SARIVA: Smartphone App for Real-time Intelligent Video Analytics

Submitted: 13th October 2014; accepted: 31th October 2014

Christopher Clarke, Plamen Angelov, Majid Yusuf, Pouria Sadeghi-Tehran

DOI: 10.14313/JAMRIS_4-2014/32

Abstract:
This paper presents the design, implementation and evaluation of a new smartphone application that is capable of real-time object detection using both stationary and moving cameras for embedded systems, particularly, the Android smartphone platform. A new object detection approach, Optical ORB, is presented which is capable of real-time performance at high definition resolutions on a smartphone. In addition, the developed smartphone application has the ability to connect to a remote server and wirelessly send image frames when moving objects appear in the camera's field of view; thus, allowing the human operator to only view video frames that are of interest. Evaluation experiments show a capability of achieving real-time performance for high definition (HD) resolution video.

Keywords: *autonomous objects detection, smartphone, mobile application, video analytics*

1. Introduction

Visual surveillance is abundant in everyday life due to the low cost and availability of video recording systems such as web cameras, digital cameras, closed circuit television (CCTV), and more recently smartphone cameras. With recent advances in camera technology, and, hence, increased resolution vast amounts of data are generated that can be processed and analysed in order to gain high-level information. This is commonly performed manually by analysts. Recent advances in processor technology and reduced footprint allows for computationally powerful systems to be embodied in smartphones [12–14]. The ability to process video frames on a smartphone reduces the need for constant human involvement. In addition, processing the video stream prior to being transmitted to a remote server for analysis has the potential to reduce the bandwidth of the data that must be sent.

An advantage of using optical cameras for object detection is that it doesn't rely on close proximity to moving objects, as opposed to proximity triggered cameras that are also commonly used for detecting objects. This enables the system to have a much deeper field of view for detection and the same techniques can be used with night vision, ultraviolet or infrared thermal imaging cameras, in addition to the visible colour spectrum, for additional applications such as search and rescue in the dark.

Since most of the embedded devices are powered by batteries, they do not achieve high performance;

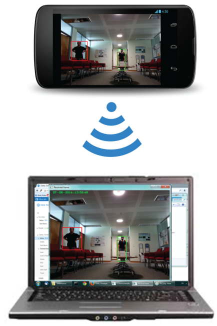

Fig. 1. The outputs from the smartphone application and desktop server when the Wi-Fi capability is enabled and moving objects are detected. When there is no moving object there is a blank screen and a message,'no motion detected'.

thus, they are unable to execute complicated computation such as extensive computer vision analysis. Nevertheless, the revolution of smartphones has removed the limitation of embedded devices. The processors used in smartphones are becoming more powerful with multiple core architecture which makes them faster with even smaller power consumption.

Previously, applications on a mobile phone were limited to the ones developed by mobile phone companies; however, since the launch of the Android operating system (OS) in 2007, mobile development has been in high demand. The aim of this paper, is to report the use of an Android based smartphone as a platform for video analysis and object recognition applications which work on images captured by the smartphone's built-in camera. The developed smartphone application is capable of performing objects detection in real-time using either a stationary or moving camera without the need for prior knowledge of the size or shape of the objects. In addition, a desktop program was developed that interfaces with the smartphone application that is capable of displaying and saving images sent from the smartphone application when a moving object has been detected [8] (Figure 1).

2. Description of the Methodology (SARIVA)

There are various approaches for detecting moving objects such as Background Subtraction (BS) [4, 10], and optical flow [17, 18]. Object detection can be performed on a single frame using image segmentation or feature matching. In this section we describe the approaches developed for both stationary and moving camera on Android smartphones.

2.1. Objects Detection App for Stationary Camera on a Smartphone

One of the most common techniques for objects detection for stationary camera is BS. The advantage of background subtraction (BS)techniques is that there are no constraints with respect to object size, shape or velocity. Any changes that deviate significantly and suddenly from the background model will be detected. However, a number of problems must be overcome when building the background model. Illumination changes must be accounted for due to the time of day or dynamic lighting conditions. Statistical fluctuations in the background model can be caused by the weather or due to noise as a result of low-quality recording equipment. It is also important for the model to account for dynamic shadows that are cast by objects and to remove detection of objects that are of no interest to the observer, such as non-stationary background objects. These limitations must be accounted for, but it is important that the alleviation of these limits does not affect the number of objects of interest that are detected. For example, removing non-stationary background objects should not affect the detection of objects that are of interest to the user.

We used two novel approaches known as Recursive Density Estimation (RDE) [2, 3] and Recursive Total-Sum-Distance-based Density Estimation (RTSDE) [1]. The advantages of using such techniques has been investigated in our previous works [4] and it has been proven that they outperform other alternatives such as Kernel Density Estimation (KDE) and Gaussian Mixture Model (GMM) [15]. RDE recursively builds the background model and updates the background model based on ***all*** previous image frames or as many or as little as necessary. On the other hand, RTSDE is approximately twice as fast as RDE due it requiring only integer calculations, which significantly reduces the computational complexity especially in cases where a floating-point unit (FPU) is unavailable [1].

2.2. Objects Detection App for a Moving Camera on a Smartphone

To date the majority of research concerning object detection with a moving camera suffers from one of the following issues: Prior information about objects to be tracked must be available; High computational complexity of the approaches means they are not capable of real-time performance. The approaches impose limitations in the camera movement, existing approaches have reduced detection rate.

In this paper, we introduce a new approach for real-time autonomous object detection application for moving camera on a smartphone known as Optical ORB. The Optical ORB approach has the ability to detect multiple objects without the need for prior knowledge regarding the objects to be detected such as their type, shape or size. The main advantage of Optical ORB is the removal of the need for image stitching, which is the most computationally complex stage of background subtraction based object detection approaches for use with moving cameras [8].

The first stage of the proposed approach is to extract the image features from the previous frame using the ORB feature detector. The ORB feature detector uses a pyramidal approach to the FAST feature detection algorithm in order to detect stable keypoints [16] which utilises a pyramid decimation ratio which can be used to control the compromise between the speed and accuracy of the FAST feature detector, i.e. the algorithm can be sped up at the expense of the quality of features detected and vice versa. FAST is one of the most computationally efficient feature detectors and another effect of speeding up the algorithm at the expense of the quality of features is the increase in the number of features detected. The higher the quality of features the less there will be because the features are scored using their Harris response [11]. As a result of the pyramidal approach the ORB detector can be tuned to detect a lot of features in a short amount of time compared with the standard FAST feature detector without the quality of the features degrading to a point of being unusable.

Once the features have been detected, the pyramidal Lucas-Kanade optical flow algorithm [6] is used to calculate each feature's new position in the current frame and, therefore, the optical flow displacement vector which describes the movement of the feature.

A recently introduced real-time evolving clustering algorithm is then applied to cluster the optical flow vectors based on their angle of motion and displacement. Mean shift [7] and Evolving Local Mean (ELM) [5] clustering were candidates because they calculate the local means of the clusters, which removes the influence of the number of moving objects and the velocity of the moving objects. For example, if one were to just use the global mean and standard deviation of the optical flow vectors then the output would be influenced by the number of moving objects and magnitudes of their velocities. Since the Optical ORB approach does not use homography calculation, the clustering must be as accurate as possible to compensate for the camera motion given the real-time constraints.

Once the optical flow vectors have been clustered the features relating to the background movement were removed. This was done by removing the cluster with the biggest number of optical flow vectors associated with it, N, which represents the relative background to camera motion. In addition, any other clusters that have $T \times N$ optical flow vectors are also removed, where $0 < T < 1$, because there may be more than one cluster which also represents the background motion of the camera. The camera motion may be represented by different clusters because features closer to the camera having a different optical flow to those further away due to perspective distortion. The value of T can be tuned based on the scene and relative distances of the background objects. For example, scenes such as motorways involve multiple moving objects moving in the same direction with approximately the same velocity and these will form one cluster; therefore, care must be taken to ensure these are not removed by the threshold, T. The remaining features that do not correspond to the motion of the camera and represent the moving objects in the scene. These can then be clustered using ELM using their spatial locations in the scene to identify individual objects. Any clusters with less than $T_{\min}$ features are discarded in the event errors occur as a result of the optical flow algorithm and this provides a minimum number of features that are expected from an object.

3. Smartphone Application Implementation

The following section describes the implementation of the object detection and clustering algorithms, the smartphone application and the desktop server. The developed object detection algorithms are multi-threaded which is more efficient for multi core processors and can significantly reduce the processing time required. The smartphone application was developed using the Eclipse Integrated Development Environment (IDE) with the Android Development Tools (ADT) plugin using the Android Software Development Kit (SDK) and Native Development Kit (NDK) for enabling the use of C++ code within the application. The object detection and clustering algorithms were implemented in Linux before being ported to the Linux-based Android platform. This includes the multi threading capability of the algorithms due to the fact Android supports the Linux based pthread implementation for multithreading. The smartphone application requires the OpenCV library manager application to be installed on the target smartphone. This reduces the storage requirements of the smartphone application because the OpenCV library manager contains the OpenCV library functions that are required by the smartphone application to run. This means that the libraries can be shared between multiple computer vision applications on the phone. The resultant APK for the smartphone application requires 5.49MB of storage space and the OpenCV manager requires 22.11MB.

3.1. Evaluation

Evaluation of the moving camera object detection approaches was performed on a Samsung Galaxy S3 smartphone powered by a quad core 1.4GHz ARM Cortex-A9 processor and 1GB of RAM with Android 4.4.4 Kitkat. RDE, RTSDE, and Optical ORB algorithms were tested on the smartphone with different resolutions (Table 1). The performance of the algorithms described in section 2 were evaluated in different scenarios in real-time (Figure 2). Results are shown in Figure 3.

The processing time is also measured from the start of one frame being processed to the start of the next frame being processed and hence includes the time required to grab the frame from the camera, convert it into a usable format (e.g. RGB), process the frame and display it to the smartphone's screen. The results show real-time performance at resolutions of 320×240 and 640×480. There appears to be a limit on the amount of frames per second that can be processed by the smartphone as can be seen by the timings for the 320×240 resolution which is a result of the time taken to grab the image frames and the underlying Android OS. Interestingly, the 640×480 resolution results in a slightly lower time to process despite the higher resolution which could be due to the processing required to reduce the frame size which requires interpolation of the pixel values. The difference in the speeds of the algorithms can be seen for the 1280×720 resolution in which RTSDE significantly outperforms all other algorithms and results in a stable real-time performance.

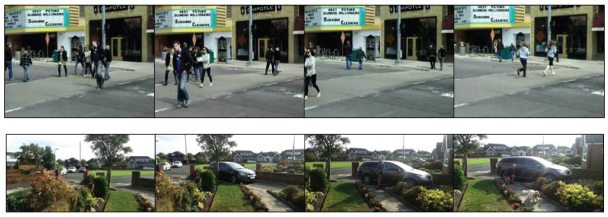

Fig. 2. Frames used for the quantitative evaluation of moving camera object detection algorithms using smartphone camera video sequence.

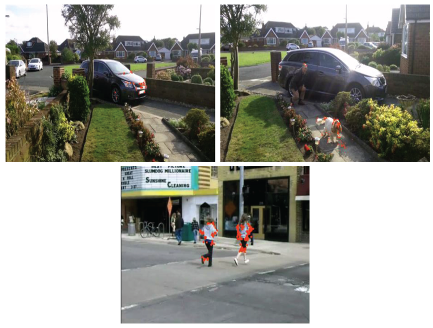

Fig. 3. Output of the Optical ORB approach for the smartphone camera video analytics.

Tab. 1. Timings of the implemented object detection algorithms on the smartphone.

Object Detection Approach	Resolution 320x240 Time per frame (ms)	320x240 FPS	640x480 Time per frame (ms)	640x480 FPS	1280x720 Time per frame (ms)	1280x720 FPS
RDE	60.1	16.6	59.8	16.7	140.3	7.1
RTSDE	60.0	16.7	59.4	16.8	75.2	13.3
Optical ORB	60.1	16.6	114.5	8.7	210.8	4.7

In addition, the quantitative analysis of the performance of the approaches is evaluated to measure the Detection Rate (DR) and False Alarm Rate (FAR) [9]. These metrics are based on True Positive (TP): moving objects that have been correctly detected as moving objects; False positives (FP): stationary objects that have been incorrectly detected as moving objects; False negatives (FN): moving objects that have been incorrectly detected as stationary objects.

The DR and FAR are defined as:

$$DR = \frac{TP}{TP + FN} \quad (1)$$

$$FAR = \frac{FP}{TP + FP} \quad (2)$$

Tab. 2. Results of the quantitative analysis of the Optical ORB smartphone camera video analytics approach.

Approach	Frame Number	TP	FP	FN	DR	FAR
Optical ORB	1	0	0	1	0.00	0.00
	2	1	0	0	1.00	0.00
	3	1	0	0	1.00	0.00
	4	2	0	0	1.00	0.00
	Total	**4**	**0**	**1**	**0.80**	**0.00**

Optical ORB approach has a tendency to produce false negatives and does not detect objects for every frame unlike the background subtraction based approaches. False positives are a rare occurrence for the Optical ORB approach and these tend to manifest as single point object detections as a result of errors introduced in the optical flow calculations which can easily be removed in the novelty clustering process.

4. Summary & Conclusion

An innovative smartphone application was successfully developed that is capable of performing real-time object detection for both stationary and moving cameras without prior knowledge of the objects to be detected. A desktop server was successfully implemented which is capable of wirelessly receiving the image frames from the smartphone application when a moving object has been detected. All of the implemented approaches to object detection were capable of achieving real-time performance, with RTSDE achieving over 13 frames per second for high-definition (HD) resolution (1280×720) and a newly proposed object detection algorithm, Optical ORB, achieving real-time performance of 8.7 fps for a resolution of 640×480 and near real-time performance (4.7 fps) for HD resolution. For more information, please visit: `http://www.lancaster.ac.uk/staff/angelov/Projects/SARIVA.htm`

5. Future Work

In this study, we developed approaches for real-time object detection using stationary and moving camera that require no prior knowledge of the object type or size. In the future these approaches could be incorporated into a larger system that tracks the detected objects and classifies the type of object and their behaviour which would allow for more high-level information to be autonomously extracted. Due to the low computational complexity of the proposed approaches they can be implemented on low-powered, portable hardware and multiple smartphones could be used to co-operatively detect objects and provide more information, such as the location of an object in real world 3D co-ordinates.

AUTHORS

Christopher Clarke – Intelligent Systems Lab, Data Science Group, Lancaster University, UK, LA1 4WA.
Plamen Angelov* – Intelligent Systems Lab, Data Science Group, Lancaster University, UK, LA1 4WA, e-mail: p.angelov@lancaster.ac.uk.
Majid Yusuf – Intelligent Systems Lab, Data Science Group, Lancaster University, UK, LA1 4WA.
Pouria Sadeghi-Tehran – Intelligent Systems Lab, Data Science Group, Lancaster University, UK, LA1 4WA.

*Corresponding author

ACKNOWLEDGEMENTS

This work is supported the SARIVA research project funded by Security Lancaster, Lancaster University, UK, 2013-14.

REFERENCES

[1] P. Angelov and A. Wilding, "RTSDE: Recursive Total-Sum-Distances-based Density Estimation Approach and its Application forAutonomous Real-Time Video Analytics", *Symposium Series on Computational Intelligence (SSCI '14; to appear)*.

[2] P. Angelov. "Anomalous system state identification". GB1208542.9, May 2012.

[3] P. Angelov, *Autonomous Learning Systems: From Data Streams to Knowledge in Real Time*, John Wiley and Sons, 2012.

[4] P. Angelov, P. Sadeghi-Tehran, and R. Ramezani, "A Real-time Approach to Autonomous Novelty Detection and Object Tracking in Video Stream", *International Journal of Intelligent Systems*, vol. 26, 2011, pp. 189–205.

[5] R. D. Baruah and P. Angelov, "Evolving Local Means Method for Clustering of Streaming Data", *IEEE World Congress on Computational Intelligence*, 2012, pp. 2161–2168.

[6] J. Y. Bouguet, "Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm", *Intel Corporation, Microprocessor Research Labs*, 1999.

[7] Y. Cheng, "Mean shift, mode seeking, and clustering", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, 1995, pp. 790–799.

[8] C. Clarke. "Smartphone Application for Real-time Object Detection". Master's thesis, School of Computing and Communications, Lancaster University, 2014.

[9] I. Cohen and G. Medioni, "Detecting and tracking moving objects for video surveillance". In: *Computer Vision and Pattern Recognition, 1999.*

IEEE Computer Society Conference on, 1999, pp. 319–325.

[10] M. Cristani, M. Farenzena, D. Bloisi, and V. Murino, "Background Subtraction for Automated Multisensor Surveillance: A Comprehensive Review", *Journal on Advances in Signal Processing*, 2010, pp. 1–24.

[11] C. Harris and M. Stephens, "A combined corner and edge detector.", *Alvey vision conference*, 1988.

[12] K. Jeong and H. Moon, "Object Detection Using FAST Corner Detector Based on Smartphone Platforms". In: *Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on*, 2011, pp. 111–115.

[13] K. Kadar, F. de Sorbier, and H. Saito, "Displayed Object Recognition for Smartphone Interaction", *International Conference on Machine Vision Applications*, 2013, pp. 20–23.

[14] K. Matusiak, P. Skulimowski, and P. Strumillo, "Object recognition in a mobile phone application for visually impaired users". In: *Human System Interaction (HSI), 2013 The 6th International Conference on*, 2013, pp. 479–484.

[15] G. Morris and P. Angelov, "Real-time Novelty Detection in Video using Background Subtraction Techniques: State of the art". In: *IEEE International Conference on Systems, Man and Cybernetics (SMC '14) (to appear)*.

[16] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF". In: *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 2564–2571.

[17] P. Sadeghi-Tehran, C. Clarke, and P. Angelov, "A Real-time Approach for Autonomous Detection and Tracking of Moving Objects from UAV". In: *IEEE Symposium Series on Computational Intelligence SSCI'14 ; to appear)*.

[18] T. Senst, V. Eiselein, and T. Sikora, "Robust Local Optical Flow for Feature Tracking", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 9, 2012, pp. 1377–1387.