DETECTING STATIC OBJECTS IN BUSY SCENES

Jia Wang [1] *and Wei-Tsang Ooi* [2]

Department of Computer Science, Cornell University, Ithaca, NY 14853-7501, USA

ABSTRACT

Detecting static objects in scenes containing significant number of moving objects has several applications in video surveillance. One example is the detection of suspicious packages which is left unattended in an airport terminal or railway station. This paper outlines an approach to automatically detect static objects from a video sequence of a busy scene. Our approach consists of two phase : foreground object extraction and object matching. In the first phase, we find the foreground objects in current video frame, using an image of a background as reference. In the object matching phase, we try to match the objects with objects that appears before in previous frames. Matching is done based on three parameter : shape and position, intensity and edge. Temporary occluded of objects is also handled. We built a system based on our approach. Preliminary experiments shows that our system are able to identify static objects in a busy scene in real time.

1. INTRODUCTION

Static object detection has not been well studied before. This problem has several significant application. One example is the surveillance system in airport. It will help to detect objects that are left unattended. If static objects are highlighted on the surveillance screen, it becomes much easier for the security people to find suspicious objects. In this paper, we present a simple yet effective approach for detecting static objects in a busy scene in real time. We defined static objects as objects that does not move, or moves very slowly, for a period of time in the scene.

The basic characteristic of the scenes described above is that they contain a number of significant objects moving in the scene. This poses a technical challenge as static objects can be occluded by other objects. Another challenge is that the detection system should operate in real time, and therefore the detection method should be simple, but yet produce few false alarms.

A work has been done to solve the similar problem using statistical method [4].This paper presents another approach. Our detection process involves two steps: *foreground object detection* and *objects matching*. The first step uses the conventional image differencing techniques to extract all foreground objects. The second step then uses three simple matching criteria to match objects from current video frame to objects from previous video frames. Section 2 describes how the foreground objects are extracted. Section 3 issues the methods we used in detection of static objects in the scene. Section 4 gives an example and the results of its execution on the example image sequence. Finally, Section 5 draws the conclusion and states some possible future works.

2. EXTRACTION OF FOREGROUND OBJECTS

We use a simple image differencing method to find foreground objects in current frame. However to do so an image of the background is needed. Due to the busy characteristic of the scene, the background is occluded frequently by some foreground objects. This makes background estimation techniques ineffective. However a snapshot of the background can be obtained easily in surveillance environment (For example, a security guard can take the snapshot early morning in the airport, when nobody is around). Therefore we assumes that an image of the background in the scene is readily available.

To extract the foreground objects from the current frames, we first find the difference image between current frame and the background image by subtracting current frame from the background and then take the threshold value. The resulting image is then smoothed by applying morphology operation *open* and *close* [1, 5, 7]. The result is a bitmap with "blobs" of objects. We then find the connected components of the blobs, which is represented as a list of bounding boxes.

Each bounding box represents some foreground ob-

1. This material is based upon work supported under a National Science Foundation Graduate Fellowship.

2. This research is supported by DARPA/ONR (contract N00014-95-1-0799), and grants from the National Science Foundation, Kodak, Intel, Xerox, and Microsoft.

jects in the scene. We ignore boxes that are too small since we are not interested in tiny objects.

Note that each bounding box can contained more than one real objects in the scene. However we treat each box as an object. This does not affect our result for the following reason : If the box contains only static objects, all the static objects can be identified as one static objects. If the box contains only dynamic objects, we are not interested in it anyway. If the box contains some static objects and some dynamic objects, the dynamic objects will move away eventually in the future frames so it does not affect our algorithms.

3. DETECTION OF STATIC OBJECTS

After we extract the list of foreground objects from current frame, we have to compare those objects with objects from previous frames. Matching is done based on three criteria : (1) similarity in shape and position: which computed by intersecting the bounding boxes of two objects and count the percentage of black pixels in the intersection. (2) similarity of intensity : computed by finding the sum of square differences in intensities of two objects. (3) similarity in edges : computed by first detecting the edge of the objects and then perform Hausdorff matching on the two images [6]. We say two objects *match well* if they perform excellently in these similarity tests.

To detect static objects, each objects in the system has 2 attributes : `age` and `type`. `age` records the number of frames the object hasn't move. `type` represents the status of the objects. It can be either *new, matched* or *occluded.*

Let O_{curr} and O_{prev} be the set of objects from current frame and previous frames respectively. The `type` attribute is assigned based on these criteria. If an object from O_{prev} matches some object from O_{curr}, it is a *matched* object. If it does not match well with any objects from O_{curr} frame, and does not overlap with any objects from O_{curr} , we assume the object does not appear in the new frame, and is not occluded by other objects, so it must have moved out of the scene. In this case the object is removed from O_{prev}. If it does not match well, but major portion of the object overlaps with some object in O_{curr}, it is a *occluded* object. The rest of the objects from O_{curr} frame which have not been matched by any objects from O_{prev} are considered as *new* objects and are added into O_{prev} before we start processing the next frames. The `age` of an object increment for every frame considered (even when it is occluded). Any *matched* object that has `age` more than a certain threshold is considered static object.

4. EXAMPLE AND RESULTS

In this section, we consider the execution of our detection system on the example video sequence. The video sequence we used are recorded in the Campus Store of Cornell University. It is about 3 minutes long and are digitized to 320x240 frames and running at one frame per second. Our system can process the frames at about 3 frames per seconds on a Sparc 5 workstation, which is faster than the input video frame rate. In our implementation, we categorized an object as static objects if it has `age` > 5. Bounding boxes of size less than 10 by 10 is ignored.

Figure 1 shows a snapshot of the background for our system. Note that there are already some bags laying on the shelve in the scene. Since they are in the background image, they should not be detected as static objects. Figure 2 - 4 shows the result of execution on our sample video sequences. Detected static objects are highlighted in its bounding box. The person in Figure 2 is detected because she stands there for at least 5 seconds without moving. A bag and a small suitcase is detected in Figure 3. Note that they are considered as one static objects since they are adjacent to each other. Somebody removes the suitcase moments later. The bag is then detected as a static object. This is shown in Figure 4.

Figure 1: Background snapshot

5. CONCLUSION AND FUTURE WORKS

We have outlined a technique for solving the problem of static objects detection in a busy scene. The technique is based on the object matching in the video sequence. A system have been built based on this technique. It runs in real time and produces excellent results in our video sequences.

One of the minor problem of our system is that a slight jitter in the video sequence can generate false alarms. An example is shown in Figure 4. A part of the

Figure 2: A person is detected as static object.

Figure 3: A static bag and suitcase is detected.

Figure 4: The bag is still highlighted as static object.

ceiling is highlighted as a static object when our camcorder that records the scene moves a little. However in real airport surveillance camera which is mounted on the wall or ceiling, this is very unlikely to happen.

We plan to run our system on longer sequences with higher frame rate in the coming months, and extent our system to use color sequences instead of gray sequences. Using color information will helps in the matching process as currently two objects with same gray intensity but different color will match each other, but they should not be. We also plan to port our system to Intel Pentium II platform running Windows NT. This should increase the processing rate of our system dramatically.

6. REFERENCES

[1] Dana H. Ballard and C. M. Brown, Computer Vision, Prentice Hall, 1982.

[2] Patrick Bouthemy and E. Francois, Motion Segmentation and Qualitative Dynamic Scene Analysis from an Image Sequence, International Journal of Computer Vision (1993) 10:2, p157-182.

[3] Jonathan D. Courtney, Automatic Video Indexing via Object Motion Analysis.

[4] D. Gibbins, G. N. Newsam and M. J. Brooks, Detecting Suspicious Background Changes in Video Surveillance of Busy Scenes.

[5] Rafael C. Gonzalez and R. E. Woods, Digital Image Processing, Addison-Wesley, 1992.

[6] Daniel P. Huttenlocher and W. J. Rucklidge, A Multi-Resolution Technique for Comparing Images Using the Hausdorff Distance.

[7] Robert M. Haralick and L. G. Shapiro, Computer and Robert Vision, Volume II, Addison-Wesley, 1993.

[8] Amar Mitiche and P. Boutheny, Computation and Analysis of Image Motion: A Synopsis of Current Problems and Methods, International Journal of Computer Vision (1996) 19(1), p29-55.

[9] J. Serra, Image Analysis and Mathematical Morphology, Academic Press, 1982.

[10] Yuan-Liang Tang, and R. Kasturi, Tracking Moving Objects During Low Altitude Flight, Machine Vision and Applications (1996) 9: p20-31.

[11] Tina Yu Tian and M. Shah, Motion Estimation and Segmentation, Machine Vision and Applications (1996) 9: p32-42.