Superflexible Welding Robot Based on the Intelligent Space Concept

Béla Takarics, Péter T. Szemes

Department of Automation and Applied Informatics
Budapest University of Technology and Economics
Goldmann György tér 3, H-1111 Budapest, Hungary
{takarics, szemes}@get.bme.hu

Abstract: Superflexible robot programming can reduce the profitable batch sizes for industrial robots. The paper presents a new way to determine the trajectory for welding robots based on the intelligent space concept. The system uses two cameras and edge detection with other image processing algorithms to find the welding path in the image. It's three dimensional position is obtained by stereo vision and then it is transformed to the robot language.

Keywords: superflexible robot programming, intelligent space, edge detection, stereo vision

1 Introduction

Increasing requirements to productivity and working environment push the need for industrial robots in the industry. This is especially important in heavy physical work were sickness, absence and recruiting problems are often the most limiting factors on the productivity. Today, industrial robots represent the best solution for both productivity and flexibility. Nevertheless, there is still a need for even more flexible robots to satisfy the industrial requirement to shorten the set up times in the production as the robot programming time is approximately 400 times the execution time of the task. Obviously, this puts requirements on a minimum batch size in the production to make it profitable to invest in industrial robots.

The paper is organized as follows: the next chapter describes the superflexible robot programming method and the proposed system design based on it. The third chapter deals with image processing with a special emphasis on edge detection. The following chapter introduces the camera calibration and stereo vision techniques, which will be followed by the results and conclusions.

2 Superflexible Robot Programming Based on the Ubiquitous Computing Concept

In the 1980s industrial robots became more and more popular in the industry. Several applications of such robots existed from welding to grinding and assembly. The batch sizes were very high an industrial robot ran the same motion scenario thousands or millions of times. Later they have lost they importance, because a lower amount of products were needed before the robot had to be reprogrammed, and they were replaced by cheaper production lines. However, the use of industrial robots can be a good solution at low batch sizes. The challenge lies in the programming time overhead of a robot, where there are only few dozens of parts to manufacture, the use of an industrial robot is way too expensive due to long programming time.

The solution is the superflexible robot programming. In this case the desired trajectory of the robot is not programmed in the robot language directly, but it is determined in the world coordinate system by some faster method and then is transformed to the robot language by an interface as seen in the figure below.

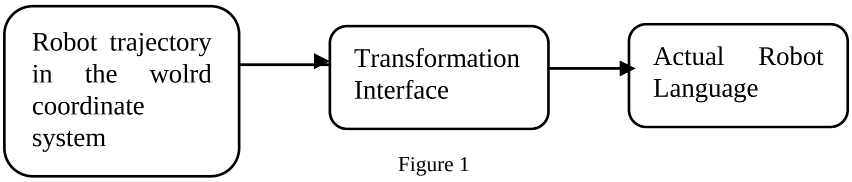
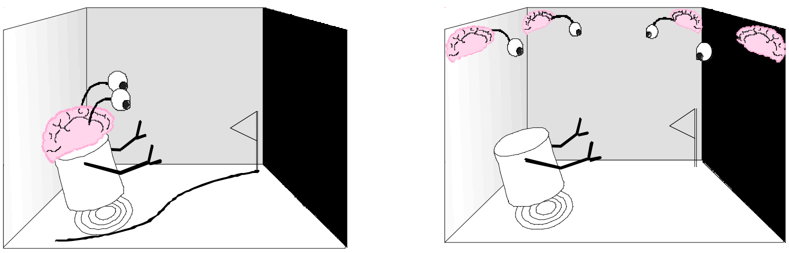


Figure 1
The block diagram of the superflexible robot programming

There are several methods to obtain the robot trajectory. A convenient method is to take advantage of the ubiquitous computing concept, such as the Intelligent Space. Intelligence Space is a space (room, corridor or street), which has ubiquitous computing and sensory intelligence. Conventionally, there is a trend to increase the intelligence of a robot (agent) operating in a limited area. The Intelligent Space concept is the opposite of this trend. The surrounding space has sensors and intelligence instead of the robot (agent). A robot without any sensor or own intelligence can operate in an Intelligent Space [1]. The difference of the conventional and Intelligent Space concept is shown in Figure 2.



Conventional concept　　　　Intelligent Space concept

Figure 2
Difference between the conventional and Intelligent Space concepts

The most widespread method to determine the robot trajectory based on ubiquitous computing concept is proposed by a Norwegian company, Productive Programming Methods As. The proposed concept is called Intuitive Robot Programming (IRP). This concept significantly reduces the programming ime of industrial robots. The IRP concept consists of:

- coordinate tracking device
- software package
- industrial robot

The coordinate tracking device captures the motions of a skilled operator that accomplishes the desired task, the software package which processes the data into a standard robot program to be uploaded directly into the robot controller. The software package also contains 3D visualization, editing functions and application packages. IRP will contain interfaces to the most common robots on the market and is connected to the robot controller via standard interfaces and thus, without any need for adaptations [2].

The aim of this paper is to introduce a superflexible robot programming method for welding robots, which is also based on ubiquitous computing concept. The robot should be able to weld the spoons of the excavator at Caterpillar Hungary Ltd. The company has recruitment problems with skilled welders who are expensive also, so the production is not profitable, which led to the motivation for using welding robots. The batch size is extremely small, only few workpieces fall in a batch, which means that conventional robot programming methods are not economic in this case. The welding path is usually obtained from the CAD model of the workpieces. Unfortunately, in this case the CAD models of the excavator spoons cannot be used as references for welding as the tolerances of the workpieces are high. This means that the particular workpiece will differ from its CAD model to such an extent that the CAD model cannot be used to obtain the welding path. The idea of the solution came from the ubiquitous computing concept. The proposed system has the following components: a vision system, a software and a welding robot. The vision system consists of two cameras and the welding path is obtained by stereo vision. An interface software is needed to transform the welding path from the world coordinate system to the actual robot programming language. This way the whole system is able to discover the welding path in the space and to give the welding trajectory of the robot. If the image processing and the transformation interface is fast enough, the system can be used even for singular workpieces.

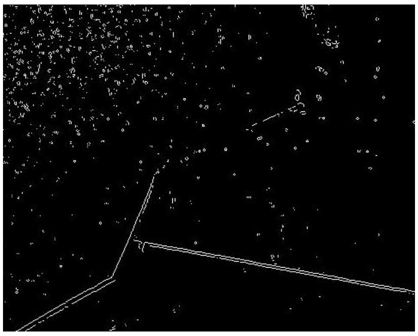
2.1 The Designed System

The above mentioned features of the ubiquitous computing concept make it suitable to use it in industrial environment for superflexible robot programming. Using cameras as sensors, connected to a computer which runs the image processing and stereo vision algorithms form the path tracking device. The computer can run the interface software also and by communicating with the robot, the robot can obtain its trajectory code. If one computer is not fast enough, the algorithms can be distributed, so separate computers are available for the image processing algorithms of the two cameras and for stereo matching. Using the intelligent space concept the welding robot does not need any extra intelligence which reduces it's cost in a large extent.

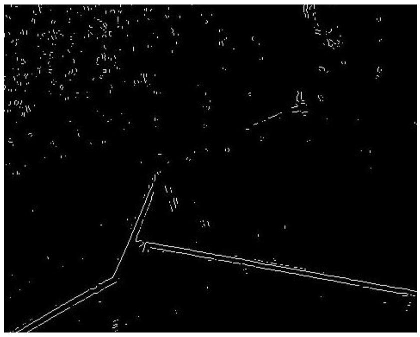
3 Image Processing

Now that we know more about the system, we can concentrate on the image processing. The aim of image processing is to find the welding path on the images obtained from the two cameras. The main image processing algorithm we use is edge detection, but since edge detection itself does not give satisfactory results, other techniques are also needed.

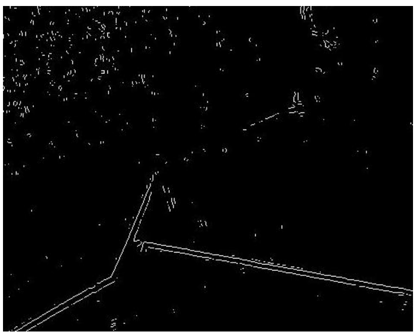
3.1 Edge Detection



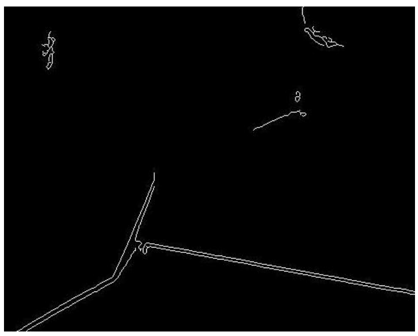
(a) Sobel method



(b) Laplacian of Gaussian method



(c) Zero-cross method



(d) Canny method

Figure 3
Edge detection results

With edge detection everything depends on a pixel and its neighbors. A comparison of that pixel's brightness and its neighbor's brightness will ultimately determine if the pixel is part of an edge or just an ordinary pixel. When a pixel's value is within a certain range of its neighbors it is most likely not part of an edge. When the pixel's value differs from its neighbors and is not within the range it is usually considered to be a part of an edge. The change in pixel value from pixel to its neighbors is usually done in a smooth manner. [3]

A lot of algorithms exist in order to process an edge detection. For instance: Sobel, Prewitt, Roberts, Frei-Chen, Laplacian or Canny. Some of it are already implemented in Matlab, so we could test more of it to see which gives the best results. Figure 3 shows the result of different edge detecting techniques.

Figure 4
Edge detection result after segmenting the image using gray scale color threshold

As we can see in the images, the Canny method gives the best result, but there is still some noise left in the image, which has to be removed! Smoothening the image and applying an edge detection after it, we get the following result seen in Figure 4, where the result was emphasized with a thicker line for better visibility.

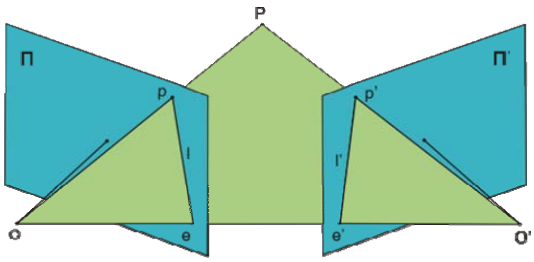


Figure 5
Epipolar geometry

Figure 5 shows that there is no more noise left in the image. There is only one more step remaining to detect the welding path in the image. We have to choose 10-20 active (white) pixels which are approximately at equal distance from each other. These points will be the basic points of the cubic spline that is fit to the welding curve. Running these steps on the particular images, the welding curve is obtained in the images. The next step is to calculate a three dimensional curve which can be transformed to the actual robot programming language. This step requires stereo vision which is described in the following chapter.

4 Camera Calibration and Stereo Vision

The simplest and one of the most versatile camera calibration techniques today is the Roger Tsai's method. Tsai's camera model is based on the pinhole model of perspective projection. Given the position of a point in 3D world coordinates the model predicts the position of the point's image in 2D pixel coordinates. More about this technique can be found in [4]. There is open source software for camera calibration, namely The Tsai Camera Calibration Software, which can be downloaded from [5]. This software package contains routines for calibrating Roger Tsai's perspective projection camera model.

4.1 Stereo Vision

Despite the wealth of information contained in a photograph, the depth of a scene point along the corresponding projection ray is not directly accessible in a single image. With at least two pictures, on the other hand, depth can be measured through triangulation.

Stereo vision involves two processes: the binocular fusion of feature observed by the two eyes, also called *stereo matching*, and the *reconstruction* of their three-dimensional preimage. The latter is relatively simple: the preimage of matching points can be found at the intersection of rays passing through these points and the associated pupil (camera) centers. Thus, when a single image feature is observed at any given time, stereo vision is easy. However, each picture consists of hundreds of thousands of pixels, with tens of thousands of image features such as edge elements, and some method must be devised to establish the correct correspondences and avoid erroneous depth measurements [6].

4.2 Stereo Matching - Epipolar Geometry

Consider the images *p* and *p'* of a point *P* observed by two cameras with optical centers *O* and *O'*. These five points all belong to the *epipolar plane* defined by the two intersecting rays *OP* and *OP'* (Figure 5). In particular, the point *p'* lies on the line *l'* where this plane and the retina **Π'** of the second camera intersect. The line *l'* is the *epipolar line* associated with the point *p*, and it passes through the point *e'* where the baseline joining the optical center *O* and *O'* intersects **Π'**. Likewise, the point *p* lies on the epipolar line *l* associated with the point *p'*, and this is line passes through the intersection *e* of the *baseline* with the plane **Π**.

Points *e* and *e'* are called *epipoles* of the two cameras. The epipole *e'* is the (virtual) image of the optical center *O* of the first camera in the image observed by the second camera, and vice versa. As noted before, if *p* and *p'* are images of the same point, then *p'* must lie on the epipolar line associated with *p*. This *epipolar constraint* plays a fundamental role in stereo vision and motion analysis [6].

The most difficult part of stereo data analysis is establishing correspondences between the two images (i.e. deciding which points in the second picture match the points in the first one). The epipolar constraint greatly limits the search for the correspondences: Indeed, since we assume that the rig is calibrated, the coordinates of the point *p* completely determine the ray joining *O* and *p*, and thus the associated epipolar plane *OO'p* and epipolar line *l'*. The search for matches can be restricted to this line instead of the whole image as shown in Figure 6.

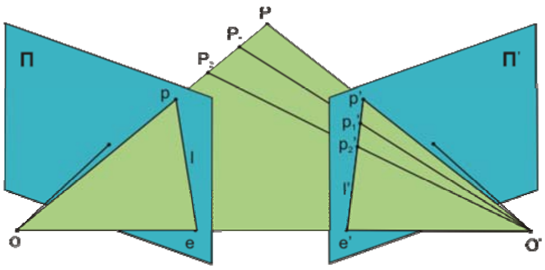


Figure 6
Epipolar constraint

The epipolar constraint and the epipolar line plays a fundamental role in our case as we have only one curve left in the image and that is the welding curve. Our aim is to choose 10-20 points of a spline of one image and to calculate their 3D position. The purpose of the stereo matching process is to find the corresponding points in the second image. As they lie on the spline of the second image, the corresponding point pairs can be determined as the intersection of the epipolar line and the spline (Figure 7). After we determined the corresponding point pairs we can start their 3D reconstruction.

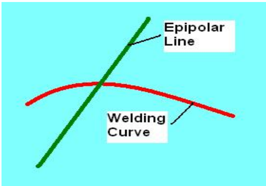


Figure 7
Stereo matching in our case

4.3 3D Reconstruction

After the camera is calibrated and its intrinsic and extrinsic parameters are known, the three-dimensional coordinates of the hand can be calculated from the two-dimensional data. The 3D reconstruction is done in the following steps.

From the definition of Tsai camera calibration [7], the captured point (X_f, Y_f) in frame coordinate system is,

$$\begin{cases} X_f = \dfrac{s_x}{d_x} \cdot X_d + C_x \\ Y_f = \dfrac{1}{d_y} \cdot Y_d + C_y \end{cases} . \tag{1}$$

Where,
(X_f, Y_f): point in pixel on the captured picture,
(C_x, C_y): center point of the captured image,
(X_d, Y_d): actual image position.

Expressing the actual image position from (1),

$$\begin{cases} X_d = (X_f - C_x) \cdot \dfrac{d_x}{s_x} \\ Y_d = (Y_f - C_y) \cdot d_y \end{cases} . \tag{2}$$

The perfect pinhole camera maps world coordinate (x, y, z) into actual image coordinate,

$$\begin{cases} X_u = f\,\dfrac{x}{z}, \\ Y_u = f\,\dfrac{y}{z}. \end{cases} \tag{3}$$

To convert real world reference coordinate (x_w, y_w, z_w), using Rotation Matrix **R** and translation vector **t**:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \mathbf{t} \tag{4}$$

Using (3) and (4), a linear inhomogeneous equation system can be deduced with three variables. With the solution of the equation system, the values of (x_w, y_w, z_w) are determined. There is only one step following the 3D reconstruction and that is to fit a spline on these points and the 3D welding curve in the world coordinate system is determined.

5 Results

The final chapter of this paper deals with the results we achieved. There are more aspects of the results of this project which should be discussed here which will follow the structure of the paper.

First of all, we have to mention the speed of the whole process, as it should be high enough to make the system applicable for small batch sizes. The algorithm runs only in Matlab at the design stage and to analysis two 2272x1704 image takes approximately 2 minutes. We can say that it is fast enough but program will run in C or C++ at the final stage and a great speed increase is expected then.

The second issue is the accuracy of the welding curve recognition in the images. This process is extremely accurate with only a few pixel errors as it can be seen in Figure 8. The test was done in the laboratory with ideal light conditions and background, but the industrial environment can be also modified for our needs.

The next criteria that has to be satisfied is that the two ends of the splines should be the end points of the welding curve. Figure 8 shows that this condition is also met.

The last item which have to be mentioned is the 3D reconstruction accuracy. The results are within 10 millimeters of tolerance. This is not extremely accurate, but these are not the final results, because only ordinary digital cameras were used for the test. We will get the industrial cameras in the near future and we expect to get satisfactory 3D reconstruction results then.

All in all we can conclude that the results confirm what we expected and the system will be able to weld excavator spoons automatically at its final stage.

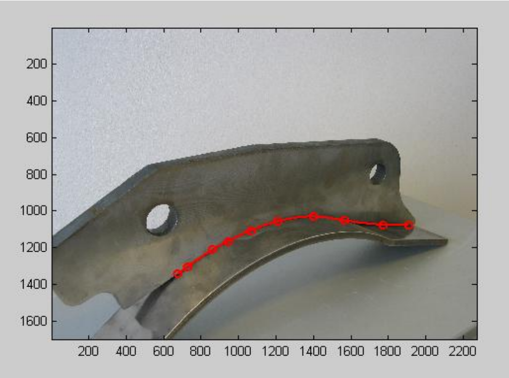


Figure 8
The result of the image processing techniques

Conclusions

The introduced system, which integrates different techniques to achieve the final result was deigned and implemented at the Budapest University of Technology and Economics and GI-Flex Ltd. The whole system is realized as two DINDs, known from the Intelligent Space. The welding path recognition in the two images and the three dimensional reconstruction algorithms were designed and implemented into the system. The algorithms are written in Matlab, but when they reach they final stage, they will be compiled to C or C++ programming languages, which are much faster. As we could see in the previous chapter, the results satisfy our needs, the next step will be to create the interface which converts the three dimensional path to the particular robot language.

Acknowledgment

The authors wish to thank the JSPS Fellowship program, National Science Research Fund (OTKA T046240), E!3168 EUREKA 'SUPERFLEXROBOT' project, Déry Miksa project (Alap2-0024/2005) and Control Research Group and János Bolyai Research Scholarship of Hungarian Academy of Science for their financial support and the support stemming from the Japanese-Hungarian Intergovernmental S & T Cooperation Programme.

References

[1] P. Korondi, H. Hashimoto: *Intelligent Space as an Integrated Intelligent System.* Electrical Drives And Power Electronics International Conference, Slovakia, September 24-26, 2003

[2] A. Gaudia, B. Reskó, P. Korondi, T. Thomessen: *Robot Programming Based on Ubiquitous Sensory Intelligence, INES 2004,* 8th IEEE International Conference on Intelligent Engineering Systems, Sept. 19-21, 2004, Cluj-Napoca, Romania

[3] Canny, J.: *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986

[4] R. Y. Tsai: *A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the shelf TV Cameras and Lenses.* IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, August 1987

[5] http://www.cs.cmu.edu/afs/cs.cmu.edu/user/rgw/www/TsaiCode.html

[6] J. Ponce, D. A Forsyth: *Computer Vision: A Modern Approach.* Prentice Hall, 2002

[7] P. Szemes: *Human Observation-based Motion Control Strategies.* Doctoral Thesis, Tokyo, 2005