Feature Extraction of an Image by using Edge Detection

Priyanka S. Barapatre
ME Student
Dept. of Electronic and Telecommunication
MGICOET, Shegaon
Shegaon, India
priyanka212patre@gmail.com

Dr. C. M. Jadhao
Principal and Professor
Dept. of Electronic and Telecommunication
MGICOET, Shegaon
Shegaon, India
cmjadhao@gmail.com

Prof. Amit S. Kakad
Assistant Professor
Dept. of Electronic and Telecommunication
MGICOET, Shegaon
Shegaon, India
ameetkakad@gmail.com

Abstract—This paper give the introduction of OpenCV. The modules of OpenCV likecv, cvaux,cvcore andhighguiare described.OpenCV is used for detecting the edges of image which isacquiring through camera.For processing on color image which iscapture from camera, there are various operations perform on it. The various operations like capturing the image from camera, then perform haartransform, then converts image color to gray scale, then smoothing operation is performed on image. After all the operationperformed on image, we will detect the edges of the capture image.

Keywords*-OpenCV, Haar cascade, smoothing, etc.,*

*****

I. INTRODUCTION TO OPENCV

What is OpenCV? This might be the 'basic' question that comes first to everyone’s mind. Well, OpenCV is nothing but an open source computer vision library, which is written in C and C++ languages.OpenCV library contains abundant advanced math functions, computer vision,image processing functions, and functions that span many areas in vision.

Open source library has C, C++, and Python interfaces running on Windows, Ubuntu, Mac, Linux and Android operating systems. Before, the use of OpenCV; let’s get familiar with the same. As, by now you must have understood that it is a 'library', so to use the functions available in the library you would need to compiler. For startingthe OpenCV, the systemneed to install 'OpenCV' and a 'compiler' and then establish a linkage between the twoie the openCV and the compiler. The Compiler is able to use the functions available with the library.

There are OpenCV modules, they are as follows:[1]

- *cv* - Main OpenCV function.
- *cvaux* - Auxiliary or experimentalOpenCV functions.
- *cxcore* - Data structures and linear algebra support.
- *highgui* - GUI functions.

OpenCV has a modular structure, which means that the package includes several shared or static libraries. The modules which are available are as follows:

a) **Core** –The basic data structures are defining by a compact module, which can include the dense multi-dimensional array Mat and basic functions in core used by all other modules.
b) **imgproc** - An image processing module that includes following points like linear and non-linear image filtering, the image transformations which is geometrical, color space conversion then histograms, and so on.
c) **Video** - a video analysis module that includes motion estimation, background subtraction, and object tracking algorithms.
d) **calib3d** –The calib3d is the single and stereo camera calibration and it is the basic multiple-view geometry algorithms, and alsoelements of 3D reconstruction, object pose estimation, stereo correspondence algorithms.
e) **features2d** - salient feature detectors, descriptors, and descriptor matchers.
f) **objdetect** –It is nothing but the object detection and instances of the predefined classes like faces, eyes, people, cars, and so on.
g) **highgui** - an easy-to-use interface to simple UI capabilities.
h) **videoio** - an easy-to-use interface to video capturing and video codecs.
i) **gpu** - graphics processing unit(GPU) –is accelerated algorithms from different OpenCV modules.

Video Input

What do you meant by video? A video is basically a collection of continuous image or a frame displayed at a certain rate (generally 30 frames per second). To extract the frames from video first we need to attach this video to the input stream and then extract those frames, when it is required. To connect or couple the video to input stream we use the following format

```
CvCapture* capture=cvCreateFileCapture("file_name.avi" );
```

And for extracting frames from input video use the following format:

```
Ipl Image* input_frame=cvQueryFrame(capture);
```

Camera Input

First camera needs to be initialized and then image is captured and further operations can be carried out on that image. Use the following command for initiating the camera:
CvCapture *capture=cvCreateCameraCapture(0);
0 in CreateCameraCapturestands for default webcam and to access other camera connected to the computer use 1 as argument, for starting the camera, it takes sufficient amount of time, so make sure that sufficient time has passed before we capture the image. This can be achieved through for

(inti=0;i<100000000&& capture!=NULL ;i++);

Finally image is captured and stored in variable of type
IplImage* frame=cvQueryFrame (capture);

Video input through camera

This is video input captured from camera. It is similar to video input which will you need to attach the video capturedfrom camera to the input stream. Following functionof video input helps us to do the function:
CvCapture *capture=cvCreateCameraCapture (0);
In this case, there is no need to initialize the camera because frame is captured regularly by the camera. Again, 0 is used for default webcam and use 1 for input captured from external camera.

Haar Cascades

For detection of particular object, by using Haar feature-based cascade classifier is an effective object detection method. Haar like features are digital image features used in object recognition system. Haar Cascades are trained classifiers used for detecting features like face, eyes, nose, lips, upper body etc. The data folder in OpenCV can stored these cascades, which are used for execution of the code. Firstly, it is needed to load cascade and then use the cascade to detect the presence of the corresponding feature. OpenCV comes with a trainer as well as detector. OpenCV can be used for creating our own classifier, if you want to train your own classifier for any object like car, planes etc.OpenCV already contains many pre-trained classifiers for face, eyes, smile etc. Those XML files are stored in opencv/data/haarcascades/ folder.
Following figure shows logo of OpenCV.[1]

Figure 1: OpenCV Logo

Consequently, the opencv library can operate on limited fixed set of primitive data types. That is, array elements should have one of the following types:

- 8-bit unsigned integer (uchar)
- 8-bit signed integer (schar)
- 16-bit unsigned integer (ushort)
- 16-bit signed integer (short)
- 32-bit signed integer (int)
- 32-bit floating-point number (float)
- 64-bit floating-point number (double)

Color to Gray

cvtColor

In Color to Gray image, the color to gray function which can converts a capture image from one color space model to another model. The first byte in a standard (24-bit) color image will be an 8-bit Blue component element, the second byte will be Green, and the third byte will be Red. In case of a transformation which is from RGB (Red, Green, Blue) color space, the order of the channels should be specified explicitly (RGB or BGR). Here, the default color format in OpenCV is often referred to as RGB but inn actual it is BGR. The fourth, fifth, and sixth bytes would then be the second pixel of image (Blue, then Green, then Red), and so on.

Following are the conventional ranges for R, G, and B channel values:

- 0 to 255 for CV_8U images
- 0 to 65535 for CV_16U images
- 0 to 1 for CV_32F images

Smoothing Image

In this paper you will learn how to apply diverse linear filters to smooth images using OpenCV functions such as:

- Blur
- Gaussian Blur
- MedianBlur
- BilateralFilter

Blur

blur (src, dst, Size (i, i), Point (-1,-1));

We specify 4 arguments for the input image. Above line shows the format of writing blur function. Here is the explanation of each word in the blur function.

src: Source image
dst: Destination image
Size (w,h): Defines the size of the kernel to be used (of width *w* pixels and height *h* pixels)
Point (-1, -1): Indicates where the anchor point (the pixel evaluated) is located with respect to the neighborhood. If there is a negative value, then the center of the kernel is considered the anchor point.

Gaussian Blur

GaussianBlur(src, dst, Size(i, i), 0, 0);

Here in Gaussian Blur we use 4 arguments are as follows. Above line shows the format of Gaussian blurs.

src: Source image
dst: Destination image
Size(w, h): The size of the kernel to be used (the neighbors to be considered). 'w' and 'h' have to be odd and positive numbers otherwise the size will be calculated using the σ_x and σ_y arguments.
σ_x : The standard deviation in x. Writing '0' implies that σ_x is calculated using kernel size.
σ_y: The standard deviation in y. Writing '0' implies that σ_y is calculated using kernel size.

Median Filter

medianBlur (src, dst, i);

We use three arguments in median filter. The above line shows the format of writing median filter.Where,

src: Source image
dst: Destination image, must be the same type as
srci: Size of the kernel .

BilateralFilter

bilateralFilter (src, dst, i, i*2, i/2);

Here we use five arguments in bilateral filter. Above line shows the format of the bilateral filter.

src: Source image
dst: Destination image
d: The diameter of each pixel neighborhood.
σ_{Color}: Standard deviation in the color space.
σ_{Space}: Standard deviation in the coordinate space (in pixel terms)

II. MORPHLOGICAL PROCESSING

Digital image processing involves the manipulation and interpretation of digital images with the aid of a computer and it is an extremely broad subject and it often involves procedures, which can be mathematically complex [6].Morphological image processing is a group of non-linear operations. Morphology is related to the shape of features in an image.Based on the image shape some simple operations are used. It is normally performed on binary images.There is a need of two inputs one original image, and structuring element, which decides the nature of operation. In morphology there are two operators are used which are Erosion and Dilation. Then it also has some forms like Opening, Closing and Gradient etc.

1. Erosion

Erosion is one of the fundamental operations in morphological image processing. It is applied to binary images. The erosion takes two data inputs. First is the image is to be eroded and second is structuring element. Erosion removes pixels on object boundaries. Erosion is used for shrinking of element A by using element B. Shrink object in binary image.

2. Dilation

Dilation adds pixel to the boundaries of objects in an image. Dilation is an operation that grows object in a binary image.Dilation operator can be applied to binary and grey scaleimages[5]. Dilation is used for expanding an element A by using structuring element B. It needs two input data one is dilated and second is structuring element.It increases the brightness of the object[5].

3. Opening and Closing

Opening generally smoothest the contour object and closing also tends tosmooth sections of contours but, it is opposed to opening [7]. Opening and closing are two important operators used in morphology. They are normally applied for binary images. Both the operators are derived from erosion and dilation.

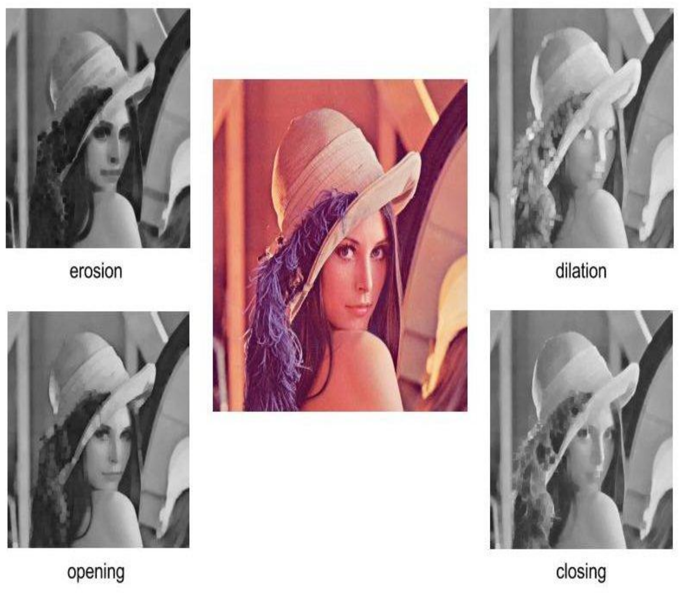

Figure 2: Different Morphological Operations

III. EDGE DETECTION

Definition of edges –

Edge detection techniques used for finding the boundaries of objects**.**Edge detection is used for image segmentation and data extraction**.**Edge detection is an important image processing technique with wide range of applications and one of the main applications of edge detection techniques is in the process of image segmentation and object detection [12]. For edge detection some operator are used like Prewitt, Sobel, Canny and Roberts etc. An edge may be regarded as boundary between two dissimilar regions in an image [9]. The operation and function of Prewitt edge detector is almost same as of Sobel detector technique but Prewitt having different kernels [10].Sobel technique extracts all of edges in an image, regardless of its direction [8]. Canny operator is the operator which is used to find edges of the image by isolating noise and the features of the edges in the image cannot be affected and it is a good technique for extracts the features in an image without disturbing its features [8]. The Roberts cross operation

ISSN: 2321-8169

performs a simple and quick 2-D spatial gradient measurement on an image [11]. Following figure shows, original image for edge detection. And by applying various operators on original image results are as follows:

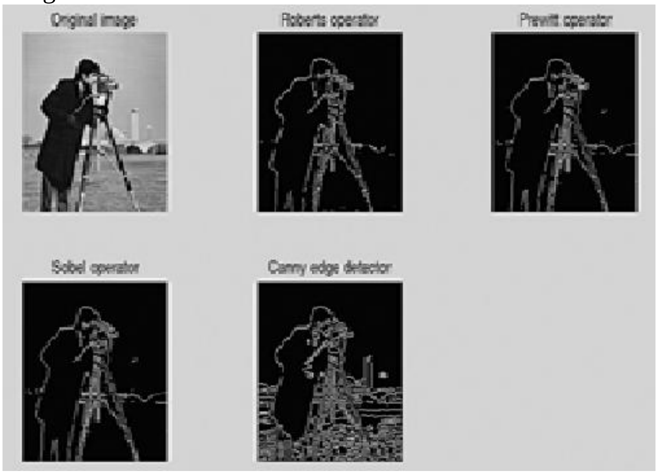

Figure 3: Edge detection

Detecting edges is very useful in a typical image understanding task such as object identification.The first step in image segmentation is edge detection.For example, the development of a low bit rate image coding system. We know that the image consists of only edges are highly intelligible.

Goal of edge detection –The main goal of edge detection is extract the various features or important features from the edges of an image like line, curves etc. These features of edges of imagesare used by higher-level computer vision algorithms.

The steps of edge detection are as:
There are four steps in edge detection like:

1. Smoothing
2. Enhancement
3. Detection
4. Localization

In smoothing we suppress noise as much as possible, without destroying the true edges. In enhancement we use a filter to enhance the quality of the edges in the image. The detection determines that which edge pixels should be discarded as noise. The localization is used for determine the exact location of an edge.
For edge detection using USB camera in OpenCV "VideoCapturecap(0)" this command is used for programming.For checking the result in OpenCV "if (!cap.isOpened())" this command is usedwhich shows that we are succeeded or not.For getting new frame "cap>>frame" this commandisused.

IV. Conclusion

Avery basic format of smoothing image function is described which is used in OpenCV. Here morphological operation is also described for the detection of edgesin OpenCV.Also edge detection can be performed in OpenCV by capturing the live images from cameraand edge detection is perform on it. By using the various operations on the image which is acquiring by camera, the edges of captured image can be detected and displayed in OpenCV.

References

[1] Nidhi, "Image Processing and Object Detection", International Journal of Applied Research 2015; 1(9): 396-399, ISSN Print: 2394-7500, ISSN Online: 2394-5869, IJAR 2015; 1(9): 396-399, www.allresearchjournal.com

[2] Guobo Xie and Wen Lu, "Image Edge Detection Based On Opencv", International Journal of Electronics and Electrical Engineering Vol. 1, No. 2, June 2013, doi: 10.12720/ijeee.1.2.104-106.

[3] Krunal Shantilal Patel, Prof. Kalpesh R Jadav, "Implementation of Embedded ARM9 Platform using Qt And openCV For Human Upper Body Detection", IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 9, Issue 2, Ver. II (Mar - Apr. 2014), PP 73-79

[4] Google Search, Opencv Library

[5] Sakshi Arora, Rahul Pandey, "Applications of Morphological Operators using Image Morphological Algorithms", SSRG International Journal of Electronics and Communication Engineering (SSRG-IJECE) – Volume 3 Issue 8 – August 2016, ISSN: 2348 – 8549

[6] Showkat Ahmad Dar, Sajjad Ahmad Lone, "AN APPLICATION OF MORPHOLOGICAL IMAGE PROCESSING TO FORENSICS", International Journal of Computer Engineering & Technology (IJCET) Volume 6, Issue 8, Aug 2015, pp. 31-40, Article ID: IJCET_06_08_005, ISSN Print: 0976-6367 and ISSN Online: 0976–6375

[7] Megha Goyal, "Morphological Image Processing", International Journal of Computer Sci ence & Technology, ISSN : 0976-8491(Online) | ISSN : 2229-4333(Print), IJCST Vol. 2, Iss ue 4, Oct . - Dec. 2011

[8] Jaspreet Kaur, Anand Sharma, "Review Paper on Edge Detection Techniques in Digital Image Processing", Conference Proceeding of 7th International Conference on Innovative Research in Engineering Science and Management (ICIRESM-16) at The Institutions of Electronic and Telecommunication Engineers (IETE), Lodhi Road, New Delhi, Delhi, India on 13th November 2016

[9] Sonam Saluja, Aradhana Kumari Singh, Sonu Agrawal, "A Study of Edge-Detection Methods", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 1, January 2013, ISSN (Print) : 2319-5940, ISSN (Online) : 2278-1021

[10] Rashmi, Mukesh Kumar, and Rohini Saxena, "ALGORITHM AND TECHNIQUE ON VARIOUS EDGE DETECTION: A SURVEY", Signal & Image Processing : An International Journal (SIPIJ) Vol.4, No.3, June 2013

[11] Indrajeet Kumar, Jyoti Rawat, Dr. H.S. Bhadauria, "A CONVENTIONAL STUDY OF EDGE DETECTION TECHNIQUE IN DIGITAL IMAGE PROCESSING", International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 3, Issue. 4, April 2014, ISSN 2320–088X

[12] Dr.S.Vijayarani, Mrs.M.Vinupriya, "Performance Analysis of Canny and Sobel Edge Detection Algorithms in Image Mining", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 1, Issue 8, October 2013, ISSN(Online): 2320-9801 ISSN (Print): 2320-9798.