Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 1, January 2014, pg.325 – 333

RESEARCH ARTICLE

IMPLEMENTATION OF SPEED UP ROBUST FEATURE FOR DETECTION AND TRACKING OF INANIMATE OBJECTS

Fari Muhammad Abubakar[1]

[1]School of Electronics Engineering, Tianjin University of Technology and Education, Tianjin P.R China

[1] mofari@hotmail.com

Abstract— ***Object detection and tracking is one of the most researched areas in computer vision and is receiving a growing attention because of its wide area of applications which include surveillance, industrial inspection, robotics, mobiles, and 3D gaming among others. This paper focuses and presents the implementation of Speed Up Robust Feature in development of a Detection and Tracking System for inanimate objects. The system can detect an inanimate object from a still image containing many other objects which have been saved as data set as well as detect and track objects using webcam. The algorithm is developed in Microsoft Visual Basic 2010 express edition using Speed Up Robust Feature available in EmguCV libraries which are used for the image processing and computer vision tasks. A Logitech C310 High Definition webcam with 5 Mega Pixel is used for the purpose of real-time detection and tracking.***

Keywords— Object detection, tracking; EmguCV; Speed Up Robust Feature (SURF); Images

I. INTRODUCTION

As the use of cameras in areas of automobile, automation, electronic gadgets such as smart phones, computers and many other devices are becoming more popular and widely used nowadays. Computer vision utilizes the power of digital cameras to make devices and machineries have amazing capabilities to identify and differentiate between objects in both still and video (camera) images. Computer vision has a variety of applications in image and video processing making it one of the most researched fields. Also, computer vision [1] has a variety of sub-domains which include scene reconstruction, event detection, video tracking, indexing, object recognition etc. In computer vision and robotics, object detection is of great importance [2]. Applications in computer vision include video surveillance, traffic monitoring, industrial inspection, digital libraries, gadgets like mobiles, cameras etc. In robotics, the task of navigation, interactions with human and environment, all require object detection. Object detection serves as the most important goal to be realized for a fully autonomous agent. Humans can be described as special machines that stand out because of their high level of sophistication due to the intellect they possess which helps them make use of their vision, eyes to perform decisive tasks and operations in the recognition and identification of a variety of objects. Such objects include

fellow humans and other categories of objects such as books, cars, cups etc. Computer vision literature includes many object detection approaches based on representing objects of interests by a set of local 2D features such as corners or edges [3].

The vision is one of the most substantive and heavily used sensors by humans. With them, we interact with our immediate and surrounding environment using these visual sensors which are called the eyes. Out of many high- level tasks that we can accomplish with vision, object detection stands out. Object detection helps us to do a wide range of daily activities like moving around, interacting with people, reading and playing etc. With object recognition, we can easily differentiate various objects by recognizing the differences between them. For example, when we see a cup, our vision sensors with the help of the processing by the brain identify that this is an inanimate object. And also, when we come in contact with a person, same procedure is applied to recognize and identify that this is an animate. Therefore, object detection can pave way for building of algorithms as well as hybrid systems used in making computers identify the difference between objects in images. With respect to the significance and importance of object detection, recognition and tracking, however, it can be used in so many areas of computer vision to help and simplify the activities of humans. Many interest point detector-descriptor algorithms have been presented and are available but the choice of this paper is *SURF* due to its scale- and rotation invariant feature as well as its good performance and computational speed. Therefore, this paper hereby proposes the implementation and development of a system for real-time moving object detection and tracking using SURF.

II. RELATED WORKS

Samet Karakaş [4] presented thesis work for real time detection and tracking of moving objects with an active camera. For this purpose, feature based algorithms are implemented due to the computational efficiency of these kinds of algorithms and SURF (Speeded Up Robust Features) is mainly used for these algorithms. The algorithm was developed in C++ environment and OpenCV library frequently used. The developed algorithm is capable of detecting and tracking moving objects by using a PTZ (Pan-Tilt-Zoom) camera at a frame rate of approximately 5 fps and with a resolution of 640x480.

Ricardo Chincha and YingLi Tian [5] of Department of Electrical Engineering, The City College of New York New York, NY, 10031, USA proposes an object recognition method to help blind people find missing items using Speeded-Up Robust Features (SURF). SURF features can extract distinctive invariant features that can be utilized to perform reliable matching between different images in multiple scenarios. These features are invariant to image scale, translation, rotation, illumination, and partial occlusion. The proposed recognition process begins by matching individual features of the user queried object to a database of features with different personal items which are saved in advance.

III. OVERVIEW OF SPEED UP ROBUST FEATURE (SURF)

A. Interest Point Using SURF

The authors of *SURF* [6] used the very basic Hessian-matrix approximation for interest point detection and integral images which was made popular by Viola and Jones [7], which reduces the computation time drastically. Given a point x = (x, y) in an image I, the Hessian matrix $H(x, \sigma)$ in x at scale σ is defined as follows [6]

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{yx}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \quad (1.0)$$

where $L_{xx}(x, \sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$ with the image *I* in point x, and similarly for $L_{xy}(x, \sigma)$ and $L_{yy}(x, \sigma)$. [2]

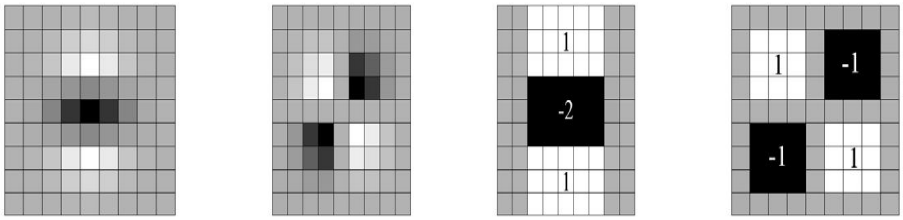


Fig. 1 Left to right: the (discretised and cropped) Gaussian second order partial derivatives in y-direction and xy-direction, and our approximations thereof using box filters. The grey regions are equal to zero and was obtained from [6].

B. Interest Point Description

The SURF descriptor [6] is based on similar properties to SIFT [8], with a complexity stripped down even further. The first step consists of fixing a reproducible orientation based on information from a circular region around the interest point [6]. Then, a square region aligned to the selected orientation is constructed; therefore the SURF descriptor is extracted from it. For more information regarding how the descriptors are described in SURF, the authors have done a very good work and have given extensive details on that which can be referred to in [6]. The process of generating distinctive descriptors involves a large number of memory accesses. SURF generates descriptors from a large number of sums of pixels distributed in a grid surrounding the interest point. SURF is intended to be highly invariant in rotation and scale and the system presented in this paper basis its implementation of the SURF feature to achieve these.

C. Interest Point Localization

Interest points need to have an exact location in the image for calculation purposes. Extracting these locations from the response maps is done by finding all local maxima. SURF finds local maxima in scale space; that is, in three dimensions, using non-maximum suppression in a 3x3x3 neighborhood around each pixel [9].

D. Interest Point Matching

Interest point matching involves figuring out which interest points in a new image correspond to that of the interest points already known from previous images. If there are correspondences between the interest points of the new image and that of the existing, the extracted descriptors around the interest points of both the two images are said to have points of correspondence making them have a successful match. The matching is often based on a distance between the vectors, e.g. the Mahalanobis or Euclidean distance [6]. The dimension of the descriptor has a direct impact on the time this takes, and a lower number of dimensions is therefore desirable [6].

E. Integral Images

One aspect of SURF that justifies the adjective Speeded-Up in its name is the use of integral images for speeding up computations. SURF makes extensive use of the integral image of a scene as part of interest point detection, and again as part of interest point description. The integral images are essential for completing these tasks with high efficiency at various scales. [9]. The integral image I_Σ of a source image I is an image where the value $I_\Sigma(x)$ of a pixel at a location $x = (x,y)^T$ is the sum of the values of all source pixels above and/or left of pixel x. More formally:

$$I_\Sigma(x) = \sum_{i=0}^{i \le x} \sum_{j=0}^{j \le y} I(i,j) \qquad (3)$$

Integral images are useful for getting the sum of pixels in a rectangular area of the source image in constant time. Only four lookups and three additions or subtractions are required regardless of the size of the area [9].

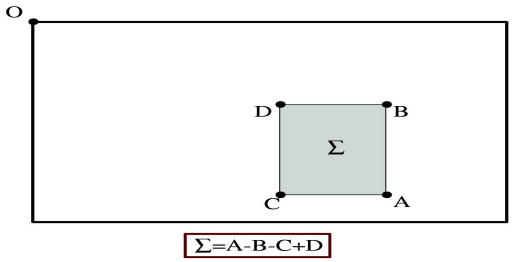


Fig.3 obtained from [6] Using integral images, it takes only three additions and four memory accesses to calculate the sum of intensities inside a rectangular region of any size.

IV. IMPLEMENTATION

A. Methodology and System set-up

The detection and tracking of the inanimate objects in still images as well as using a webcam can be carried out using the GUI of the system. For the proposed system, the Visual Basic setting is set to Expert Setting for the design. Thanks to Open CV and Emgu CV as to this system made use of the SURF feature in developing this system. Also, we will like to mention our sincere appreciation to the tutor in [10] for providing helpful and useful videos that gave us ideas on how to configure the EmguCV with VB 10 and understanding on how SURF implementation works as the system employed some of the codes from the tutorials. The development of the system goes into the following stages; firstly, the GUI is created using Microsoft Visual Basic 2010 express edition for which eight essential Emgu CV .dll reference library files were added and are as follows;

- Emgu.CV.DebuggerVisualizers.VS2010
- Emgu.CV
- Emgu.CV.GPU
- Emgu.CV.ML
- Emgu.CV.OCR
- Emgu.CV.Stitching
- Emgu.CV.UI
- Emgu.Util

The next step is, seventeen OpenCV .dll library files which come with the Emgu CV package were added which include;

opencv_calib3d240.dll, opencv_contrib240.dll, opencv_core240.dll, opencv_features2d240.dll, opencv_ffmpeg240.dll, opencv_flann240.dll, opencv_gpu240, opencv_highgui240.dll, opencv_imgproc240.dll, opencv_legacy240.dll, opencv_ml240.dll, opencv_nonfree240.dll, opencv_objdetect240.dll, opencv_photo240.dll, opencv_stitching240.dll, opencv_video240.dll, opencv_videostab240.dll

The system algorithm uses the following library packages from Emgu CV into visual basic 2010 which is .NET which were imported in the code window so as to achieve real- time moving object detection and tracking using a single active camera and also object detection and tracking in still images and if not found, the system will not be able to perform the detection and tracking task. These library packages that were imported include;

System.Drawing, System.Diagnostics, Emgu.CV, Emgu.CV.CvEnum, Emgu.CV.Features2D, Emgu.CV.Structure, Emgu.CV.UI, Emgu.CV.Util

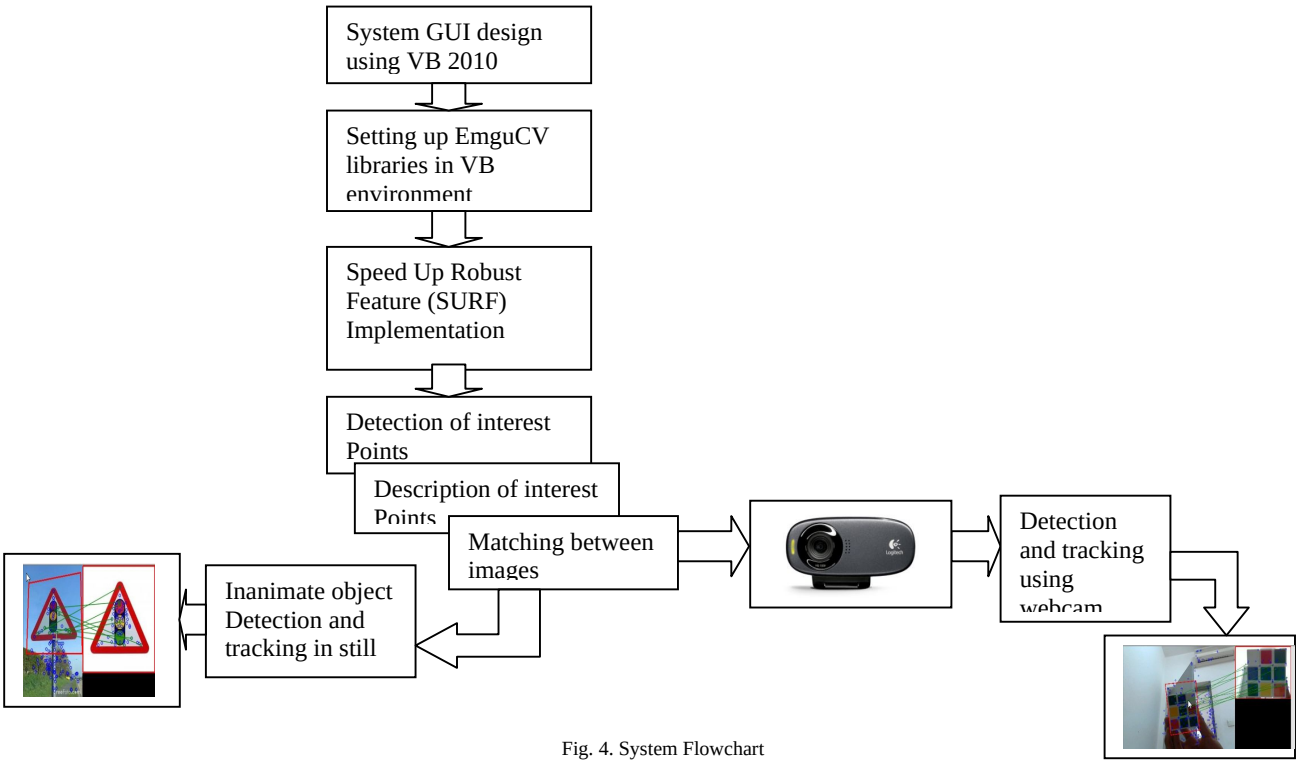
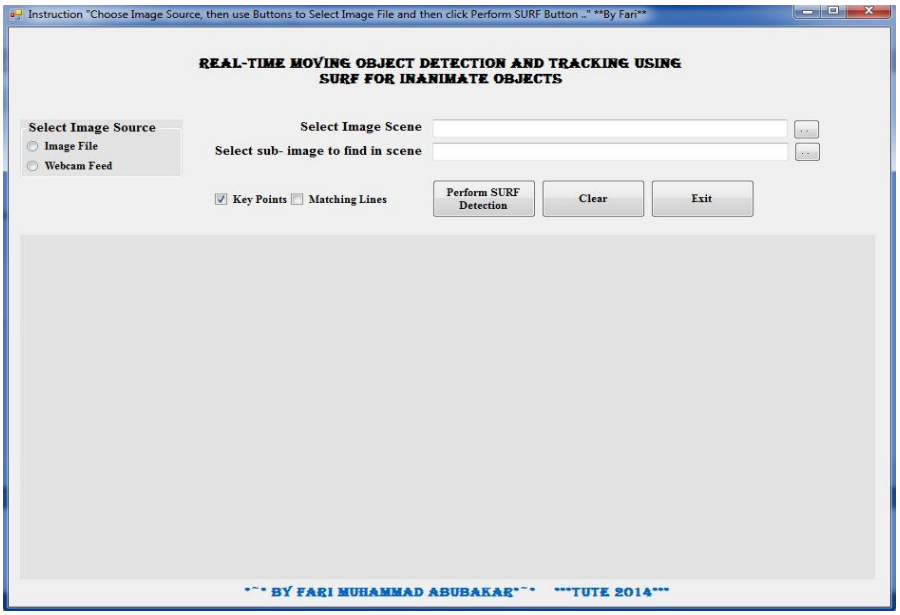


Fig. 4. System Flowchart

B. Equipment used for System Development

The paper focuses to present a system to perform object detection and tracking on inanimate with using webcam as well as for still images. For this purpose, C310 HD webcam produced by Logitech is used. The camera comes with a 5MP and has a built in mic for noise reduction. The camera is operated at 640*480 resolutions. The hardware requirement used to develop and run the system is standard PC, Dell N4050 which runs an Intel Core i3-2310M CPU @ 2.10GHz and 8 GB RAM. The GUI is developed using Microsoft Visual Basic 2010 express edition development environment and EmguCV library (EmguCV 2.4.0) is frequently employed. The EmguCV is a wrapper library that allows calling of OpenCV functions which is an open source having. The figure below shows the GUI of the proposed system;



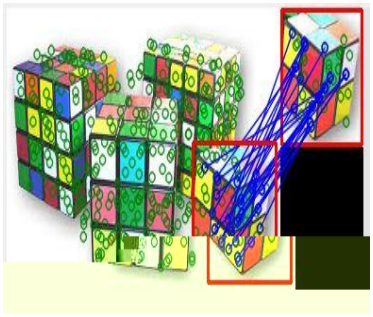
The Fig 4. shows the GUI of the proposed system

V. EXPERIMENTAL RESULTS

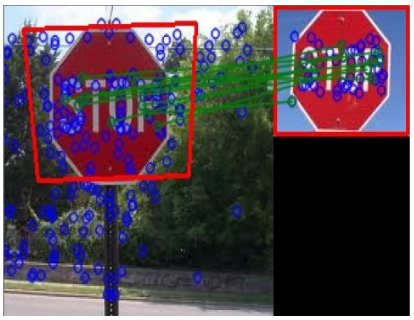
This section demonstrates some tests carried out on various still images obtained across the internet via Google search. These experimental tests are aimed at showing the main objective of this paper which is the implementation of SURF feature available in EmguCV in developing an algorithm to perform real-time moving object detection and tracking using a single webcam and also object detection and tracking in still images for inanimate objects. The detection and tracking is based on interest point detectors that checks for point of correspondence between the image scene containing the object with other objects and the sub-image which contains only the object of interest to be detected and tracked. These experimental results are obtained using the proposed interest point detection, description algorithm that has been discussed above. The proposed algorithm is based on implementation of the SURF which is a novel scale- and rotation-invariant detector and descriptor. SURF approximates or even outperforms previously proposed schemes with respect to repeatability, distinctiveness, and robustness, yet can be computed and compared much faster according to the authors [6]. These experiments were carried out on image scenes with few objects, medium, as well as one with many objects. For the real-time moving object detection and tracking using the webcam, a real live object was also tested on the system to check its performance in terms of scale, rotation, illumination.

A. System Test on Various Images

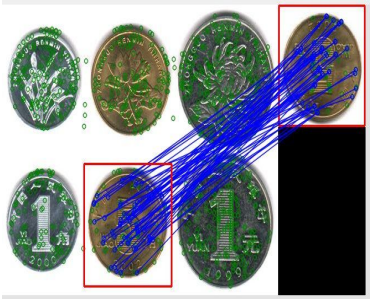
This section presents the two types of tests carried out on the system to detect inanimate objects. The first is detection of inanimate objects in still images whereas the is real- time moving object detection and tracking using a single active webcam for detection and tracking purposes is the second. For some images used for the experimental tests, the objects were cropped from the original image scene while for others; objects were obtained differently from the image scene. Before the test commenced, the C310 Logitech was connected to the PC after loading the system. When the webcam is detected by the system, the detection and tracking can be started. The aim is to test this system on various images in order to see how the system will be able to detect and track objects based on scale, rotation and illumination differences between the two images which one is the image scene containing many objects and the second is the object scene; that is the interested object to be found in the images scene. The figures below show results of system's detection and tracking;



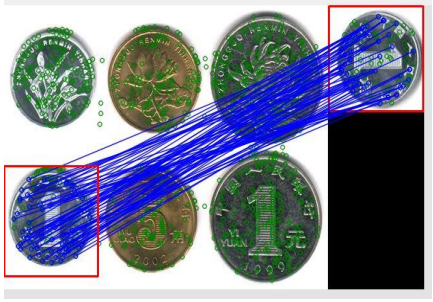
a.



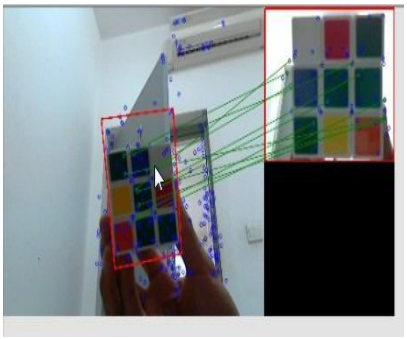
b.



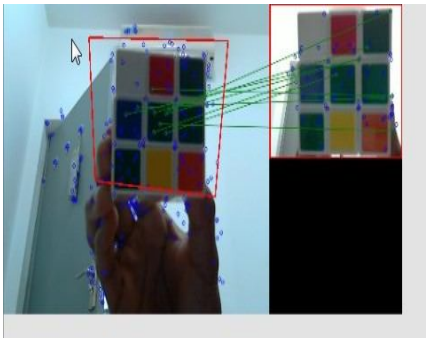
c.



d.



e.



f.

Fig. 5 shows two images where the object of interest (object) is by the upper right and image scene by the left hand side (Image scene). (a) To (f) shows successful objects detection based on scale, illumination and rotation parameters as both the two images matched each other due to same points of correspondence.

TABLE I

Image scene	**Objects of Interest (Image to find in scene)**	**Scale Different (✓) Same (X)**	**Illumination Different(✓) Same (X)**	**Rotated Yes (✓) No (X)**	**Detected (✓) Not Detected (X)**
Image scene 1(a)	Object (a)	X	X	X	✓
Image scene (b)	Object (b)	✓	✓	X	✓
Image scene (c)	Object (c)	✓	X	✓	✓
Image scene (d)	Object (d)	X	X	✓	✓
Image scene (e)	Object (e)	✓	X	✓	✓
Image scene (f)	Object (f)	X	X	X	✓

Table 1. Showing various tests performed on the system based on scale, rotation and illumination changes, object detection and tracking is achieved for inanimate objects successfully between two images having same points of correspondence making the interest points match between each other.

VI. CONCLUSION

This paper has made use of Microsoft Visual Basic 2010, a streamlined, easy-to-use and easy-to-learn IDEs for users other than professional software developers, such as hobbyists and students to develop the GUI and the algorithm for the system then implementation SURF feature and making use and image processing libraries available in EmguCV 2.4.0 was also done in VB development environment. The system implements the SURF which is a scale and rotation invariant feature which is based on interest points detector, descriptor. The paper aimed at developing real-time moving object detection and tracking system using a single webcam for inanimate objects. The system takes two image sources; in which the first is from a webcam feed and the second using still images.

For both real-time and still images, firstly, interest points are detected across distinctive locations in the image such as corners, edges etc., A distinctive property of the SURF detector is its repeatability i.e. whether it reliably finds the same interest points under different viewing conditions. Secondly, the neighborhood of every interest point is represented by a feature vectors. This descriptor has to be distinctive and, at the same time, robust to noise, detection errors, and geometric and photometric deformations [6]. Lastly, these features are matched for point of correspondence between the object and image scene. Both detection and tracking were achieved in real-time as well as using still images. Experimental results shown in Figure 5 above have shown that the SURF feature implementation in the system achieved interest point's detection with a robust performance under different scale, rotation and illumination parameters. These experimental tests are aimed at showing the main objective of this paper which is the implementation of SURF feature available in EmguCV in developing an algorithm to perform real-time moving object detection and tracking using a single webcam and also object detection and tracking in still images for inanimate objects.

ACKNOWLEDGMENT

Thanks to Open CV and Emgu CV as to this system made use of the SURF feature in developing this system. Also, we will like to mention our sincere appreciation to the tutor in [10] for providing helpful and useful videos that gave us ideas on how to configure the EmguCV with VB 10 and understanding on how SURF implementation works as the system employed some of the codes from the tutorials.

REFERENCES

[1] http://en.wikipedia.org/wiki/Computer_vision

[2] Yousuf A. "Visual Object Detection Using Frequent Pattern Mining," *A Thesis of Masters of Science submitted to Dept. of Computer Science and Engineering*, Indian Institute of Technology Madras, Jan 2011"

[3] Benhimane, Selim, et al. "Real-Time Object Detection and Tracking for Industrial Applications." *VISAPP (2)*. 2008.

[4] Samet Karakas "Detection and Tracking Moving Objects with an Active Camera in Real-Time," *MSc Thesis Submitted to the Graduate School of Natural & Applied Sciences of Middle East Tech. Univ.,* Sept 2011

[5] Ricardo Chincha and YingLi Tian [20] "Finding Objects for Blind People Based on SURF Features," *Department of Electrical Engineering The City College of New York New York, NY, 10031, USA {rchinch00, ytian}@ccny.cuny.edu*

[6] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer vision and image understanding*, *110*(3), 346-359.

[7] P.A. Viola and M.J. Jones. "Rapid object detection using a boosted cascade of simple features." *In CVPR* , pages 511 ,518, 2001.

[8] Lowe, David G. "Distinctive image features from scale-invariant keypoints." *International journal of computer vision* 60.2 (2004): 91-110.

[9] S.F. Arnesen "Structure From Motion in CUDA," *Master Thesis,* 1. November 2010

[10] http://www.18f4550.com/

