INTERNATIONAL CONFERENCE ON
SCIENCE AND TECHNOLOGY

CERTIFICATE

of Apreciation

Presented to

Puspaningtyas Sanjoyo Adi

as

Participant

in The International Conference on Science and Technology (ICOSAT 2016)

"Promoting A New Innovation on Food, Energy, and Environment Sustainability Through Science and Technology for Development"

Syahida Inn,
Syarif Hidayatullah State Islamic University Jakarta,
November 16th - 17th, 2016

Rector
Syarif Hidayatullah State Islamic University Jakarta

Prof. Dr. Dede Rosyada, MA
NIP. 19571005 198703 1 003

Chairman of Conference
Vice Dean Academic Affair

Dr. Elpawati, Ir. MP
NIP. 19641204 199203 2 001

Proceedings of the International Conference of Science and Technology

ICOSAT 2016

Syahida Inn,
Syarif Hidayatullah State Islamic University
Jakarta,
November 16th - 17th, 2016

Theme:
Promoting A New Innovation Sustainability Through Science and Technology for Development

Co - Host :

Ikatan Sarjana Ekonomi Indonesia

International Society for Southeast Asian Agricultural Sciences (ISSAAS)

PERTANIKA JOURNAL
Universiti Putra Malaysia

Publisher UIN Press

ISSN 2541-6758

International Conference of Science and Technology
(ICOSAT 2016)
Faculty Science and Technology
Syarif Hidayatullah State Islamic University Jakarta

TABLE OF CONTENTS

ENERGY AND ENVIRONMENT

ORB Image Detection in Android

Puspaningtyas Sanjoyo Adi
Universitas Sanata Dharma
Yogyakarta, Indonesia
puspa@usd.ac.id

Sri Hartati Wijono
Universitas Sanata Dharma
Yogyakarta, Indonesia
tatik@usd.ac.id

ABSTRACT

In recent years, there has been rapid development of the technologies used in smart devices. Hence, research into these technologies is required in order to expand our understanding of these devices. One of the main features of smart devices is the mobile camera. Many studies have been conducted about how to utilizing service using camera. A key component of camera applications is image processing software. In this paper, we analyze the performance of the ORB algorithm for image detection in mobiles. We use Android applications with OpenCV library. Through this study, we find that ORB algorithm can work well for image detection in mobile applications.

Index Terms—
Feature detector, Computer vision, Image processing, ORB, OpenCV

I. INTRODUCTION

Currently the image features can be summarized as two categories: global features (such as color, texture, outline, etc.) and local features. They are used to describe global information and local details of the images respectively. A lot of experimental results show that the integrated features which combine these two kinds of features can improve recognition accuracy effectively [2].

Scale Invariant Feature Transform (SIFT) is the most representative local feature descriptor currently, which is invariant to image scale and rotation, and proven to be able to provide robust matching across a substantial range of affine distortion, variations of 3D viewpoint and illumination [3]. But the computational complexity of feature extraction is very high. Therefore, a variety of fast local feature extraction algorithms have been proposed, such as Speeded Up Robust Features (SURF)[4], Oriented FAST [7] and Rotated BRIEF [5] (ORB)[6] etc. Compared with SIFT, the extraction speed of these features are much faster than that of SIFT and they can achieve comparable performance when they are used in image classification, retrieval and other applications.

II. Computer Vision Technologies

A. ORB feature detection and descriptor matching

Currently, common feature point matching algorithms include SIFT [3], SURF [4], BRIEF [5], ORB [6]. The key factor affecting speed and accuracy of feature point matching is feature detection. There are two kinds of feature detection: descriptor based on absolute value and descriptor based on comparison. SIFT and SURF which stand for descriptor based on absolute value generally quantify gray or gradient to get a histogram, then construct descriptor based on histogram. Compared with SIFT and SURF, BRIEF and ORB based on comparison have great advantage. They construct the descriptor by comparing the characteristic value which is pre-trained or belongs to random points. This kind of descriptor is designed to improve speed. As can be seen from [5] and [6], it takes respectively 5228.7 (ms), 217.3 (ms), 8.87 (ms), and 15.3 (ms) for SIFT, SURF, BRIEF-32 and ORB to process one frame. Compared with ORB, BRIEF is faster but it do not have the rotation invariance. When people use the smart phone, the change of rotation and position is relatively frequent. So BRIEF is not suitable for feature point matching for smart phone. ORB is proposed based on BRIEF. It has rotation invariance because of adding a direction variance to each descriptor. So this paper chooses ORB descriptor.

B. Open CV

OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it is free for both academic and commercial use [1]. It has C++, C, Python and Java interfaces, and supports Windows, Linux, Mac OS, iOS and Android operating systems. OpenCV is designed for computational efficiency, with a strong focus on real-time applications. Since it is written in optimized C/C++, the library can take advantage of multi-core processing. Also, it supports low-level and high-level APIs (Application Programming Interfaces), and is suitable for real-time applications. OpenCV has a modular structure, which means that the package includes several shared or static libraries. The following modules are available:

(1) core: A compact module defining basic data structures, including the dense multi-dimensional array "Mat" and basic functions used by all other modules.
(2) imgproc: An image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on.
(3) video: A video analysis module that includes motion estimation, background subtraction, and object tracking algorithms.
(4) calib3d: Basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence algorithms, and elements of 3D reconstruction.
(5) features2d: Salient feature detectors, descriptors, and descriptor matchers.
(6) objdetect: Detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on).
(7) highgui: An easy-to-use interface for video capturing, image and video codecs, as well as simple UI capabilities.
(8) gpu: GPU-accelerated algorithms from different OpenCV modules.

III. Simulation Design

In order to improve development efficiency, we choose to do with OpenCV. OpenCV is an open source computer vision library, and it implements many common algorithms of image processing and computer vision. OpenCV provides two kind of interface, C++ and Java, the

paper chooses Java interface in order to maintain consistency with the architecture of Android developing tools.

We simulate an ORB algorithm based on the Android operating system in real-time camera. The flowchart of object detection process is shown in figure 1. At first, reference image is saved in application and application will compute and get key point extractor and its descriptor. Application will reference image descriptor. We use ORB key point extractor and descriptor. After that, video picture is taken from smart phone camera. Application will get a sample video frame as scene image. ORB algorithm calculates key point and descriptor. Furthermore, the application will calculate the level of accuracy (matching) or descriptor matcher between two descriptors, reference descriptor and scene descriptor. The calculation result is a list of list of a couple of key points reference image and scene image along with the distance between keypoint. Furthermore, the application will find good key point with the steps: 1) calculate the minimum distance of all couples are formed. 2) find a couple keypoint within less than 1.5 times the minimum distance. Couple this keypoint hereinafter referred to as good key point. Furthermore, the application will calculate homografi with the input parameter is a good key point and the key point of the scene image. Homografi calculation will result in recognition status outputs reference image, known or not. If recognized, then the application will draw a box poses as a sign that the reference image is recognized on the scene image.

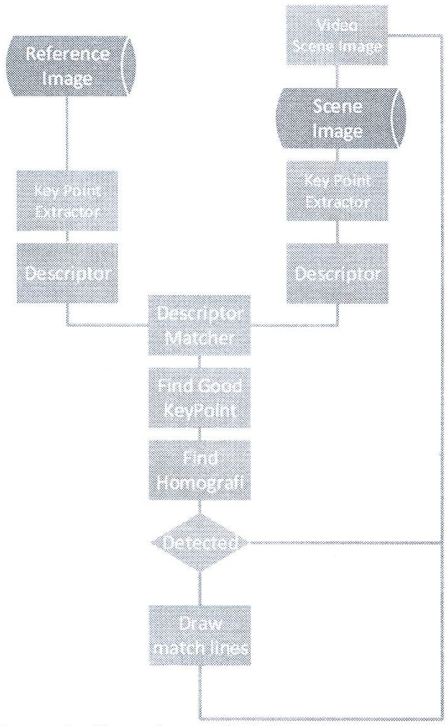


Figure 1. flowchart of object detection process

The Android application is designed based on the class diagram shown in Figure 2. `MainActivity` class is the main class in android application. `MainActivity` class implements the interface `CvCameraViewListener2`. This interface is the interface of the package OpenCV. This interface provides three functions: `onCameraFrame ()`, `onCameraViewStarted ()`, `onCameraViewStopped ()`. `OnCameraFrame()` is a

function that constantly works when the camera taking pictures. In `onCamera()`, the function `ImageDetectionFilter.apply()` is called. Function `apply ()` on class `ImageDetectionFilter` work according to flowchart image detection, corresponding figure 1.

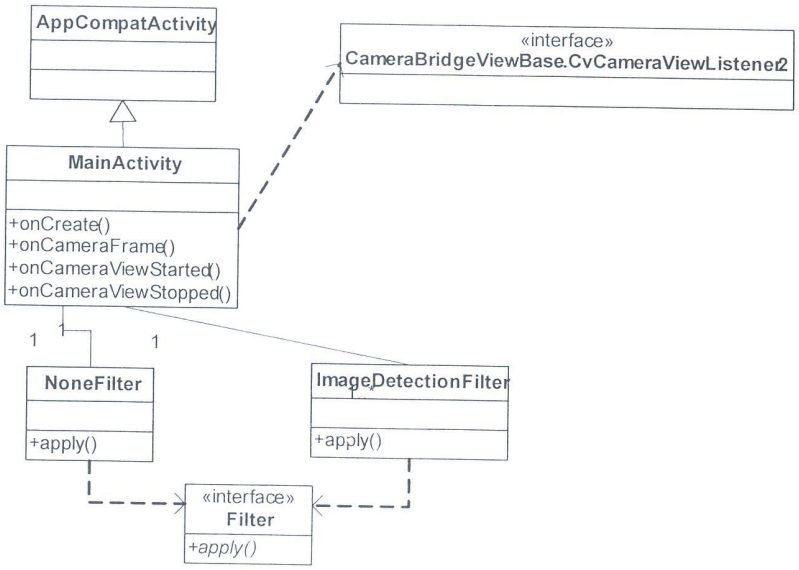
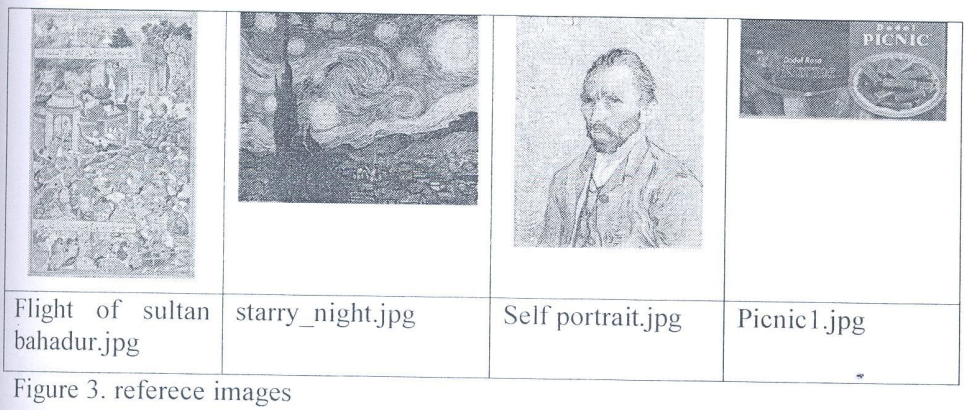


Figure 2. application class diagram

IV. Analysis of Result

Application tested by detecting four reference image, see figure 3. The application installed on the smartphone Samsung Galaxy Tab S2 series SM-T705. Smartphones fitted with a tripod to steady his position and exposed to the notebook display images or objects that have the same image with the reference image. Notebook will display a variety of images in which there are refence image.



Flight of sultan bahadur.jpg	starry_night.jpg	Self portrait.jpg	Picnic1.jpg

Figure 3. referece images

Image detection results can be seen in figure 4. Application can detect the reference image, picnic1.jpg, and making pose boxes and connecting lines between good key point in the scene image.

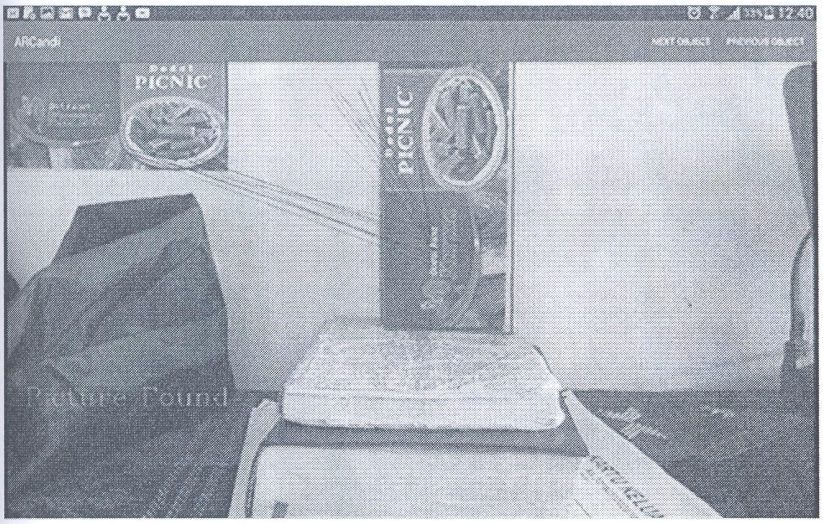


Figure 4. ORB detection result

Table 1. object detection time

file	average good keypoint	average detection time (mS)
flight_of_sultan_bahadur.jpg	10	773,07
starry_night.jpg	12	854,70
self_portrait.jpg	15	887,64
Picnic1.jpg	16	905,06

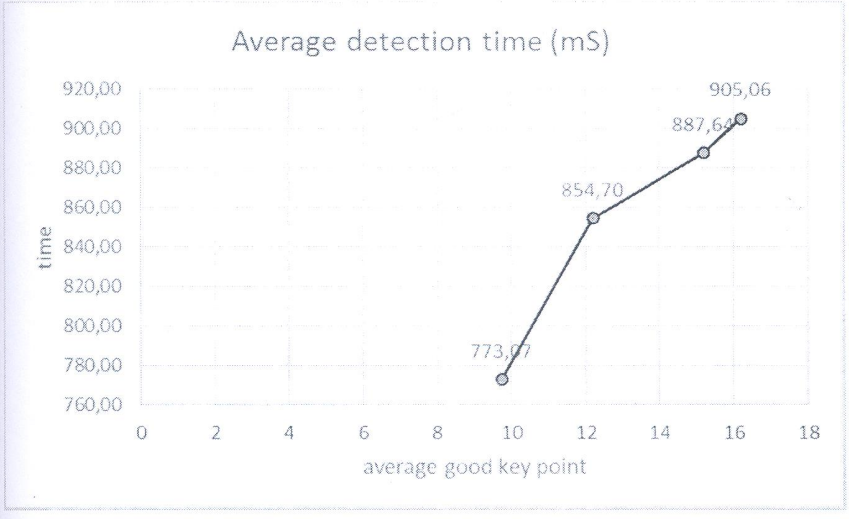


Figure 5. Object detection time graphic

Table 1 shows the mean time image detection. There are 4 image files are trying to identify. Each picture has a average good key point, which amounts vary between images. Figure picnic1.jpg who had a mean good key point of the most high has image detection time of 905.06 ms, while images flight_of_sultan_bahadur.jpg having a good key point average of 773.07 to have image detection time of 773.07 ms. Of figure 5, the average time image

detection increases in proportion with the addition of good key point. Good key point is the key point that has a distance of 1.5 times the minimum distance. This experiment has not been tried for various alternative minimum distance variables to find the optimal solution.

V. Conclusion

The result of this simulation Showed that the increasing of good key point will increase of detection time. Good key points is key points that have distance of 1.5 minimum distance. For further research, the minimum distance variables can be used as study materials to find the most optimum value.

Acknowledgment

The research work was supported by Penelitian Hibah Bersaing Kemenristekdikti Indonesia under Grant No. 010/HB-LIT/III/2016.

References

[1] OpenCV (Open Source Computer Vision), http://opencv.org.

[2] Yan Sun, Research on Some Algorithms of Content-based Image Retrieval and Sensitive Image Filtering (Ph.D. Dissertation), Jilin University, Jilin, 2011.

[3] David G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2), 2004, 91–110.

[4] B. Herbert, T. Tuytelaars, L.V. Gool, Surf: speeded up robust features, Computer Vision – ECCV, vol. 3951, 2006, pp. 404–417.

[5] M. Calonder, V. Lepetit, C. Strecha, P. Fua, Brief: binary robust independent elementary features, Computer Vision – ECCV 2010, vol. 6314, 2010, pp. 778–792.

[6] E. Rublee, V. Rabaud, K. Konolige, G. Bradsk, ORB: an efficient alternative to SIFT or SURF, in: International Conference on Computer Vision (ICCV), 2011, pp. 2564–2571.

[7] E. Rosten, R. Porter, T. Drummond, Faster and better : a machine learning approach to corner detection , IEEE Trans. Pattern Anal. Mach. Intell. 32(1), 2010, 105–119.

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA YOGYAKARTA
Email : fst@usd.ac.id

SURAT TUGAS

Nomor: 113/DKN/FST/ST/XI/2016

Pimpinan Fakultas Sains dan Teknologi Universitas Sanata Dharma Yogyakarta dengan ini memberikan tugas kepada:

N a m a	:	1. Puspaningtyas Sanjoyo Adi, S.T.,M.T. 2. Sri Hartati Wijono, S.Si.,M.Kom.
Pekerjaan/Jabatan	:	Dosen Prodi Teknik Informatika
Unit Organisasi	:	Fakultas Sains dan Teknologi Universitas Sanata Dharma
A l a m a t	:	Kampus III Paingan , Maguwoharjo,Depok, Sleman Yogyakarta Telp. (0274) 883037, Fax. (0274) 886529
Untuk Keperluan	:	sebagai Pemakalah dalam " INTERNATIONAL CONFERENCE ON SCIENCE AND TECHNOLOGY".
Tempat	:	Syahida Inn 6 th, floor,Kertamukti No 5 Ciputat, Kampus II Syarif Hidayatuah, State Islamic University Jakarta, Indonesia
Waktu	:	Rabu - Kamis, 16 - 17 November 2016
Biaya	:	Dana Penelitian DIKTI

Tugas ini harap dilaksanakan sebaik-baiknya, dan apabila sudah selesai agar segera memberikan laporan.

Yogyakarta, 14 November 2016
Dekan,

Sudi Mungkasi, S.Si.,M.Math.Sc.,Ph.D.

Yang bersangkutan telah melaksanakan tugas dengan baik sesuai waktu yang tertera dalam surat tugas.

Jakarta, 16 November 2016

(Dr. Cicatwati, Ir. MP)
Jabatan: Waki Dekan Bidang Akademik

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI (UIN)
SYARIF HIDAYATULLAH JAKARTA
FAKULTAS SAINS DAN TEKNOLOGI

Jl. Ir. H. Juanda No. 95 Ciputat 15412 Indonesia
Telp.: (62-21) 7493606, 7493547, 7401925 Fax.: (62-21) 7493315

Email : fst@uinjkt.ac.id
Website : fst.uinjkt.ac.id

Ref. No : Un.01/F9/HM.00.5/ 9820 /2016

Date : Oct 19, 2016

Subject : **Notification of Acceptance for The International Conference on Science and Technology (ICOSAT)**

Dear Puspaningtyas Sanjoyo Adi and Sri Hartati Wijono,

I am writing on behalf of committee of International Conference on Science and Technology (ICOSAT) 2016. Your abstract entitled "**ORB Image Detection in Android**" has been accepted for oral presentation at International Conference on Science and Technology (ICOSAT) that will be held on November 16-17, 2016 at Syahida inn 6th floor, Kertamukti street No. 5 Ciputat, Kampus II Syarif Hidayatullah State Islamic University Jakarta, Indonesia

You should submit your full paper by November 10, 2016. The paper will be published in the conference proceedings and selected papers will be published in an International Journal indexed by Scopus. Schedules and further information will be informed by email.

Yours Sincerely,

Dr. Elpawati, Ir. MP
NIP. 19641204 199203 2 001
Vice Dean of Academic Affairs

Cc:
Dean Faculty of Science and Technology

ORB Image Detection in Android

Puspaningtyas Sanjoyo Adi
Universitas Sanata Dharma
Yogyakarta, Indonesia
puspa@usd.ac.id

Sri Hartati Wijono
Universitas Sanata Dharma
Yogyakarta, Indonesia
tatik@usd.ac.id

Silverio R.L. Aji Sampurno
Universitas Sanata Dharma
Yogyakarta, Indonesia
silverio@usd.ac.id

Abstract—

In recent years, there has been rapid development of the technologies used in smart devices. Hence, research into these technologies is required in order to expand our understanding of these devices. One of the main features of smart devices is the mobile camera. Many studies have been conducted about how to utilizing service using camera. A key component of camera applications is image processing software. In this paper, we analyze the performance of the ORB algorithm for image detection in mobiles. We use Android applications with OpenCV library. Through this study, we find that ORB algorithm can work well for image detection in mobile applications.

Index Terms—

Feature detector, Computer vision, Image processing, ORB, OpenCV

I. Introduction

Currently the image features can be summarized as two categories: global features (such as color, texture, outline, etc.) and local features. They are used to describe global information and local details of the images respectively. A lot of experimental results show that the integrated features which combine these two kinds of features can improve recognition accuracy effectively [2].

Scale Invariant Feature Transform (SIFT) is the most representative local feature descriptor currently, which is invariant to image scale and rotation, and proven to be able to provide robust matching across a substantial range of affine distortion, variations of 3D viewpoint and illumination [3]. But the computational complexity of feature extraction is very high. Therefore, a variety of fast local feature extraction algorithms have been proposed, such as Speeded Up Robust Features (SURF)[4], Oriented FAST [7] and Rotated BRIEF [5] (ORB)[6] etc. Compared with SIFT, the extraction speed of these features are much faster than that of SIFT and they can achieve comparable performance when they are used in image classification, retrieval and other applications.

II. Computer Vision Technologies

A. ORB feature detection and descriptor matching

Currently, common feature point matching algorithms include SIFT [3], SURF [4], BRIEF [5], ORB [6]. The key factor affecting speed and accuracy of feature point matching is feature detection. There are two kinds of feature detection: descriptor based on absolute value and descriptor based on comparison. SIFT and SURF which stand for descriptor based on absolute value generally quantify gray or gradient to get a histogram, then construct descriptor based on histogram. Compared with SIFT and SURF, BRIEF and ORB based on comparison have great advantage. They construct the descriptor by comparing the characteristic value which is pre-trained or belongs to random points. This kind of descriptor is designed to improve speed. As can be seen from [5] and [6], it takes respectively 5228.7 (ms), 217.3 (ms), 8.87 (ms), and 15.3 (ms) for SIFT, SURF, BRIEF-32 and ORB to process one frame. Compared with ORB, BRIEF is faster but it do not have the rotation invariance. When people use the smart phone, the change of rotation and position is relatively frequent. So BRIEF is not suitable for feature point matching for smart phone. ORB is proposed based on BRIEF. It has rotation invariance because of adding a direction variance to each descriptor. So this paper chooses ORB descriptor.

B. Open CV

OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it is free for both academic and commercial use [1]. It has C++, C, Python and Java interfaces, and supports Windows, Linux, Mac OS, iOS and Android operating systems. OpenCV is designed for computational efficiency, with a strong focus on real-time applications. Since it is written in optimized C/C++, the library can take advantage of multi-core processing. Also, it supports low-level and high-level APIs (Application Programming Interfaces), and is suitable for real-time applications. OpenCV has a modular structure, which means that the package includes several shared or static libraries. The following modules are available:

(1) core: A compact module defining basic data structures, including the dense multi-dimensional array “Mat” and basic functions used by all other modules.
(2) imgproc: An image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on.
(3) video: A video analysis module that includes motion estimation, background subtraction, and object tracking algorithms.
(4) calib3d: Basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence algorithms, and elements of 3D reconstruction.
(5) features2d: Salient feature detectors, descriptors, and descriptor matchers.
(6) objdetect: Detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on).
(7) highgui: An easy-to-use interface for video capturing, image and video codecs, as well as simple UI capabilities.
(8) gpu: GPU-accelerated algorithms from different OpenCV modules.

III. Simulation Design

In order to improve development efficiency, we choose to do with OpenCV. OpenCV is an open source computer vision library, and it implements many common algorithms of image processing and computer vision. OpenCV provides two kind of interface, C++ and Java, the

paper chooses Java interface in order to maintain consistency with the architecture of Android developing tools.
We simulate an ORB algorithm based on the Android operating system in real-time camera. The flowchart of object detection process is shown in figure 1. At first, reference image is saved in application and application will compute and get key point extractor and its descriptor. Application will reference image descriptor. We use ORB key point extractor and descriptor. After that, video picture is taken from smart phone camera. Application will get a sample video frame as scene image. ORB algorithm calculates key point and descriptor. Furthermore, the application will calculate the level of accuracy (matching) or descriptor matcher between two descriptors, reference descriptor and scene descriptor. The calculation result is a list of list of a couple of key points reference image and scene image along with the distance between keypoint. Furthermore, the application will find good key point with the steps: 1) calculate the minimum distance of all couples are formed. 2) find a couple keypoint within less than 1.5 times the minimum distance. Couple this keypoint hereinafter referred to as good key point. Furthermore, the application will calculate homografi with the input parameter is a good key point and the key point of the scene image. Homografi calculation will result in recognition status outputs reference image, known or not. If recognized, then the application will draw a box poses as a sign that the reference image is recognized on the scene image.

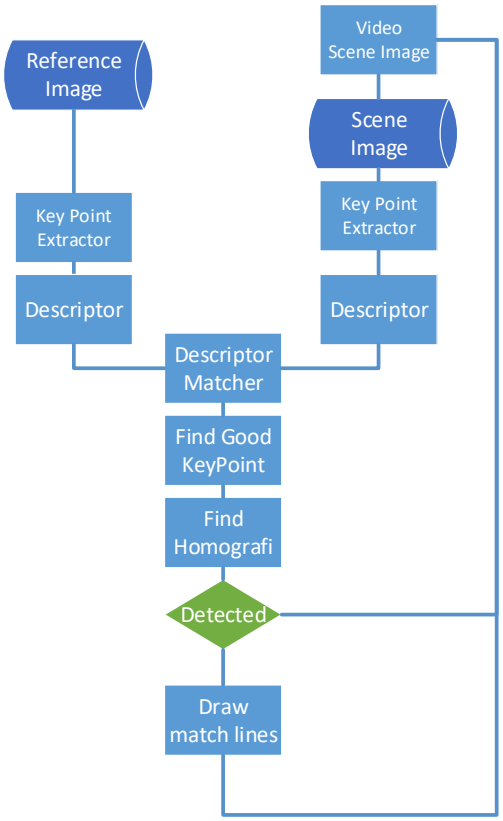


Figure 1. flowchart of object detection process

The Android application is designed based on the class diagram shown in Figure 2. `MainActivity` class is the main class in android application. `MainActivity` class implements the interface `CvCameraViewListener2`. This interface is the interface of the package OpenCV. This interface provides three functions: `onCameraFrame ()`, `onCameraViewStarted ()`, `onCameraViewStopped ()`. `OnCameraFrame()` is a

function that constantly works when the camera taking pictures. In `onCamera()`, the function `ImageDetectionFilter.apply()` is called. Function `apply ()` on class `ImageDetectionFilter` work according to flowchart image detection, corresponding figure 1.

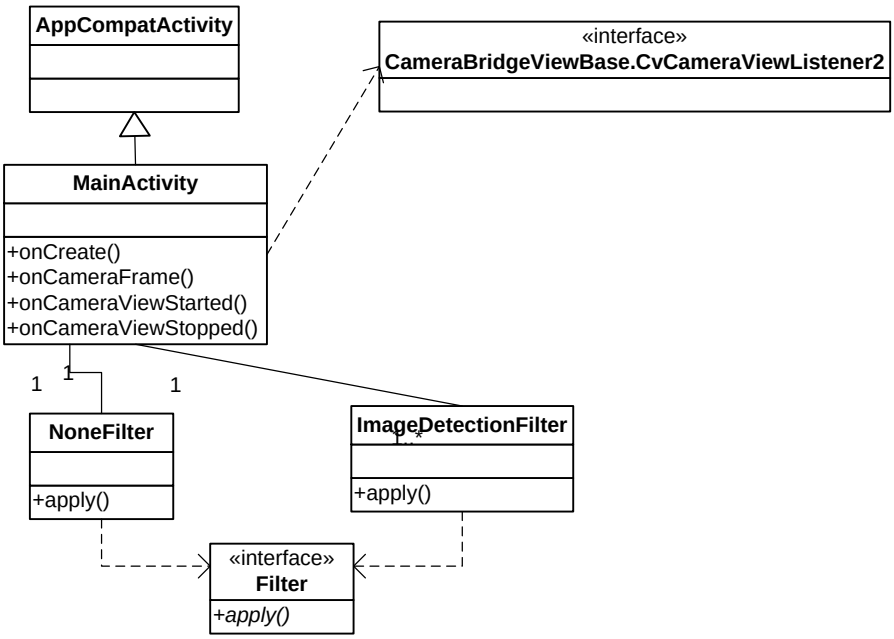


Figure 2. application class diagram

IV. Analysis of Result

Application tested by detecting four reference image, see figure 3. The application installed on the smartphone Samsung Galaxy Tab S2 series SM-T705. Smartphones fitted with a tripod to steady his position and exposed to the notebook display images or objects that have the same image with the reference image. Notebook will display a variety of images in which there are refence image.

			
Flight of sultan bahadur.jpg	starry_night.jpg	Self portrait.jpg	Picnic1.jpg

Figure 3. referece images

Image detection results can be seen in figure 4. Application can detect the reference image, picnic1.jpg, and making pose boxes and connecting lines between good key point in the scene image.

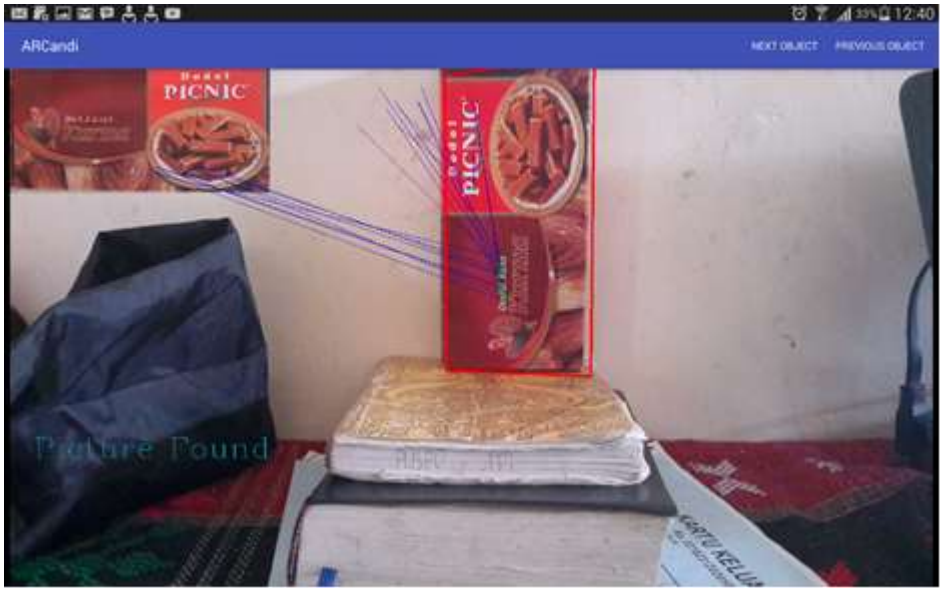


Figure 4. ORB detection result

Table 1. object detection time

file	average good keypoint	average detection time (mS)
flight_of_sultan_bahadur.jpg	10	773,07
starry_night.jpg	12	854,70
self_portrait.jpg	15	887,64
Picnic1.jpg	16	905,06

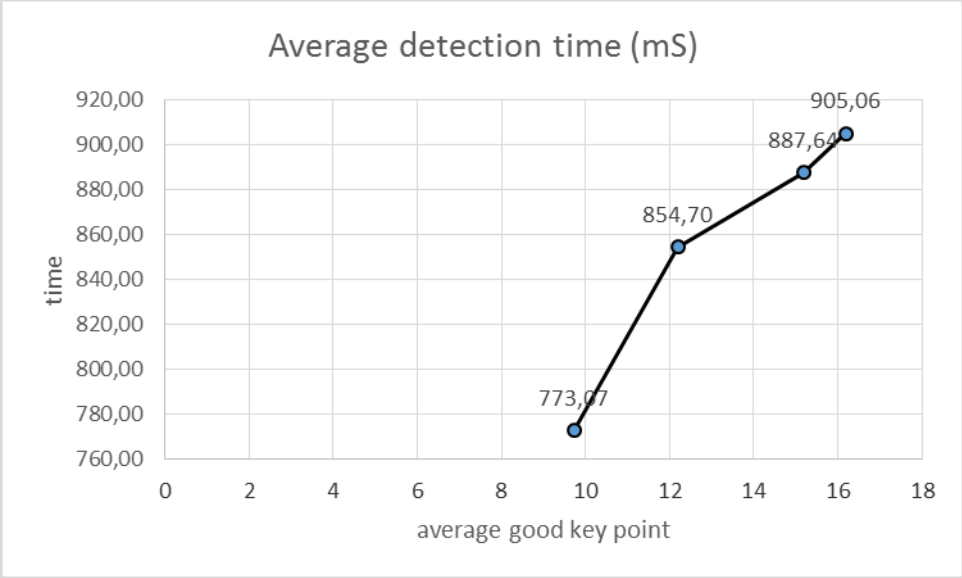


Figure 5. Object detection time graphic

Table 1 shows the mean time image detection. There are 4 image files are trying to identify. Each picture has a average good key point, which amounts vary between images. Figure picnic1.jpg who had a mean good key point of the most high has image detection time of 905.06 ms, while images flight_of_sultan_bahadur.jpg having a good key point average of 773.07 to have image detection time of 773.07 ms. Of figure 5, the average time image

detection increases in proportion with the addition of good key point. Good key point is the key point that has a distance of 1.5 times the minimum distance. This experiment has not been tried for various alternative minimum distance variables to find the optimal solution.

V. Conclusion

The result of this simulation Showed that the increasing of good key point will increase of detection time. Good key points is key points that have distance of 1.5 minimum distance. For further research, the minimum distance variables can be used as study materials to find the most optimum value.

Acknowledgment

The research work was supported by Penelitian Hibah Bersaing Kemenristekdikti Indonesia under Grant No. 010/HB-LIT/III/2016.

References

[1] OpenCV (Open Source Computer Vision), http://opencv.org.

[2] Yan Sun, Research on Some Algorithms of Content-based Image Retrieval and Sensitive Image Filtering (Ph.D. Dissertation), Jilin University, Jilin, 2011.

[3] David G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2), 2004, 91–110.

[4] B. Herbert, T. Tuytelaars, L.V. Gool, Surf: speeded up robust features, Computer Vision – ECCV, vol. 3951, 2006, pp. 404–417.

[5] M. Calonder, V. Lepetit, C. Strecha, P. Fua, Brief: binary robust independent elementary features, Computer Vision – ECCV 2010, vol. 6314, 2010, pp. 778–792.

[6] E. Rublee, V. Rabaud, K. Konolige, G. Bradsk, ORB: an efficient alternative to SIFT or SURF, in: International Conference on Computer Vision (ICCV), 2011, pp. 2564–2571.

[7] E. Rosten, R. Porter, T. Drummond, Faster and better : a machine learning approach to corner detection , IEEE Trans. Pattern Anal. Mach. Intell. 32(1), 2010, 105–119.