A Query Answering Surveillance System for Detecting and Tracking Moving Objects using Bounding Rectangles

Azzam Sleit¹, Wesam Almobaideen², Mohammed Qatawneh³, Ala’a Barakat⁴, Heba Saadeh⁵
P.O. Box 13898
Department of Computer Science
King Abdullah II School for Information Technology
University of Jordan
Amman 11942
Jordan
¹azzam.sleit@ju.edu.jo, ²wesmoba@ju.edu.jo, ³Mohd.qat@ju.edu.jo
⁴alabarakat@yahoo.com, ⁵blue_way85@yahoo.com

ABSTRACT: In many computer vision applications such as surveillance systems, detecting moving objects is the first step of information extraction. This paper introduces a query answering system that aims at responding to user queries without the need to scan the whole video tape. The proposed system is based on finding the minimum boundary rectangles (MBRs) of moving objects which appear in the video. It detects moving objects in the video frame set, keeps track of those objects, and saves information to answer future queries.

Categories and Subject Descriptors
I.4 [Image Processing and Computer Vision]; I.2.10 [Vision and Scene Understanding]; Video analysis; H.5 [Information Interfaces and Presentation]; Video

General Terms: Computer vision applications, Query answering systems

Keywords: Surveillance system, MBR, Object tracking, Background subtraction, RGB Difference.

Received 8 May 2008; Revised and accepted 12 July 2008

1. Introduction

Video parsing and retrieval techniques have been of much interest for researchers over the past two decades. Researches have shifted interest gradually from fixed image analysis based on object segmentation into video shot boundary detection, content based analysis, and object tracking. Initial research activities in the video analysis field focused on low-level features such as colors while many researches nowadays concentrate on high-level features (i.e. object features). Techniques that depend on low-level features tend to be simple and easy to implement.

In a traditional surveillance system, video cameras record scenes for specific purposes. By the end of the recording process, video tapes may hold many useless scenes such as empty room. The desire to seek for a specific piece of information may require parsing the whole video tape. The task of interpreting these high-level videos is usually performed by human operators, who have to sequentially view large amounts of visual information presented to them through one or several monitors [1]. This approach is extremely time-consuming and may be frustrating.

This paper presents a query answering solution for video-based surveillance systems in a controlled indoor environment. The proposed technique for object identification and classification depends on utilizing low-level features of the video frame set such as binary difference, and morphological functions. The intention is to detect objects using their enclosing minimum bounding rectangles (MBRs) and keep track of the movement of MBRs along the entire tape. The system consists of three crucial modules as shown in Figure 1. The system accepts a colored or gray level video tape along with a reference frame “background image” as an input, and splits the video into frames.

The Find_MBRs module detects all moving objects and encloses each of them by a rectangle which will be dealt with as a representative of the object. The Identify_and_Track_Objects module focuses on tracing the movement of detected objects across the frames. Accumulated information is utilized by the Answer_Queries module. This paper is organized as follows. Sections 2 presents related work. Section 3 discusses the proposed system and relevant algorithms. Section 4 displays our system implementation interfaces. The article concludes in section 5.

2. Related Work

Video surveillance systems with detection functions usually include a module aiming at detecting relevant changes in the guarded environment [1] [2] [5]. If the output of the change
Object segmentation is one of the approaches used to identify objects, keep track of their motion, and answer queries. The segmentation process is best suited for fixed images but not for video tapes with thousands of frames. This task requires massive amount of computations and may not be practical for videos. In surveillance applications, functionalities like detecting abnormal or dangerous objects for alarm generation require video content analysis. Stringa and Regazzoni presented a surveillance system to detect abandoned objects and highlight the people who left them in an indoor environment. In this system, foreground objects can be easily segmented since the background scene information is static and the camera is mounted at fixed location. A multilayer perception is trained offline to classify the foreground objects as abandoned objects, person, lighting or structural changes. By motion analysis, the person who leaves an abandoned object is highlighted. The shape and color information of the object will be indexed for future reference.

Several researches presented algorithms for scene change detection based on statistical soft morphology (SSM) which can be implemented as binary SSM instead of binary filters. It is also possible to apply SSM filtering before the binarization step which makes the system more stable with respect to the threshold settings. Unsupervised segmentation can be utilized for video shot detection. A survey prepared by Yilmaz, Javed and Shah characterized Point tracking as the correspondence of detecting objects represented by points across the frames. Kernel tracking is typically performed by computing the motion of the object, which is represented by a primitive object region from one frame to the next, whereas, silhouette tracking provides an accurate shape description for complex objects. Most detection algorithms utilize frame difference, background subtraction, optical flow and statistical learning methods. Statistical learning requires a training phase. Likewise, Optical flow is complex and time consuming. Because of the computational time and complexity of the preceding two methods, they are not suitable for real-time applications. Background subtraction and frame difference are more popular to be utilized for detection algorithms. Chaohui et al. introduced a motion detection algorithm based on frame difference and edge detection. Since the edge has no relation with the brightness, the moving edge method can effectively suppress the noise caused by light. The proposed algorithm applies Canny’s edge detector (frame K and preceding frame k-1), takes the difference between the two edge images, divides the difference image into several small blocks and decides if a specified block is a moving area according to a certain threshold. Finally, it takes the smallest rectangle that contains the moving object by applying the block-connected component labeling algorithm.

3. System Structure and Algorithms

Geometric shapes such as polygons and circles can be used to approximate the dimensions of a pictorial object. Along with other features, the dimensions and location of the bounding geometric shape can be utilized to identify the object. The minimum bounding rectangle (i.e. MBR) of an object is one of the most commonly accepted bounding container shapes in the imaging and spatial databases research community. This is due to the simplicity of rectangles when utilized in the recognition process, in addition to the fact that most multi-dimensional indexing structures are based on rectangles.

The proposed technique aims to provide the human operator of a surveillance system with the following information:

- Only frames which contain moving objects.
- Number of moving objects in a frame.
- Time at which a moving object appears / leaves the scene.
- Frames which have moving objects of the same type.

The classical method to obtain the bounding rectangle of an object is to raster scan the object in the image and to subtract all the pixels of the object from the image in order to find the pixels with the minimum X, minimum Y, maximum X and maximum Y coordinates. Also, recent work for finding bounding boxes in textual document images is based on skew estimation which requires multiple passes. Figure 2 demonstrates the process flow diagram for the Find_MBRs module which is responsible for browsing the frame set and drawing the MBRs of the moving objects in a video stream.

The Find_MBRs module accepts a reference frame A which only contains the background of the scene with no moving objects and a current frame B which may or may not contain moving objects. We have found that reducing the frame sizes by 50% enhances the computational speed of the system with no impact on accuracy. The RGB-Difference function between the current and reference frames is given in Figure 3 to segment objects which appear in the current but not reference frame. It converts the difference image into black and white and returns the difference image in the output array BW based on a threshold value.
The choice of the threshold value $th$ is critical. If $th$ is selected to be small, a large number of pixels will be detected as change. If it is selected to be large, pixels belonging to a particular object could be lost. Figure 4 shows an example of change detection obtained with different threshold values. You can notice from this figure that the results vary depending on different threshold settings.

The proposed RGB-Difference function prepares the $BW$ image for the application of morphological functions such as image smoothing to eliminate noise and image dilation to fill in the gaps between an object’s parts which may result from the smoothing process. It is essential to observe that the RGB-Difference function may cause some holes to appear in the objects which can be filled with white pixels.

The last step in the Find-MBRs module is to draw the minimum bounding rectangles which enclose the moving objects. The Draw_MBR function is described in Figure 5.

The idea of the Find_MBRs module is to enclose each object with the minimum boundary rectangle which will be the representative of the object in the next steps. Figure 6 demonstrates examples of frames after applying the RGB-Difference followed by the Draw_MBRs function.

As the last step of Draw_MBRs, we manipulate the overlapping rectangles by merging them together as per Figure 7. Overlapping rectangles may appear because of illumination.
or shadows which tend to separate an object into smaller objects. The bounding rectangles merge function minimizes the effect of illumination.

Figure 7. (a) Overlapping bounding rectangles. (b) Corresponding merged rectangles

Once the dimensions of an MBR have been determined, we can identify and trace the corresponding object in the video stream. For a specific camera in a surveillance system, a video stream has one reference frame because each stream refers to one specific scene. The Identify_and_Track_Objects module identifies the class to which an object belongs; e.g. human, bird, car, ...etc. The first step is to determine if the dimensions of the MBR are similar to an MBR that appeared in the previous inspected frame with possible offset taking into consideration the object’s average speed and the direction of its motion in previous frames. If a new object is detected which has not appeared in the previous inspected frame, information will be saved to be used later during future identification and query answering phases. This information includes object identification (Object_id) used to track the object along the stream, MBR coordinates, and time of object’s appearance using SMPTE standard (Society for Motion Pictures and Television Engineers). When a new object first appears in the stream, the object’s class is determined using domain specific heuristics to classify the object depending on the surveillance domain, potential object types and position of the camera. Table 1 demonstrates a rule-based decision scheme which accepts the coordinates of the MBR (minX, minY, maxX, maxY) as input and generates a corresponding classification as output using a similarity measure. In this research, we used the Euclidean distance measure. The following record is registered in the Objects table:

(Object_id, Class), where Object_id is a sequentially system generated integer, and Class is the object type. Also, for each appearance of a moving object in an inspected frame, a record of the following format is registered in the Motion table:

(Object_id, MBR_Coordinates, Time), where MBR_Coordinates is a quadruple of the format (minX, minY, maxX, maxY) and Time is hours:minutes:seconds:frames.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Class A</th>
<th>Class B</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition 1</td>
<td>H_{AL} &lt; H &lt; H_{AH}</td>
<td>H_{BL} &lt; H &lt; H_{BH}</td>
<td></td>
</tr>
<tr>
<td>Condition 2</td>
<td>W_{AL} &lt; W &lt; W_{AH}</td>
<td>W_{BL} &lt; W &lt; W_{BH}</td>
<td></td>
</tr>
<tr>
<td>Condition 3</td>
<td>R_{AL} &lt; R &lt; R_{AH}</td>
<td>R_{BL} &lt; R &lt; R_{BH}</td>
<td></td>
</tr>
<tr>
<td>Condition 4</td>
<td>Y_{AL1} &lt; Y &lt; Y_{AH1}</td>
<td>Y_{BL1} &lt; Y &lt; Y_{BH1}</td>
<td></td>
</tr>
<tr>
<td>Condition 5</td>
<td>Y_{AL2} &lt; Y &lt; Y_{AH2}</td>
<td>Y_{BL2} &lt; Y &lt; Y_{BH2}</td>
<td></td>
</tr>
<tr>
<td>Condition 6</td>
<td>X_{AL} &lt; X &lt; X_{AH}</td>
<td>X_{BL} &lt; X &lt; X_{BH}</td>
<td></td>
</tr>
<tr>
<td>Condition 7</td>
<td>X_{AL} &lt; X &lt; X_{AH}</td>
<td>X_{BL} &lt; X &lt; X_{BH}</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Proposed Object Classification Scheme

In order to detect the existence of a certain object and to avoid the effect of illumination and shadow spots, we process the first frame of each six frame block and skip the other five consecutive frames. In this way, we reduce the number of false object detections as well as the number of frames for which the object tracking process will be applied. For a frame rate of 25 fps, 4 frames will be analyzed per second.

While tracing a specific object along the video, we scan the areas that surround the object in the next frames to identify the direction of its movement, calculate the new coordinates of its MBR, and then shift the MBR to the new position. Object tracking continues until the object disappears by leaving the frame from one of its boundaries.

Using the Objects and Motion tables which are populated during the execution of the Identify_and_Track_Objects module, the query answering module will be able to answer inquiries such as the following:

- Return all frames which contain a specific number of objects from a particular class.
- Return all frames which contain a specific object.
- Return the frames in which an object appears and/or disappears.
- Return the objects which appear in a specific window.

It is also possible to utilize spatial indexes such as R-tree and SB+-tree to enhance the performance of the query answering module. This is particularly important for systems with a large number of frames.

4. System Implementation

We implemented the modules in Figure 1 using MATLAB. The first interface, as shown in Figure 8, corresponds to the Find_MBRs module with several buttons to implement the steps in Figure 2.
The second interface, as shown in Figures 9 and 10, is designed to implement the Identify_and_Track_objects module. It displays the sequential frame number and the number of objects in each frame.

The third interface is shown in Figure 11. It allows the user to pause queries such as the ones listed in the previous section.

5. Conclusion

This research proposes a system that depends on analyzing the minimum bounding rectangles of moving objects. The system enables users to obtain needed information through an efficient, simple, and user friendly interface. The underlying algorithms are studied and evaluated in a controlled indoor environment. The advantage of this system is that it is customizable to meet surveillance domain specific parameters. Furthermore, various similarity measures can be utilized depending on the nature of the objects and scenery. Future work will follow to compare and contrast various parameters and similarity measures.

References


---

Azzam Sleit is an associate professor with the Computer Science Department, University of Jordan. His research interest includes imaging databases, information retrieval, and distributed systems. Before Joining the University of Jordan in 2005, Dr. Sleit was the Chief Information Officer at Hamad Medical / Qatar’s Ministry of Public Health where he developed and executed the information technology strategy of healthcare. Dr. Sleit has over sixteen years of experience and leadership in the information technology field including work at all levels of government, private and international sectors. Before joining Hamad Medical, Dr. Sleit was the Vice President of Strategic Group & Director of Professional Services of Triada, USA. Dr. Sleit served as the Director of Professional Services at Information Builders, USA. Dr. Sleit holds B.Sc., M.Sc. and Ph.D. in Computer Science.

Wesam AlMobaideen was born in Amman, Jordan. He received his Ph.D. in computer Networks from the University of Bologna, Bologna Italy in 2003, Master degree in computer science from the University of Jordan in 1999 and B.Sc. degree in Computer Science from Mu’ta University, Karak, Jordan in 1997. He is currently an Assistant Professor of Computer Science Department at the University of Jordan. He held several administrative positions including the Assistant Dean of King Abdullah II School for Information Technology.

Mohammed Qatawneh is an associate professor and Chairman of the Computer Science Department at the University of Jordan. He received his Ph.D. in Computer Engineering from Kiev University in 1996. Dr. Qatawneh published several papers in the areas of parallel algorithms, Networks and Embedding systems. His research interests include Parallel Computing, Embedding System, and Network Security.

Ala’a Barakat is a graduate student with the Computer Science Department, University of Jordan. Her research area of interest is image processing.

Heba Saadeh is a graduate student with the Computer Science Department, University of Jordan. Her research area of interest is image processing.