1998

INTEGRALLY DIFFERENTIAL ALGORITHM OF BOUNDARY ALLOCATION FOR COMPUTER VISION SYSTEMS

In this paper, we propose the algorithm of allocation of an object contour, based on the analysis of the histogram of the image. The algorithm is simple in realization and suitable for use in computer vision systems working in real time.

PhD A. Lipanov
Doctor of Science, Prof. E. Putyatin

The process of moving object tracking begins with a definition of an object that the system should monitor [1]. The operator defines with the help of index a point on the object which is a start point for one of the algorithms of segmentation, by means of which the background around the object is cleaned in order to define the sizes of a tracing frame.

By solving a problem of tracking it is expedient to allocate an object figuration; that allows to carry out accumulation of contours (figurations) and, therefore to watch (observe) the dynamics of an object, to define the sizes and forms of frames on a contour. At technical realization of tracking systems tracing frames of square or rectangular form are used more often. For allocation of objects figurations there are various algorithms based on spatial derivation with the use of convolution.

In this paper, we describe the developed integrally differential algorithm of allocation of an object boundary, where we first define the threshold of points belonging to a boundary, and afterwards occurs its allocation.

Let's enter the general designations and the definitions necessary for the description of algorithm. Then we make an assumption that the sizes of an area, from which the algorithm works, equal to half of linear sizes of a vision area (field of vision). We shall notice that in a concrete computer model of the automated tracking system the size of this area is 128x128 points (viewing field 256x256). We shall designate this set of points **D'**.

Definition 1. Neighborhood δ of image point **B (i, j)** we shall name a set of points lying directly around the point **B (i, j)**. In that specific case neighborhood δ has the size 3x3 points. Record $\delta(\overline{B}(i,j))$ should be understood as a neighborhood of point **B (i, j)**, excluding point **B(i,j),** δ**(B(i,j))** – neighborhood **B (i, j)** together with point **B (i, j)**.

We consider the point of target detection as a start point for the algorithm, and its coordinates as coordinates of the object center. In neighborhood of this point we define average brightness C_0:

$$C_0(\delta(B(i,j))) = \frac{1}{card\delta(B(i,j))} \sum_{B_{i,j}\in\delta(B(i,j))} B(i,j) \quad , \qquad (1)$$

Where **B (i, j)** is a function of brightness of the image, **B(i,j)**$\in\delta$**(B(i,j))**.

Let **K** be a set of values of brightness of the image, $k=\overline{0,255}$, **k**$\in$**K**, N_K –quantity of points of brightness **k**, defined on function of brightness **B (i, j)** in viewing area **D'**. We shall designate N_{cp} as a quantity of points which brightness is equal to the value C_0 in viewing D '.

We enter the function of quantity of points N_K for each brightness **B (i, j)=k**, i.e. the histogram of distribution of image point brightness in working area **D'** (fig. 1).

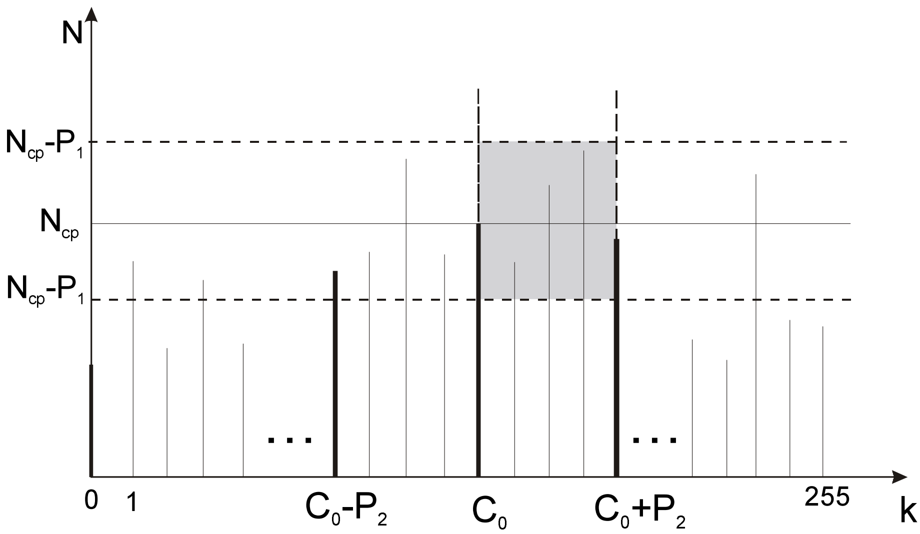

Fig. 1 - Histogram of distribution of working area brightness

We define the size of threshold **T** of points belonging to a contour as follows

$$T = \sum_{k=C_0}^{255} \left(T_0 + 1 \mid \left\{ \left| N_k - N_{cp} \right| \geq P_1 \cap \left| C_0 - k \right| \leq P_2 \right\} \right), \tag{2}$$

where $\mathbf{T_0=0}$ in the beginning of determination procedure of threshold **T**,

$\mathbf{N_к}$ – quantity of points of brightness **k**,

$\mathbf{N_{cp}}$ – quantity of points, which brightness corresponds to brightness $\mathbf{C_0}$,

k – current brightness,

P1> 0, P2> 0 – thresholds set in the beginning of allocation procedure of a contour.

At the determination of a threshold **T** the oversight of brightness range begins with value $\mathbf{k=C_0}$; that allows to remove part of random interferences having low brightness and capable to influence accuracy of threshold **T** size definition. Threshold **T** is a quantity of the brightness exceeding value $\mathbf{N_{cp}-P_1}$, got into the painted over (colored) area on fig. 1.

It is experimentally established, that for CVS it is possible to recommend value $\mathbf{P_1=64}$**,** $\mathbf{P_2=50}$. Generally the sizes of thresholds are chosen empirically, depending on a class of images with which the tracking system works or on the basis of the statistical analysis.

Set **D'** contains three subsets: a subset of object points, a subset of points of an object boundary, a subset of points of a background (all the points not included into the first two subsets). Thus, the set of contour (figuration) points can be determined by removal from **D'** of all the points related to an object and background.

Let's enter set $\mathbf{D^1}$, equipotent to set **D'**, and its element $\mathbf{d^1(i,j)\in D^1}$ we define as follows

$$d^1(i,j) = \begin{cases} 1, & \left| B(i,j) - \overline{B}(m,n) \right| < T, \ \forall\, B(i,j) \in \delta(\overline{B}(i,j)) \\ 0, & \left| B(i,j) - \overline{B}(m,n) \right| > T, \ \forall\, B(i,j) \in \delta(\overline{B}(i,j)) \end{cases}, \tag{3}$$

where $i = \overline{1,128},\ j = \overline{1,128}$,

$\overline{B}(m,n)$ – brightness of points of neighborhood $\delta(\overline{B}(i,j))$,

B(i,j) – brightness of the central pint of neighborhood.

Indexes m and n run across neighborhood $\delta(\overline{B}(i,j))$.

In set $\mathbf{D^1}$ the element $\mathbf{d^1(i,j)=1}$ specifies the point of a contour, and $\mathbf{d^1(i,j)=0}$ specifies the points of an object, background or interference.

On fig. 2 there is a possible arrangement of points in neighborhood $\delta(\overline{B}(i,j))$ of point **B(i,j)**.

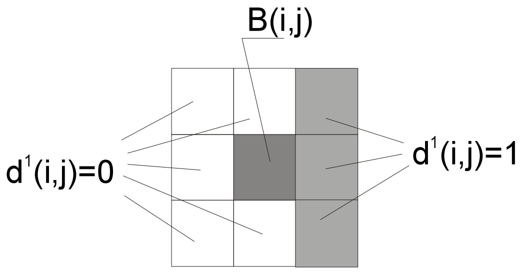

Fig.2 – Neighborhood of point B (i, j)

With light grey color are those points marked which correspond with $\mathbf{d^1(i,j)=1}$, white $\mathbf{d^1(i,j)=0}$, point **B(i,j)** is compared to the points of neighborhood, but value 0 or 1 isn't assigned to it on this step.

Such formation method of set **D^1** allows allocating in set **D** a group of points with close values of brightness. On the other hand, random glitches and noise interferences will be removed, as their brightness as a rule differs considerably from the brightness in the points of neighborhood $\delta(\overline{B}(i, j))$.

Let's enter set **D^2**, equipotent to set **D'**. Element **$d^2(i,j)\in D^2$** we define as follows

$$d^2(i,j) = \begin{cases} 1, & C_0 - T < B(i,j) < C_0 + T, \ \forall\ B(i,j) \\ 0, & B(i,j) \notin (C_0 - T, C_0 + T), \ \forall\ B(i,j) \end{cases}, \quad (4)$$

where $i = \overline{1,128},\ j = \overline{1,128}$.

Set **D^2** will include those points which brightness is close to the brightness **C_0**, within the limits of T.

The difference in formation of sets **D^1** and **D^2** is that when we chose in the first set we get point brightness of neighborhood $\delta(\overline{B}(i, j))$ in comparison with brightness **B(i,j)**, and in set **D^2** we select points which brightness gets in brightness interval **$(C_0 - T, C_0 + T)$**, and at the same time all points of neighborhood **$\delta(B(i,j))$**, including **B(i,j)** are being analyzed.

Set **D^1** already includes instructions on the presence of boundary points. If we use only **D^1** for definition of a boundary of an object, it will not be possible to allocate it correctly in many cases. Using only D1 to define a contour, the algorithm becomes less jam-resistant and strongly depends on what point of an object has become a starting one for it. In order to eliminate of this lack we use set **D^2**, that was defined earlier. It specifies the presence of points of an object and a background **$d^2(i,j)=1$**, and also **$d^2(i,j)=0$** - points of a contour and interference.

Set L of points of a contour **l(i,j)** we define by removing points from **D'** in accordance with the following condition

$$l(i,j) = \begin{cases} 255, & ((d^1(i,j)=1) \wedge (d^2(i,j)=0)) \vee (d^1(i,j)=1) = 1, \\ B(i,j), & ((d^1(i,j)=0) \wedge (d^2(i,j)=1)) \vee (d^1(i,j)=0) = 0, \end{cases} \quad (5)$$

where – $i = \overline{1,128},\ j = \overline{1,128}$, **$d^1(i,j)\in D^1$, $d^2(i,j)\in D^2$, $l(i,j)\in L$**.

The removal of a point corresponds with assignment to it of a code 255 (white color), provided that the expression **$(d^1(i,j)\wedge d^2(i,j))\vee d^1(i,j)$**, for corresponding elements from **D^1** and **D^2**, is equal 1. At formation of set L, if the condition is not fulfilled, the point with brightness **B(i,j)** remains, in order to increase clearness of a contour it is possible to replace this point with a point of black color. In other interpretation, elements from sets **D^1** and **D^2**, making the expression **$(d^1(i,j)\wedge d^2(i,j))\vee d^1(i,j)$** equal 1, specify those points belonging to an object or to a background, they are marked with white color. In this condition the preference is given to the points from set **D^1** as there is an object and a contour in this set. So the points of a contour in set **L** are not marked with white color, i.e. the procedure of allocation of an object and of a background is done, then all these points are being whitened, and there is only a contour left – points of a boundary **l(i,j)=B(i,j)** or **l(i,j)=0** (a 0-code of black color). Set **L** contains an object boundary, i.e. it is a frame of the wrong form. That solves the given task – defining of tracing frame **S** in the form of an object boundary.

The received boundary **L** can be used to define the sizes of a rectangular frame. In order to this, we should view set **L** from left to the right, from right to the left, from top to the bottom, from bottom to the top and define coordinates of the first points that we see while viewing the set. So the initial values of parameters of tracing frame **S** are defined: **W** - width, **H** - height of a tracing frame. During the process of allocation of a contour there can be parts of a background left that can be considered as interferences. The presence of interferences will influence the choice of tracing frame size, as procedures of definition of the size can take them for some part of object by mistake.

The described algorithm gives good results of allocation of an object contour. The advantage of the algorithm is its simplicity that allows increasing the reliability of a tracing system. Working speed of algorithm is high since the elementary logic operations are used, that enables to use it during tracking for periodic updating of a contour in real time and accumulation of contours with the purpose of their specification.