Journal of Automation,
Electronics
and Electrical Engineering

Identification of the Object Motion Trajectory in a Digital Image Using the Kalman Filter

Tomasz PIĄTKOWSKI

Bydgoszcz University of Science and Technology, Faculty of Mechanical Engineering,
Al. prof. S. Kaliskiego 7, 85-796 Bydgoszcz, Poland, topiat@pbs.edu.pl

DOI: https://doi.org/10.24136/jaeee.2023.009

Abstract – The paper concerns identification of the object motion trajectories in a digital image. An image analysis procedure was used consisting of two methods: BLOB identification and Kalman filter. BLOB identification is the superior method, and the Kalman filter is an alternative one, used to determine the motion trajectories only in the event of incorrect BLOB identification. Data from the correct BLOB identification are used on an ongoing basis in the process of training the Kalman filter. In this way, the filter becomes a model of a physical object that allows predicting the future state based on the object's current behaviour. The use of two complementary image analysis methods allows to determine the motion trajectories of the object's characteristic points in conditions of uneven scene lighting. The effectiveness of the method was confirmed in experimental rests.

Key words – camera, BLOB, experiment, digital image, Matlab

Introduction

Numerical image analysis has many practical applications, e.g. in the detection of surface imperfections in industrial product quality control, creation of digital geographic maps, robotics, medicine, and transport [1], [2]. Images are generated by cameras: photographic, ultrasonic, X-ray and other imaging devices. Computerized vision systems support or replace people in many processes of analysis and recognition of specific classes of images, such as: fingerprints in access security systems, banknotes in vending machines. Vision systems expand the perception range of our sight sense [2]. They enable observations of objects in the infrared, ultraviolet and X-ray ranges. They also allow to notice differences that are undetectable through direct observation. They are perfect for mass analysis and classification of millions of images, based on numerical criteria, using devices that are not subject to emotions or fatigue [3]-[9].

The essential feature of computer image analysis is the non-invasive (non-contact) nature of examining objects and physical phenomena. This feature means that the examination does not interfere with the properties of the object and does not disturb its functioning. When using a highspeed camera, the method is also suitable for recording and analysing rapidly changing processes, e.g. automatic sorting or positioning of unit loads [10]-[12].

Nowadays, there are programs supporting engineering calculations, e.g. Matlab, Scilab, Python [13], [14], which contain libraries with functions for performing basic image analyses. Examples of solutions proposed in these environments concern simple systems presented in idealized conditions. They do not include, for example, algorithms for simultaneously tracking the motion trajectories of several selected objects [10] or identifying these paths in conditions of uneven scene lighting. The general resources of these programs do not provide a ready-made target solution, but they constitute an important starting point, giving the opportunity to develop own application taking into account contemporary trends in the field of interest.

Identification of a moving object moving relative to a stationary scene is typically solved by comparing the differences between two adjacent frames. Then, the pixels constituting the difference are subjected to BLOB detection (Binary Large OBject detection, Fig. 3b, Fig. 4b) [15], [16]. This detection involves detecting an region of a digital image whose pixels have common, previously defined features, e.g. brightness, shape or colour, different from the rest of the image background.

Another way to identify moving objects is to use deep learning algorithms (YOLO, SSD, ACF) using neural networks. This technique requires the prior selection of objects to be recognized in the image and the process of training and testing the neural network. This is a time-consuming process, but it allows you to recognize, for example, faces, people and other objects.

BLOB detection is a high-throughput technique. It is suitable for combination with frame colour thresholding, enabling the selection of objects with a previously defined colour - preferably contrasting with the background. However, this solution is susceptible to the quality of scene lighting, which may cause unintended changes in colour tones and incorrect BLOB analysis.

The aim of the presented research is to develop and experimentally verify a method for digital identification of object motion trajectories in the process of positioning with an oblique barrier on a belt of conveyor (Fig. 2). In this process, the object, hitting the stationary barrier, moves relative to the belt of conveyor. Therefore, there are two moving bodies: a belt of conveyor and a positioned object dislocating relative to the belt. For this reason, the technique of identifying the movement of an object by comparing two adjacent frames is unreliable. The research used BLOB detection with frame colour thresholding, supplemented with a Kalman filter to become independent from scene lighting disturbances. The Kalman filter is usually used in control systems, e.g. in robotics, aviation, and space [17]. This filter allows to predict the future state of a physical object, based on knowledge about its current state. For this reason, the Kalman filter in the image analysis method should support BLOB detection when the lighting quality is insufficient. The proposed method for identifying the object's trajectory, its experimental verification and the obtained research results are presented in the following sections.

1 PROCEDURE FOR DETERMINING THE MOTION TRAJECTORY OF THE CUBOID CORNERS

The flowchart of the proposed image analysis procedure is shown in Fig. 1. The conditional expressions and instructions follows the convention used in the Matlab environment in which this algorithm was implemented. Two Matlab libraries: *Image Processing* and *Computer Vision* were used, from which two functions were taken: *BlobAnalysis* and *configureKalmanFilter*, which are crucial from the point of view of the program's operation.

Before starting the analysis procedure of all image frames, first should be indicated manually the *xy* coordinates of the centres of characteristic points *P*, represented by red rectangular markers on the first frame (Fig. 2a). These markers are distinguished with a reference (4) and are placed on two opposite corners of the tracked object. The red colour was chosen because it was assumed that it would be unique enough in the frame to facilitate digital identification of the object's corners. The variable *xy* is a four-element row vector:

$$xy=[x_1, y_1, x_2, y_2]; \tag{1}$$

where:

x_1, y_1, x_2, y_2 – coordinates of the marker centres of the corners P_1 and P_2 (Fig. 2a).

The algorithm also allows to track a larger number of markers at the same time. This is determined by the number *k* of initial points entered into the *xy* variable. The number *k* of tracked points is given by the expression:

$$k=length(xy)/2; \tag{2}$$

To narrow down the search regions for markers in frames, these markers are contained in square regions (ROI - Region Of Interest, marked with reference (5) in Fig. 2a). The *H* edge of the ROI square should be as small as possible, but at the same time large enough so that in the next frame, the characteristic corner marker fits entirely within the ROI, despite the movement of the tracked object. The centre of the ROI in the current frame coincides with the centre of the marker detected in the previous frame. The corners of the ROI are described in coordinates represented by the rows and columns of the pixel matrix in the frame:

$$ROI=frame(yI-H:yI+H,xI-H:xI+H); \tag{3}$$

where:

xI=floor(xL); *yI=floor(yL)*; – coordinates of the centre of the *P* marker, as integers indicating the column and row numbers in the pixel matrix in the previous frame,

xL=xy(2j-1); *yL=xy(2j)*; – coordinates of the centres of the *P* markers in the previous frame, as real numbers,

j=1, 2,..,k – index of point *P*,

2H+1 – length of the side of the ROI square.

Too large an ROI dimension is not desirable because it increases the frame processing time and may increase the probability of containing hues of the frame with colours similar to the colour of the marker, which may make it difficult to correctly recognize the marker coordinates (4) – Fig. 2a.

The basic process identifying the geometric centres of points *P* is carried out by function *BlobAnalysis*, which is initiated just before searching the frames:

$$BA=vision.BlobAnalysis(); \tag{4}$$

When analysing ROI regions, the BA parameter is called with an argument, just like the function:

$$[F,S]=BA(BW); \tag{5}$$

where:

F – blob area in pixels,

S – coordinates of the geometric centres of the blobs,

BW – ROI binary image (black and white).

The *BW* binary image is obtained by thresholding the grey shades of the image pixels GREY with the *a* parameter:

$$BW=GREY\geq a; \tag{6}$$

The GREY image is created by converting of the RGB format of the ROI:

$$GREY=rgb2gray(ROI); \tag{7}$$

The threshold value *a* is defined by the relationship:

$$a=c(j)*f; \tag{8}$$

where:

f – threshold factor, assumed in the paper *f*=0.85,

c(j) – shade of grey of the pixel located in the geometric centre of point *P(j)*, read from the previous frame:

$$c(j)=GREY(floor(y\text{-}yL+H), floor(x\text{-}xL+H)); \tag{9}$$

The shade of gray *c(j)* stored in eight bits has a value in the range 0-255, where 0 corresponds to black and 255 to white. Conversion of a GREY image, in shades of grey, to a black and white *BW* image, performed according to Eq. (6), will assign pixels smaller than *a* (Eq. (8)) the value 0 (black), and in the opposite case – the value 1, i.e. white, interpreted as the corner marker *P*. Using the factor *f*<1 (e.g. *f*=0.85) expands the range of GRAY image pixels that are interpreted as the *P* marker. In this way, the variation in the colour shades of registered marker in the image, caused by the change in the position of this marker in relation to the lighting and camera in individual frames, is taken into account.

The effect of the *BA* function (Eq. (5)), which searches for regions of white pixels (so-called blobs) in the BW image, are the areas *F* and the centres' coordinates *S* of these regions.

The possibility of searching for multiple blobs generates the problem of assigning the blob to the *P* marker of the object corner. It was assumed that the blob whose geometric centre is the closest to the centre of the ROI represents the marker *P*. For this purpose, the distances *R* of the geometric centers of the blobs from the centre of the ROI are calculated:

$$R=sqrt((H+1\text{-}S(:,1)).\hat{}2)+(H+1\text{-}S(:,2)).\hat{}2)); \tag{10}$$

Then, the minimum distance *R* is searched, to which the *iF* index of the blob is associated. This index allows to read the surface area *F* of the blob representing the marker *P*:

$$[\sim,iF]=min(R); F=F(iF); \tag{11}$$

and the *S* coordinates of this blob in the frame:

$$x=S(iF,1)+xL\text{-}H; y=S(iF,2)+yL\text{-}H; \tag{12}$$

In case of frames in which *P* markers are found in over-illuminated regions (Fig. 2b) or under-exposed regions (Fig. 2c) where the markers are difficult to distinguish from the background, the blobs found by the *BA* function (Eq.(5)) may not represent a *P* marker, as e.g. in Fig. 3 and Fig. 4. There may be blobs with too large surface areas (Fig. 3, Fig. 4) or no blobs at all. At the same time, before starting the frame search process, we can determine the maximum number *Fmax* of pixels that can be assigned to the *P* marker. It is assumed that the blob indicates the *P* marker if the condition is met:

$$\text{if } Fmax>F>0 \tag{13}$$

If this condition is met, the coordinates of point *P(j)* (determined in Eq. (12)) are entered into the Kalman filter (presented later in the work) as next training data:

$$predict(K\{j\}); \tag{14}$$

$$TK=correct(K\{j\},[x,y]); \tag{15}$$

These data are also used to read the shade of grey *c(j)* of the marker *P(j)* (according to Eq.(9)) and to update the *xy* vector containing the coordinates of points *P* located on the frame *i*:

$$xy(2j-1:2j)=[x,y]; \tag{16}$$

At this stage, the index *j* of point *P* is incremented:

$$j=j+1; \tag{17}$$

If the condition is met:

$$\text{if } j \le k \tag{18}$$

then the another marker *P* in the current frame exists, for which the process of determining the coordinates of its centre must be performed again. Otherwise, all *P* points located on frame *i* have already been found. Then we need to add the *xy* vector to the matrix *T*, which collects the coordinates of points *P* from all frames:

$$T=[T;xy]; \tag{19}$$

and increment the index *i* of frame:

$$i=i+1; \tag{20}$$

to repeat the search process for a new frame:

$$frame=read(M,i); \tag{21}$$

starting from point *P* with index:

$$j=1; \tag{22}$$

The frame analysis process ends when the condition:

$$\text{if } i \le n \tag{23}$$

is no longer met. Then a matrix *T* is generated, containing the motion trajectories of the markers *P*, ending the image analysis procedure.

An important support for image analysis using the BLOB method is the use of the Kalman filter. The purpose of this filter is to determine the coordinates of the *P* marker when the BLOB method is ineffective, i.e. when the Eq. (13) condition is not met. However, it should be emphasized that the Kalman filter cannot replace the BLOB method, but can only state its sporadic supplement.

The initiation of the Kalman filter is performed during the analysis of the first frame, individually for each tracked marker *P(j)*, when the condition is met:

$$if\ i = 1 \tag{24}$$

First, a Kalman filter object *K{j}* is created for point *P* with index *j*:

K{j}=configureKalmanFilter(A,[x,y],B,[xK,vK],C); (25)

where:

A='ConstantVelocity' – a parameter specifying the type of adopted motion model to track markers *P*; an alternative traffic model is *'ConstantAcceleration'*,

*[*x,y] – initial location of *P(j)* marker,

*B=1E7*ones(1,2)* – initial estimation error,

[xK,vK] – motion noise,

xK=5 – motion noise concerning location,

vK=0.5 – motion noise concerning velocity,

C=100 – measurement noise.

The Kalman filter parameters were adopted based on examples available in the Matlab environment (Matlab R2023 help centre) and on the basis of our own preliminary research.

Then, the first training data, i.e. the initial *xL* and *yL* coordinates of point *P(j)*, are introduced into the newly created object *K{j}*, according to Eq. (15).

In case of the first frame, the *xL* and *yL* coordinates are simultaneously treated as the *x* and *y* coordinates of the current frame:

x=xL; y=yL; (26)

which are used to read the shades of grey *c(j)* of point *P(j)*, according to Eq. (9).

The intervention of the Kalman filter in determining the coordinates of points *P* manifests itself during the analysis of subsequent frames (at *i*>1), when the BLOB method fails, i.e. when the condition Eq. (13) is not met. Then, the Kalman filter, based on the training data previously provided to it, determines the coordinates of the centre of the marker *P(j)*:

TK=predict(K{j}); (27)

x=TK(1); y=TK(2); (28)

The further process of searching for the coordinates of points *P* proceeds according to the previously presented scheme relating to the BLOB method.

In order to improve the practical implementation of the method of tracking the trajectory of *P* markers in image, the structure of this method has been interpreted graphically in the flowchart presented in Fig. 1. This flowchart contains references to logical relationships and instructions, the meaning of which is explained in detail in the text above.

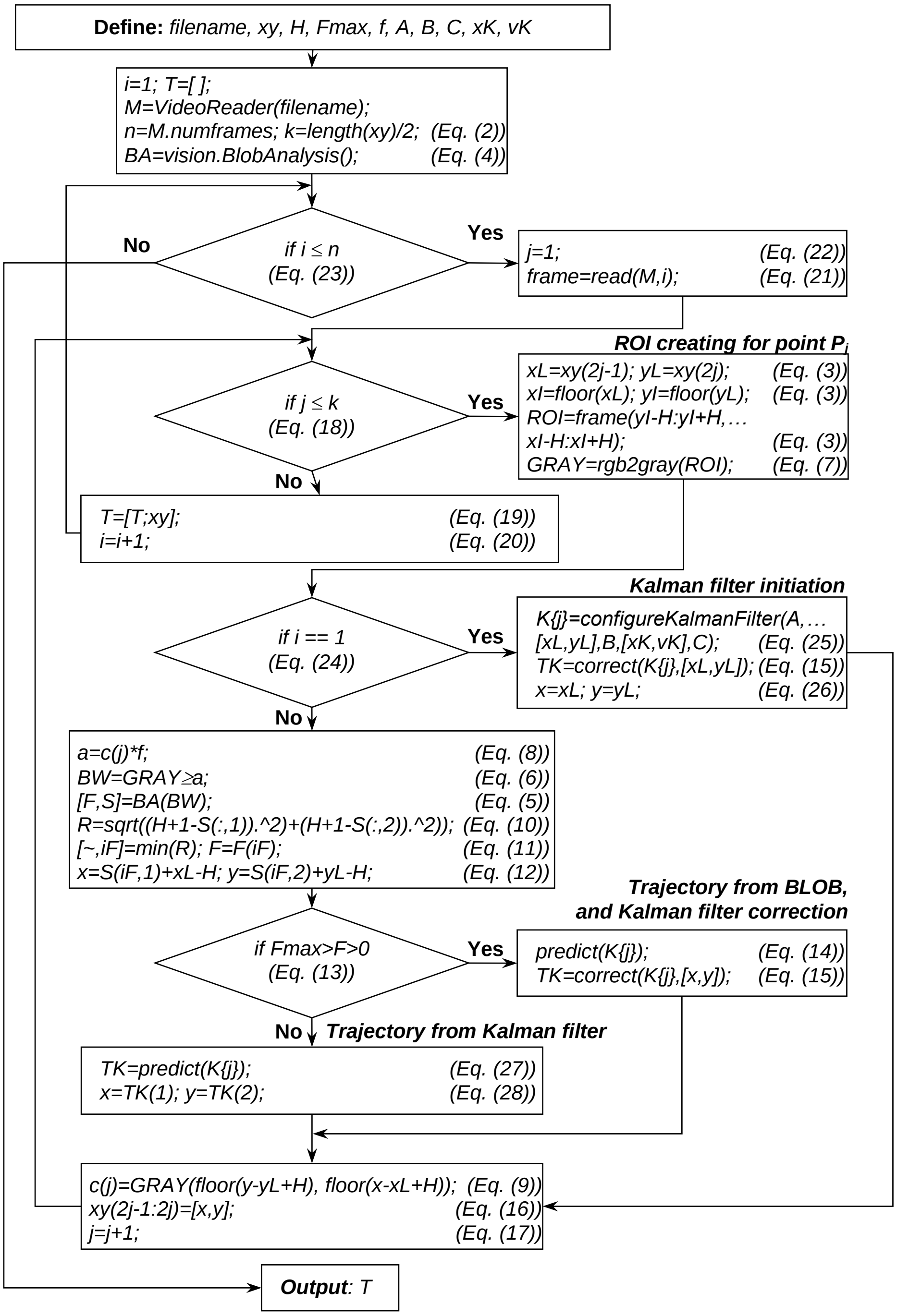
Define: filename, xy, H, Fmax, f, A, B, C, xK, vK
i=1; T=[ ];
M=VideoReader(filename);
n=M.numframes; k=length(xy)/2; (Eq. (2))
BA=vision.BlobAnalysis(); (Eq. (4))
No
if i ≤ n
(Eq. (23))
Yes
j=1; (Eq. (22))
frame=read(M,i); (Eq. (21))
ROI creating for point Pj
xL=xy(2j-1); yL=xy(2j); (Eq. (3))
xI=floor(xL); yI=floor(yL); (Eq. (3))
ROI=frame(yI-H:yI+H,…
xI-H:xI+H); (Eq. (3))
GRAY=rgb2gray(ROI); (Eq. (7))
if j ≤ k
(Eq. (18))
Yes
No
T=[T;xy]; (Eq. (19))
i=i+1; (Eq. (20))
Kalman filter initiation
K{j}=configureKalmanFilter(A,…
[xL,yL],B,[xK,vK],C); (Eq. (25))
TK=correct(K{j},[xL,yL]); (Eq. (15))
x=xL; y=yL; (Eq. (26))
if i == 1
(Eq. (24))
Yes
No
a=c(j)*f; (Eq. (8))
BW=GRAY≥a; (Eq. (6))
[F,S]=BA(BW); (Eq. (5))
R=sqrt((H+1-S(:,1)).^2)+(H+1-S(:,2)).^2)); (Eq. (10))
[~,iF]=min(R); F=F(iF); (Eq. (11))
x=S(iF,1)+xL-H; y=S(iF,2)+yL-H; (Eq. (12))
Trajectory from BLOB,
and Kalman filter correction
if Fmax>F>0
(Eq. (13))
Yes
predict(K{j}); (Eq. (14))
TK=correct(K{j},[x,y]); (Eq. (15))
No
Trajectory from Kalman filter
TK=predict(K{j}); (Eq. (27))
x=TK(1); y=TK(2); (Eq. (28))
c(j)=GRAY(floor(y-yL+H), floor(x-xL+H)); (Eq. (9))
xy(2j-1:2j)=[x,y]; (Eq. (16))
j=j+1; (Eq. (17))
Output: T

Fig. 1. Scheme for identifying the motion trajectories of P markers in an image

2 EXAMPLE OF APPLICATION OF THE PROCEDURE FOR SEARCHING CHARACTERISTIC POINTS IN AN IMAGE

The procedure of characteristic points tracking in a digital image was carried out on the example of the process of positioning a cuboid object with oblique barrier – Fig. 2. This barrier, marked in Fig. 2 with the reference (2), is placed above the surface of the belt of conveyor (1). The object (3) hits the barrier, rotates, moves along the barrier, and then, when lowering the barrier, positions parallel to the edge of the conveyor. This process has practical applications, i.e. in logistics centres when performing automatic picking of transport batches.

The aim of the positioning process is to position object parallel to the conveyor axis from any initial location, with the assumed precision in the shortest possible time and in the smallest working space, without damaging the objects. Achieving this goal requires proper configuration of the design and operational parameters, which are: velocity and width of the conveyor, frictional properties of the barrier and the conveyor, angle and length of the barrier, geometric and mechanical properties of the positioned objects. The proper selection of such a large number of parameters is most effectively carried out by numerical optimization. The main element of optimization are physical, mathematical and numerical models of the positioning process, which should be calibrated and verified in experimental tests.

In this research, the most important issue is to obtain the motion trajectories of the positioned object, which will then allow to determine its velocity, acceleration and reaction forces. A natural way to register the motion of an object, without disturbing its course, is video recording.

During the experimental tests, a digital camera was placed in a tripod above the conveyor so as to cover the entire working space in which the positioning process is carried out (Fig. 2a). It was assumed that the translational and rotational motion of the positioned object in the conveyor plane will be identifiable if the motion trajectories of two characteristic points (4), located along the diagonal, at opposite corners of the tracked object, are recorded. The markers (4) were placed on the upper wall of the object, covered with black paper, to increase the contrast of the markers against the background of the image. One of the critical factors of video recording is scene lighting. This lighting can be properly configured if image recording is performed in a studio or laboratory. In case of an active production hall, the introduction of laboratory lighting is problematic because it generates limitations and disruptions in the company's current economic processes. The least invasive solution is to make video recordings using standard lighting in the production hall, without any prior preparations. However, then there is a high probability of uneven lighting occurrence (Fig. 2), i.e. parts of the scene with excessive lighting, mirror reflections of the lighting on the recorded object (Fig. 2b) or poorly lit fragments (Fig. 2c). These imperfections mean that the method of finding characteristic points must be resistant to them. It is best when the method is equipped with alternative ways of image analysis that complement each other, e.g. the BLOB function and the Kalman filter – as proposed in the presented work.

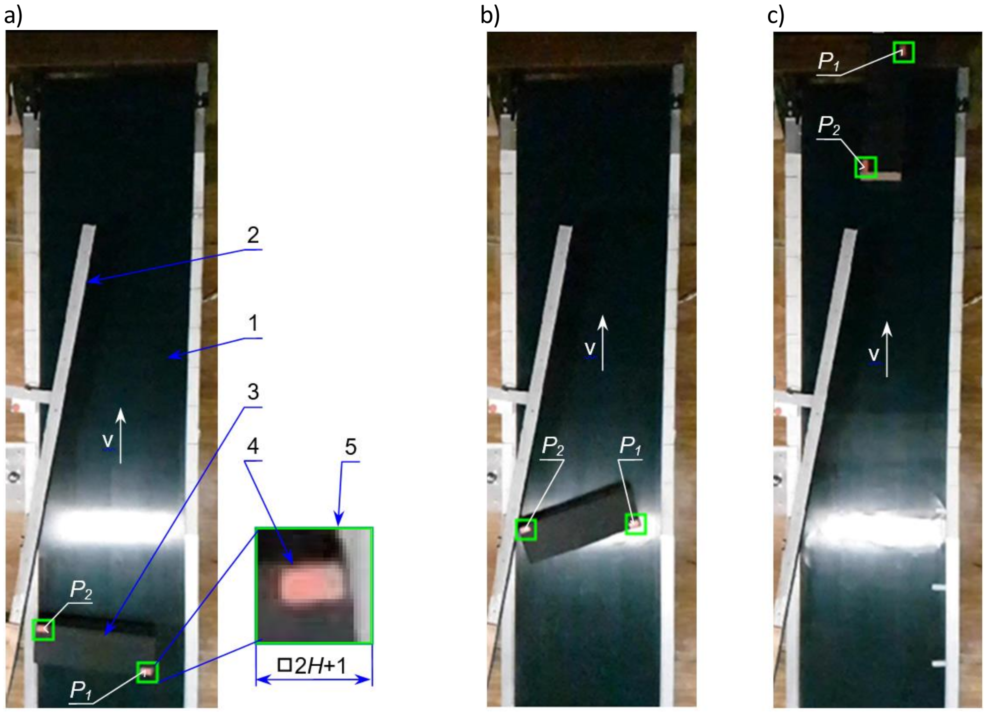

Fig. 2. A cuboid object transported on a belt of conveyor with oblique barrier: a) initial position of the object, before coming into contact with the barrier – the first frame of the video, b) position of the object in the most illuminated zone, c) final position of the object, after losing contact with a barrier – the last frame of the video; 1 – belt of conveyor, 2 – oblique barrier, 3 – transported object, 4 – rectangular marker (red) at the corner of the object, 5 – ROI (Region Of Interest, a square with a green frame in which the marker (4) is searched), *P_1* and *P_2* – corners of the object included in the ROI, *2H*+1 – side of the square in px

In the recorded image (Fig. 2), digital tracking of the motion trajectory encounters two problematic positions of the object, shown in Fig. 2b and Fig. 2c. The first position concerns the situation when the object's corner markers are located in an over-lighted area of the image Fig. 2b, Fig. 3), and the second one – when the markers are in an under-lighted zone, and additionally the side wall of the object is visible, with a colour similar to the marker (Fig. 2c, Fig. 4).

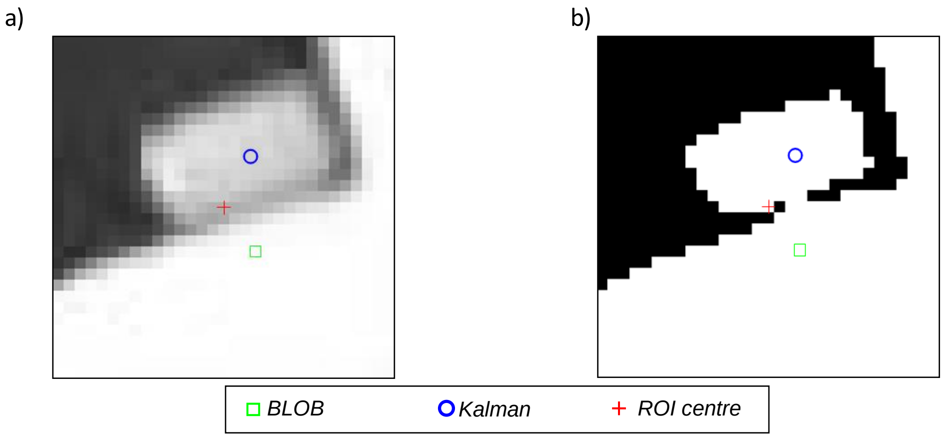

Fig. 3. Identification of the corner P_1 according to the frame from Fig. 2b, located in the most illuminated zone: a) ROI area in grey, a) ROI area in black and white

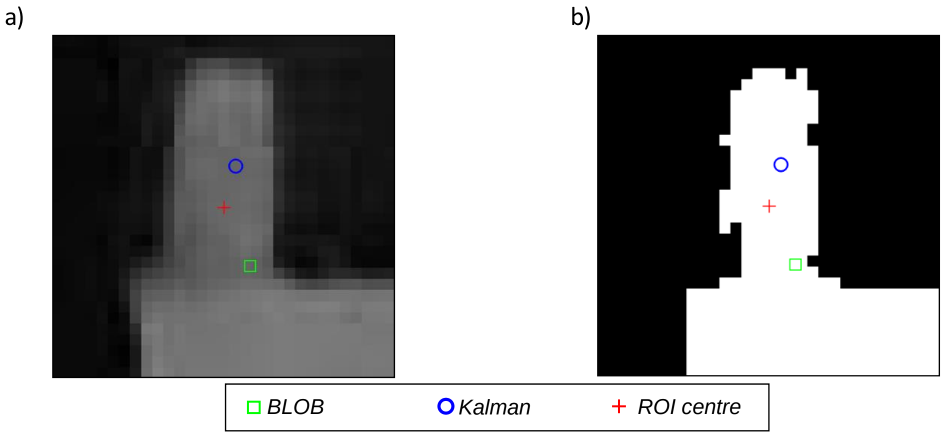

Fig. 4. Identification of the corner P_2 according to the frame from Fig. 2c, in the final position of the object, after losing contact with the barrier: a) ROI area in grey, b) ROI area in black and white

Both situations in BLOB analysis give the same effect – the marker interpenetrates with the surroundings (Fig. 2b-Fig. 3 and Fig. 2c-Fig. 4). This case is recognized by the instruction in Eq. (13), which detects a disproportionately large blob surface (white area) relative to the actual marker (4). The geometric centres of such blobs, which are marked with green square markers in Fig. 3 and Fig. 4, are obviously incorrect – they do not represent points *P*. Then the Kalman filter proposal is used, i.e. the coordinates marked in Fig. 3 and Fig. 4 by circular blue markers. In the remaining figures (Fig. 5 and Fig. 7), the points of the motion trajectories determined by the Kalman filter are also marked with round markers, but in red.

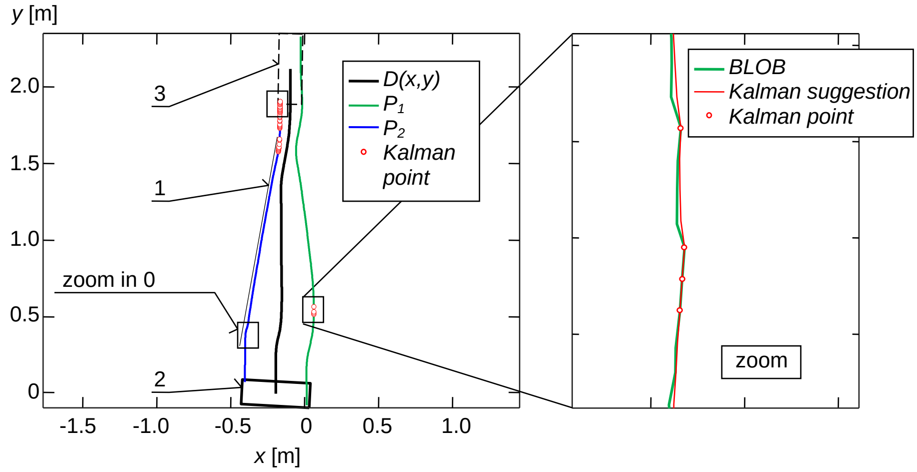

Fig. 5. Motion trajectories of the object's corners; 1 – oblique barrier, 2 – initial position of the object, 3 – final position of the object, *D(x,y)* – trajectory of the geometric centre of the object, P_1 and P_2 – corners of the object, BLOB – trajectory determined on the basis of the BLOB method, Kalman point – trajectory determined on the basis of the Kalman filter ; [*a*,*b*,*c*]=[0.46, 0.16, 0.1] m – dimensions of the object, *m*=2.12 kg – mass of the object, *v*=1.59 m/s, α=10.66o – barrier inclination angle, *L*=1.703 m – length barrier, FrameRate=240 frame/s, Resolution=[1280,720] px

To determine the actual dimensions of objects and the geometry of motion trajectories recorded in the digital image, a scaling method was used, presented, among others in work [11]. The result of using this method and the procedure for searching for *P* markers of object is shown in Fig. 5. The motion trajectories of characteristic points P_1 and P_2 were obtained based on the image analysis procedure, and the *D* trajectory of the object's centre of geometry was obtained based on the geometric relationships between these points. The motion trajectories of points P_1 and P_2 are determined mainly using the BLOB function. A Kalman filter constantly accompanies this function. The coordinates of points P_1 and P_2, determined by the BLOB function, are also the training data for the Kalman filter. This filter, in response, suggests the next locations of the *P* points in subsequent slides, as shown in the enlarged fragment of Fig. 5 and Fig. 6. In the training process, the Kalman filter becomes a model of the physical object. The properties of the Kalman filter are controlled by configuration parameters. One of the most important parameters is *'MotionNoise' vK*. The influence of this parameter on the motion trajectory determined by the Kalman filter is shown in Fig. 6. The higher the value of this parameter, the greater the loss of the Kalman filter's ability to remove noise from the prediction of the coordinates of point *P*. The lack of noise removal by the Kalman filter comes down to copying the trajectory determined by the BLOB function (Fig. 6c). Then, in case of stand-alone prediction of Kalman filter, the coordinates of the motion trajectory are incorrect, i.e. they are burdened with noise that has not been removed in the learning process. The lack of the ability to remove noise causes the Kalman filter to become an ordinary recorder and not a model representing a physical object. When the noise from the data provided by the BLOB procedure is eliminated, the motion path determined by the Kalman filter is smoother (Fig. 6a,b). For the needs of the presented work, the parameter *vK*=0.5 was adopted.

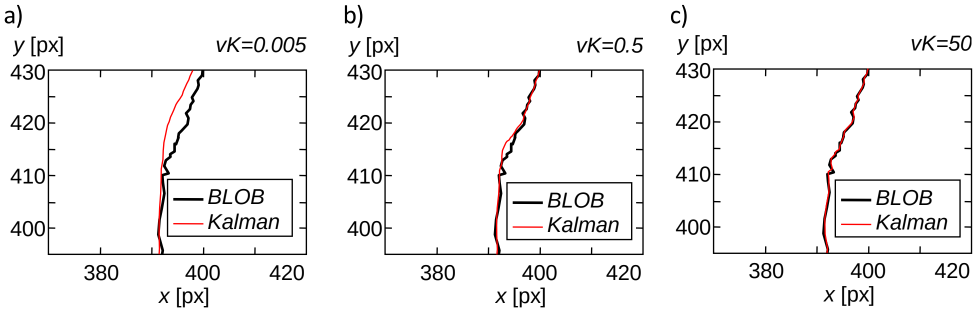

Fig. 6. Zoom from Fig. 5 – influence of the *'MotionNoise' vk* parameter on the shape of the object's motion trajectory suggested by the Kalman filter when: a) *vK*=0.005, a) *vK*=0.5, c) *vK*=50

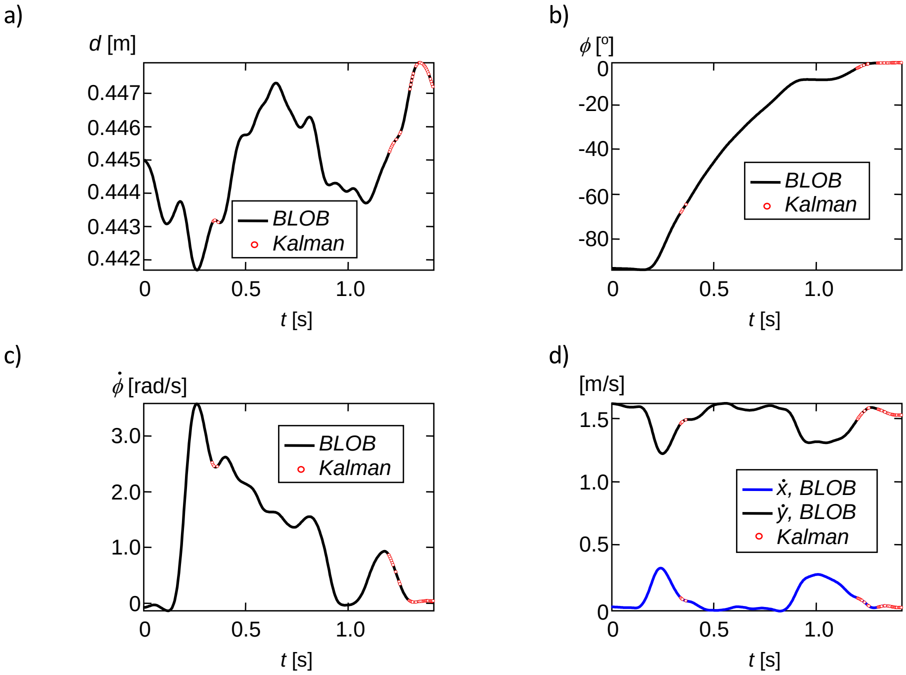

Fig. 7. Analysis charts of the object's motion trajectory: a) object's diagonal (distance between the corners P_1 and P_2), b) object's rotation angle, c) object's angular velocity, d) velocity components of the object's centre of geometry

The Kalman filter available in Matlab can operate in two operating modes: *'ConstantVelocity'* or *'ConstantAcceleration'*. Both modes are linear models. Hence, for a non-linear system such as the object positioning process, the Kalman filter should be constantly corrected with training data from the BLOB function. Moreover, the Kalman filter should be used to determine the coordinates of points *P* only incidentally, e.g. as in Fig. 5 – without long-term continuous operation.

The *'ConstantVelocity'* mode was used in the research due to the constant velocity *v* of the conveyor, even though a significant part of the positioning process is in a state of unsteady motion. The adopted operating mode has proven successful, especially in predicting the final section of the motion trajectory of point P_2, invisible to the BLOB function, when the object has lost contact with the barrier and starts moving at the conveyor velocity *v* (Fig. 5).There is also the so-called extended Kalman filter, intended for nonlinear real-time applications. As a consequence, however, it is characterized by a much larger number of configuration parameters. For this reason, a simpler, basic version of the Kalman filter was used in the presented research.

Fig. 7a shows a graph of the diagonal length of the object, the value of which changes by only 0.006 m throughout the recorded image. The obtained result should be considered satisfactory, especially if we take into account that the image frame covers the working space of the conveyor with a length exceeding 2 m (Fig. 5). The diagram of the angle ϕ of rotation of the object, from Fig. 7b, allows us to recreate the course of the positioning process and its final effect. After coming into contact with the object, the object begins to perform an intense rotation, which is confirmed by a sudden increase in angular velocity (Fig. 7c) and a decrease in the translational velocity towards the *y* axis and an increase in this velocity towards the *x* axis Fig. 7d. This stage lasts until the object is placed with its longer side along the barrier, which is represented by a horizontal section of trajectory ϕ (Fig. 7b) achieved in approx. 1 s. Then, the object leaves the barrier, and is rotated by the free end of the barrier, until the object is positioned parallel relative to the edge of the conveyor, i.e. the angle ϕ=0°, represented by the horizontal line of the final trajectory section ϕ in Fig. 7b.

Fig. 8 shows changes in the shades of grey *c(j)* of the tracked object markers in individual frames.

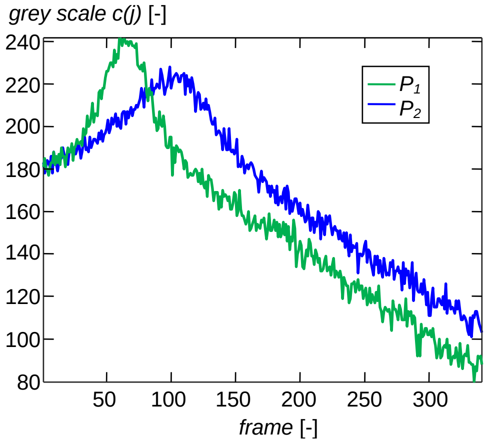

Fig. 8. The shade of grey *c(j)* of pixels identifying points P_1 and P_2 in an 8-bit greyscale, represented by a numerical range of 0-255, where 0 means black and 255 - white

These changes are mainly caused by uneven lighting in the scene, but also by changes in the position of the moving object relative to the camera and the penetration of background colours and markers. Fig. 8 shows that the process of identifying marker coordinates is carried out in very difficult conditions. The key to meeting these conditions is to use a procedure in which the starting point for the analysis of the marker colour in the current ROI is the marker colour from the previous ROI and to use two alternative image analysis methods: BLOB and the Kalman filter, which complement each other.

3 CONCLUSIONS

A procedure for digital identification of the motion trajectory of a cuboid object marked with coloured markers that moves relative to the belt of conveyor is proposed. The algorithm allows to track from one to many markers simultaneously. It was assumed that the procedure should take into account that it will be used to process digital images recorded in an industrial hall with standard lighting, so as not to disturb the economic functioning of the enterprise. These conditions mean that the lighting of the scene, especially of large dimensions, will be uneven. Then there may be areas of excessive or low lighting. To meet these conditions, it is necessary to propose an algorithm for analysing object motion trajectory that will be resistant to uneven lighting. For this reason, a solution was proposed that uses two alternative and complementary image analysis methods: BLOB identification and the Kalman filter. The leading method is BLOB identification, and the Kalman filter is a supplement, used only for incidental determination of the motion trajectory when BLOB identification is unreliable. The BLOB identification operation is constantly accompanied by a Kalman filter. The coordinates of the object's motion trajectory determined by the BLOB function are also the training data for the Kalman filter. In the learning process, the Kalman filter becomes a model of a physical object that allows to suggest the future state of the object based on its current state. To facilitate the practical implementation of the proposed image analysis procedure, it is presented graphically in the form of a flowchart, and its functional elements are also explained in the text of the paper. The procedure was verified in experimental tests, which confirmed the effectiveness of identifying the object's motion trajectory in conditions of uneven lighting.

REFERENCES

[1] Dalka P., Metody algorytmicznej analizy obrazu wizyjnego do zastosowań w monitorowaniu ruchu drogowego, Praca doktorska, Politechnika Gdańska, 2014.

[2] Rudnicki Z, Metody komputerowej analizy obrazów w badaniach tribologicznych, Wydawnictwo AGH, Kraków 2010.

[3] Guilluy W., Oudre L., Beghdadi A. (2021) “Video stabilization: Overview, challenges and perspectives”. Signal Processing: Image Communication, 90.

[4] Lee T.H., Lee Y., Song B. C. (2014) “Fast 3D video stabilization using ROI-based warping”. Journal of Visual Communication and Image Representation, 25, 943-950.

[5] Wang Y., Huang Q., Jiang C., Liu J., Shang M., Miao Z., (2023) “Video stabilization: A comprehensive survey”, Neurocomputing, 516, 205-230.

[6] Ren Z., Chen C., Fang M. (2021) Self-calibration method of gyroscope and camera in video stabilization. 2021 International Conference on Electronic Information Engineering and Computer Science, September, 23-26 2021, Changchun, China.

[7] Pavlovic M., Banjac Z., Kovacevic B. (2022) "Digital Video Stabilization Verification Based on Genetic Algorithm Template Matching". Advances in Electrical and Computer Engineering, 22, 53-60.

[8] Goshtasby A., Piecewise linear mapping functions for image registration," Pattern Recognition, 19, 1986, 459-466.

[9] Goshtasby A., Image registration by local approximation meth-ods,"Image and Vision Computing, 6, 1988, 255-261.

[10] Piatkowski T., Wolski M., Tomaszewski T., Strzelecki P., Sempruch J., Analysis of the positioning process of objects on an oblique plane with barrier, Mechanism and Machine Theory, 2021.

[11] Piątkowski T., Zagadnienie skalowania obrazu z kamery cyfrowej w badaniach eksperymentalnych torów ruchu sortowanych obiektów, Współczesne wyzwania transportu i elektrotechniki: monografia pod red. Tomasza Ciszewskiego i Jerzego Wojciechowskiego, UTH Radom, 2021.

[12] Piatkowski T., Wolski M., Osowski P., Method for determining motion trajectories of characteristic points registered by a video camera, Engineering Mechanics, 24rd International Conference Engineering Mechanics, Svratka, Czech Republic, 14-17.05.2018, 677-680.

[13] Galda, H., Development of a segmentation method for dermoscopic images based on color clustering, Kobe University, August, 2003.

[14] Gonzales R. C., Woods, R.E. and Eddins, S.L., Digital Image Processing Using MATLAB, Gatesmark Publishing, 2020.

[15] Moeslund T. B., Introduction to video and image processing: building real systems and applications. Undergraduate Topics in Computer Science book series, Aalborg, Dania, 2012.

[16] Jia, T., Sun N., Cao M., Moving object detection based on blob analysis. IEEE International Conference on Automation ad Logistics, Qingadao, China, 2008. 322-325.

[17] Noureldin A., Karamat T., Georgy J., Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration, Springer, 2013.