Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 14, No. 1, 2025

Design of Electronic Connector Vision Inspection System Based on VisionPro and C#

Zhelu Wang[1, a], Ruyi Yu[1, b], Yongle Ju[1, c]

[1]Wenzhou Polytechnic, Wenzhou 325035, China
[a]382211678@qq.com, [b]2103129420@qq.com, [c]1027559891@qq.com

Abstract: With the rapid development of the electronic manufacturing industry, the quality inspection of electronic connectors is crucial for ensuring the reliability and performance of electronic devices. This paper proposes a vision-based inspection system for electronic connectors using VisionPro and C#. The system leverages the powerful image processing capabilities of VisionPro and the flexibility of the C# upper computer software to achieve rapid detection of the pins of electronic connectors. Through the research on the overall system design, detection system design, upper computer software development, function debugging, and summary, the effectiveness and practicality of the system have been verified, providing a new solution for the automated inspection of electronic connector pins.

Keywords: Electronic connector; Visual detection; VisionPro; C# programming; Automatic detection.

1. Introduction

In modern electronic manufacturing, electronic connectors are the key components to connect each electronic module, and their quality directly affects the performance and reliability of electronic equipment. Traditional detection methods mainly rely on manual eye detection, which is not only inefficient, but also easy to miss and misdetect. With the development of machine vision technology, visual inspection system has gradually become an important means of industrial automated inspection. The detection of pins in the connector is an important part of product quality detection. The bending and missing of pins will cause short circuit, open circuit and other serious consequences. [1-3]

This paper designs a visual inspection system for electronic connectors based on VisionPro and C#. The powerful image processing and analysis capabilities of VisionPro vision software are used to capture images of connector pins, establish the project flow, script writing and debugging of pins. The upper computer is written in VS2015 C# language, and the joint debugging is carried out by loading ToolBlock files. This paper aims to design a set of efficient and accurate electronic connector visual inspection system combining VisionPro and C#, which can be applied in the electronic connector industry to realize the automatic and high-precision inspection function of the connector. [4-5]

2. Overall System Design

The visual inspection system of electronic connector is composed of light source, lens, camera, computer and actuator. The system software system is programmed by VisionPro, and the interface is developed by VS2015 C#. The main function is to measure defects such as pin bending, offset and missing according to the size and type of connector, and then send instructions to the execution unit for operation based on the matching results of image processing.

In terms of hardware selection of visual inspection system, industrial camera is the core component of machine vision, which essentially converts optical signals into electrical signals. In this paper, Hikvision's 12-megapixel MV-CU120-10GM black and white camera is mainly selected. Optical lens is a dioptric component in machine vision, which is the key to obtain high quality image. This paper selects 12mm fixed focal length lens according to the size of connector. The appropriate light source design can best separate the object information from the background information in the image, this paper selects the ring light source; Finally, the camera is connected to the industrial computer through Ethernet. The industrial computer mainly undertakes the process control, image analysis, data processing and communication of the detection system, which is the core part of the detection system.

3. Vision Software Design Based on VisionPro

3.1. Development environment setup

The design of image processing algorithm in electronic connector vision inspection system is the key. In this paper, the development environment is constructed using the VisionPro software development kit developed by Cognex Company. VisionPro integrates processing algorithms such as positioning, detection, identification and communication. The C# programming language of this system is designed and developed on the basis of VisionPro software package. VisionPro simplifies the integration of vision systems with other master control programs for high-performance vision system development.

3.2. Design of visual image processing flow

In the design of Image processing flow for visual inspection, first open VisionPro QuickBuild software, click Image Source to select image database or camera to capture images, and then create CogToolBlock for pre-processing, which mainly includes the following steps:

(1) CogIPOneImageTool was added for image quantization processing, and then grayscale morphology processing was selected for corrosion and expansion preprocessing;

(2) Add CobBlobTool for spot analysis;

(3) Add caliper tool CogCaliperTool, add calipers on the basis of the previous output image, set the edge mode, and

make the polarity setting from dark to light, and from light to dark;

(4) Script processing, using high-level language writing, and finally ending the process. The specific detection process is shown in Figure 1.

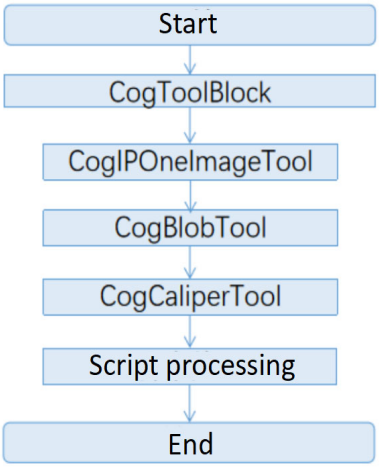

Figure 1. Visual inspection image processing flow

#

3.3. Visual inspection script programming

The VisionPro script is the basic unit of the QuickBuild project. Each project has at least one Job, and the jobs are parallel and do not affect each other. By default, each Job contains a toolGroup where users can add tools and toolblocks. Tool blocks and tool groups are tool "containers" that enable project modularization and tool reuse. Through the "polymorphism" technology, each Job, toolGroup, toolBlock object has an interface object, and users can rewrite the interface method to achieve custom extension function. The script programming of this project mainly uses C# for programming, including pin bending, offset, missing and other problems for programming detection.

(1) Pin bending script programming. First get and edit the results of the VisionPro Blob tool, and then determine whether there is a bend according to the Angle of the spot. The pin bend script determines the procedure as follows:

```
CogBlobTool blob = mToolBlock.Tools["CogBlobTool1"] as CogBlobTool;
bool flag = true;
gc.Clear();
foreach( CogBlobResult b in blob.Results.GetBlobs() ) {
CogRectangleAffine rect = new CogRectangleAffine();
rect = b.GetBoundingBox(CogBlobAxisConstants.Principal); //Minimum external rectangle
//pins bend determination ----------
double angle = Math.Abs(CogMisc.RadToDeg(b.Angle));
if( Math.Abs(angle - 90) > 5 )
{
gc.Add(rect); //Add to graph collection
flag = false;
}
```

(2) Pin offset scripting. First, determine the top/bottom position of the pin according to the Y coordinate of the center of mass, and then determine whether there is deviation according to the minimum external rectangle height is too long/short. The pin deviation script determines the procedure as follows:

```
double h = b.GetMeasure(CogBlobMeasureConstants.BoundingBoxExtremaAngleHeight);
// ImageBoundHeight
if( b.CenterOfMassY < 250 ) {
if( h < 100 || h > 120 )
{ rect.Color = CogColorConstants.Red;
gc.Add(rect); //Add to graph collection
flag = false;
} }
else{
if( h < 50 || h > 65 )
{
rect.Color = CogColorConstants.Magenta;
gc.Add(rect); //Add to graph collection
flag = false;
} }
```

(3) Pin missing script programming. First, each pin is located through the caliper (find edge pairs), the caliper result is sorted in ascending order according to position x, and then according to the leftmost/right pin position x to determine whether there is missing, the third step calculates the spacing between adjacent pins, according to the spacing between adjacent pins to determine whether there is missing in the middle. Pin missing script judgment procedure is as follows:

```
CogCaliperTool c1 = mToolBlock.Tools["CogCaliperTool1"] as CogCaliperTool;
List<CogCaliperResult> cList1 = new List<CogCaliperResult>();
for(int i = 0 ;i < c1.Results.Count ;i++)
{
cList1.Add(c1.Results[i]);
}
cList1.Sort(( m,n) => { return m.PositionX.CompareTo(n.PositionX); });//Sort from smallest to largest in x coordinates
if(cList1[0].PositionX > 50) { //Check whether the left and right ends are abnormal. The left side is used as an example
CogRectangleAffine rect = new CogRectangleAffine();
rect.CenterX = cList1[0].PositionX / 2;
rect.CenterY = cList1[0].PositionY;
rect.SideXLength = cList1[0].PositionX - 10;
rect.SideYLength = 200;
rect.Color = CogColorConstants.Red;
gc.Add(rect);
flag = false;
}
//----------Intermediate pin missing judgment ----------
for(int i = 0 ;i < cList1.Count - 1 ;i++)//calculated interval
{
double dis = cList1[i + 1].PositionX - cList1[i].PositionX;
if(dis > 80)
{
CogRectangleAffine rect = new CogRectangleAffine();
rect.CenterX = (cList1[i + 1].PositionX + cList1[i].PositionX ) / 2;
rect.CenterY = cList1[i].PositionY;
rect.SideXLength = dis - 20;
rect.SideYLength = 200;
rect.Color = CogColorConstants.Red;
gc.Add(rect);
flag = false;
}
}
```

4. Development of Upper Computer Software for Vision System

4.1. Build the upper computer interface of the vision system

To build the upper computer interface in C#, usually use Windows Forms or WPF (Windows Presentation Foundation) to design the graphical user interface (GUI), the main steps are as follows:

(1) First create a new electronic plug-in Visual inspection system Windows Forms in Visual Studio;

(2) Secondly, add controls of the visual inspection system, such as Button, Label, TextBox, image display controls, etc. At the same time, set the size, position, text and other properties of the controls through the property window to meet the needs of the interface layout;

(3) Finally, write visual detection event handlers: Write visual detection event handlers for buttons and other controls, add click buttons to load visual images and run detection, and realize user interaction functions. The C# upper computer software interface is shown in Figure 2.

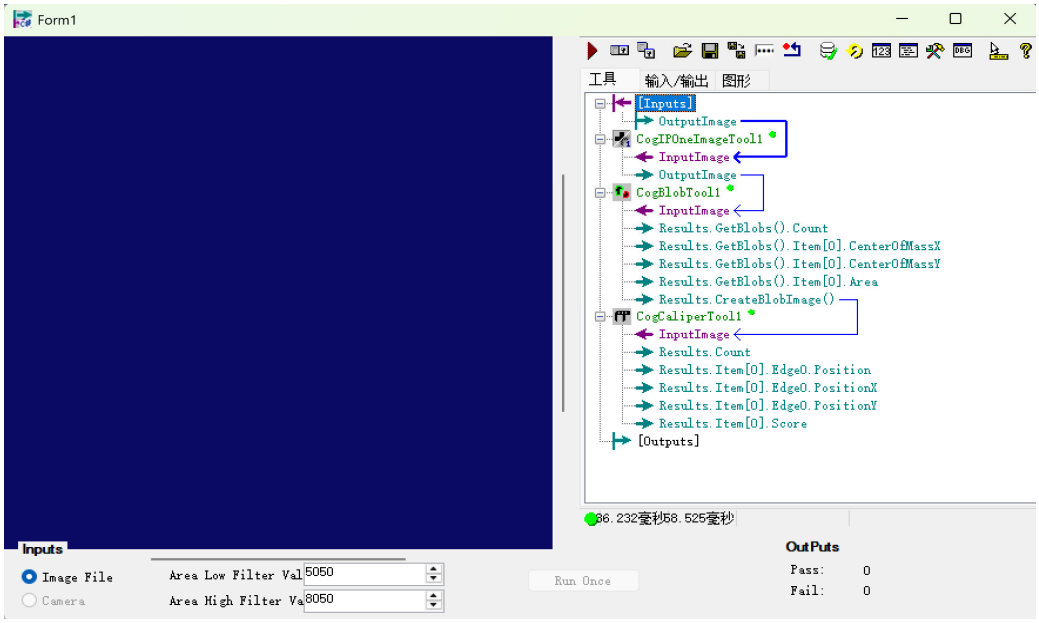

Figure 2. C# upper computer software interface

4.2. C# load ToolBlock File Debugging

In c # programming environment, first loads the VisionPro ToolBlock files and debugging, and then add VisionPro related assembly references, use CogSerializer. LoadObjectFromFile method loading ToolBlock file from the specified path. After setting parameters, call Toolblock.run () method to Run ToolBlock and obtain visual detection results. The detection of pin is shown in Figure 3 and Figure 4.

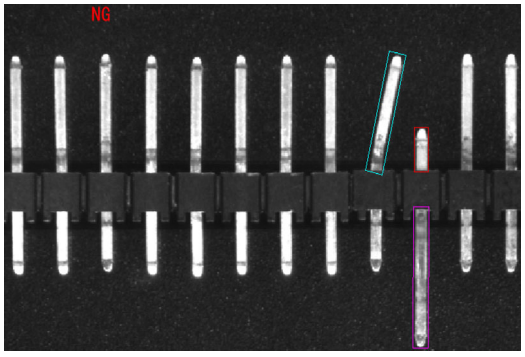

Figure 3. pin bending and offset problem detection

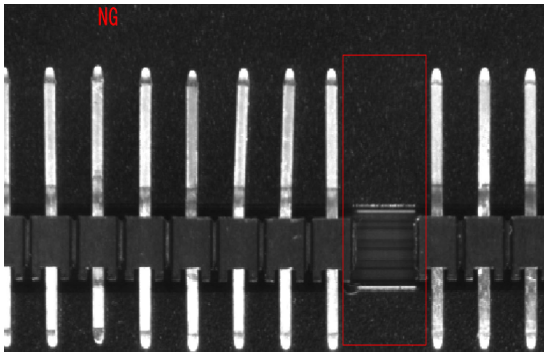

Figure 4. pin missing problem detection

5. Debug System Functions

After completing the writing of the plug-in visual inspection program and C# interface setting, the next step is to enter the system function debugging. First, ensure that hardware equipment such as industrial cameras, light sources, and plug-in detection platforms are properly connected; Load VisionPro's ToolBlock file in a C# program. Load the ToolBlock file with the following code; In C# interface, when clicking the "Start detection" button, whether the program can correctly trigger the detection process and display the result on the interface.

Observe the performance of the program, especially the speed of image processing and detection. By optimizing the algorithm in ToolBlock and adjusting the resolution and frame rate of image acquisition, the pin bending, offset and missing of different types of plug-ins were tested several times to ensure the stability and accuracy of the program. Through the system function debugging, it was verified that the whole system was functional and the recognition rate was

high.

6. Conclusion

In this paper, after obtaining the image of the electronic connector, the connector pin pin is processed by VisionPro. Through C# script docking plug-in pin bending, offset and missing problems are detected and displayed, and determine whether the electronic connector is grid. It has been proved that this detection method can accurately detect whether the connector pin meets the assembly requirements. Compared with the traditional manual inspection, the detection accuracy and detection efficiency of machine vision are greatly improved.

Acknowledgements

This work was supported by the 2023 "School-Enterprise Cooperation Project" of Domestic Visiting Engineers 2023 "School-Enterprise Cooperation Project" (Project No. FG2023033), the 2023 Science and Technology Innovation Activity Plan for College Students in Zhejiang Province (New Talent Plan-Project) No. 2023R452010, the second batch of municipal level "Higher Vocational College Teacher Teaching Innovation Team" construction project - Industrial Robot Technology Integrated Teacher Teaching Innovation Team，the Ministry of Education Supply and Demand Connection Employment and Human Resources Project No.20220100911.

References

[1] Fan Liangyu, Pan Feng. Design of pin positionality detection System for Connector based on vision [J]. Journal of Jiangnan University (Natural Science Edition), 2015,14(06):762-768.

[2] Yuan Fei, Huangshan, Wu Fangfang. Pin Position Detection of Automotive Connector based on Machine Vision [J]. Journal of Wuhan Polytechnic Institute of Technology, 2021, 33(03): 40-43.

[3] Yu Yang, Jin Bin. Miniature electronic connector size measurement based on machine vision [J]. Journal of laboratory research and exploration, 2021, 40 (04) : 13 to 18, DOI: 10.19927 / j.carol carroll nki syyt. 2021.04.004.

[4] Guo Zhanling. Connector pins defect detection based on machine vision research [D]. Sichuan university, 2023. The DOI: 10.27342 /, dc nki. Gscdu. 2023.001479.

[5] Fang Jianlong, Yu Bo. Level measurement connector terminal detection based on machine vision system [J]. Journal of digital technology and application, 2021, 33 (3) 6:127-131. The DOI: 10.19695 / j.carol carroll nki cn12-1369.2021.03.42.