Vision feedback for SMD placement using neural networks

Gilles Burel, Frank Bernard, Wim J. Venema

▶ To cite this version:

Gilles Burel, Frank Bernard, Wim J. Venema. Vision feedback for SMD placement using neural networks. IEEE International Conference on Robotics and Automation, May 1995, Nagoya, Japan. pp.1491-1496, 10.1109/ROBOT.1995.525486 . hal-03221756

HAL Id: hal-03221756
https://hal.univ-brest.fr/hal-03221756

Submitted on 17 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

IEEE Int. Conf. on Robotics and Automation, Vol. 2, pp. 1491-1496, May 21-27th, 1995, Nagoya, Japan

Vision Feedback for SMD Placement using Neural Networks

G. Burel$^{(*)}$, F. Bernard$^{(\dagger)}$ & W.J. Venema$^{(\ddagger)}$

$^{(*)}$ Thomson CSF/LER, Av. de Belle Fontaine, 35510 Cesson-Sévigné, France
$^{(\dagger)}$ Philips/LEP, 22 av. Descartes, BP 15, 94453 Limeil-Brévannes, France
$^{(\ddagger)}$ Philips I&E, Building BC-p, P.O. Box 218, 5600 MD Eindhoven, The Netherlands

***Abstract*—**
The paper[1] describes an approach to include vision control and neural networks in the assembly process of Surface Mounted Devices (SMD). The need of vision feedback is due to the decreasing size of SMD. The proposed approach includes image processing, estimation of positioning error by a neural network, and geometrical computations. The hardware implementation to achieve real time constraints is also described.

***Keywords*— Industrial Vision, SMD Placement, Control by Vision, Neural networks.**

I. Introduction

Surface Mounted Devices or SMDs are electronic components designed for mounting on a printed circuit board surface. As no component leads have to be inserted through holes in the printed circuit board (PCB) during assembly, SMDs are well suited for high speed, reliable placement and for miniaturisation, while manufacturing costs are reduced.

As the use of SMD components increases, more component types become available as SMD, leading to a large component variety. Miniaturisation puts high demands on SMD placement accuracy. As a rule, the placement accuracy must be 1/6 lead pitch to prevent soldering problems. With lead pitches reduced to 600 microns for the newest SMDs, placement accuracy must be better than 50 microns.

The SMD assembly process can be divided in three steps [5]: Picking (the component is picked with a pipette), Alignment (of the component relative to the board), and Placement. Here, we will deal with the problem of alignment of the component with respect to the board. More precisely, our objective is to measure the relative positioning error between pins and footprints, through the use of vision and neural networks, in order to provide a correction signal to the mechanical system. The images are taken when the component is above the PCB, just before it touches it.

SMD components show smaller and smaller pins and narrower spaces between pins, so that even a very precise mechanical placement is not accurate enough. Furthermore, the Printed Circuit Board (PCB) can be slightly dilated by small changes of external temperature. So, only the relative position of pins and footprints is relevant, and, hence, use of vision feedback is required.

In this paper, we describe a look-while-place approach which takes profit of learning capabilities of neural networks. This work has been done inside the Galatea project (Esprit project n^o 5293), which aims at promoting the application of neural networks by European industries. The vision workpackage is in charge of developing neural solutions to industrial vision problems.

The paper is organized as follows: in section 2, we present the context (placement machines, SMD, and requirements). In section 3, we describe the proposed algorithm to control by vision the alignment of the component relative to the board. Finally, the hardware implementation is detailed in section 4.

II. Need of vision feedback in placement machines

The placement cycle of a current SMD mounter, as made by Philips IE/EMT (figure 1) is as follows:

- The printed circuit board is fed in the machine

[1] IEEE International Conference on Robotics and Automation, May 21-27th, 1995, Nagoya, Japan

©1995 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

- Due to mechanical tolerances, the location of the solder pads with respect to the machine is only approximately known. The exact position of the PCB solder pads is determined by a vision system on the machine that measures the position of alignment markers (fiducials) on the PCB.
- A component is picked up by a gripper from one of the machine feeders.
- A vision system is used to measure the position of the component with respect to the gripper.
- The movement the gripper has to make to place the component leads on the PCB solder pads is calculated, using the PCB CAD data that gives the position of the PCB solder pads with respect to the fiducials, the measured position of the fiducials w.r.t. the machine, and the measured position of the component w.r.t. the gripper.
- The gripper is moved to this position, and the component is placed.

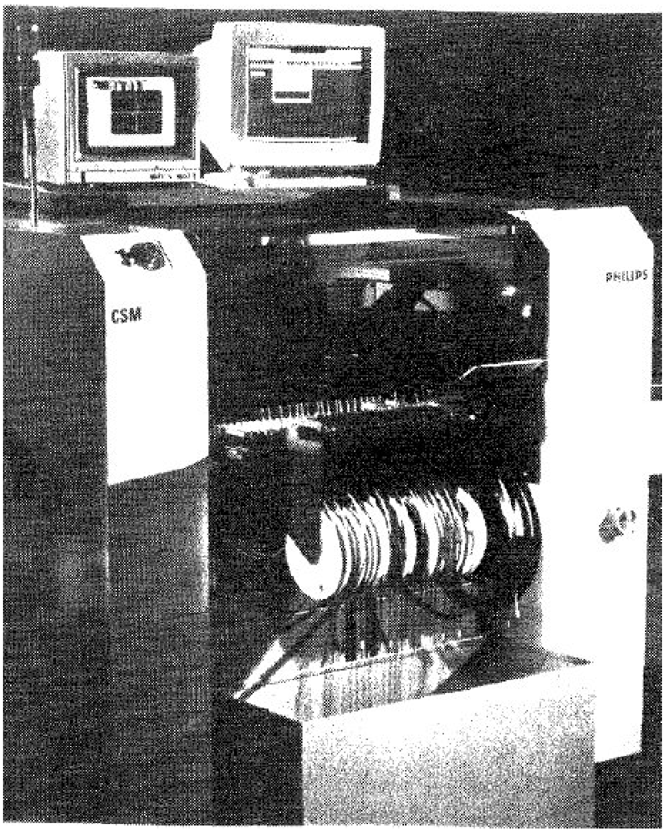

Figure 1: *Philips IE/EMT SMD component placer*

This scheme makes the vision tasks easy: both fiducial and component can be measured under ideal circumstances. This is needed to obtain the required reliability (50 part-per-million misplacements) in practical situations with component and PCB variations. To get the required accuracy however, interpolation on the PCB, including stretch/shrink effects is needed; the gripper has to move with high accuracy; and the vision systems must be carefully calibrated w.r.t. the machine.

A more straightforward approach would be a look-while-place approach. During component placement, the component lead to PCB solder pad distances are measured, and driven to zero. PCB stretch and machine inaccuracies are then automatically compensated, while the vision system calibration is not critical.

A problem with the look-while-place approach is the difficulty of the vision task of measuring reliably lead to pad distance in one image. As the correction movement has to wait for the vision measurement to complete, the time for the vision task has to be short because it adds to the placement cycle time. This has prevented from using the look-while-place approach in existing SMD placers.

Due to their learning capabilities, neural networks could allow the development of a look-while-place approach because the vision task could be learned from examples instead of being explicitely developed. They may also facilitate the adaptation to new component types and, therefore, simplify the task of the end user. Furthermore, it is hoped that a neural network approach will provide more robustness against image noise due to changes in the lighting conditions, and to imperfections of component surface.

III. A neural approach for component alignment

A. Principles of neural networks

Neural networks have gained popularity among the scientific community during the last decade because of their success as non-linear adaptive systems [3]. Many neural network models can be described as a non-linear parametric function $s = G_w(e)$, where e is the input vector, s the output vector, and G_w a function parameterized by a vector w. The entries of w are the weights of the network.

Let us consider for instance the well known multilayer perceptron (MLP). The output vector of the 3-layer perceptron depicted on figure 2 is given by:

$$s = f\left[W_2 f\left(W_1 e + b_1\right) + b_2\right]$$

where f is a non-linear function (usually the hyperbolic tangent), W_1 and W_2 are matrices, and b_1 and b_2 are bias

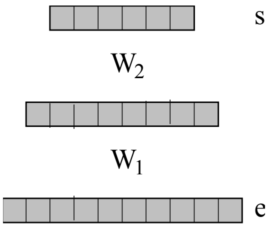

Figure 2: *A 3-layer perceptron*

vectors. The learning algorithm, known as "backpropagation" [4], updates the components of the matrices and of the bias vectors according to the gradient of the mean-square error $e_{MS} = E\{||s - s_{desired}||^2\}$.

B. Alignment control

B.1. Principle of the approach

Figure 3 shows an overview of the approach developed by Thomson CSF/LER [1]. There are two cameras which take images of two opposite corners of the component. The images are coded at 8 bits per pixel, and their size is 512x512 pixels. Figure 4 shows an example of image. On this image, we can see the component body, the pins (bright areas connected to the component body), and the footprints (bright areas on the PCB, under the pins). Here, there is clearly a positioning error due to insufficient accuracy of the mechanical system.

The images are obtained by two CCD cameras: it is not possible to obtain a single image of the whole component, because the pipette which moves the component prevents from having a camera just above the center of the component. On each image, two windows are extracted around the corner (figure 5). Each window contains a partial view of a side of the component, as shown on the figure. The window which contains a horizontal side of the component is called WH and the window which contains a vertical side is called WV. The size of a window is LVxLH, where LV=96 and LH=128 (the windows are not square in order to take into account the fact that the pixel is not square).

On each window, projections are computed, a horizontal projection $ProjH(y)$ and a vertical projection $ProjV(x)$:

$$ProjH(y) = \sum_x I(x,y)$$

$$ProjV(x) = \sum_y I(x,y)$$

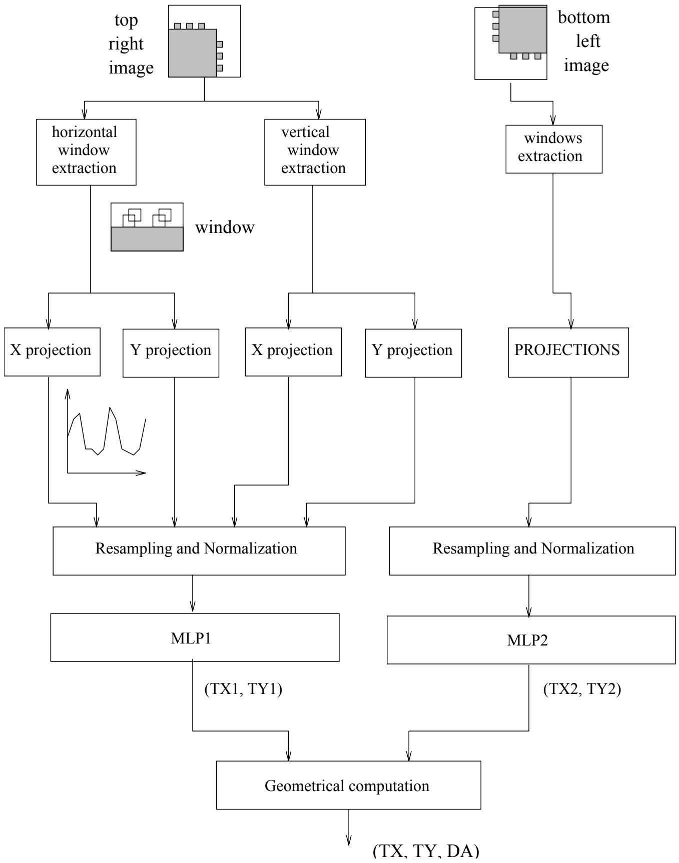

Figure 3: *Overview of the approach*

where I(x,y) is the brightness of a pixel.

Hence, we obtain four projections: ProjH_WV, ProjV_WV, ProjH_WH, ProjV_WH, the length of which is respectively LH, LV, LH, LV.

Then, the projections are normalized in amplitude, in order to obtain values between zero and one. Finally, they are resampled (using a linear interpolation): a projection orthogonal to the side of the component (ProjV_WH, ProjH_WV) is represented on 64 samples, while a projection parallel to the side of the component (ProjH_WH, ProjV_WV) is represented on 32 samples.

The normalized and resampled projections are then concatenated to provide the input vector of the neural network. The order is ProjH_WV, ProjV_WV, ProjH_WH, ProjV_WH. The size of the input vector is 192.

The neural network is a 3-layer MLP. The sizes of the lay-

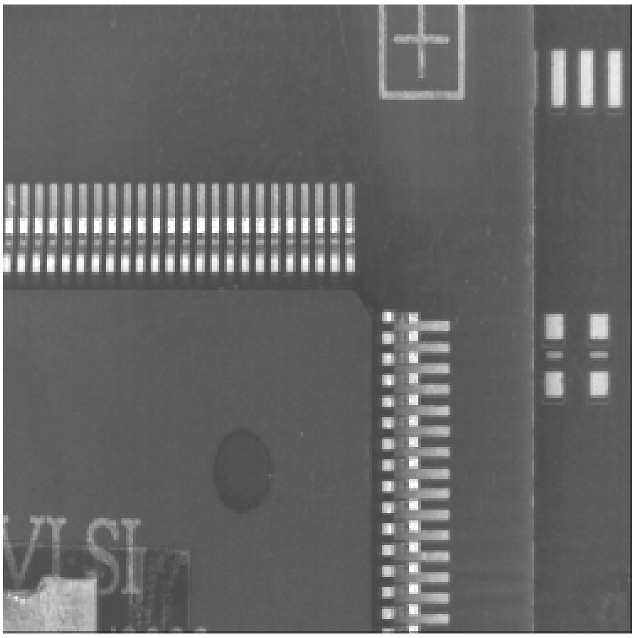

Figure 4: *An example of a source image: $QFP160$ on PCB*

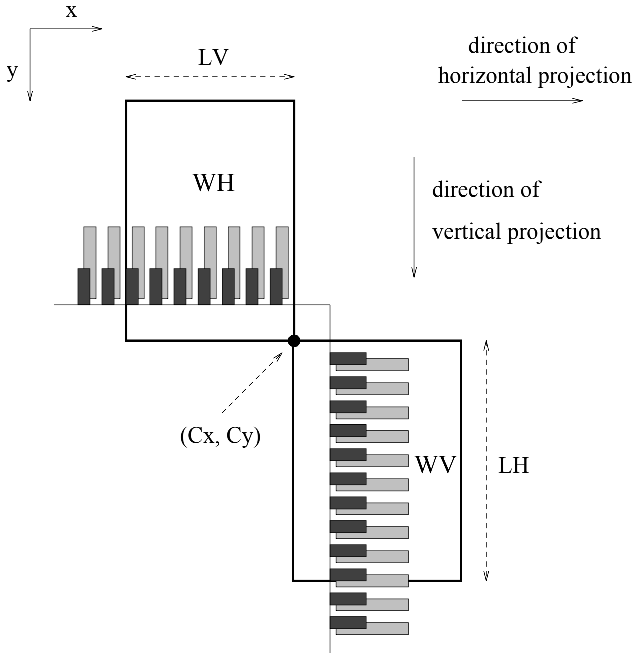

Figure 5: *Windows extracted around the top-right corner*

ers are 192, 48, and 10. The output layer can be divided into 2 groups of 5 neurons. The first group codes for the X-translation(horizontal), and the second group for the Y-translation (vertical). Inside a group, the neurons code for translation amplitudes $-L_p/2$, $-L_p/4$, 0, $+L_p/4$, $+L_p/2$, where L_p is the pitch, and interpolation is used for intermediate amplitudes. The network is previously trained on a database showing various examples of positioning error. These examples are collected using a precisely calibrated mock-up. In order to increase the precision in the recall phase, a linear interpolation between the two highest activations inside each group of neurons is performed.

A good estimation of the local translation can be obtained. Then, a postprocessing module combines the local translation estimations provided by both MLPs in order to compute the global rotation and translation. This process is explained in the section below. This information is given to the placement machine for correction of the mechanical placement.

B.2. Computation of global translation and rotation from local translations

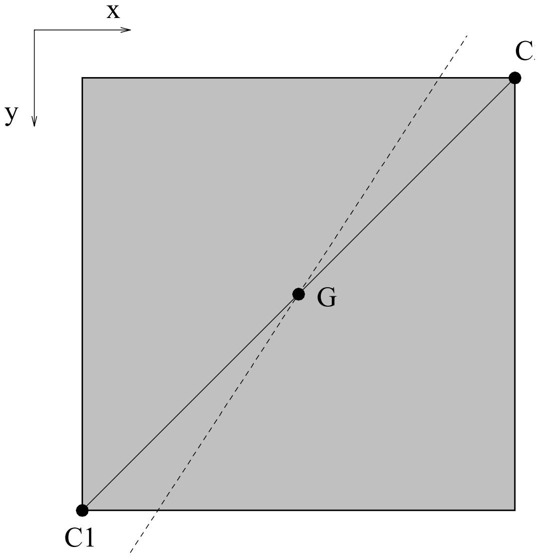

Figure 6: *Local and Global Estimations*

Let us consider figure 6, and let us note:

- $\vec{T}_{C1}$ and $\vec{T}_{C2}$ the translations estimated around the corners C1 and C2

- $\vec{T}_G$ the global translation of the component
- θ the global rotation of the component

$$\vec{T}_{C1} = \begin{pmatrix} T_{X1} \\ T_{Y1} \end{pmatrix} \qquad \vec{T}_{C2} = \begin{pmatrix} T_{X2} \\ T_{Y2} \end{pmatrix}$$

$$\vec{T}_G = \begin{pmatrix} T_{Gx} \\ T_{Gy} \end{pmatrix} \qquad \overline{C_1C_2} = \begin{pmatrix} a \\ b \end{pmatrix}$$

Using the basic rules of geometry, we can show that the global parameters are given by:

$$\begin{cases} \theta &= \dfrac{1}{a^2+b^2}\left\{a(T_{Y2}-T_{Y1}) - b(T_{X2}-T_{X1})\right\} \\ \vec{T}_{Gx} &= \frac{1}{2}(\vec{T}_{X2}+\vec{T}_{X1}) \\ \vec{T}_{Gy} &= \frac{1}{2}(\vec{T}_{Y2}+\vec{T}_{Y1}) \end{cases} \qquad (1)$$

C. Results

Figure 7 shows the precision of the local estimation of the horizontal translation error. The maximum estimation error is $0.07(L_p/2)$, where L_p is the pitch. Since the required precision is $L_p/6$, these results are very satisfactory.

From equation (1), it appears that the accuracy of the global translation estimation is the same as the local estimation accuracy. The accuracy of the global rotation estimation can be easily derived from the local translation estimation accuracy. From equation (1) we obtain (when $a = b$, which is the usual case):

$$\Delta\theta = \frac{2}{a}\Delta T$$

For example, for a QFP160, we have $a = 28\text{x}10^3\mu m$, $L_p = 650\mu m$, and $\Delta T = 0.07(L_p/2) = 23\mu m$. Hence $\Delta\theta = 3.3\text{x}10^{-3}rad = 0.2deg$.

Experiments with changing lighting conditions have shown that the estimation error stays inside the tolerance area for a lighting power comprised in the interval $[\frac{P_N}{8}, 4P_N]$, where P_N is the nominal lighting. This means that it is very robust with respect to lighting conditions.

IV. Hardware implementation

A. Running neural nets on multiple DSPs

Within the framework of the Galatea Esprit project (funded at 50% by the EEC), a full environment including software and hardware tools dedicated to neural networks based algorithms has been developed. At the highest level, a user friendly graphic interface set up by Mimetics (France) allows the user to design his own neural network by means of a set of predefined tools on a hierarchical object-oriented way. He can also add his own objects by describing them as C++ classes. Once the learning method has been chosen among a large library, the corresponding code (for both learning and recall stages) is automatically generated and cross compiled into Virtual Machine Language (VML). This language dedicated to address parallel hardwares is at the level of the C language. The standardization of VML is in progress.

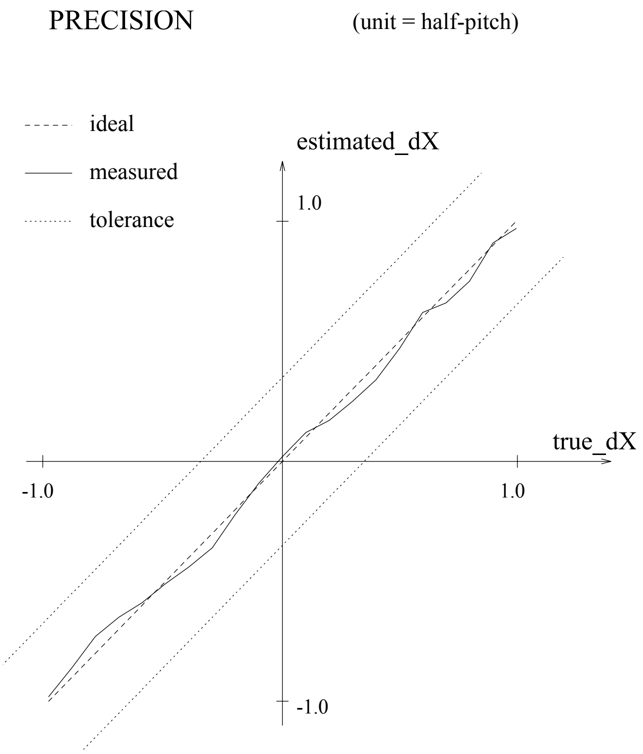

Figure 7: *Precision of the local estimation*

Then the on-line implementation is addressed by LEP who has developed a VML cross-compiler towards a set of 6 Texas Instruments DSP TMS320C40 chips embedded on two VME bus based boards. This set of boards with the corresponding software is referred as the LEP's GPNC (General Purpose Neuro Computer). All the cross compilation stages down to the DSPs remain hidden to the user who monitors all the preprocessing on a Sun Sparc 2. An accurate design of the VML cross-compilation towards C40's assembler allows to reach more than 95% of the theoretical peak performance. In the recall stage, up to 75 times faster execution compared with a Sun Sparc 2 workstation can be obtained. A master-slave implementation has been chosen where the master DSP controls the program flow, the I/O with the Sun, the slave data management, and the scalar operations. From 0 up to 5 slaves are supported depending upon the requirements of the user in terms of CPU. Peripherals dedicated to the C40 based platform such as a color frame grabber, a VME bus shared memory and semaphores, and PC communication via a RS232 link are supported at the VML level (figure 8). All of that enables the LEP system to cope with the three major stages of a neural network application, i.e.: the data base acquisition, the learning stage, and the on-line execution of the recall stage.

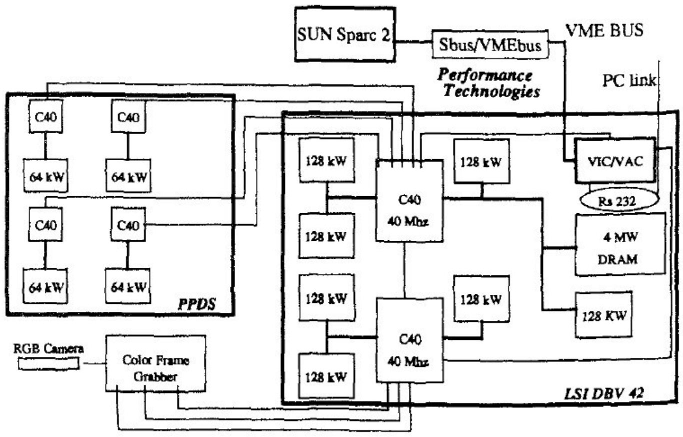

Figure 8: *LEP's GPNC*

B. Implementation of the SMD application

In the SMD application, the database has been acquired by means of a single CPU board, the learning done on a 6 DSPs set and the recall stage performed on a single DSP board. In order to estimate the global transformation (2 translations + 1 rotation), 2 images have to be recorded corresponding to each of the opposite corners. The database acquisition takes about 4 minutes, the learning consumes about 10 minutes (twice 1750 examples learned). Then the net weights are used to compute the correction. Those three steps lead to an average time of 15 minutes to "learn" a new component. Estimating the misalignment needs about 1.7ms per image on a single CPU board (it corresponds to some 150 Koperations and 50 Kinstructions per image). The synchronization with the positioning machine is achieved by a simple signal transmission on the RS232 link. The on-line throughput is about 21000 components per hour when image grabbing is taken into account.

C. Further work

Several other applications have been successfully addressed by LEP on its multiple DSP platform such as an OCR, an orange video grading machine (together with CRAM-Italy). LEP is now working on a more cost-effective solution for embedded hardwares. This will be achieved by the new LNeuro2.3 SIMD processor [2]. An entire software and hardware compatibility with Galatea's environment is scheduled for the end of 1994.

V. Conclusion

We have proposed an approach to include vision control and neural networks in the SMD assembly process. We have defined an approach that combines classical image processing (projections), Neural Networks (MLP), and geometrical computations. The first experimentations have shown a good accuracy.

The method has the advantage to be very easy to use (even for the creation of the learning set, very little human effort is required), and to be fast (on the DSP board, the processing of an image requires less than two milliseconds). Furthermore, the experimentations we have performed show the robustness of the method: it still works very well in presence of noisy and low contrast images.

Since it is a look-while-place method, the main interest with respect to current SMD mounters vision algorithms is the fact that PCB stretch and machine inaccuracies are automatically compensated, while the vision system calibration is not critical.

References

[1] G. Burel, "Vision Feedback during SMD Placement", Esprit Project 7807 (Galatea2), Report V5-M40/2-LER, march 9th, 1994

[2] P. Friedel, "Réseaux de Neurones, Traitement des données, et Extraction de connaissance", Entretiens Science et Défense, Tome 2, 1994, pp 371-382, ref IBSN 2.10.002399.3

[3] R.P. Lippmann, "Pattern Classification using Neural Networks", IEEE Communication Magazine, November 1989

[4] D.E. Rumelhart, G.E. Hinton, R.J. Williams, "Learning internal representations by error backpropagation", Parallel Distributed Processing, D.E. Rumelhart and J.L. Mc Clelland, Chap8, Bradford book - MIT Press - 1986

[5] Ivo Van RIJSSEL, "User requirements of the Galatea vision workpackage dedicated to SMD placement", Galatea Report (European Esprit project n^o 5293), April 15th, 1991