ACCENT JOURNAL OF ECONOMICS ECOLOGY & ENGINEERING

Peer Reviewed and Refereed Journal (International Journal) ISSN-2456-1037

Vol. 05, Special Issue 01, (ICOSD-2020) January 2020, Available Online: www.ajeee.co.in/index.php/AJEEE

SURVEILLANCE OF CRIMINAL FOR SECURITY USING FACE DETECTION AND MOVING OBJECT

Ekta Sahu[1], Khushboo Sawant[2]

M. Tech Scholar, Department of Computer Science and Engineering, LNCT Indore (India)[1]

Assistant Professor, Department of Computer Science and Engineering, LNCT Indore (India)[2]

Abstract- An algorithm of surveillance video thesis is presented in this paper. At first, the detection method of moving object for surveillance video is proposed. Then the ROI (Region of Interest) of moving object is extracted and a no interactive matting algorithm of moving object based on Grab Cut is presented. These matting moving objects are placed without overlapping in a background image, so a frame that several moving objects have been placed in a background image can be obtained. Finally, a series of these frame images can be obtained in timeline and a single camera surveillance video synopsis can be formed. The experimental results show that our detection method of moving object is superior to those methods such as Gaussian mixture model, background subtraction, frame difference method and Li's method. The synopsis obtained contains all objects appeared in surveillance video and is a concise expression for original video.

Keywords: Security and Privacy, Criminal Identification; CCTV; facial recognition, Open CV.

1. INTRODUCTION

This chapter gives an overview and objective of the dissertation. It also states the problem in the existing system and provides a solution to it.

1.1 Overview

With the development of social economy and computer technology, the demand of social public security is also proposed. Many surveillance cameras are installed in many big public places and used to monitor and record the state of the places, so a massive video data is formed. If we need query and retrieve the recorded video data, the traditional forward and backward method can be used to browse the surveillance video, but it is time-consuming and may omit the important information. It has become a hotspot and difficult problem for the researchers to describe briefly, browse, query and analyze the video content in video monitoring field.

Video synopsis is a video summary technology that concentrates the original video information to the greatest extent by analyzing and processing a long video file and extracts the key information. The traditional video summaries only compress the video content in timeline. However, video synopsis compresses the video content not only in time but also in space. The length of video synopsis is shorter than that of the original video and the storage space of surveillance video can be greatly saved. Meanwhile, the basic content of the original video is retained and the user can quickly browse and search the video content.

The video synopsis technology has the broad application prospects and the important research significance. The surveillance video synopsis technology based on object can be divided into two categories. One is the conventional video summarization method that the surveillance video is divided into a so-called shots (the actual surveillance video has not the shot), and then the key frames are extracted to form a video summary. Another is a new video analysis method proposed in recent years. The frame image can be changed by this new method while the frame image as the minimum basic unit of video can not be divided in the traditional video summarization technology, so the length of the video synopsis is further shortened and the storage space is significantly saved.

The representative methods are the

video montage technology and video synopsis technology. Y Pfitch, A Rav-Acha, S Peleg proposed the overall framework of a video synopsis technology in their papers. It is a video to video conversion mode. In recent years, Teng Li et al. have proposed a multiple video summary method in which many auxiliary videos are placed in a main video in timeline. Yael Pritch et al. have proposed a video summary using the similar activity clustering method. The human eye focus position of surveillance video operator has been used to obtain a real-time surveillance video synopsis presented by U. Vural.

Uros Damnjanovic has adopted a method of detecting and clustering the important events of surveillance video to form the synopsis, and the so-called important event here is the moving object with more energy change in successive frames. Xu Min has also presented a video synopsis technology based on target motion information. In this paper, our algorithm of video synopsis is different from those methods mentioned above. After background modeling and detecting of moving object, the ROI of moving object is extracted and a no interactive matting algorithm of moving object based on Grab Cut is used to obtain the matting moving object. These matting objects are placed without overlapping in a background image and a series of these frame images can form a single camera surveillance video synopsis.

2. LITERATURE SURVEY

2.1 Title – Selection And Update Of Background Image

Introduction

- **2.1.1** The surveillance video synopsis is a procedure of processing the recorded video. In order tom prove the efficiency of generating video synopsis, the surveillance video will be segmented into a blank section without motion information and motion section with motion information according to the features of monitoring video.
- **2.1.2** Video segmentation can be performed by the histogram difference of two adjacent frames. When the difference value of histogram is less than a predetermined threshold, the frame is in blank section and there is not the moving object between the two frames. If the difference value is greater than the threshold, the motion section begins. When the difference value is less than the threshold again, the motion section is ended and the blank section starts. In the blank section, the posterior frame can be used as the background image. The background image of motion section can use this background image as the initial background image, and then the Gaussian mixture model is utilized to model and update the background image.
- **2.1.3** Fig. (1) is an example. The A and C are the video section without containing the moving object and the section B contains the moving object. After background modeling for the A, B and C respectively, the background images of A and B video clip can be compared. If the difference is within a certain threshold range, the change of background image in two video clips is little. The background image of video clip A is selected as the background of A and B. If the difference exceeds the threshold value, the change of two video background images is large. Then the background image of B is compared with that of C. If the difference is within a certain threshold range, the background image of video clip C is selected as background. If the difference exceeds the threshold value, the background image of video clip B is different with A and C, then the background image of B is chosen as the background. The background image of each motion section is selected by this method for the whole surveillance video.

2.2 Title- Detection Of Moving Object

Introduction

- **2.2.1** The detection of moving object is a foundation of surveillance video analysis. Many algorithms have been proposed for object detection in video surveillance application such as background subtraction, optical flow, Gaussian mixture model (GMM), frame difference method, and so on.
- **2.2.2** Background subtraction is the most popular choice to detect the stationary foreground object because it works well when the camera is static and the illumination changes gradually. This method detects the foreground object by measuring the difference between the current image and the background image. Even though the background subtraction technique works well to extract the moving object, it may suffer to dynamic background changes such as the entrance of a new background object or sudden illumination change.

- **2.2.3** Optical flow based motion segmentation uses characteristics of flow to detect independently the moving object even in the presence of camera motion. However most optical flow methods are computationally complex and very sensitive to noise. Gaussian mixture model is widely used in moving object detection in image sequences. However, the existing moving object detection methods based on Gaussian mixture model are not so efficient especially when dealing with noise or illumination mutation. Frame difference method is a temporal differencing technique. It uses the pixel-wise difference between two or three frames in frame sequences to extract the moving object. It is computationally simple, fast and adaptive to dynamic environment. But the frame difference method is most sensitive to the threshold value when determining pixel-wise difference between consecutive frames and it may produce the holes in the foreground object. When a foreground object moves slowly or stops moving, the temporal differencing.

Technique fails in detecting a change between consecutive frames and loses the target object information. A survey of moving object detection method can refer to literature [10, 11]. In this paper, we propose a detection method of moving object that combining the background subtraction with the Gaussian mixture model method. At first, the median smooth filter and image equalization are adopted for the video frame in order to reduce the influence of noise and luminance change. Then the region filling is used to fill the holes of object detected by the background subtraction. Considering the real-time requirements, we utilize the following method to fill the holes.

The background is firstly filled by the foreground white color (point (0, 0) as the seed point), so the black holes can be obtained. Then this binary image is reversed (0 is changed to 1 and 1 is changed to 0). The image with white color holes is achieved and it is added with the original binary image in order to fill the holes of object. Fig is an example at frame 364. Fig. (2a) is a video frame and Fig is a foreground image without image filling. Fig. (2c) is a foreground image using our image filling method and the number of hole in this image is decreased appreciably. Finally, the detection of moving object is achieved by the logical AND operation of the binary foreground image of GMM and the binary foreground image of background subtraction.

3. FEATURES

1. **Interface:-** Android's default user interface is mainly based on direct manipulation, using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, along with a virtual keyboard. Game controllers and full-size physical keyboards are supported via Bluetooth or USB. The response to user input is designed to be immediate and provides a fluid touch interface, often using the vibration capabilities of the device to provide haptic feedback to the user. Internal hardware, such as accelerometers, gyroscopes and proximity sensors are used by some applications to respond to additional user actions, for example adjusting the screen from portrait to landscape depending on how the device is oriented, or allowing the user to steer a vehicle in a racing game by rotating the device, simulating control of a steering wheel. Android devices boot to the home screen, the primary navigation and information "hub" on Android devices, analogous to the desktop found on personal computers. Android home screens are typically made up of app icons and widgets; app icons launch the associated app, whereas widgets display live, auto-updating content, such as a weather forecast, the user's email inbox, or a news ticker directly on the home screen. A home screen may be made up of several pages, between which the user can swipe back and forth. Third-party apps available on Google Play and other app stores can extensively re-theme the home screen, and even mimic the look of other operating systems, such as Windows Phone. Most manufacturers customize the look and features of their Android devices to differentiate themselves from their competitors.
2. **Applications:-** Applications ("apps"), which extend the functionality of devices, are written using the Android software development kit (SDK) and, often, the Java programming language. Java may be combined with C/C++, together with

a choice of non-default runtimes that allow better C++ support. The Go programming language is also supported, although with a limited set of application programming interfaces (API). In May 2017, Google announced support for Android app development in the Kotlin programming language. The SDK includes a comprehensive set of development tools, including a debugger, software libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Initially, Google's supported integrated development environment(IDE) was Eclipse using the Android Development Tools (ADT) plugin; in December 2014, Google released Android Studio, based on Intelli J IDEA, as its primary IDE for Android application development. Other development tools are available, including a native development kit (NDK) for applications or extensions in C or C++, Google App Inventor, a visual environment for novice programmers, and various cross platform mobile web applications frameworks. In January 2014, Google unveiled an framework based on Apache Cordova for porting Chrome HTML 5 web applications to Android, wrapped in a native application shell.

3. **Memory Management:-** Since Android devices are usually battery-powered, Android is designed to manage processes to keep power consumption at a minimum. When an application is not in use the system suspends its operation so that, while available for immediate use rather than closed, it does not use battery power or CPU resources. Android manages the applications stored in memory automatically: when memory is low, the system will begin invisibly and automatically closing inactive processes, starting with those that have been inactive for the longest amount of time. Life hacker reported in 2011 that third-party task killer applications were doing more harm than good.
4. **Hardware:-** The main hardware platform for Android is ARM (the ARMv7 and ARMv8-A architectures), with x86 and x86-64 architectures also officially supported in later versions of Android. The unofficial Android-x86 project provided support for x86 architectures ahead of the official support. The ARMv5TE and MIPS32/64 architectures were also historically supported but removed in later Android releases. Since 2012, Android devices with Intel processors began to appear, including phones and tablets. While gaining support for 64-bit platforms, Android was first made to run on 64-bit x86 and then on ARM64. Since Android 5.0 "Lollipop", 64-bit variants of all platforms are supported in addition to the 32-bit variants. Requirements for the minimum amount of RAM for devices running Android 7.1 range from in practice 2 GB for best hardware, down to 1 GB for the most common screen, to absolute minimum 512 MB for the lowest spec 32-bit smartphone. The recommendation for Android 4.4 is to have at least 512 MB of RAM, while for "low RAM" devices 340 MB is the required minimum amount that does not include memory dedicated to various hardware components such as the baseband processor. Android 4.4 requires a 32-bit ARMv7, MIPS or x86 architecture processor (latter two through unofficial ports), together with an OpenGL ES 2.0 compatible graphics processing unit (GPU). Android supports OpenGL ES 1.1, 2.0, 3.0, 3.1 and as of latest major version, 3.2 and since Android 7.0 Vulkan (and version 1.1 available for some devices). Some applications may explicitly require a certain version of the OpenGL ES, and suitable GPU hardware is required to run such applications.
5. **Development:-** Android is developed by Google until the latest changes and updates are ready to be released, at which point the source code is made available to the Android Open Source Project (AOSP), an open source initiative led by Google. The AOSP code can be found without modification on select devices, mainly the Nexus and Pixel series of devices. The source code is, in turn, customized and adapted by original equipment manufacturers (OEMs) to run on their hardware. Also, Android's source code does not contain the often proprietary device drivers that are needed for certain hardware components. As a result, most Android devices, including Google's own, ultimately ship with a combination of free and open source and proprietary software, with the software required for accessing Google services falling into the latter category.

4. CONCLUSION

We are able to detect and recognize faces of the criminals in an image and in a video stream obtained from a camera in real time. We have used Haar feature-based cascade classifiers in Open CV approach for face detection.Detecting human beings accurately in a surveillance video is one of the major topics of vision research due to its wide range of applications. It is challenging to process the image obtained from a surveillance video as it has low resolution. A review of the available detection techniques is presented. The detection process occurs in two steps: object detection and object classification.

In this paper, all available object detection techniques are categorized into background subtraction, optical flow and spatio-temporal filter methods. The object classification techniques are categorized into shape-based, motion-based and texture-based methods. The characteristics of the benchmark datasets are presented, and major applications of human detection in surveillance video are reviewed. At the end of this paper, a discussion is made to point the future work needed to improve the human detection process in surveillance videos. These include exploiting a multi-view approach and adopting an improved model based on localized parts of the image.

REFERENCES

1. Y. Pfitch, A. Rav-Acha, S. Peleg, “Non chronological video synopsis and indexing,” IEEE Trans. PAMI, vol. 30, no.11, pp.1971- 1984, 2008.
2. Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg, “Webcam Synopsis: Peeking Around the World,” in 11th International Conference on Computer Vision, pp. 1-8, 2007.
3. A. Rav-Acha, Y. Pritch, and S. Peleg, “Making a Long Video Short: Dynamic Video Synopsis,” in IEEE Conf. Computer Vision and Pattern Recognition, pp. 435-441,
4. Teng Li, “VideoM: Multi-Video Synopsis,” in IEEE International Conference on Data Mining Workshops, pp. 854-861, 2008.
5. Y. Pritch, “Clustered Synopsis of Surveillance Video,” in 6th IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 195-200, 2009.
6. U. Vural, Y.S. Akgul, “Eye-gaze based real-time surveillance video synopsis,” Pattern RecognitLett, vol. 30, no. 12, pp. 1151-1159, 2009.
7. U. Damnjanovic, “Event Detection and Clustering for Surveillance Video Summarization,” in IEEE Ninth International Workshop on Image Analysis for Multimedia Interactive Services, pp. 63-66, 2008.
8. Xu min, “Research of video synopsis technology based on object motion information,” M. S. thesis, University of Science and Technology of China, Hefei, China, 2009.
9. J. Nascimento and J. Marques, “Performance evaluation of ob-ject detection algorithms for video surveillance,” Multimedia, IEEE Trans, vol. 8, no. 4, pp. 761-774, 2006.
10. W. Hu, T. Tan, L. Wang, and Steve Maybank, “A survey on visual surveillance of object motion and behaviors,” IEEE Trans. Syst, Man, Cybernet – Part C: Appl. Rev., vol. 34, no. 3, pp. 334-352, 2004.
11. M. Antony, J. Anitha, “A survey of moving object segmentation methods,” Int. J. Adv. Res. Electron. Commun. Eng. (IJARECE), vol. 1, no. 4, pp. 73-80, 2012.
12. L. Li, W. Huang, I.Y.H. Gu, and Q. Tian, “Foreground object detection from videos containing complex background,” in Proc. ACM Int. Conf. Multimedia, Berkeley, CA, pp. 2-10, 2003.
13. C. Rother, V. Kolmogorov, and A. Blake, “Grabcut-interactive foreground extraction using iterated graph cuts,” ACM Trans. Graph, vol. 23, no. 3, pp. 309-314, 2004.
14. Belhumeur, P. N., Hespanha, J. P., &Kriegman, D. J. (1997). Eigenfaces vs. Fisher faces: Recognition Using Class Specific Linear Projection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 19, pp. 711-720. IEEE Computer Society.
15. Viola, P. and Jones, M. Rapid object detection using boosted cascade of simple features. IEEE Conference on Computer Vision and Pattern Recognition, 2001.
16. Brunelli, R., &Poggio, T. (1993). Face Recognition: Features versus templates. IEEE Transaction on Pattern Analysis and Machine Intelligence, 15 (10), 1042-1052.
17. Bureau of Justice Statistics, U.S. Department of Justice, April 1990), pp. 43-66; SEARCH Group, “Legal and Policy Issues Relating to Biometric Identification Technologies”.