1998

CONSTRUCTION OF ALGORITHMS FOR TRACING FRAME NORMALIZATION

In this paper, the methods of calculation of tracing frame coordinates increment, scale distortion and angle of rotation in the tracing process of moving object are introduced. Furthermore a parallel-serial algorithm of size correction of tracing frame in the tracing process is presented.

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

The normalization (compensation of geometric transformations) of a tracing frame is often applied in practice for the groups of offset, rescaling and rotation, though the situations are possible, where there is frame normalization at affine and collineatory transformations is necessary. Shift, rescaling and rotation of an object are the most common transformations. The image inside the frame is an area **D**.

The movement of an object during the time $\mathbf{t=t_{k+1}-t_k}$ causes the transformation of the image field from one condition into the other one, which can be described with the change of intensity distribution. Time **t** can be interpreted as a time interval between two television images or as an interval equal to a time interval that is necessary to make one cycle of calculation according to the normalization algorithms.

The tracing task is solved on a real background for a real object that describes some path with a certain speed.

Let's take a look at the picture inside the tracing frame **S** in two time points $\mathbf{t_1}$ и $\mathbf{t_2}$ (fig. 1). Suppose the image is not centered and coordinate system must be combined with the object centre.

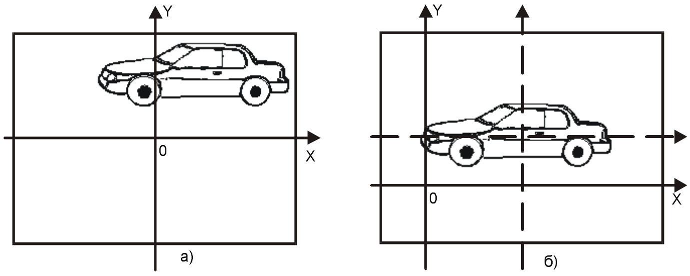

Fig.1 – Object in the tracing moment in two sequential time points: a) in time point $\mathbf{t_1}$ (before the normalization); b) in the time point $\mathbf{t_2}$ (after the normalization)

Directly from the theory of coordinate normalizers [2] we can get an expression to define frame offset in the process of tracing.

$$\Delta m = \frac{\iint\limits_D B(x,y,t_1)xdxdy}{\iint\limits_D B(x,y,t_1)dxdy} - \frac{\iint\limits_D B(x,y,t_0)xdxdy}{\iint\limits_D B(x,y,t_0)dxdy}, \qquad (1)$$

where $\mathbf{t_0}$ is a previous time point, $\mathbf{t_1}$ is a current time point, $\Delta\mathbf{m}$ is movement along the axis **X**.

This expression allows calculating the movement of image center of gravity along the axis **X**. Similar expression can be written down for the movement along the axis **Y** as well. The disadvantage of this calculation method of the movement is that there is a necessity to divide by value of energy at each step, which react on every background changes and introduces considerable errors.

To increase noise immunity (1) we apply the approach of comparison between the parts of the image with respect to the axes **X** and **Y**. Let's divide vertically the tracing frame into two equal parts. Then at balance state the difference between left and right parts should equal zero, i.e. the image becomes equal along the axes **Y**:

$$\iint\limits_{D_1} B(x,y,t)xdxdy - \iint\limits_{D_2} B(x,y,t)xdxdy = 0,$$

where area $\mathbf{D_1}$ is a left part of the image and area $\mathbf{D_2}$ is a right part of it.

If there is a movement along the axes **x**, then this difference will show the movement of the object with respect to the beginning of coordinate system along the axis **x**

$$\iint\limits_{D_1} B(x,y,t)xdxdy - \iint\limits_{D_2} B(x,y,t)xdxdy = \Delta x. \qquad (2)$$

Now we divide the image horizontally into two equal parts and write down a similar expression

$$\iint\limits_{D_3} B(x,y,t)ydxdy - \iint\limits_{D_4} B(x,y,t)ydxdy = \Delta y. \qquad (3)$$

where D_3 is a top part of the image and D_4 is a bottom part of it.

It should be noted that formulae (2) and (3) help us find not the real movement of an object inside the frame, but some kind of an error, the value of which enables to compensate (balance) the movement of a tracing frame along the axes and to combine the center of gravity with the frame center.

We need the normalization of scale distortion so that the control system could change the size of a frame on time and not to lose the object from the viewing field.

Calculation of the rescaling can be done while observing the changes of image energy inside of the tracing frame. Energy of the image is a sum of brightness in every point of the image.

$$\lambda^2 = \iint_D B(x,y,t)dxdy .$$

If we calculate $\lambda_2^2 / \lambda_1^2$, i.e. the relation of the energies on the current step **(λ_2)** and on the previous step **(λ_1)**, we can define whether the sizes of the tracing frame should be scaled up or reduced. If we scale up the sizes of the object, it will fill up the most part of the frame. If we reduce the sizes, it will up the smaller part. The energy of the image will change accordingly.

Such an approach is quite easy to implement, provides high speed of algorithm functioning, but it has its own disadvantages and that is weak noise immunity. The algorithm works well with uniform or quite uniform background. If any local interferences or any parts of other objects appear in the frame area, the algorithm can calculate the value of factors inaccurately.

Let's take a look at another method of rescaling evaluation. Let's present the change of scale coefficients in exponential form:

$$\mathbf{M_x(t)} = e^{\lambda_1 t} , \mathbf{M_y(t)} = e^{\lambda_2 t} ,$$

Where λ_1, λ_2 – scale coefficients along the axes **X** and **Y.**

If the scale along the axes **X** and **Y** changes identically it is defined in the following way:

$$\Delta M = e^{\lambda_1 t_0} - e^{\lambda_2 t_1} ,$$

where λ_1, λ_2 are scale functions in time points $\mathbf{t_0}$ and $\mathbf{t_1}$.

As the time point $\mathbf{t_0}$ goes with the functioning start of the control system or with the moment, when we begin to track the parameters of tracing frame, we can assume, that function λ_1**=1**. We must find function λ_2. We find function λ_2 as a relation of the points μ_{20} in time points $\mathbf{t_0}$ и $\mathbf{t_1}$:

$$\lambda_2 = \frac{\iint_D B(x,y,t_0)x^2dxdy}{\iint_D B(x,y,t_1)x^2dxdy} = \frac{\mu_{20}(t_0)}{\mu_{20}(t_1)} .$$

Formula evaluation of rescaling:

$$\Delta M = 2\left(e - e^{\frac{\mu_{20}(t_0)}{\mu_{20}(t_1)}} \right). \qquad (4)$$

Coefficient 2 in (4) takes into account the rescaling along the axis Y. The experiments have shown, that the rescaling, that is being evaluated according to the formula (4), meets the demands of precision while using them in control system.

To increase the accuracy of rescaling finding we can also take into account the change of image energy μ_{00} during the time point Δ**t** by intensification of the value Δ**M** in (3) with value Δ**M'**:

$$\Delta M' = 1 - \frac{\mu_{00}(t_0)}{\mu_{00}(t_1)}. \quad (5)$$

To adjust the sizes of a tracing frame we need to multiply the values of linear sizes of the frame by value **1-ΔM** regardless of the fact whether the size of the frame grows or becomes smaller. Outcome of experiment shows that formula (4) taking in account formula (5) makes it possible to define the rescaling quite accurately. It should be mentioned that be the value of rescaling **ΔM<0.1** it is not reasonable to change the size of a tracing frame. It can lead to the wrong adjustment of the frame size, because at minor (small) scale change the calculations are much more inaccurate.

The third and the most wide spread method of image transformation tracing system will be working with is rotation.

Angle of rotation increment is defined with the following relation

$$\varphi' = \frac{\mu_{02}}{\mu_{20}} = \frac{\iint_D B(x,y,t)y^2dxdy}{\iint_D B(x,y,t)x^2dxdy}.$$

Such definition of the increment allows defining small angles of rotation up to 2 degrees. On the other part for the calculation this increment we need to calculate only point μ_{02} at each step (point is calculated while defining scale change). In order to get true value of the angle, we need to accumulate value **φ'** in the process of tracing:

$$\varphi' = \varphi'(t_0) + \frac{\iint_D B(x,y,t_1)y^2dxdy}{\iint_D B(x,y,t_1)x^2dxdy}. \quad (6)$$

where $\mathbf{t_1}$ is a present time moment, $\mathbf{t_0}$ is a previous time point.

After the angle increment is found we need to adjust the tracing frame sizes using formula evaluation:

W'=Hsin(φ')+Wcos(φ'), H'=Wsin(φ')+Hcos(φ'), (7)

where **W'** is width of the tracing frame after adjustment, **H'** is height of the tracing frame after adjustment, **W** is a given width of the frame, **H** is a given height if the frame.

Taking in account the scale change we have found above, transformation (7) will look like:

W'=HΔMsin(φ')+WΔMcos(φ'), H'=WΔMsin(φ')+HΔMcos(φ'), (8)

where **ΔM** is a scale change.

According to the algorithms above it is possible to state a parallel-serial tracing algorithm. After the tracing object is captured, we calculate its initial angle of rotation with the help of normalizer of coordinate type [2] and if it is necessary, we also perform the adjustment of CVS platforms. Then using normalizers (2) and (3) we trace a moving object. We define at the same time values **Δx** and **Δy** and add them to coordinates of the top left corner of the tracing frame. At the same time we calculate the points of second order and trace scale and rotation changes. We perform the adjustment of the sizes of a tracing frame using formula evaluation (8) when necessary. This algorithms has all been run on a computer model CVS, which proved their work efficiency.

Список литературы

1. Путятин Е.П., Липанов А.В., Прокопенко Д.О., Теоретические предпосылки слежения за протяженными объектами. Сборник научных трудов. Системы обработки информации. с.80-84. Харьков: НАНУ, ХВУ, 1998. – 160 с.

2. Путятин Е.П., Аверин С.И. Обработка изображений в робототехнике. - М.: Машиностроение, 1990. - 320 с.