Cultivating Research in Computer Vision within Graduates and Post-Graduates using Open Source

Nilesh J. Uke
Department of Information Technology
Sinhgad College of Engineering Vadgaon, Pune

Ravindra C. Thool
Department of Information Technology
S.G.G.S. College of Engineering and Technology, Nanded

ABSTRACT

In this paper we discuss suitability of the using an open source environment; OpenCV for providing effective solutions for complex image processing and vision algorithm for real time application for UG and PG students projects. Computer Vision (CV) applications require extensive knowledge of digital signal processing, mathematics, statistics and perception. OpenCV is an open source vision library suitable for such computer vision programs. It's always benefits to students to learn theoretical aspects of CV concepts by practicing in labs and getting hand-on expertise. In this paper, we describe our experience of using open source library for post-gradates and undergraduate students' teaching. We also report the experience of developing the projects based on computer vision to Bachelor of Engineering in Information Technology (BE-IT) and Master of Engineering in Information Technology (ME-IT) courses. Initial tutorials designed help students to understand basic concepts of computer vision. Small applications development gives lot of confidence in executing complex assignments and moderate level projects. Statistics shows that projects undertaken by students had increased due to extensive practice on openCV environment. Finally we describe few example project and dissertation submitted in last two years by the students.

General Terms

Computer Vision, Open Source Environment

Keywords

Computer Vision, OpenCV, Object detection, segmentation, Object tracking

1. INTRODUCTION

There is an exponential growth of secondary storage, memory capacity, processor speed, cheap cameras. This had increased the applications areas in Computer Vision (CV) and related research. CV area is vast and continues to grow but there is no standard API for development of complex algorithms. To enhance the joy of learning and solving real life problems, design of curriculum should be such where a student enhances learning interest and motivation towards study. Many technical institutes are trying to add Computer Engineering and Information Technology courses along-with the core courses such as Civil Engineering, Mechanical Engineering and Electronics. Computer vision is a compulsory course in most of the Universities. Sometimes it is offered as electives course in few colleges.

Computer Video based systems are becoming an important part of applications around us, such as human-computer interface, automated video surveillance, object identification, gesture recognition, and immersive gaming. A considerable number of computer vision and image processing software packages have been developed for various applications. Mostly these software packages are written in C/C++ to take advantage of computational speed [1].

OpenCV stands for Open Source Computer Vision Library and is designed in C & C++ specifically for increased computational efficiency and supported by most Operating Systems [2]. Example applications of the OpenCV library include Human-Computer Interaction (HCI), Object Identification, Segmentation and Recognition, Face Recognition, Gesture Recognition, Camera and Motion Tracking, Ego Motion, Motion Understanding, Structure-From-Motion (SFM), Stereo and Multi-Camera Calibration and Depth Computation and Mobile Robotics [2]. Basic goal of Computer Vision application is to make useful decisions about real objects in the scene. It is an emerging discipline and still there is need to formulate the standardization of many algorithms for Computer Vision problems.

This paper investigate and reports the experience of developing the projects based on Computer Vision to Bachelor of Engineering in Information Technology (BE-IT) and Master of Engineering in Information Technology (ME-IT) course.

2. OPEN SOURCES FOR COMPUTER VISION

Unlike computer graphics standard libraries like OpenGL and DirectX, computer vision currently have no standard API for developing application. The library is actively used by a large number of companies (such as Intel, IBM, Microsoft, SONY, Siemens, Google) and research centers (Stanford, MIT, CMU, Cambridge, INRIA) for research and development [2]. Following are few of the other open source libraries used for CV applications:

- The Open Source Vision Library (OpenVL) and its Utility Toolkits (VLUT) are designed to address efficiency, reusability and scalability to facilitate progress in computer vision. By providing a hardware development middleware that supports different hardware architectures for acceleration, OpenVL allows code reusability without compromising performance [3].
- IPP provides cross-platform interface to highly-optimized low-level functions that perform domain-specific operations, particularly, image processing and computer vision primitive operations [4].
- DIPlib is a platform independent scientific image processing library written in C language [5]. It consists of a large number of functions for processing and analyzing multi-dimensional image data. The library provides functions for performing transforms, filter

operations, object generation, local structure analysis, object measurements and statistical analysis of images. Key design features include ample support for different data types (binary, integer, floating point, complex) and dimensionalities. The current release contains over five hundred (500) documented functions, and more than three hundred (300) of these functions provide image processing functionality. I/O functionality is available in a separate library called dipIO, and provides support for ICS, TIFF and JPEG files.

- cvBlob is a library for computer vision to detect connected regions in binary digital images [6]. Some of the features include Binary image labeling, Features extraction and render, Contours functions and Basic blob tracking.

- VXL (the Vision-something-Libraries) is a collection of C++ libraries designed for computer vision research and implementation. It was created from TargetJr and the IUE with the aim of making a light, fast and consistent system [7].

- The VLFeat open source library implements popular computer vision algorithms including SIFT, MSER, k-means, hierarchical k-means, agglomerative information bottleneck, and quick shift [8]. It is written in C for efficiency and compatibility, with interfaces in MATLAB for ease of use, and detailed documentation throughout.

- SimpleCV is a Python module integrating classic computer vision algorithms and libraries. Developer can have access to high level algorithms in feature detection, filtering, and pattern recognition in a unified framework. [9] Integrated iPython interactive shell makes experimenting easy for non programmer too.

- VIGRA stands for "Vision with Generic Algorithms". It's a novel computer vision library that puts its main emphasis on customizable algorithms and data structures. By using template techniques similar to those in the C++ Standard Template Library, you can easily adapt any VIGRA component to the needs of your application, without thereby giving up execution speed. [10]

- EJS+EjsR - Easy Java Simulations (EJS), an open-source tool [11] which provides support for creating applications with a full 2D/3D interactive graphical interface, and EjsRL, a high-level Java library specifically designed for EJS which provides a complete functional framework for modeling and simulation of arbitrary serial-link manipulators, computer vision algorithms and remote operation.

Many commercial image processing and computer vision software packages have also been developed [2]. They provide high-level image processing and machine vision such as LabView [12], can be accelerated with special image processing hardware. Developed by Microsoft, the vision SDK is a low-level C++ library for image manipulation and analysis [13]. Image processing toolkit in MATLAB and digital image processing package in Mathematica are convenient image processing tools, taking the advantage of many advanced numerical functions at the same time. Developed by Aurora Co., LEADTOOLS [14] is a set of commercial computer vision packages. It includes an Image Server which can be used to build a web-based image processing server. But they do not include advanced functions such as moving objects tracking, pose recognition, face recognition and 3D reconstruction, etc. And it is complicated to interface with existing computer vision library and code in C/C++ material.

2.1 Important Features of OpenCV

An important advantage of adopting OpenCV for teaching and learning purposes is that it is open source. This not only reduces costs but also allows the increasing number of students with their own computer and a web cam to work at home and to continue using the library after the project [15]. OpenCV Developers Library includes over 500 functions implementing computer vision, image processing and general-purpose numeric algorithms [2], [16]. Few important features are listed below:-

- Image data manipulation (allocation, release, copying, setting, conversion).
- Image and video I/O (file and camera based input, image/video file output).
- Matrix and vector manipulation and linear algebra routines.
- Various dynamic data structures (lists, queues, sets, trees, graphs).
- Basic image processing (filtering, edge detection, corner detection, sampling and interpolation, color conversion, morphological operations, histograms, image pyramids).
- Structural analysis (connected components, contour processing, distance transform, various moments, template matching, Hough transform, polygonal approximation, line fitting, ellipse fitting).
- Camera calibration (finding and tracking calibration patterns, calibration, fundamental matrix estimation, homograph estimation, stereo correspondence).
- Motion analysis (optical flow, motion segmentation, tracking).
- Object recognition (Eigen-methods, HMM).
- Basic GUI (display image/video, keyboard and mouse handling, scroll-bars).
- Image labeling (line, conic, polygon, text drawing). always

3. SETTING UP IDE

Developers always need a free, open-source Integrated Development Environment (IDE) for software application development. OpenCV libraries could be configured and used with many IDE [16]. Most prominently used IDE’s are Visual C++, Eclipse, Dev C++, NetBeans, Qt, Code::Blocks.

The linking of OpenCV library with any DevC++ by BloodShed is discussed here. Bloodshed Dev-C++ is distributed under the GNU General Public License and free for use. In windows, assuming we have installed OpenCV into C:/Program Files/OpenCV, the settings you require are:

```
Additional Include Directories:
"C:\Program Files\OpenCV\cxcore\include";
"C:\Program Files\OpenCV\cv\include";
"C:\Program Files\OpenCV\cvaux\include";
"C:\Program Files\OpenCV\otherlibs\highgui";
"C:\Program Files\OpenCV\otherlibs\cvcam\include"
Additional Library Directories:
"C:\Program Files\OpenCV\lib"
Additional Dependencies (libraries):
“cv.lib cxcore.lib highgui.lib”
```

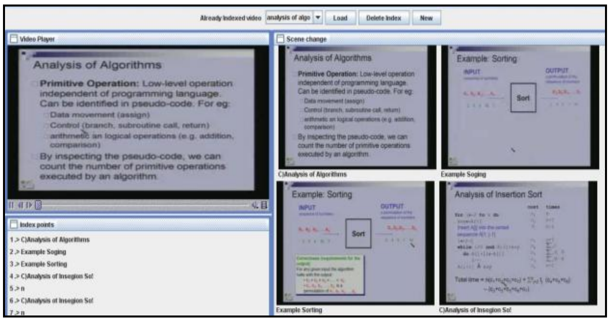
In [16] these IDE setting tasks are presented in a way that is independent of any particular environment and operating system, so you are free to use the one of your choice. But it is warned to use the compiled version of the OpenCV library that is appropriate to the compiler and operating system you are using. In order to use OpenCV under the environment of your choice, you need to generate the library binary files using the appropriate C++ compiler. To build OpenCV, you need to use the CMake tool. CMake is another open source software tool designed to control the compilation process of a software system using platform-independent configuration files.

4. PROJECTS DEVELOPED

By the time students appears for final year, they become moderately expert in C and C++ language. These two languages become their mother tongue. Advancement towards OpenCV becomes easy with few tutorials and examples programs. This tutorial enables students to appreciate the components of a computer vision system, to develop their beginner's skills in programming complex algorithms. The project normally occupies about 208 hours, spread over 2 semesters, and can take the form of an individual (PG course) or a group project (UG course). This section briefly explains the projects studied by the students in last 3 academic years.

4.1 Organization of pre recorded lecture video

This work focuses on playing lecture videos which are captured while teaching in the class using presentations and its contents organizations. Initially the long video is divided into slides frames. Lecture video is indexed for effective retrieving and navigation though a long video. This is achieved by selection of important frames (slides) for indexing using the contents of the frames. Figure 1 shows a pre recorded lecture video (Analysis of Algorithm) played in a video player. Slide change detection procedure is extracted from the video being played and the index points are generated [17]. Clicking on the index point, video can be browsed effectively from any random position depending upon the selection.



Analysis of Algorithms

Primitive Operation: Low-level operation independent of programming language. Can be identified in pseudo-code. For eg:

- Data movement (assign)
- Control (branch, subroutine call, return)
- arithmetic an logical operations (e.g. addition, comparison)

By inspecting the pseudo-code, we can count the number of primitive operations executed by an algorithm

Example: Sorting

Analysis of Insertion Sort

1.> C)Analysis of Algorithms
2.> Example Soging
3.> Example Sorting
4.> C)Analysis of Insegion So!
5.> n
6.> C)Analysis of Insegion So!
7.> n

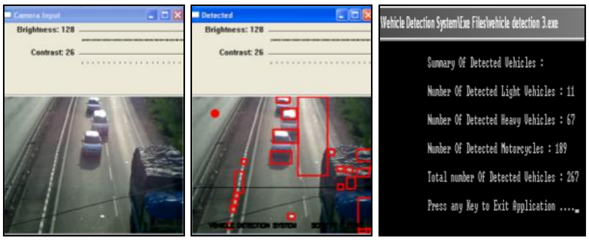
C)Analysis of Algorithms
Example Soging
Example Sorting
C)Analysis of Insegion So!

Fig. 1: Lecture video and its index points

4.2 Moving Vehicle Detection Using Background Subtraction

Vehicles detection must be implemented at different environment where the light and the traffic status are changing. In this system, students accepted the traffic flow video from a camera and converts video into frames. The proposed system consists of three stages.

1) System Initialization: System gets initialized and set up in this stage. Camera records continuous stream of data and sends to the system for analysis.
2) Background Subtraction: In this stage, a set of frames are taken into focus and on successive analysis and operations background subtraction takes place.
3) Vehicle Detection: In this stage, using the subtracted background image all the moving vehicles/objects can be tracked and counted.
System works in two modes, pre-recorded video mode and real-time camera mode. We can provide pre-recorded traffic flow video for detection and counting of vehicles [18]. Real time camera mode application accepts the video from the camera and tracks the vehicles. This system was tested on a laptop powered by an Intel Core Duo (1.83 GHZ) CPU and 2GB RAM equipped with iNTEX IT- 305 WC camera. We tested the system on image sequences of highway scenes. The system is able to track and classify most vehicles successfully. Figure 2 shows results of the system.



Brightness: 128
Contrast: 26

```
Summary Of Detected Vehicles :

Number Of Detected Light Vehicles : 11

Number Of Detected Heavy Vehicles : 67

Number Of Detected Motorcycles : 189

Total number Of Detected Vehicles : 267

Press any Key to Exit Application ....
```

Fig. 2 (a) Real time capturing of traffic (b) Moving vehicle detection (c) Counting of the vehicular movement

4.3 Drowsy Driver Detection and Warning System for Commercial Vehicle Drivers

This project is focused on the localization of the eyes, which involves looking at the entire image of the face, and determining the position of the eyes, by a self developed image-processing algorithm. Once the position of the eyes is located, the system is designed to determine whether the eyes are opened or closed and detect fatigue.

Camera continuously focused on drivers face to generate its video. The recorded video is converted to respective frames at running time. After reading the image (RGB image), it is converted into grayscale image. Edges of the image are detected by applying Sobel filters. Assuming circle matrix values, numbers of center points of circles are calculated. This uses Circular Hough Transform. The possible circles are plotted on original frame. After checking every eight consecutive frames, we can conclude that the eyes are closed if numbers of circles are less than 2 else eyes are opened. Detection of Drowsiness in user/driver is divided into the following categories/steps:

1) Video Capturing: Continuous Monitoring of users/Drivers Face through Video Camera.
2) Framing: Converting input video into its respective frames.
3) Binarization: Converting each frame of RGB into to Grayscale frame for further image processing.
4) Edge Detection: Finding out edges for every grayscale frame which acts as input for Circular Hough Transform.
5) Circular Hough Transform: It is used to locate all possible circles on original Images.
6) Alarm Generation: Once the eyes found closed, an alarm/sound is generated to give an alert to driver.

4.4 Controlling Computer Using Hand Gesture Recognition

Basic purpose of this gesture recognition is to develop system such that we are able to operate computer with the help of hand gestures without using mouse. This project helps us in introducing new computer interface, provided interface will be more natural and user friendly. It also reduces physical contact with devices and minimizes hardware required for interfacing. Screen shot in Figure 3 shows segmented hand, ROI of wrist band along with separated hand from whole background.

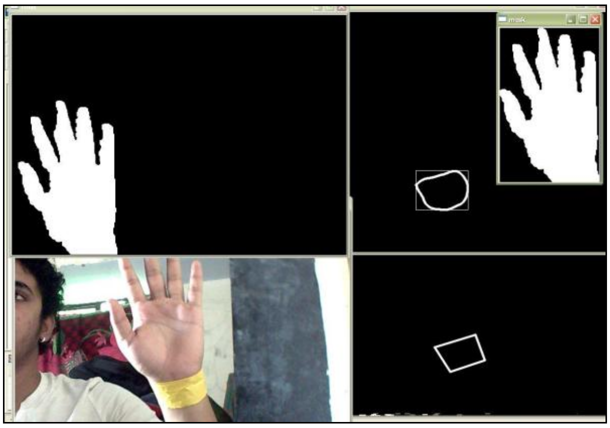
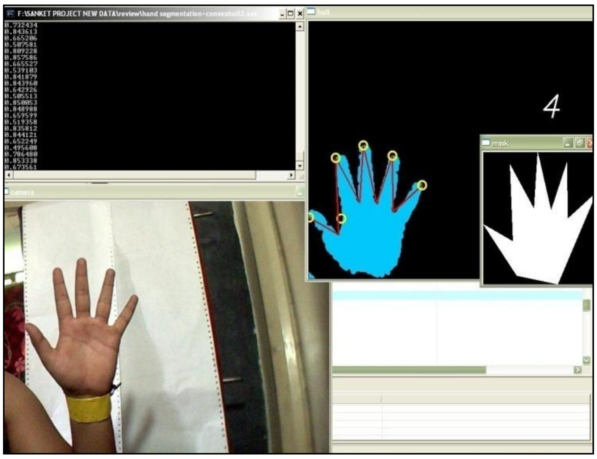


Fig. 3 Detecting hand and wrist band

In this project convex hull and convex defect techniques are used to segment and detect hand correctly. Convex hull and convex defect techniques determine exact portion of hand which required for us to perform operations. If current hand gesture is different from previous then gesture is detected. We have predefined some gestures in the system. Figure 4 describes the way of gestures identification.


F:\SANKET PROJECT NEW DATA\review\hand segmentation-convexhull2.exe

```
0.732434
0.843613
0.665206
0.507581
0.809228
0.857586
0.665527
0.539103
0.841879
0.843960
0.642926
0.505513
0.850053
0.848988
0.659599
0.519358
0.835812
0.844121
0.652249
0.495608
0.786480
0.853338
0.673561
```

Fig. 4 Gesture identification

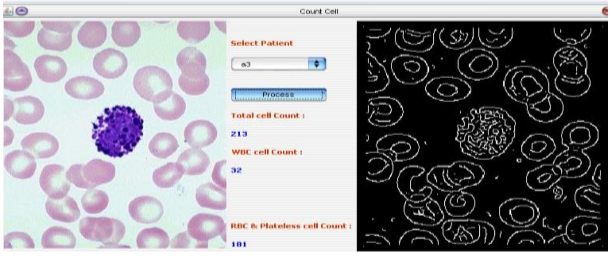
4.5 Digesting Commercial Clips from TV Streams

News, serials, movies, etc. are interrupted by groups of advertisements. While commercial detection is certainly appealing for home users to skip unwanted content, also many professional applications in the domain of video indexing, retrieval, archiving, or summarization require removal of commercial breaks in order to focus on the actual content.

We present a simple approach for commercial digest to perform individual commercial segmentation from given video streams. Firstly, the recorded TV programs pass through the shot change detector. The shot change detector's results are used to identify both cuts and strong cuts in a minute for labeling candidates of commercial segment. These candidates are further analyzed to find the exact commercial boundaries. Finally, commercials can either be removed or kept according to the purposes of applications.

4.6 Blood Cell segmentation and Counting

Today in this busy hectic schedule; pathologists need some help in terms of software for blood cell analysis. Thus, the idea of this project is to serve the pathologists, medical technicians for the same, by using Image Processing techniques. After analyzing the principle nature of PCNN, we use an algorithm for cell segmentation and counting, where the image is de-noised using PCNN model combined with median filter. Then, in order to segment blood cells from its background we adopt the PCNN segmentation algorithm based on minimum cross-entropy. Finally, the cells' number can be counted and a specific cell can be segmented from its background through labeling each blood cell [19]. The user interface developed for the system is shown in Figure 5.


Count Cell

Select Patient
a3

Process

Total cell Count :
213

WBC cell Count :
32

RBC & Plateless cell Count :
181

Fig. 5 Patient sample of blood and its count

4.7 Obstacle Detection and Distance Estimation Using Stereovision

This project deals with finding the distance of an object from a camera, which is missing in the 2D image shot by the camera. The disparity between images from left and right cameras is used to find out the distance of the obstacles from the camera. The image processing part of the project involves image capture, segmentation, feature extraction and matching. Basic aim of this system is to capture stereoscopic images of the obstacle and then evaluate the disparities of corresponding points in left and right images. Using this we can find the distance from the camera at which the obstacle is located. The algorithm we developed detects the distance up to 150 cm with an error of 15%. The algorithm is developed for rectangular objects but can be further developed for irregular and overlapped objects also.

4.8 Real Time Object Detection and Classification for Human

Understanding any activity involves being able to detect and classify targets of interest and analyze what they are doing. In this PG project, a student demonstrates a process for detecting moving targets and extracting boundaries. Main purpose behind the project is to implement robust method to detect human if present in video under dynamic environment. This can be used as efficient visual surveillance method. The software will monitor for movement or motion in a localized area. If motion is detected it will check for object whether it is human or nonhuman. We collected a total of indoor and outdoor video sequences captured from stationary cameras. The method implemented gives very good performance and detects human about 90% of time [20]. The snapshots are given in Figure 6.

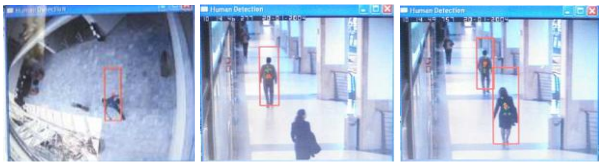


Fig. 6 (a) Human detection for upside camera (b) Human detection from front side camera (c) Multiple human detection

4.9 Real Time Detection of Traffic Signs

Students are working on a system which exploits the visual information already available to the driver and it recognizes traffic signs by analyzing the images/video taken from a camera installed on the car. If an image contains signs, the system gives an output to the driver, indicating the respective sign. This system will help drivers to gain better awareness of the road.

4.10 Abandoned Object Detection System

Intelligent video analysis offers a promising solution to the problem of active surveillance. Automatic threat detection systems can assist security personnel by providing better situational awareness. In this proposed work we investigate how to detect static objects, which are left abandoned in a scene using a moving camera.

5. EVALUATION AND PROBLEMS IN TEACHING-LEARNING

Programme educational objectives of project work to expose students with project-product development cycle using industrial experience, use of state-of-art technologies, which is mate with cultivating this type of application development in the technical educational institutes. OpenCV is becoming widely used by the research community. Since its release as freeware, the OpenCV library has been downloaded over half a million times and the official yahoo group has over 5000 members [4].

The explanations of complicated functions are not well documented even on the group. Beginners finds it’s too difficult to grasp initial concepts and setting up the environment. Students find it easy to go through small example code from the available resources. Appreciation from subordinates and peers enhances their interest of developing small applications in computer vision. It also encourages junior students to take up computer vision applications as their projects.

6. ACKNOWLEDGMENTS

We would like to deeply acknowledge the effort of students Iresh Dhotre, Prashant Dhotre, Vinod Kimbhahune, Manoj Shelar, Gouri Tagunde, Pallavi Bangare, Smita Kakade, Ranveer Puri, Sanket Dhuri for their contribution in effective execution of the projects and setting up the new trends in IT graduates to inculcate vision culture in their projects.

7. CONCLUSION

Use of open source environment from Intel’s allowed most of the students not only to build an interesting solution to computer vision problems but it helps us to bridge the theoretical concepts with practical approach. It helped them to learn the theory behind the complex image processing and computer vision algorithms. In this paper we discuss suitability of the OpenCV libraries for providing effective software solutions that could be programmed and tested for real time computer vision application. This study had cultivated the motivation towards basic research in computer vision problems. This study will culminate in approaching various agencies for research funding and specialized lab development.

8. REFERENCES

[1] Qingcang Yu, Cheng, H.H., Cheng, W.W., Xiaodong Zhou, Interactive open architecture computer vision, 2003 Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence, pp. 406- 410, 2003

[2] G. Bradski, A. Kaehler, 2008, Learning OpenCV-Computer Vision with the OpenCV Library, O'Reilly

[3] Changsong Shen, Fels, S.S., Little, J.J., OpenVL: Towards A Novel Software Architecture for Computer Vision, Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on , vol., no., pp.1-8, 17-22 June 2007

[4] Chaczko, Z., Braun, R., Teaching Computer Vision for telemedicine systems using OpenCV, Information Technology Based Higher Education and Training (ITHET), 2010 9th International Conference on , vol., no., pp.17-20, April 29 2010-May 1 2010

[5] Delft Scientific Image Processing Library. http://www.diplib.org/diplib

[6] cvBlob. ttp://opencv.willowgarage.com/wiki/cvBlobsLib

[7] VXL. http://vxl.sourceforge.net/ Vision-something-Libraries

[8] VLFeat. http://www.vlfeat.org/

[9] SimpleCV. http://simplecv.org/

[10] VIGRA library website [Online].Available:http://kogs-www.informatik.uni-hamburg.de/~koethe/vigra.

[11] Jara, C.A., Candelas, F.A., Gil, P., Torres, F., Esquembre, F., Dormido, S, “Ejs+EjsRL: An interactive tool for industrial robots simulation, Computer Vision and remote operation”, Robotics and Autonomous Systems. Vol. 59 (6). Pp. 389-401. 2011.

[12] National Instrument Company. http://www.ni.com/

[13] Microsoft Research Center http://research.microsoft.com/projects/VisSDK/.

[14] Aurora Company www.hallogram.com/leadtool/imgsrv/

[15] Cavallaro, A, Image analysis and computer vision for undergraduates," Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on , vol.5, no., pp. v/577- v/580 Vol. 5, 18-23 March 2005

[16] Robert Laganière, OpenCV 2 Computer Vision Application Programming Cookbook, Packt Publishing-2011

[17] MD Shelar, NJ Uke, Video Segmentation and Video Indexing For E-Learning, Proceedings of International Conference on Communication and Electronics Information (ICCEI), 2012

[18] Nilesh J. Uke, Ravindra C. Thool, Moving Vehicle Detection for Measuring Traffic Count Using OpenCV ,Proceedings of 4th International Conference on Conference on Electronics Computer Technology (ICECT), 2012 – Paper accepted

[19] VV Kimbahune, NJ Uke, Blood Cell Image Segmentation And Counting, International Journal of Engineering Science and Technology (IJEST), Vol. 3 No. 3, PP. 2448, 2011

[20] S. Kakade, NJ Uke, Real Time human detection using covariance matrices as human descriptor , Proceedings of 4th International Conference on Computer and Automation Engineering (ICCAE), 2012

9. AUTHORS PROFILE

Nilesh J. Uke received the BE degree in computer science and engineering from Amravati University, Maharashtra, India, in 1995 and the ME in computer Engineering in 2005 from Bharathi University Pune. He is Assistant Professor with the department of Information Technology, Sinhgad College of Engineering Pune. Currently he is pursuing his PhD in Computer Vision from SRTM University, Nanded, India. His research area includes Computer Vision, Human Computer Interface and Artificial Intelligence. He is member of IEEE, ACM, Life member of ISTE and Indian Science Congress.

Dr. Ravindra C. Thool received his BE, ME and Ph.D. in Electronics from SGGS College of Engineering & Technology, Nanded, India, in 1986, 1991 and 2003 respectively. He is professor and head with Information Technology in the same organization. His research area includes Computer Vision, Image processing and multimedia information systems. He has published several research papers in refereed journals and professional conference proceedings. He is member of IEEE, Life member of ISTE.