Image processing techniques for coin classification using labview

BHAGIRATHI V
Department of Mechatronics
Acharya institute of technology
Bhagi.suryavanshe@gmail.com

MEGHANA M SINTHRE
Department of Mechatronics
Acharya institute of technology
meghs.sinthre@gmail.com

DILIP R
Department of Mechatronics
Acharya institute of Technology
dilipr.me@gmail.com

Abstract

The aim of the project was to develop a simulation for detecting and classifying currency coins by machine vision using NI 1742 Smart Camera. Powered by a 533 MHz PowerPC processor, it greatly enhances processing speed as it is a dedicated processor simplifying machine vision by analyzing images directly. For programming the same, LabVIEW RT (Real- Time) software was used in conjunction with Vision Assistant 2010.

Keywords: Lab View RT module, Vision assistant, NI camera

1. INTRODUCTION

Coin numeration can be done by finding out the weight or by using a vision system. We have used real time vision and image processing techniques for the identification of different coins. An extension of such a system would allow separation and counting of large number of coins of different denominations in places such as banks, temples, etc. This application can be easily extended to sort out coins of different countries.

As long as the background conditions can be controlled sufficiently, the coin detection task becomes almost trivial. If the conveyor belt is homogenous and always darker (or brighter) than the coins, a simple threshold operation suffices for the separation. Once the current coin is separated from the background, the desired features can be extracted.

Prime Objectives of the project were:

• To get acquainted with the LabVIEW 2010 software and its applications.

• To develop an algorithm for the specified problem and build a LabVIEW code for the same.

• To construct a hardware setup mimicking the real world problem using NI-1742Smart Camera.

• To test and validate the simulation.

2. METHODOLOGY

Software Used: LabVIEW 2010 with RT module, Vision Assistant 2010.
Hardware used: NI 1742- Smart Camera.

3. WHAT IS LABVIEW?

LabVIEW (short for Laboratory Virtual Instrumentation Engineering Workbench) is a platform and a graphical programming environment developed by National Instruments (NI). It allows one to program with graphical functional blocks (that can be dragged- and - dropped) instead of writing lines of text. An important aspect is Dataflow Representation which allows for easy development and understanding of the code.

LabVIEW is used worldwide to develop sophisticated measurement, test, and control systems using intuitive graphical icons and wires that resemble a flowchart. It offers consummate integration with thousands of hardware devices and provides hundreds of built-in libraries for advanced analysis and data visualization – all for creating virtual instrumentation (VI). It creates automated forms of measurement and processing instruments used in any typical laboratory setup. The LabVIEW platform is scalable across multiple targets and Operating Systems (OS), and, since its introduction in 1986, it has become an industry leader. [1]

The latest version of LabVIEW is version LabVIEW 2010, released in August 2010. LabVIEW Core I and II courses were taken at NI, Bangalore during 23rd-27th May, 2011.

356

3.1 hardware integration with labview

A significant benefit of LabVIEW over other development environments is the easy access of instrumentation hardware through built in libraries and thousands of instrument drivers.

Drivers and abstraction layers for many different types of instruments and buses are included or are available for inclusion. The provided driver-interfaces save program development time. Thus we can write programs and deploy test solutions in a reduced time frame when compared to more conventional systems.

3.2 real time module in lab view

The National Instruments LabVIEW Real- Time Module is an add-on component for the LabVIEW Development System.

Fig 1: lab view real time module

Programming graphically in LabVIEW can greatly improve one's programming efficiency, and this same graphical approach can be used with the LabVIEW Real-Time Module to create stand-alone systems that run for extended periods of time.

While LabVIEW is commonly used to develop applications that run on desktop OSs such as Windows, Linux etc., these OSs are not optimized for running critical applications for an extended period of time. The LabVIEW Real-Time Module features real-time OS (RTOS) software that runs on NI embedded hardware devices. Thus using LabVIEW, we can easily develop or debug codes which can be then directly downloaded to and executed on embedded hardware devices such as NI CompactRIO, NI Single-Board RIO, PXI, vision systems, or even third-party PCs made possible by using LabVIEW RT. We have used it for programming NI 1742 Smart Camera which is embedded with a 533 MHz Power PC processor.

3.3 vision assistant 2010:

It is an independent module but once installed, it creates additional options in the functions toolbar of LabVIEW Block Diagram - Vision and Motion. It provides an express VI, Vision Assistant which can be used to create, edit and/or run vision algorithms using Vision Assistant 2010. When this VI is placed in the Block Diagram, Vision Assistant is launched where we can build algorithms using the abundant functions offered for processing images.

A few of the functions employed in the present project were:

- Threshold
- Morphology Fill holes, Particle Filtering
- Particle Analysis-report
- Pattern Matching.

Controls and Indicators that are to be programmatically used through LabVIEW can also be selected in case of Vision Assistant Express VI. Alternately, all the functions of Vision Assistant can be selected individually for building the algorithm directly in **LabVIEW through the drag-and-drop functions available in Vision and Motion Functions Toolbar. It is to be noted that these functions will be available in LabVIEW only after Vision Assistant is installed.**

4. NI 1742 SMART CAMERA

NI 17xx Smart Cameras simplify machine vision by analyzing images directly on the camera with a powerful, embedded processor capable of running the entire suite of NI vision algorithms. [2]

A multifunctional vision system that can transmit not just raw acquired images but also the examined results, it's an amalgamation of the onboard processor with a charge-coupled device (CCD) image sensor.

Housed in a rugged metal case, all NI Smart Cameras offer built-in I/O, multiple industrial protocols, built-in Web servers, and many other features. The NI Smart Camera was programmed with LabVIEW Real-Time.

- Real time Machine Vision
- High-quality monochrome VGA (640x480) CCD image sensors with image acquisition rate up to 60 fps.
- High-performance embedded processors – 533
- MHz processor
- Isolated 24 V digital I/O
- Dual gigabit Ethernet ports – used for cross connection with the operating computer RS232
- serial support.

357

BHAGIRATHI V, MEGHANA M SINTHRE, DILIP R

Fig 2: NI 17xx Smart Cameras

4.1 HARDWARE DETAILS

Processor Configuration:	Processor	533 MHz Power PC
Memory:	System Memory	128 MB
	Firmware and job storage	128 MB
Image Sensor	Resolution	640x480
	Image Size	1/3 in.CCD
	Pixel Depth	8–bit
	Color	Monochrome (grayscale)
	Acquisition rate (fps)	60
	Partial Image Acquisition	✓
Display		Web Interface
Operation Temperature Range	Camera	0 to 45 °C
Application Development		LabVIEW / Vision Builder AI
Power Supply		regulated 24 V supply through DB15 pin connector

Fig 3: hardware details

Vision Development Module – Vision Assistant 2010 recognizes Smart Camera 1742 as soon as it is connected via a cross cable to the computer. But for image acquisition programmatically through LabVIEW, an IMAQ session has to be created using IMAQ Init.vi. Subsequently, IMAQ Snap.vi or IMAQ Grab Acquire.vi can be used for Snap (acquiring one image at a time) or Grab (Continuous acquisition of images) functions.

5. SIMULATION SETUP

The temporary setup was made by screwing the NI 1742 Smart Camera to a wood board .The illumination was set to optimum limit by using a 25W electric bulb in series with a normal fan regulator. A temporary stand was used for adjusting the height in order to set proper focus.

5.1 labview code

The code must be deployed on the Camera, so a Real-Time Project (.lvproj) was created from the Getting Started Window. In the Browse Target option, we selected Smart Camera from Select a new Target option. The code was written on a VI that was already added to the Smart Camera, here Procyon (IP-192.168.1.2). The code comprised of three stages:

- Wait Phase: The system waits for an input from the user.
- Learning Phase: The user gives training data for the machine to learn a particular type of coin and its attributes/features.
- Classification Phase: The user gives random coins for classifying according to the types learnt in the learning phase.

The user has an option of choosing from:
- Learning: training the system,
- Classification: classifying random samples
- Done: finishing the run

5.2 theoretical background

Essentially, the problem of coin separation boils down to feature based classification. It can be perceived as the act of taking in raw data and taking an action based on the "category" of the pattern/ feature. [3]
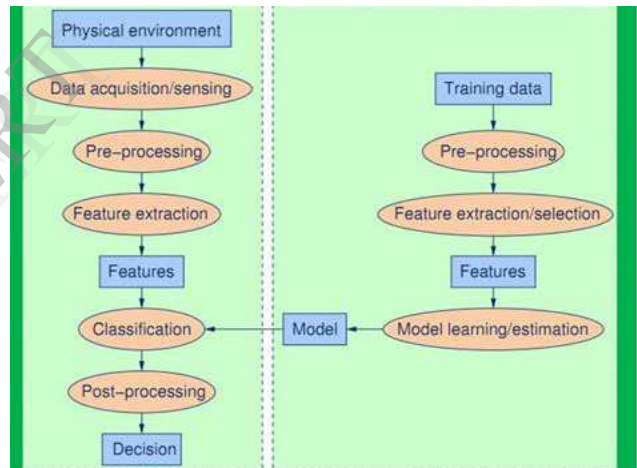

Fig 4: Process Flow Diagram

1. Data acquisition Images were acquired using IMAQ Snap.vi in LabVIEW after initializing a session with IMAQ Init.vi and initializing an image with IMAQ Create. Unless the code is written to a VI added to Procyon (Smart Camera), there occurs a Camera interface name conflict error. (Error -1074397163; Possible reason(s): NI-IMAQ: The passed in interface or session is invalid.)

2. Pre-Processing Camera's signals are pre- processed to simplify the subsequent images without losing relevant information. This stage may include filtering for removal of noise, Image sharpening, color plane extraction: [RGB (red/blue/green) or HSL (Hue/ Saturation/ Luminance) or HSV (Hue/ Saturation/Value) or HSI (Hue/ Saturation/Intensity)], segmentation (to isolate coins from

358

the background), threshold etc. It is required to know the coin area from the background in order to be able to extract features. Such a technique is called segmentation. Segmentation is to subdivide an image into its component regions or objects. [4]

Segmentation algorithms generally are based on one of the two basic properties of intensity values namely discontinuities and similarities. Discontinuity based algorithms partition an image based on sharp changes in intensity (such as edges) and the latter partition an image into regions that are similar according to a set of predefined criteria. Threshold (Fig 5) takes in both discontinuity and similarity criteria.

Since NI 1742 Smart Camera produces gray scale images, intensity plane extraction was not required as a separate step. Threshold was directly applied to the image taken by the camera.

3.IMAQThreshold.vi

NI_Vision_Development_Module.lvlib was employed for this project with its Replace Value input equal to 1 and Front Panel controls for the Range input. This was done to allow the user to set appropriate threshold range limits during learning phase. This information would then be utilized by the feature extractor in the classification stage. The output image after this stage is a binary image with only two regions, background and coin area.

This image was given as an input to Vision Assistant.vi where a script containing the following functions was written.

- Advanced Morphology -> Remove Small Objects: This was done to remove any unscrupulous particles that may have come in the image.
- Advanced Morphology -> Fill Holes: To cover the entire coin area including the portions which may not have been identified by earlier step of threshold due to imperfect threshold range suitability.
- Advanced Morphology -> Separate Objects: If more than one, coins are placed, they must be separated.
- Particle filter: Removes or keeps particles from an image specified by the filtering criteria. Particle filtering was done with Heywood circularity factor (HCF). Particles having HCF between 0.8 and 1.2 were retained. This was done to reject other unwanted particles in the image like dust particles etc.
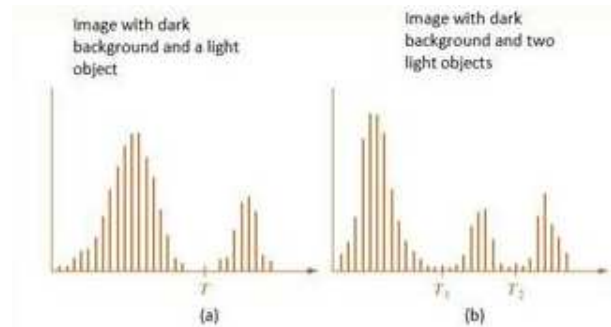

FIG 5: Grey Level histograms a) single threshold and b) multiple thresholds

4. FEATURE EXTRACTION

A set of characteristic measurements (numerical or non-numerical), and their relations are extracted to represent patterns for further process.[5] The task of feature extraction is problem and domain dependent and thus requires knowledge of the domain.

For this, it is important to look for distinguishing features that are invariant to irrelevant transformations like rotation, scaling, translation, occlusion, projective distortion, etc. Distinguishing features are those for which feature values are similar in the same category and very different in different categories. For the present project area, perimeter and HCF were the features selected.

Area represents the number of pixels lying within the region identified as coin. Perimeter represents the number of pixels lying on the boundary of this area. Heywood Circularity Factor (HCF) is defined as the ratio of the particle perimeter to the perimeter of a circle of the same area. A perfect circle would have an HCF of 1. It is a useful tool for shape analysis.

In the Vision Assistant.vi script, the last step was Particle Analysis where the above three parameters were selected under Select Measurements Option. The particle measurement report was stored in an array. During the learning phase, these reports were all combined to form a database from where the comparisons were made in the classification phase. The user is prompted to add a name for each type of coin which was also added to the particle's attributes in the database.

5. CLASSIFICATION

The process or events with similar properties are grouped into a class. The number of classes is task-dependent. [5] So it can be seen as the act of assigning an object to a category by using the feature vector. Difficulty of classification depends on the variability of the feature values in the same category relative to the difference between feature values in different categories. The variability of feature values in the same category may come from noise.

Features Extracted from the sample coins of unknown denomination were matched within tolerance limits of the default values in each class or coin type in the database. In the classification phase, all the steps starting from

359

BHAGIRATHI V, MEGHANA M SINTHRE, DILIP R

image acquisition to particle analysis report generation are the same as that in learning phase, as discussed above. A classifier stage was designed which takes this report and compares each feature to the database created in the learning phase. When a match for all the three features is found, the classifier returns the type of the coin as entered by the user during learning phase.

360

6. FINDINGS AND CONCLUSIONS

In this project we have successfully been able to sort Indian currency coins of different denominations on a Real-Time embedded device. A sample image and the output images after every step of the image processing part of the algorithm are shown below:

Fig 6: Acquired Image   Fig 7: Threshold Image
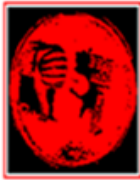
Fig 8: After removing small particles   Fig 9: filling holes

Fig 10: After separating particle   Fig 11: after particle filtering

Results ...	1	
Perimeter	666.03643	
Area	18585.00000	
Heywood Circularity Factor	1.37820	

Fig 12: Particle analysis report

The particle analysis reports for each coin image were generated after all the preceding steps. These reports built up the database in the learning phase. And for classification phase, this report was searched for in the database created. Accordingly, match/ no match were found.

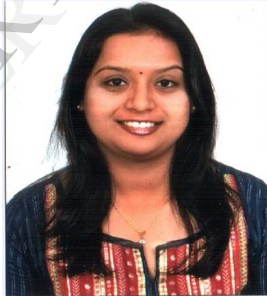A small live demo of the project was given by us at the System Integrators Business and Technical Meet 2011 organized by NI, India on June 28, 2011 at The Park, Kolkata.

The system is not yet capable of sorting heaps of mixed coins. But coins coming on a conveyer belt can be easily classified using an extension of the program. Unknown coins are rejected. Further research may be carried out to improve the recognition result and speed

REFERENCES

[1] www.ni.com

[2] NI 1742 Smart Camera Datasheet, ©2008 National Instruments.

[3] Duda RO, Hart PE, Stork DG. Pattern Classification, 2nd ed. New York: Wiley Inter science; 2000.

[4] Gonzalez RC, Woods RE. Digital Image Processing. 3rd ed. Addison-Wesley Pub (Sd); 1992.

[5] Qi X, Basic Components of Pattern Recognition and Feature Selection, REU Site Program in CVMA; 2011.

BIOGRAPHY OF THE AUTHORS

Bhagirath i V has received her M.Tech degree in Digital communication from Visvesvaraya Technological University in the year 2007. She is currently working as Assistant professor in Acharya institute of technology Bangalore. Her research interest is in the area of Communication system, signal conditioning and RF technology.

Meghana M Sinthre has received her M.Tech degree in VLSI and Embedded system from Visvesvaraya Technological University in the year 2011. She is currently working as Assistant professor in Acharya institute of technology Bangalore. Her research interest is in the area of Communication system, signal conditioning and RF technology.

361

DILIP R has received his M.E degree in Control and instrumentation from Bangalore University in the year 2012. He is currently working as Assistant professor in Acharya institute of technology Bangalore. His research interest is in the area of Control System, Signal Conditioning and Process and Control Instrumentation