



***ISSN: 2278 – 0211 (Online)***

## **A Review Of Face Recognition System Using Wavelet Transform On Beagleboard-Xm**

**S. N. Gaikwad**

PG Student, Government College of Engineering, Aurangabad (M.S.), India

**P. H. Bhagat**

Assistant Professor,

Dept. of Electronics & Tele-Communication Engineering, Government College of Engineering, Aurangabad (M.S.), India

***Abstract:***

*The principle aim of this research paper is to investigate alternative methods to be used for face recognition, in particular the use of wavelets. This paper represents the standalone low cost face recognition system using beagleboard-xm, which is design specially to address the Open Source Community. Paper deploys USB camera as images acquisition equipment, processing and recognition is on images is perform using the Opencv software. Beagleboard use as a hardware platform for the performing the high speed image processing having the DM3730CBP processor operates on 1GHz. This paper's face recognition algorithms based on wavelet transform. This paper gives the brief review about the implementing wavelet transform and face recognition on beagleboard-xm.*

***Keywords:*** Face Detection, Face Recognition, Beagleboard-xm, OpenCV, wavelet transform, DM 3730.

## 1.Introduction

As compared with other biometrics systems using fingerprint/palm print and iris, face recognition has distinct advantages because of its non-contact process. Face images can be captured from a distance without touching the person being identified, and the identification does not require interacting with the person. In addition, face recognition serves the crime deterrent purpose because face images that have been recorded and archived can later help identify a person.

Over the past decade face detection and recognition have transcended from esoteric to popular area of research in computer vision. Since 2002, face detection can be performing using OpenCV's face detector, working in roughly 90-95% of clear images of person looking at camera. However face recognition is much less reliable than face detection, generally 30-70% accurate face recognition has been strong field of research since 1990s, but is still far from reliable and more techniques are being invented each year[1]. This paper shows how to implementing a system for face recognition using wavelet on the beagleboard-xm platform, having DM3730 which is the combination of Arm and DSP processor use for the high speed video processing. Using the open source software Opencv, which is the C and C++ library for the implementing the image processing algorithm. This system is operating on the Linux platform.

## 2.Literature Review

### 2.1.Hardware Design

This system uses beagleboard-xm as a hardware platform as shown in figure1, for performing images processing operation. The beagleboard-xM processor is the DM3730CBP 1GHz version and comes in a .4mm pitch POP package which is the combination of ARM and DSP processor and provide 512MB RAM[2].

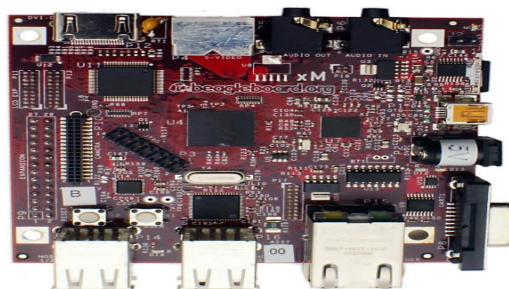


Figure 1: Beagleboard-xm

DM3730 is digital media processor compatible with OMAP3 architecture working upto 1GHz ARM Cortex-A8 Core also supports 300,600 and 800MHz operation. High performance image, video, audio (IVA2.2) accelerator system by using TMS320C64+ DSP Core and it also support operation on 260, 520 and 660 MHz operation [3]. BeagleBoard features powerful graphics capabilities application processor containing an integrated SGX hardware graphics accelerator. The SGX hardware is available to programmers through the standard OpenGL® ES 1.1 or 2.0 for 3D graphics and OpenVG™ for 2D scalable vector graphics [4]. In this system ARM Core is used for the data handling and DSP Core is used for the performing the high speed image processing. An USB camera is direct connecting to one of the USB slot of beagleboard-xm for the capturing the input and output is taken from the DVI-D port. Beaglaboard provide interface of micro SD memory card where the operating system can be install on that memory card. Minimum 4Gb microsd card is require for implementing this system.

### *2.2. Software Platform*

The entire Face recognition system has been implemented using C/C++ programming language. The platform has been carried out using the OpenCV (Open Source Computer Vision Library) On the Linux platform. Programming is done using GCC/ G++ compiler on Linux system.

OpenCV is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms and it support multi operating system. OpenCV provide a modular structure, which means that the package includes several shared or static libraries[5]. The following modules are used for system:

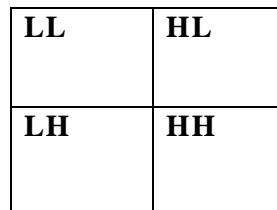
- core - a compact module defining basic data structures, including the dense multi-dimensional array Mat and basic functions used by all other modules.
- imgproc - an image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on.
- Video - a video analysis module that includes motion estimation, background subtraction, and object tracking algorithms.
- calib3d - basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence algorithms, and elements of 3D reconstruction.
- features2d - salient feature detectors, descriptors, and descriptor matchers.

- objdetect - detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on).
- highgui - an easy-to-use interface to video capturing, image and video codec's, as well as simple UI capabilities.
- GPU - GPU-accelerated algorithms from different OpenCV modules.
- ML -namespace contains machine-learning interfaces[6].

### 2.3. Wavelet Analysis

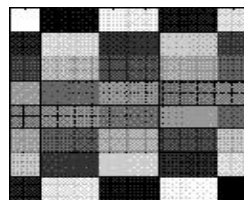
Wavelet transform is an increasingly popular tool in image processing and computer vision. Many applications, such as compression, detection, recognition, image retrieval. Wavelet transform has the nice features of space-frequency localization and multi-resolutions.

This paper use wavelet transform for face recognition in order of row and column. Wavelet decomposed image into four sub-bands as shown in figure 2. According to the wavelet theory sub-band LL keep main energy of original signal, other sub-bands contain detail information and overall characteristic of the image. [7]



*Figure 2: Wavelet Decomposition*

Now let us see how to wavelet transform perform on image. Image is represented by pixels in 2D matrix form. For performing wavelet transform on 2D matrix we have to perform two times 1D wavelet transform. First apply the 1D wavelet transform on rows of matrix then apply 1D wavelet on the columns of matrix. Consider the 8x8 image taken from a specific portion of a typical image shown in Figure3. The matrix (a 2D array) representing this image is shown in Figure 4.



*Figure3: A 8x8 image*

64	2	3	61	60	6	7	57
9	55	54	12	13	51	50	16
17	47	46	20	21	43	42	24
40	26	27	37	36	30	31	33
32	34	35	29	28	38	39	25
41	23	22	44	45	19	18	48
49	15	14	52	53	11	10	56
8	58	59	5	4	62	63	1

*Figure 4: 2D array representing figure3*

32.5	0	0.5	0.5	31	-29	27	-25
32.5	0	-0.5	-0.5	-23	21	-19	17
32.5	0	-0.5	-0.5	-15	13	-11	9
32.5	0	-0.5	0.5	7	-5	3	-1
32.5	0	0.5	0.5	-1	3	-5	7
32.5	0	-0.5	-0.5	9	-11	13	-15
32.5	0	-0.5	-0.5	17	-19	21	-23
32.5	0	0.5	0.5	-25	27	-29	31

*Figure5: Transformed array after operation on each row of Figure 4*

32.5	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	4	-4	4	-4
0	0	0	0	4	-4	4	-4
0	0	0.5	0.5	27	-25	23	-21
0	0	-0.5	-0.5	-11	9	-7	5
0	0	0.5	0.5	-5	7	-9	11
0	0	-0.5	-0.5	21	-23	25	-27

*Figure 6: Final Transformed Matrix after one step*

Now we perform the operation of averaging and differencing to arrive at a new matrix representing the same image in a more concise manner. Let us look how the operation is done. Consider the first row of the Figure4.

Averaging:  $(64+2)/2=33$ ,  $(3+61)/2=32$ ,  $(60+6)/2=33$ ,

$(7+57)/2=32$

Differencing:  $64-33=31$ ,  $3-32=-29$ ,  $60-33=27$  and

$7-32=-25$  So, the transformed row becomes  $(33\ 32\ 33\ 32\ 31\ -29\ 27\ -25)$ . Now the

same operation on the average values i.e.  $(33\ 32\ 33\ 32)$  is performed. Then we perform the same operation on the averages i.e. first two elements of the new transformed row.

Thus the final transformed row becomes  $(32.5\ 0\ 0.5\ 0.5\ 31\ -29\ 27\ -25)$ . The new matrix

we get after applying this operation on each row of the entire matrix of Figure 4 is shown

in Figure 5. Performing the same operation on each column of the matrix in Figure 5, we

get the final transformed matrix as shown in Figure 6. This operation on rows followed by columns of the matrix is performed recursively depending on the level of

transformation meaning the more iteration provides more transformations. Note that the

left-top element of the Figure 6 i.e. 32.5 is the only averaging element which is the overall average of all elements of the original matrix and the rest all elements are the

details coefficients [8]. Opencv provide image in the 2D matrix using `cv::Mat` command form so that it is easy to perform the pixel operation in c/c++ language[9].

### **3.Implementation Procedure**

For the beginning of the system we require to start the beagleboard-xm, for that we have port the operating system on the memory card and then this card interface with beaglboard-xm. For porting operating system on the microSD card we have to partition the memory card into 2 partitions one for the kernel file system and other for operating system file system [10]. This system use Angstrom distribution of linux comes with opkg packages. At the time of building image from their official site [11] we select additional gcc/g++ and opencv packages for the compiling c/c++ program and image processing. After completing the porting operating system we are ready to perform face recognition task. Face Recognition generally involves two stages:

#### *3.1.Face Detection*

where a photo is searched to find any face, then image processing cleans up the facial image for easier recognition.

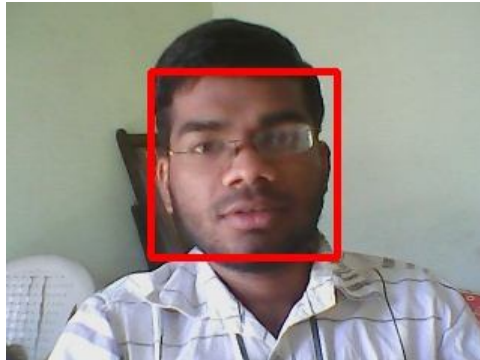
#### *3.2. Face Recognition*

where that detected and processed face is compared to a database of known faces, to decide who that person is.

OpenCV uses a type of face detector called a Haar Cascade classifier. Given an image, which can come from live video, the face detector examines each image location and classifies it as "Face" or "Not Face." Classification assumes a fixed scale for the face, say 50x50 pixels. Since faces in an image might be smaller or larger than this, the classifier runs over the image several times, to search for faces across a range of scales. We will need to tell the classifier where to find the data file you want it to use. The one we will be using is called `haarcascade_frontalface_default.xml`. In OpenCV version 2.0, it's located at `[OPENCV_ROOT]/data/haarcascades/haarcascade_frontalface_default.xml` where `[OPENCV_ROOT]` is the path to OpenCV installation[12].

After detecting the face from the image or live video we have to preprocessing like cropping the face part from the remaining image and convert it to gray image and it is used to apply wavelet transform on it. After the applying wavelet transform image is decomposed into four sub-bands LL, HL, LH, HH. Sub-band LL keeps main energy of

system so that is LL is use as feature from the image, other side band are not used. After extracting the feature from image save it as .xml file in the data base. Basically, to create a face recognition database from training images, we create a text file that lists the image files and which person each image file represents. Figure 7, show the result of face detection from the camera, preprocessing and LL part of wavelet transform



*(a) Face detection using camera*



*(b) Cropping face*



*(c) LL part of Wavelet*

*Figure 7: Result of Face Detection and wavelet Transform*

For the face recognition, take input as live video from camera using the haar cascade classifier detect the faces and apply the wavelet transform on it. After getting wavelet transform compare the LL sub-band with .xml files from data base. If there is any matching with any .xml file then it will show the name of person from database.



**4. Conclusion**

This review article gives the brief overview of face recognition using wavelet transform on beagleboard-xm. This approach appears to be quite perspective, robust against facial hair, glasses and also generally very robust compared to other methods. This paper uses the open source software for implementing the face recognition so that the cost of system is low as compare to other systems. This paper expanded the application fields for the DM3730 chips, meanwhile provided a hardware platform which can be more rapid developed an application system like face recognition system and iris recognition system.

Face recognition systems are no longer limited to classification, identity verification and surveillance tasks. Growing numbers of applications are starting to use face recognition as the initial step towards interpreting human actions, intention, and behavior, as a central part of Next-Generation Smart Environments.

**5.Reference**

1. Shervin Emami, Introduction to face Detection and Face Recognition  
<http://www.shervinemami.info/faceRecognition.html>
2. Beagleboard-xm system reference manual Revision C.1.0, April 2010 by  
beagleboard.org
3. Texas Instruments Incorporated. DM3730, DM3725 Digital Media Processors,  
SPRS685D–AUGUST 2010–REVISED JULY 2011
4. Diego Dompe, Embedded Software Team Lead RidgeRun, Introduction to  
creating 3D UI with BeagleBoard, White paper ESC Bosten2008 Class:ESC-341
5. OpenCV Documentation,  
<http://opencv.willowgarage.com/documentation/cpp/index.html>
6. Valentin Petrut SUCIU, Facial Recognition using OpenCV, Journal of Mobile,  
Embedded and Distributed Systems, vol. IV, no. 1, 2012, ISSN 2067 – 4074
7. Kamireddy Archana and D.R. Sandeep,A palm-print recognition system based on  
DM3730, International Journal of Engineering Research & Technology (IJERT)  
ISSN: 2278-0181 Vol. 1 Issue 8, October - 2012
8. Kamrul Hasan Talukder and Koichi Harada, Haar Wavelet Based Approach for  
Image Compression and Quality Assessment of Compressed Image, IAENG  
International Journal of Applied Mathematics, 36:1, IJAM\_36\_1\_9
9. Robert Laganieri, OpenCV 2 computer vision Application Programming  
cookbook
10. Porting Angstrom and Ubuntu to Beagle Board XM Revision C Article,  
<http://beaglexmc.blogspot.in/>
11. <http://www.angstrom-distribution.org/builder/>
12. Robin Hewitt , Seeing With OpenCV, Part 2: Finding Faces in Images Article,
13. [http://www.cognotics.com/opencv/servo\\_2007\\_series/part\\_2/index.html](http://www.cognotics.com/opencv/servo_2007_series/part_2/index.html)