Research Article

Performance Analysis of Shape Recognition Algorithms on Embedded Boards RASPBERRY PI and ODROID C2

Sivasankari S[1*], Sivamurugan V[2]
Department of Information Technology, SRM University, Chennai, India.
Department of Information Technology, SSN College of Engineering, Chennai, India.
***Corresponding author's E-mail:** sivasankari.s@ktr.srmuniv.ac.in

Received: 07-01-2017; **Revised:** 28-02-2017; **Accepted:** 15-03-2017.

ABSTRACT

Shape recognition plays an important role in machine learning and vision applications.1 Shape recognition analyzes geometric shapes of objects automatically, for example using a computer to detect similarly shaped objects or parts that fit together. The shapes of different objects are to be predicted using different edge detection algorithms namely Canny, Laplacian and Sobel. These algorithms are made to run on boards Raspberry Pi and odroid c2. The performance analysis for the above mentioned algorithms on two different boards are compared based on execution time, memory usage and accuracy. From the results, accurate shape recognition algorithm that best fits the board is found.

Keywords: shape recognition, canny, sobel, LOG.

INTRODUCTION

Shape recognition represents a very important computer vision domain, consisting of recognizing image objects, based on their shape information. The shape recognition is mainly used in computer vision applications like fingerprint analysis, handwriting mapping, face recognition etc. Shape analysis is mainly used for automatic analysis of geometric shapes, using a computer to detect similarly shaped objects in a database or parts that fit together. The objects have to be represented in a digital form[2]. Most commonly boundary representation is used to describe the object with its boundary. However, other volume based representations or point based representations can be used to represent shape[3]. Once the objects are given, either by modeling, by scanning or by extracting shape from 2D or 3D images, they have to be simplified before a comparison can be performed. The simplified representation is often called a shape descriptor[4] (or fingerprint, signature).

These simplified representations will carry most of the important information, while being easier to handle. A complete shape descriptor is a representation that can be used to completely reconstruct the original object. Edge is the symbol and also reflection of discreteness of partial image.

Two different platforms are used for running different object recognition algorithms. The first platform is based on the open source ODROID C2 board based on the Application Processor Samsung's Exynos 5422.

The different algorithms used for shape recognition is implemented on ODROID C2 powered by ARM big. LITTLE technology, comes with a Heterogeneous Multi-Processing (HMP) Solution. ODROID-C2 is a new generation computing device with powerful and efficient hardware and smaller form factor. By having e-MMC 5.0 and USB 3.0 interface, the ODROID-C2 boasts fast data transfer speed, a feature is required to support advanced processing power on ARM devices that allows users can fully experiences an up gradation in computing such as faster booting, web browsing and 3D game experience. The performance of different algorithms is compared for the different performance metrics namely power consumption, energy efficiency while running the application on both Linux and Android operating system.

The second Platform used for the implementation of shape recognition algorithm is the Raspberry Pi , a series of credit card–sized and a single-board computers developed in Pencoed, Wales by the Raspberry Pi Foundations. All Raspberry Pi include processors with an ARMv6-compatible core or newer ARMv7 cores and have included on a chip with Video Core IV GPU, and have at 256 megabytes of RAM, except later models (models B and B+) that have 512 MB. The different shape recognition algorithms are compared for accuracy of shape recognion and execution time while running the application on this embedded board.

Related Work

Edge detection operator is mutation in nature. There are two main types: the first derivative-based edge detection to detect image edges by computing the image types[5] gradient values, such as Roberts operator, Sobel operator, Prewitt operator; the other is the second derivative-based edge detection operator,

Such as LOG operator, canny operator. Edge lies on the differential value of the maximum, minimum or zero.

$$Gx = f[x+1, y] - f[x, y]$$

$$Gy = f[x, y+1] - f[x, y] \quad (1)$$

Sobel operator[6] image use the two nuclear convolutions. One checks maximum response of the vertical edge and the other checks maximum response of the horizontal edge. The maximum value of two convolutions is referred as output value of the changing point.

LOG (Laplacian of Gaussian) operator finds the optimal filter of edge detection by ratio of the signal to noise of image. Canny[7] operators basic idea is to use a Gaussian function to smooth image firstly. Then the maximum value of first derivative corresponds to the minimum of the first derivative. The Canny algorithm is not susceptible to noise interference and enables its ability to detect true weak edges.

Proposed Work

In this method, the shape recognition system is designed and implemented using Android/Linux and ARM based Embedded Multicore platform. The shape recognition algorithms implemented in system level has to be run using the boards Raspberry Pi and ODROID C2 and its performance in these boards is evaluated based on performance metrics namely speed, accuracy and memory usage. Now a comparative study is made from the above mentioned algorithms and a graph will be generated based on the results obtained.

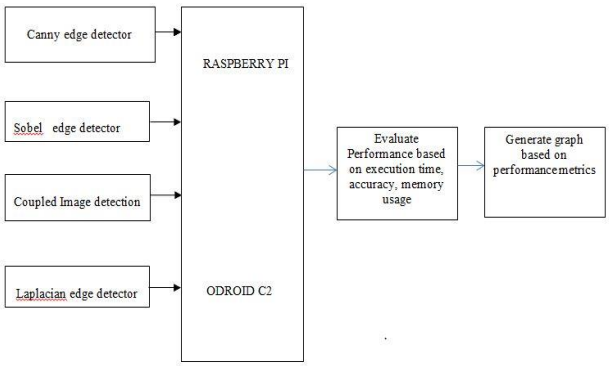

Figure 1: Architecture Diagram

Algorithm

Step 1: Load the input image using image read function

Step 2: Convert it to grayscale

Step 3: Use Canny, Sobel, Prewitt, Roberts and LOG edge detectors to detect edges of image.

Step 4: Find the contours of image.

Step 5: Use Clone function to copy source image to destination.

Step6: Approximate contour using approxPolyDP function.

Step 7: if (approx.size() == 3)

{

setLabel (dst, "TRI", contours[i]);

}

Step 8: Calculate the cosine angles to determine the shape as square or rectangle

If (vtc == 4 && mincos >= -0.1 && maxcos <= 0.3)

Step 9: Find ratio = std::abs(1 – r.width/r.height), if it is <=0.02 then it is SQUARE otherwise RECTANGLE.

Step 10 : Similarly vertex=5, for PENTAGON and 6 for HEXAGON.

Step 11: For Circles, calculate contours of bounding rectangle with radius equal to half of width and ratio should be <=0.2 for CIRCLE.

Step 12: Use image show function to display both source and destination image.

imshow("src", src);

imshow("dst", dst);

Step 13: Run on Raspberry pi and ODROID XU3 and compute the performance.

Step 14: Display the results and generate graph.

EXPERIMENTAL RESULTS

Raspberry Pi

- Installation of Raspbian OS in Raspberry pi.
- Configuration of Raspberry Pi
- Installation of opencv in Raspberry Pi
- Execution of canny, sobel, laplacian edge detection algorithm
- Execution of canny video file and coupled image detection algorithm

Steps to install openCV on Raspberry pi

- Sudo apt-get update
  sudo apt-get upgrade
- Install required libraries for openCV
- Download opencv from http://opencv.org/downloads.html
- Unzip openCV and prepare for build
- Create a new directory say 'release'
- ccmake ../
- Press 'c' to configure
- Change the settings as you want
- Press 'c' again to configure with your new settings
- Press 'g' to generate the makefile
- And finally build – make
- sudo make install

ODROID C2

- Get emmc card prebuild with ubuntu
- Configuing odroid C2
- Installation of opencv in odroid C2
- Execution of canny, sobel, laplacian, video file, coupled images detection in odroid C2.

The input image file consists of different geometric shapes namely square, rectangle, triangle, oval, circle, pentagon and hexagon.

Table 1: Time consumption of Raspberry Pi Vs Odroid C2

ALGORITHM	RASPBERRY PI	ODROID C2
CANNY	0.06 µs	0.000483 µs
SOBEL	0.31 µs	0.0035 µs
LAPLACIAN	0.08 µs	0.002723 µs
COUPLED	0.38 µs	0.03 µs

Time consumption of algorithms

Sobel>Laplacian>Canny
Best – Canny

Accuracy of algorithms:

Canny> laplacian&Sobel
Best - Canny

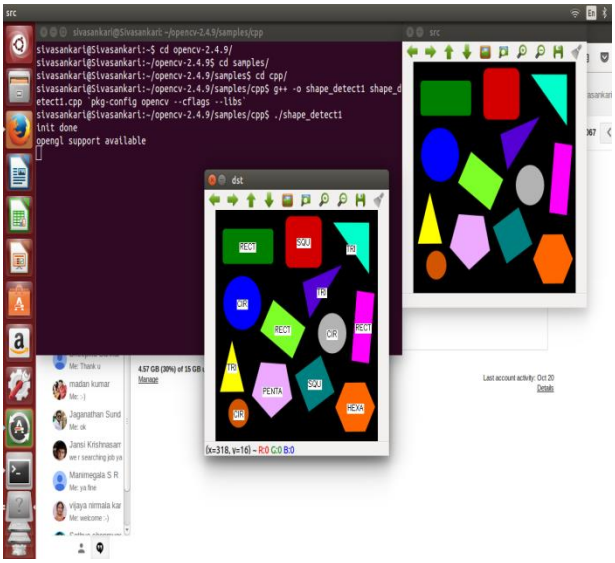

Figure 2: Output Image

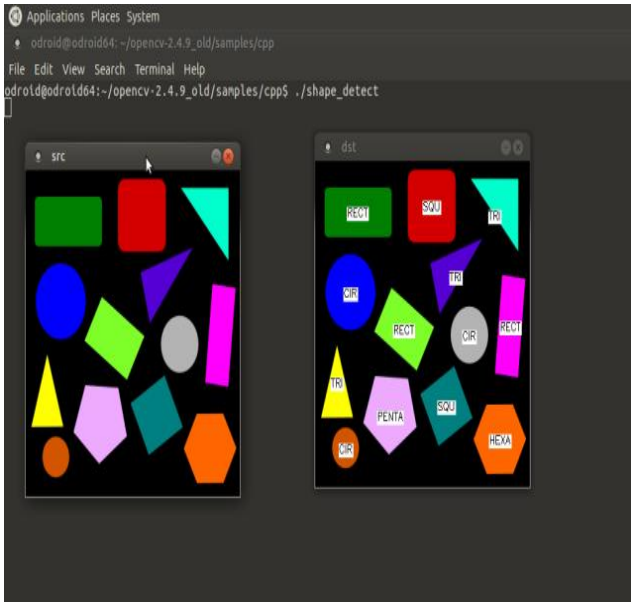

Figure 3: ODROID C2 output for Canny algorithm

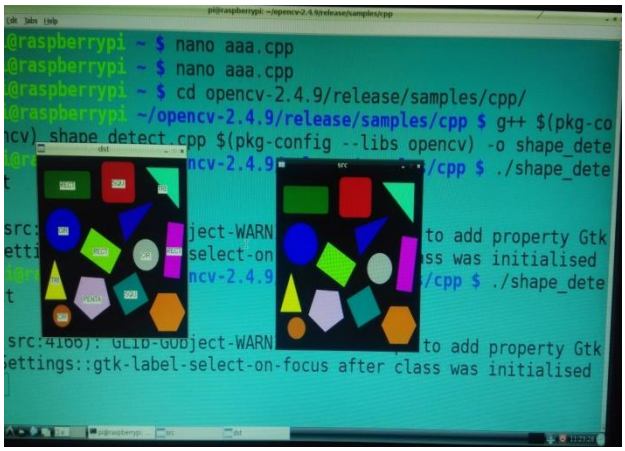

Figure 4: Raspberry Pi output for Canny algorithm

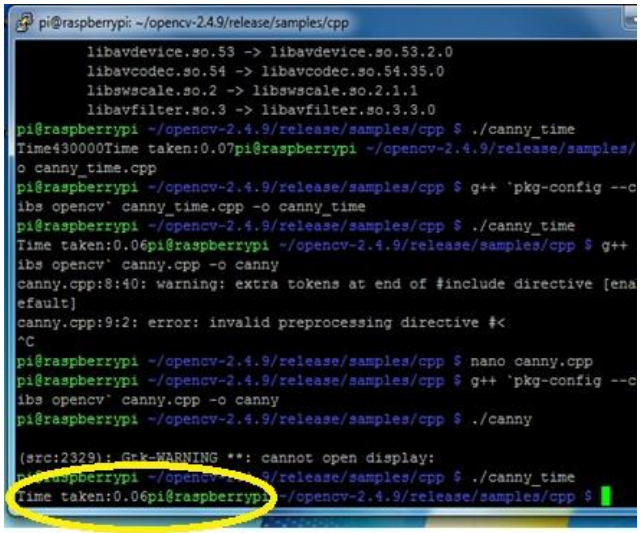

Figure 5: Time taken by canny algorithm in raspberry pi

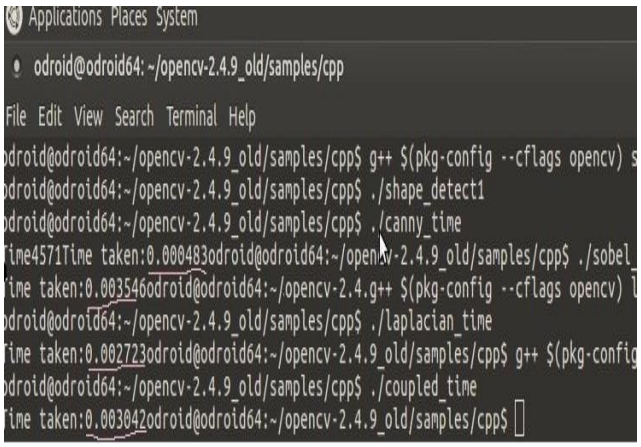

Figure 6: Time taken in Odroid C2

CONCLUSION

In this paper we implemented shape recognition system using edge detection algorithms that uses number of vertices and cosine angle to identify the shapes. These algorithms are made to run on boards raspberry pi and odroid c2 and their performance in these boards are evaluated based on performance metrics namely memory usage, execution time and accuracy which proves

- ODROID C2 gives good performance both in time and accuracy than raspberry pi.
- Canny is the best algorithm compared to sobel and laplacian.
- ODROID C2 supports video file whereas raspberry pi was not able to run video file to recognize shapes.

Table 2: Accuracy of Algorithms

ALGORITHM	RASPBERRY PI	ODROID C2
CANNY	83.33%	100%
SOBEL	16.66%	66.66%
LAPLACIAN	0%	66.66%

REFERENCES

1. Shikha Garg, Gianetan Singh Sekhon. Shape Analysis and Recognition Based on Over segmentation Technique. International Journal of Recent Technology and Engineering ISSN: 2277-3878, Volume-1, Issue-3, August 2012
2. Almaz J, Fornés A, Valveny E. A Non-Rigid Feature Extraction Method for Shape Recognition, International Conference on Document Analysis and Recognition, Spain, 2011
3. Thourn K, Kitjaidure Y. Multi-view Shape Recognition Based on Principal Component Analysis, International Conference on Advanced Computer Control Department of Electronics, Bangkok, 10520, 2008.

Source of Support: Nil, **Conflict of Interest:** None.