

Multicamera Systems in the Moving Body Recognition

Aleš Procházka, Miroslav Kubíček, and Aleš Pavelka

Prague Institute of Chemical Technology, Technická 1905, 166 28 Prague 6, Czech Republic
E-mail: A.Prochazka@ieee.org

Abstract - *The contribution presents the process of the space localization of the moving object using a multicamera system, possible remote data transmission through communication links and image acquisition problems. The moving body recognition and its mathematical modelling including spline interpolation of its trajectory in the three-dimensional space form the main part of the paper. Final results are presented in the computer environment of virtual reality allowing the detail study of the whole system that can represent an engineering or a biomedical process.*

Keywords - *Image acquisition, object recognition, 3D modelling, virtual reality, spline interpolation*

1. INTRODUCTION

Mathematical analysis of videosequences and motion modelling belong to an interdisciplinary area of digital signal and image processing [1] allowing detection, localisation, identification and prediction of moving objects components. Applications can be found in engineering [2], biomedicine [3] and in many further disciplines.

The paper is related to the extensive research of these topics including analysis of multiple marker association [4], geometric algebra application [5], specific methods of image features extraction [6] and position estimation [7]. The paper contributes to these topics in the area of system analysis and its virtual reality modelling.

2. IMAGE DATA ACQUISITION

Data acquisition can be realized by a synchronized two camera system in the simplest case to detect the precise position of a selected object in the three-dimensional space. Fig. 1 presents the principle of the whole system arrangement using two cameras Dragonfly connected to the PC. The Dragonfly is an OEM-style IEEE 1394 board level camera providing control and flexibility for industrial machine vision tasks. Cameras used in the system have a color CCD sensor with 1024x768 resolution and 15 frames per second. Partial Image Format (sub-sampled) allows the user to transmit a sub-sampled 640x240 image at the speed up to 50 fps. The 6-pin 1394 standard cable provides the camera [8] with both power and a connection to computer having IEEE 1394 plug-in board. For transmission of images from cameras to PC the MATLAB Image Acquisition Toolbox has been used.

Multiple Dragonfly's on the same IEEE-1394 bus are automatically synchronized to each other at the hardware level. The maximum deviation

in the synchronisation is 125 μ s of each other as states the Dragonfly Technical Reference Manual. This synchronisation of cameras has been verified by analysis of the record of the display of the counter counting 1 kHz pulses. It means that numbers on the display were changing thousand times per second. On the corresponding picture from each camera the same number on the display has been captured.

The system has been based upon the Image Acquisition Toolbox supporting a wide range of image acquisition operations from the professional grade frame grabbers to USB-based Webcams. The toolbox allows the connection of hardware, its configuration, video preview, and

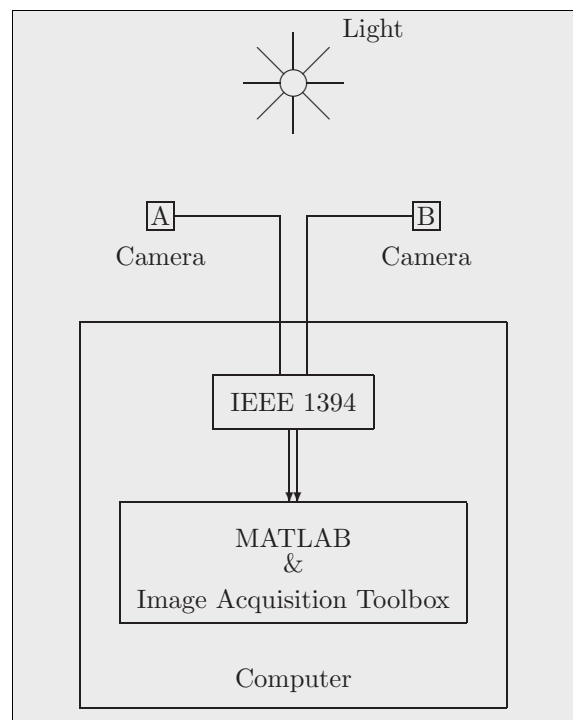


Fig. 1. The two camera measurement system

```

-----
clear all
%% Definition of the input format %%
cam1=videoinput('dcam',1,'Y8_1024x768');
cam2=videoinput('dcam',2,'Y8_1024x768');
%% The number of pictures for acquisition %%
deda.FramesPerTrigger=30;
strejda.FramesPerTrigger=30;
%% The search of the source property %%
src=getselectedsource(cam1);
src1=getselectedsource(cam2);
%% Shutter speed definition %%
src.Shutter=5; src1.Shutter=5;
%% Start images acquisition %%
start(cam1); start(cam2);
%% Wait for the acquisition to stop %%
wait(cam1); wait(cam2);
%% Transfer images to the MATLAB workspace %%
[f,t]=getdata(cam1); [g,t]=getdata(cam2);
-----

```

Fig. 2. Algorithm of the camera image acquisition

transfer of the stream of images directly into the MATLAB environment for their analysis and visualization.

A short program for capture of 30 pictures by cameras and saving them to the hard disc (acquiring 15 frames per second in the monochrome mode) is presented in Fig. 2. The program for final display of differences of all saved pictures concentrated in one picture is presented in Fig. 3.

```

-----
k=input('Number of figures = ');
d(:,:,1)=g(:,:,1);
%% Compute differences between pictures %%
for i=1:k-1
    d(:,:,1)=imabsdiff(d(:,:,1),g(:,:,i+1));
end
imview(d(:,:,1),[])
-----

```

Fig. 3. Algorithm for the final image display

3. MOVING BODY DETECTION

The whole system presented in Fig. 4 consists of two cameras A and B located in the fixed distance c . The moving object C represented by a light in this case forms together with positions A and B a triangle that can be used for space detection of the moving body.

3.1 Calibration

System calibration presented in Fig. 4(a) assumes known values of distances $a_1(1)$, $b_1(1)$ and c of the triangle ABC . The cosine theorem can be then used to evaluate initial angles $\alpha_1(1)$ and $\beta_1(1)$ by relation

$$\alpha_1(1) = \arccos \frac{b_1(1)^2 + c^2 - a_1(1)^2}{2 b_1(1) c} \quad (1)$$

$$\beta_1(1) = \arccos \frac{a_1(1)^2 + c^2 - b_1(1)^2}{2 a_1(1) c} \quad (2)$$

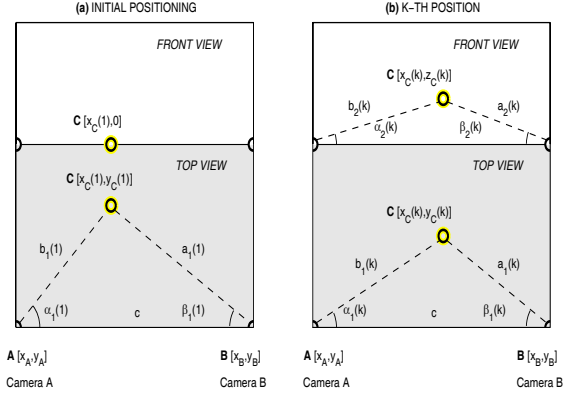


Fig. 4. Two camera system allowing (a) the calibration of the system and (b) the space localization of the moving object using the set of synchronized camera observations acquired with a chosen frequency

In the following step it is necessary to evaluate horizontal and vertical angles of both cameras. Using the calibration grid table according to Fig. 5(a) placed in the distance d from camera A it is possible to find both horizontal $s_{horizontal}$ and vertical $s_{vertical}$ sizes of the figure. These parameters can then be used for evaluation of the limits of angles presented in Fig. 5(b) using rectangular red and blue triangles estimating in the case of camera A values

$$\alpha_{horizontal} = 2 \arctan \frac{s_{horizontal}/2}{d} \quad (3)$$

$$\alpha_{vertical} = 2 \arctan \frac{s_{vertical}/2}{d} \quad (4)$$

implying limits α_{1min} , α_{1max} , α_{2min} and α_{2max} using the initial point positioning and initial angles evaluated before. The similar process can be used for camera B .

3.2 Observation

During the observation process precisely synchronized pictures taken with a chosen frequency by both cameras are acquired. The situation in

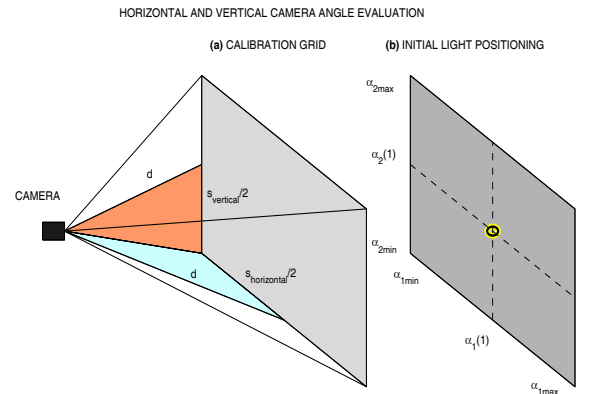


Fig. 5. System calibration based upon (a) the use of the calibration grid table and (b) evaluation of horizontal and vertical camera angle limits

the k -th observation step is given in Fig. 4(b). The moving object represented by a light can be detected using a simple thresholding method applied to individual images. Using the results of calibration given in Fig. 5(b) it is possible to convert the row and column positioning of the light to horizontal $\alpha_1(k)$ and vertical angles $\alpha_2(k)$ in the case of camera A and to $\beta_1(k)$ and $\beta_2(k)$ angles in the case of camera B .

For the k -th observation step the set of two pictures is acquired defining triangle ABC in the space with the top and front view presented in Fig. 4(b). Using the system of coordinates with the origin in the position of camera A and choosing the axis x in the direction of camera B and y axis in the plane of the initial light positioning it is possible to evaluate coordinates of point C for each set of camera observations. The top view enables to find the size of $b_1(k)$ using the sine theorem and coordinates of point C in the form

$$b_1(k) = c \sin(\beta_1(k)) / \sin(\pi - \beta_1(k) - \alpha_1(k)) \quad (5)$$

$$x_C(k) = b_1(k) \cos(\alpha_1(k)) \quad (6)$$

$$y_C(k) = b_1(k) \sin(\alpha_1(k)) \quad (7)$$

The z coordinate of point C can be found in the similar way.

$$b_2(k) = c \sin(\beta_2(k)) / \sin(\pi - \beta_2(k) - \alpha_2(k)) \quad (8)$$

$$z_C(k) = b_2(k) \sin(\alpha_2(k)) \quad (9)$$

Definition of the three dimensional positioning of the moving body is given in this way in the chosen coordinate system for each set of camera images.

3.3 Visualization

Resulting three column matrix of three coordinates of the moving object in each its line can be further processed and visualized. Results of a selected set of 30 observation is presented in Fig. 6. The following spline interpolation having similar effect as the increase of the camera frequency image acquisition has been further used to smooth results.

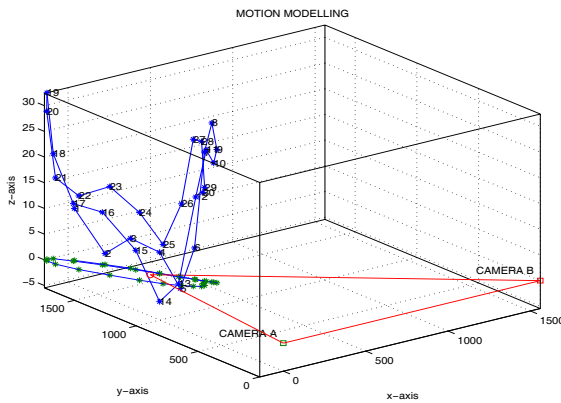


Fig. 6. Localization of the moving object in the three dimensional space based upon the camera systems observations

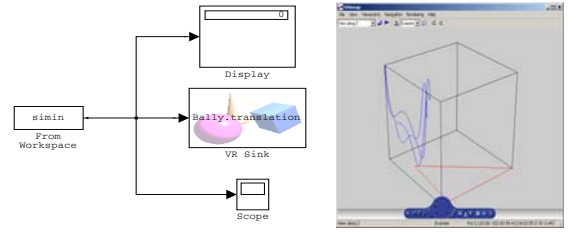


Fig. 7. Simulink model Fig. 8. VRML model

4. VIRTUAL REALITY MODELLING

Modelling in the virtual reality (VR) environment [9] is very efficient both in engineering, biomedical and other applications to study the moving object properties. This kind of modelling allows separate definition of different objects that can form the final model to make it as realistic as possible. Moreover it enables the incorporation of mathematical or physical analysis results to make the virtual system close to the real system conditions. To build such a model in the case under study it is possible to use results of data acquisition and processing in the MATLAB environment presented above. The final image can be transformed into the VR model whose objects can be controlled from the Simulink environment presented in Fig. 7. The data set evaluated in MATLAB can be used to control the specific movement of those objects.

The proposed VR model and its control system in Simulink is created as the whole in the MATLAB environment. For the user of the program it is necessary to capture the image sequence only. The rest of the process including the image identification, calculation of position of the light, spline interpolation of its movement, Simulink model creation and the final VR model creation is fully automatic. The creation of the VR file using results of MATLAB modelling with the Virtual Reality Toolbox is enabled by the `vrm1(gcf,name_file)` command.

Fig. 9 describes how easy the creation of the VR model is. The most complicated problem in the whole process is in the calculation and plot of a simple and transparent graph without too much detail objects. Fig. 8 has been created from the classical MATLAB figure using the `vrm1(gcf,'my_vrml_file.wrl')` command only having in mind that the MATLAB's Virtual Reality Toolbox is necessary for its execution.

The same principle has been used during the creation of the corresponding Simulink model to enable the movement of the light (the yellow ball) on the trajectory. The object of the light in the VR model has a special property that determines the position of the object and this property is changed in the VR model by Simulink in the real time.

```

-----
%% VRML Creation %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(5)
% Trajectory of moving object
plot3(siminin(:,1),siminin(:,2),siminin(:,3));
hold on;
% Zero Camera Plane
plot3([0 c xc0 0],[0 0 yc0 0],[0 0 zc0 0],'r');
hold off
% Removing useless graphics
axis tight; box on
set(gca,'XTick',[]); set(gca,'YTick',[]);
set(gca,'ZTick',[]);
% Correction of camera position
set(gca,'CameraPosition',...
get(gca,'CameraPosition')/1.8);
% Generating of unique file name
name_file=['ball_',num2str(now),'.wrl'];
name_file=[name_file(1:11) name_file(13:end)];
% Create VRML file
vrml(gcf,name_file);
-----

```

Fig. 9. Algorithm for the simple VRML creation

The real system presented in the virtual reality environment can be studied and observed from any arbitrary location of the viewer in the space using any zoom level. Results presented in Figs 10 and 11 show different views towards the object represented by a moving yellow ball.

Further modelling steps can enable even more realistic views and the study of the motion enabling control of the motion speed as well. System described above presents a very simple example of possibilities of three dimensional motion modelling following one moving object only.

Similar approach can be used in the case of detection of several moving objects allowing the study of their interaction as well.

5. CONCLUSION

The paper presents both technical principles of the moving body detection using the image acquisition toolbox and a synchronized two video cameras system. Specific mathematical methods of image components localization for processing of each couple of observed images are then used.

Basic principles of the virtual reality visualization presented in the paper enable a very convenient way of the study of a moving object both in the simple example presented in the paper and in more complex applications. The paper presents how such a model based on the MATLAB computational system can be created. The most important algorithms are included in the paper as well.

The future work will be devoted to the multiple moving objects detection, recognition and modelling. The purpose of such a study will be in the analysis of the complete body movement. Prediction methods will be incorporated as well.

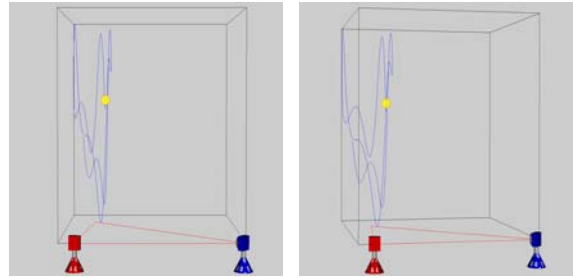


Fig. 10. The back view of the system Fig. 11. Camera A view towards the object

ACKNOWLEDGMENTS

The work has been supported by the research grant of the Faculty of Chemical Engineering of the Institute of Chemical Technology, Prague No. MSM 6046137306.

REFERENCES

- [1] R. C. Gonzales, R. E. Woods, and S. L. Ed-dins. *Digital Image Processing Using MATLAB*. Prentice Hall, 2004.
- [2] M. Ringer and J. Lasenby. Modelling and Tracking of Articulated Motion from Multiple Camera Views. In *Proc. British Machine Vision Conf (BMVC)*, pages 172–181, 2000.
- [3] R. Boulic, P. Fua, L. Herda, M. Silaghi, J.S. Monzani, L. Nedel, and D. Thalmann. An Anatomic Human Body for Motion Capture. In *Technologies for the Information Society: Developments and Opportunities*. EMM-SEC98, 1998.
- [4] M. Ringer and J. Lasenby. Multiple Hypothesis Tracking for Automatic Optical Motion Capture. *Lecture Notes in Computer Science*, 2350, 2002.
- [5] J. Lasenby and A. Stevenson. Using Geometric Algebra for Optical Motion Capture. In E. Bayro-Corrochano and G. Sobczyk, editors, *Applied Clifford Algebras in Computer Science and Engineering*. Birkhauser, Boston, U.S.A., 2000.
- [6] M. Nixon and A. Aguado. *Feature Extraction & Image Processing*. NewNes Elsevier, 2004.
- [7] M. Ringer, T. Drummond, and J. Lasenby. Using Occlusions to Aid Position Estimation for Visual Motion Capture. In *Proc Computer Vision and Pattern Recognition (CVPR)*. IEEE USA, 2001.
- [8] M. Kubíček. Using Dragonfly IEEE-1394 Digital Camera and Image Acquisition Toolbox. In *Sborník konference MATLAB 2004*, pages 280–282. VŠCHT Praha, 2004.
- [9] A. Pavelka, M. Kubíček, and A. Procházka. Motion Observation and Modelling in the Virtual Reality Environment. In *Sborník konference MATLAB 2005*. VŠCHT Praha, 2005.