World Applied Sciences Journal 4 (6): 781-786, 2008
ISSN 1818-4952
© IDOSI Publications, 2008

Real-time Human Tracking System with Dynamic Camera

[1]Ibrahiem M.M. El Emary, [2]Mazen Abu Zaher and [2]Rashad J. Rasras

[1]Faculty of Engineering, Al Ahliyya Amman University, Amman, Jordan
[2]Faculty of Engineering Technology, Albalqa' Applied University, Amman, Jordan

Abstract: Human tracking system plays a critical role in many applications such as surveillance and robot applications. Our earlier implementation [1] while quite successful was restricted to applications with approximately fixed camera. In this paper we present some recent work that removes this restriction. Such systems are required for machine vision from moving platforms such as robots, intelligent vehicles and unattended large field of regard cameras with a small field of view. Our approach is based on the use of a modified motion estimation technique coupled with our previous work in [1] for human tracking. This allows introducing an invariant human tracking system that is robust to background and lighting changes in addition to shadows remo val, handles unknown camera/ human motion and does not require special hardware components or camera calibrations.

Key words: Human tracking system . Dynamic camera . Jordan

INTRODUCTION

Human tracking system plays a critical role in many applications such as surveillance and robot applications. In the situations of tracking with moving camera, the tracking process become more complex and time consuming, especially with real-time implementations. From this fact comes the importance of this work to enhance the presented system in [1] to deal with moving camera in real-time (indoor or outdoor) performance using ordinary personal computer without the need for camera calibrations, or predefined conditions. We use one at time algorithm (OTA) motion estimation technique in modified fashion to accomplish our goal.

The rest of this paper is organized as follows. In Section 2, the design and implementation of the proposed human tracking system is presented followed by experimental results in Section 3, then the summary and future work in Section 4.

In this paper we will focus on the algorithm needed to handle moving camera. For invariant process and tracking we will not detail it; so for detailed information review the work in [1].

AN INVARIANT REAL-TIME HUMAN TRACKING SYSTEM

Introduction: The challenge in this paper is to create an invariant real-time human tracking system with

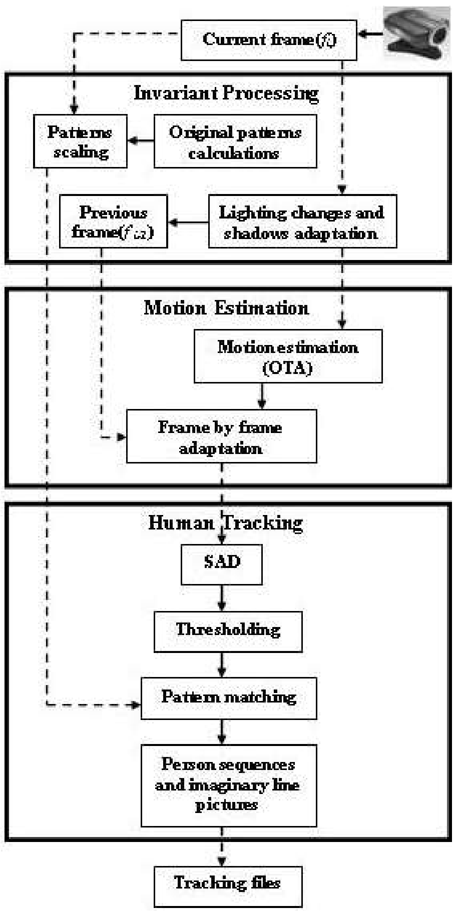

Current frame(f_i)
Invariant Processing
Patterns scaling
Original patterns calculations
Previous frame(f_{i-1})
Lighting changes and shadows adaptation
Motion Estimation
Motion estimation (OTA)
Frame by frame adaptation
Human Tracking
SAD
Thresholding
Pattern matching
Person sequences and imaginary line pictures
Tracking files

Fig. 1: System block diagram

Corresponding Author: Dr. Mazen Abu Zaher, Faculty of Engineering Technology, Albalqa' Applied University, Amman, Jordan

moving camera using an ordinary personal computer without any additional hardware, initial conditions, or assumptions.

Description of the proposed algorithm: A block diagram of the proposed human tracking system is shown in Fig. 1. The proposed system is divided into three parts: invariant processing, motion estimation and human tracking. These three parts can be described as follows:

Invariant processing: This part consists of three stages; the first stage is based on applying a modified power law transformation to each pixel in the captured frame; this operation compensates lighting changes and removes shadow from image sequences. The second stage is concerned with pattern design to just extract human from the scene and ignore other moving objects. In this stage a classification technique have been used to expand one pattern with different human body shapes in various scales of the same object. The third stage is frame by frame adaptation which have been used to overcome background change problems by using frame by frame method to extract changes between two image frames $f(x,y,t_i)$ and $f(x,y,t_j)$ taken at times t_i and t_j, respectively by comparing these two frames pixel by pixel. Using this method, we can adapt to different backgrounds. In addition, we save time and increase robustness against light changes which make this method suitable for real-time implementations. We move the third stage in [1] to be in motion estimation step to increase the speed and enhance the algorithm to deal with moving camera.

Motion estimation: This step handles camera movement to find the direction of camera motion. Using this technique we don not need any signal from the motor to indicate the direction of movement because this will be done automatically using motion estimation.

One idea is that a traditional technique can be applied here by calculating the relative motion of the moving object and that is done by assigning a reference point (or line) in the background at the starting of the application such as a line on the wall or the edge of the door in a room, but with this criteria we must take into account that this reference should still in the scene no matter how much the camera moves, in addition that we must pre initialize this reference and so we must test any new location to beck the reference manually which restrict the tracking process.

So we decide to use one of motion estimation algorithm than do not require any initialization. To perform the best selection of a motion estimation algorithm we restrict our search on block matching motion estimation algorithm since the basic human tracking system deals with block matching.

One of the most common techniques used for motion estimation is through block matching. The technique assumes a 2D translational model for each block and a single vector is found by minimizing the total pixel intensity difference. In practice, a distortion function is used to quantify the similarity between the source and target blocks. Due to the amount of data that needs to be processed, the selected distortion function should be easy to compute and result in good matching. Tow most commonly used matching criteria are mean absolute difference (MAD) and mean square difference (MSD). In our work we use MAD because of its simplicity and speed. MAD mathematically can be expressed by the following equation [2]:

$$\mathrm{MAD}(dx,dy) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left|F(i,j) - F(i+1,j+1)\right|$$

We decide to use one at a time algorithm (OTA) for motion estimation since this search algorithm requires very little time, which help in real time constraint. OTA is a simple, but effective way of trying to find a point with the optimal block. During the horizontal stage, the point on the horizontal direction with the minimum distortion is found. Then, starting with this point, the minimum distortion in the vertical direction is found. The algorithm may be described as follows [3]:

Step 1: Pick three points about the centre of the search window (Horizontal).

Step 2: If the smallest distortion is for the centre point, start the vertical stage, otherwise look at the next point in the horizontal direction closer to the point with the smallest distortion (from the previous stage) Continue looking in that direction till you find the point with the smallest distortion. (Going in the same direction, the point next to it must have a larger distortion)

Step 3: Repeat the above, but taking points in the vertical direction about the point that has the smallest distortion in the horizontal direction.

One particular search path for the algorithm is shown in Fig. 2.

For automatic motion estimation we select four 3×3 blocks; two at the upper and lower right corner of the frame and the others at the upper and lower left corner of the frame. We apply OTA to find the new

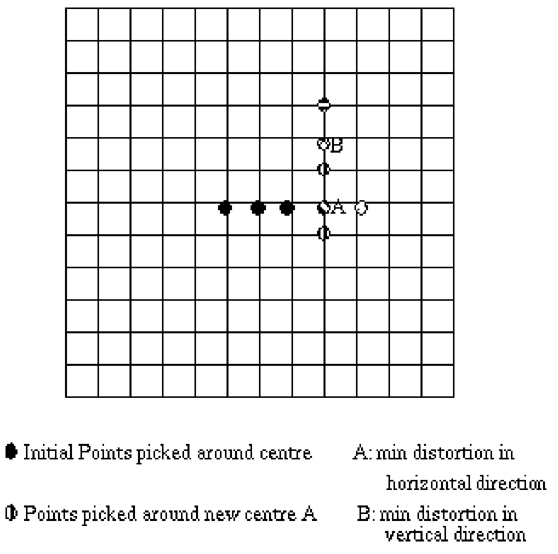

B
A
● Initial Points picked around centre
◐ Points picked around new centre A
A: min distortion in horizontal direction
B: min distortion in vertical direction

Fig. 2: Example path for convergence of one at a time algorithm [2]

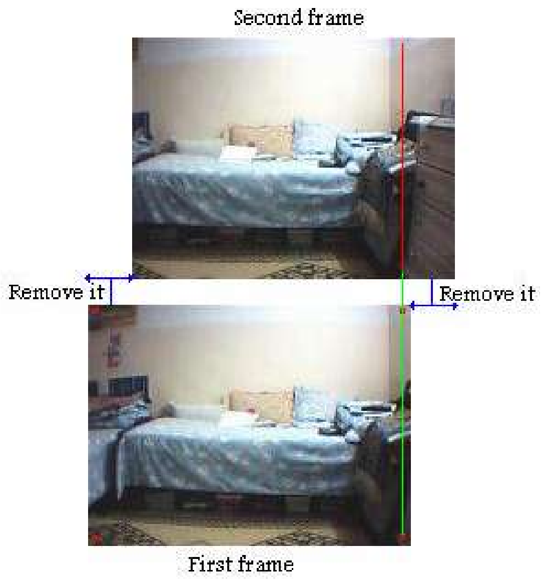


Fig. 3: Motion estimation using OTA

position for these four blocks and then extract the direction of camera movement. We select these four locations because they are on the frame boundary in addition that these two positions are less affected by pedestrian. According to the new location of the four blocks or one of them we truncate the two subsequent frames and perform frame by frame adaptation which described in the previous section. The next figures describe the algorithm.

If we take the case when the camera moves to left; the upper and lower left blocks disappear and OTA find the upper and lower right block. From the distance

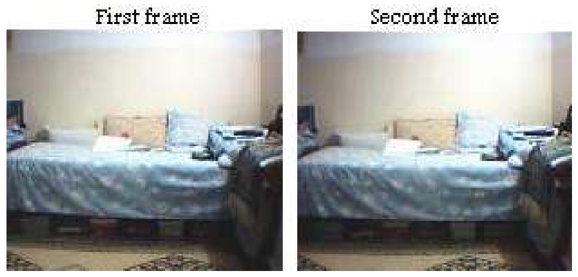
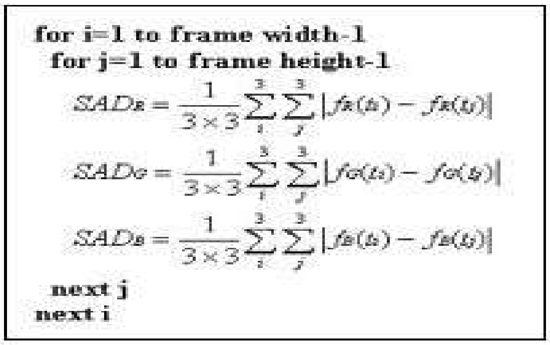


Fig. 4: Result after frames truncation



```
for i=1 to frame width-1
  for j=1 to frame height-1
    $SAD_R = \frac{1}{3\times 3}\sum_{i}^{3}\sum_{j}^{3}|f_R(t_i) - f_R(t_j)|$
    $SAD_G = \frac{1}{3\times 3}\sum_{i}^{3}\sum_{j}^{3}|f_G(t_i) - f_G(t_j)|$
    $SAD_B = \frac{1}{3\times 3}\sum_{i}^{3}\sum_{j}^{3}|f_B(t_i) - f_B(t_j)|$
  next j
next i
```

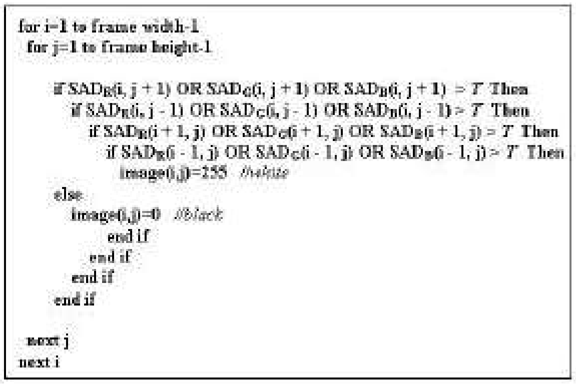
Fig. 5: Pseudo code for SAD calculation

between the locations of the upper and lower right bocks in the first frame and the second frame, we find the truncated area from left for the first frame and from right for the second frame (Fig. 3). The two frames after truncating shown in Fig. 4. Tracking process will be applied to these two frames and so on for the next two frames.

The same operation is done in the case of moving right, up, or down. As seen from Fig. 4 the aim of motion estimation step is to extract equal frames so that the only change in the scene in foreground objects. In Fig. 4 the truncated areas are large that's because the two frame taken at rate of one frame per second just to show the idea, but in the real implementation the frame rate is higher than that, so the truncated areas are very small. The output of this stage represents the input for the human tracking step.

Human tracking: This part presents the design of a human tracking algorithm that is compatible with the presented invariant image processing algorithm and motion estimation process. This part is subdivided into four stages. In the first stage a difference image between two images taken at time t_i and t_j is defined using sum of absolute difference (SAD) for each RGB color channel. This process shown in the next figure.

The second stage deals with thresholding where the resulting difference image from the SAD step is used to extract moving objects in contiguous frames.



```
for i=1 to frame width-1
 for j=1 to frame height-1

    if SAD_R(i, j + 1) OR SAD_G(i, j + 1) OR SAD_B(i, j + 1) > T Then
      if SAD_R(i, j - 1) OR SAD_G(i, j - 1) OR SAD_B(i, j - 1) > T Then
        if SAD_R(i + 1, j) OR SAD_G(i + 1, j) OR SAD_B(i + 1, j) > T Then
          if SAD_R(i - 1, j) OR SAD_G(i - 1, j) OR SAD_B(i - 1, j) > T Then
            image(i,j)=255  //white
    else
      image(i,j)=0  //black
          end if
        end if
      end if
    end if

 next j
next i
```

Fig. 6: Pseudo code of thresholding process

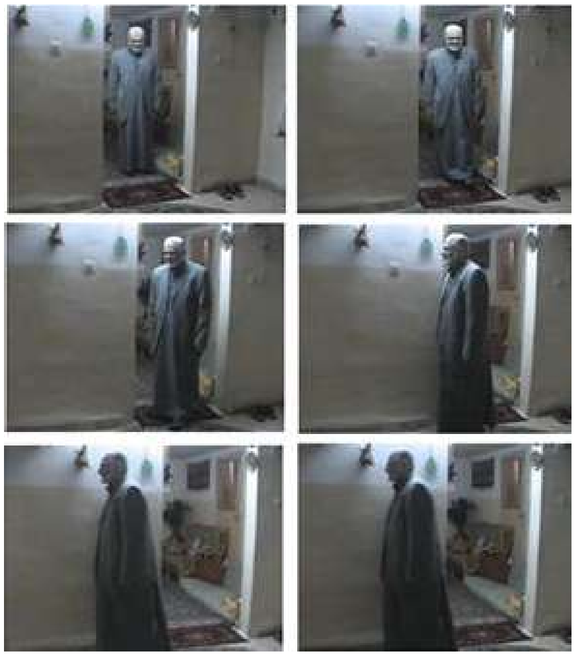


Fig. 7: Tracking sequence

Thresholding of this difference image yields a binary image containing “foreground” regions where movement was detected. Threshold value, *T*, was selected to be 50 using subjective evaluation after various studies and experiments to lighting and shadow properties. Figure 6 shows the Pseudo code of the thresholding process.

At the end of this step the output frame represents moving objects in white and background objects in black. Here, we called the resulting image binary image.

The third stage concerned with recognizing human within a frame according to a person model which compared with the output of the thresholding process, this increases speed and saves time in performing matching to the entire frame using SAD.

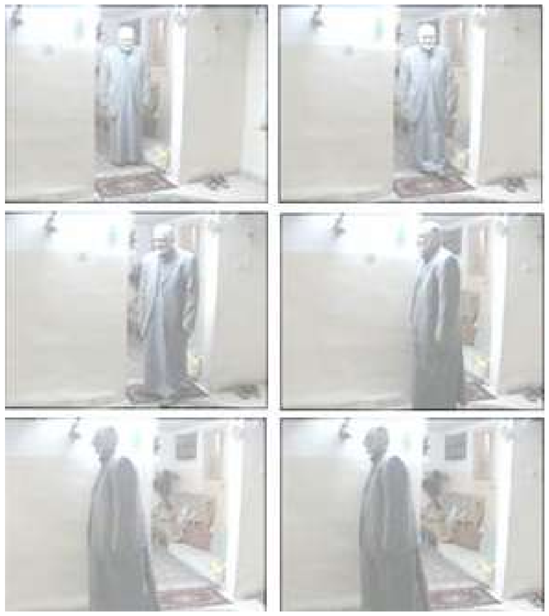


Fig. 8: Results of invariant processing

The last stage concerned with creating a special file containing all paths for every tracked person. Within the tracking process, if the person enters predefined imaginary line (to make sure that the person stands in good position to take his face detail) the system will capture a picture for that person, which can be used in any face recognition technique.

EXPERIMENTAL RESULTS

Implementation issues: We have implemented our system with two camera types: Panasonic NV-DS65 and Kinstone PC CAMERA.

Our system is implemented using a personal computer Pentium(R) 4 with the following specifications: CPU speed 1.70GHz with 256 MB of RAM.

Our system has been implemented using visual basic 6 (VB6). To increase our system speed in VB6, we use multithreading with six threads, one for the invariant processing and four for motion estimation process for each of the four blocks and one for the tracking process.

In the case of human tracking real-time performance will be more than established using three frames per second as stated in [4]. This is because in the case of human motion no large displacements occur within a time that is less than half second. In our system each captured frame is of size 176×144 pixels. This size is selected to help in noise elimination and increases the system speed. On the other hand frame of size 320×240 pixels can be used with speed of 65 *ms* per frame. It can be seen that our system speed can be improved using faster personal computer (more than 1.70GHz).

RESULTS

For tracking experiments different image sequences in different environments and conditions

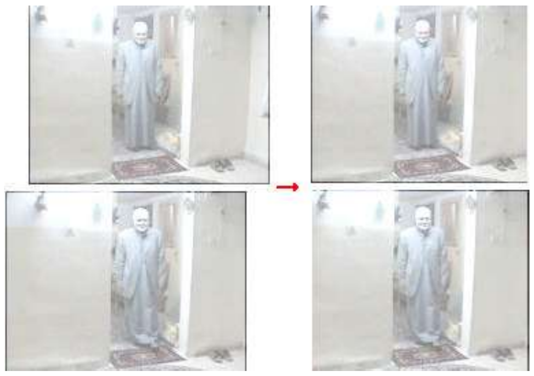


Fig. 9: Results of motion estimation processing frame1 and frame2

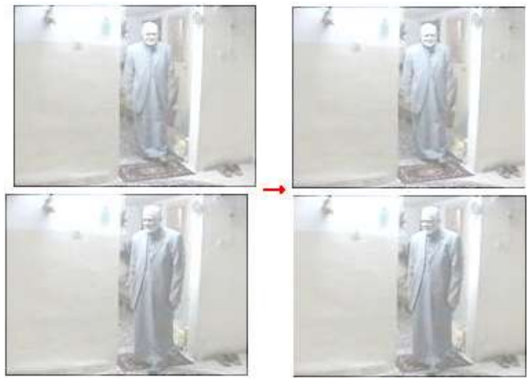


Fig. 10: Results of motion estimation processing frame2 and frame3

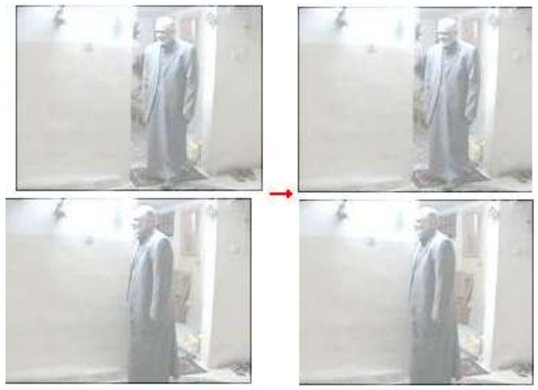


Fig. 11: Results of motion estimation processing frame3 and frame4

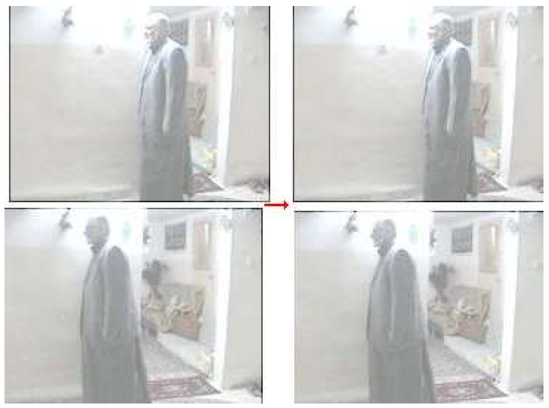


Fig. 12: Results of motion estimation processing frame4 and frame5

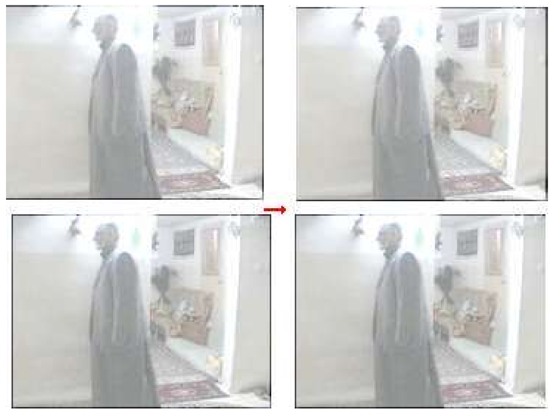


Fig. 13: Results of motion estimation processing frame5 and frame6

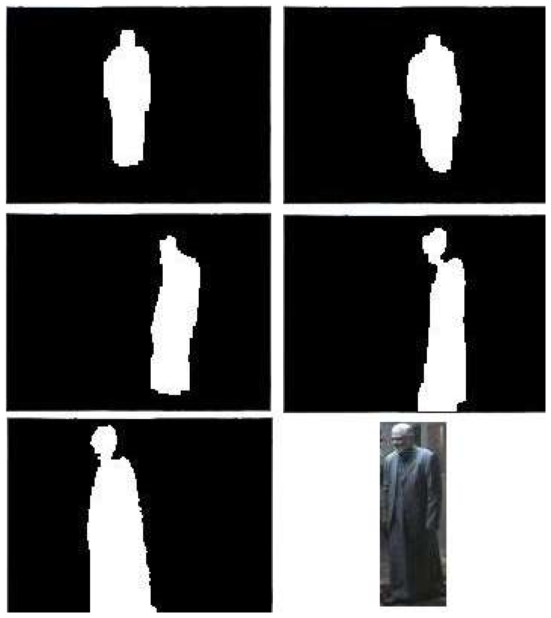


Fig. 14: Tracking results

were selected. The result of tracking process is shown as a binary image (black and white). Black area represents background and stationary objects; white regions represent the tracking person. In each tracking sequence, a picture is attached which is captured for the tracked person when that person enters the predefined imaginary line. We select the following sequence from our experimental results, which taken with moving camera to right and to left.

CONCLUSION AND FUTURE WORKS

In this paper, we have developed a framework for tracking humans from moving camera. Our approach used the OTA motion estimation algorithm combined with our tacking system in [1]. As a result, we could handle general situations where there are no restrictions on the dynamics of the camera or the human being tracked and there are no assumptions on the viewing angle. As a future work and to enhance the system, we will try to include real time face recognition mechanism.

REFERENCES

1. Mazen Abu_Zaher, Rashad J. Rasras and Ibrahiem M. El Emary, 2007. Building an Advanced Invariant Real-Time Human Tracking System. ComSIS.
2. Professor GZ Yang, “Motion Picture Processing”, Multimedia University, http://www.doc.ic.ac.uk/~gzy.
3. Deepak Turaga and Mohamed Alkanhal, 1998. Search Algorithms for Block-Matching in Motion Estimation. Carnegie Mellon University.
4. Siebel, N. and S. Maybank, 2001. Real-Time Tracking of Pedestrians and Vehicles. In the Second IEEE International Workshop on Performance Evaluation of Tracking and Surveillance.