

## REAL-TIME RECOGNITION AND TRACKING OF MOVING OBJECTS

Karol DOBROVODSKÝ, Pavel ANDRIS

*Institute of Informatics  
Slovak Academy of Sciences  
Dúbravská cesta 9  
845 07 Bratislava, Slovakia  
e-mail: {utrrdobr, utrrandr}@savba.sk*

**Abstract.** A multi-camera monitoring system for online recognition of moving objects is considered. The system consists of several autonomous vision subsystems. Each of them is able to monitor an area of interest with the aim to reveal and recognize characteristic patterns and to track the motion of the selected configuration. Each subsystem recognizes the existence of the predefined objects in order to report expected motion while automatically tracking the selected object. Simultaneous tracking by two or more cameras is used to measure the instant distance of the tracked object. A modular conception enables simple extension by several static and mobile cameras mechanically oriented in space by the pan and tilt heads. The open architecture of the system allows additional subsystems integration and the day and night image processing algorithms extension.

**Keywords:** Image processing, pattern recognition, tracking

**Mathematics Subject Classification 2010:** 68U10, 97P99

### 1 INTRODUCTION

Detecting moving objects from a video sequence is a fundamental task in many computer vision applications such as human detection and tracking, traffic monitoring and military applications. A general approach to this problem lies in performing background subtraction, which identifies moving objects from the video frame that

differs significantly from a background model. The paper deals with the various types of position coordinates starting with the camera coordinates on the one side and ending with the 3D scene coordinates of the virtual world. The goal is to predict the motion and to support tracking of the object already caught by one or more of the vision subsystems. The combination of an automatic camera orientation control with the motion prediction and the automatic recognition facilitates the ability of the whole system to monitor and track moving objects while working in a real-time.

There are several specific problems which a motion detection subsystem must deal with. First of all, it should eliminate motion caused by camera sensor displacements and changes in global scene brightness. The system consists of several relatively independent subsystems having the ability to recognize object from the background, track its position and report the velocity and acceleration of the tracked object.

The recognition of moving objects from the moving camera picture in a real-time is an extremely complex problem. The active motion of cameras makes impossible to eliminate the static background as the existing monitoring systems with static cameras do. Moreover, multi-level image segmentation accentuates the edges of the clouds as well as objects flying on the cloudy background [7, 9]. Many spurious objects are produced by the subsequent image processing analyzer from these edges. For this reason, it is not possible to identify the movements using a simple comparison of the sequence of consecutive images [10]. Potential objects of tracking must first be recognized without specifying their identity in the context of consecutive images. Then the estimation of their identity in the context of the previous positions may be detected [11]. An extensive approach to the recognition of potential objects to be tracked is necessary. The mechatronic tracking system recognizes many potential objects for tracking. At this stage, the system is working in search mode and recognizes the greatest number of objects that may be subject to tracking. Simple differential filtration is used in this mode. An operator makes the choice from the offered objects and the system starts the track mode. From this moment the tracked object is the only one to be recognized by the visual subsystem. Online differential filtration with measuring the size and brightness of the object is introduced to increase the reliability of the tracking process. Especially for flying targets, it is difficult to initiate the tracking process by putting the optical axis of the camera on the object to be tracked [12]. Moreover, transition from extensive to intensive recognition is critical at the start of the tracking process. Finally, only one object becomes the subject of the recognition during the tracking process. The recognition is coupled with automatic search of the object in the predicted position in this phase.

## 2 GEOMETRIC RELATIONS

The mechatronic vision system for automatic perception and monitoring of moving objects consists of several geometric subsystems. Screen coordinates  $x_s, y_s$  of the

recognized object measured in pixels actually represent a pair of angles in respect of the camera optical axis. The distance  $z_s > 0$  of the object from the camera makes the position information complete. Let  $x_r, y_r, z_r$  be the coordinates of the rotating rectangular base with  $z_r$  identical with the optical axis of the camera. The fish-eye transformation between the extended screen coordinates SCR:  $x_s, y_s, z_s$  and the rectangular coordinates ROT:  $x_r, y_r, z_r$  rotating with camera is defined by the relations (1)

$$\begin{aligned} c &= z_s / \sqrt{1 + \tan^2(x_s) + \tan^2(y_s)} \\ x_r &= c \tan(x_s) \\ y_r &= c \tan(y_s) \\ z_r &= c \end{aligned} \quad (1)$$

and vice versa by the relations (2)

$$\begin{aligned} x_s &= \text{atan2}(x_r, z_r) \\ y_s &= \text{atan2}(y_r, z_r) \\ z_s &= \sqrt{x_r^2 + y_r^2 + z_r^2}. \end{aligned} \quad (2)$$

The orientation of the optical axis of the camera is controlled by the two degrees of freedom – azimuth and elevation in respect of the base. The left-handed rectangular coordinate system ROT is attached to the moving camera. In the case of the zero azimuth and elevation, the following equalities hold:

$$x_r = x_b, \quad y_r = z_b, \quad z_r = y_b. \quad (3)$$

The system ROT is rotated by the azimuth angle in respect of the axis  $z_b$  of the system BASE and then by the elevation angle around the axis  $x_r$  of the system ROT.

Assigning  $s_1, c_1$  the values sin and cos of the azimuth and  $s_2, c_2$  the values sin and cos of the elevation, the transformation formulas between the triples of coordinates ROT:  $x_r, y_r, z_r$  and BASE:  $x_b, y_b, z_b$  obtain the form

$$\begin{aligned} x_b &= c_1 x_r + s_1 s_2 y_r - s_1 c_2 z_r \\ y_b &= s_1 x_r - c_1 s_2 y_r + c_1 c_2 z_r \\ z_b &= c_2 y_r + s_2 z_r \end{aligned} \quad (4)$$

and vice versa

$$\begin{aligned} x_r &= c_1 x_b + s_1 y_b \\ y_r &= s_1 s_2 x_b - c_1 s_2 y_b + c_2 z_b \\ z_r &= -s_1 c_2 x_b + c_1 c_2 y_b + s_2 z_b. \end{aligned} \quad (5)$$

The screen coordinates SCR of the tracked object are transformed into the Cartesian coordinates ROT, and, using actual azimuth and elevation, into the Cartesian coordinates BASE. Finally, the history of the coordinates BASE is used for future positions prediction of the moving object.

Let us point out at this place that the missing distance  $z_s > 0$  is added by default or by the distance measuring process discussed later in Section 10.

### 3 SEGMENTATION

The aim of an effective segmentation is to separate objects from the background and to differentiate pixels having nearby values for improving the contrast. The proposed segmentation technique belongs to multilevel category. Some small regions are expected to have homogeneous characteristics (grey level) indicating that pixels belong to the same object. Nevertheless, several background levels as well as several object levels are expected. Thus, multilevel segmentation is needed for general real world images. The following procedure for 256 grey levels leading practically from 2 up to 128 brightness levels is proposed:

- Calculation of histogram  $p(k)$ ,  $k \in \langle 0, 255 \rangle$ .
- Outer limits setting  $u(0) = 0$ ,  $u(256) = 256$ .
- The total number of pixels counted and the average brightness level

$$\begin{aligned} n(128) &= \sum_j p(j); \\ u(128) &= \sum_j j * p(j); \\ u(128) &:= 1 + u(128)/n(128), \end{aligned}$$

where integer division is expected.

- The number and average level of pixels with the brightness below and above the average level.

$$\begin{aligned} n(64) &= \sum_{u(0) \leq j < u(128)} p(j); \\ if[n(64) &= 0; u(64) = u(0); \\ u(64) &= \sum_{u(0) \leq j < u(128)} jp(j), \\ u(64) &:= 1 + u(64)/n(64)] \end{aligned}$$

$$\begin{aligned} n(192) &= \sum_{u(128) \leq j < (256)} p(j); \\ if[n(192) &= 0; u(192) = u(128); \\ u(192) &= \sum_{u(128) \leq j < u(256)} jp(j), \end{aligned}$$

$$u(192) := 1 + u(192)/n(192)],$$

where  $64 = 128/2$  and  $192 = 128 + 64$ .

- In general, for pixels with brightness from  $u(i)$  to  $u(j)$  we have

$$\begin{aligned} n(l) &= \sum_{u(i) \leq k < u(j)} p(k); \\ \text{if } n(l) &= 0; u(l) = u(i); \\ u(l) &= \sum_{u(i) \leq k < u(j)} kp(k), \\ u(l) &:= 1 + u(l)/n(l)], \end{aligned}$$

where  $l$  is  $128, 128 \pm 64, 128 \pm 64 \pm 32, 128 \pm 64 \pm 32 \pm 16, 128 \pm 64 \pm 32 \pm 16 \pm 8, 128 \pm 64 \pm 32 \pm 16 \pm 8 \pm 4, 128 \pm 64 \pm 32 \pm 16 \pm 8 \pm 4 \pm 2, 128 \pm 64 \pm 32 \pm 16 \pm 8 \pm 4 \pm 2 \pm 1$ .

Thus we have 1, 2, 4, ... up to 128 average levels (255 altogether). We can generate the values  $i, j, l$  in cycles:

$$\begin{aligned} \text{outer cycle: } m &= 0, 1, \dots, M \\ &(\text{minimum } M = 0, \text{ maximum } M = 7) \\ \text{inner cycle: } i &= 0, i < 256, \text{ step } 2^{8-m} \\ &\text{and } j = i + 2^{8-m}, l = (i + j)/2. \end{aligned}$$

The value  $M$  is a degree of segmentation:

$$u(0), u(q), u(2q), \dots, u(256); \quad q = 2^{7-M}.$$

The maximal value  $M = 7$  is leading to 128 average levels in the last group of average levels. Having an image with the constant histogram, the final series of average brightness levels should be  $u(i) = i$ .

It should be pointed out at this point that the final series of average levels defines the intervals, which are closed from the left and open from the right side. Thus, the equality of the left and the right side of any interval means emptiness of such interval.

Let us consider the following simple example with  $4 \times 4$  pixels whose brightness is

$$\begin{matrix} 12, & 13, & 6, & 5 \\ 15, & 12, & 11, & 10 \\ 14, & 13, & 11, & 11 \\ 13, & 12, & 11, & 10. \end{matrix}$$

The corresponding histogram is

$k$	0	...	4	5	6	8	9	10	11	12	13	14	15	16	...	255
$p(k)$	0	...	0	1	1	0	0	2	4	3	3	1	1	0	...	0

outer limits	$u(0) = 0, \quad u(256) = 256$
number of pixels	$n(128) = 1 + 1 + 2 + 4 + 3 + 3 + 1 + 1 = 16$
average brightness level	$u(128) = 5 + 6 + 20 + 44 + 36 + 39 + 14 + 15 = 179$ $u(128) := 1 + 179/16 = 1 + 11 = 12$
number below	$n(64) = 1 + 1 + 2 + 4 = 8$
average below	$u(64) = 5 + 6 + 20 + 44 = 75$ $u(64) := 1 + 75/8 = 1 + 9 = 10$
number above	$n(192) = 3 + 3 + 1 + 1 = 8$
average above	$u(192) = 36 + 39 + 14 + 15 = 104$ $u(192) := 1 + 104/8 = 1 + 13 = 14$

For  $M = 1$  we have  $q = 64$  and the last group of average levels is  $u(64), u(192)$ . Final series of average levels  $u(64), u(128), u(192)$  together with outer limits  $u(0), u(256)$  defines the intervals, which are closed from the left and open from the right side:

$$\langle 0, 10 \rangle, \quad \langle 10, 12 \rangle, \quad \langle 12, 14 \rangle, \quad \langle 14, 256 \rangle.$$

The new brightness levels of pixels with brightness from the above intervals will be 0, 10, 12, 14, respectively. Finally, in order to improve the contrast, we can separate the final four brightness levels into the values 0, 64, 128, 192, respectively. Thus the new image will be

12, 12, 0, 0		128, 128, 0, 0
14, 12, 10, 10	and in case of contrast improving	192, 128, 64, 64
14, 12, 10, 10		192, 128, 64, 64
12, 12, 10, 10		128, 128, 64, 64.

#### 4 DIFFERENTIAL FILTRATION

Differential filtration is an extensive image information pre-processing procedure to point out all possible objects of recognition. The benefits will mainly address the following tasks:

- symmetric differential filtration aimed at detecting critical jumps of brightness level between adjacent pixels
- an efficient algorithm searching for local extremes of brightness level in rows and columns
- the identity estimation of the detected moving objects using continuously measured shape and surface texture parameters of objects
- continuous recognition of the tracked object near its predicted position without the background elimination.

The discussed mechatronic monitoring and tracking system is composed of several relatively independent subsystems. These subsystems operate in real-time with a sample period of 100 milliseconds. Some modules operate independently on the results of other modules, others work together to achieve a common goal to identify potential objects for automatic tracking.

The system works with day and night cameras oriented in the space with two degrees of freedom. In the SEARCH mode operator manually guides the optical axis of camera using a joystick. In this mode the extensive recognition of all potential objects of monitoring is performed. All detected objects are tentatively attributed by default distance. After the transformation of the position coordinates, objects are placed in a common 3D model of the outside world without specifying their identity. For new objects, pairing is performed with objects that already exist in the model from previous observations. The pairing is allocating a new object identity, i.e. identification of objects in new positions. During the extensive phase of the SEARCH mode, the system recognizes the scene with tens of objects. The system offers the operator to select from the objects around the optical axis. As soon as an object has been selected, the TRACK mode begins. An extended filtration is followed by an intensive phase of pattern recognition. In this phase the number of objects to be recognized is reduced to one.

#### **4.1 Simple Differential Filtration**

The SEARCH mode of the image processing is devoted primarily to the elimination of the changing background of the moving scene. This ensures the stability of the recognition process. It should be distinguished between the background and objects according to the dark and light alternation without direct comparison of the image with previous images. The images are in fact shifted in time, but also in space, because the cameras are not static. In many cases highlighting the potential object of tracking by simple differential filtration may work successfully. In particular, simple differential filtration of subcritical jumps of background luminance level creates sufficient conditions for a successful recognition of flying objects, especially in clouds. The output of the filtration is further processed by the algorithm of connectivity analysis discussed in the next chapter.

#### **4.2 Extended Differential Filtration**

In some cases in the SEARCH mode, but namely in the TRACK mode, the image information processing is focused on further suppress of spurious objects that remain after simple differential filtration. These are the cases of substantially different background alteration between the sky and the earth's surface on the horizon. In such cases, searching local extremes of brightness levels appears to be effective.

In comparison with the simple differential filtration, couple of opposite critical jumps of brightness level is searched. Moreover, there is limited distance requirement

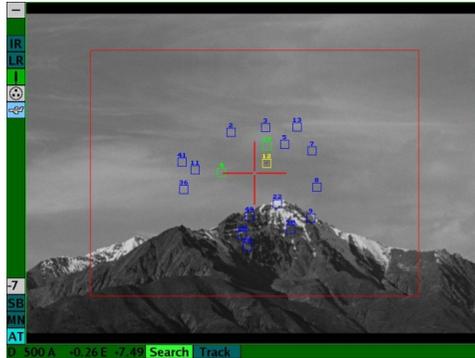


Figure 1. Unfiltered image in differential mode

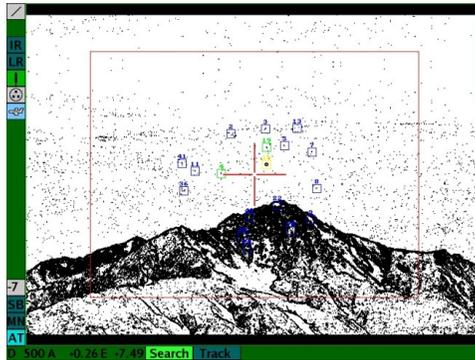


Figure 2. Filtered image in differential mode

of these jumps and coincidence of local extremes in rows and columns. Extremal differential filtration (EDF) generally leads to the suppression of clouds.

### 4.3 Intensive Recognition

In some cases in the TRACK mode, namely in complex surroundings, advanced image processing is needed. Intensive recognition with single object search is performed. The underlying approach is the search of location of the tracked moving object using continuously measured parameters of movement, shape and surface texture. The object is continuously searched near its expected location without the possibility of an elimination of the changing background on a large scale.

#### 4.4 Extremal Differential Filtration

Simple differential filtration is based on critical jumps of brightness level between the adjacent pixels of the processed image. We define critical jump simply as the jump exceeding the selected value. A set of added conditions may extend the definition. We speak about an extended differential filtration in such case. Moreover, special combinations of critical jumps may take place. In the case of filtration through a search of local extremes, we search for two consecutive critical jumps opposite to each other.

#### 4.5 EDF Algorithm

Let us introduce the following terms:

**CJ** – Critical jump

**CJD** – Critical jump down

**CJU** – Critical jump up

**RTP** – Record the Type and Position

**WTO** – Write To Output.

We localize two consecutive CJs opposite to each other in rows as well as in columns. We write the section between them into the output as black or white strip of minimum length 1 pixel. This is the case when two opposite CJs are closest to each other. The maximum length of the section in pixels may be limited by the maximum size of the object, for example, 40 pixels in rows and 30 pixels in columns. Basic background of the output will be medium gray, that is to say 128 from the interval  $\langle 0, 255 \rangle$ .

At the beginning of the line, reset of RTP is performed, that is to say it is empty. The next RTP overwrites the previous one, thus RTP is only one or none.

Overrun of the maximum lengths of the section is checked after each move of 1 pixel. The RTP reset is done after reaching a maximum without having appeared opposite CJ. The algorithm is continued as if starting a new line in such case. Once the opposite type of CJ appears before reaching the maximum, WTO is performed according to the order of types: CJD-CJU black strip, CJU-CJD white strip. At the same time, CJ is overwritten by the opposite type of CJ and the algorithm is continued. In case of no opposite CJ, the row or column remain without WTO. Conversely, if CJU and CJD occur alternately, alternation of black and white sections will occur. In case of the same type of CJ repetition, medium gray appears.

#### 4.6 Row and Column Oriented Output

In order to achieve symmetry in the sense that it will not matter whether you start with rows or columns, we finish outputs from rows and columns separately.

The resulting output is a logical product of the two outputs. Finally, we will use information from the operator on whether the object of interest is darker or paler than its immediate surroundings. According to information from the operator, grey background turns white or black by whether the object is dark or pale, respectively.



Figure 3. Unfiltered image in extended differential mode



Figure 4. Row oriented output of EDF

#### 4.7 Experimental Results

The combination of simple or extended filtration with extensive or intensive approach to automatic recognition is motivated by the authors' experience gained during the implementation of the automatic fire control system for an anti-aircraft gun [12].

In the search mode the system works extensively, i.e. recognizes the many objects that could happen in the next phase to become objects of tracking. The operator is

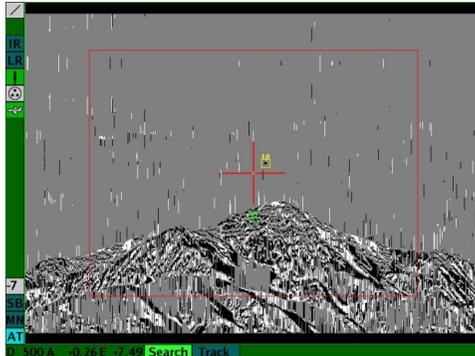


Figure 5. Column oriented output of EDF

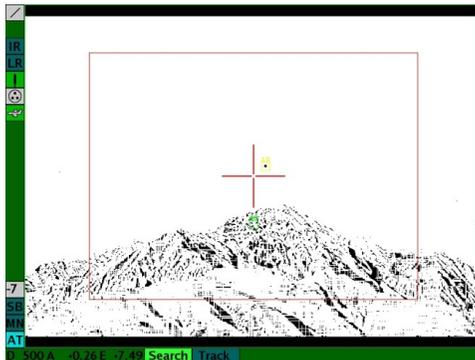


Figure 6. Filtered image in extended differential mode

selecting the object of tracking from the offer of the extensively working visual subsystem. Object-oriented extremal filtration is applied in order to suppress spurious objects at the beginning of the track mode. In this mode, extremal filtration extended by measuring the size and brightness of the tracked object takes place. The proposed extremal differential filtration offers much more reliable results in comparison with the simple one, especially in complex situations with a lot of spurious objects.

It should be pointed out that more sophisticated approach to image processing based on rough unfiltered image is needed in the most difficult circumstances. An intensive approach to recognition process may be introduced when extended filtration does not fulfill stability requirements of the tracking process. In this case a part of the unfiltered image at the selected object along with its immediate surroundings becomes a sample object.

The visual system may operate under the intense recognition when the object of interest is the only one reference object in the scene. In this mode search algorithm

for a new position of the tracked object in the relative vicinity of the predicted position is repeatedly performed. It should be noted that a portion of the object pattern is actually the changing environment acting as a disturbing factor of the recognition process.

## 5 CONNECTIVITY ANALYSIS

The goal of the connectivity analysis is to find areas with nearly the same brightness [1, 2, 3]. Our connectivity analysis algorithm is based on the definition of the row and the column noodles [11]. Noodle is a sequence of adjacent pixels with the same brightness according to the filtering mask used to process the given picture. In the proposed algorithm any number of brightness levels is acceptable. Both row and column noodles are specified in the first phase of the image processing. The shortest noodle is one pixel long and the longest one is the whole row (column). The row noodles are given names (they are numbered) consecutively and, in the second phase of the algorithm, the row noodles are joined by renaming them in accordance with the column noodles. The implemented algorithm is fully symmetric in respect of the rows and columns.

### 5.1 Passive Filtering

It is desirable to decrease the number of the resulting contiguous areas by decreasing the number of brightness levels. The aim of an effective segmentation is to separate objects from the background and to differentiate pixels having nearby values for improving the contrast. We suppose the final mask has been chosen using some multilevel segmentation method, but any other problem oriented method comes into account [9] or [7]. The same final mask resulting from the segmentation process is used to generate row and column noodles.

### 5.2 Numbering

All row noodles are numbered. The maximum number of row noodles (the worst case) is equal to the total number of pixels (1 pixel chess board). The connectivity algorithm is ready to process any picture with  $m$  rows and  $n$  columns with the maximum equal to  $m \times n$  objects. In principle, no row noodle may be crossed by the column noodle longer than 1, thus all row noodles may remain single, i.e. not joined with any other one. On the other hand, the minimum number of row noodles is  $m$ . Every new row means automatically a new row noodle with no regard on the mask. Analogously, the minimum number of the column noodles is  $n$ .

### 5.3 Joining

In general, the row noodles are crossed by the column noodles. The length of the crossing column noodle defines the number of the row noodles to be joined. Such

row noodles are given the same alias to indicate the case. Each alias used as the name generates the next alias creating the chain of aliases. Since the starting alias name is the same name, the last item in this chain of aliases is the name generating the same alias name (noodle not renamed yet). To increase the efficiency of the algorithm, a shortcut between the first and the last item in the (name, alias) – chain has been successfully implemented.

#### **5.4 Centre of Shape**

Each row noodle is a germ of the future contiguous area, i.e. an object. Data for future centre of mass calculation is initially set during the row noodle generation process. This data is finally used in the last phase of the algorithm to calculate the centre of mass of the joined row noodles.

### **6 STATIC IDENTITY ATTRIBUTES**

Each picture provides mutually disjunct areas with the same level of grey resulting from the mask used to filter single pixels. The problem is how to classify these areas according to the arbitrary criteria in problem-oriented applications.

#### **6.1 Nearness to the Centre**

We suppose the moving camera to have two rotational degrees of freedom. In the case of too many objects in the picture only the nearest 50 or 100 objects surrounding the centre of the picture may be passed to further investigation. Thus the two degrees of freedom should be used to move interesting localities into the centre.

#### **6.2 Magnitude**

The concrete problem orientation provides the minimum and maximum number of pixels of any object coming into account. The concrete numbers depend on the distance between the camera and the monitored object as well as on the total number of pixels coming from the camera in one picture.

#### **6.3 Size**

Having defined the row noodles and the column noodles in connectivity algorithm, it is easy to restrict the processing of noodles by an upper limit for the length of the row noodles and, separately, an upper limit for the length of the column noodles.

#### **6.4 Relation to the Background**

Another easily to implement feature coming from the noodle definition process is the relation between the level of grey of an object and the level of its background.

We can easily distinguish between dark objects in a pale background and vice versa.

## **6.5 Fourier Coefficients**

In the phase of the intensive recognition in the vicinity of the predicted position Fourier coefficients of the selected area of interest are calculated [4, 5, 6]. The coefficients are sophisticatedly compared with those calculated and adapted in the previous step in order to make a decision about the result of the intensive recognition process.

## **7 DYNAMIC IDENTITY ATTRIBUTES**

The dynamic identity attributes are the most sensitive ones. On one hand, positions of objects from the two consecutive pictures may be completely the same, but, on the other hand, they may be completely different. Of course, we expect nearly the same objects in nearly the same positions; but the algorithm must be ready to process any theoretically possible situation.

### **7.1 Magnitude of the Velocity Vector**

Mutual distance of the two objects from the two consecutive pictures is supposed to be the distance between the two consecutive positions of the same object. Thus, the value of this parameter means the maximal velocity magnitude, i.e. the distance in pixels per one sampling period. The concrete value of this parameter depends on the character of the application, the typical distance between the camera and the monitored objects and the duration of the sampling period.

### **7.2 Orientation of the Velocity Vector**

The monitored objects are supposed to move with an inertia in a straight line or nearly the straight line. The change of orientation is limited by an angle to eliminate objects with chaotic movements.

### **7.3 Minimal Velocity**

Some objects are static or they seem to be static because of the motion in the line towards the camera. Tracking such objects may result in delusive motion caused by the discrete pixels forming the contiguous object area. The minimal velocity value must be achieved to make the above two parameters (magnitude and orientation of the velocity vector) active.

## 8 PREDICTION

On the assumption that the motion is performed under constant acceleration, the following formula expresses the time dependence of the predicted position

$$r(t) = r + vt + at^2/2. \quad (6)$$

At least the three consecutive positions are needed for the three unknown constants  $r$ ,  $v$ ,  $a$  of the prediction formula.

Let us denote by  $r_i$  the triples of position coordinates (transformed into the BASE) measured in times  $t_i$ , so that we have the equations  $r(t_i) = r_i$ . We need at least three equations for the unknown constants  $r$ ,  $v$ ,  $a$ . Particularly, the relevant solution for  $t_1 = 2T$ ,  $t_2 = T$  and  $t_3 = 0$  is

$$r = r_3, \quad (7)$$

$$v = (1/2T)(r_1 - 4r_2 + 3r_3), \quad (8)$$

$$a = (1/T^2)(r_1 - 2r_2 + r_3). \quad (9)$$

Generally

$$D = (t_2t_3(t_3 - t_2) + t_1t_2(t_2 - t_1) + t_3t_1(t_1 - t_3))/2 \quad (10)$$

$$r = D^{-1} [t_2t_3(t_3 - t_2)r_1 + t_3t_1(t_1 - t_3)r_2 + t_1t_2(t_2 - t_1)r_3] / 2 \quad (11)$$

$$v = D^{-1} [(t_2^2 - t_3^2)r_1 + (t_3^2 - t_1^2)r_2 + (t_1^2 - t_2^2)r_3] / 2 \quad (12)$$

$$a = D^{-1} [(t_3 - t_2)r_1 + (t_1 - t_3)r_2 + (t_2 - t_1)r_3]. \quad (13)$$

From the practical point of view (noise, numerical errors), it is recommended to make much more measurements and to solve an overconstrained system of equations leading to such  $r$ ,  $v$ ,  $a$  that guarantee the minimum of the sum of squares of differences between the left and the right sides of equations. In such case we have  $n$  equations:

$$r(t_i) = r_i, \quad i = 1, 2, \dots, n, \quad (14)$$

and the following solution takes place:

$$p_k = \sum r_i t_i^k, \quad k = 0, 1, 2 \quad (15)$$

$$s_k = \sum t_i^k, \quad k = 0, 1, 2, 3, 4 \quad (16)$$

$$D = s_0s_2s_4 + 2s_1s_2s_3 - s_2^3 - s_1^2s_4 - s_0s_3^2 \quad (17)$$

$$r = D^{-1} [(s_2s_4 - s_3^2)p_0 + (s_2s_3 - s_1s_4)p_1 + (s_1s_3 - s_2^2)p_2] \quad (18)$$

$$v = D^{-1} [(s_2s_3 - s_1s_4)p_0 + (s_0s_4 - s_2^2)p_1 + (s_1s_2 - s_0s_3)p_2] \quad (19)$$

$$a = 2D^{-1} [(s_1s_3 - s_2^2)p_0 + (s_1s_2 - s_0s_3)p_1 + (s_0s_2 - s_1^2)p_2]. \quad (20)$$

## 9 TRACKING

In order to have the object near the optical axis, it is desirable to predict future positions of the moving object chosen for tracking. From the vision subsystem we have data  $x_k$ ,  $y_k$ ,  $t_k$  expressing the two mutually rectangular angular deviations from the optical axis and the moment of exposure. For the coordinates in a rotating rectangular base ROT we have

$$x_r = z_r \tan(x_k) \quad (21)$$

$$y_r = z_r \tan(y_k) \quad (22)$$

where the positive value of the coordinate  $z_r$  (the distance) comes from the distance measuring process [13]. We discuss the distance measuring later in Section 10. From the recorded values of azimuth and elevation we have the angles at the moment of exposure. The relevant BASE coordinates of the object are

$$x_b = c_1 x_r + s_1 s_2 y_r - s_1 c_2 z_r = z_r (c_1 \tan(x_k) + s_1 s_2 \tan(y_k) - s_1 c_2) \quad (23)$$

$$y_b = s_1 x_r - c_1 s_2 y_r + c_1 c_2 z_r = z_r (s_1 \tan(x_k) - c_1 s_2 \tan(y_k) + c_1 c_2) \quad (24)$$

$$z_b = c_2 y_r + s_2 z_r = z_r (c_2 \tan(y_k) + s_2). \quad (25)$$

The positive value  $z_r$  has no effect on the azimuth and elevation of the object

$$\text{azimuth} = \text{atan2}(-x_b, y_b), \quad (26)$$

$$\text{elevation} = \text{atan2}\left(z_b, \sqrt{x_b^2 + y_b^2}\right). \quad (27)$$

The history of coordinates  $(x_b, y_b, z_b)$  (pseudo coordinates when default value for  $z_r$  is used) is recorded in the memory. Having three or more positions, we can predict future positions of the object as well as the future optical axis orientation.

The time axis of the relevant events with the sampling period  $T$  is as follows:

0 – snap shot, image processing

$T$  – object coordinates, prediction calculation

$2T$  – predicted azimuth and elevation, servo positions

$3T$  – optical axis orientation.

Tracking with prediction is a process and it may take several seconds until tracking is successful. Under successful tracking we mean stable convergent process with the optical axis pointing on the moving object. Having omitted prediction, the time delay  $3T$  would cause a steady error resulting in a constant angular difference between the actual and the desired optical axis orientation.

### 10 DISTANCE MEASURING

Monitoring the object by two or more cameras offers the possibility to measure the distance of the object [13]. The triangle camera<sub>1</sub>, camera<sub>2</sub>, object is given by the constant distance of the two cameras and the two varying direction angles of the object in respect of the relevant camera.

#### 10.1 Notation

Let us assign  $a, b, c$  the three distances:

$a$  – the distance of the object from camera<sub>1</sub>

$b$  – the distance of the object from camera<sub>2</sub>

$c$  – the distance between the two cameras.

Let the relevant position vectors be  $\mathbf{a}, \mathbf{b}, \mathbf{c}$  so that

$$\mathbf{a} + \mathbf{b} + \mathbf{c} = \mathbf{0} \tag{28}$$

and let the corresponding angles be  $\alpha, \beta, \gamma$  (Figure 7).

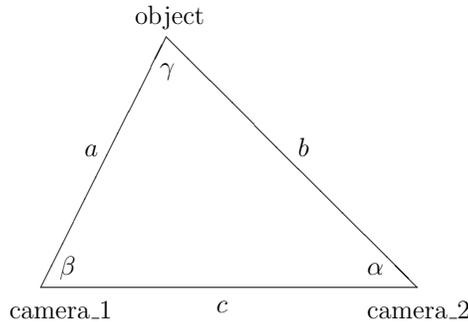


Figure 7. Triangle camera–camera–object

#### 10.2 Vector and Scalar Products

We can write for vector products

$$\mathbf{a} \times \mathbf{c} = ac \sin(\pi - \beta) \mathbf{v} = ac \sin(\beta) \mathbf{v} \tag{29}$$

$$\mathbf{c} \times \mathbf{b} = bc \sin(\pi - \alpha) \mathbf{v} = bc \sin(\alpha) \mathbf{v} \tag{30}$$

$$\mathbf{a} \times \mathbf{c} = (-\mathbf{b} - \mathbf{c}) \times \mathbf{c} = \mathbf{c} \times \mathbf{b} \tag{31}$$

where  $\mathbf{v}$  is the unit vector perpendicular to the plane of the triangle.

We can write for scalar products

$$\mathbf{a} \cdot \mathbf{c} = ac \cos(\pi - \beta) = -ac \cos(\beta) \quad (32)$$

$$\mathbf{c} \cdot \mathbf{b} = bc \cos(\pi - \alpha) = -bc \cos(\alpha) \quad (33)$$

$$\mathbf{a} \cdot \mathbf{c} = (-\mathbf{b} - \mathbf{c}) \cdot \mathbf{c} = -\mathbf{b} \cdot \mathbf{c} - c^2. \quad (34)$$

### 10.3 Solution for Distances

Joining Equations (29), (30) in respect of Equation (31) and joining Equations (32), (33) in respect of Equation (34), we have two equations for two unknown distances  $a$ ,  $b$ :

$$a \cos(\beta) + b \cos(\alpha) = c \quad (35)$$

$$a \sin(\beta) - b \sin(\alpha) = 0. \quad (36)$$

The solution of the above equations for the unknown distances  $a$ ,  $b$  of the object from cameras is:

$$a = c \sin(\alpha) / \sin(\alpha + \beta) \quad (37)$$

$$b = c \sin(\beta) / \sin(\alpha + \beta) \quad (38)$$

### 10.4 Orientation Angles

The monitored object and the two cameras create the triangle  $ABC$  with cameras in  $A$  and  $B$  and the object in  $C$ . The constant distance between the two cameras is  $c = AB$ . This constant is known from the rectification process. The two unknown distances  $a = BC$  and  $b = AC$  will be calculated from Equations (37), (38) using the online calculated angles  $\alpha$ ,  $\beta$  at  $A$  (camera.2) and  $B$  (camera.1).

Let us assign  $X()$  the orientation matrix representing the rotation around the axis  $x$  and analogously  $Y()$ ,  $Z()$ . In internal notation azimuth is the rotation around the  $z$  axis, elevation is the subsequent rotation around the new  $x$  axis, and finally the axis  $y$  is the optical axis of the relevant camera. Both  $\alpha$  and  $\beta$  represent the change of the optical axis orientation from the instant position of the tracked object to the constant position of the other camera.

We have to calculate  $\alpha$  from azimuth and elevation of the camera.2 in  $A$  looking towards the camera.1 in  $B$  :  $(a_{AB}, e_{AB})$  and looking towards the object in  $C$  :  $(a_{AC}, e_{AC})$  from the symbolic equation

$$Z(a_{AB})X(e_{AB})Y()Z(\alpha)Y() = Z(a_{AC})X(e_{AC}), \quad (39)$$

where  $Y()Z(\alpha)Y()$  is the theoretical orientation change of the camera optical axis  $y$  from the tracked object to the other camera.

Equation (39) is leading to the orientation matrix decomposition into the three rotations

$$Y()Z(\alpha)Y() = X(e_{AB})Z(-a_{AB})Z(a_{AC})X(e_{AC}), \quad (40)$$

where  $\alpha$  is the only one angle we have to calculate.

Analogously, we have to calculate  $\beta$  from azimuth and elevation of the second camera in  $B$  looking towards the first camera in  $A : (a_{BA}, e_{BA})$  and looking towards the object in  $C : (a_{BC}, e_{BC})$  from the symbolic equation

$$Z(a_{BA})X(e_{BA})Y()Z(\beta)Y() = Z(a_{BC})X(e_{BC}). \tag{41}$$

Equation (41) is leading to the orientation matrix decomposition into the three rotations

$$Y()Z(\beta)Y() = X(-e_{BA})Z(-a_{BA})Z(a_{BC})X(e_{BC}), \tag{42}$$

where  $\beta$  is the only one angle we have to calculate.

### 10.5 Orientation Matrix Decomposition

Equations (40) and (42) are analogous, thus we can analyze Equation (43) for decomposition purposes.

$$Y(u_1)Z(u_2)Y(u_3) = X(v_1)Z(v_2)X(v_3) \tag{43}$$

In this equation, we consider angles  $u_1, u_2, u_3$  to be unknown and angles  $v_1, v_2, v_3$  to be given. Moreover, from the unknown angles  $u_1, u_2, u_3$  only  $u_2$  (the change of optical axis orientation) has to be calculated.

We have the following orientation matrices in Equation (43):

$$Y(u_1) = \begin{pmatrix} \cos u_1, & 0, & -\sin u_1 \\ 0, & 1, & 0 \\ \sin u_1, & 0, & \cos u_1 \end{pmatrix} \tag{44}$$

$$Z(u_2) = \begin{pmatrix} \cos u_2, & \sin u_2, & 0 \\ -\sin u_2, & \cos u_2, & 0 \\ 0, & 0, & 1 \end{pmatrix} \tag{45}$$

$$Y(u_3) = \begin{pmatrix} \cos u_3, & 0, & -\sin u_3 \\ 0, & 1, & 0 \\ \sin u_3, & 0, & \cos u_3 \end{pmatrix} \tag{46}$$

$$X(v_1) = \begin{pmatrix} 1, & 0, & 0 \\ 0, & \cos v_1, & \sin v_1 \\ 0, & -\sin v_1, & \cos v_1 \end{pmatrix} \tag{47}$$

$$Z(v_2) = \begin{pmatrix} \cos v_2, & \sin v_2, & 0 \\ -\sin v_2, & \cos v_2, & 0 \\ 0, & 0, & 1 \end{pmatrix} \tag{48}$$

$$X(v_3) = \begin{pmatrix} 1, & 0, & 0 \\ 0, & \cos v_3, & \sin v_3 \\ 0, & -\sin v_3, & \cos v_3 \end{pmatrix}. \tag{49}$$

Having analyzed the product of matrices on both sides of Equation (43), the unknown angle  $u_2$  may be calculated from the equality of the central elements  $a_{22}$  of the resulting matrices:

$$\cos u_2 = \cos v_1 \cos v_2 \cos v_3 - \sin v_1 \sin v_3. \quad (50)$$

Since  $u_2$  is from the interval  $(0, 180)$ , we can write the equality

$$u_2 = \text{atan2} \left( \cos u_2, \sqrt{1 - \cos^2 u_2} \right) \quad (51)$$

and thus the solution of Equation (43) for  $u_2$  is given by Equation (51) where  $\cos u_2$  will be replaced by the right side of Equation (50).

Finally, we can solve Equations (40) and (42) for the unknown angles  $\alpha, \beta$  at  $A$  (camera\_2) and  $B$  (camera\_1).

### 10.6 Solution for $\alpha, \beta$

We have in case of Equation (40)

$$\begin{aligned} v_1 &= -e_{AB} \\ v_2 &= a_{AC} - a_{AB} \\ v_3 &= e_{AC} \end{aligned} \quad (52)$$

i.e. the element  $a_{22}$  in Equation (50) is

$$a_{22A} = \cos(-e_{AB}) \cos(a_{AC} - a_{AB}) \cos(e_{AC}) - \sin(-e_{AB}) \sin(e_{AC}) \quad (53)$$

and the angle  $\alpha$  at camera\_2 is

$$\alpha = \text{atan2} \left( a_{22A}, \sqrt{1 - a_{22A}^2} \right). \quad (54)$$

On the other hand, in case of Equation (42) we have

$$\begin{aligned} v_1 &= -e_{BA} \\ v_2 &= a_{BC} - a_{BA} \\ v_3 &= e_{BC} \end{aligned} \quad (55)$$

i.e. the element  $a_{22}$  in Equation (50) is

$$a_{22B} = \cos(-e_{BA}) \cos(a_{BC} - a_{BA}) \cos(e_{BC}) - \sin(-e_{BA}) \sin(e_{BC}) \quad (56)$$

and the angle  $\beta$  at the camera\_1 is

$$\beta = \text{atan2} \left( a_{22B}, \sqrt{1 - a_{22B}^2} \right). \quad (57)$$

### 10.7 Azimuth and Elevation of the Object

We considered azimuth and elevation of the tracked object to be given directly by actual positions of servos. This may not be the true, if high precision of orientation angles is needed. Tracking the object is a steady state of the control process with the time delay in the feedback. The steady error with magnitude directly proportional to the angular velocity of the tracked object around the camera should be expected. Consequently, the tracked object is not exactly on the optical axis and an additional transformation of the so called camera coordinates of the object should be performed to correct actual azimuth and elevation of the optical axis.

Let us introduce camera coordinates of an object recognized from the image:

$x_c$  – left–right angle

$y_c$  – up–down angle

$z_c$  – distance.

An object on optical axis has coordinates  $x_c = y_c = 0$ . The direct and the inverse transformations with the corresponding rectangular Cartesian coordinates of the recognized object are

$$\begin{aligned} z_r &= z_c / \sqrt{1 + \tan^2(x_c) + \tan^2(y_c)} \\ x_r &= z_r \tan(x_c) \\ y_r &= z_r \tan(y_c) \end{aligned} \quad (58)$$

$$\begin{aligned} x_c &= \operatorname{atan2}(z_r, x_r) \\ y_c &= \operatorname{atan2}(z_r, y_r) \\ z_c &= \sqrt{x_r^2 + y_r^2 + z_r^2}. \end{aligned} \quad (59)$$

Instead of actual azimuth and elevation of servos, i.e. actual optical axis orientation, we will use the camera coordinates  $x_c, y_c$  of the tracked object in the default distance  $z_c = 1000$ . From Equation (58) we get its rectangular coordinates in the rotated coordinate system.

Let us denote sin and cos values of the actual azimuth  $a$  and elevation  $e$  of servos

$$s_1 = \sin(a), \quad c_1 = \cos(a), \quad s_2 = \sin(e), \quad c_2 = \cos(e). \quad (60)$$

Using actual azimuth and elevation, we find the equivalent position of the object in the default distance

$$\begin{aligned} x_b &= x_r c_1 + y_r s_1 s_2 - z_r s_1 c_2 \\ y_b &= x_r s_1 - y_r c_1 s_2 + z_r c_1 c_2 \\ z_b &= y_r c_2 + z_r s_2. \end{aligned} \quad (61)$$

Finally, the corrected azimuth  $a_C$  and elevation  $e_C$  of the tracked object  $C$  is

$$a_C = \text{atan2}(y_b, -x_b), \quad e_C = \text{atan2}\left(\sqrt{x_b^2 + y_b^2}, z_b\right). \quad (62)$$

The analogous correction procedure has to be performed for camera\_1 and camera\_2.

## 11 CONCLUSIONS

A general concept of the perception and monitoring of moving objects with the concrete algorithmic consequences has been presented. Both forward and inverse kinematic transformations as well as operator activities and active filtering of data for final image processing have been considered. We presented the cooperation between the two subsystems having the common goal, namely automatic control subsystem with the two kinematic degrees of freedom and the vision subsystem. These two subsystems create an intelligent mechatronic solution of the practically formulated problem to catch and track a class of moving objects in a natural environment. Furthermore, the combination of an extensive and an intensive approach to the image recognition process has been presented. In respect of the previous work, activities related with various modes and detailed geometric analysis of the tracked object, online measuring of the size and the brightness has been proposed.

Finally, a general concept of the multi camera mechatronic monitoring system has been presented. In respect of the previous work activities related with various modes and the detailed geometric analysis of the tracked object, online distance measuring has been worked out. The relatively autonomous vision subsystems are working independently with the ability of the common control system to recognize tracking of the same object. In case of tracking the same object the control system is able to calculate the distance of the tracked object and thus to navigate other subsystems toward the object.

The realization output is oriented both toward the civil and the military sector and is based on the real-time kit [8]. The system is developed in cooperation with the Slovak producer of the camera head orientation servo systems.

## Acknowledgments

We thank for the support of the European Operating Program Research and Development, project ITMS code 26240220060, and of the Scientific Grant Agency of the Ministry of Education of the Slovak Republic and the Slovak Academy of Sciences, project No. 2/0096/13.

## REFERENCES

- [1] BRITAŇÁK, V.—SLOBODA, F.—TREBATICŤÝ, I.: A Modular Vision System for Inspection, Materials Handling and Assembly. Proceedings of the 3<sup>rd</sup> Scandinavian Conference on Image Analysis, Copenhagen 1983, pp. 199–204.
- [2] BRITAŇÁK, V.—KUCHTA, M.—SLOBODA, F.—TREBATICŤÝ, I.: The System Performance of the SM 54/30 Vision System. Proceedings of Artificial Intelligence and Information Control Systems of Robots, Amsterdam, North Holland 1984, pp. 101–104.
- [3] BRITAŇÁK, V.—KUCHTA, M.—SLOBODA, F.—TREBATICŤÝ, I.: Vision System as a Sensor for an Intelligent Robot. Paper presented at the IFAC Proceedings Series, Vol. 1, 1987, pp. 261–263.
- [4] BRITAŇÁK, V.: A Unified Discrete Cosine and Discrete Sine Transform Computation. Signal Processing, Vol. 43, 1995, No. 3, pp. 333–339.
- [5] BRITAŇÁK, V.: A Unified Approach to Fast Computation of Discrete Sinusoidal Transforms I: DCT and DST Transforms. Computers and Artificial Intelligence, Vol. 17, 1998, No. 6, pp. 583–607.
- [6] BRITAŇÁK, V.: A Unified Approach to Fast Computation of Discrete Sinusoidal Transforms II: DFT and DWT Transforms. Computers and Artificial Intelligence, Vol. 18, 1999, No. 1, pp. 19–36.
- [7] LIAO, P.-S.—CHEN, T.-S.—CHUNG, P.-C.: A Fast Algorithm for Multilevel Thresholding. Journal of Information Science and Engineering. Vol. 17, 2001, No. 5, pp. 713–727.
- [8] DOBROVODSKÝ, K.—ANDRIS, P.—KURDEL, P.: Computer Kit for Development, Modeling, Simulation and Animation of Mechatronic Systems. Proceedings of the 15<sup>th</sup> International Workshop on Robotics in Alpe-Adria-Danube Region, CD, Balatonfured, Hungary, June 2006.
- [9] ARORA, S.—ACHARYA, J.—VERMA, A.—PRASANTA, K.—PANIGRAHI, K.: Multilevel Thresholding for Image Segmentation Through a Fast Statistical Recursive Algorithm. Pattern Recognition Letters, Vol. 29, 2008, No. 2, pp. 119–125.
- [10] DOBROVODSKÝ, K.—ANDRIS, P.: Automatic Perception and Monitoring of Moving Objects. Proceedings of the 16<sup>th</sup> International Workshop on Robotics in Alpe-Adria-Danube Region, CD, Ljubljana, Slovenia, June 2007.
- [11] DOBROVODSKÝ, K.—ANDRIS, P.: Connectivity Analysis with Online Identity Estimation. Proceedings of the 17<sup>th</sup> International Workshop on Robotics in Alpe-Adria-Danube Region, Ancona, Italy, September 2008, CD, ISBN 978-88-903709-0-8.
- [12] DOBROVODSKÝ, K.—ANDRIS, P.: Mechatronic Vision and Fire Control System. Proceedings of the 18<sup>th</sup> International Workshop on Robotics in Alpe-Adria-Danube Region, Brasov, Romania, May 2009, CD, ISSN 2066-4745.
- [13] DOBROVODSKÝ, K.—ANDRIS, P.: Dual Camera Mechatronic Tracking System. Proceedings of the 19<sup>th</sup> International Workshop on Robotics in Alpe-Adria-Danube Region, Budapest, Hungary, June 2010, CD, ISBN 978-1-4244-6884-3.



**Karol DOBROVODSKÝ** received his M.Sc. degree in 1975 in automation and his Ph.D. degree in technical cybernetics in 1980 from the Slovak Technical University in Bratislava. He is currently a principal research worker at the Institute of Informatics of the Slovak Academy of Sciences, and a member of the Scientific Committee of the International Conference Robotics in Alpe-Adria-Danube Region. His main research interests include robot control, real-time mechatronic systems and problem oriented algorithms for image processing.



**Pavel ANDRIS** received his M.Sc. degree in numerical mathematics in 1979 and his Ph.D. degree in technical cybernetics from the Slovak Academy of Sciences in Bratislava in 1990. He is a research worker at the Institute of Informatics of the Slovak Academy of Sciences. His main areas of research include robots and other mechatronic devices control, image processing, and embedded hard real-time systems.