Real time video processing and object detection on android smartphones essay

Real Time Video Processing and Object Detection on Android Smartphone

Abstract aˆ“ As Smartphone is getting more potent, can do more superior stuffs that previous required a computer. For employing the high processing power of Smartphone is mobile computer vision, the ability for a device to capture; process; analyze; understanding of images. For mobile computer vision, Smartphone must be faster and real time. In this study two applications have been developed on Android platform using OpenCV and core library called as CamTest with own implemented algorithms. Efficiency of two Android applications have been compared and found that OpenCV performs faster than CamTest. The results of examining the best object detection algorithm with reverence to efficiency shows that FAST algorithm has the finest blend of speed and object detection performance. Next projected object recognition system using FAST algorithm, which uses SVM, BPNN for training and validation of object in real time. The application detects the object perfectly with recognition time around 2 ms using SVM and 1 ms using BPNN.

Keywordsaˆ” Android; Video Processing; object detection; SVM; FAST corner detector; BPNN

I. INTRODUCTION

As Smartphone is the perfect combination of personal digital assistant, media player, camera and several other stuffs. It has entirely changed the past about mobile phone. In the early days of Smartphone application development only mobile company was able to develop. After the introduction of Android OS in 2007, Smartphone application development is high in demand. Android was developed by Google with Linux core kernel and GNU software stuffs. [16].

The introduction of Smartphone with camera Real Time video processing becomes very trendy now and having most critical computation tasks. Nearly all Smartphone applications uses a camera to use mobile computer vision technology [2]. Mobile computer vision technologies playing vital role in developing our day to day activities applications [1]. This technology having many objectives like object finding, segmenting, location recognition [2].

As Smartphone processors such as MediaTek, ARM, NVIDIA Tegra, and Snapdragon are achieving more computation capability showing a fast growth of mobile computer vision applications, like image editing, augmented reality, object recognition. Long processing time due to the high computational difficulty averts mobile computer vision algorithms from being practically used in mobile phone applications. To overcome this problem, researchers and developers have explored the libraries such as OpenGL and OpenCV [2]. Application developers will face lot problems as he does not having basic idea to process real time video. OpenCV library is the solution which is written in C, C++ language, reduces the complexity for development and research [17] [2].

Real-Time recognition and detection of objects is complex and favorite area for research in the todayaˆ™s fast growing mobile computer vision technology. Applications like machine vision, visual surveillance; robot navigation are the best examples [4]. Object detection and recognition consist of three steps basically, first is the feature extraction, second classification and third is the recognition of object using machine learning and several other technologies [3].

Due to the growth of Scale Invariant Feature Transform (SIFT)[10], the object detection method using matching filter changed to key point matching based object detection method [8] [10]. SIFT is more focusing on invariant key point matching. On the similar concept new algorithms were born such as the Speeded-Up Feature Transform (SuRF)[11], Center Surrounded Extrema (CenSurE)[22], Good Features to Track (GFTT)[26], Maximally-Stable Extremal Region Extractor (MSER)[24], and Oriented Binary Robust Independent Elementary Features (ORB)[21], and Features from Accelerated Segment Test (FAST)[12] [4] [6] [8].

In this paper, real time video processing efficiency was find using OpenCV [17] and CamTest with support of core library. Next analyze best object detection algorithm with respect to efficiency in support with OpenCV library. Projected real time object recognition system using FAST algorithm [12], SVM [15] and BPNN [25]. All the stuffs have been conducted on LG Optimus Vu Smartphone with Android 4. 0. 4 OS.

II. ANDROID ARCHITECTURE

The Android operating system is like other Smartphone OS, with stacked structure [2][16]. Android operating system stack consist on several layers such Kernel Layer, System Libraries, Dalvik Virtual Machine layer (i. e Android Runtime layer), Application Framework layer and on top Applications layer [2][16]. The Kernel gives basic funtionalities like network management memory management, process management, device management. Libraries are used for different oprations like internet security [2][16]. Android Runtime consist of Dalvik Virtual Machine which is optimized for Android and provides core libraries. The Application Framework layer gives services to the installed applications in the form of Java Class Library. [2][16]. Application developers takes the services of this layer for application development [2][16]. Application layer is the top layer in the stack where your application will get install [2][16].

III. OPENCV IN ANDROID

The OpenCV library was officially developed and introduced by Intel in 1999 to enforce CPU and GPU exhaustive application [17]. The earlier version of OpenCV was written in C[27]. From the edition 2. 0 OpenCV provided both C and C++ interfaces[27]. In the next edition of 2. 2 they had introduces Android port with some sample applications of image processing. Currently it has several optimized methods with the version OpenCV 2. 4. 9[27] [17].

IV. real time video processing methods

To find and compare the efficiency of OpenCV and CamTest, each processing method of mobile computer vision was applied and average value was calculated [2]. The input format of video frame should be in standard form such as RGB space[2][27].

The input video frame to RGB conversion is done by following relation [28]

R = 1. 164(Y – 16) + 1. 596(V – 128)

G = 1. 164(Y – 16) aˆ“ 0. 813(V – 128) aˆ“ 0. 391(U – 128)

$B = 1.164(Y - 16) + 2.018(U - 128)$ (1)

Each pixel of video frame is threshold with a constant number T. If it is greater than T, pixel will be set 1, otherwise 0.

$g(x, y) = 1, \text{ if } f(x, y) > T$

$= 0, \text{ otherwise}$ (2)

Where *f(x, y)* is the original frame and *g(x, y)* is the threshold frame. The descriptions of processing methods are shown in Table I.

TABLE I. FRAME PRCESSING METHODS AND ITSDESCRIPTION

Sr. No

Frame Processing Method

Description of method

1

RGB

Input video frame to RGB color space

2

Grayscale

Luma color component to 0 ~ 255 grayscale

3

Threshold

Frame thresholding with value 70

4

Gaussian

2D convolution with Gaussian 3 X 3 kernel

5

Laplacian

2D convolution with Laplacian 3 X 3 kernel

6

Sobel

Frame filtering using Sobel operator

7

Mean

Frame filtering using average values of pixels

8

Median

Filtering using median values of pixels

V. METHODOLOGY

First designed application layout using JAVA and XML. Then, the processing methods and object detection algorithms are written using JAVA and OpenCV. The tools used for designing and programming are Android SDK [16], OpenCV [17] and JAVA SDK. Application file is then installed to the LG Optimus Vu. If there are no errors, then started to measure the result regarding frame processing rate. After all the data had been collected, and the result is analyzed and compared with the theory. The Application flow is shown in Fig. 1. 0 and Fig. 1. 1

A) System Flow of Real Time Video Processing and Object Detection Algorithms

No

Yes

Next

Real

Time

Video

Frame

z

Fig. 1. 0: Real time video processing flow

B) System Flow of Real Time Object Detection Algorithms

No

Yes

Next Real

Time Video

Frame

Fig. 1. 1: Real time object detection algorithms flow.

VI. EXPERIMENT RESULTS

A) Performance of Real Time Video Processing Methods

For the calculation of processing efficiency of OpenCV and CamTest is calculated by following formula.

(7)

The unit of FPR is frames processed per second i. e. fps. If the value of Frame Processing Rate(FPR) is high for the particular processing metohd then theat method is more efficient. Higher the value of FPR represents the method is more efficient. Table II. Shows real time video processing methods and frames processed per second by CamTest, OpenCV test.

TABLE II. REAL TIME VIDEO PROCESSING METHODS AND FPS OF CAMTEST AND OPENCV TEST

Real Time Video Processing Method

CamTest FPS

OpenCV FPS

RGB

2. 8

15

Grayscale

21

14

Threshold

2. 50

14. 50

Mean

1. 60

14

Gaussian

1. 80

12

Laplacian

2. 05

8

Sobel

1. 40

7. 20

Median

2. 03

7

Frame Processing Ratio is as follows,

FPR Ratio = (OpenCV FPR aˆ“ CamTest FPR)/OpenCV FPR (8)

As from Table II, FPR shows significant differences between OpenCV and CamTest. If there is Positive FPR ratio value e. g N, then OpenCV is 1/N times better than CamTest. If there is Negative FPR ratio value e. g aˆ“ M, then CamTest is 1/M times better than OpenCV. As shown in Table III, Frame Processing Rate Ratio(average) is 0. 64, leads to a conclusion that OpenCV (1/0. 64 times) 1. 56 times faster and better than CamTest.

TABLE III. REAL TIME VIDEO PROCESSING METHODS AND FPR RATIO

Real Time Video Processing Method

Ratio of Frame Processing Rate

Graysclale Processing Method

-0. 50

Threshold Processing Method

0. 82

RGB Processing Method

0. 81

Laplacian Processing Method

0. 75

Sobel Processing Method

0. 80

Median Processing Method

0. 71

Gaussian Processig Method

0. 85

Mean Processing Method

0. 88

Average of Procesing Method

0. 64

Fig. 2. 0: Frame processing rate using CamTest and OpenCV test for eight image processing methods.

B) Performance of Real Time Object Detection Algorithms

TABLE IV. REAL TIME OBJECT DETECTION ALGORITHMSAND THEIR FPS

Object Detection Algorithm

Object(Socket) 1 FPS

Object(Glue Stick) 2 FPS

FAST

11

14

SURF

2

5

SIFT

2. 78

3. 60

MSER

5

7

ORB

4. 50

6

STAR

2

4. 40

GFTT

1. 90

3. 30

Fig. 2. 1: Frame Processing Rate for object detection algorithm.

As shown in Table IV and Fig. 2. 1, FAST algorithm is having the highest fps value and 10 times faster as compare to SIFT and SURF. The minimus fps for real time object recognition should be at least 15 fps and FAST achieves the almost same thing. So that FAST is having optimum performance in real time scenario while executing real time object detection operation.

VII. APPLICATION

As from experimental results shown above in Table IV, we concluded that FAST algorithm [12] is almost several times faster than other algorithms. To recognize the object in real time video FAST algorithm almost achieves 15 fps. As FAST algorithm extracts the corner features accurately and it requires less time for it. So proposed a Real Time Object Recognition system using FAST algorithm is as follows.

A) System Flow of Real Time Object Recognition

As shown in Fig. 3. 0 Input object image is captured by Smartphone camera and it is saved to internal storage. FAST corner detector [12] algorithm is applied on the captured image to extract the features. The extracted features should have the same number and location as the viewpoint and corner changes. So the extracted features should be adjusted to the same number and it called as normalization. After the features are adjusted to the same number, weight is calculated for SVM [15] and BPNN [25] for training the features. After that feature database will get created. After the preparation of database object will get recognized in real time video via SVM [15] and BPNN [25]. As system recognizes the object it shows the feature count and recognition time on the display of Smartphone.

No

Input Database

Yes

Fig. 3. 0: Real Time Object Recognition Flow

A) Results

The Real time object recognition system shown above in Fig. 3. 0 was developed for LG Optimus Vu and Android platform 4. 0. 4. The development environment consist of Microsoft Windows 7 with Intel Core i3, 2GB RAM, Android SDK, NDK and JAVA SDK. The object used for training was Hand Watch and training time was 102 ms using SVM and 1115 ms using BPNN. The Table V presents the recognition time for object (Hand Watch) using FAST corner detector, SVM and BPNN.

TABLE V. RECOGNITION TIME FOR HAND WATCH OBJECT

Algorithm

Recognition Time(Hand Watch)

Sopport Vector Machine

2ms

Back Propagation Neural Network

1ms

VIII. CONCLUSION

As per the above experimentation and results, Most of the real time video processing methods executed using OpenCV having high performance with respect to efficiency than the CamTest. OpenCV gives more attention towards the efficinecy than the CamTest. As per the result obtained from the real time object detction application, FAST algorithm achieves high efficiency, almost 15 fps compared to other algorithms. For the futurescope, like to enhance the FAST algorithm in terms of accuracy. The proposed real time object recognition system gives faster and accurate recognition of object(Hand Watch) on the Smartphone using SVM and BPNN. In future would like to introduce multi object recognition, location tracking on Smartphone platforms, also like to introduce the concept like GPU and parallel computing with OpenCL.

REFERENCES

[1] Nasser Kehtarnavaz and Mark Gamadia, aˆ? Real-Time Image and Video Processing: From Research to Realityaˆ?, Synthesis Lectures On Image, Video and Multimedia Processing Lecture 5, 2006.

[2] Khairul Muzzammil bin Saipullah and Ammar Anuar, aˆ? Real-Time Video Processing Using Native Programming on Android Platformaˆ?, 8th IEEE International Colloquium on Signal Processing and its Applications, 2012.

[3] Kanghun Jeong and Hyeonjoon Moon, aˆ? Object Detection using FAST Corner Detector based on Smartphone Platformsaˆ?, First ACIS/JNU International Conference on Computers, Networks, Systems, and Industrial Engineering, 2011.

[4] Paul Viola, Michael Jones, aˆ? Robust Real-time Object Detectionaˆ?, Second International Workshop on Statistical and Computational Theories of Vision, July2001.

[5] L. Zhang and D. Yan, aˆ? An Improved Morphological Gradient Edge Detection Algorithmaˆ?, IEEE International Symposium on Communications and Information Technology (ISCIT), Vol. 2, pp. 1280-1283, 2005.

[6] O. Folorunso, O. R. Vincent and B. M. Dansu, aˆ? Image edge detection, A knowledge management technique for visual scene analysisaˆ?, Information Management and Computer Security, Vol. 15, No. 1, pp. 23-32, 2004.

[7] D. G. Kamdar and C. H. Vithalani, aˆ? Simulation and Performance Evaluation of Edge Detection Techniques in Differential Time Lapse Videoaˆ?, IEEE International conference on Computational Intelligence and Computing Research (ICCIC), 2012

[8] David G. Lowe, aˆ? Object Recognition from Local Scale-Invariant Featuresaˆ?, the Proceedings of the Seventh IEEE International Conference on Computer Vision, pp. 1150-1157, 1999.

[9] Clemens Arth, Christian Leistner, aˆ? Robust Local Features and their Application in Self Calibration and Object Recognition on Embedded Systemsaˆ?, Computer Vision and Pattern Recognition, 2007. IEEE Conference, June 2007.

[10] D. G. Lowe, aˆ? Distinctive Image Features from Scale-Invariant Key pointsaˆ?, International journal of Computer Vision, 60(2), pp. 91-110, 2004.

[11] H. Bay, T. Tuytelaars, L. V. Gool, aˆ? SURF: Speeded Up Robust Featuresaˆ?, Proceedings of the European Conference on Computer Vision, 2006.

[12] E. Rosten and T. Drummond, aˆ? Machine learning for high-speed corner detectionaˆ?, European Conference on Computer Vision, Vol. 1, pp. 430-443, 2006.

[13] E. Rosten and T. Drummond, aˆ? Fusing points and lines for high performance trackingaˆ?, Tenth IEEE International Conference on Computer Vision (ICCV'05), Vol. 2, pp. 1508-1515, 2005.

[14] Y. Hirose, K. Yamashita and S. Hijiya, aˆ? Back-Propagation Algorithm Which Varies the Number of Hidden Unitsaˆ?, Neural Networks, Vol. 4, pp. 61-66, 1991.

[15] E. J. A. K. Suykens and J. Vandewalle, aˆ? Least Squares Support Vector Machine Classifiersaˆ?, Neural Processing Letters (ICCV'05), Vol. 9, pp. 293-300, 1999.

[16]K. Owen, an Executive Summary of Research in Android &Integrated Development Environments, April 2011.

[17]OpenCV, Open source Computer Vision library. In 2009.

[18] Edward Roston and Tom Drummond, aˆ? Machine Learning for high-speed corner detectionaˆ?, Department of Engineering, Cambridge University, UK.

[19] Y. Khairul Muzzammil bin Saipullah and Ammar Anuar, aˆ? Coparision of Feature Extractors for Real-time Object Detection on Android Smartphoneaˆ?, Journal of Theoretical and Applied Information Technology, Vol. 47, 2013.

[20] Y. Khairul Muzzammil bin Saipullah and Ammar Anuar, aˆ? Analysis of Real-time Object Detection Methods for Android Smartphoneaˆ?, 3rd International Conference on Engineering and ICT (ICEI2012), 2012.

[21] Ethan Rublee Vincent, Gary Bradski, aˆ? ORB: an efficient alternative to SIFT or SURFaˆ?

[22] Agrawal M, K. Konolige, aˆ? CenSurE: Center Surround Extremas for Realtime Feature Detection and Matchingaˆ?, Computer Vision aˆ" ECCV, Springer Berlin/Heidelberg, vol. 5305, pp. 102-115, 2008

[23] Calonder M, V. Lepetit, aˆ? BRIEF: Binary Robust Independent Elemetary Featuresaˆ?, Computer Vision aˆ" ECCV, Springer Berlin/Heidelberg, vol. 6314, pp. 778-792, 2010

[24] Donoser M, Bischof H, aˆ? Efficeint Maximally Stable External Region(MSER) Trackingaˆ?, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 533-560, June 2006

[25] Y Hirose, K. Yamashita, aˆ? Back-Propagation Algorithm which varies the Number of Hidden Unitsaˆ?, Neural Networks, vol 4, pp. 61-66, 1991

[26] Shi J, Tomasi C, aˆ? Good Features To Track In Computer Vision and Pattern Recognitionaˆ?, CVPR 1994, pages 593-600, 1994

[27] Amar Anuar, Khairul Muzzamil Saipullah, aˆ? OpenCV Based Real-Time Video processing Using Native Android Smartphoneaˆ?, International Journal of Computer Technology and Electronics Engineering(IJCTEE), Vol 1, Issue 3

[28] Khairul Muzzamil Saipullah , Amar Anuar, aˆ? Measuring Power Consumption For Image Processing On Android Smartphone aˆ?, American Journal of Applied Sciences, pages 2052-2057, 2012