World Applied Sciences Journal 4 (2): 244-249, 2008
ISSN 1818-4952

Graphical User Interface (GUI) Development for Object Tracking System in Video Sequences

S. Sulaiman, A. Hussain, N. Md. Tahir and S.A. Samad

Department of Electrical, Electronics and Systems, Faculty of Engineering,
Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor DE, Malaysia

Abstract: In an intelligent surveillance system, object tracking provides better sense of security using visual information. Hence, the main purpose of this work is to develop an object motion detection system that detected the presence of motion in video sequences. The proposed system employed background modelling and image differential techniques to isolate the motion objects from its background. Additionally, morphology processes are performed to enhance the object pixels that have been extracted so that a more accurate output image that comprised of the detected object can be determined. Subsequently, the plotting method is intentionally developed to denote the object being tracked. Two different approaches were utilized namely, edge detection and boundary box. The plotting process will be updated as the object moves in the video surrounding. In addition, a Graphical User Interface (GUI) for the system is built to facilitate users to apply the developed system and understand its hierarchy. GUI that acted as the intermediate medium creates a form of communication between users and the developed object detection system. The feasibility and workability of the proposed detection system along with the built GUI is demonstrated throughout the experimental results that clearly detect and track the moving objects in the video sequences recorded in the outdoor environment.

Key words: Edge detection . Graphical User Interface (GUI) . Object tracking

INTRODUCTION

The use of vision-based camera systems in surveillance has grown exponentially in recent years wherein one of the important tasks is object motion tracking. To realize automatic or semi-automatic surveillance video analysis, low-level motion data must be initially extracted by tracking algorithms [1]. There have been many algorithms developed to track and detect the motion of interest subjects in a video surrounding; most of them applied the four stages of image differencing, thresholding of the difference image, morphological filtering and connected-component labelling [2].

The main objective of this paper is to describe the development process of an object tracking system along with its GUI that will provide a communication path to the system users.

MATERIALS AND METHODS

Two major works in this study consist of developing the object tracking algorithms followed by designing of the GUI. Figure 1 illustrated the developed

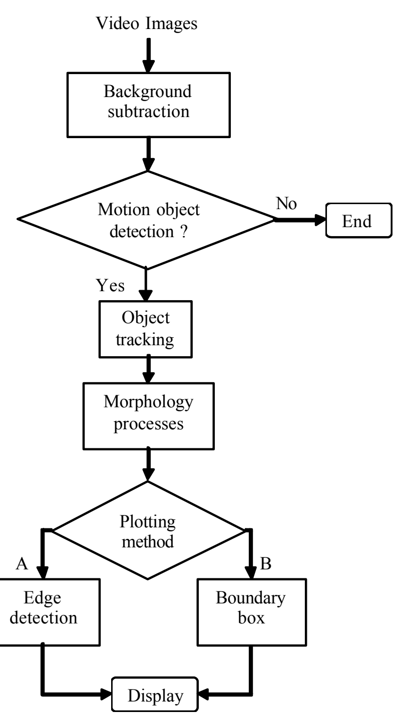

Fig. 1: Flow chart of the object tracking system

Corresponding Author: S. Sulaiman, Department of Electrical, Electronics and Systems, Faculty of Engineering, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor DE, Malaysia

object tracking system based on background subtraction, motion detection and tracking of object.

Object detection algorithm based on background subtraction and morphological processes: The basic idea underlying the background subtraction is image differentiation between the present frame image which contains the subject of interest and the reference images [11]. Thus, a set of background image frames will be functioned as a reference set of images. In this work, a better technique involves the background model as mentioned in [5] is applied. These background image pixels are represented by three values as below:

- Minimum *m(x),* intensity values
- Maximum *n(x)* intensity values
- Maximum difference *d(x)* between consecutive frames

Next, each of the sequence images from the input video that consist of object motion will be processed to extract the object silhouette. The previous background model is used to segment out the foreground pixels from the background scene. By taking the minimum *m(x) and* maximum *n(x)* intensity values and the maximum difference *d(x)* between consecutive frames that represent the background scene model *B(x)*, pixel *x* from image *I* is classified by following the below equation:

$$B(x) = \begin{cases} 0 \text{ background} & \begin{cases} (I(x) - m(x)) < kd(x) \\ (I(x) - n(x)) < kd(x) \end{cases} \\ 1 \text{ foreground} & \text{otherwise} \end{cases} \quad (1)$$

Where *k* is the Gaussian distributions constant, in this case we consider *k* is ranging from 2 to 5.

It is well known that thresholding is an important aspect in background subtraction procedure and it is quite crucial to determine the most suitable threshold value. Therefore, morphological processes are applied for a more perfect silhouette. Alternatively, morphology is a broad set of image processing operations that process images based on shapes [5]. In this work, we applied morphological operations namely dilation and erosion. Both of these operations can be manipulated using single or combination of specific structuring elements.

The main purpose of this step is to filter the presence of noisy pixels in the foreground image. Subsequently, after the object has been tracked, the system will apply one of the plotting methods based on users' choice, namely, edge detection and boundary box. The objective of these methods is to denote the tracked object and update it according to each of the image frames of the input video.

GUI design and development: A GUI is designed to present the developed tracking algorithm in a visual and graphical form that will ease users to interact better with the system. The actions are performed through direct manipulation of the graphical elements such as push buttons, radio buttons, list boxers and menus. These elements are arranged systematically to ensure that its usage is trouble-free and comprehend by the potential users. Generally, UI provides input, allowing the users to manipulate a system and an output, permitting the system to produce the effects of the users' manipulation. A good UI is one that can present the information clearly and effectively. In designing a GUI, three main factors are taken into consideration namely, usability, simplicity and interactivity [7].

Usability is tangible and can be quantified. Generally, it can be defined as "ease of use", including such measurable attributes as learnability, speed of user task performance, user error rates and subjective user satisfaction. However, an easy-to-use system that does not support its users' needs, in terms of functionality, is of little value. Thus, usability has evolved toward the concept of "usability in the large", ease of use, plus usefulness.

Simplicity does not mean lack of functionality, in fact it denotes a fast initial user learning curve and consideration for the number of concepts the user actually needs to understand. The goal for a good UI design should be easy to learn, but with a built-in path for higher proficiency. The golden rule of simplicity is to design a system with components that will actually be used by the user, without any unnecessary elements.

In designing a UI, interactivity factor is another important thing that should be deemed. An interactive and attractive appearance of the UI will provide users a comfortable feeling for further use of the system. It should react the way users expect the system to respond. The style of interaction offered by a UI is the dialog employed for communicate between the human and computer. The most common manner of interaction are menu, instructions language, WIMP (window, icon, menu, pointing device), form-filling, questions along with answers dialog, 3D interface and direct manipulation. Each of these dialog types has their own structures and standard parameters. For example, direct manipulation is a type of interaction style that allows users to interact with objects by manipulating representation of objects presented by the UI. Users will normally select the objects presented on the screen using a pointer device. Moreover, a system that offers error prevention and error message can be categorized as an interactive system from the perspective of users' error handling.

RESULTS AND DISCUSSION

In this work, recorded video scenes consist of various human movements are utilized to examine the developed object tracking algorithm. Video sequences are recorded using one static camera placed at a distance approximately between six to ten meters. The objects of interest which is specifically human in our study; will move in the grey region as indicated in Fig. 3. The detected human identified on the original frame images is tracked using either the edge detection or boundary box method.

Sample results and the ability of the developed algorithms to track and detect object of interest in video scenes are as illustrated in Fig. 4. The object is traced using either one of the plotting method and will be updated according to their movements in the particular video input. Additionally, the algorithm can also be applied to detect fast actions such as jumping and running. Moreover, it performed well for tracking object movement in circular view as depicted in the first two columns in Fig. 4.

Besides that, another major work accomplished was designing and developing the GUI (graphical user interfaces) for the tracking system. It is accomplished using GUIDE, the MATLAB GUI development environment that provides a set of tools for creating GUI.

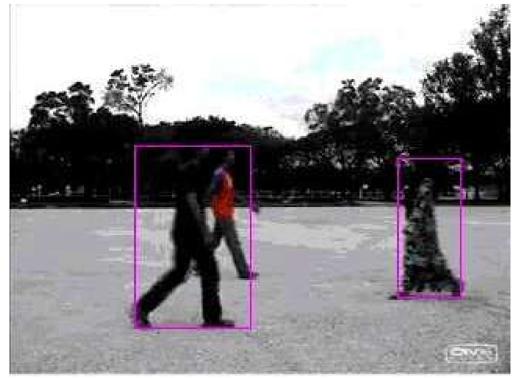

Fig. 2: The bounding box plotting on the original RGB image

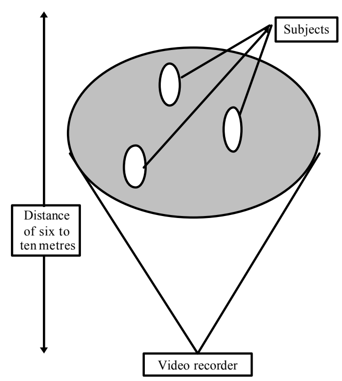

Fig. 3: Camera placement for the input recorder video system

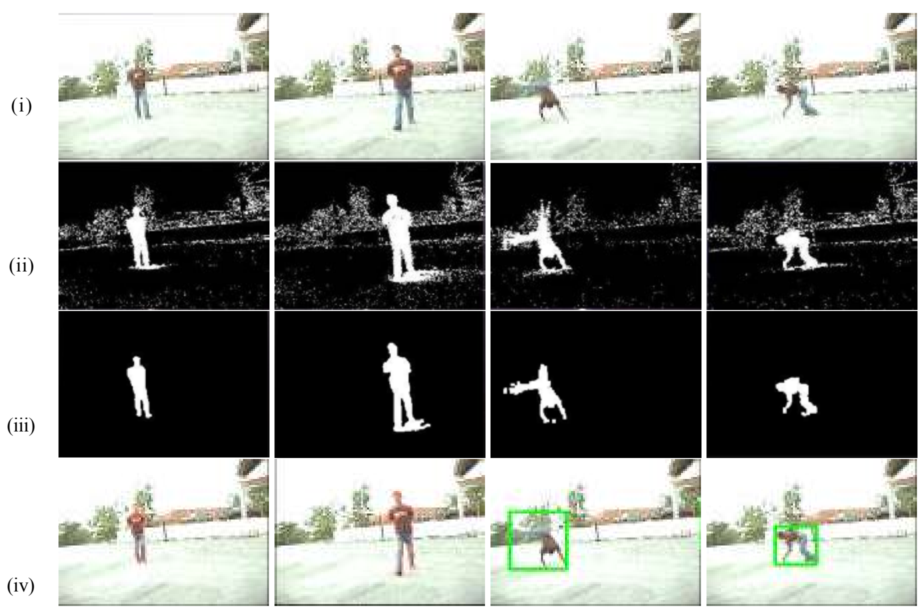

Fig. 4: Sample results of tracking algorithm (i) original RGB image (ii) foreground object tracking with the presence of noisy pixels (iii) binarized image (iv) tracking outcome using edge detection method (first two columns) and boundary box technique (third and forth column)

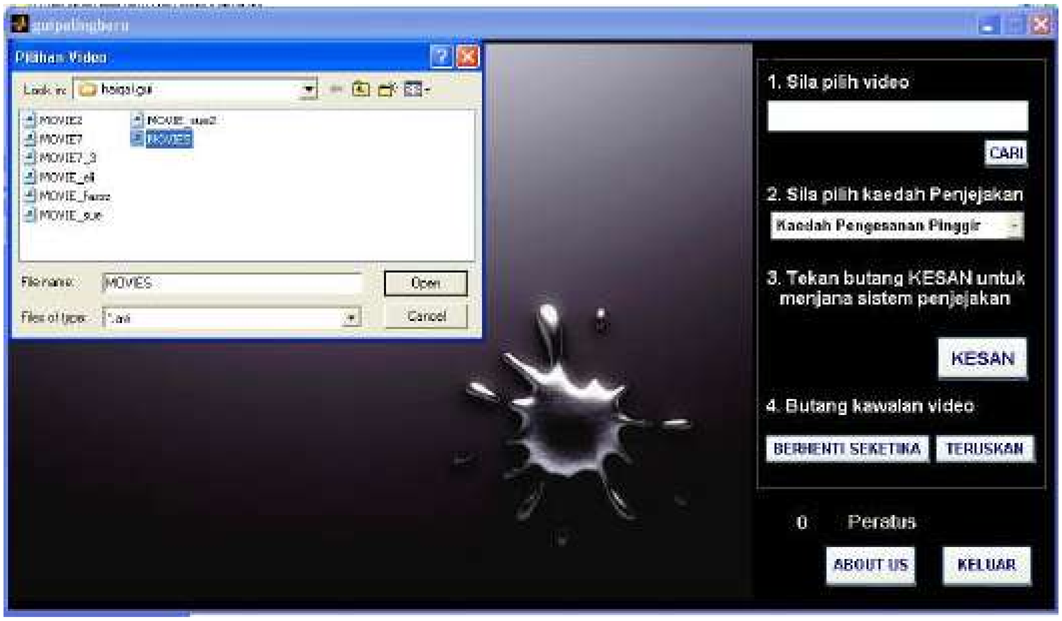

Fig. 5: A file-browsing window

Table 1: Sample of m-code for input files selection

```
% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton8 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
err.message = 0;
[FileName,PathName] = uigetfile('*.avi','Pilihan Video');
cd(PathName);
h = waitbar(0,'Sila Tunggu Sebentar...');
try
 VideoInfo = aviinfo(FileName);
 waitbar(1/3)
 handles.BilFrame = VideoInfo.NumFrames;
 handles.mov = aviread(FileName);
 waitbar(2/3)
catch
 err = lasterror;
end
if err.message ~= 0
 close(h)
 errordlg('Sila pilih video jenis.avi sahaja.','Error');
 return
end
```

GUI is a communication tool that linked the users to the system, creating a path of interaction between each other. Appropriate graphical panels are selected and positioned aptly for users to explore and utilize the system. Additionally, we also emphasized the interface delivery method for better and interactive system design and to attract users to communicate with the system through the GUI. For example, colour and pattern selection is one of the major characteristic that must be considered in a GUI design. There are six main push buttons provided on the GUI named *search*, *process*, *pause*, *continue*, *about* and *exit*. Each of this buttons has its own functions and executes specific processes when activated by the user. For instance, the *search* button is created to help users browsing for the video file that will be the input file of the developed tracking system. It is performed using the MATLAB function *uigetfile* that will pop up a file browsing window. Figure 5 illustrates the GUI display when the users hit the *search* button.

After choosing a specific input file, the system will subsequently read it and set to process that particular file according to the user's next action. The *m-code* written for this specific button is shown in Table 1.

Another important push button is the *process* button. It is developed to execute the object tracking system subsequent to video input file uploaded by the user. At this state, the GUI manipulates the tracking algorithms that have already been developed separately. Four output are displayed namely the original video file, background subtraction, noise removal and output video. All these *axis graph* display depicted the results of the tracking algorithm in a video surrounding and proven to track the motion object in that video images successfully. As the analysis process is being run, two other push buttons that are utilized are the *pause* and

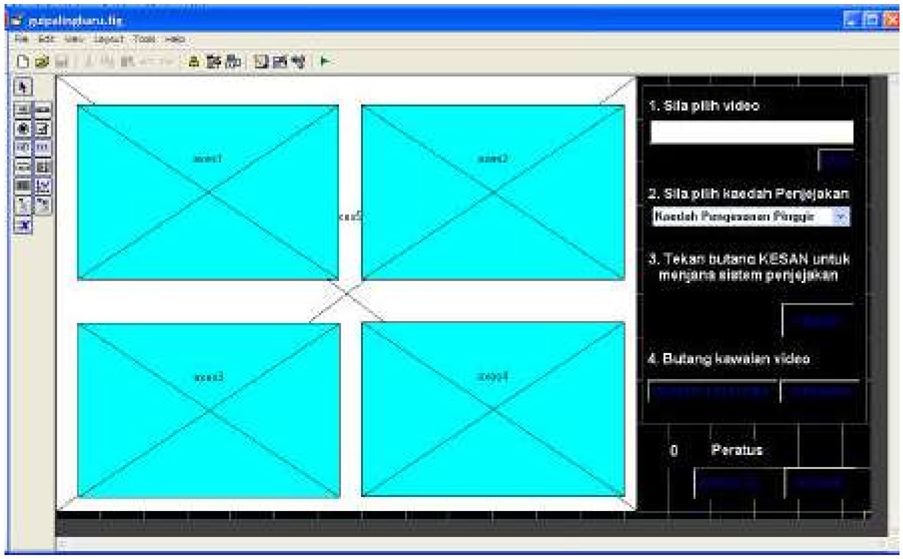

Fig. 6: A preliminary design of an object tracking GUI layout

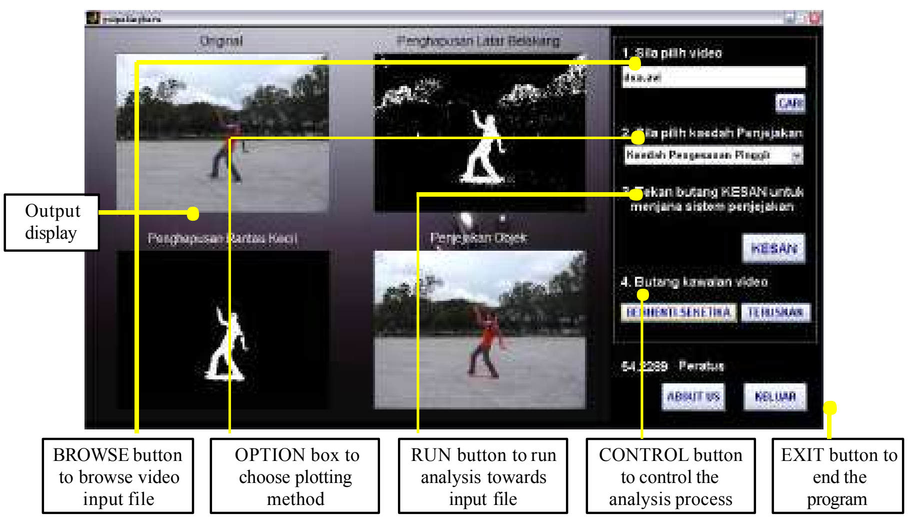

Fig. 7: The GUI display for object tracking system

Table 2: Partial code for set up of push buttons

```
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
uiwait;

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
uiresume
```

continue button. These two buttons will request for the *uiwait* and *uiresume* instruction function, respectively, as shown in Table 2.

Further, we have also applied a *listbox* graphical panel in our GUI design. This element is developed to provide options for the plotting method. Users can select either the edge detection method or bounding box method. Additionally, a percentage meter showing the analysis progress is provided at the right bottom of the GUI layout. An *exit* push button is also placed if the users wish to quit from the programme. Figure 6 depicted the preliminary design of the GUI layout whilst Fig. 7 illustrated a sample of the GUI display with the tracked object being analyzed.

CONCLUSION

In conclusion, the ability of both techniques namely the edge detection and boundary box methods for tracking objects in video sequences has been tested. The functionality of the built GUI has also been demonstrated and confirmed. These approaches can be applied for pedestrian detection, monitoring of human action, human activity recognition and smart surveillance system.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the contribution of Mohd Fariq Haiqal Masekh and Abdul Marwan Mohamad Shah and other members of the DSP research team who volunteered as subjects in the video data.

REFERENCES

1. Bhanu, B. and X. Zou, 2004. Movings Human Detection Based on Multi-modal Sensor Fusion. Proc of IEEE Computer Vision and Pattern Recognition.
2. Curio, C., J. Edelbrunner, T. Kalinke, C. Tzomakas and W. Seelen, 2000. Walking Pedestrian Recognition. In IEEE Transactions On Intelligent Transportation System, 1 (3): 155-163.
3. Elbasi, E., L. Zuo, K. Mehrotra, P. Mohan and P. Varshney, 2005. Control Charts Approach for Scenario Recognition in Video Sequences. Turk. J. Elec. Eng., Vol: 13 (3).
4. Gavrila, D.M., 2000. Pedestrian Detection from a Moving Vehicle. In: Proc. Euro. Conf. Comp., Vision, pp: 37-49.
5. Haritaoglu, I., D. Harwood and L.S. Davis, 2000. W4: Real-Time Surveillance of People and Their Activities. In: IEEE Trans. Pattern Analysis and Machine Intelligence, 22 (8): 809-830.
6. McIvor, A.M., 2000. Background Subtraction Techniques. In Proc. of Image and Vision Computing, 1 (3): 155-163.
7. Hartson, H., 1998. Human Computer Interaction: Interdisciplinary Roots and Trends. In The Journal of System and Software, 43 (2):103-118.
8. Monnet, A., A.M.N. Paragios and V. Ramesh, 2003. Background Modeling and Subtraction of Dynamic Scene. In Proc. 9th IEEE Intl. Conf. Comp. Vision, 2: 1305-1312.
9. Stauffer, C. and W.E.L. Grimson, 1999, Adaptive Background Mixture Models for Real-Time Tracking. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2: 246-252.
10. Xiong, Q. and C. Jaynes, 2004. Multi-resolution Background Modelling of Dynamic Scenes using Weighted Match Filters. In Proc. of the ACM 2nd International Workshop on Video Surveillance and Sensor networks, IEEE Conference on Multimedia, New York, NY, USA, pp: 88-96.

Book:

1. Gonzalez, R. and R. Woods, 2000. Digital Image Processing. 2nd Edn. Prentice Hall, New Jersey, USA.