TUJNAS

Thamar University Journal of Natural & Applied Sciences ISSN: 2073-0764
2009 A(**1**) 61-74

Development of Advanced Monitoring System in Real Time Environment

Khalil Saeed M. Al-Wagih

Information Technologies Dept., Faculty of Computer Sciences and Information Systems, Thamar Univ., Yemen, khalilwagih@gmail.com

ABSTRACT

In this paper, the problem of providing a good computerized monitoring system is discussed. As an application, Advanced Monitoring System (AMS) is designed and built for this purpose. AMS is an image processing, a soft real time system, which is concerned with enhancement of the monitoring services with lower cost and a high performance. It seems to be very broad area of research as it relates to many important subjects of computer science. The produced system has accomplished its intended functions and services in trusted fashion.

1. INTRODUCTION

Since computer invention, there have been many persistent attempts to use it in a range of areas. The common main goal is to achieve easier life by using the modern technology. Day by day, computer has become the important element in most of daily life activities such as education, medicine, engineering, management…etc.

Today, digital image processing represents one of the most important fields of the computer science. Here, the applications and needs of digital image processing include very long list that contains: Games, Motion Detection, Pattern Recognition…etc. The main issue of digital image processing is the huge computation which might be required [1,5].

"Real time system is any system where a timely response by the computer to external stimuli is vital". The term timely means that the real time system runs tasks that have dead lines [2].

This work is an application of digital image processing in real time environment that aims at developing an automatic monitoring software system which can help in security purposes. The intended system should process a stream of image frames. For the purpose of monitoring, an on-line camera which is connected to the computer must focus on the monitored place.

Practically, the intended system is indeed a real time software system. Here, the task of developing digital image processing software to work in real time mode takes its attraction. This is actually because of time sensitivity and required huge computation.

AMS takes its importance because of the following reasons:

Firstly, monitoring rigid places is a hard human job, considering the time those humans remain to monitor a needed place.

Secondly, there are some noticed cases at many places in which the computer is used. However, no alert actions can be considered. It is just to use the computer screen in human monitoring. Consequently, it is a computer abuse manner without high benefits.

Finally, full automated monitoring systems may achieve the monitoring targets but with a huge costs.

AMS is a trial to solve the above problems with simple solutions.

It starts working when the user requests the monitoring via connected camera for creating the base frame. Then, it starts monitoring services.

During the monitoring, the AMS senses any change in the monitored place, as a result of this, it informs the user by running a suitable alert response. The other system response is done by recording the occurred actions.

Digital image processing remains a challenging domain of programming for several reasons. First, the issue of digital image processing appeared relatively late in computer history, it had to wait for the arrival of the first graphical operating systems to become a true matter. Secondly, digital image processing requires the most careful optimizations and especially for real time applications. Comparing image processing and audio processing is a good way to fix ideas. Let us consider the necessary memory bandwidth for examining the pixels of a 320x240, 32 bits bitmap, 30 times a second: 10 Mo/sec. Now with the same quality standard, an audio stereo wave real time processing needs 44100 (samples per second) x 2 (bytes per sample per channel) x 2 (channels) = 176Ko/sec, which is 50 time less [1,5].

Obviously, it could not use the same signal processing techniques in both audio and image. Finally, digital image processing depends almost on definition of two dimensional domain; this somehow complicates things when elaborating digital filters.

2. PROBLEM DETERMINATION AND OBJECTIVES DEFINITION

There are many problems associated with monitoring systems, which can be listed as follows:

- The monitoring using human capabilities without computerized system is a hard job.
- The computerized monitoring systems without any advanced services such as (change detection, user alarm) are also not efficient systems, because the users of those systems also need continuous watching of the monitoring screen(s).
- The cost needed to develop an efficient monitoring system is very high, because the hardware components will cost a lot of money.

Consequently, the main objectives of the AMS system are to:

- minimize human effort during the monitoring job;
- enhance the monitoring services by using the changes detection, recording the changes and user alert services;

- reduce the cost of the computerized monitoring system, by using the normal hardware attributes required for system work;
- make the monitoring more easily and more powerful; and
- allow the users of the monitoring systems to do another works during monitoring such as reading.

3. SYSTEM DESIGN AND IMPLEMENTATION

The intended AMS has to perform the following basic functions:

- monitor the considered places;
- alert the user when any change has occurred during monitoring**; and**
- record the happened changes.

In addition, to provide system integration the following functions are included:

- user-friendliness;
- reliability; and
- efficiency

As any real time system, AMS Design Phase is very crucial, therefore the designer has to make the desired design simple but highly efficient [2]. The design phase starts from the general design specification toward more detailed design specification i.e., from top to bottom.

3.1 General Design Specifications

The system picks up the input from a user and a connected online camera. Then, it processes the input to produce some results that represent the output. Consequently, the input falls into two categories. The first category is user's input (user name, user password and system setting). The camera's input represents the second category of the intended system input (captured frames). Here, the major component of the AMS system is said to be Core Processing Unit (AMSCPU). The AMSCPU is responsible for most of AMS work functions. Briefly, it gets the input from a user and camera. Then it applies the input to the processing operations. Specifically, the AMSCPU is responsible for accomplishing four main successive tasks. In those tasks, AMSCPU should firstly get user settings and check if a base frame is captured. In case of no base frame, the user must acquire the base frame to continue the work. Secondly, acquired frames via the connected online camera are compared with the base frame. Thirdly, during the monitoring phase, if there is any change detected in the acquired frame, then it alerts the user, and records a stream of frames that have the detected change. All the previous steps are presented in the following points:

a. store each action in the log files; and
b. control the possible errors.

Side by side, the output could include:

a. informing the user with any performed action;
b. displaying the stream of captured frames;
c. storing to log files; and
d. Alerting actions.

Figure (1) summarizes the general view of the AMS system.

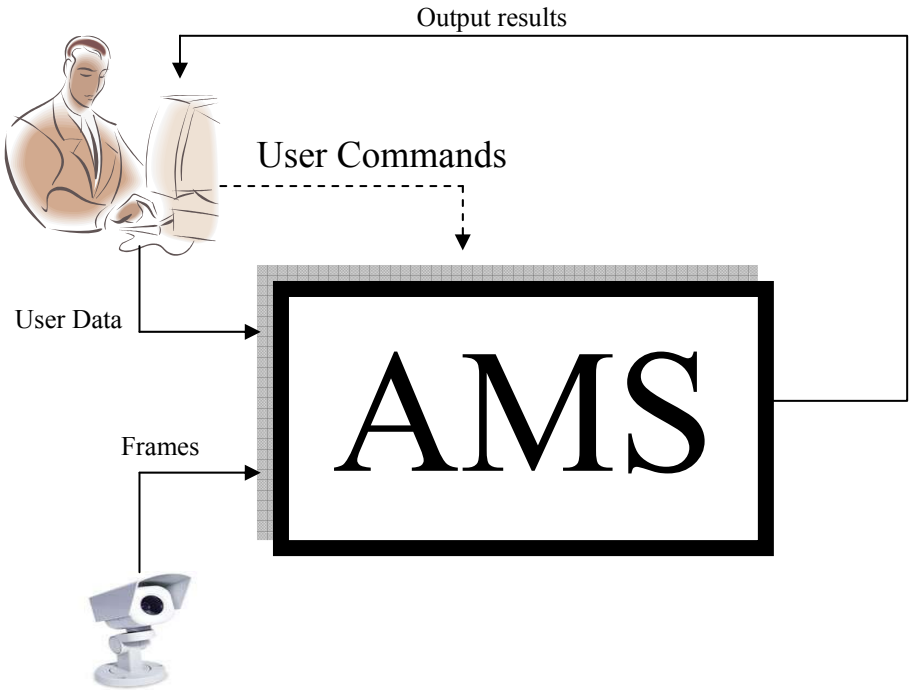


Figure (1): AMS general view.

3.2 Detailed Design Specification

According to figure (2), it is easy to note that the AMS design is constructed by using layering with modularity approach. Clearly, three main layers can be distinguished as in the following:

1- AMS Input Unit Layer (Upper layer);
2- AMS Core Processing Layer (Middle layer); and
3- AMS Output Unit Layer (Lower layer).

3.2.1 AMS Input Unit Layer (AMSIU)

The AMS Input Unit Layer is built of AMS Input Interaction Unit (AMSIIU). The AMSIIU is one of the AMS system main components, which is responsible for acquiring and processing user and camera input. In other words, AMSIIU should:

1. identify system input;
2. control possible input error;
3. deliver the received input to the lower layer; and
4. store the input actions to the log file.

According to the two input categories, the AMSIIU is divided into two modules including User Input Manipulation Unit (UIMU) and Camera Input Manipulation Unit (CIMU).

The UIMU module is responsible for:

1. getting the configuration setting of the user;
2. getting the configuration setting of the system depending on the user's needs;
3. sending user commands to Camera Input Manipulation Unit;
4. sending user setting to Camera Input Manipulation Unit;
5. sending User setting and/or system configuration setting to AMS Core Processing Unit (AMSCPU);
6. informing user against the system errors; and
7. receiving commands come from AMSCPU.

The CIMU module is responsible for:

1. interfacing with camera device;
2. delivering acquired frames to the AMS Core Processing Unit (AMSCPU);
3. receiving and analyzing user commands;
4. sending connection states to the user;
5. sending connection commands to the AMS Core Processing Unit (AMSCPU); and
6. informing the user with the error(s) in the system.
7.

3.2.2 AMS Core Processing Unit Layer (AMSCPU)

The AMSCPU layer can be considered as the heart of the AMS system .This layer is responsible for providing the most important AMS functions. The AMSCPU consists of three separated and related modules including Input Interface Unit (IIU), Processing unit (PU) and Output Interface Unit (OIU).

The Input Interface Unit (IIU) represents an interface to the AMSIU layer (Most Upper layer). It performs the initialization of the PU services. Logically, IIU module is divided into Camera Interface Unit (CIU) and User Interface Unit (UIU). The CIU is responsible for manipulating the Camera Input Manipulation Unit output.

CIU consists of two parts:

- Commands unit (Camera interface command unit. CICU) which is responsible for receiving the CIMU unit and Processing unit (PU) commands, and sending the command(s) to the appropriate unit(s): CIMU, Commander and Frame Receiver; and
- Data manipulation unit (Frames Receiver) which is responsible for receiving the commands from CICU and the data from CIMU, and sending the data to the (Transformer in PU).

UIU is responsible for manipulating the User Manipulation Unit intended output. UIU consists of two parts:

- Command unit (User Interface Command Unit. UICU) which is responsible for receiving the commands from the User Input Manipulation Unit (UIMU), and Processing unit (PU), and sending the command(s) to appropriate unit(s) (Recording Checker, UIMU and Commander in PU).

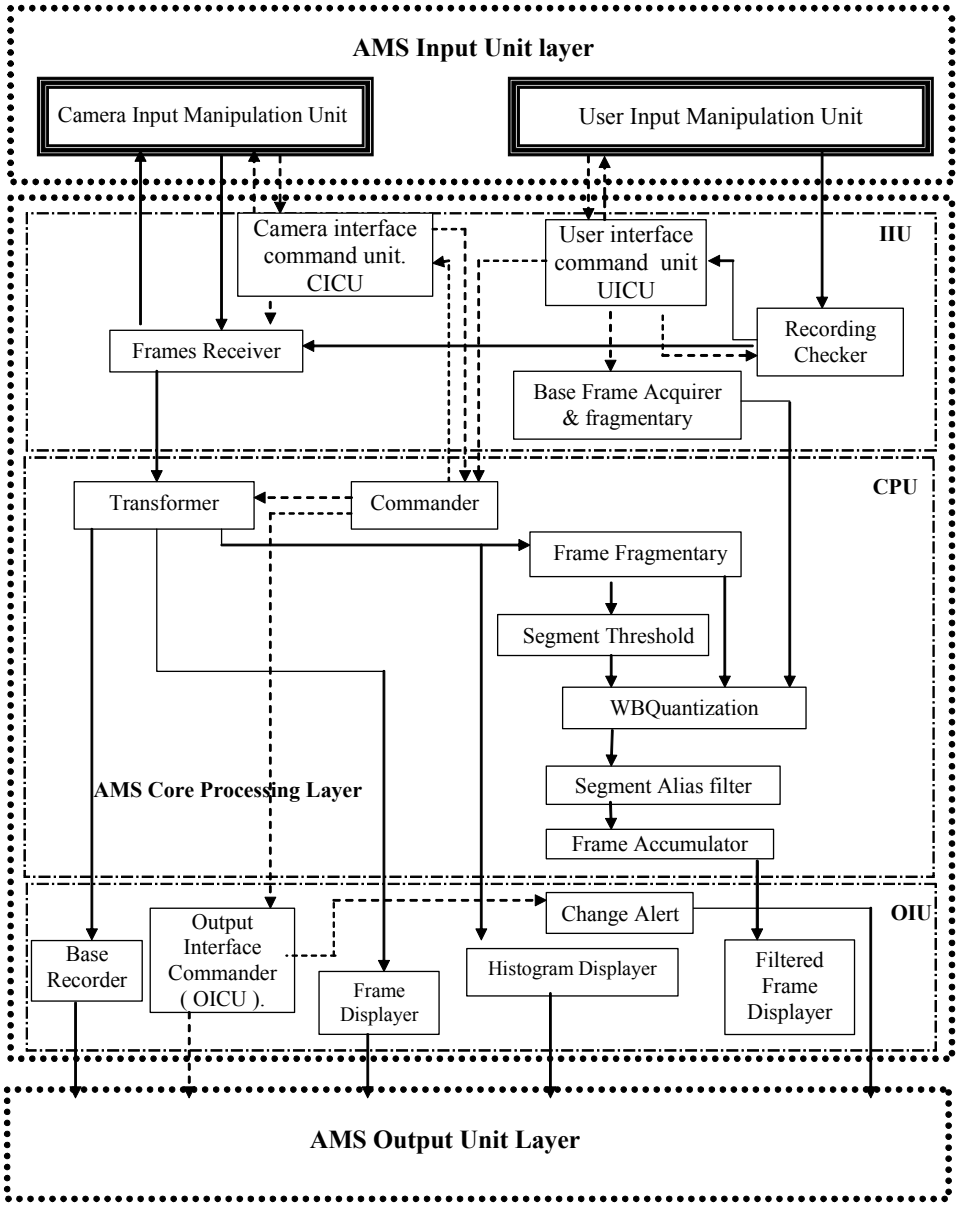


Figure (2): AMS layered view.

- Data manipulation subunits which consist of:

  - Base Frame Acquirer& fragmentary which is responsible for loading and fragmenting the base frame.

- Recording Checker which is responsible for checking the recoding space and determining the best path of the stream file.

PU is the engine of the AMS monitoring services. Logically, PU consists of two main subunits:

1. Command processing unit (Commander) which is responsible for routing the commands inside the AMSCPU.
2. Data processing unit which is responsible for managing received data.

Data processing unit is in turn divided into six modules:

(1) Transformer which is responsible for filtering, transforming and transferring the frames to the appropriate unit(s).

(2) Frame Fragmentary. The main role of Frame Fragmentary is to fragment the frames.

(3) Segment Threshold. The Segment Threshold is an important module, and its importance dwells in finding the threshold value, which is the backbone of matching process. Here, threshold value is computed via the *Adaptive thresholding algorithm*. During its work, it has only one case, which receives the segment from Frame Fragmentary.

(4) WBQuantization. WBQuantization module is responsible for quantizing the (white black) frame segments from the filtered input frame. During its work, it has one case which receives the segment from the Frame Fragmentary and then quantizes it.

(5) Segment Alias filter. Segment alias filter module is responsible for the linear matching process. It has a case which receives segment from Frame Fragmentary, then matches it by comparing the number of white pixels in it with the threshold value.

(6) Frame Accumulator. Frame Accumulator is responsible for accumulating the filtered and fragmented frame .During its work, it has a case which receives the segment from the Segment Alias filter, and then reassemble the fragmented frame.

The next section demonstrates some algorithms that are used to implement the above modules.

The OIU unit is the interface between the AMSCPU layer and the AMSOU layer. It manages the output coming from PU unit.

Briefly, OIU reforms the output to appropriate a suitable form for the outputting. It consists of two main subunits:

1- Command manipulation unit is implemented by:

- Output Interface Commander (OICU) which is responsible for routing the commands between OIU and PU.

2- Data manipulation units are implemented by:

- Change Alert;
- Histogram Displayer;
- Base Recorder (Gateway);
- Frame Displayer (Gateway); and
- Filtered Frame Displayer (Gateway)

3.2.3 AMS Output Unit (AMSOU) Layer

AMSOU is responsible for manipulating the system's output. Normally, it receives the output from (OIU in AMSCPU) and manipulates them.

3.3 AMS Implementation

AMS is an image processing and soft real time system. Therefore the computation speed is more valuable than other processing constraints (space …) .Here a programming

language is a very sensitive factor. It affects the implementation of similar systems. The decision is based on the used programming language facilitates. These facilitates can support the implementation of the presented design. The needed facilitates should include:

1. the smallest size of the executive program image (.exe file);
2. high speed run time code;
3. abilities of good memory management; and
4. the power of interaction with the operating system

Regarding the above facilitates, (C++) programming language is the best choice .Due to including graphical user interface MS Visual C++ 6.0 is used to the AMS implementation [3,4]. The most important implementation constraints can be summarized as the following:

1. The monitored object must have a different color than the background color in acceptable distance between these colors.
2. The monitored object must have an acceptable size, regarding the monitoring camera qualities.

Figure (3) shows one of the ASM interfaces.

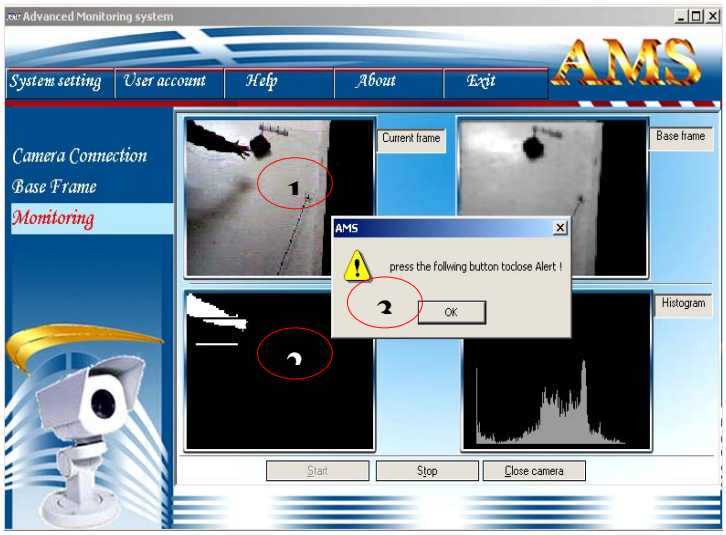


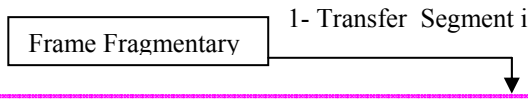
Figure (3): Monitoring with alert.

As seen in Figure(3) above, when any change is detected (1), you will see the change in the filtered frame (2) with the white color and the alert will be run. With this action, a message box will appear with this message ("press the following button to close Alert") (3), which allows you to stop the alert.

4. AMS IMAGE PROCESSING ALGORITHMS

Here, the researcher demonstrates some algorithms [1,5] which are used to implement the data processing unit. These algorithms affect directly the performance of the AMS system. In fact, the used image processing algorithms affect the AMS computation speed and accuracy. That is why the considered algorithms have to be stated. On the other hand, many improvements could be done in this area by enhancing or replacing a distinct algorithm.

4.1 Adaptive Thresholding Algorithm



For each received segment, do the following algorithm.
Adaptive thresholding algorithm.
Input: the segment
Output: adaptive threshold value of that segment

begin

1- compute the segment i histogram.
2- Store the result histogram into matrix (**Histo[256]**)
3- Y axis is the value in the Histo matrix ,and X axis is the index of the Histo matrix..
4- Find Maximum Y value **(MaxY**) and it is index (**MaxX**).
5- Find Minimum Y value (**MinY**) and it is index **(MinX**).
6- The distance (**d**) equation is : (see specification).

d= root (Square (MaxX - MinX) + Square (MaxY - MinY))

7- initialize the threshold value .

MinX-MaxX)(r = (MinY-MaxY) /

Threshold= MaxX-1
MaxY= r*(Threshold – MinX) + (MinY)
newX =(y – MinY + (MinX *r)) / r
MinY =Histo [Threshold]
/ Compute d_threshold */*
d_threshold = root (Square (Threshold - newX) + Square (newY - MinY))

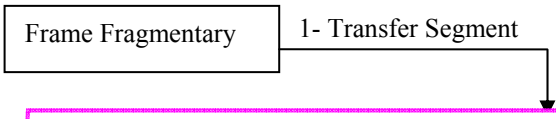
8- search for the threshod value

For i =MaxX-2 down to i>MinX
begin
y = r (i – MinX) + (MinY)
newX =(y – MinY + (MinX *r)) / r
newY =Histo [i]
/ Compute di */*
di = root (Square (i - newX) + Square (newY - MinY))
if di > d_threshold then
begin
Threshold=i
d_threshold=di
end if
end for

9- Send the threshold value to the WBQuantization unit.

end

4.2 WBQuantization Algorithm



begin

1- Get the threshold value (**Threshold**) of the segment i (**Frame_Seg**) from the Segment Threshold unit.
2- Get the segment i (**Base_Seg**)from the Base Frame Acquirer& fragmentary.
3- Compute the difference **(D)**between the colors in the (**Base_Seg** and **Frame_Seg**)

```
For  i=0 to Base_Seg.width-1
 Begin
  For j=0 to Base_Seg.hieght-1
   Begin
   C1=getpixel (Base_Seg,i,j)
   C2= getpixel (Frame_Seg,i,j)
/*
    get RGB values from C1 and C2 ,respectively and store them in the
   R1,G1,B1 for C1 ,and  R2,G2,B2 for C2
*/
```

$$D(C1,C2)=\sqrt{(R1-R2)^2+(G1-G2)^2+(B1-B2)^2}$$

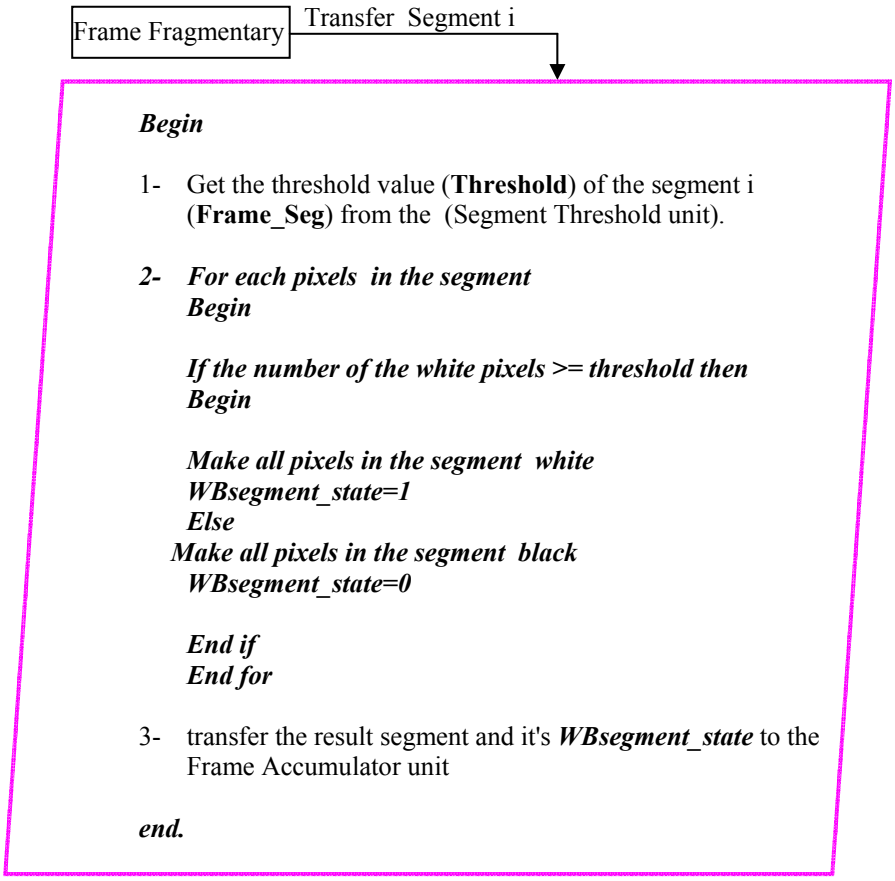
4- Quantize the Frame_Seg into white black segment (WBSegment)

```
    if  D>= Threshold then
     begin
      setpixel (Frame_Seg,i,j,WHITE)
      else
       setpixel (Frame_Seg,i,j,BLACK)
    end if
  end for
end for
```

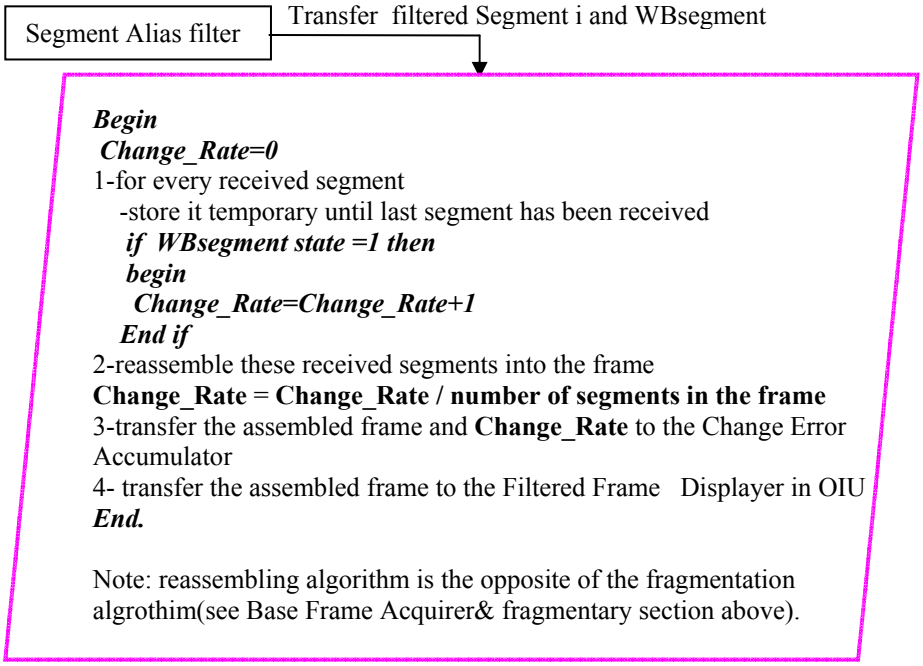
5- Transfer the result segment (Frame_Seg) to the Segment Alias filter

end

4.3 Segment Alias Filter Algorithm



4.4 Frame Accumulator Algorithm



5. DISCUSSION, CONCLUSION AND FURTHER WORKS

In this section, some important discussion is given to ensure the AMS assessment. Then, the conclusion is provided. After that, some ideas for further works are derived.

5.1 Discussion and Conclusion

There are many challenges in image processing applications, especially the applications that work in a noisy environment, but they also have the motivation and good research areas to carry on. AMS has been considered the situations of Noise Analysis and Cancellation, Image Thresholding and Computation Speedup.

The noise that AMS system processes is *an Amplified* type. This type of noise can be canceled by smoothing and blurring the image to balanced form, i.e. saving image form and removing the noise. However, some colored noise may still be in the smoothed image. So gray scaling is used to remove that noise.

Finding the image threshold value can be simple, but the computed value may not be precise depending on the image thresholding technique. There are many techniques that could be used to find image threshold. The AMS system uses *Triangle algorithm* (due to Zack). This algorithm is very efficient in finding threshold value, but when applying this algorithm on the image as a whole block, the computed threshold value may not be efficient in matching process. However, by applying it in region by region of the image then

calculating the regions threshold values will improve the matching process. This is done by using region by region matching.

One key element of the real time systems succession is the implementation algorithm that is used. The major challenge in designing efficient implementation algorithm is the application nature. In image processing applications, many processes are usually performed on the same image. In other words, the processes are performed on the same array as serious procedures which may increase the computation time. So, AMS system uses the parallelism technique when loading the image from camera to memory.

Here, it is important to clarify that the system in question has been checked for a suitable time at the laboratories of Faculty of Computer Science and Information System, Thamar University. As a result of this, the researcher can conclude clearly that the built AMS system can indeed do its function and provide the intended services of monitoring in real time environment.

5.2 Further works

As any security system, AMS system may have some imprecise issues. For example, finding threshold value may not be very effective regarding some mathematical approaches. So there are more powerful development approaches that can be applied to AMS system to increase its efficiency on achieving monitoring functions.

Suggested further development approaches include Thresholding by minimizing fuzziness.

REFERENCES

[1] Gonzalez R. C. and Woods R. E. (3rd edition),(2008), Digital Image Processing, Prentice Hall.

[2] Krishna C.M. and Shin K. G.(International Edition), (1997) , Real time systems ,McGrow-Hill.

[3] Microsoft Corporation (2001), Microsoft Development Network: Visual C++ 6.0 tutorial, MSDN.

[4] Microsoft Corporation (1990), Microsoft Windows Guide to programming V 3 , Microsoft Press.

[5] http://www.cs.datamouth.edu/~farid

تطوير نظام مراقبة متقدم في بيئة زمن حقيقي

خليل الوجيه

كلية الحاسوب و تقنية المعلومات – جامعة ذمار

ملخص

تناقش هذه الورقة مسألة توفير نظام جيد للمراقبة بواسطة الحاسوب. وكتطبيق لذلك فقد تم تصميم وبناء نظام المراقبة المتقدم (AMS). يعد النظام المطور نظام زمن حقيقي للمعالجة الصورية لغرض تعزيز خدمات المراقبة بكلفة منخفضة وأداء عالٍ. يأتي هذا التطبيق ضمن مساحة بحثية واسعة لعلاقته بعدة مواضيع مهمة في علم الحاسوب وقد جرى التحقق من إنجاز الوظائف والخدمات المستهدفة للنظام بشكل موثوق.