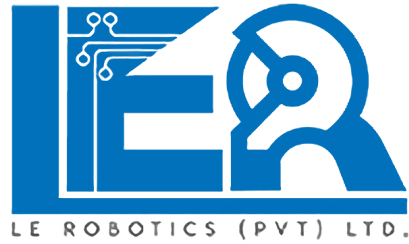


L.E. Robotics (Pvt.) Ltd.

No. 100/4, Divulapitiya Road, Minuwangoda, Sri Lanka



Machine Vision based Real Time Trajectory Generation

Prepared by Thalagala B.P.
Last Modified on January 13, 2022

Contents

1	Feasibility Study	5
1.1	Introduction	5
1.2	Challenges Encountered when Incorporating a Vision Unit to a Robotic System	5
1.3	A Traditional Machine Vision Approach as the Solution	6
1.4	High Level Representation of the Algorithm	6
1.4.1	Object Detection: Connected Component Analysis (CCA) and Background Subtraction	7
1.4.2	Object Modeling: Contour Analysis and Representation	7
1.4.3	Object Tracking: Template Matching followed by Object moments interpretation	8
1.4.4	Related functions in OpenCV computer vision library	8
2	Underlying Concepts of the Algorithms	9
2.1	Object Detection: Connected Component Analysis (CCA) and Background Subtraction	9
2.2	Object Modeling: Contour Analysis and Representation	10
2.2.1	Topological Structural Analysis of Digitized Binary Images by Border Following	10
2.3	Object Tracking: Template Matching followed by Object moments interpretation	10
	Bibliography	11

Chapter 1

Feasibility Study

1.1 Introduction

Most of the robotic arms used in industrial environments operate in a pre-programmed cycle. When it comes to the way a human does the same task is much different as the path planning for picking an object may change from cycle to cycle because of the perception obtained through human vision.

Machine vision is the technology and methods incorporated to mimic the human vision in order to gain the insights about the operating environment of the robotics system. However, when it comes to the real time object detection using machine vision, there is an inevitable trade-off between the accuracy and the speed of the operation. This depends entirely on the used machine vision algorithms and the computational power of the available hardware.

1.2 Challenges Encountered when Incorporating a Vision Unit to a Robotic System

If the robotic system/ arm in interest is not controlled through a dedicated industrial PC with adequate computational resources, the amount of resources that can be allocated to the vision unit becomes limited. This will eventually result in great delays (which is not desirable when it comes to real time operations) to produce the required outputs by processing the acquired images through the associated camera.

Due to this limitation in computational resources it is not practical to use Deep Learning based algorithms in time critical industrial applications. Because, by the time the decisions are made the environment may have already changed and the made decisions may no longer valid.

1.3 A Traditional Machine Vision Approach as the Solution

To overcome the challenges mentioned above traditional machine vision algorithms can be incorporated inside the vision unit. Unlike the deep learning based algorithms, traditional machine vision algorithms are less computationally expensive and no initial data is required to train the algorithm. As they are built upon strong mathematical foundations they work well in almost all the situations when the parameters are tuned properly.

1.4 High Level Representation of the Algorithm

For real time trajectory generation we first need to track the objects on the conveyor belt. A typical object tracking framework generally consists of three major modules which can be identified as, 1. Object Detection, 2. Object Modeling and 3. Object Tracking. The following traditional computer vision methods will be used to implement those three modules.

Module	Method
Object Detection	Connected Component Analysis and Background Subtraction
Object Modeling	Contour Analysis and Representation
Object Tracking	Template Matching followed by Object moments interpretation

Table 1.1: Modules of a Typical Object Tracking Framework

Through the above framework we can identify and track the centroids of the objects on the conveyor belt and that information can be handed over to the trajectory generation algorithm to generate the required path in the real time to pick the object. Following subsections highlight the contribution from each module tabled above where a conveyor belt which carries hexagonal nuts is used as the example.

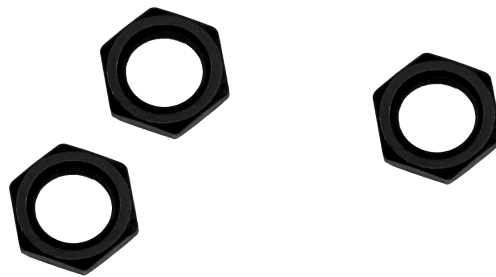


Figure 1.1: Single frame of the video captured from the camera

1.4.1 Object Detection: Connected Component Analysis (CCA) and Background Subtraction

As the first step of the object detection and tracking, the foreground (objects) must be separated from the background. This segmentation is done at this stage using *connected component analysis* (CCA). Through the CCA different connected objects found in the image can be assigned different labels. Those label information can then be used to subtract the background from the image. Prior to CCA, thresholding followed by a suitable morphological transformation must be done. For this example *Otsu's thresholding* and *morphological closing* are used.



(a) Labeling of connected objects using CCA



(b) Subtracting the background

Figure 1.2: Connected Component Analysis (CCA) and Background Subtraction

1.4.2 Object Modeling: Contour Analysis and Representation

Background subtracted image is used in this stage to find contours. These contours can be then used as a representation of the objects on the conveyor belt.

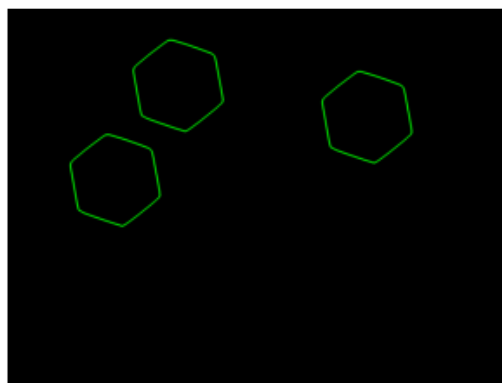


Figure 1.3: Finding contours in the image

1.4.3 Object Tracking: Template Matching followed by Object moments interpretation

In order to identify different objects a procedure called as *template matching* is carried out. For this template matching, reference images known as template images should be created using a properly captured images of the objects that are intended to convey using the conveyor belt. Contours of these templates can be compared with the object contours found in the previous stage to identify similar objects for further processing.



Figure 1.4: Sample template image for a conveyor belt carrying hexagonal nuts

1.4.4 Related functions in OpenCV computer vision library

1. `cv.threshold`
2. `cv.morphologyEx`
3. `cv.connectedComponentsWithStats`
4. `cv.findContours`
5. `cv.drawContours`
6. `cv.matchShapes`
7. `cv.moments`

Chapter 2

Underlying Concepts of the Algorithms

2.1 Object Detection: Connected Component Analysis (CCA) and Background Subtraction

2.2 Object Modeling: Contour Analysis and Representation

2.2.1 Topological Structural Analysis of Digitized Binary Images by Border Following

Authors of the Paper : SATOSHI SUZUKI, KEIICHI ABE

“Two border following algorithms are proposed for the topological analysis of digitized binary images. The first one determines the surroundness relations among the borders of a binary image. Since the outer borders and the hole borders have a one-to-one correspondence to the connected components of l-pixels and to the holes, respectively, the proposed algorithm yields a representation of a binary image, from which one can extract some sort of features without reconstructing the image. The second algorithm, which is a modified version of the first, follows only the outermost borders (i.e., the outer borders which are not surrounded by holes). These algorithms can be effectively used in component counting, shrinking, and topological structural analysis of binary images, when a sequential digital computer is used. © 1985 Academic press, IC.”

2.3 Object Tracking: Template Matching followed by Object moments interpretation

Bibliography