text, image, video and animation based on two criteria, first the full perception of the information, and secondly, in accordance with the costs for their production, based on the chart data it can be seen that the level of the full perception of the new media course saturated already using video and animation, using the same simulation models is costly, but ineffective, as is clearly happening glut of supply of new material of media course, so it is advisable to take into account the point of optimum use in the construction of container media material of TIAV [16,17].

Conclusion

The main advantages of container TIAV include:

- Keeping a simultaneous text and graphic information;
- The content of the audio tracks in different languages;
- The ability to extract audio and video data without special editors;
- The container is easily edited;
- No hang when playing large files;
- Enhanced storage overhead;
- Broadcasting over the Internet (for those interested: the protocols HTTP, RTP);
- Robustness (According to the manufacturer);
- Modular expandability;
- The packaging does not use compression protocol, the files in the container TIAV can store a large number of information flows.

Scope of multimedia information systems expands with each passing day. Today, they cover a wide range of applications including classrooms, conference rooms, comprehensive monitoring systems, situational centers and control centers, etc. Development of multimedia systems is characterized by the increasing complexity of their architecture. All processes in multimedia systems are discrete-continuous, as a result, there is a need to develop models of efficient algorithms, software package such automated multimedia systems, which would include the development of online designers, create multimedia courseware, training multi contents that based on the use of discrete-continuous object class TIAV.

The dimension of the binary image is usually the same as the dimension of the original image. A single value at a point in a binary image corresponds to a decision about the presence of an object at that point [16]. A value of zero corresponds to a decision about the absence of an object. In practice, in the process of selecting objects, any algorithm will make mistakes, which are manifested in the fact that at some points belonging to objects, a decision will be made about their belonging to the background and, conversely, at some points of the background, a decision will be made about the presence of an object. They can be eliminated by applying binary image and segment list processing procedures. Therefore, it is necessary to introduce additional processing steps in order to remove the "false" segments and restore the lost ones. At the end of processing of each frame the consumer is given a list of such the control. Device is necessary to minimize the mismatch between the actual direction of the optical axis of the sensor and the desired one; in most cases, it coincides with the direction of the object of interest. The essence of the task of maintenance is the combination of these directions in order to maintain constant visual contact with the selected object.

Acknowledgements:

Foundation item: The National Project of India. Authors are grateful to the Department of Science and Technology, Government of India for financial support to carry out this work.

Corresponding Author:
Dr. Beknazarova Saida Safibullayevna
Doctor Of Technical Science, Professor Audiovisual Technologies Of Tashkent University Of Information Technologies Named after Muhammad Al-Khwarizmi, Tashkent, Uzbekistan, 100096
Telephone: 998-90-3276666
E-mail: saida.beknazarova@gmail.com

References

1. J. P. Tardif, Non-iterative approach for fast and accurate vanishing point detection//ICCV. 2009. P. 1250-1257.
2. (http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5459328) Pratt W. Digital image processing.- M. Mir, 1982-kN 2 -480 s.
3. Obukhova, N. A. video detection And tracking systems for mobile objects / N. A. Obukhova, B. S. Timofeev / / telecommunications. - 2003-No. 12-Pp. 36-44.
4. Gaganov V., Konushin A. Segmentation of moving objects in the video stream. Computer graphics and multimedia. Issue #3. 2004. Pp. 45-47.
5. Talantbek K. Detection and determination of coordinates of moving objects point objects in the sequence of images / / Novosibirsk 2009.
6. D. Prewer, Kitchen L. Weighted Linked Pyramids and Soft Segmentation of Colour Images // Taiwan: ACCV2000. vol. 2. 2000. P. 989-994.
7. Aridgides A., Fernandez M., Randolph D., Bray D. Adaptive three-dimensional spatial-temporal filtering techniques for infrared clutter suppression. Proc. of SPIE Vol. 1305, Signal and Data Processing of Small Targets. -Oct, 1990, pp. 63-74.
8. Aridgides, M. Fernandez, D. Randolph 3 Ferris D. Adaptive 4-D sh clutter suppression filtering technique // Proc. of SPIE Vol. 1481, Signal and Data Processing of Small Targets. - Aug 1991, pp. 110-116.
9. D. Farin, Peter H. N. de With, W. Effelsberg, Robust background estimation for complex video sequences // International Conference on Image Processing. - Sept. 2003, pp. 145-148.
10. Alpatov BA. Algorithm of detection and selection of a moving image fragment / / communication equipment. Television Technique series. -1991.- No. 2. Pp. 72-76.
11. Beknazarova S.S. Discrete-continuous processes in TIAV multimedia system, LAP LAMBERT Academic Publishing GmbH & Co. KG, Saarbrucken, Germany, 2015, 57.
12. Loops J.R., Loops Yu, Cool Edit Pro 2. Secrets Excellence. BHV - St. Petersburg, 2004.
13. Greeks A.S. Digital video. Tutorial on the computer. Peak - Russia; 2004.
14. Means of automated information systems and technologies [electronic resource] // Finam: [site]. [1999-2012]. URL: http://www.finam.ru/dictionary/wordf02B55/default.asp?n=27 (date accessed: 29.11.2012).
15. Fine V.S. Image Recognition. M.: Nauka, 1970.299 s. Samal D.I., Starovoitov V.V. Selection of features for recognition based on statistical data // Digital image processing. - Minsk: ITK, 1999. – S.105-114.
16. Sedova, N., Sedov, V., Bazhenov, R., Karavka, A., Beknazarova, S. Automated Stationary Obstacle Avoidance When Navigating a Marine Craft (Conference Paper) SIBIRCON 2019 - International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings October 2019, 8958145, Pages 491-495 2019 International Multi-Conference on Engineering, Computer and Information Sciences, SIBIRCON 2019; Novosibirsk; Russian Federation; 21 October 2019 do 27 October 2019; Nomer kategoriiCFP1911E-ART; Kod 156894.
17. D. Luchaninov, R. Bazhenov, T. Gorbunova, S. Beknazarova, L. Putkina, A. Vasilenko. Using Social Network Communities As A Tool For Organizing It Education Inted 2019 Proceedings Pages: 4104-4109 Publication year: 2019 ISBN: 978-84-09-08619-1ISSN: 2340-1079 doi: 10.21125/inted.2019.1029Conference name: 13th International Technology, Education and Development Conference Dates: 11-13 March, 2019 Location: Valencia, Spain.

3/7/2020