International Conference on Mechanical, Industrial and Materials Engineering 2015 (ICMIME2015)
11-13 December, 2015, RUET, Rajshahi, Bangladesh.

Paper ID: AM-08

Automatic Exploration of Object from Partial Region of Interest (ROI)

Md. Rokunuzzaman*[1], S. Mandal[1], Md. Kamrul Islam[1] and Basu Deb[1]
[1]Departmentof Mechanical Engineering, Rajshahi University of Engineering & Technology (RUET), Bangladesh
*Email: rzaman.mte@ruet.ac.bd

Abstract

Intelligent system tries to explore things when it finds insufficient or partial information. The reason behind this behavior is curiosity. Usually curiosity drives the intelligent system to uncover the truth by searching and exploring. This paper is an attempt to observe this behavior with a simple experiment using vision system. In this experiment, a database has been prepared with features of some simple desktop objects. A web camera has been mounted on the top of an experimental base to capture the image of the simple objects. For processing the image and extraction of features of the observed object, OpenCV library has been used. A simple probability of recognition algorithm has been proposed using feature matching. The web camera has been fixed with a rack and pinion mechanism so that it can move up and down to explore the object according to the probability of recognition. The camera explores the object by moving itself up and down until it finds the maximum probability of recognition. The system then identifies the object and displays its name. This type of system is helpful in autonomous searching of objects, exploring un-identified objects, inspecting of products, collecting information from database and investigating of intelligent behavior in the research area of Artificial Intelligence (AI).

Keywords: Automatic exploration; Image Processing; Partial ROI.

1. Introduction

Robot object detection is concerned with determining the identity of an object being observed in the image from a set of known labels. Central to robot object recognition systems is how the consistency of an image, taken under different lighting and positions, is extracted and recognized. To work, algorithms are made to adopt certain representations or models, either in 2D or 3D, to capture these characteristics, which then facilitate procedures to tell their identities. The recognition process, which could be generative or discriminative, is then carried out by matching the test image against the stored object representations or models in the database. With more reliable representation schemes and recognition algorithms being developed, more progress continues to be made towards recognizing objects even under variations in viewpoint, illumination and under partial occlusion.

An image may be considered to contain sub-images sometimes referred to as regions-of-interest, ROIs, or simply regions. This concept reflects the fact that images frequently contain collections of objects each of which can be the basis for a region. In a sophisticated image processing system it should be possible to apply specific image processing operations to selected regions.

Automatic exploration is a built-in ability of intelligent systems. This ability gives a driving force to extract information from unknown environment. After extracting information, the intelligent system analyzes it and makes decision about the next task. To make such systems, many researches has been carried out in the field of object learning through curiosity [1] or active exploration [2], object detection from partially occluded objects[3], object detection by partial shape matching [4], recognition by components[5], object recognition from Nearest Neighbor Classification technique[6], object recognition from Bayesian inference where geometric primitives are used as prior knowledge for probabilistic integration[7][8], image searching from large database [9], automatic object detection and shape analysis[10] and so on. Although these researches are very relevant, our research is unique in terms of novelty and simplicity of its approach.

2. Experimental set up

Figure 1 shows the experimental setup for automatic exploration of object from partial ROI. The setup consists of a laptop, a web camera, rack-pinion mechanisms for webcam movement and controller circuits with a stepper motor.

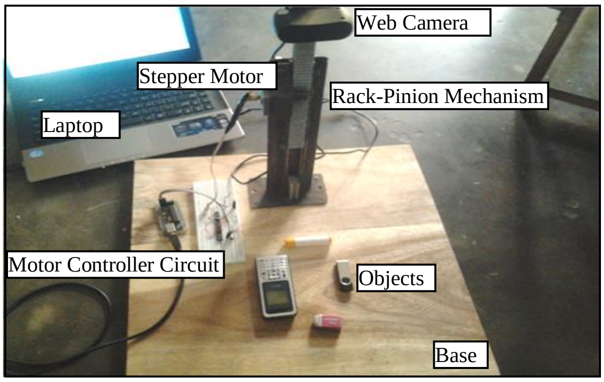

Fig 1: Experimental Setup

3. Methodology

3.1 Input

In this system, four simple desktop objects are selected to test our approach. These are computer, mobile phone, pen and pen-drive.

3.2 Tools and Platform

We have utilized the following state of the art tools and platform to implement our system:

- Software tools and Platforms: Open CV 2.4.5, Code::Blocks 10.05, C make 2.8
- Hardware Tools: Laptop: Pentium Core-2 Duo, 2GB Memory, Webcam: USB2.0 VGA UVC Webcam

3.3 Flow Diagram

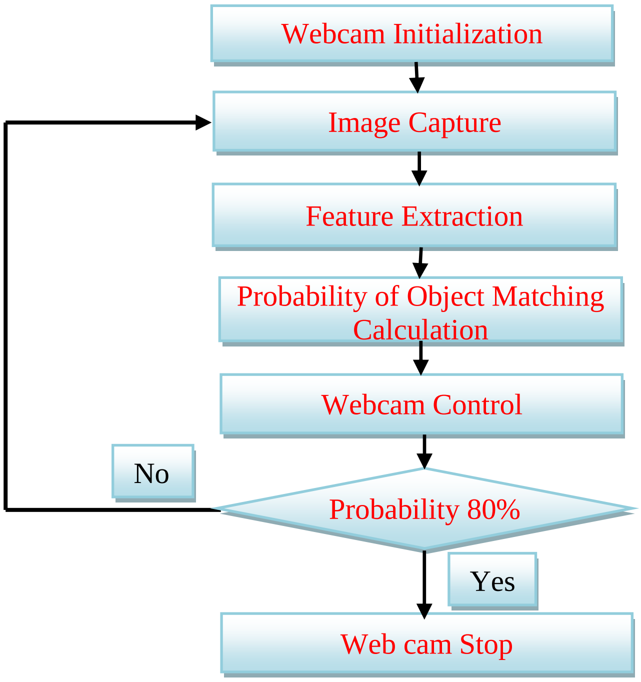

Fig 2: Flow diagram for automatic exploration of objects from partial ROI

Figure 2 shows the flow diagram for automatic exploration of objects from partial ROI. In this process, the webcam is initialized and object is placed on the base of the setup. Then the image of the object is captured and processed by using an image processing function developed by OpenCV [11]. After processing the image, features are extracted and probability of recognition is calculated. The details of feature extraction and probability calculation are stated in the following paragraphs:

3.3.1 Feature Extraction

A feature is an interesting part of an image, such as a corner, blob, edge, or line. Feature extraction process involves mathematical operation which produces feature vectors from input image. To identify an object, geometrical shape is an important parameter and can be easily evaluated by its area, perimeter, bounding box area, bounding box ratio, density etc. Therefore we have calculated these features using mathematical formulas. The process of feature extraction is depicted in figure 3.

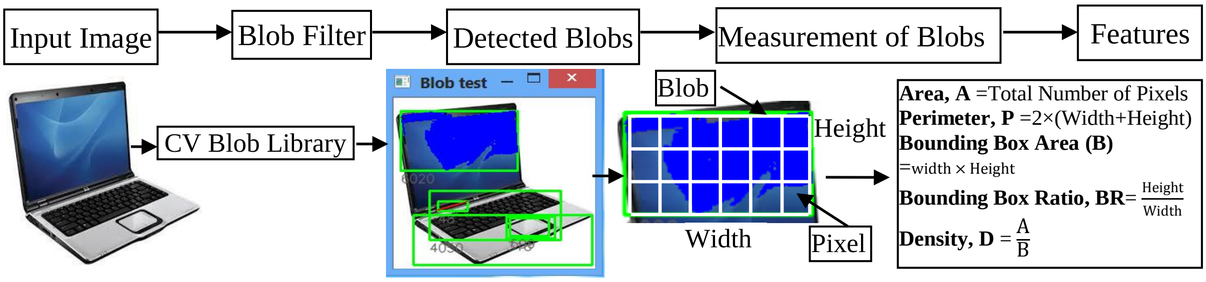

Fig 3: Feature extraction process

In the feature extraction process the image (here laptop) is processed by a Blob filter and detects Blobs (Binary large objects). In the field of computer vision, blob detection refers to mathematical methods that are aimed at detecting regions in a digital image that differ in properties, such as brightness or color, compared to areas surrounding those regions. Informally, a blob is a region of a digital image in which some properties are constant or vary within a prescribed range of values; all the points in a blob can be considered in some sense to be similar to each other. To implement blob detection, we have used CV Blob library for this purpose. The blob detection algorithm can be summarized as follows [12]:

Algorithm for extracting blobs from an image:

1. Convert the source image to binary images by applying thresholding with several thresholds from minThreshold (inclusive) to maxThreshold (exclusive) with distance thresholdStep between neighboring thresholds.
2. Extract connected components from every binary image by findContours and calculate their centers.
3. Group centers from several binary images by their coordinates. Close centers form one group that corresponds to one blob, which is controlled by the minDistBetweenBlobs parameter.
4. From the groups, estimate final centers of blobs and their radiuses and return as locations and sizes of keypoints.

This class performs several filtrations of returned blobs. One should set filterBy* to true/false to turn on/off corresponding filtration. Available filtrations:

- **By color**. This filter compares the intensity of a binary image at the center of a blob to blobColor. If they differ, the blob is filtered out. Use blobColor = 0 to extract dark blobs and blobColor = 255 to extract light blobs.
- **By area**. Extracted blobs have an area between minArea (inclusive) and maxArea (exclusive).
- **By circularity**. Extracted blobs have circularity ($4*\pi*Areaperimeter*perimeter$) between minCircularity (inclusive) and maxCircularity (exclusive).
- **By ratio of the minimum inertia to maximum inertia**. Extracted blobs have this ratio between minInertiaRatio (inclusive) and maxInertiaRatio (exclusive).
- **By convexity**. Extracted blobs have convexity (area / area of blob convex hull) between minConvexity (inclusive) and maxConvexity (exclusive).

Default values of parameters are tuned to extract dark circular blobs.

After detection of blobs, features are extracted which helps to identify the objects.

3.3.2 Probability of object matching calculation

Probability of object matching is calculated based on the feature matching between database objects and input object. We have created object databases using the measured features and stored it in the memory of the laptop. Each time when an input object is placed under the webcam, its feature is matched with the object database and probability of matched features are calculated and displayed in the monitor of the laptop.
We assume each object is defined as a set of measured features, *f*. Therefore, object at input and object in database can be formulated by

$$Obj[ID]_{input} = [f_a, f_b, f_c, f_d, \dots\dots f_z] \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (1)$$
$$Obj[ID]_{database} = [f_{a'}, f_{b'}, f_{c'}, f_{d'}] \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (2)$$

Therefore, the probability of matching feature *a* is calculated as

$$P_a = \frac{f_a}{f_{a'}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. (3)$$

Other probabilities are calculated in the same way. The probability of object matching is then calculated as

$$P_{Obj[ID]} = \frac{P_a + P_b + \cdots .. P_z}{N} \times 100\% \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Where, *ID* is the object name and *N* is the no. of feature considered.

3.3.3 Webcam Control: As shown the experimental setup, the motion of the webcam is controlled by a rack and pinion mechanism. A stepper motor is attached at the stand of the setup and moves a pinion which in turn gives a linear motion of the rack. A microcontroller (**AT89C51**) is used to drive the stepper motor. According to the program output the signal from the Laptop is fed through the USB port and connected to a microcontroller. The microcontroller sends the signal to the driver circuit of the motor and the motor drives accordingly. Initially the motor rotates to adjust the webcam so that the probability of identifying object reaches maximum. In the program we set the probability is 80% which means when an input object is matched 80% with the database object then the system will be stable and webcam will be fixed at that instant.

4. Results

Table 4.1: Probability of object matching for mobile and pen

Parameter	Object	Total	Probability
Area (A)	mobile	7745	0.14
	pen	1099	
Perimeter (P)	mobile	3797	0.14
	pen	546	
Bounding Box (B)	mobile	13958	0.38
	pen	5337	
Bounding Box ratio (BR)	mobile	131.808304	0.03
	pen	4.148256	
Density (D)	mobile	17.786625	0.05
	pen	0.881864	

Table 4.1 shows the probability of object matching for mobile and pen.The calculated value is

$$\text{Probability} = \left(\frac{0.14+0.14+0.38+0.03+0.05}{5}\right) \times 100\ \% = 14.2\ \%$$

Here we have found that the matching probability between the input image and database's image the probability value do not increase 80 % so the the input image is not pen.

Table 4.2 Probability of object matching for mobile and computer

Parameter	Object	Total	Probability
Area (A)	mobile	7745	0.65
	computer	11847	
Perimeter (P)	mobile	3797	0.65
	computer	2469	
Bounding Box (B)	mobile	13958	0.39
	computer	35720	
Bounding Box ratio (BR)	mobile	131.808304	0.02
	computer	2.518421	
Density (D)	mobile	17.786625	0.11

	computer	1.992728	

Table 4.2 shows the probability of object matching for mobile and computer.The calculated value is

$$\text{Probability} = (\frac{0.65+0.65+0.39+0.02+0.11}{5}) \times 100\% = 36.4\ \%$$

Since the matching probability value do not increase 80 % so the the input image is not computer.

Table 4.3 shows the probability of object matching for mobile and pen drive. The calculated value is

$$\text{Probability} = (\frac{0.7+0.48+0.48+0.09+0.20}{5}) \times 100\% = 39\ \%$$

Since the probability values do not increase 80 % so the input image is not pen drive.

Table 4.3: Probability of object matching for mobile and pendrive

Parameter	Object	Total	Probability
Area (A)	mobile	7745	0.7
	pendrive	11060	
Perimeter (P)	mobile	3797	0.48
	pendrive	1853	
Bounding Box (B)	mobile	13958	0.48
	pendrive	28974	
Bounding Box ratio (BR)	mobile	131.808304	0.09
	pendrive	11.807933	
Density (D)	mobile	17.786625	0.20
	pendrive	3.540914	

Table 4.4: Probability of object matching for mobile and mobile

Parameter	Object	Total	Probability
Area (A)	mobile	7745	0.98
	mobile	7912	
Perimeter (P)	mobile	3797	0.95
	mobile	3632	
Bounding Box (B)	mobile	13958	0.96
	mobile	14502	
Bounding Box ratio (BR)	mobile	131.808304	0.94
	mobile	140.0256	
Density (D)	mobile	17.786625	0.91
	mobile	16.26852	

Table 4.3 shows the probability of object matching for mobile and pen drive. The calculated value is

$$\text{Probability} = \left(\frac{0.98+0.95+0.96+0.94+0.91}{5}\right) \times 100\% = 95\ \%$$

The matching probability value increase 80% so the input image is mobile.

The program calculates the probability of matching for the input object with several objects in the database and show the results in the monitor. When the program finds the probability less than 80%, then it moves the camera ups and down to further calculate the value of probability. However, if the probability value is more than 80%, then the camera does not move which means it is stable.

5.Conclusion

We have successfully developed an automatic exploration system for object identification. This system is very simple and mimics an artificial intelligent system. However, the system needs lots of improvements. The system can only take images in 2D. Therefore, 3D objects cannot be processed with this system. The pose and orientation of the object is fixed in this system. In real world, object can be viewed in different poses and angles which give different recognition of the same object. Therefore, we need to build more robust system which can be developed with SIFT or SURF algorithms. There are lots of scopes are open for its development. The probabilistic object detection is very powerful method and works very well in the real world. Therefore, we can use Bayes Network to calculate probability for the object which will give better result. In conclusion, we have introduced a very simple system which solves automatic object exploration problem. This type of system is helpful in autonomous searching of objects, exploring un-identified objects, inspecting of products, collecting information from database and investigating of intelligent behavior in the research area of Artificial Intelligence (AI).

References

[1]Sao Mai Nguyen, Serena Ivaldi, Natalia Lyubova, Alain Droniou, Learning to recognize objects through curiosity-driven manipulation with the iCub humanoid robot, IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL), 18-22 Aug.pp. 1-8, 2013

[2]S. Ivaldi, S. M. Nguyen, N. Lyubova, A. Droniou, V. Padois, D. Filliat, P.-Y. Oudeyer, O. Sigaud, Object Learning Through Active Exploration, IEEE Transactions on Autonomous Mental Development, Volume: 6, Issue: 1, pp.56 – 72, 2013

[3]Bo Gun Parka, Kyoung Mu Leeb, Sang Uk Leea, Jin Hak Leec, Recognition of partially occluded objects using probabilistic ARG (attributed relational graph)-based matching, Computer Vision and Image Understanding, Vol. 90, Issue 3, pp. 217–241, June 2003

[4]Tianyang Ma and L.J. Latecki, From partial shape matching through local deformation to robust global shape similarity for object detection, IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2011, pages 1441 –1448, June 2011

[5]Biederman, I. Recognition-by-components: a theory of human image understanding. Psychological Review, Vol: 94, Issue: 2, pp.115‑147, 1987

[6]R.Muralidharan, Object Recognition Using K-Nearest Neighbor Supported By Eigen Value Generated From the Features of an Image, International Journal of Innovative Research in Computerand Communication Engineering Vol. 2, Issue 8, August 2014

[7]Daniel Kersten, Object Perception: Generative Image Models and Bayesian Inference, Chapter, Biologically Motivated Computer Vision, Volume 2525 of the series Lecture Notes in Computer Science, pp 207-218, 2002

[8] Leonardo Chang, Miriam Monica Duarte, Luis Enrique Sucar, Eduardo F. Morales, Object Class Recognition Using SIFT and Bayesian Networks, Proceedings of the 9th Mexican International Conference on Artificial Intelligence, MICAI, volume 6438 of the series Lecture Notes in Computer Science, pp. 56-66, Mexico, November 8-13, 2010

[9]Feng Jing, Mingjing Li, Member, IEEE, Hong-Jiang Zhang, and Bo Zhang, An Efficient and Effective Region-Based Image Retrieval Framework, IEEE Transactions on Image Processing, Vol. 13, No. 5, May 2004 pp. 699-709

[10]Kaichang Di, Zongyu Yue , Zhaoqin Liu, Automated Rock Detection and Shape Analysis from Mars Rover Imagery and 3D Point Cloud Data, Journal of Earth Science, Vol. 24, No. 1, pp. 125–135, February 2013

[11] Bradski, G, Kaehler, A "Learning OpenCV: Computer Vision with the OpenCV Library", O'Reilly Media, September, 2008

[12] OpenCV-3.0.0-dev, Simple blob detector-OpenCV documentation, retrieved 25 January 2014 from the web site http://docs.opencv.org/master/d0/d7a/ classcv_1_1SimpleBlobDetector.html