RESEARCH ARTICLE OPEN ACCESS

Image Detection and Count Using Open Computer Vision (Opencv)

K. B. Neelima[1], Dr. T. Saravanan[2]
[1]Research Scholar, Assistant Professor.
[2]Professor & HOD, Bharath University, Chennai.

ABSTRACT
The purpose of this paper is to introduce and quickly make a reader to provide basics of OpenCV (Open Source Computer Vision) without having to go through the lengthy reference manuals and books. OpenCV is actually an open source library for image and video analysis, originally introduced more than decade ago by Intel. The latest major change took place in 2009 (OpenCV2) which includes main changes to the C++ interface. Nowadays the library has >2500 optimized algorithms. It is extensively used around the world, having >2.5M downloads and >40K people in the user group. Regardless of whether one is a novice C++ programmer or a professional software developer, unaware of OpenCV, the content should be interesting mainly for the researchers and graduate students in image processing and computer vision areas.

***Keywords**:* OpenCV, image processing, computer vision, calibration, face detection.

I. INTRODUCTION

OpenCV means Intel Open Source Computer Vision Library. It is a collection of C functions and a few C++ classesuses to implement the Image Processing and Computer Vision algorithms. The Key features about it are Cross-Platform API of C functions FREE for commercial and non-commercial uses. This means the user can take advantage of high speed implementations of functions commonly used in Computer Vision/Image Processing. OpenCV was designed for computational efficiency and with a strong focus on real time applications.

OpenCV can be written in optimized C and takes the advantage of multicore processors. OpenCV automatically uses the appropriate integrated Performance Primitives (IPP) library at runtime. One of OpenCV's goals is to provide a simple infrastructure of computer vision to help people for build the fairly sophisticated vision applications quickly. The OpenCV library contains over 500 functions that span many areas in vision, including factory product inspection, medical imaging, security, user interface, camera calibration, stereo vision, and robotics.

II. DOWNLOADING AND INSTALLING OPENCV

The OpenCV site is on the Source Forge at http://SourceForge.net/projects/opencvlibrary and the OpenCV Wiki [OpenCV Wiki] page is at http://opencvlibrary.SourceForge.net. For Linux, the source distribution is the file opencv-1.0.0.tar.gz; for Windows, you want OpenCV 1.0.exe. However, the most up to date version is always on the CVS server at Source Forge. Once installed anyone can download the libraries and then install them. In the text file named INSTALL directly under the .../opencv/ directory, the installation instruction on Linux or Mac OS are availed and also describes how to build and run the OpenCV testing routines. INSTALL lists the additional programs you'll need in order to become an OpenCV developer, such as automake, autoconf, libtool, and swig. Windows-Get the executable installation from Source Forge and run it. It will install OpenCV, register Direct Show filters, and perform various post-installation procedures. The system is ready to start OpenCV. You can always go to the .../opencv/_make directory and openopencv.sln with MSVC++ or MSVC.NET 2005, or open opencv.dsw with lower versions of MSVC++ and build debug versions or rebuild release versions of the library. To add the commercial IPP performance optimizations to Windows, obtain and install IPP from the Intel site (http://www.intel.com/software/products/ipp/index.htm);use version 5.1 or later. Make sure the appropriate binary folder (e.g., c:/program files/intel/ipp/5.1/ia32/bin) is in the system path.

III. OPENCV STRUCTURE AND CONTENT

OpenCV is broadly structured into five main components, four of which are shown in Figure 1.0. The CV component contains the basic image processing and higher-level computer vision algorithms; ML is the machine learning library, which includes many statistical classifiers and clustering tools. For storing and loading video and images, the HighGUI contains I/O routines and functions and CXCore contains the basic data structures and content.

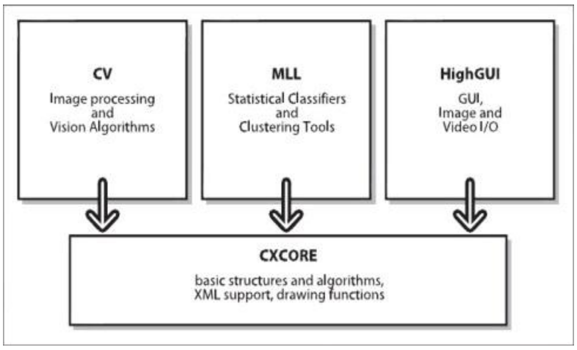
CV
Image processing and Vision Algorithms
MLL
Statistical Classifiers and Clustering Tools
HighGUI
GUI, Image and Video I/O
CXCORE
basic structures and algorithms, XML support, drawing functions

Figure 1.0

Figure 1.0 does not include CvAux, which contains both defunct areas (embedded HMM face recognition) and experimental algorithms (background/foreground segmentation). CvAux is not well documented in the Wiki and is not documented at all in the .../opencv/docs subdirectory.

IV. HIGH GUI

The OpenCV function that allows the user to interact with the operating system, the file system, and hardware such as cameras are collected into a library called HighGUI ("high-level graphical user interface"). HighGUI allows the user to open windows, to display images, to read and write graphics-related files (both images and video), and to handle simple mouse, pointer, and keyboard events. The HighGUI library in OpenCV can be divided into three parts: the hardware part, the file system part, and the GUI part. The hardware part is primarily concerned with the operation of cameras. In most operating systems, interaction with a camera is a tedious and painful task. HighGUI allows an easy way to query a camera and retrieve the latest image from the camera.

V. IMAGE PROCESSING

"Image processing", mean that just: using higher-level operators that are defined on image structures in order to accomplish tasks whose meaning is naturally defined in the context of graphical, visual images.

A. SMOOTHING-Smoothing, also called blurring is a simple and frequently used image processing operation. There are many reasons for smoothing, but usually it is done to reducecamera artifacts and noise. Smoothing is also important when there is importance to reduce the resolution of an image in a principled way. OpenCV offers five different smoothing operations at this time. All of them are supported through one function, cvSmooth(),* which takes user's desired form of smoothing as an argument.

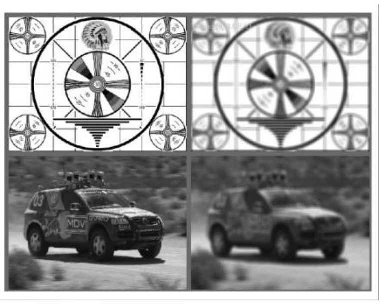

Figure 2.0 Image smoothing by block averaging: on the left are the input images; on the right, the output images.

The simple blur operation, as exemplified by CV_BLUR in Figure 2-1, is the simplest case. Each output pixelis the simple mean of all the pixels in a window around the corresponding input pixel. Simple blur supports 1–4 image channels and works on 8-bit images or 32-bit floating-point images.

B. IMAGE MORPHOLOGY--OpenCV provides a fast, convenient interface for doing morphological transformations on image. Dilation and erosion are the basic Morphological transformationsand they arise in a wide variety of contexts such as removing noise, isolating individual elements, and joining disparate elements in an image.

C. RESIZE--The user may often encounter an image of some size that is to be converted to an image of someother size and may want to upsize (zoom in) or downsize (zoom out) the image; this can be accomplished by using the either task cvResize(). This function will fit the source image exactly to destination image size.

VI. IMAGE TRANSFORMATION

Image transforms, are methods for changing an image into an alternate representation of the data entirely. The original image is converted to an alternate representation of the data. The result of this operation is stored in an OpenCV "image" structure, but the individual "pixels" in this new image represents input spectral components. OpenCV provides complete implementations of some of the more common ones as well as building blocks to help the user implement own image transforms.

VII. TRACKING AND MOTION

Identification amounts to finding the object of interest from one frame in a subsequent frame of the video stream.Techniques for tracking unidentified objects typically involve tracking visually significant key points (more on what constitutes "significance"), rather than extended objects. OpenCV provides two methods for achieving this: the Lucas-Kanade* [Lucas81] and Horn-Schunck [Horn81] techniques, which represent what are often referred to as sparse or dense optical flow respectively.

VIII. CAMERA CALIBRATION

The actual calibration is done via cvCalibrateCamera2(). In this routine, the method of calibration is to target the camera on a known structure that has many individual and identifiable points. By viewing this structure from different angles, it is possible to compute the (relative) location and orientation of the cameraas well as the intrinsic parameters of the camera. In order to provide multiple views, rotate and translate the object.

IX. MACHINE LEARNING

The goal of machine learning (ML) is to turn data into information. After learning from a collection of data, a machine to be able to answer questions about the data: What other data is most similar to this data? Is there a car in the image? What ad will the user respond to? Often there is cost component, so this question could become: “Of the products that the user makes the most money from, which one will the user most likely buy if we show them an ad for it?” Machine learning turns data into information by extracting rules or patterns from that data.

X. CONCLUSIONS

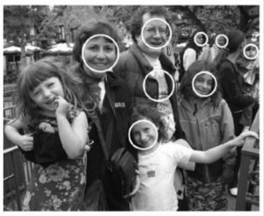

Figure 3.0. Face detection on a park scene: some tilted faces are not detected, for the image shown; more than a million sites and scales were searched to achieve this result in about 1.5 seconds on a 2 GHz machineThe Figure 3.0 shows the detected faces in the shown photo. This detection on a given photo takes place in nano seconds and the output is shown in just about 1.5 seconds on a machine having a speed of 2 GHz. OpenCV has long received support from Intel Corporation and Willow Garage (www.willowgarage.com).One of the key new development areas for OpenCV is robotic perception. Computer vision has a rich future ahead; OpenCV seems likely to be (at least in part) one of the key enabling technologies for computer vision. Endless opportunities for creativity and profound contribution lie ahead.

REFERENCES

[1] Intel corporation, “*Open Source Computer Vision Library – [Reference Manual]* Copyright © 1999-2001 Intel Corporation.

[2] Gary Bradski and Adrian Kaehler, O’REILLY Learning OpenCV Copyright © 2008 Printed in the United States of America.

[3] *[OpenCV Wiki] Open Source Computer Vision Library Wiki*, http://opencvlibrary.sourceforge.net/.

[4] *[OpenCV] Open Source Computer Vision Library (OpenCV)*, http://sourceforge.net/projects/opencvlibrary/.

[5] D. Chen and G. Zhang, “*A new sub-pixel detector for x-corners in cameracalibration targets*,” WSCG Short Papers (2005).

[6] T. A. Clarke and J. G. Fryer, “*The Development of Camera Calibration Methods and Models,*” Photogrammetric Record 16 (1998).

[7] A. Colombari, A. Fusiello, and V. Murino, “*Video objects segmentation by robust background modeling*,” International Conference on Image Analysis and Processing, September 2007.

[8] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski, “*Selfsupervised monocular road detection in desert terrain,*” Robotics: Science and Systems, Philadelphia, 2006.