ISSUES IN DEVELOPING A CAMERA FOR A TENNIS PLAYING ROBOT

ABSTRACT

The paper is concerned with how to develop a camera application suitable for a robotic tennis player that can perfectly identify a ball which is moving and can detect the fence in the back of its court so that it will not hit the fence during a game. The steps required to gain this functionality are described briefly in this paper without any detailed demonstration of the codes and architectures of algorithms involved in each step.

INTRODUCTION

Computer vision is a sub-branch of artificial intelligence (AI) that deals with the processing of images from the real world. It is related to the generation of artificial images and visualization which ranges from a graphical presentation of numerical data to virtual reality. The identification of objects in the world of robotics is generally done through an image. This process starts with applying image processing techniques. The total process is a combination of low-level image processing in order to increase the quality of an image to high-level pattern recognition for fully understanding images and recognizing the specific features of objects.

As discussed by Webber [1] on image recognition, for low-level image processing the first step is the noise removal. The next step is feature extraction to locate edges, lines, and regions. An object can appear very different from different views with different angles. One important aspect is the decision about the features of objects. It is also important to distinguish an object from its shadow and background.

In order to manipulate an image several kinds of algorithms are used in image processing, such as convolution, Fast Fourier Transform, Discrete Cosine Transform, skeletonisation, edge detection, and enhancement of the contrast [2]. These techniques are generally implemented through software programming. However, special purpose hardware may also be used for better performance. Image processing has a huge impact in robot navigation as capturing images of objects and obstacles is analogous to the use of the human eye.

In higher level image processing, prior knowledge is utilized to classify data and extract patterns that provide statistical information and knowledge on the classification of measurements to define any object or point in a multidimensional space. The classifications of observations are vital for robotic path planning as description of each observation.

Sensors and cameras are used for observations. As written in an online Dictionary of Computing [3] by Denis Howe, a mechanism for feature extraction is included to compute numeric or symbolic information received through the observations. A scheme is generally used for classifying or describing observations based on the extracted features and on the availability of a set of patterns, also known as training set. There are three main approaches for feature extraction: syntactic (based on the structural interrelationships among features), statistical (based on the statistical characteristics of patterns generated through a probabilistic system), and neural (based on neural computing with neural networks).

THE ROBOTIC CAMERA AND IMAGE PROCESSING

The human eye is an ideal example of a natural vision instrument. In robotics, vision follows the same basic working principle as biological vision. This parallel study is coined as biomimetics. The designing of a camera for a tennis-playing robot with all the functionality analogous to human eyes consists of a few steps.

The design of the camera of a robot for playing tennis is complicated, as the robot needs to recognize several objects simultaneously while playing a game. Though the tennis ball is the main object to be identified, there are at least seven more things to be considered. These are the recognition of the position and height of the net between the two courts, the measurement of the distance between the net and the camera, varying positions of the opponent throughout a game, perfect analysis of the area of the opponent's court, the position of the fence behind the opponent, the continuous measurement of the distances of the robot from the sidelines along with the analysis of the current position of the robot in its own court through image analysis, and the distance to the fence in the back of the court.

As discussed in tutorials in Society of Robots [4] several initial steps are required for image analysis. These include the capturing and storing of real life images and storing of data, adjusting the file size through manipulating resolutions of images for improved processing time, and color correction by changing the color contrast to facilitate image recognition. Images are just information that is stored in computers. Different types of digital cameras work differently though there are mainly two types of technology available in the market: CCD and CMOS. The important thing is how to handle an image. An image consists of dots of color called pixels. When an image file is stored, the machine contains the information (color values and location) for each pixel, and each image has a fixed number of pixels per size that is generally known as its resolution. So it is easy to understand that more pixels per size mean more resolution resulting in more computational time for image processing. Decreasing resolution is a good way to start working with the problem, and there are many ways to reduce the file size through image resolution. [4, 5]

Another good way is through thresholding and [6] heuristics that reduce the color value of each pixel by statistically analyzing an image as a whole for overall brightness and counting the number of pixels containing the color information. This process is often used to improve image contrast, as it makes the bright pixels brighter and dark pixels darker. Determining the threshold value is vital to output a satisfactory result, as a poor threshold may result in a poor quality image. The image may be whited out, or it may become completely dark or closer to dark, decreasing the robot's ability to recognize an object in an image. 4D image analyses including time data is the way to develop the entire procedure for a mobile robotic camera, as it is required to identify the proximity of tennis balls, fences and the sidelines of a tennis court from one time to another. This is simple, as the time data is stored as sets of 2D matrix images where each pixel contains range data, and as each unit of time passes new 2D matrices are stored that are generally analyzed separately and then compared with previous images to process the changes over the course of time.

The next step in the image processing is called feature extraction [7], which is necessary to handle huge data with less useful information. Generally data is transformed and reduced into representation of sets of features that are generally called feature vectors. They provide relevant information to be utilized in processing. Edge detection is the technique for detecting sudden changes of color values in the pixels and lighting to locate the edges of an object, such as the edges of tennis balls (useful to get the shape of the ball though detection), as well as edges and corners of the courts, body edges of opponents, and the fences in the back. In this case choosing a good threshold value is vital as a poor threshold may result poor image output. In the previous step an image heuristics program may be used to get the average value that should be used as the threshold for the procedure in edge detection [8].

Now the camera can detect the edges of the objects, but that does not tell the robot whether a particular object is a ball or the net between two sides of the court. So it is necessary to identify the shapes of the objects perfectly to make advanced decisions. It can be done through the application of simple logic. The code for the edge detection detects the borders of an object. Now the number of continuous edges can be counted with a simple logic. If any sharp change in the direction of any line occurs, it should be counted as a different line. This is done by getting the average value of any two adjacent pixels in the image. Now it can be concluded whether an object is rectangular or circular depending on the number of edges. A rectangular shape should have four edges, whereas a circle has only one. But this is just a simple example how the logic works. In reality, more complicated logic that employs the application of probability is required for shape detection algorithm, as it is necessary to detect shapes of not only a ball, but also the shapes of the net, a human body (body of opponent), and the fence surrounding the court. [8]

The corner detection algorithms provide more information; not only on the points of the intersection of two or more edges, but also on any isolated point with maximum or minimum light intensity, any point on any curve where the curvature is locally maximal, and any point that is the end of any particular line [8]. But these kinds of algorithm are generally known as the detection of points of interest. There are several kinds of algorithms that can be used in this context, such as the Moravec corner detection algorithm [9], the Harris & Stephens / Plessey corner detection algorithm [9], the Shi and Tomasi corner detection algorithm [9], and the Wang and Brady corner detection algorithm [10]. This process works along with bold detection procedures.

Good middle mass and blob detection algorithms [8] are necessary, as it is important to identify the ball object in an image where in the background there are many static objects, such as the net and the fence behind the opponent, and dynamic ones, such as the ball and opponent. These kinds of algorithms are very crucial in this context because the robot is supposed to identify the ball perfectly in order to hit the ball back to the opponent's side of the court. For a threshold, color for tennis balls should be fixed and a middle mass detection is necessary to identify the mid-point (average of all pixels with the threshold color) of the ball. All the pixel values in the array will be read. If the color of any pixel matches with the threshold color, it will be labeled with 1 otherwise it will be labeled with 0. Moreover, if there is any pixel found with a similar color to the threshold value but not adjacent to the previous match, it should be marked with 2 or any other number, as this particular logic is helpful in determining whether there is more than one ball moving on the court or to avoid misinterpretation of the actual ball for hitting. Along with blob detection, it is advisable to run an algorithm for the removal of non-threshold colors from the background to get a better result.

Next, the step for motion detection procedures is necessary to find or track the motion of the tennis ball. There are many approaches, but the most common approach is comparing the current image with previous frames where images are in grayscale. The objective is to find the part of the image where they differ. It is generally done by counting the pixels, and if the value is greater than the threshold value considered with the unit of time interval, it can be said the motion of the ball is successfully detected. However, there may be a number of noisy images that may produce faulty motion where there is no motion at all. This situation can be avoided through erosion filters based on the basics of noise reduction techniques.

There are mainly two types of noise in images, salt and pepper noise [14, 15] and Gaussian noise. Salt and pepper noise appears as black and white dots on the image and is also known as independent or random noise The pixels have huge differences in colors from the surrounding pixels with no relation to the colors of other pixels of any image. Gaussian noise, also known as dependent noise, is the noise present in every part of the image, where every pixel in the image is changed from its original value by a small amount. Gaussian filters, or non-linear filters, can be used in this context to avoid noisy images. So a combination of a frame-to-frame comparison algorithm with respect to time variation, along with a noise reduction procedure, can successfully detect the actual motion of a tennis ball and even the motion of the opponent on the other side of the court to determine where to hit the ball back to the opponent.

As discussed in Society of Robots [10] tracking for advanced motion detection can be achieved by utilizing the calculations for the motion of the middle mass of the ball to identify the changes of direction of the ball due to the hit of the opponent player on the other side of the court. Using vector math and the ratio of pixels to distances, the measurement for displacement or acceleration can be done. The difference between the middle mass of a frame and another frame after a unit time interval from the previous one can be multiplied with distance per pixel to get the speed of the ball. For this algorithm "determining the distance to pixel ratio" [10] is an issue as the camera may be at an angle with the horizon, or there may be some lens effect in the camera. A separate algorithm is required to map the ratio for given pixels on the X-Y plane. The distortion constant of a lens can be calculated from the datasheet of the camera or experimentally. During tracking, crossover is a major problem to be considered. When more than two objects cross over each other, the tracking algorithm gets confused about which one to track. So to avoid this problem it is essential for the algorithm to remember some object-specific features for each object. It is necessary because by using the stored features, the algorithm one can use the techniques of template matching [10] to distinguish one object from another.

According to Society of Robots [8] the issues related to pixel classification becomes relevant, as it is particularly useful in identifying the region of the court along with its boundary lines and in identifying the obstacle (here the net in the middle of the court) for avoidance. This is done through the assignment of each pixel of any image to object classes. For example, the light green color would be a ball, where the dark green color would reflect the region of the court, and thick white lines would be regarded as the sidelines. Just using the techniques for middle mass detection or maximizing the RBG values of each pixel would work efficiently. For example, pixels with more green than others would become dark green for the identification of the court area. This process may not be highly perfect, but it could be utilized in this context along with template matching procedure.

In the discussion [8] of Society of Robots the techniques for stereo vision will work well to determine the 3D location of the ball and the position of the opponent player on the opposite side of the court. A single pinhole camera or two parallel or non-parallel cameras can perfectly measure the actual three dimensional position of the ball.

Visual servoing is also a useful method for determining the position data of the tennis robot utilizing visual feedback [8, 11]. If a robot needs to pick up a tennis ball, visual servoing can be used to move the arm to that location. Visual features enable vision system to control a robotic system through image information for measuring the difference between the present position and the expected position in future. To move inside the court, visual servoing would track the whole court with the sidelines. For visual servoing, it is necessary to identify the ball and judge its motion. Here an algorithm is required to predict the position of the ball while it is moving, and then the robot needs to make a decision about how to orient itself in order to reach the location predicted previously. A Proportional Integral Derivative Controller, modified form of "Hierarchical Partially Observable Markov Decision Process" [8, 12] can be used for controlling robot-motions.

A number of steps are discussed so far, but the whole process needs the acquisition of images first. According to Pulkit Gaur's information in Robotics India [13] there are several ways, such as real time acquisition and still acquisition, where still acquisition is a comparatively easier process than real time acquisition, but this process works better for a fixed distance of an object, so the real time acquisition fits better in this particular case. In real time acquisition video cameras convert images into electrical signals that describe light intensity for each line that may be used for detecting the fence and the net through the method of Fourier Transform that represents an image in a frequency domain where the actual image represents the spatial domain. Signals are generally converted to pixels for image processing using frame grabbers to describe color intensity of pixels, but sometimes problems arise as the amount of data is huge for an image, and the speed of acquiring images in video cameras is very high. However, the problem can be solved through the reduction of resolution [13].

CONCLUSION

The development of a tennis playing robot, a prefect example of an expert system is complicated, so special attention and expertise are required to make the robots work at expert-human level. The steps for successfully developing the vision system for a tennis robot are discussed without any complicated details on programming and hardware interfaces. Among huge numbers of established techniques in machine learning and computer vision, only relevant processes are mentioned so that students can understand how machine vision works for a particular type of system and they can develop expertise through further learning on each step discussed in the paper.

The architectures of artificial vision systems differ depending on the desired functionalities of the whole systems, but the basics are the same in each development. Development of realistic vision systems contributes a lot in industrial automation, biomedicine, modern robotics, bio-medicine, navigation, and even in satellite observation and space explorations.

Artificial vision in a tennis playing robot is a good example of the application for designing engineering systems with cutting edge technology through mimicking the natural processes and systems. These applications are categorized in a newly emerging field of science, biomimetics that will continue to become more sophisticated through continuous researches and applications of basic biological working principles.

REFERENCES

1. Webber, H., C., Image Processing And Transputers, Washington, WA: IOS Press, 1992.

2. Pitas, I., Digital Image Processing Algorithms And Applications, New York: Wiley-IEEE, 2000.

3. Howe, Denis, Image Processing, 1997, http://burks.brighton.ac.uk/burks/foldoc/32/56 .htm, Nov 2007.

4.Society of Robots, Computer Image Processing, 2005-2007, http://www.societyofrobo-ts.com/programming_computer_vision_tutorial_pt2.shtml, Nov 2007.

5. Wikipedia, Thresholding (Image Processing), Sep 2007, http://en.wikipedia.org/ wiki/Thresholding_%28image_processing%29, Nov 2007.

6. Wikipedia, Heuristic, Nov 2007, http://en.wikipedia.org/wiki/Heuristics, Nov 2007.

7. Wikipedia, Feature Extraction, September 2007, http://en.wikipedia.org/wiki/Feature_e xtraction, Nov 2007

8. Society of Robots, Programming – Computer Vision Tutorial Computer Vision Algorithms, 2005-2007, http://www.societyofrobots.com/programming_computer_vision _tutorial_pt3.shtml, Nov 2007.

9. Wikipedia, Corner Detection, Oct 2007, http://en.wikipedia.org/wiki/Corner_detectio-N, Nov 2007.

10. Society of Robots, Computer Image Processing, 2005-2007, http://www.societyofrob ots.com/programming_computer_vision_tutorial_pt4.shtml, Nov 2007.

11. Kragic, D., Christensen, H.I., Survey on Visual Servoing for Manipulation, http://www.societyofrobots.com/robottheory/Survey_on_Visual_Servoing_for_Manipulat ion.pdf, Nov 2007.

12. Theocharous, G., Rohanimanesh, K., Mahadevan, S., Approximate Planning with hierarchical partially observable Markovdecision Process models for Robot Navigation, Robotics and Automation Proceedings, IEEE International Conference, *2*, 1347-1352, 2002.

13. Gaur, P., Robotics India, Theory of Image Processing, Sep 2007, http://www.roboticsindia.com/modules.php?name=News&file=article&sid=61, Nov 2007.

14. Farid, H., Fundamentals of Image Processing, http:// www.cs.dartmouth.edu/farid/t-utorials/fip.pdf, Nov 2007.

15. Kuiper, A., Sigmund, M., Simulating of Authentic Movie Faults, Computer as a Tool, 2005. EUROCON 2005.The International Conference on, *2*, (2005), 1015 – 1018, 2005.