Recognition of Hand Gestures on the Video Stream Based on a Statistical Algorithm with Pre-treatment

Palochkin Vitaly, Maksimovskiy Alexander
P.G.Demidov Yaroslavl State University
Yaroslavl, Russia
palochkin1991@mail.ru, maxal9999@mail.ru

Priorov Andrey
P.G.Demidov Yaroslavl State University
Yaroslavl, Russia
andcat@yandex.ru

***Abstract*–The goal of this work is human hand detection and gesture recognition. This is a tremendously difficult task as hands can be very varied in shape and viewpoint, they can de open or closed, they can have different finger articulations. It is proposed a combined method of hand gesture recognition based on a statistical image processing algorithm. As a pre-processing algorithm it was applied Lucas–Kanade method and background subtraction algorithm. Object recognition was performed with using of Haar classifiers. The possibility of using gestures for remote control of various devices with different hands position from the camera location was shown.**

I. Introduction

Object recognition is a classical problem of computer vision, image processing and machine learning. This is important for solving the problems of detection, assessment of the objects position, digital image processing and for remote systems management.

One of the main problems of object recognition is to select function according to which the accessory of image to the object of interest will be set by the control value. Selection of the decision function requiresa minimum of recognition errors.

Gesture recognition is one of the most difficult and important trends in the field of objects recognition. Gesture recognition systems are designed to identify specific human gestures for remote control of a variety of devices and for transmission them the information. In this paper the problem of gesture recognition on the video stream in real time is considered.

The goal isto develop an algorithm for recognizing hand gestures on the video stream in real time. This is achieved by combining the algorithms of texture allocation, color allocation and other machine learning algorithms.

II. Proposed approach

To achieve this goal it is necessary to solve the following tasks:

A. Preliminary processing

On the received digital image from the camera the areas of interest in which palm of the hand can be located are searched using the object detection algorithms, color and texture allocation algorithms and search algorithm for point features. Some of them may be false and not relevant to the image of a hand.

B. Statistical algorithm

Hand gestures are searched on the basis of image pre-processing. In this case Viola-Jones method with Haar features and AdaBoost-classifier is used.

C. Gesture classification

It is necessary to develop a specific decision rule or a set of decision rules, based on which the decision will be made about the gesture type.

III. Preliminary processing

Pattern recognition problem is closely connected with the problem of preliminary classification.

During pre-processing color allocation algorithm, texture allocation algorithm and Lucas–Kanade method were used.

For the object detection the pixel coordinates are determined both by the spatial coordinates and by the coordinates in RGB. On each frame of video stream objects, which pixels fall within ε - neighborhood of the selected pixel in RGB color system, are segmented [1]. On Fig.1 the results of color detection algorithm are illustrated for test object.

Fig. 1. The results of color detection algorithm

Despite the widespread use of textures in image processing and their importance, there is still no rigorous approach to defining and describing them. Thus, methods of distinguishing textures are developed separately for each case.

Texture is "the spatial organization of the elements within a certain section of the image." This is due to the statistical distribution of the intensity of gray tones or different colors shades. The plot can be considered texture, if the number of observed differences of intensity or color changes is sufficiently large[8].

Texture can be described by some attributes. Under the textures attributes characteristic properties are considered. They are common to all textures of a certain class. Texture attributes play a crucial role for their classification and the separation of images into separate areas.

At the beginning it is necessary to determine the size of the sliding window, through which the object of interest will be allocated. Selecting the window size is due to the fact that the image outline is determined by the neighborhood of the image. The size of the sliding window depends on what features characterize the properties of the objects and their statistical characteristics.

As such attributes you can use the statistical moments of the spatial distributions calculated as a measure of homogeneity in a one-dimensional histogram of signal values (characteristics of the first order) and two-dimensional histograms of signal values (characteristics of the second order). The following statistical data are used for numerical evaluation of the texture in a one-dimensional histogram:

- k –initial moment
  $T_1^k = n^{-2}\sum_{i=1}^{n}\sum_{j=1}^{n}[f(i,j)]^k$
- entropy:
  $T_2 = -\sum_{g=0}^{N-1} F(g) log_{10} F(g)$
- energy:
  $T_3 = \sum_{g=0}^{N-1}[F(g)]^2$
- variation:
  $T_4 = \sum_{g=0}^{N-1}(g-\mu)^2 F(g)$

where n is the size of sliding window ($n = 2W + 1$) in pixels;

$f(i, j)$ – the pixel brightnessof sliding window at point (i, j);

N –gradation number of image brightness;

$F(g)$– number of pixels with brightness g;

μ – window average.

Rating textural features calculated on a one-dimensional histogram of frequencies does not consider the mutual arrangement of neighboring pixels in a sliding window and allows you to evaluate the properties of a group of pixels.

There is an approach for the formation of textural features that take into account the mutual arrangement of the pixels within the sliding window. It is based on the use of adjacency matrix.

The analyzed image is considered to be rectangular and it has N_x horizontal elements and N_y vertical elements. in this $G = \{1,2,\dots,N\}$ is a set of N quantized luminance values. Thus the image is described by the luminance values from the set G, that is $f: L_x \times y \to G$, where $L_x = \{1,2,\dots,N_x\}$ and $L_y = \{1,2,\dots,N_y\}$ are horizontal and vertical spatial areas. The set of N_x and N_y is the set of bins in the bitmap. Adjacency matrix contains the relative frequencies p_{ij} on the images of neighborhood elements, which are located at a distance d from each other with luminance $i, j \in G$.

By calculating the adjacency matrices it is possible to calculate directly the numerical estimates of a number of textural features:

- average:
  $T_5 = \mu_i = \mu_j = \sum_{i=0}^{N-1}\left[i\sum_{j=0}^{N-1}P(i,j)\right]$
- energy:
  $T_6 = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}[P(i,j)]^2$
- variation:
  $T_7 = \sigma_i^2 = \sum_{i=0}^{N-1}\left[(i-\mu_2)^2\sum_{j=0}^{N-1}P(i,j)\right]$
- uniformity:
  $T_8 = \sum_{i=0}^{n}\sum_{j=0}^{n}P(i/j)\Big/(1+|i-j|)$

where $P(i, j)$ is the frequency of occurrence of two pixels in the sliding window with the brightness (i, j) under an angle α at a distance d;

σ_i – standard deviation of brightness in a sliding window.

Autocorrelation function:

$$A(\xi,\eta,j,k) = \frac{\sum_{m=j-W}^{j+W}\sum_{n=k-W}^{k+W} f(m,n)f(m-\xi,n-\eta)}{\sum_{m=j-W}^{j+W}\sum_{n=k-W}^{k+W}[f(m,n)]^2}$$

is calculated in the window which size is equal $(2W + 1) \times (2W + 1)$ for every point on the image (j, k) and with deviation $(\xi, \eta) = 0; \pm 1; \pm 2;$ ... Here $f(m, n)$ is the brightness of the pixel at point (m, n), W – number of pixels in each dimension

At a fixed shear (ξ, η) large values $A(\xi, \eta, j, k)$ will correspond to the image area with a large area with the same texture component, i.e. the grain size is proportional to the width of the texture of the autocorrelation function.

It is necessary for practical application of object detection algorithm that not all objects will be allocated on each frame of a video stream if their pixels fall within the ε – neighborhood. But only that object will be allocated if the pixel is within the chosen part of image. The pixel coordinates of the user object are defined in the spatial coordinates, and the tracking under this pixel is implemented. An Lucas–Kanade method is used as such algorithm.

Let the intensity of the selected by user pixel is equal $I(x, y, t)$. It becomes equal $I(x + \delta x, y + \delta y, t + \delta t)$ because of the pixel displacement in the next frame. These intensities are equal. Assuming that the displacement between frames is small, the intensity $I(x + \delta x, y + \delta y, t + \delta t)$ is expanded in a Taylor series:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x} * \delta x + \\ + \frac{\partial I}{\partial y} * \delta y + \frac{\partial I}{\partial z} * \delta z$$

.

Hence the optical flow vector is a solution of equations:

$$\begin{cases} I_x(q_1) * V_x + I_y(q_1) * V_y = -I_1(q_1) \\ I_x(q_2) * V_x + I_y(q_2) * V_y = -I_1(q_2) \\ \dots \\ I_x(q_n) * V_x + I_v(q_n) * V_v = -I_1(q_n) \end{cases}.$$

This equation can be written in matrix form:

$$Av = b.$$

Optical flow vector is calculated by solving this system using the method of least squares:

$$\begin{pmatrix} V_x \\ V_y \end{pmatrix} = \begin{pmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i) * I_y(q_i) \\ \sum_i I_x(q_i) * I_y(q_i) & \sum_i I_y(q_n)^2 \end{pmatrix}^{-1} * \\ * \begin{pmatrix} -\sum_i I_x(q_i) * I_1(q_i) \\ -\sum_i I_y(q_i) * I_1(q_i) \end{pmatrix}.$$

During the processing of a pyramidal iterative algorithm is used and the point is searched that minimizes the weighted quadratic model of restrictions on first-order derivatives in the search window.

Before application of the algorithm it is necessary to smooth the input image to remove random noise, because the algorithm is local and it depends strongly from any noise.

The main limitations of the algorithm are its assumptions - constant brightness of moving points and the same direction of movement within the search window.

The results of detection algorithm based on the Lucas-Kanade method are illustrated for test object on Fig.2.

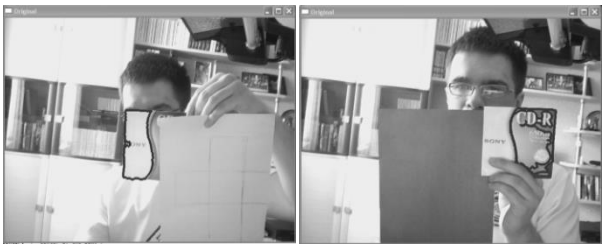

Fig. 2. Detection algorithm based on the Lucas-Kanade method

IV. Main part

Regular links are used between the characteristics detected on the training sample when constructing decision functions. Training sample is a hand gesture image for recognition [2].

Training set has to be moved into a new space whose dimension is much greater than 2. The principal component analysis gives the possibility to form a space so that the data are located optimally in it. The choice of the required space dimension depends on the desired quality hand gesture recognition. As a result space dimension is formed, in which each image of the original sample is presented in summary form.

There are various methods for detecting hands in the image. These methods can be divided into two groups: vision-based approach and 3D hand model based approach.Hand detection method is implemented on the video stream with using Haar classifier. It is based on the image pre-processing algorithm and background subtraction algorithm.

Nowadays Viola-Jones method with using Haar features and AdaBoost-classifier is one of the best algorithms for solving problems of finding objects on the video stream in real time. In this paper, Viola-Jones method is used to solve the problem of hand detecting in real time.

Haar-like features are calculated by the formula::

$$f(x) = \sum_{S_1} I - \sum_{S_2} I$$

where I – pixels intensity; S_1 – all pixels in the area of dark rectangle; S_2 – all pixels in the area of white rectangle. The simple and advanced Haar-like features used in this paper are illustrated on Fig. 3 and Fig. 4.

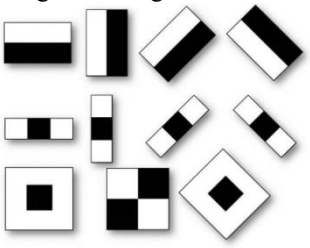

Fig. 3. Simple Haar-like features

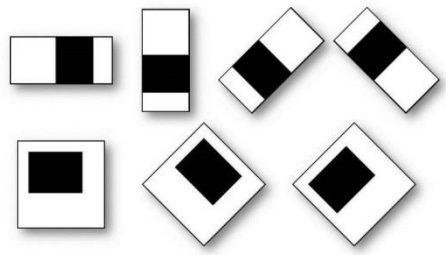

Fig. 4. Advanced Haar-like features

Integral image. Integral representation of the image is a matrix with the same size as the original image. The sum of the intensities of all pixels located above and to the left of this element is stored in each its element. Matrix elements are calculated using the following formula:

$$L(x,y) = \sum_{i=0,j=0}^{i\le x, j\le y} I(i,j)$$

where I(i, j) — pixel brightness of the original image.

Using the integral image allows to calculate the Haar-like features more quickly. And it does not depend on the size of image.

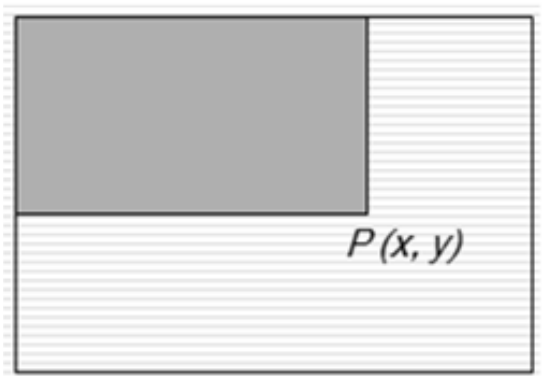

Fig. 5. Integral representation

Each element from the matrix L[x, y] represents the sum of pixels in the rectangle from (0,0) to (x, y), i.e.value of each pixel (x, y) is the sum of all the pixels to the left and above the current pixel (x, y). Matrix calculation takes linear time proportional to the number of pixels in the image, so the integral image is calculated in a single pass [5].

Boosting is the set of methods that improve the accuracy of the analytical models. Effective model that admits little misclassification is called "strong". "Weak" model does not allow to provide a significant separation between classes or to give accurate predictions. It makes a lot of errors. Therefore, boosting means "strengthening" of "weak" models. It is a procedure for constructing the serial composition of machine learning algorithms, where each following algorithm tries to compensate the shortcomings of the composition of all previous algorithms.

Auxiliary set of R, called the space of estimates, is introduced along with the sets X and Y. Superposition algorithms are considered such as $a(x) = C(b(x))$, where function $b: X \to R$ is called an algorithmic operator function, function $C: X \to R$ n is called a decision rule.

Many classification algorithms have exactly this structure: first assessment accessories to object classes are computed, and then the decision rule translates these estimates in the class number. Estimation value are generally characterizes the degree of classification confidence.

Algorithmic composition is an algorithm $a: X \to R$ with form: $a(x) = C(F(b_1(x), \ldots, b_T(x)), \; x \in X$, which is composed by algorithmic operators $b_t : X \to R, t = 1, \ldots, T$, which corrects operation $F: R_T \to R$ and decision rule $C: R \to Y$.

Generic algorithms are denoted by function $a_t(x) = C(b_t(x))$, and operators $b_t(x)$ are general algorithms too if a decision rule C is fixed. Superposition with type $F(b_1, \ldots, b_T)$ is mapping from X to R, and it is an algorithmic operator.

In classification problems set of real numbers is commonly used as a guest space into two disjoint classes. Decision rules may have customized settings. Thus, in the algorithm of Viola-Jones a threshold decision rule is used. At the beginning the operator at zero is constructed, and then the optimal value is selected. Process of sequential learning of basic algorithms is used often in the construction of compositions.

Various criteria may be used depending on the specific of the problem. It is also possible to use several criteria:

- a predetermined number of basic algorithms is built;
- specified accuracy is achieved on the training set;
- it is impossible to improve accuracy over the last few steps at a certain algorithm parameter.

Development of this approach was the design of the improved family of boosting algorithms Adaboost [3].

Decision:

1. Initialization of objects weights:

$$w_i := 1/\ell, i = 1, \ldots, \ell;$$

for $t = 1, \ldots, T$

2.1. $b_t := \arg min_b Q(b; W^l)$

2.2. $a_t := {}^1/_{2ln1} - \frac{1-Q(b;W^l)}{Q(b;W^l)}$

3. Recalculation of object`s weights. The rule multiplicative conversion scales. Weight of the object increases when b_t admits mistake on it, and the same factor

decreases when b_t correctly classifies x_i. Thus, just before setting the basic algorithm accumulates the greatest weight is in those objects that are often difficult for the previous algorithms:

$$w_i \coloneqq w_i \exp\left(-a_t y_i b_t(x_i)\right), \quad i = 1, \dots, l$$

4. Normalization of object weights

$$w_0 = \sum_{j=1}^{l} w_j; w_{i=} w_i / w_0, \quad i = 1, \dots, l$$

Video stream obtained with a help of a video camera, is a sequence of frames. For each frame integral image is calculated. Then, the window frame is scanned by small size window containing Haar-like features. For each j-th feature the corresponding classifier is defined by the formula.

$$h(x) = \begin{cases} 1, & p_j f_j(x) < p_j \theta_j \\ 0 & \end{cases}$$

where x – small size window; θ_j– threshold; p_j– direction of the inequality sign; f_j–Haar-like feature.

AdaBoost-algorithm can improve classification accuracy through a series of stages of weak classifiers. As a result, weighted combination of weak classifiers is calculated.

$$H(x) = \sum_{i=1}^{N} a_i h_i$$

(N –number of weak classifiers; a_i– a coefficient derived from the database; h_i – weak classifier).

V. GESTURE CLASSIFICATION

Hand gesture recognition algorithm was created based on the detection algorithm. This algorithm builds a hand envelope, counts straighten fingers in real time and performs tracking that can be used for remote control of radio-physical and other technical devices. The number of defective pixels are counted in progress of this algorithm. Depending on their amount concludes amount fingers. Conclusion about the amount of fingers depends on the number of defective points [4].

Sliding window is used for background detection:

$$BG[i][j] = \alpha BG[i][j] + (1 - \alpha) Frame[i][j]$$

Background pixels are not affected by the weighting, because background is a static object. Pixels that are constantly changing, are not the background image. After each iteration the background pixels are becoming more noticeable especially if they are compared with the pixels of the moving object (hands). Background subtraction involves the calculation of the reference image, the subtracting of each new frame of the image and the image segmenting which identifies nonstationary areas of objects.

Fig. 6. Background subtraction algorithm

Convex hull is in the fingers, because they are limbs. And hence this fact can be used to detect "NO fingers". But on the hand there are other special points - convex defects - the deepest points of diversion on the contour. To implement the hand tracking center of the palmhave to be find. There are 2 convex defects for every convex point. Hence they form a triangle with approximately the same distance between maximum and minimum. The position of minima should be close to the circumference of the palm. In addition, the ratio of the radius to the length of the finger palm should be about the same. It is assumed that the palm is inclined so that it represents a circumference. Because of the noise averaging of the images is produced that will give us the exact position of the center of the palm and its size.

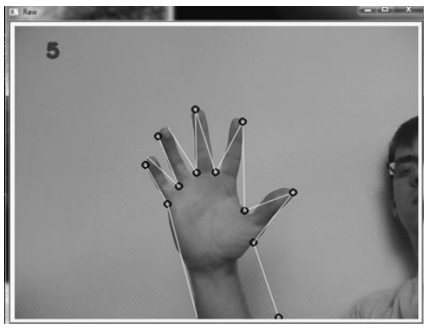

Fig. 7. Convex hull and defects

VI. RESULTS AND TABLES

Considered hand detection algorithm was tested on a database of digital images. Database size was 1000 images. It was a set of gestures.

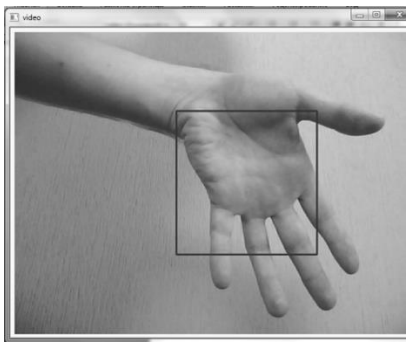

Fig. 8. Hand detection algorithm

All measurements were performed on a personal computer with the following configuration: Intel(R) Pentium(R) D 930 CPU 3.00GHz, RAM 4GB 400MHz, Windows 7 64-bit.

TABLE I. RESULTS OF TESTING THE ALGORITHM

File mane	Found	Missed	False positives
DSC_0001.jpg	1	0	0
DSC_0002.jpg	1	0	0
DSC_0003.jpg	1	0	0
DSC_0004.jpg	0	1	0
…	…	…	…
Total	782	218	232

Cascade order: 10.
Number of weak classifiers: 691.
Total time: 27.862 ms.

Gesture classification was tested on real video stream. During execution two Haar classifiers based on different training samples were created, a comparative analysis was performed. The first classifier was built using 3123 positive samples and 1492 negative samples. The second classifier was based on the 7294 and 3853 samples, respectively. The classification results qualitatively changed if the second classifier has been used. Because it was trained on larger positive and negative samples. The number of false positives is decreased for the second classifier. The proportion of detected objects (hands) is increased. Processing time of one picture has decreased significantly. Second classifier detects a hand with better quality in terms of a minimum share of false positives. This is a consequence of the choice of higher quality and volume of training samples.

The method of detection the error curve or ROC-curve (Receiver Operation Characteristic) is used for rating the performance of the algorithm. It is a characteristic curve of a binary classifier. This is a ratio of the level of true positive classifications to the level of false positive classifications by varying the threshold of the decision rule [6].

ROC-curves quite clearly characterize the predictive ability of the constructed model. The most effective classifiers are described by curves, which are located as close as possible to the upper left corner of the coordinate system. As you can see, the sensitivity of the pattern was more than 85%.

The level of true positive decisions of a classifier (TP) is denoted as $TPR = \frac{TP}{TP+FN}$, and the level of false positives (FP) solutions - $FPR = \frac{FP}{TN+FP}$, sensitivity of model – $Se = TPR$, specificity of model – $Sp = 1 - TPR$. Here FN (false negative) is the number of false negative decisions of the classifier, and TN (true negative) is the number of true negative decisions.

From these definitions it follows that the model with high sensitivity gives the true result in the presence of a positive outcome in most cases. Model with high specificity often gives the true result with a negative outcome [7].

Among the most important requirements to the parameters of the algorithm are:

- a minimum value of the sensitivity of the model (in this case, the optimal threshold value will be the value of model specificity, which is attained at a given minimum sensitivity);
- the maximum value of the total sensitivity and specificity:

$$\text{Thresh} = \max_k(\text{Se}_k + \text{Sp}_k).$$

- the balance between sensitivity and specificity of the model:

$$\text{Thresh} = \min_k |\text{Se}_k - \text{Sp}_k|.$$

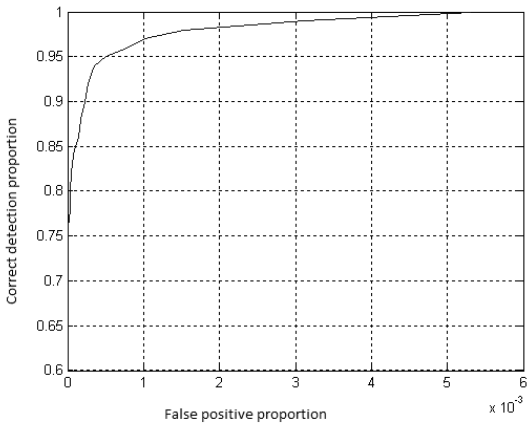

Fig. 9. ROC-curve for detection algorithm

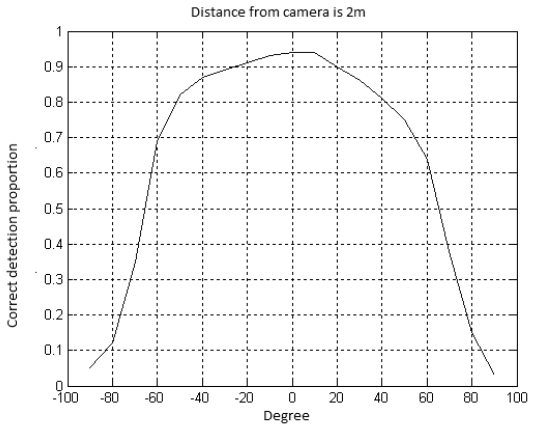

Fig. 10. The dependence of correct detection from the angular position

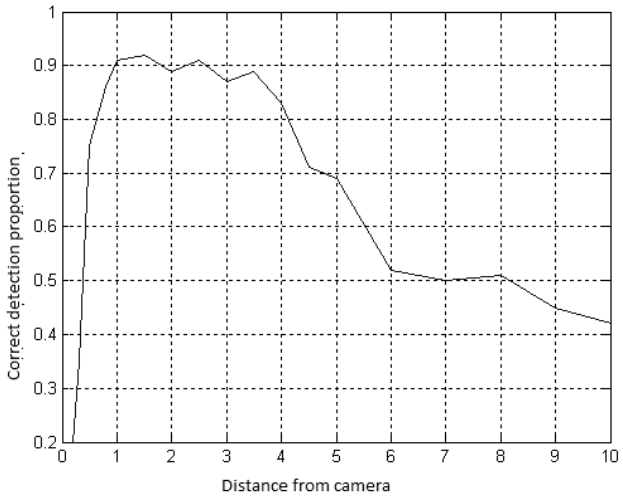

Fig. 11. The dependence of correct detection from the distance

Based on these dependencies we can make a conclusion that hand is correctly detected with a probability of 91% at distances up to 0.5m to 4.5m. For other values of the distance the proportion of recognition errors is increased. This is due to the fact that hand has few characteristic features. The size of the scanning window decreases when distance increases. Consequently, the algorithm requires low level of noise in the image, which greatly affect the object detection.

It is revealed in the study of dependence of correct detection from angular position that at a distance of 2 meters from camera probability of detection is more than 80% if the angular aperture is 93˚. It can be explained by the boundaries of the applicability of Viola-Jones method.

VII. Conclusion

It is proposed a combined method of hand gesture recognition based on a statistical image processing algorithm. As a pre-processing algorithm there were applied Lucas–Kanade method, background subtraction algorithm, color detection and texture detection methods. Object recognition was performed with using of Haar classifiers. The possibility of using gestures for remote control of various devices with different hands position from the camera location was shown.

The analysis of the applicability of the algorithm is made under different external conditions. The accuracy of the algorithm is high enough and recognition is less exposed to errors of the first and second kind when the camera is oriented within the permissible angular aperture, which carried out the detection arm.

References

[1] Wang R. Y., Popovic J. *Real-time hand-tracking with a color glove*. New Orleans, Louisiana: ACM, 2009

[2] Stockman G, Shapiro L. *Computer vision*. Moscow, Binom. Labor science, 2006.

[3] Gonsales R., Vuds R., Jeddins S. *Digital image processing in MATLAB*. Moscow: Tehnosfera, 2006

[4] Song P., Winkler S., Gilani S., Zhou Z. *Vision-Based Projected Tabletop Interface for Finger Interactions*. Human–Computer Interaction, 2007.

[5] Lukyanitsa A.,Shiskin A. *Digital video processing*. Moscow:Press, 2009.

[6] Priorov A., Volokhov V., Sergeev E., Mochalov I., Tumanov K. *Parallel Filtration Based on Principle Component Analysis and Nonlocal Image Processing* // Proc. International MultiConference of Engineers and Computer Scientists 2013. Hong Kong, 2013. vol. 1. pp. 430-435.

[7] Priorov A., Apalkov I., KhryashchevV. *Digital image processing*. Yaroslavl: Yaroslavl State University, 2007.

[8] Kolodnikova N.V.*Overview of textural features for pattern recognition problem.* TURUS reports:Automated data processing systems, 2004, pp. 113-124.