Multi-robot Assembly of IC's

C. Michaud, A.S. Malowany and M.D. Levine

Computer Vision and Robotics Laboratory
Department of Electrical Engineering
McGill University
Montréal, Québec, Canada

Abstract

This paper presents a multi-robot system which assembles integrated circuits. The problems addressed are the calibration of the robotic station, the control of several robots working concurrently, and pattern recognition. These problems are discussed in the context of a distributed system.

The use of image processing is omnipresent. For instance, vision feedback is used to acknowledge almost every move of the robots and also to calibrate the camera itself. Pattern recognition techniques are employed to detect the exact orientation of an IC die.

Several robots are used in order to give more flexibility to the system and allow us to acquire experience which could lead toward a multirobot multitasking system.

keywords: Multirobot, Flexible Manufacturing Systems, Integrated Circuit assembly, Pattern Recognition, Vision Feedback, Error Recovery

Résumé

Cet article présente un système assemblant des circuits integrés avec plusieurs robots. Les problèmes adressés sont: la calibration de la station robotique, le contrôle de plusieurs robots oeuvrant à differentes tâches simultanément et la reconnaissance d'images.

L'utilisation de la vision par ordinateur est omniprésente dans le système. Premièrement, elle permet de vérifier l'exactitude de presque tous les mouvements des robots. Deuxièmement, elle est nécessaire pour la calibration de la caméra. Finalement, elle est indispensable afin de procéder à la reconnaissance d'image. Cette dernière est utilisée dans le but d'obtenir les positions et orientations exactes des puces électroniques.

Plusieurs robots sont utilisées dans le but de rendre le système plus flexible et surtout pour nous permettre d'acquérir l'expérience nécessaire à la conception de systèmes robotisés "multi-tâches" "multi-robots"

Mots Clé: Multi-robots, Systemes de Production Flexible, Assemblage de Circuits Intégrés, Reconaissance d'Images, Recouvrement d'Erreurs

1. Introduction

Until recently, few attempts have been made to create multirobot cooperative systems which have practical application in industry. The designers of such systems have to take into account factors such as ease of programming, task parallelism, error recovery, collision avoidance and so on. F.Ruoff[1] has proposed such a system but every robot must be programmable with the same programming language. Unfortunately, industrial robots usually come with their own programming language and generally, robot vendors offer little support to those who want to implement their own language. As a result, few customers implement their own because it is expensive, takes a lot of time, and includes risk of damaging the robot. Here, we will discuss the implementation of a flexible IC assembly station which deals with some of these problems. The proposed station does not require any special robot programming language. Paler et al.[2] have already implemented a system that assembles IC's. However, they used one robot, which involved a single task. Therefore, they were not concerned about synchronization between robots. Here, the assembly of IC's is used as a means to work in a multi-robot environment. This paper is divided in five sections. First, a description of the robotic environment is given, which is followed by a functional description of the tasks to be done. Thereafter, the important features of the system are discussed. This includes the start-up procedure, task synchronization and vision feedback.We conclude with some ways to improve the system.

The start-up procedure discusses efficient ways to model the environment and to calibrate the cameras The modeling is based on *basic guiding*[3] and the camera calibration is done by inferring the position of the camera and correcting its position by evaluating a proper correction offset. This technique has been described by Mansouri[4].

Task synchronization is done by means of message passing[5][6][10]. Processes are synchronized with each other by means of a *"rendez-vous"* scheme. The vision feedback will mostly be used for locating the position and the orientation of objects. It requires the use of template matching and the local features method[7]. Finally, the implementation of forward recovery techniques[8] and a simple scheme for collision avoidance based on the method of Freund and Hoyer[9] are intended.

2. Description Of The Robotic Environment

The McGill University Computer Vision and Robotics Laboratory is based on three VAX minicomputers and a wide variety of peripheral equipment for research. A VAX 11/780 running VAX/VMS and Eunice, a Unix emulator, is linked with two VAX 11/750 minicomputers running UNIX 4.2BSD, by the Ethernet local area network. The VAX 11/780 is used for the AI oriented programs which control the overall station. The two VAX 11/750s are used for the robots, the control of peripheral equipment and the vision processing. A Matrox frame grabber system which can handle up to 4 different video inputs, is linked to the vision VAX 11/750. The system is configured as a distributed system shown in Figure 1.

There are two robots available in the laboratory: A PUMA 260 from Unimation and a CASTOR from Microbo. Each robot has its own command language, VAL for the PUMA 260 and IRL for the CASTOR. Fortunately, software packages have been written in the C programing language to emulate these two robot languages. Basically, these programs convey commands as remote terminals to the robots by way of RS-232 links. This greatly simplifies the control of the robots by allowing them to be accessed by a single program. There are advantages and disadvantages to do this. The main advantage is that no special language is required to use both robots at their full capabilities. This saves a great deal of time at the implementation level and seems to be one of the easiest ways to use several robots efficiently. However, the user is limited by the power of the emulation package available as well as the vendor's programming language. For instance, the CASTOR's controller offers an easy way to do concurrent processing because every time a command is received, the CASTOR's controller returns an acknowledgement immediately, which is not the case for the PUMA's controller. It is also possible to change the robot trajectory while it is moving by sending a new trajectory command before the previous one is completed. On the other hand, the IRL command language does not have any facilities for making the robot move along a straight line. The only available motion is joint interpolated.

A variety of automatically interchangeable end effectors have been built to work with the PUMA 260 robot. They will soon be available for the CASTOR robot as well. Meanwhile, a specialized multitool has been built for the Microbo. These exchangeable tooling facilities provide great flexibility to the present robotic station. The robots can perform a wide range of complex tasks without any human intervention.

In addition, a set of high resolution CCD cameras and a 50x microscope, with remote focus and zoom, are available for the vision task. Also, an XY stage can achieve extremely fine positioning under the microscope. Finally, several kinds of feeders have been built to fulfill the different needs of the laboratory.

The minimum set-up requirements for the present project includes:

- Two robots to perform the assembly task.
- IC feeder.
- Dice array carrier to hold the dice array for assembly.
- Fixture for holding the dice array during the operation.
- Vacuum end effector to pick up the 40 pin IC dice carrier.
- Vacuum end effector to pick up a 3mm x 3mm die.
- Syringe to dispense the soldering paste on the IC dice carrier.
- End effector to pick up the dice array.
- Set of CCD cameras and a microscope with a frame grabber system.
- XY stage.

The Matrox frame grabber generates images of 512 x 512 pixels with 128 grey levels. However, only 256 x 256 pixels of resolution and 64 grey levels are used. This has been chosen for two reasons. First, it greatly reduces the amount of processing required per image which speeds up the processing time by several orders of magnitude. Second, knowing that the welding pads are $\approx$ 10 pixels square when full magnification is used, this 256 x 256 image resolution allows locating their centers with a possible precision of 10 percent, which is acceptable. Because of its higher repeatability, the CASTOR robot is used to handle the dice. Its repeatibility is about +/- 10 microns. The repetability of the PUMA 260 is only 40 microns, so it is used for the less demanding tasks. The XY stage used for positioning under the microscope has an accuracy of 5 microns.

The multi-tool used by the CASTOR robot consists of a small vacuum tube that can accurately take a die and a syringe that is used to put the solder paste on the IC dice carrier. Presently, this microbo tool is not yet automatically interchangeable, which constrains the use of the CASTOR to these two tasks only. On the other hand, the PUMA 260 uses two interchangeable tools. The first one is a vacuum tube built to pick up 40 pin IC dice carrier. The second tool consists of two fingers for picking up a dice array by holding it by the border.

3. Functional Description

The overall process consists of putting dice on a IC dice carrier on which solder paste has been applied. This will be done in several steps involving many concurrent operations. Consequently, the functional description put stress on this.

The operation begins with the calibration of the station. Then, the PUMA 260 robot takes a dice carrier and puts it on the XY stage. Meanwhile, the CASTOR robot has moved close to the microscope and is ready to put on the paste. As soon as the PUMA 260 has finished its move, it goes to get the dice array and puts it in the fixture close to the CASTOR robot while the latter is putting on the paste. Then, the CASTOR goes to take a die and puts it on the XY stage. Thereafter, the XY stage moves the die under the microscope. When it is in position, a picture of the die is taken and the XY stage moves immediately back to where it was and the vision VAX 11/750 computes the exact position of the center of the die. When the result is known and the XY stage is in position, the CASTOR picks up the die at the center and moves it over the dice carrier. By that time, the orientation of the die is known, which enables the CASTOR to put it in place. Meanwhile, the PUMA 260 has replaced the dice array by the next required kind of dice. This interleaved sequence is shown in Figure 2.

The reader may have noticed that sometimes during the process one robot is idle, doing nothing! Our goal would be to make the robotic station work with an efficiency close to 100 %. This problem has not been fully resolved. A great deal of research and experimentation are in progress involving image processing algorithms, networking and operating systems

4. Start-Up

We feel that the station start-up procedure is very important. In order to do this nicely, special routines have been written to ease the world modeling and the calibration of the camera. The world modeling of the station is done by basic guiding, which means that one can teach the desired position using the robot teach pendant. Both robots can be used for the modeling of the environment. The routines written allow the user to modify part or all of the environment in the database. When a modification is done or a new element is introduced, the database is immediately updated. Furthermore, a graphic routines package will be added to help the user during the modeling. It will guide the user by displaying the suggested sequence of points for teaching any particular object in the robotic station. For instance, if someone wants to teach the position and orientation of an object X, the routines will display the particular robot to be used for the teaching of this particular object and the object itself.

The camera calibration is done automatically. The only requirement is that the robot program must know the approximate location of the camera. This can be provided during the basic guiding. During the camera calibration process, the robot is brought under the camera and the equations relating the camera and the robot coordinate systems are evaluated.

The calibration of the XY stage is also done automatically with the use of the camera. It is done by precisely locating the position of a marker on the XY stage.

5. Task Synchronization

Anyone working in a distributed environment must deal with the synchronization of tasks running in parallel on different computers. This is being done by way of message passing. Every time a process has to be synchronized with another, one of the processes acts as message sender and the other one as message receiver. When the sender process has reached its synchronization point, it sends a message to the other process. It will not go further until a proper acknowledgement message is sent back. In the case of the receiver process, it will run until it reaches the "rendez-vous" point where it expects a message. If there is no message, it waits for the message unless a watchdog procedure stops it. In this latter case, a recovery procedure is fired. Otherwise, it reads the message and sends an acknowledgement.

Every process involving direct object movement runs on the same VAX 11/750. This was done because if a sensor detects a possible collision, it can stop the robots immediately with a minimum of delay.

6. Vision Feedback

Extensive use of vision feedback is made starting with the calibration of the station until the termination of a working session. Every time the robots move objects, vision feedback is used to detect any malfunctioning. However, most of the vision processing will be used for the detection of the position and orientation of the IC dice.

It is very important to detect the position of the die accurately. We propose to do it in two steps. First, an approximative evaluation of the position of the die is done. Then, using this knowledge, the program tries to match known local features from a database to the present object. When this is done, the knowledge of the exact position of these features can be used to obtain the precise position of the die.

The finding of the aproximate position is done as follows. First, a predefined grey level threshold is applied to the picture of the dice. Then, thinned edges are obtained by running a contour tracing program which stores the contours in chain codes. The chain codes are noise filtered by applying a filter that retains only the relatively long straight edges. Thereafter, the shape of the desired dice is fitted to the exterior of the residual edges. The quality of the fitting is critical.

The detection of the orientation is done using local feature extraction. In order to do so, the program uses the database containing available features of the dice and their relative positions. Since these features can be matched whithin a few passes and involve only small part of the picture, the process can be very fast. Having accurately located the IC die, other operations would be possible. For example, getting the position of the welding pad.

7. System Reliability

In order to improve the reliability of the system, forward recovery will be extensively applied. Forward recovery implies that every time an error is detected, the program tries to recover without having to restart from a predetermined previous position. It also does not require knowledge of the state of the program before it failed.

In addition, the implementation of a collision avoidance program is being attempted. However, since the topic of collision avoidance is still under active research, we have decided to apply Freund and Hoyer's approach[9]. He considered the problem in the XY plane where all the robots are described an translational and rotational joints. It turns out that most robots can be described that way. This approach, based on decision rules and tables, leads toward a relatively simple algorithm to plan the robot's trajectory. This method is much more efficient than freezing one robot when another is in the common working area, and will significantly increase the performance of the system by eliminating most of the robot's waiting.

8. Conclusion

This system will provide us with some first hand experience with multiple robots. It will be used as a basis for implementing a multirobot multitasking system for repairing IC and hybrid circuits. In the future, other repair and assembly tasks with concurrency will be added. This will ensure a better percentage of robot use and will provide much flexibility to the user.

The required tooling and programming development for the assembly of hybrid circuits and IC dies are presently in progress.

Acknowledgements

The financial support of NSERC and FCAC are gratefully acknowledged. Martin Levine would like to acknowledge the support of the Canadian Institute for Advanced Research.

References

[1] Ruoff F., "Multitasking Robot System", Industrial Robot, June 1980

[2] K. Paler et al., "Automatic Packaging of Integrated Circuits", Proc. of 1983, RoViSec 3. SPIE Vol 449, Cambridge Massachusetts, 1983

[3] Lozano-Perez, "Robot Programming", Proceeding of IEEE, Vol 71(7), July 1983

[4] A.R. Mansouri and A. Malowany, "Using Vision Feedback in Printed Circuit Board Assembly", Proc. of the 1985 IEEE Microprocessors Forum Atlantic City,NJ, 1985

[5] Manning and Peebles, "A Homogeneous Network for Data-Sharing Communications", Computer Networks, Vol 1, (2), 1976

[6] C.A.R. Hoare, "Communicating Sequential Processes", CACM, Vol 21, August 1978

[7] Robert C. Bolles and Ronald A. Cain, "Recognizing and Locating Partially Visible Objects:The Local-Features-Focus Method", Int. J. of Robotics Research, Vol 1,(3), Fall 1982

[8] T.Anderson & D.A. Lee, *Fault Tolerant Principle and Practice*, Prentice Hall International, 1981

[9] E.Freund and H.Hoyer, "Collision Avoidance for Industrial Robots with Arbitrary Motion", Journal of Robotics System, Vol. 1(4), June 1984

[10] D. Gauthier, G. Carayannis, P.Freedman, A.Malowany "A Session Layer Design for the CVaRL Local Area Network",Technical Report Tr-85-7R, Computer Vision And Robotic Lab., Dept. of Elec. Eng. McGill University, February 1985

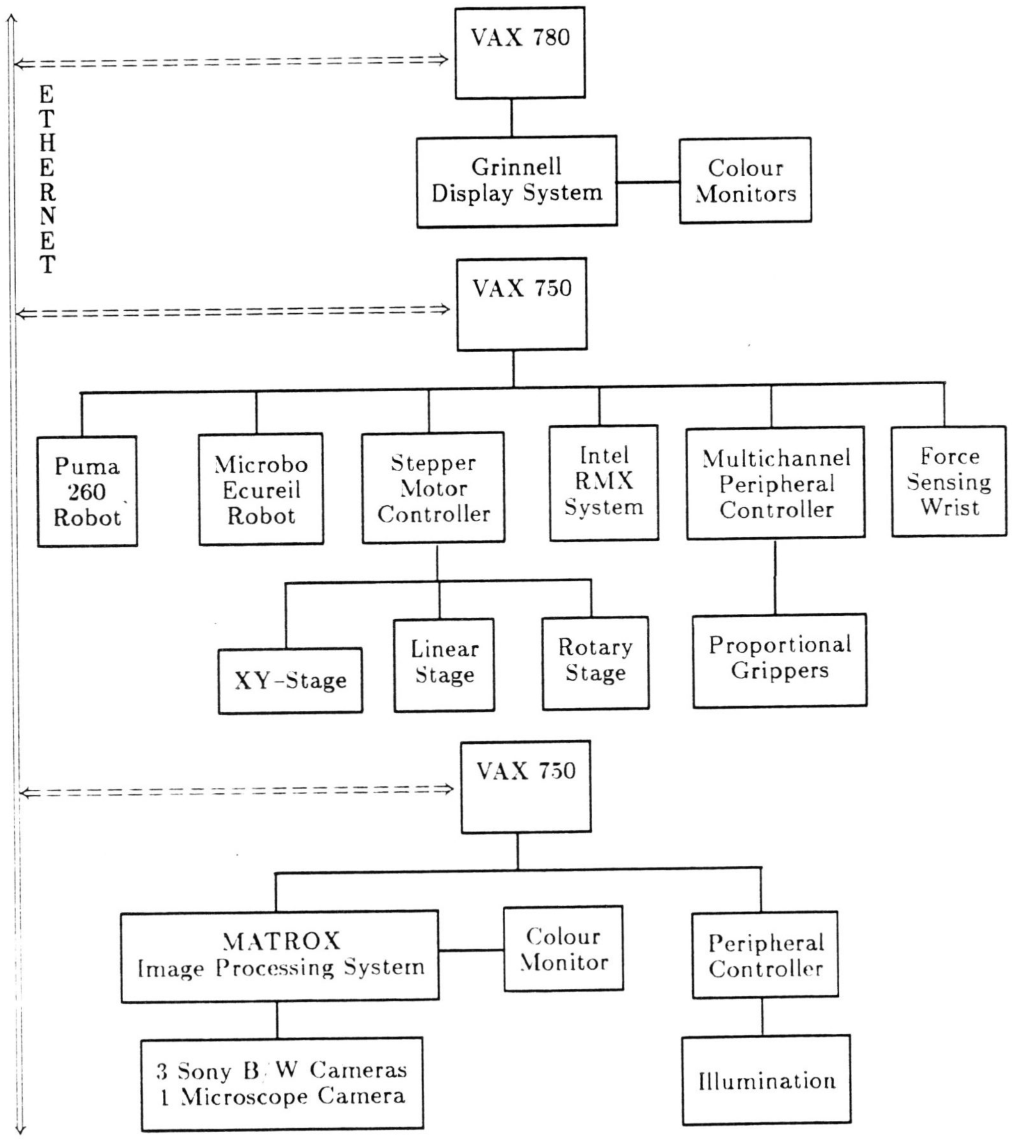

Figure 1 The CVaRL research environment

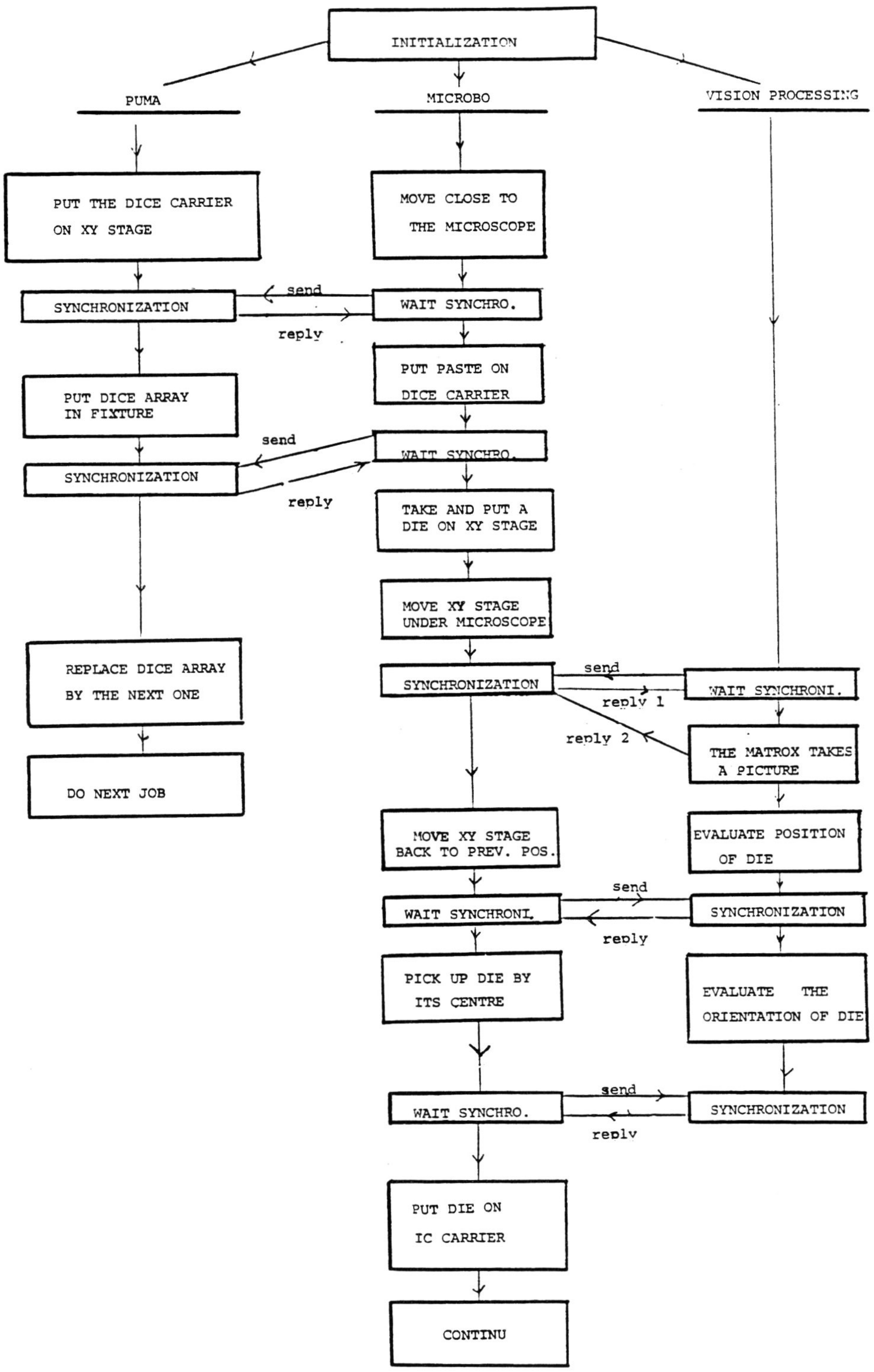
INITIALIZATION
PUMA
MICROBO
VISION PROCESSING
PUT THE DICE CARRIER ON XY STAGE
MOVE CLOSE TO THE MICROSCOPE
SYNCHRONIZATION
send
reply
WAIT SYNCHRO.
PUT PASTE ON DICE CARRIER
PUT DICE ARRAY IN FIXTURE
send
WAIT SYNCHRO.
SYNCHRONIZATION
reply
TAKE AND PUT A DIE ON XY STAGE
MOVE XY STAGE UNDER MICROSCOPE
REPLACE DICE ARRAY BY THE NEXT ONE
SYNCHRONIZATION
send
reply 1
WAIT SYNCHRONI.
reply 2
THE MATROX TAKES A PICTURE
DO NEXT JOB
MOVE XY STAGE BACK TO PREV. POS.
EVALUATE POSITION OF DIE
WAIT SYNCHRONI.
send
reply
SYNCHRONIZATION
PICK UP DIE BY ITS CENTRE
EVALUATE THE ORIENTATION OF DIE
WAIT SYNCHRO.
send
reply
SYNCHRONIZATION
PUT DIE ON IC CARRIER
CONTINU

Figure 2: Process Flow Chart