

Android Application Based on Image Detection, Recognition and Audio Association

¹Aarti Bawche, ²Abhijeet Chavan, ³Priya Dhivar

^{1,2,3}MAEER'S MIT College of Engineering, Kothrud, Pune, Maharashtra, India

Abstract

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class in digital images and videos. An attempt is made in this study to detect and extract objects in an image implemented on android platform. Color images are converted to binary images using the thresholding technique. Morphological opening and closing filters are used in sequence for object detection. Contour based learning techniques are adopted for drawing contours of the objects detected. In the process objects are extracted and stored in an array for further analysis.

Keywords

Image Processing, Object Detection, Recognition, Smartphone, Android.

I. Introduction

Object detection technology is used for the object tracking & recognition of the object moving direction. The detection of object is a image on the android platform. The object is one detected & save in the mobile phone can be used for further use [1]. The information is associated with the image is tagged to the detected object thus user can get the information as & when required. An attempt is made in this study to detect and extract objects in an image implemented on android platform. Color images are converted to binary images using the thresholding technique [2]. Morphological opening and closing filters are used in sequence for object detection. Contour based learning techniques are adopted for drawing contours of the objects detected. In the process objects are extracted and stored in an array for further analysis [3]. Compared to PC platforms, smart phone platforms have limited resources, so computation-intensive SIFT and SURF descriptors are less usable in such resource-limited environments [6]. Opencv functions are used to implement these algorithms on android.

II. Edge Detection

Edge detection is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The Sobel operator is used for edge detection [1]. We use a 3X3 Sobel matrix which is convoluted with our source image [1]. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The same problem of finding discontinuities in 1D signal is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

III. Grey Scale

In photography and computing, a grayscale or greyscale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information [3]. Images of this sort, also known as black and white, are composed exclusively

of shades of gray, varying from black at the weakest intensity to white at the strongest. Grayscale images are distinct from one-bit bi-tonal black and white images, which in the context of computer imaging are images with only the two colors, black, and white (also called bilevel or binary images).



Fig. 1:

Grayscale images have many shades of gray in between [1]. Grayscale images are often the result of measuring the intensity of light at each pixel in a single band of the electromagnetic spectrum (e.g. infrared, visible light, ultraviolet, etc.), and in such cases they are monochromatic proper when only a given frequency is captured. But also they can be synthesized from a full color image; See the section about converting to grayscale.

IV. Thresholding

The simplest Thresholding methods replace each pixel in an image with a black pixel if the image intensity is less than some fixed constant T (that is, $I_{ij} < T$), or a white pixel if the image intensity is greater than that constant [1]. In the example image on the right, this results in the dark tree becoming completely black, and the white snow becoming complete white.

V. Hough Line Transformation

Hough line transform is used to detect line in edge detected image [4].

Theory: Consider three data points, shown here as black dots.

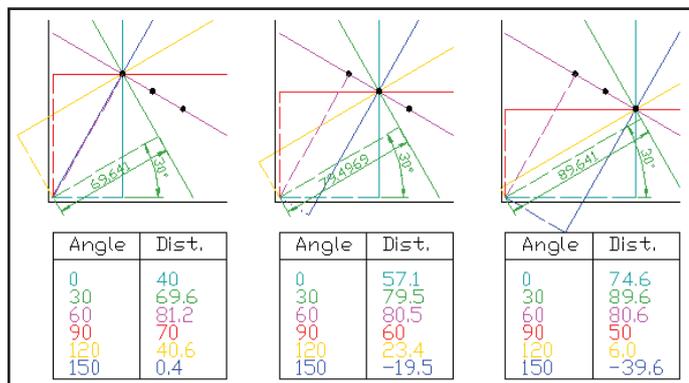


Fig. 2:

- For each data point, a number of lines are plotted going through it, all at different angles. These are shown here as solid lines.
- For each solid line a line is plotted which is perpendicular to it and which intersects the origin. These are shown as dashed lines.
- The length (i.e. perpendicular distance to the origin) and angle of each dashed line is measured. In the diagram above, the results are shown in tables.
- This is repeated for each data point.
- A graph of the line lengths for each angle, known as a Hough space graph, is then created.

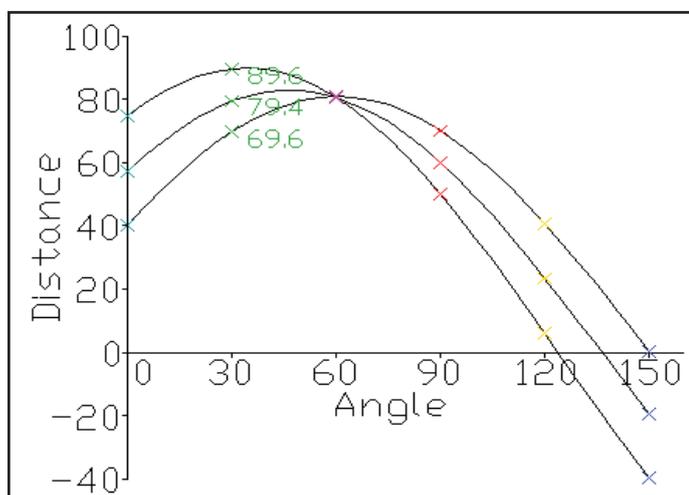


Fig. 3:

The point where the curves intersect gives a distance and angle [5]. This distance and angle indicate the line which intersects the points being tested. In the graph shown the lines intersect at the pink point; this corresponds to the solid pink line in the diagrams above, which passes through all three points [4].

The following is a different example showing the results of a Hough transform on a raster image containing two thick lines.

Modification

we needed to detected larger lines of pixels greater than 50 pixels. So, we applied Hough line transform for first 10 pixels from any reference point to find whether those pixels are white (Edge

detected image is binary image). If these pixels are white, then find directly 50th pixel on same line slope. If that it also white then save this line in a data variable.

Loop through all pixels in the image, and save all lines. Now, find the largest line, second largest and third largest line amongst all lines.

Generally in a analog wrist watch, largest line will be of second hand, second largest of minute hand, and third largest of hour hand.

Our project is focused on detecting wrist watch from image obtained from android camera. For this I used Visual Studio 2010 and opencv 2.3.1. used “video capture” function of opencv to obtain image from the android camera. Then applied deblurring algorithm on the image which is available in opencv [5]. There are several well known methods of removing blur from images. The chief among them are the Richardson-Lucy method, the Weiner filter, Radon transform or some form Bayesian filtering etc. All of these methods perform the converse of convolution known as de-convolution. Almost all approaches are based on the following equation

$$b(x) = k(x) * i(x) + n(x) \tag{1}$$

Where $b(x)$ – is the blurred image, $i(x)$ is the latent (good) image, $k(x)$ is the blur kernel also known as the Point Spread Function (PSF) and $n(x)$ is random noise.

There are two types of de-convolution blind and non-blind. In the non-blind de-convolution method an initial blur kernel is assumed and is used to de-blur the image. In the blind convolution an initial estimate of the blur kernel is made and the latent image is successively iterated to obtain the best image [4]. This is based on a method known as maximum-a-posteriori (MAP) to iterate through successive estimations of better and better blur kernels. The Richardson-Lucy algorithm also tries to estimate the blur kernel from the blurred image [5]. We performed de-convolution using an estimated blur kernel on the image.

Process: steps: DFT=Discrete Fourier Transform

1. Perform DFT on blurred image
2. Perform DFT on blur kernel.

We used opencv function for DFT.

Now, we have deblurred image.

This image is sent for character recognition. (Theory of opencv and character recognition)

We have stored images of digits of 1 to 12 and from 0 to 60. The characters detected will be compared with these stored images and tell time. We have stored “.mp3” files of audio of 1 to 60 numbers. This file is played after the characters (digits) are recognized.

If the character recognition fails, then it means we have analog wrist watch.

This image is sent for probabilistic hough transformation. This algorithm detects line for set of points from the image. (Theory of probabilistic Hough transformation)

Now all the houghlines are kept inside a data variable “cv::Vec4i”.this variable stores the endpoints of line segment as (x0,y0,x1,y1)==(line[0],line[1],line[2],line[3]) so we have array of Vec4i storing all such lines.

For finding time on wrist watch we are interested in area near centre of dial. So it's necessary to filter out the lines which lies beyond certain radii from center. This minimum radii is dynamic depending on the size of image. The near lines are kept inside array of `cv::Vec4i`. The numbers of near lines are kept in a counter int variable.

Now, the near lines can be repetitive as some overlapping lines are also detected as different lines. To filter this out, we found out that "no two lines should have angle less than 5 degree". This angle is based on experimentations. If the angle is less than 5 degree then select only one line out of other lines. Thus, this forms our distinct lines. The number of distinct lines is kept in another counter int variable.

Now, find the intersection of pair of distinct lines. The intersection point and the farthest endpoint of each line is taken into consideration for calculating time. Sum of lengths of pair of lines is stored in a variable. The minimum sum is selected. The sum referring to pair of lines is of minute hand and hour hand. Depending on the individual lengths of lines, minute and hour hand is found out. Actual time is obtained considering the angles between hour hand line and minute hand line. As previously, described audio file is associated to tell the time according to time detected.

VI. Scope

1. Find angle between intersecting lines obtained through hough line transformation on edge detected image.
2. This angle help to tell time on analog wrist watch if the watch is approximately vertical.
3. We have database of known objects which have specific dimensions. The line detected help to detect these known objects inside any image.
4. Same technique is useful to detect shapes of objects in any image.
5. In object detection application user can instantly take a picture through a camera and any object in picture can be tagged with the associated information.
6. The detected image can be send to another user the associated information is visible.
7. This application is used for the blind person.
8. The object that they can't see if the information is related to that object is store in his mobile then by using that they can easily search the object and control them.

VII. Conclusion

The image processing and recognition algorithm dedicated for blind users are proposed. The developed software tools were implemented and tested on the smart phones equipped with a digital camera: HTC Desire HD, HTC Exploer and Sony Xperia S. We noted that performance of these algorithms can depend on the quality of the built –in camera and image acqution lighting condition.

References

- [1] Neha Bari, Nilesh Kamble, "Android based object recognition and motion detection to aid visually impaired", International journal of Advanced in computer science and Technology Oct 2014.
- [2] Andreas Savakis, Roy Melton, "Low Vision Assitance Using Face Detection and Traking on Android Smartphones", 2012 IEEE.

- [3] K.Matusiak, P.Strumillo, "Object Recognition in a mobile phone application for visually impaired users", 2013 IEEE.
- [4] [Online] Available: http://ww.camera-sdk.com/p_253-how-to-accomplish-line-detection-in-c-onvif.html
- [5] [Online] Available: http://www.camera-sdk.com/p_271-how-to-create-rectangle-detector-in-c-onvif.html
- [6] Kanghun Jeong, Hyeonjoon Moon, Object Detection Using FAST Corner Detector Based On Smartphone Platforms, 2011 First ACIS/JNU International Conference on Computers, Networks, Systems, and Industrial Engineering.