

Defect Detection of Vehicle Surface

Armen Petrosyan

Institute for Informatics and Automation Problems, National Academy of Sciences of Armenia, National Instruments Armenia
e-mail: armen.petrosyan@ni.com

Hakob Sarukhanyan

Institute for Informatics and Automation Problems, National Academy of Sciences of Armenia, Yerevan, Armenia
e-mail: hakop@iiap.sci.am

Abstract—The task of quality control and automatic defect detection on production lines is one of the most important tasks of machine vision and digital image processing (DIP). This article provides one example of solving such a problem using various DIP algorithms. For the algorithm realization, NI LabVIEW and NI Vision Development Module have been used. The suggested method of defects detection allows to automatically find the location of the defect with high accuracy.

This is a practical example, which was tested on the production line.

The main idea used in the image processing part of the system is particle analysis.

Keywords—Edge Detection, Particle and Contour Analysis, Digital Image Processing, LabVIEW, Machine Vision.

I. INTRODUCTION

To solve the problem on the production line, 4 cameras were installed on the ceiling of the production room. All cameras were connected to industrial controller, IC-3173 (see Fig. 1). This industrial controller has the ability to simultaneously collect video data from 4 cameras with GigE interface, as well as two of them have USB 3.0 ports, which were not used in the task. Also, the controller has digital inputs and outputs, so it is possible to use and integrate other sensors into the system, such as sensors for the horizontal movement of car parts, thus, it is possible to detect the moment, when the part is fed under the angle of view of cameras for synchronous collection of video data. LabVIEW environment [1] was used as software.

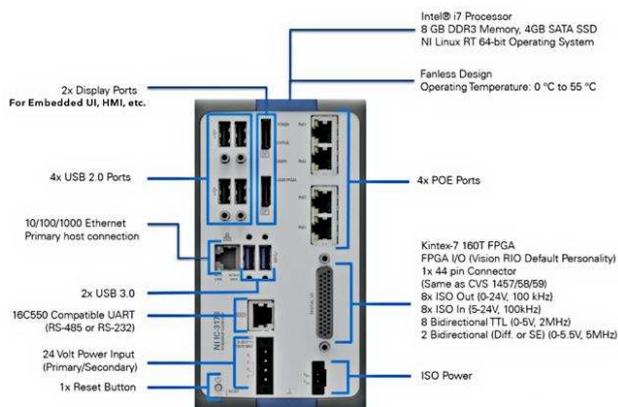


Fig.1. Industrial controller for video data collection

II. PROBLEM STATEMENT

The machine vision system, consisting of 4 cameras, must process and measure the parts of the automotive structure on the production line to check compliance with the prescribed dimensions, considering both the external shape and the internal cut (see Fig. 2).

At the preliminary stage of the task, several questions were posed to create an optimal system, both in terms of hardware and software. Below are the questions posed and the answers to them.

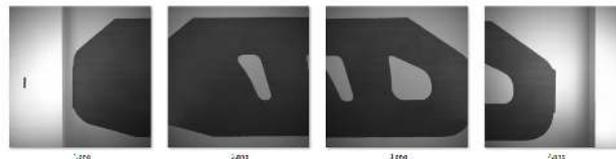


Fig.2. Car parts taken from 4 cameras

A) Is there an ideal template for each part with which the tested part should be compared, or should the vision program be repelled only by factory parameters with a given tolerance?

Answer: There are templates, but not in the form of images, but in the form of schemes, so it will be necessary to match the coordinates of the schemes to the processed images.

B) Is there a document in accordance with GOST, where all the parameters and tolerances for each detail are spelled out?

Answer: The program must be able to enter and read these parameters.

C) How many parts does the nomenclature have, exact or approximate, and can this number change and increase over time?

Answer: There are currently 44 parts to be integrated into the program.

D) Should the program have a training part for each new part, or is the list of parts known and is it possible to initially prescribe / integrate them into the program?

Answer: You can skip the learning phase. The initial requirement is to close the list of available parts, in the future you can revise the list of parts as needed and on the principle of providing an updated version of the program.

E) Should the program be able to combine 4 images received from 4 cameras and save them into one whole, or

is it an optional condition and is offered only for visual viewing?

Answer: The program should connect and save the composited image. To simplify the program, the number of cameras can be reduced taking into account the final field of views of the cameras, so for the available parts, the problem was solved with three cameras.

III. MACHINE VISION PROBLEMS AND SOLUTIONS

A) Organization of uniform illumination over the entire area covered by 4 cameras. An example is shown below [2]. The first shot is bad, it should be like the second one (see Fig. 3).



Fig.3. Reflection of light from a surface, the importance of choosing the right lighting

Since this problem could not be solved on the production line due to multiple metal parts on it, it was decided to formulate the inverse problem by gluing black reference points on the production line. With their help, the program will be able to switch to a binary image using the threshold intensity filter (thresholding), finding these black points and starting from them (see Fig. 4).



Fig.4. Fiducials added for auxiliary purposes for DIP

B) Organization of rigid mounting of cameras at an equal vertical distance and calibration of the viewing angle (pixel-physical ratio).

The problem at the hardware level was solved. Regarding the software part, it should be borne in mind that the measurement of physical quantities was carried out with a certain accuracy, which depended on the characteristics of the cameras.

C) Connecting / gluing adjacent frames (if necessary).

Here you need to divide the task into two parts: horizontal and vertical bonding. Considering that the parts move along the production line, first it is necessary to solve the issue of synchronous video stream collection from three cameras. The ideal solution is to take a signal from the motion sensor on the line and start collecting frames based on this sensor. At the preliminary stage, the system was created without taking into account the data of this sensor, so the collection was carried out in a continuous mode and it was necessary to programmatically match the

frames, relying on the data described in point A (black dots). The software solution may have one-pixel inaccuracy.

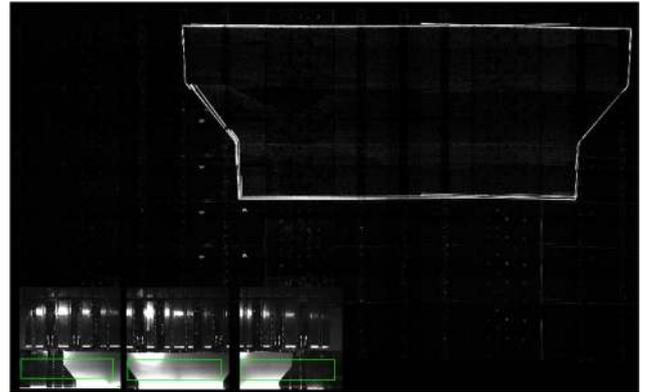


Fig.5. Automotive part processing software and pattern matching

Having solved the issue of horizontal frame matching, the issue of vertical gluing was carried out, taking into account the fixed position of the cameras and the possibility of preliminary calculation of vertical positions (see Fig. 5).

D) Measurement of the shape / dimensions of the tested part and comparison with a template.

As you can see in the picture above, initially the template data was read into the program and after obtaining the complete image of the part (from the top right in the image), this data was compared with the resulting contour of the complete image of the part.

E) Conclusion or saving of the necessary measurements and the decision on the suitability of the part according to the specified criteria (mathematical comparison with the prescribed tolerance).

If the parameters did not match the specified tolerance, an indicator of defects was displayed. Further, the contours of parts with a defect were saved in separate files for further viewing.

IV. PRACTICAL IMPLEMENTATION

The program was written using NI software - LabVIEW, both ready-made algorithms from the NI Vision image processing package [3] were used, and a number of mathematical algorithms were written to match individual frames from three cameras into one large image and other sub-functions [4].

In general, the solution to the problem can be defined as a set of a number of algorithms and subsequent steps that correspond to the classical approach to image processing, such as preliminary filtering, then processing in a spatial domain using algorithms for edge detection and pattern matching [5], and then switching to a binary plane with threshold setting for contour analysis (see Fig. 6) and for analysis of other objects in the scope of interest (Fig. 7).

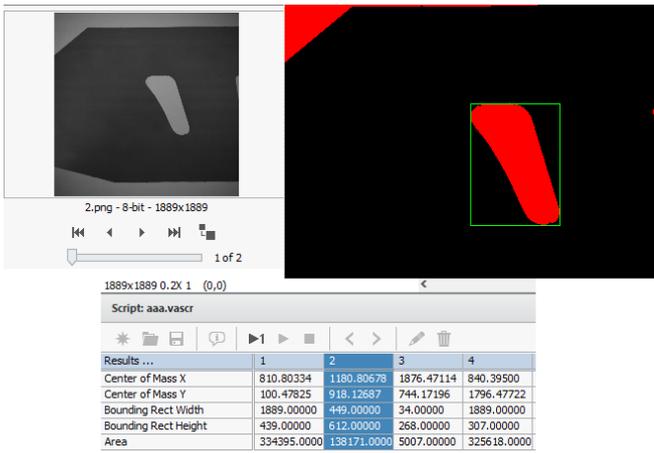


Fig.6. Contour analysis in a binary plane

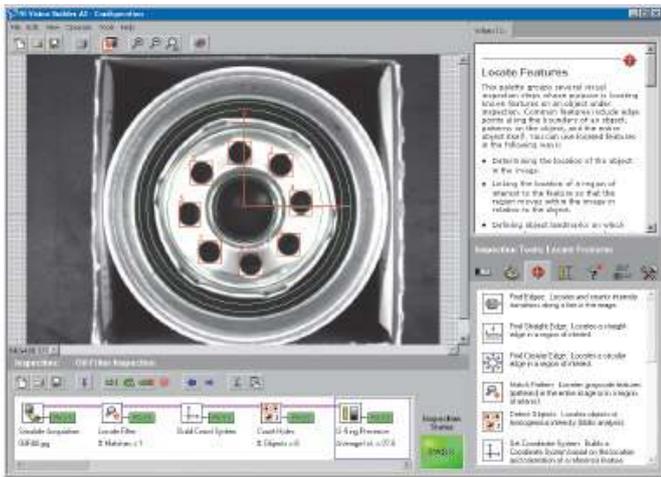


Fig.7. Analysis of contours in the spatial plane

V. CONCLUSION

To solve the problem of automating a machine-building line using machine vision, the appropriate equipment, software and computer technologies of NI were selected, which have ample opportunities for building control and measuring systems of technical vision with subsequent integration into an industrial system. NI provides a combination of powerful software, high-quality drivers, and hardware to help you create custom solutions. By leveraging NI approaches and tools, you can accelerate your development process with an open, software-based platform, you can leverage modular hardware and an extensive system. At the same time, for the development of image processing algorithms and the extraction of information parameters of the image, an application is available called NI Vision Assistant. The implementation of video control systems based on NI solutions allows obtaining stable results in the conditions of uncontrolled image acquisition parameters, therefore the NI software environment is well suited for scientific research of various objects.

As a result, the problem was solved, and the production line was automated with the installation of four cameras, of which three cameras were used in the

software, taking into account the dimensions of the parts (see Fig. 8).



Fig.8. Location of cameras on the production line

The importance and innovation of solving the problem is an integrated approach to the problem, taking into account both hardware solutions and software additions to existing algorithms [6], which ultimately led to the creation of an automated line.

REFERENCES

- [1] National Instruments LabVIEW Development System, www.ni.com.
- [2] K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall Inc, Englewood Cliffs, 1989.
- [3] National Instruments Corporation, "NI Vision Concepts Manual", 2000-2009.
- [4] E.R. Davies, *Computer and Machine Vision: Theory, Algorithms, Practicalities*, Fourth Edition, ELSEVIER, 2012.
- [5] A. Petrosyan, "Infrared image processing for solar cell defect detection", *Proceedings of International Conference Computer Science and Information Technologies*, pp. 300—304, 2017.
- [6] M.K. Bhuyan, *Computer Vision and Image Processing, Fundamentals and Applications*, CRC Press, Published October 27, 2019.