Dynamic alignment for real-time CAD-driven PCB inspection

Ong, S.-A.

Award date:
1995

EINDHOVEN UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING
Control and Measurement Group (ER)

Dynamic Alignment for real-time CAD-driven PCB inspection

by S.A. Ong (ITE 234205)

M.Sc. Thesis
March 1991
commissioned by
prof.ir. F.J. Kylstra (TUE)
under supervision of
ing. A.J.M van Lier (Philips CFT)
ir. A.C.M. Gieles (Philips CFT)
ir. R.G. van Vliet (TUE)

Dynamic Alignment for real-time CAD-driven PCB inspection

by S.A. Ong (ITE 234205)

M.Sc. Thesis
March 1991

commissioned by
prof.ir. F.J. Kylstra (TUE)

under supervision of
ing. A.J.M van Lier (Philips CFT)
ir. A.C.M. Gieles (Philips CFT)
ir. R.G. van Vliet (TUE)

Abstract

This report deals with the work performed during my final project period at the Philips Centre for Manufacturing Technology (CFT). It can be fitted in with the whole of ESPRIT 2017.

ESPRIT 2017 involves the automated, CAD-oriented inspection of bare printed circuit boards (PCB), hybrid boards and fully mounted surface mounted devices (SMD) boards. Philips CFT is a main contributor to the project by means of the development of the 3D laser scanner and the belonging data processing hardware for inspection of bare printed circuit boards.

Concerning data processing for bare PCB inspection, ESPRIT 2017 adopts CAD-driven measurements. These however require the CAD reference data to be dynamically aligned with respect to the flow of image data because of misalignment introduced by miss-orientation, product deformation, and/or non-ideality of the scan pattern.

For alignment purposes, elementary board patterns are designated to serve as alignment features whose positions are known in advance. By measuring their locations in the image of the PCB under test, the misalignment prevailing in the near vicinity of an alignment feature can be determined. The determination of the alignment features' location is being performed by means of *multi-level input binary template matching*. In order to assess for its measuring performance, an evaluation in view of alignment purposes has been performed.

The evaluation results serve as condition for the elaboration of a concept for dynamic alignment. So do the requirements put by the concepts adopted by other data processing units, the real-time restrictions and the laser scanner's scan pattern.

A concept for dynamic alignment has been elaborated, and evaluated by means of computer simulations. Computer simulations also provide for visualization of the concept, in order to ease giving demonstrations for educational purposes

The concept for dynamic alignment takes practical situations as false alarms, missed recognitions, and measuring inaccuracy into account. Computer simulations show that 1.5 mm misalignment can be corrected for within a tolerance of 40-60 μm, which is the order of magnitude of the finest patterns comprised in future PCBs.

Contents

1 Introduction

Efficient production of bare printed circuit boards (PCBs) requires automatic inspection at different stages during and after manufacturing. At the moment, Philips CFT already provides for this need by means of the PCB inspection machine FLAIR. FLAIR is based on laser scanning and real-time processing of picture data, which makes it possible to inspect large product volumes at high speed and high resolution.

Because of recent developments in PCB manufacturing, a more advanced automatic inspection system is required. Main reason for this is the development of track sizes in the coming five years; tracks are estimated to become in the order of 50 μm wide, with interspacings in the same order.

Additional to 2D information obtained by vision, height information must be available in order to be able to guarantee sufficient copper cross section area for transport of electricity. Research project ESPRIT 2017 meets these requirements of automated process and assembly inspection for very fine line and spacing patterns. It was started early 1989.

The approach in which the PCB under test is scanned and the information data processing is performed, differs considerably from the concepts being applied in FLAIR. Concerning data processing for bare PCB inspection, ESPRIT 2017 makes use of CAD-driven measurements. These require the CAD reference data to be dynamically aligned with respect to the flow of image data. A dynamic alignment system has therefore to be introduced.

To meet this objective, the dynamic alignment system's environment will first be presented in Chapter 2 to Chapter 5. Chapter 6 presents the main causes for misalignment and the resulting design specifications for the dynamic alignment system.

At Philips CFT, concepts for performing the alignment have been proposed. These make use of alignment features which are to be recognised. Recognitions are performed by means of multi-level input binary template matching which will be discussed in Chapter 7. Its measuring performance has been deduced by means of software simulation (Program EVALUATE) in order to evaluate its aptitude for alignment purposes. Software simulation has also been carried out in order to ascertain the minimal hardware configuration needed for detecting alignment features properly. The evalution is being presented in Chapter 8.

To demonstrate the concept of recognizing alignment features, Program DUAL_DRC has been written.

An algorithm for calculating correction vectors needs to be elaborated and evaluated. This will be discussed in Section 9.6 to Section 10.6. Elaboration and evaluation have been carried out by means of Program ANAALIGN. The algorithm derived is based on the framework

described in Section 9.1 to Section 9.5. Finally program AMU demonstrates the simplified operation of the dynamic alignment system and is to be used for educational purposes which of course puts certain demands on man-machine interfacing. Appendix 9 presents the user's guide for the various programs written.

2 Origins of PCB inspection

2.1 Intelligent automation

Today's market often requires many variants of each product to be manufactured, and new designs are being introduced frequently. The very long production runs on a single design of product that enabled inflexible and usually expensive automation to be used are more and more being replaced by runs on a smaller number of products. For this reason dumb automation is inadequate, and the need for some intelligence in automation emerges. The flexibility introduced when intelligence is added to the line can make short production runs, possibly even of single items, economical. It sometimes enables tasks to be performed which otherwise would be impracticable.

A formal and precise definition of "intelligence" in automation cannot be given. It can however be stated that intelligence is concerned with describing and quantifying the ability to process information. The more sophisticated the processing, then, the higher the intelligence.

As a minimum, the following attributes are required of a system for it to be regarded as being intelligent:

1. An ability to process information.
2. An ability to accept information from the world outside the system.
3. An ability to output information to the outside world.
4. An ability to vary this output in response to what has been received as input.

More sophisticated intelligent systems can do rather more, for example:

5. The ability to perform rational operations on data, including the making of decisions, the perception of relationships and the quantification of similarity and difference.
6. The ability to output information in a form that may be understood and exploited by other intelligent systems, or human beings.

Speaking of automation involving intelligent processing, or speaking of robotics, are the same issues. There are two distinct classes of robotic activity. In passive robotics (for example, automated visual inspection), samples are sensed and the information thus obtained is used to characterize and/or classify the samples, but the physical action is minimal. In active robotics, on the other hand, complex manipulations are performed, and an ability to perceive and anticipate position and orientation in three dimensions is important.

Intelligent automation is already being applied extensively in industry. The chief objective is of course to adapt to the requirements of the market in order to increase, or if not possible maintain productivity and hence profitability. Other aspects such as a need to make production

less boring, or more safe, or to eliminate a dependence on scarce skills, have also been important considerations. Among the operations intelligent automation can be applied to, we mention:

1. Inspection of discrete items (e.g. engineering components, PCBs) for flaws such as missing and extra holes.
2. Verification of items, for example to detect the presence of an incorrect product in batch production lines.
3. Detection, sizing and identification of flaws on sheet material such as steel strip, paper, glass, and textiles.
4. Gauging, i.e. checking that critical dimensions on components are within a certain tolerance.

etc.

Adding intelligence to automation generally improves upon the performance of existing arrangements, which results into increased speed, accuracy, reliability, consistency, and cost-effectiveness. Adding intelligence to automation also yields an increase of flexibility which makes it possible to adapt an automated line when the product is altered or replaced just by reprogramming.

Concerning this report, we are merely interested in passive robotic; inspection in particular. Inspection is perhaps the most common activity which can be performed by intelligent instrumentation. Of course, the aspects previously discussed concerning intelligent automation, also apply to inspection operations.

2.2 Inspection

Inspection involves the detection and classification of defects on materials or objects. A defect is some property of material or object which is not normally present (i.e., represents departure from the "design specification") and which causes the product to be less or even non-acceptable to a purchaser or a user. We must distinguish critical and non-critical defects. Non-critical defects do not involve the functional operation of the product, contrary to critical errors which are brought to light by the product's malfunction or may cause malfunction in due course.

Inspection of unpopulated PCBs involves the indication whether electrical paths are correct and whether certain board elements such as solder pads are being present. In order to inspect unpopulated PCBs, a purely electrical approach is inadequate since it merely determines whether electrical paths are correct at the time of inspection; latent defects (which will cause trouble later, possibly after components have been inserted into the board and the whole system has been delivered to the end user) are not detected. We therefore need to perceive the board's defects visually. This can however impossibly be done by human operators; the

operation is costly, inefficient and above all impracticable. Thus a need exists for instrumentation to automate the inspection of the boards.

2.3 Achievements

As already stated, efficient production of bare PCBs requires automatic inspection at different stages during and after manufacture. At the moment Philips CFT already provides for this need by means of the PCB inspection machine FLAIR. FLAIR features the following specifications:

Size of board	full size: 650x400 mm^2
Scanning system	2D laser scanning
Resolution	maximum pixel resolution 20 μm/pixel
Inspection time	2 minutes for a full size board
Defects	The defect detection is based on the "design rules" method. This method requires a system to be programmed with the acceptance criteria for the particular PCB being inspected. The system can be programmed with the minimum line width, and the minimum conductor to conductor spacing. Violations which appear like real patterns are however not found. The "design rules" approach is analogous to a human inspector in that it does not need to know about the circuit pattern on the board. It looks only for violations of the design rules, such as open circuits, short circuits, nicks and spurs on the whole boards. This approach makes masking of positions not to be inspected necessary.
Performance	100% of violations against design rules must be detected, though possibly with some "false alarms" to be corrected later by a human inspector
Trainability	The different kind of types of PCBs requires all different system settings. These settings have to be stored on disk. Because defects are determined based on design rules, no track pattern or what so ever has to be stored, only the mask pattern.
Indication	When the machine has completed its inspection of the PCB, the operator receives a report on the monitor, showing the number of errors found. The operator can now display the errors one at a time and decide whether the detected flaw is non-critical, repairable or harmful.

The chief difficulty in inspecting printed circuit boards is the high resolution required in sensing because many significant defects are very small. Thus, the quantity of information to be sensed and processed is large. For this reason, FLAIR is based on laser scanning and

processing of picture data in real time (the time available to inspect an assembly produced or supplied at one per second is equal or less than one second). The structured light of the laser beam increases contrast and makes inspection possible on a large variety of products. Another advantage of laser scanning is that the inspection resolutions can be changed without changing lenses and focusing procedures.

2.4 Current state of the art

Although FLAIR's operation amply suffices in practice, provision of a more intelligent automation system is under study. Due to developments in PCBs to come, a more advanced automatic inspection system is required; ESPRIT 2017 was started. Main cause for this is the expected development of track sizes in the coming five years; tracks are estimated to become not more than 50 μm wide, with interspacings in the same order. See Figure 1.

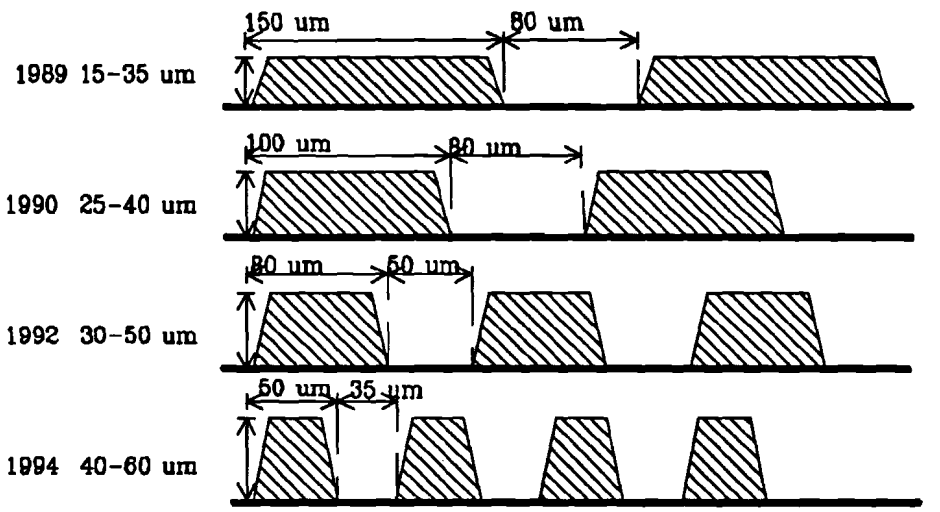


Figure 1. *Developments in track width and interspacings*

If we start from the structure of an intelligent system for industrial automation following the schematic form presented in Figure 2. then it can be stated that the basic methodologies and concepts of FLAIR and ESPRIT 2017 are quite different.

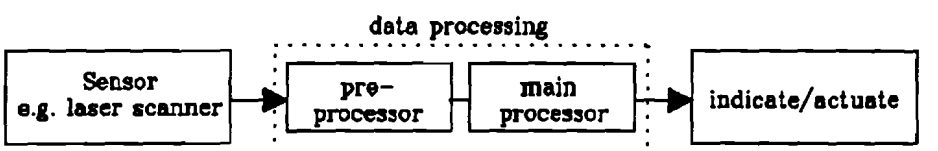


Figure 2. *Generalized structure of an intelligent system*

Concerning *the sensor*. Because the two dimensional reflectance image acquired by laser scanning of the board under test does not comprise a complete specification of the intactness

of the track pattern, height information is required. By acquiring this height information (third dimension), it can be determined whether tracks have sufficient copper cross section area for transport of electricity.

Concerning *data processing*. The use of Computer Aided Design (CAD) techniques is now widespread in industry; maximum benefit is obtained if CAD systems are used to guide automated assembly and inspection. An instrument to inspect printed circuit boards could, for example, be linked directly to the CAD system used to design the boards. This however requires a completely different approach of the system's data processing part compared to the FLAIR. Another limitation of information processing hardware of FLAIR is that it makes use of binary operators which yield a limited accuracy and thus a limited applicability when sizes become smaller. Full grey level operators are preferred.

Concerning *defect indication*. Human operators are now being deployed for judgement, whether defects appeared are harmful or not. By adding intelligence, we aim at minimizing human involvement as much as possible. In the ideal situation, the indications of defects encountered only consist of critical errors.

With the above mentioned desiderata in mind, research project ESPRIT 2017 was started. It specified the requirements of automated process and assembly inspection for very fine line and spacing patterns. ESPRIT 2017 however does no only tackle the problem of unpopulated printed circuit board inspection. Besides the inspection of unpopulated printed circuit boards, it also considers the automatic inspection of printed circuit boards with components mounted, since the work to be undertaken is similar to a certain extent. This work is being carried by other partners.

Thus, ESPRIT 2017 involves the automated, CAD-oriented inspection of bare printed circuit boards (PCB), hybrid boards and fully mounted surface mounted devices (SMD) boards. Philips CFT is a main contributor to the project by means of the development of the 3D laser scanner and the data processing hardware for inspection of bare printed circuit boards.

If the progress a project such as ESPRIT 2017 can be comprised in a flow diagram as presented in Figure 3. on page 9, then ESPRIT 2017 can be considered to be at a stage involving methods evaluation and approach selection. This report will mainly discuss the data processing hardware for bare PCB inspection, and in particular the necessary arrangements to be made in order be able to perform CAD-driven PCB inspection. It will discuss the evaluation of proposed concepts involving the alignment of CAD reference data to acquired image data of a bare PCB under test.

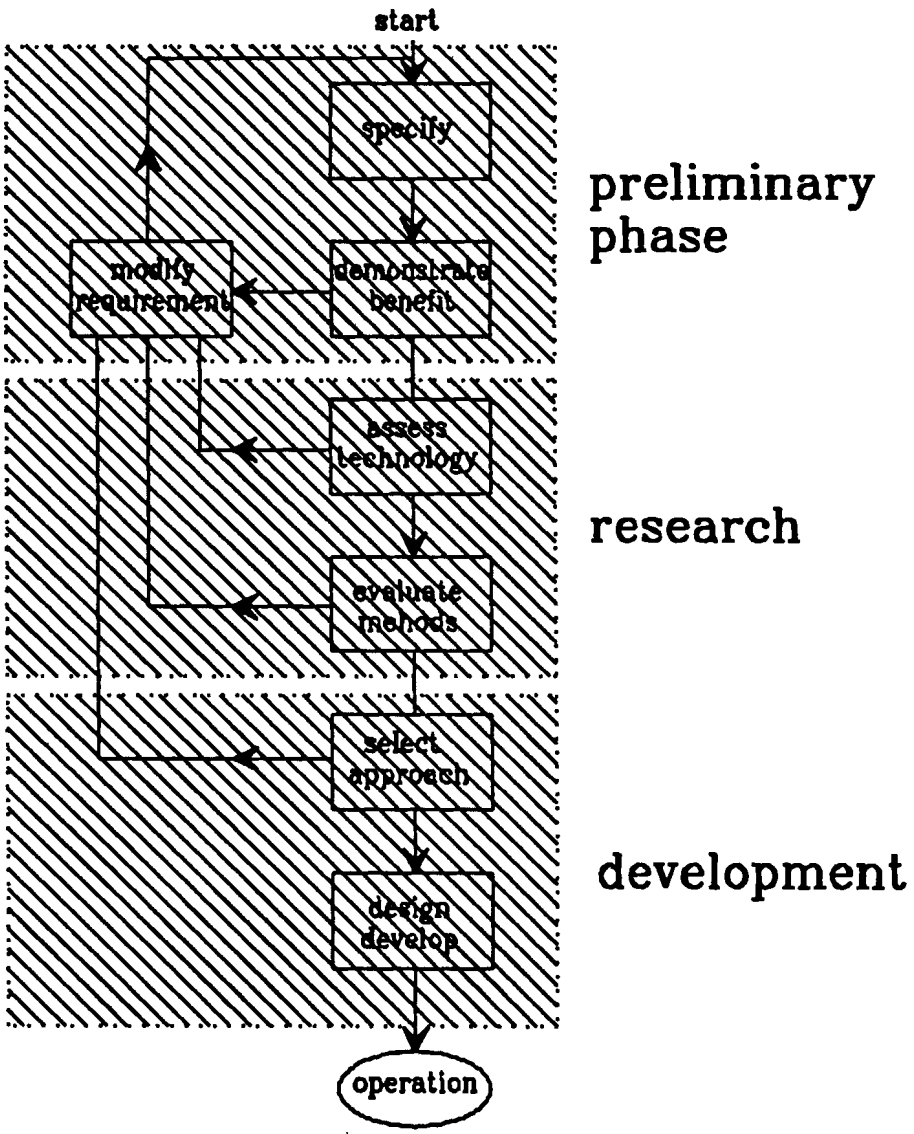


Figure 3. *Stages in the provision of an intelligent automation system*

3 Image acquisition

The way the image of a printed circuit board under test is being acquired, determines in every sense which methodologies and concepts can be adapted for data processing. Therefore, an introduction into the basic principles of image acquisition is required. It will however be beyond the scope of this report to present the discussion in all its details.

3.1 Introduction

The means by which information about a scene is transferred to a processing unit involves, in general, three factors. These are the illumination of the scene, the conversion of the optical pattern into electrical form, and the transfer of the electrical information to the processor. The conversion of the pattern is performed by sampling the brightness or reflectance of individual points in the scene and will usually involve a scanning action either within the optoelectronic converter or before it. In some cases the illumination of the scene is incorporated into the scanning system. In many systems an optical unit, e.g. lenses, will be required to form an image of the scene within the optoelectronic converter. Optical systems may also be required in the illumination of the scene.

Scanning systems can be divided into two groups, those that have a light spot scanning the scene and detect the strength of the returned light, and those in which an image of the scene is formed within the optoelectronic converter and scanned electronically. Although the second is by far the larger and more useful group, it is not suitable for fine line pattern inspection of bare PCBs due to the amount of data to be processed and the resolution required.

3.2 Properties

3.2.1 Scanning patterns

Scanning patterns determine the way in which the picture elements or pixels, are being acquired and comprise the following types:

The *raster pattern* is the most common pattern and is a format suited for picture processing in that it can be stored and processed in computer memories in a flexible and logical way. It is the pattern used in television systems but generally in the form of two interlaced scans.

The *circular pattern* can be used to determine orientation with respect to the centre of the scan. Special solid state arrays exist in this format.

The *spiral pattern* covers an area. This pattern is useful in determining the orientation of an object if the scan can be centred on a specific point of the object.

In the *random pattern*, usually the scan is directly under the control of a computer and consequently has a slow pixel rate. On the other hand it scans only the interesting points and therefore is more efficient in that it acquires only the required information.

The *linear pattern* is suitable for position sensing, but is also used for scanning scenes where the scene itself is moving, for example on a conveyor belt. This principle is being adopted for bare PCB inspection by the laser scanners of FLAIR and ESPRIT 2017.

3.2.2 Scanning rate

Two factors affect the scanning rate, the time required to move from one scanning element to the next and the minimum value of either the time the scan spot has to remain on each element or the time between successive scans of each element or both. Both times relate to the correct operation of the photo-sensitive device.

3.2.3 Signal-to-Noise Ratio

The noise superimposed on the output of a scanning system arises from several sources. Noise is generated within the electronic circuits and there is a random function in the collection of photons. Both of these effects are worse when the light level, and hence the signal level, is low. Noise may also be introduced when the sensitivity of the system or DC offsets, vary with the position of the scanning element. For this reason it is known as spatial or fixed pattern noise. It can be measured because of its nature and cancelled from the signal by processing.

3.3 Image quality

Image quality obtained by a scanning system, can be defined as being the relationship between resolution and contrast. *Contrast* is defined as the difference between the intensities in light and dark areas of the image detail normalised by the difference between solid black and white areas. As the detail in the object becomes finer, the contrast in the image will become worse. *Resolution* basically refers to the effective size of the scanning element at the scene. Usually this element does not have sharp boundaries and may vary in size across the scanned area. Contrast becomes worse as detail in the object becomes finer, because cross-talk between adjacent scanning elements appears.

3.3.1 Aberrations

To obtain a resolution as high as possible, the size of the scanning element must be minimized. Optics however reproduce increasingly fine detail with increasingly poor contrast. These defects due to optics' limitations are called aberrations. They cause sharp points and edges in the image to become blurred. Related to laser scanners, aberrations yield a limitation to the minimization of the scan spot size which results into a maximum resolution.

In monochromatic light such as laser light, the following aberrations may be present:

- *Spherical aberration.* Light rays passing centrally and those passing off-centre through a lens come to focus at different distances from the lens. This causes a blurred image.
- *Coma.* When light rays form a relative large angle with the main optical axis, the off-centre rays come to focus to one side of the central ray position. This produces comet-shaped images of point objects.
- *Astigmatism.* This occurs when various details in the image come to focus at different distances from the lens.
- *Field curvature.* The surface of best focus is not flat. Focus therefore varies across the image or plane of projection.
- *Distortion.* A square grid in the object plane is translated into a distorted grid in the image plane, or plane of projections.

3.3.2 Modulation transfer function

Resolution can be completely expressed in terms of the modulation transfer function (MTF). If the object is a bar pattern with sinusoidally varying intensity distribution, then the amplitude ratio is the ratio between image contrast and object contrast. The modulation transfer function of an optical system is the variation of this as a function of the fineness of the pattern (spatial frequency). The typical form of the MTF is shown in Figure 4. on page 12. Even in the absence of aberrations the resolution and contrast reproduction of an optical system is limited by diffraction and the MTF is not a straight horizontal line. Aberrations take the MTF further below this ideal curve.

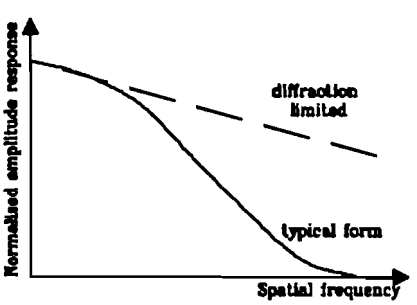


Figure 4. *The MTF*

3.4 Laser Scanners

The light emitted by the laser has the property of being almost perfectly parallel, and all of the light can be focused into a spot on a distant surface, except for transmission and reflection losses at the optical components. By the addition of a means for moving the spot across the scene and of detecting some of the light scattered from the spot and/or its direction, an image acquisition system can be produced.

A laser scanner consists of a spot projection and scanning system, comprising a laser, beam shaping optics, focusing optics and a scanning element.

The 3D laser scanner to be built is in some respects similar to the one used in FLAIR. In principle, both scanners consist of a helium-neon (He-Ne) laser, whose beams are deflected by a fast rotating polygon mirror and focused on the PCB surface under test by a focusing lens and field flattener.

Design specification for the 3D scanner have been formulated as follows:

- 300 mm scan width
- 15 x 15 μm^2 pixel size, preferably 12,5 x 12,5 μm^2
- 15.10^6 pixels/s data speed
- pseudo 3rdD contrast enhancement

Pseudo 3rdD contrast enhancement will be discussed later on.

3.4.1 Spot projection

Spot projection of the 3D laser scanner of ESPRIT 2017 and the 2D laser scanner applied in FLAIR, are being performed using a rotating mirror drum. This is a polygonal prism with polished faces, rotating around its axis. Due to this rotation, the polygon sweeps the beam over a certain angle, causing the spot to be moved along the scan line. As the object under test is moved under the scanner, a complete 'image' can be built up line-by-line. This results in a linear scan pattern and a continuous flow of pixels.

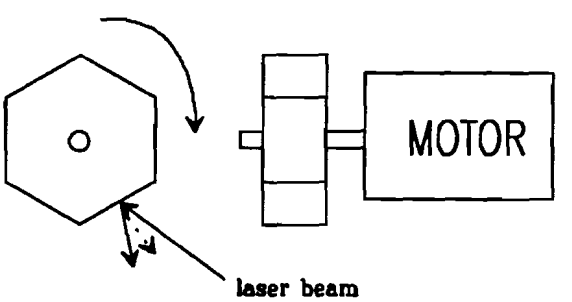


Figure 5. *Polygon scanner type*

3.4.2 Spot generation

Equipment for generating the scan spot to be projected on the surface of an object under test, comprises beam shaping and focusing optics. Apart from necessary provisions and adaptations in order to actually make the optical system operable, and apart from the deflection detection mechanism, the most elementary components are shown in the following schematic diagram.

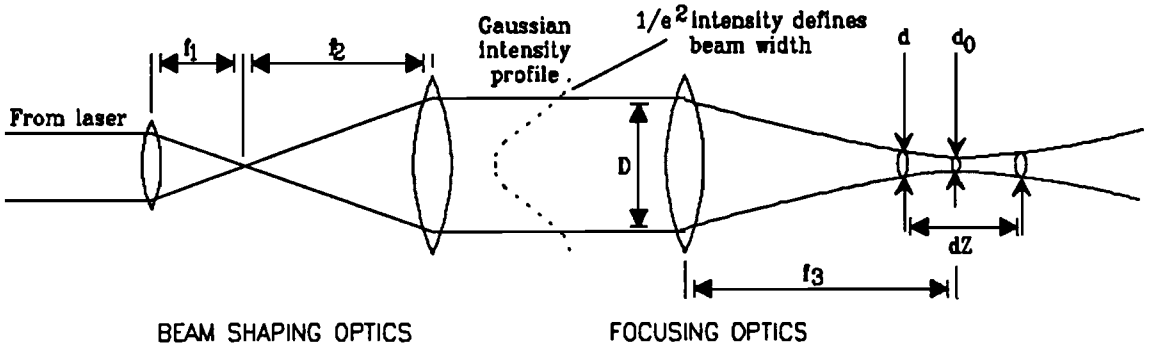


Figure 6. *Basic principles of scan spot generation*

To focus the laser beam down into a spot of a chosen size, the raw beam must first be expanded to a certain diameter D. The beam is expanded by the ratio between the focal lengths of the two lenses, f_1 and f_2. By applying a focusing lens, the laser beam is projected to its final size. The scan spot size determines the resolution of the inspection system.

Most elementary, diffraction limited spot generation is being considered. This means that the limit to how small the scan spot can be made is not due to optics' aberrations but to fundamental optical properties of diffraction. Only the last property is taken into account.

The spot size d_0 is defined to be:

$$d_0 = \frac{4\lambda}{\pi}\left(\frac{f}{D}\right)$$

$$= 8.05 \cdot 10^{-4}\left(\frac{f}{D}\right) \; mm, \; for \; He\text{-}Ne \; laser$$

The depth of focus dZ is the length over which the spot size is within $\sqrt{2}$ of the best focus spot size d_0. A larger beam diameter leads to higher resolution at the expense of less depth of focus.

$$dZ = \pm\frac{4\lambda}{\pi}\left(\frac{f}{D}\right)^2$$

$$= 1.61 \cdot 10^{-3}\left(\frac{f}{D}\right)^2 \; mm, \; for \; a \; He\text{-}Ne \; laser$$

3.4.3 Field flattening

There are two basic types of projector/scanner systems, post-objective and pre-objective. These names refer to whether the scanning element, c.q. the device performing the scan movement, is placed after or in front of the objective, which is mostly a lens which focuses the beam into a spot on the surface. The two configurations are shown in Figure 7.a. and Figure 7.b. The post-objective scanner is cheaper to build and simpler to implement for wide scans because it uses a simple objective lens working on-axis. It, however, produces a curved scan field giving problems of depth of focus and of scan speed variation across the width of a flat object. The pre-objective scanner can be designed to correct these problems to give a flat scan field, at the cost of a higher complexity.

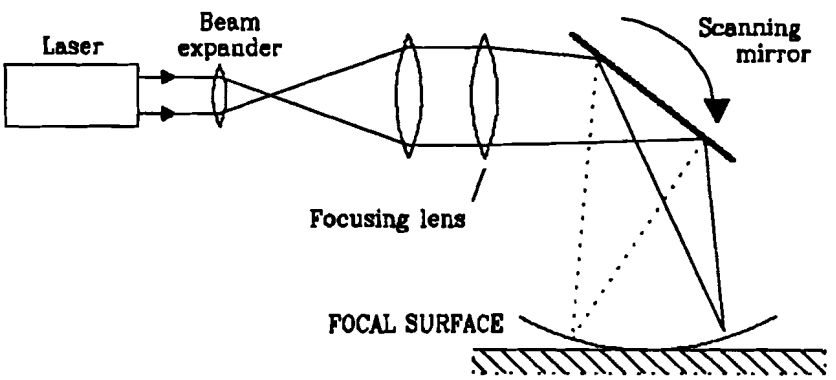


Figure 7.a. *Post-objective type laser scanner*

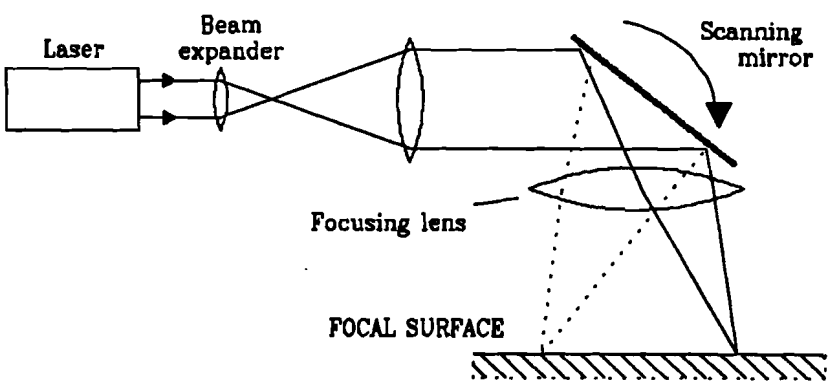


Figure 7.b. *Pre-objective type laser scanner*

The 3D laser scanner of ESPRIT 2017 and the 2D scanner used in FLAIR, both adopt the pre-objective scanning method. A patented telecentric field flattening system is used consisting of a set of two aspherical correction mirrors in combination with a focusing lens. This also enables the laser beam to hit the surface to be inspected perpendicularly. By adopting this system a relatively large scan format can be achieved. Its telecentricity enables synchronized sensing. The property of telecentricity features the possibility to allow the light

reflected to travel almost the same path in reverse direction and directing the light onto a detector, making synchronized sensing possible. The light scattered from the scan element under test may still be collected directly by a light collector and directed to a detector.

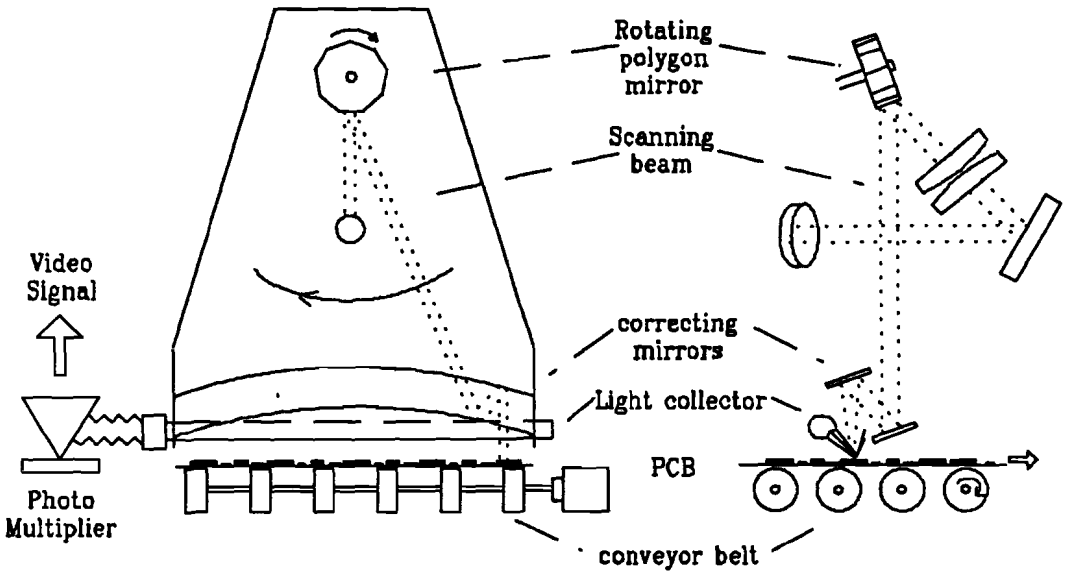


Figure 8. *The FLAIR principle, detection by scattered light collection*

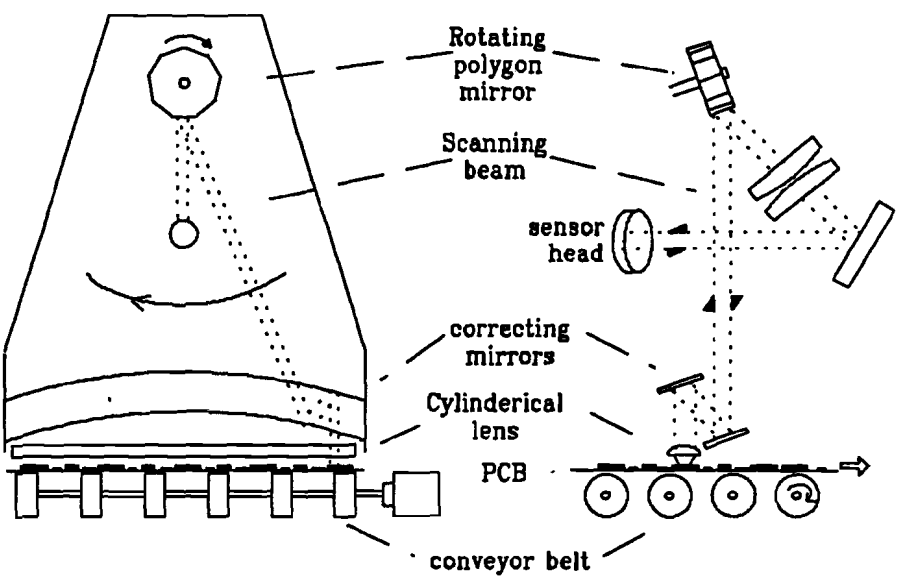


Figure 9. *ESPRIT laser scanner principle, laser beam traverses reversely onto detector*

3.4.4 Detector

The laser scanner used in FLAIR makes use of a photomultiplier to transform optical information into electrical signals. The reflected beams are collected by a glass tube which directs the collected light into the photomultiplier. See Figure 8. A photomultiplier is a device constructed with an evacuated glass envelope, having a front window coated on the inside with a photoelectric material (the photocathode) which emits electrons when photons strike it. The photoelectrons are accelerated by a voltage difference between the cathode and a plate, or dynode. By hitting several dynodes, the number of electrons is magnified. Finally the electrons reach the final positive electrode causing an output voltage. See Figure 10.

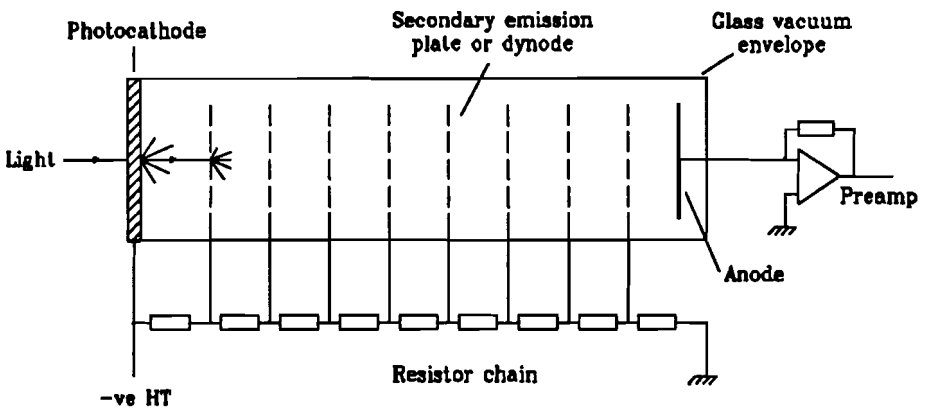


Figure 10. *The principle of the photomultiplier tube*

The 3D laser scanner of ESPRIT 2017 doesn't in contrast to the FLAIR system, collect and detect the reflected light of the PCB directly, but make use of the telecentricity of the scanning element. The reflected beams are now being collected by an optical system and directed to a Position Sensitive Device. See Figure 8. The position the beam hits the PSD varies dependent of the height of the underlying object. By determining the position the beam hits the PSD, the height of the underlying surface can be derived. See Figure 11.

A position sensitive diode can be used as such a Position Sensitive Device. It determines the position of a spot of light with a resolution up to 1 in 10,000, in X and Y. The structure is a thin, large area junction (Figure 12.). The junction is reversely biased and conduction will occur through the junction where the spot of light falls on the device. One surface has electrodes on the X edges and the other on the Y edges. For each surface, the two electrodes are at the same potential and, due to the resistance of the material, the currents flowing to the electrodes are inversely proportional to the distances of the electrodes from the light spot. In practice the values of $(X_a-X_b)/(X_a+X_b)$ and $(Y_a-Y_b)/(Y_a+Y_b)$ are derived from the four currents and from these the position can be calculated.

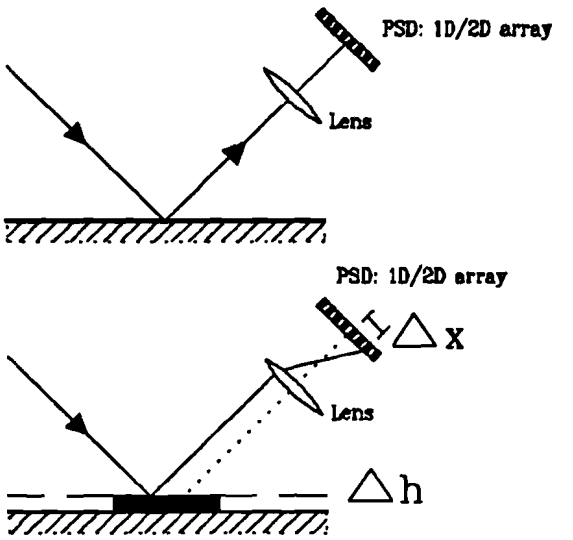


Figure 11. *The principle of triangulation for height measurement*

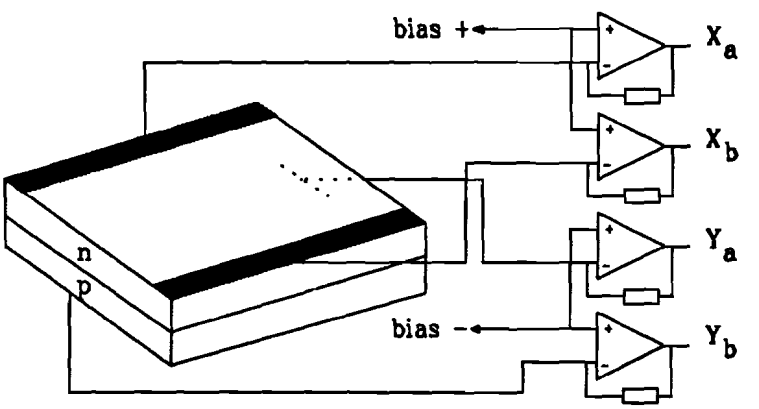


Figure 12. *The two-dimensional position-sensitive diode. the outputs depend upon the position of a light spot on the diode.*

3.5 Edge transition

Image edges are not perfectly sharp, because of cross-talk between adjacent scanning elements, and the optical will not necessarily be spatially synchronised with the element spacing. One or more elements in the vicinity of each image edge will therefore give a signal level part way between black and white. This causes uncertainty in the position of each edge. See Figure 13.

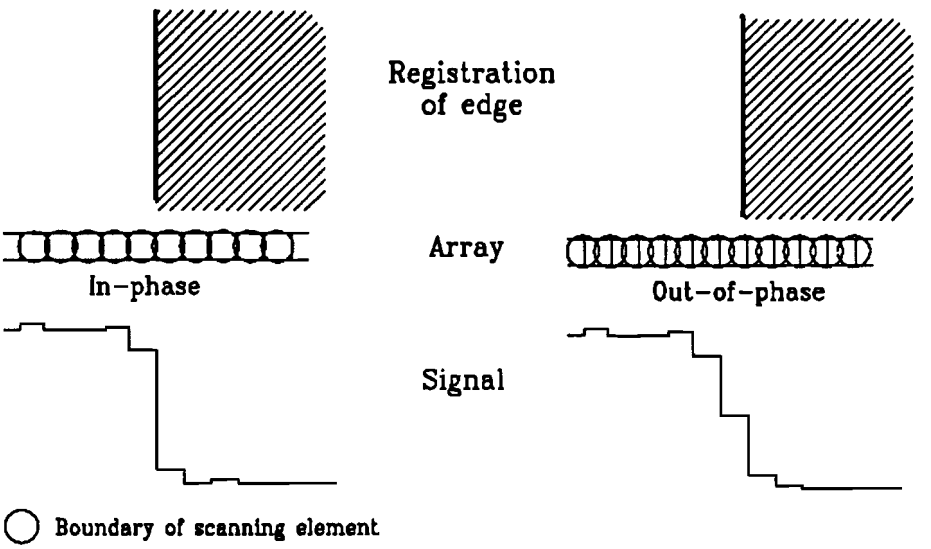


Figure 13. *The effect of spatial phase and over-talk of scanning elements on reproduction of an edge*

To improve edge measuring accuracy, sub-pixel interpolation can be applied. This is an analogue interpolation technique which uses the signal levels from several adjacent elements through which a high order polynomial, for example a third order polynomial, is fitted. By positioning the edge transition at a fixed point, for example the 50% point of the smooth curve transition described by the high order polynomial, a more accurate estimation of the actual edge position can be made. See Figure 14. on page 21.

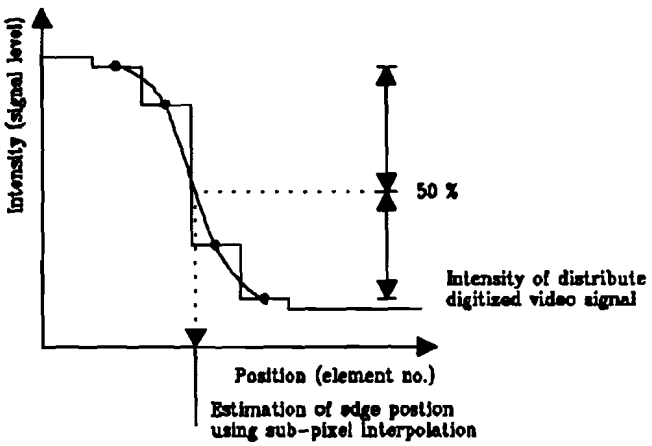


Figure 14. *Third order polynomial curve fitting, for edge transition estimation (sub-pixel interpolation).*

3.6 Benefits of 3D laser scanning

Images acquired by 2D laser scanning of a bare printed circuit board under test do not comprise a complete specification of the intactness of a track pattern as already mentioned; height information is required. By acquiring this height information (third dimension), it can be determined whether cross sections of tracks have sufficient copper area for transport of electricity.

At Philips CFT a real 3D measurement of the tracks and spacings with μm accuracy, is regarded as not feasible and not needed. It is expected that a pseudo (lower accuracy) 3D measurement will be sufficient in order to obtain improved resolution and contrast. A sufficient intensity of the beam reflected validates the obtained height information.

Because of the improved contrast, an effective 5x5 μm^2 2D resolution can be realized by means of two-dimensional sub-pixel interpolation. Sub-pixel interpolation cannot be applied on images based on reflection only such as in FLAIR, because at very fine line patterns, oxidation and surface roughness have a relatively strong and fatal influence on reflectivity.

The data obtained by the 3D laser scanner to be applied for ESPRIT 2017 is thus a combination of height and reflectivity data. After combination of both kinds of data, the image obtained will be more reliable and of better quality than images obtained with the FLAIR system and what is most important, suitable to inspect fine line image patterns and suitable for applying sub-pixel interpolation yielding a higher effective resolution.

4 Data processing architecture

The requirement that the inspection system should be used in flow production, the high data speed of the laser scanner (15 MHz pixel), the linear scan pattern, and high data volumes, make it necessary to perform real-time data processing. At Philips CFT, a concept of pipelined machines has been adopted to meet the computational requirements for real-time inspection, c.q. image processing in ESPRIT 2017.

4.1 Pipelining

Pipelines consist of a set of processors configured in a line, and processing data in parallel. The data is passed from one of the engines to the next. The first result takes N units of time to be computed, after that a new result appears every unit of time. These computational engines can perform any task that is required, for example if the data being passed along the pipeline is an image, then one processor might be a shape recognition engine, or the data might be a flow of control codes and the engines might be controllers.

Figure 15. presents a generalized pipeline processor. It comprises a sequence of stages each holding a vector of data. In synchronism with each clock pulse (the pixel clock for ESPRIT 2017), the data in each stage proceeds (with processing according to the task in-hand) one stage to the right. A new vector of raw data is drawn into the left-hand side, and a new vector of processed data appears at the right. The processing between stages is controllable by software.

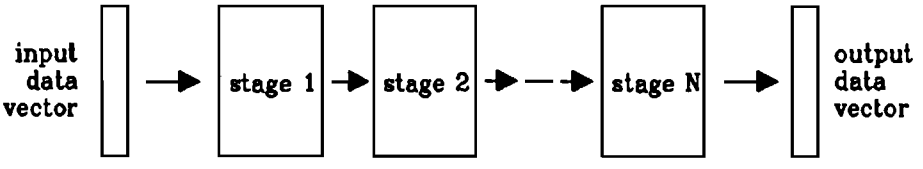


Figure 15. *Generalized pipeline processor*

The concept of pipelining for ESPRIT 2017 is implemented around the PAPS II (Picture Acquisition & Processing System II) bus system by Philips.

4.2 The PAPS II bus system

The PAPS II bus system includes a bus structure and bus architecture suitable for almost any picture processing application. The bus structure specifies electrical and mechanical characteristics such as connector type and pin assignment, control protocols, and voltage

levels. The bus architecture also describes the various pathways available for communicating data and control signals.

4.2.1 System buses

The backbone of the PAPS system is formed by its bus structure - the Picture Bus and the Control Bus. These buses allow the connection of all types of PAPS modules which offer maximum flexibility.

For each type of sensor, such as a 3D laser scanner in the case of ESPRIT 2017, there is a PAPS module called Picture Acquisition Unit (PAU) which acquires the picture information and puts the picture on the Picture Bus for transport to other PAPS modules such as Hardware Processors. Hardware Processors perform all the filtering, pre-processing and processing that a particular system requires, in real time. The results are passed to a general purpose computer by the Picture Transfer Buffer (PTB) module via the General Purpose Bus Unified (GPBU). See Figure 16.

The Picture Bus contains seven channels of eight bits each. Six channels take care of the transport of picture data and will be referred to as data channels. The remaining channel is called the control channel. Each data channel can be assigned as a Hardware Processor input or output. In this way various kinds of pathways can be configured.

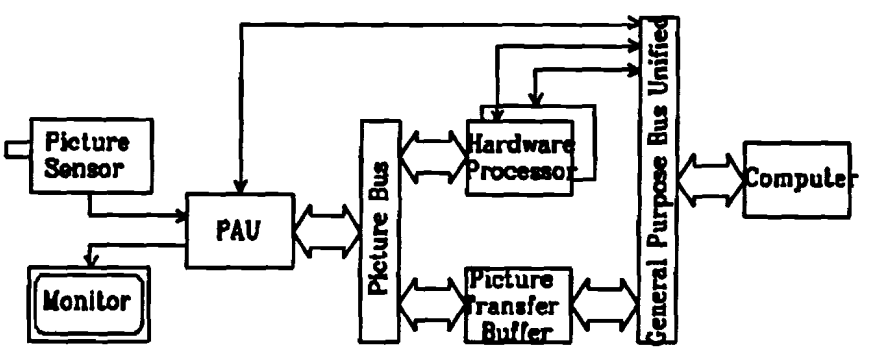


Figure 16. *PAPS II bus architecture*

4.2.2 Control signals

The control channel transports signals, like clock, picture synchronisation and read/write signals to manage the transfer of picture data over the data channels. The following control signals are very important

LIE : LIne Enable (picture synchronisation signal)
PIE : PIcture Enable (picture synchronisation signal)
CLK : Pixel clock (clock)

The signal PIE is high until the end of an image is encountered and indicates that valid picture data is being generated, the signal LIE is high until the end of a line is encountered and indicates the validity of the pixels in a picture line and the signal CLK is high for each pixel. Note that the CLK pulses are supplied by the 3D laser scanner. See Figure 17. In the system the PAU provides for these signals which are to be used by all the data channels. So there is only one PIE and one LIE signal. Let us refer to these signals as the master PIE and LIE.

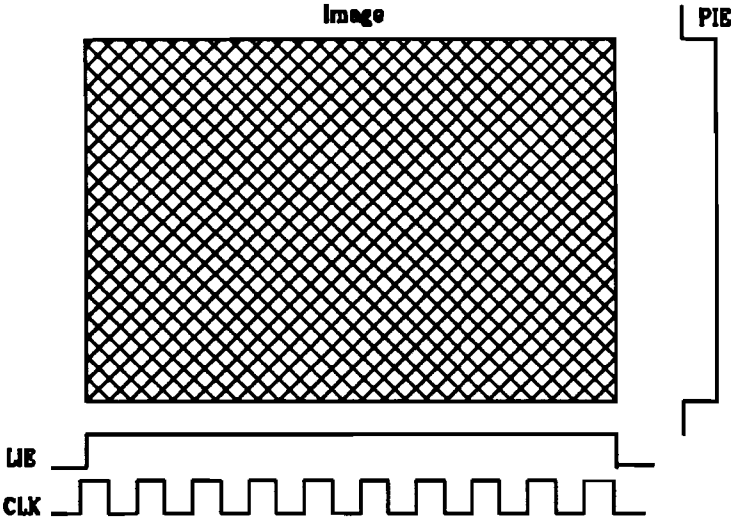


Figure 17. *PIE and LIE control signals*

4.2.3 Possible configurations for the PAPS II bus system

There are two ways to connect Hardware Processors to the picture bus:

- parallel processing
- pipeline processing

When the system is configured for parallel processing, then one data channel is reserved to connect the inputs of the Hardware Processors and another data channel to connect the outputs of the Hardware Processors. This means that a picture can be processed by several Hardware Processors at the same time. See Figure 18., for simplicity reasons the control channel is not drawn.

In the other situation when the system is configured for pipeline processing, each Hardware Processor occupies a data channel for input and another data channel for output. See Figure 19.

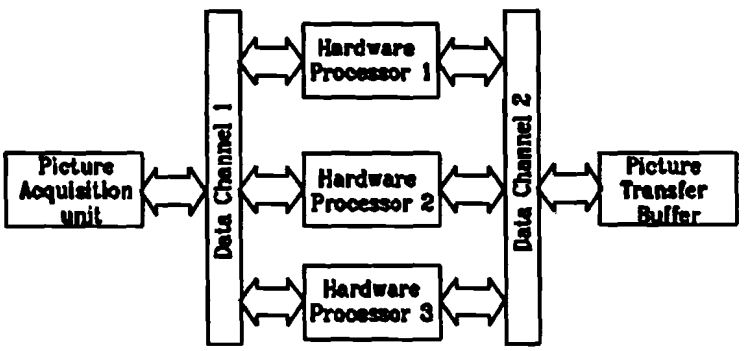


Figure 18. *PAPS II configured for parallel processing*

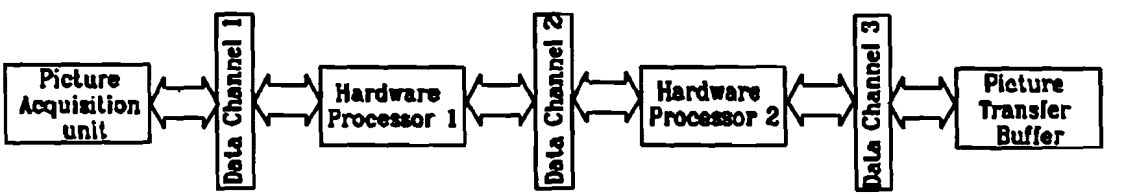


Figure 19. *PAPS II configured for pipeline processing*

Each of the Hardware Processors uses an interface to the picture bus. Registers on the interface can be set by software to select the input and output data channel(s) which the Hardware Processor is going to use. So dependent on the settings of the registers a configuration is created. This can be parallel, pipeline or even a mixed configuration.

4.2.4 Delay of signals

Hardware Processors introduce a certain delay between input and output data stream. As processing for example requires a delay of three lines, then the picture data on the output channel is not synchronized any more with the master PIE and LIE (See Figure 20.) The PAPS II system therefore provides for measures for delaying the control signals PIE and LIE locally on a Hardware Processor. PIE can be delayed up to 254 lines and LIE can be delayed up to 254 clock signals. The amount of delay is software selectable by register setting on the interface.

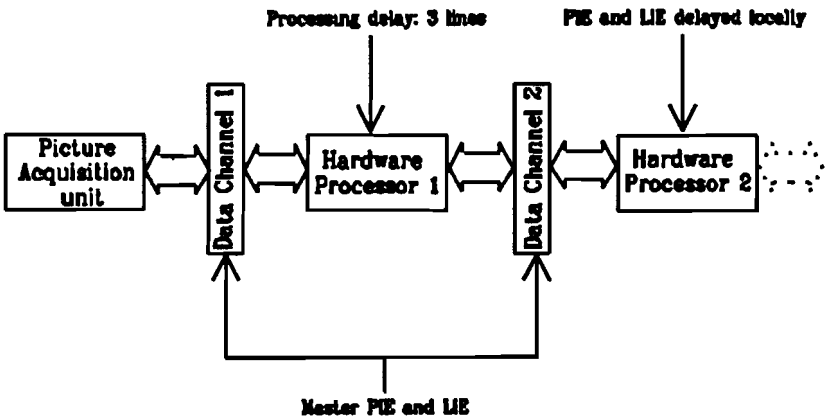


Figure 20. *Delay compensation by use of locally delayed PIE and LIE signals*

Another possibility for delaying picture data is to build cards which store the picture data in a buffer which is as long as the required delay (in pixels). The buffer is first filled and when it is full the picture will be transported to the output data channel. In this way the picture data is delayed. Of course it can be implemented on a Hardware Processor card itself. The only difference is that the output of the buffer will then be used directly by the Hardware Processor (not using a data channel on the picture bus).

5 ESPRIT 2017 System description

5.1 Design specifications

ESPRIT 2017 involves the automated, CAD-oriented inspection of bare printed circuit boards (PCB), hybrid boards and fully mounted surface mounted devices (SMD) boards. It features the following characteristics:

- Data obtained by 3D laser scanner, 8-bit height and 8-bit intensity data.
- High data throughput: data speed of 15 Mpixels/s.
- High data volumes: up to several gigabytes for maximum board size.
- Short processing time: 2 minutes for full size board (60 cm x 60 cm).
- Very fine line and spacing patterns: down to a size of 12.5 μm.
- Fully CAD-driven measurements.
- Virtual resolution increase by use of sub-pixel interpolation.
- Application of full grey level operators instead of binary operators.
- System should be suitable for Flow Line production.
- Real-time implementation.

Figure 21. on page 29 presents an overview of the proposed approach for data processing by means of its functional diagram. Here we see a pipeline oriented system, in which the data throughput consists of flows of pixels, further referred to as images or pictures. They may contain all kind of information, e.g. picture information, control information, error information.

The following subdivision into four different system parts can be made:

I : Pre-processing of image (See Figure 22.a.)
II : Pre-processing of data (See Figure 22.b.)
III : Measuring of patterns (See Figure 22.c.)
IV : Post-processing (See Figure 22.d.)

These system blocks will be discussed in following sections in order to establish the need for dynamic alignment CAD reference data with respect to image data.

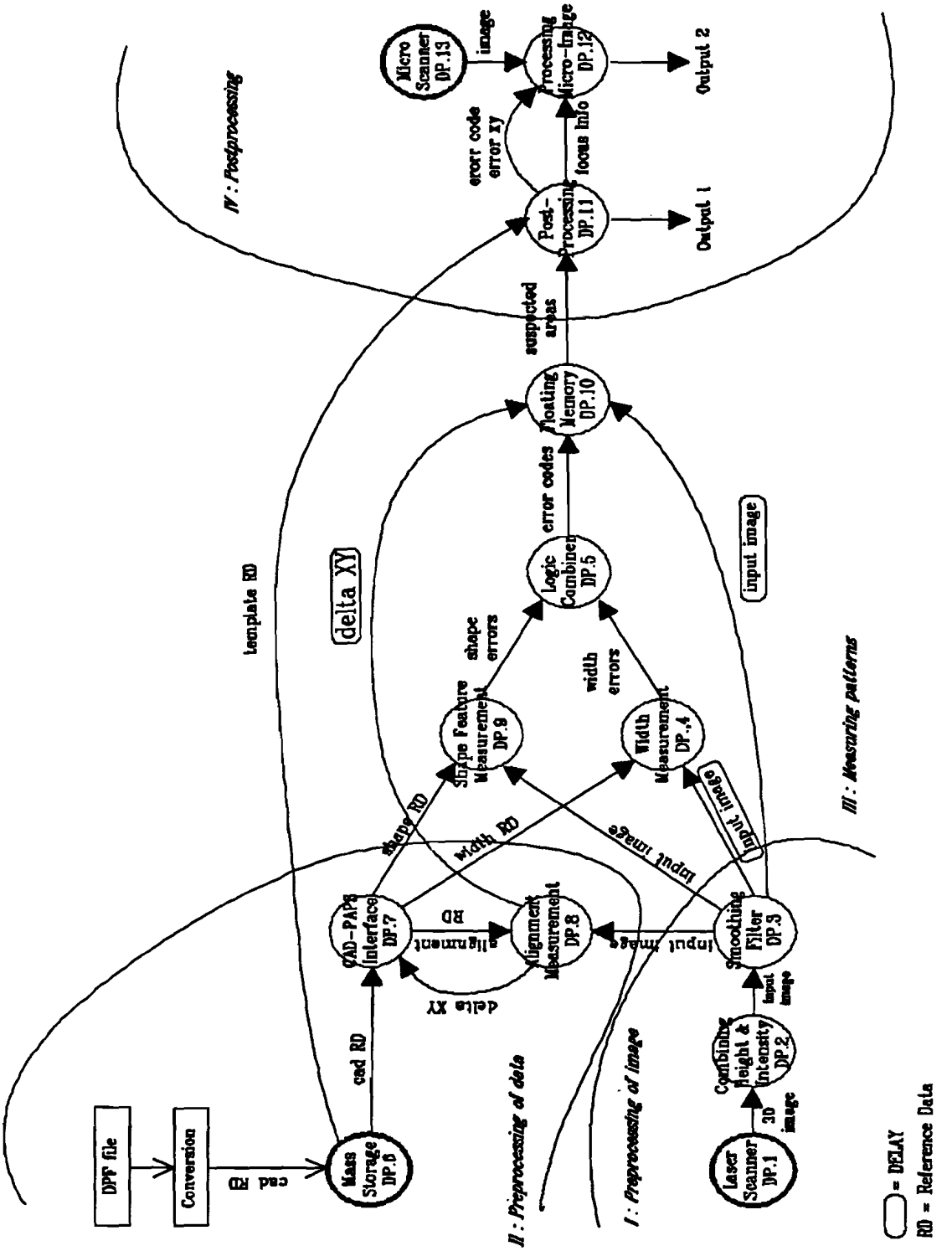


Figure 21. *Functional diagram of ESPRIT 2017*

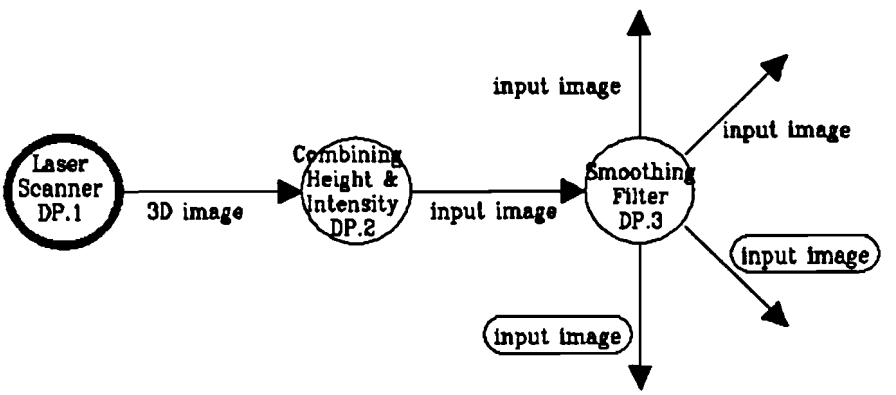


Figure 22.a. *Pre-processing of image*

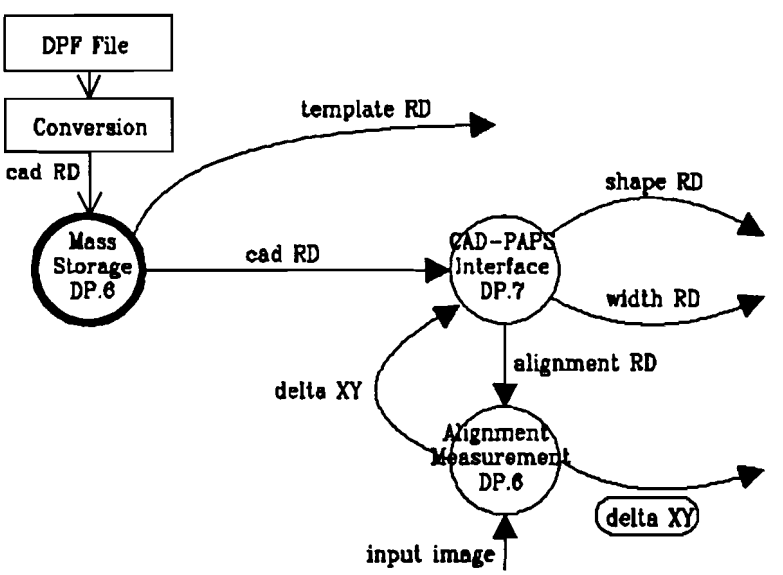


Figure 22.b. *Pre-processing of data*

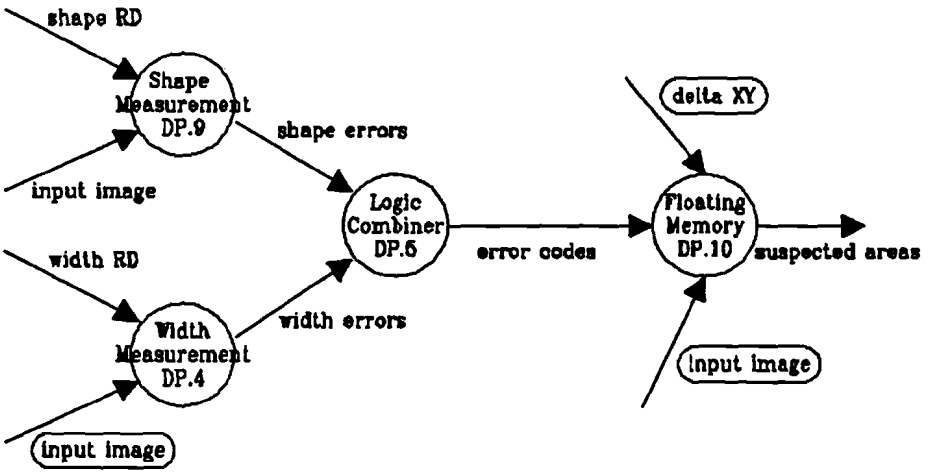


Figure 22.c. *CAD-driven measurements and storage of suspected areas*

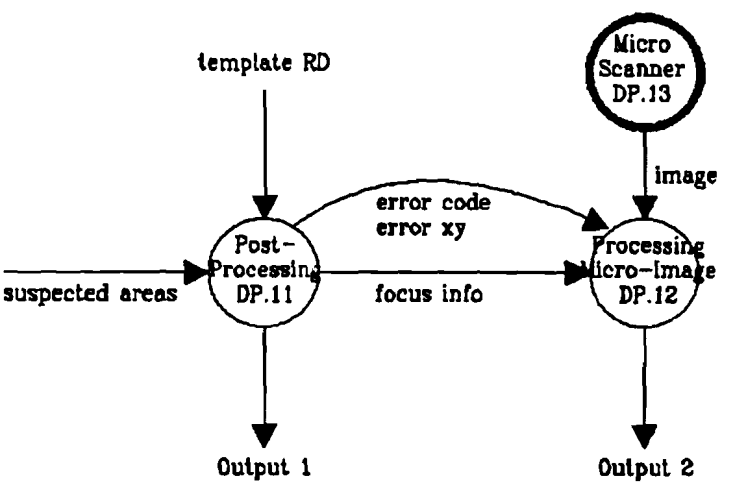


Figure 22.d. *Post-processing*

5.2 Part I: Pre-processing of image

DP.1 The intensity and height information acquired by the 3D laser scanner (DP.1) have to be combined in order to get picture information with high image quality. This is
DP.2 performed in unit DP.2

The input image for the data processing hardware consist of a 8-bit grey level image which may contain some noise, having the behaviour of so called pepper and salt
DP.3 noise. By means of a special Edge-preserving Smoothing Filter (DP.3) this noise can be removed without distorting the edges. See Appendix 3. The obtained image is now suitable as input for other modules following the Edge-preserving Smoothing Filter (DP.3) in the pipeline.

5.3 Part II: Pre-processing of data

The system performs CAD-driven inspection. This means that the system is managed according to CAD reference data specifying the contents of the PCB under test. The CAD reference data is the key for detecting errors on the PCB. The CAD plot files comprising the reference data are delivered on tapes and will first be converted to a condensed line oriented format. This conversion is done off-line. The condensed line
DP.6 oriented data will be stored in a RAM bank (DP.6), because the picture information throughput of 15 Mpixels/s is rather demanding. A hard disk for example would be too slow. The expansion is needed, because the inspection techniques rely on pixel oriented data (referred to as images).

The various units expect CAD reference data containing control information in order to be able to perform their CAD-driven measurements. These information flows are
DP.7 provided for by the CAD-PAPS interface unit (DP.7) which takes care of the expansion of data coming from the mass storage, i.e. RAM bank (DP.6) into the proper real-time data stream for each unit.

The data stored in the mass storage is based on an ideal scan of the PCB under test. There are however several uncertainties introducing misalignments which have to be compensated for.

In order to correct for these misalignments, a dynamic alignment system is necessary, i.e. the alignment is carried out during the inspection task and "continuously" being
DP.8 adapted. The combined action of the alignment measuring unit (DP.8) and CAD-PAPS interface unit (DP.7) causes the CAD reference data to be aligned to the picture information flow within a certain tolerance. A certain amount of reference data is therefore buffered in the CAD-PAPS interface unit in order to be able to advance or retard the data stream to the succeeding units. Forwarding the data stream is necessary

if it appears that the scan position is ahead of the CAD reference data for the various units. Backwarding is necessary if the scan position is behind the throughput of CAD reference data.

Aligning the CAD data to the scanned information is made possible by feeding reference alignment positions to the alignment measurement unit during scan and comparing them to the measured alignment position on the PCB under test. In this way the vector connecting above mentioned positions can be calculated and proper adjustments can be made.

We therefore need to be able to recognise the alignment positions and calculate the measured misalignment vector ($\Delta X,\Delta Y$) pointing from the reference alignment position to the measured alignment position in order to adjust the reference data stream with a certain correction vector ($\Delta x,\Delta y$). See Figure 23.

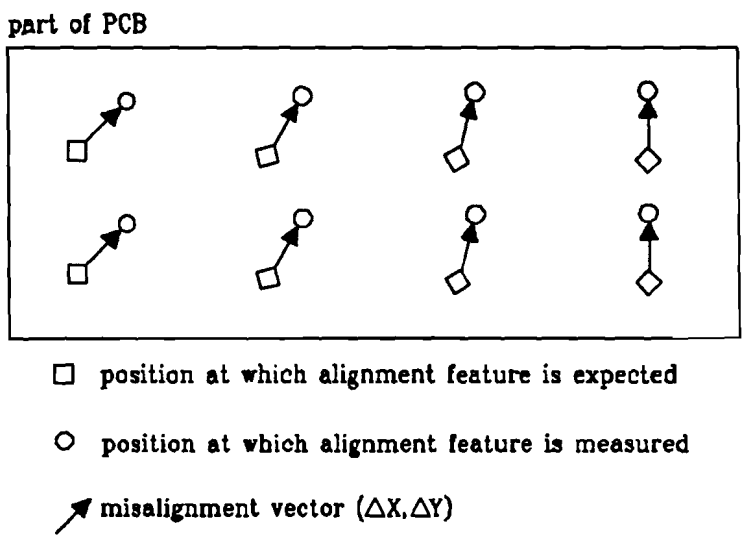


Figure 23. *Measuring the misalignment by means of alignment features*

The buffering and control of reference data containing control data for the succeeding units is managed by the CAD-PAPS interface. The reference data consists of:

- a file with alignment control data, which is a small file of about 20kB, which is loaded completely at initialisation time;
- data file + pointer file for width measurements (= buffered via circular buffer and corrected with correction vectors ($\Delta x,\Delta y$));
- data file + pointer file for shape measurements.

5.4 Part III: Measuring patterns

Real time measurements of features to be inspected are performed within this part of the inspection system.

DP.4 The Width Measurement Unit (DP.4) performs width measurements with sub-pixel accuracy by means of sub-pixel interpolation in the image pattern, especially tracks and their interspacings. The unit's main objective is to determine the width and continuity of very thin lines. Measurements can be performed in horizontal, vertical, and if necessary diagonal direction. The positions in the image where the measurements are to be performed and which values are to be expected, are offered to the unit as width feature reference data. The width feature reference data has to be offered in real time and contains a certain tolerance with respect to the input image. The output of the width measurement unit is just a new image containing codes at those locations where measurement requirements are not matched. The code represents the errortype which has been found.

The inspection of larger image patterns such as footprints, solder or interconnection islands and corners in tracks, is difficult to perform using width measurements because of their geometry and the position inaccuracy. Their presence can however

DP.9 be determined by the shape feature measurement unit (DP.9) using template matching techniques. The presence of various shape features contained in larger image patterns such as corners, edges etc is now determined. The positions in the image where the measurements are to be performed are passed to the unit as shape features reference data. Of course the obtained measurement results are less accurate than the ones obtained using the width measurement unit but this is of no importance because of their size. When a feature is not recognised, an error code instead of zeros (implying no errors) is sent to the logic combiner (DP.5).

DP.5 The logic combiner (DP.5) is just an **AND/OR** gating circuit. The error images from the width measurement and the shape feature measurement unit are logically **OR**ed to one image containing codes depending on the errors occurred. The obtained error image can be **AND**ed with a masking image which contains spots which do not have to be inspected. This will only be used, when no CAD-driven masking is available.

DP.10 An error code causes the floating memory unit (DP.10) to store the surrounding 96^2 pixels of the location at which the error occurred, together with all kinds of information about the error. See Figure 24.

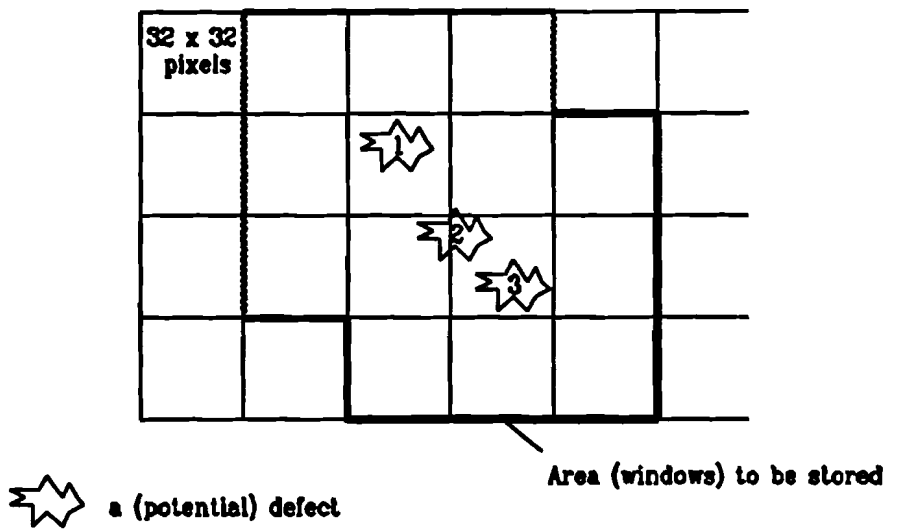


Figure 24. *Floating memory*

5.5 Part IV: Post-processing

In case errors have been detected, the post-processing module looks whether it can classify the errors with a specified certainty. If the errors can be classified then an output is generated indicating at which positions errors have occurred.

DP.11 The stored areas are first to be processed by the post-processing unit (DP.11) and if
DP.12 necessary by the 3D micro image processing unit (DP.12). These two units decide whether an defect is harmful or not. The 3D micro image processing unit (DP.12),
DP.13 bases its decisions on more detailed images passed by the micro scanner (DP.13). The micro scanner has a 3D resolution of 5x5x5 μm^3, its data rate can be up to 2 Mpixels/s, and its scanlength up to 3 mm. Units DP.12 and DP.13 are being carried out by another partner of the project.

The units of course introduce considerable delays. In order to compensate for these delays image delay units (IDU) are inserted. In they are schematically presented in Figure 21. as blobs which are labelled with DELAYS.

6 Misalignment

The alignment measurement unit's objective is to measure and correct for the misalignment of the CAD reference data with respect to the scanned location of the PCB under test. The causes for misalignment are manifold. They can be divided into three classes:

- misalignment due to miss-orientation.
- misalignment due to deviations inherently due to the acquisition system's operation.
- misalignment due to product deformation.

These classes will be discussed in following sections.

Two kinds of misalignment can be distinguished, static and dynamic misalignment. Static misalignment does not change along the scan of a PCB, but remains the same for every scanline. Dynamic misalignment however changes gradually during the scan and has to be dynamically compensated for. See Figure 25. on page 37

6.1 Miss-orientation

During operation of the inspection system, the product to be inspected is placed on a transport table, where it is held in place by vacuum. The transport table moves the object under test, so that the surface can be scanned line-by-line. The actuation of this movement is synchronized by the rotation of the polygon mirror. In this way no noticeable misalignment will beintroduced. See section 6.2.1. for the details.

The initial positioning of the product on the table can however cause a certain miss-orientation with respect to a reference position in the form of translation and rotation. Translation causes static misalignment, rotation causes dynamic misalignment. See Figure 25.

The alignment measuring unit has to be designed to compensate for an expected misalignment due to translation of maximum 0.5 mm and an expected misalignment due to rotation of max 0,32 mm/300mm. These correspond with translation of 40 pixels and rotation of 20 pixels over 300mm of the board, when the maximum resolution of 12.5 μm is applied.

All together the alignment measurement unit has to be able to compensate for a misalignment due to miss-orientation of 0,8 mm, which is about 64 pixels, using a resolution of 12.5 μm.

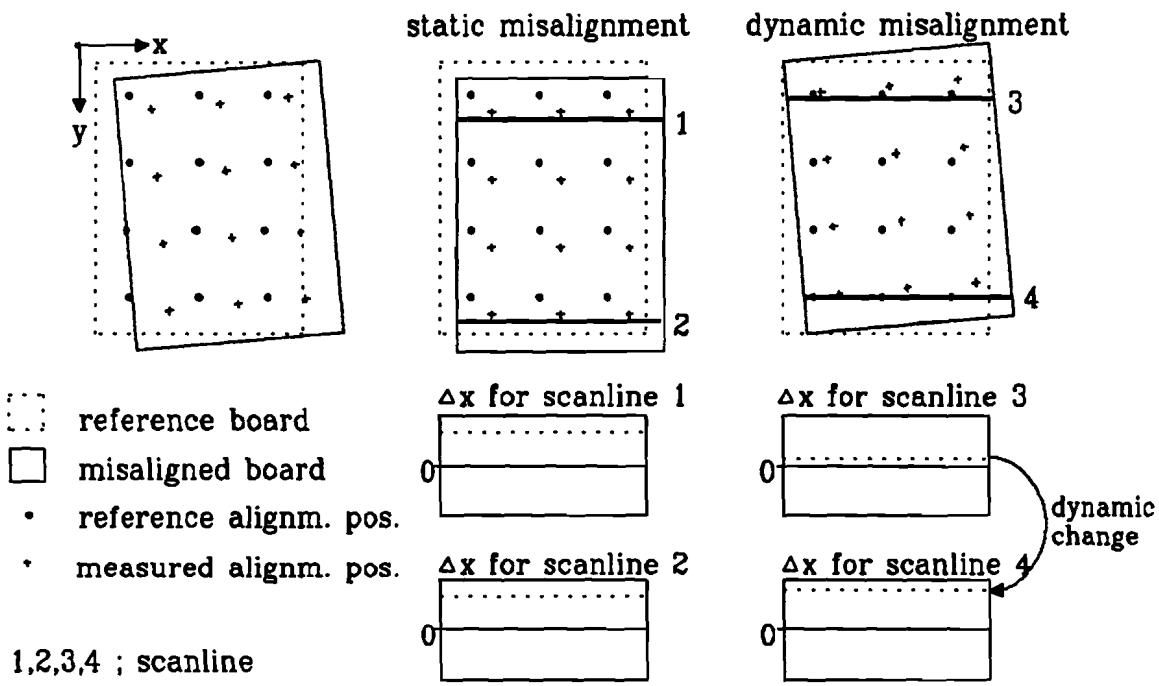


Figure 25. *Types of misalignment, static and dynamic*

6.2 Deviations of the acquisition system

In Chapter 3 the basic operation of a laser scanner has been discussed. The laser scanner's performance is basically determined by:

a) its scan spot characteristics.
b) the deviation with respect to the nominal position, which is inherently due to the projection of the scan spot upon the surface of the object under test.
c) the deviation inherently due to the deflection of the scan spot along the scanline.
d) the deviation inherently due to the collection of reflected rays and the way they are being directed to the detector.
e) the detector's performance itself.

At Philips CFT a number of scanner configurations have been calculated by ray tracing, analyzed and optimized. It would be beyond the scope of this report to discuss these research activities in detail, because we are merely interested in the consequences the 3D laser scanner's main design specifications have for dynamic alignment.

6.2.1 3D laser scanner specifications

The 3D laser scanner comprises the following main design specifications:

a)	scan length	:	300mm, gross (290-295, net)
b)	total optical scan angle	:	27.5°
c)	effective focal length	:	≈ 620 mm
d)	spot size (50%)	:	15-20 μm, wave length= 633 nm (He-Ne laser)
e)	position non-linearity	:	< 0.2%
f)	spot aberration	:	5 μm, long. & transv.
g)	polygonal mirror	:	220 mm diameter
h)	number of facets	:	24
i)	rotating speed	:	1875 r.p.m at 750 scans/s

Ad a) The scan beam is deflected along a scanline with the above specified scan length. See Figure 26.

Ad b) The scan beam traverses an angle of 2 x 27.5° when performing the scan. See also Figure 26.

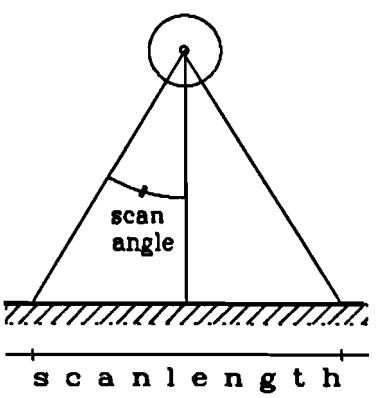


Figure 26. *Laser scanner's scan length and scan angle*

Ad c/d) A spot size of about 20μm will be obtained if f/D= 25 is assumed. Calculation is based on the equations as presented in Chapter 3. λ= 633 nm for He-Ne laser. The modulation transfer achieved is now 100%. A spot size of 15 μm is estimated to yield a modulation transfer of about 50%. A modulation transfer of 50% will be adopted as criterium for resolution of 12.5-15.0 μm.

Ad e) The polygon mirror drum rotates with a constant angle velocity by means of a control system. The pixel stream is synchronized by a master clock, whose signals are derived from the rotation angle of the polygon mirror drum. In this way it can be guaranteed that the pulses of the master clock always occur at the same positions along the scanline; i.e. the spots are always projected on the same positions along the scanline. The interval between clock pulses is however not fixed in time, due to small deviations in the angle velocity.

The constant angle velocity of the rotating polygon mirror yields a linearly increasing optical scan angle w, which is the angle between the laser beam and the line l. Line l is defined to be in between the rotation axis of the polygon and position w_0. See Figure 27.a.

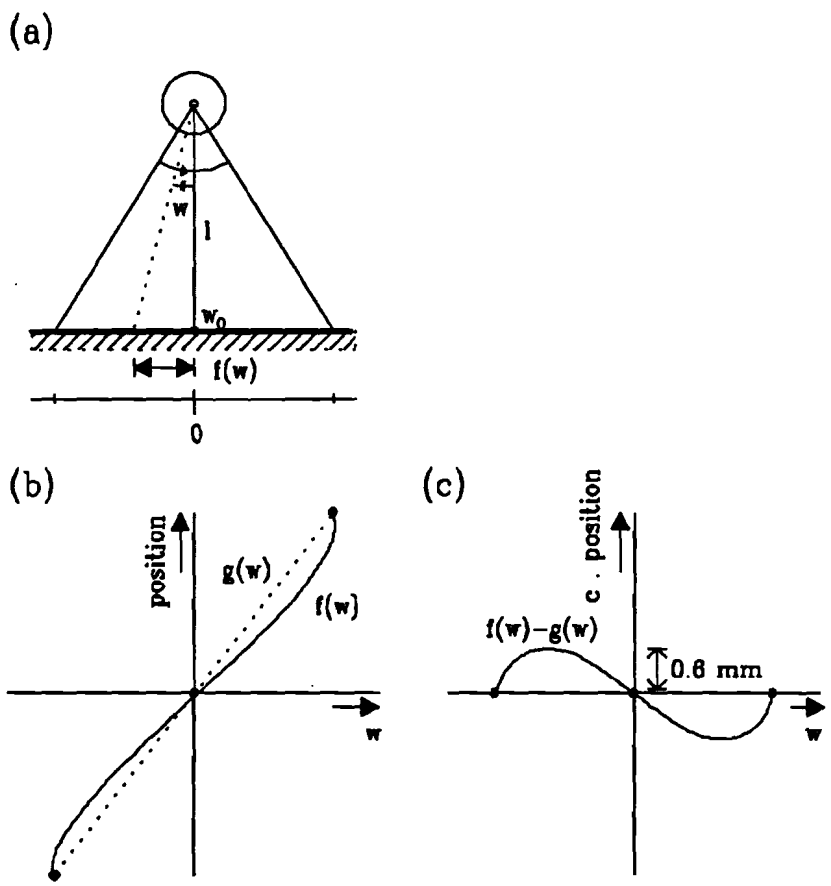


Figure 27. *The laser scanner's a-linear behaviour along the scanline*

The scan spot is projected at position f(w), which is basically the tangent of the optical scan angle w. For this reason it does not progress in a linear way along the scan-line. See Figure 27.b.

The master clock is passed to other units. The scan position for these units is defined by function g(w) which is linear to w. Because of this linearity misalignment exists. In order to minimize the misalignment, g(w) is defined in such a way that it fits through the three dots as specified in Figure 27.b.

The clock signals passed to the other units cannot be designed in such a way that g(w) superimposes f(w). This causes a misalignment f(w)-g(w), as presented in Figure 27.c. yielding position non-linearity. The position non-linearity is specified to be smaller than 0.2% (+/-) which is about 600 μm (+/-).

Ad f) The causes for spot aberration are:

- field curvature, causing the scan spot top be out of focus.
- astigmatism - spot geometry.
- lateral aberration in the cross scan direction.
- lateral aberration in the scan direction by coma.

Spot aberration in both longitudinal and transversal direction is estimated to be about 5 μm. See Figure 28.

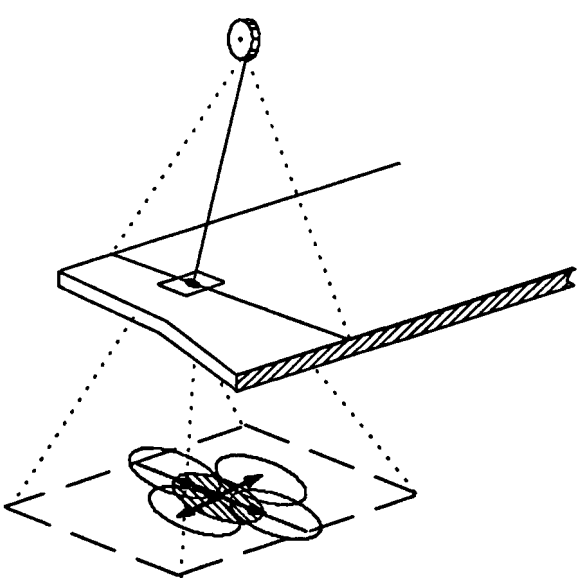


Figure 28. *Spot aberration*

Ad g/h/i) At 15 μm sampling distance, each scan contains 20,000 non-interpolated (full) pixels, so at 1875 r.p.m, or 750 scans/s a data rate of 15.10^6 full pixels per second is produced.

6.2.2 Consequences

There are two main causes for misalignment due to deviations comprised in the acquisition system:

- spot aberration
- position non-linearity

Spot aberration causes minor misalignment of 5 μm which is less than half a pixel, when the maximum resolution of 12.5 μm is applied.

Position non-linearity on the other hand causes serious misalignment of about 600 μm maximum along about a quarter of the scanline. This causes a misalignment of about 50 pixels, when the maximum resolution of 12.5 μm is applied. Position non-linearity has the property of causing static misalignment; e.g. the introduced misalignment is the same for every scanline. Because of this similarity for every inspection run under the same system settings, the misalignment can be pre-determined. This offers the possibility to compensate for this kind of misalignment by means of a look up table. The look up table contains compensation values for the misalignment introduced by position non-linearity for each scan position. The current address of it is determined by the progress of the scan spot along the scanline.

It can therefore be stated that misalignment due to deviations in the acquisition system, is such that it can be easily compensated for yielding a maximal misalignment of about half a pixel.

6.3 Product deformation

Fabricating multi-layer PCBs always yields a certain variance in production. However, the resulting misalignment is expected to be small and can be neglected.

Production deformation may however also be introduced on purpose. The pressure which is to be applied during putting the innerlayers together, causes deformation. By giving the patterns on the film as produced by the plotter a deformation in the opposite direction, the deformation can be compensated for. The inner layers have to be inspected before being put together, so at the moment of inspection this deformation with respect to the CAD reference data still exists.

The intended product deformation can be up to 0.1% which yields a misalignment of 300 μm at 30 cm (scan length), which corresponds with 24 pixels.

6.4 Design specifications

In overall, when scan resolution of 12.5 μm is applied, misalignment over 300 mm (scanlength) may comprise:

- about 40 pixels deviation due to translation
- about 20 pixels deviation due to rotation
- about 24 pixels deviation due to product deformation

The first cause is static, the last two may contain both a static and a dynamic component.

In vertical direction PCBs can be up to 600 mm. This yields misalignment which may amount to:

- about 40 pixels deviation due to translation
- about 40 pixels deviation due to rotation
- about 48 pixels deviation due to product deformation

Corrected has therefore to be for misalignment which may amount up to 128 pixels. Dynamic misalignment maximally yields 44 pixels / 300mm.

7 Multi-level input binary template matching

To be able to detect the presence of certain objects or meaningful parts of objects which serve as alignment positions in a scene, a feature extraction operator is required. The approach to scene matching called template matching is perhaps the simplest. Given a template of a scene, determine the part of the scene matching this template.

In practice, image quality is mainly determined by lighting, background and object reflectivity. Satisfactory image quality can be achieved if images contain foreground and background levels which are sufficiently even, and contrast between objects and their background is sufficiently high.

Although the three factors just mentioned do not apply for images acquired by the 3D laser scanner, the criteria which these images must meet are the same. Throughout the discussion of template matching techniques, there will be made no distinction between images obtained using different kind of acquisition techniques. We thus do not make any difference between images based on intensity information and images based on both height and intensity information. To avoid misunderstanding, a general nomenclature will be introduced in the following section.

7.1 Picture function

A *picture function* f(x,y) is a real-valued function of two variables, having values that are non-negative and bounded, that is $0 \leq f(x,y) \leq L_f-1$ for all (x,y). When a picture is digitized, the resultant digital picture function f, or simply digital picture or image, can be represented by an array of picture elements, or pixels. The digital picture function f can be regarded as a mapping from $\{0,..,M_f-1\}$ x $\{0,..,N_f-1\}$ to $\{0,..,L_f-1\}$. The set $\{0,..,L_f-1\}$ is called the grey level set, and L_f is the number of distinct grey levels. In our case L_f=256. The two integers M_f and N_f represent the size of the digital picture f. Each point (x,y) in this rectangular grid has a pixel value denoted by f(x,y).

7.2 Basic template matching

Suppose the shape of an upper left-hand corner of a white square as presented in Fig. 29.a. is considered to be a template and its contents is defined by picture function w. A template is also called window or mask, and can be considered to be a small sized image. Given is a part of a larger image f_1 as presented in Fig. 29.b. It is desired to determine the location where template w matches a part of the larger image f_1.

The upper left-hand corner of a template and the image to operate upon will be taken as the two reference coordinate systems. The location of the detail which is looked for may then be specified by two translation parameters with respect to the large image coordinate system. For example, consider the template w. If the template is first placed at the (0,0) location in the larger image f_1, and translated 3 units in the x and 3 units in the y directions, then the two patterns will match exactly. Also note that the template cannot be superimposed at every location in the large image. A strip of N_w-1 pixels wide at the right-hand side and M_w-1 high at the bottom side of the large image cannot be operated on, due to the template's size of M_wxN_w elements.

A simple measure of similarity at each test location could now be, to both count the "1" and "0" matches. An exact match would in this case result in a count of 16 and all mismatch locations would produce a count less than 16. The count operation can be considered to be a picture function, and the output is a grey-level intermediate "count" image as presented in Fig. 29.c.

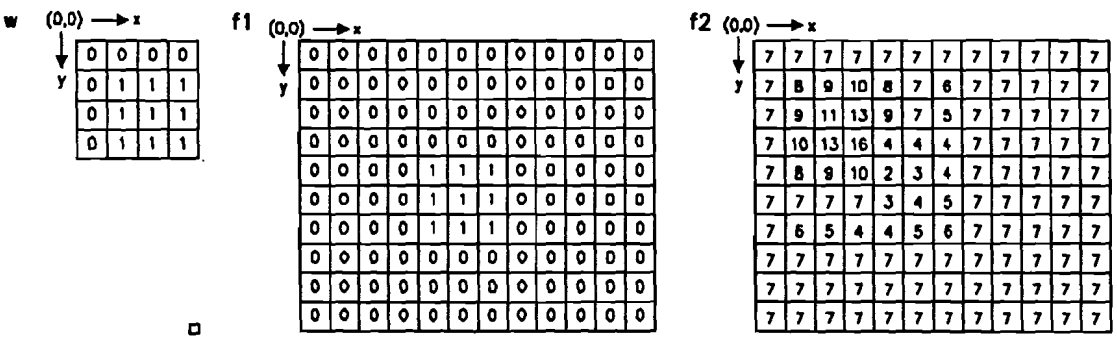


Fig. 29.a. *Template* **Fig. 29.b.** *Scene containing the pattern* **Fig. 29.c.** *Count image*

The template matching technique previously discussed however requires binary images as input. In order to match full grey level images, these images must first be transformed into binary images. This transformation can be done by performing a thresholding operation.

The thresholding operation T_t transforms a full grey-level image into a binary image by applying a cutoff threshold to the original image as follows:

$$T_t(x,y) = \begin{cases} 1 & if\ f_1(x,y) \geq t \\ 0 & otherwise \end{cases}$$

For example. Consider the full grey scale picture fg as presented in Figure 30. Applying the thresholding operation with threshold value 100 on fg yields fg_1 which is equal to f_1; fg_1= T_{100}(fg). The intermediate "count" image" obtained is of course the same as f_2 as presented in Fig. 29.c. on page 45.

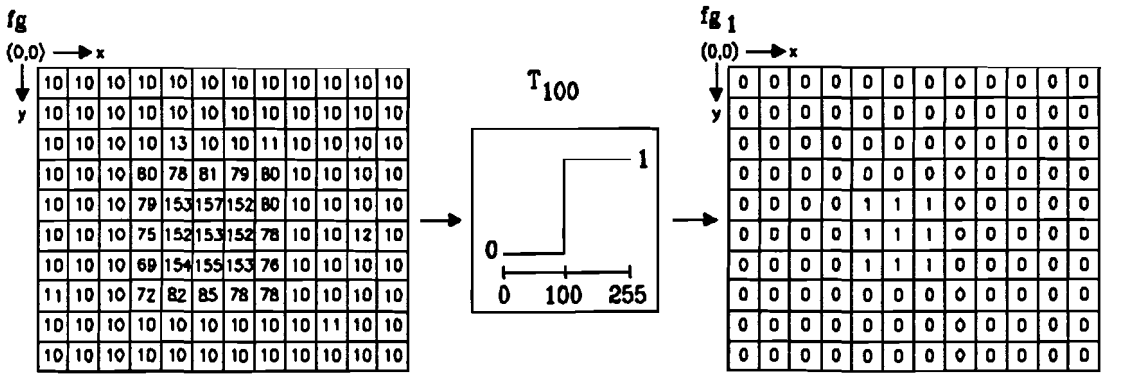


Figure 30. *The thresholding operation*

7.3 Binary template matching

Images usually contain edges which are not perfectly sharp. Among the causes we mention noise and over-talk of adjacent scanning elements. For more details see Chapter 3 discussing image acquisition. For this reason, acquisition of scenes containing the same patterns and acquired under the same lighting conditions, may yield images showing small differences. At first sight, these differences seem to be negligible. When performing a thresholding operation upon these images in order to perform template matching, boundary curvature of crossings between dark and light areas of the various patterns sampled, are however not uniquely defined. In order to overcome the problem of unsharp edge transition, don't care template elements are being introduced. The template matching method now obtained, is called binary template matching.

Binary template matching is based on two templates, whose elements are binary. One of these templates is the "white/black" template w; "ones" represent white pixels, "zeros" represent black pixels. The other one is the "do care"/"don't care" template d, "ones" represent do care elements, "zeros' represent don't care elements. The templates are finite in size and have the same dimensions: $M_d x N_d = M_w x N_w$. The combination of template d and w is represented as template t. See Figure 31.

Matching is as follows. Both templates w and d are put on the input image and shifted synchronously from pixel to pixel. While doing so, at each location a counter is incremented for every input pixel that corresponds with a do care element of template d, and have the same value as the corresponding element in template w. The value of input pixels corresponding with don't care elements of template d, does not matter.

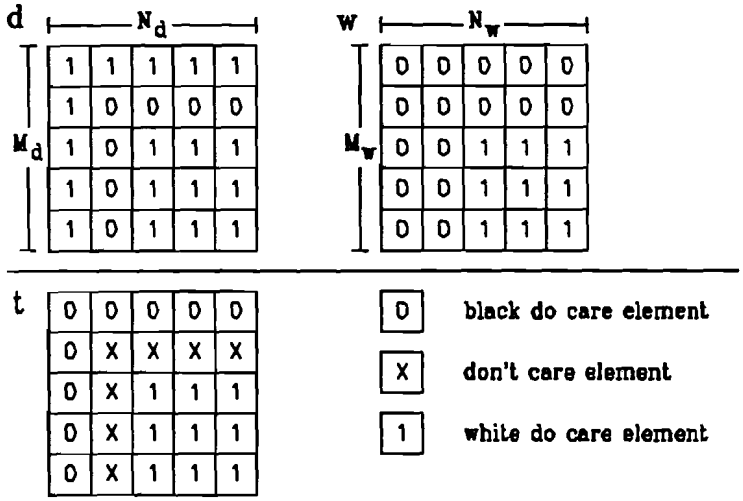


Figure 31. *Mask for binary template matching*

$$\text{template } d, \text{ sized } M_d x N_d,\ L_d = 2$$
$$d(i,j) = \begin{cases} 1; & \text{template element } i,j \text{ is do care} \\ 0; & \text{template element } i,j \text{ is don't care} \end{cases}$$

$$\text{template } w, \text{ sized } M_w x N_w,\ L_w = 2$$
$$w(i,j) = \begin{cases} 1; & \text{template element } i,j \text{ is white, if } d(i,j) = 1 \\ 0; & \text{template element } i,j \text{ is black, if } d(i,j) = 1 \end{cases}$$

Picture function $o_b(x,y)$ specifies the binary template operator. Before giving its definition, we first introduce the intermediate picture function $o_g(x,y)$ which generates a intermediate "count" image. See Figure 32.b.

In formula:

$$o_g(x,y) = \sum_{k=0}^{M_d-1} \sum_{l=0}^{N_d-1} \{\neg(i(x+k,y+l) \oplus w(k,l)) \wedge d(k,l)\}$$

So it follows that:

$$0 \le o_g(x,y) \le \sum_{k=0}^{M_d} \sum_{l=0}^{N_d} d(k,l) = S$$

S is the number of do care elements in template d.

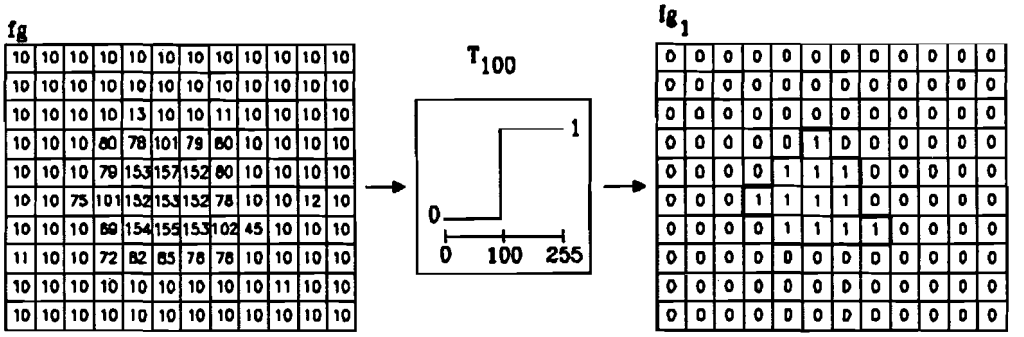


Figure 32.a. *Tresholding operation*

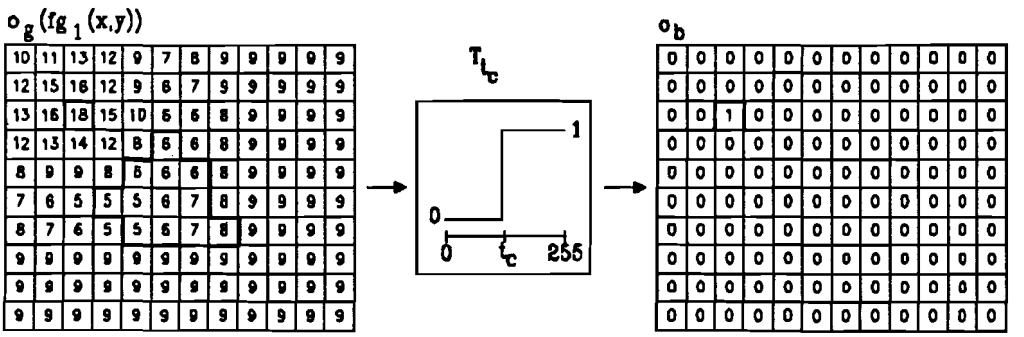


Figure 32.b. *Thresholding the count image, $t_c = S$*

Occurrences of similarity can be defined to be at those positions where $o_g(x,y)$ equals S, which requires complete matching of input image and template. However, in order to be able to cope with noise and deformations in an other way than by introducing don't cares, a recognition or match is defined to be at those positions where $o_g(x,y)$ is higher than or equal to a correlation threshold t_c (this is not to be confused with the threshold for obtaining a binary input image). So the binary output $o_b(x,y)$ of binary template matching is defined as:

$$o_b(x,y) = \begin{cases} 1; & if\ o_g(x,y) \geq t_c \\ 0; & if\ o_g(x,y) < t_c \end{cases} \qquad 1 \leq t_c \leq S$$

which equals

$$o_b(x,y) = T_{t_c}(o_g(x,y)) \qquad 1 \leq t_c \leq S$$

7.4 Split binary template matching

Although binary template matching suffices for many situations in practice, it however requires the number of black do care elements to be in the same order as the number of white do care elements. Especially if some tolerance regarding noise is to be allowed. Consider for example the detection of a thin line using templates d_{tl} and w_{tl} and sized 5x7 pixels^2 as presented in Figure 33.a. The combination of the templates d_{tl} and w_{tl} is given in Figure 33.b.

A complete match requires that all of the 25 do care elements match with corresponding pixels of the input image. However, if we want to tolerate some noise, we can lower the correlation threshold. If five noise pixels are allowed then the presented template also match on a constant black input, which is of course not desirable. Making the thresholds one higher does not help too much, because now isolated white pixels are also considered to be white lines.

To obtain better control over the matching operation, the template is now splitted into a black part and a white part with separate correlation thresholds as is the case in Figure 35. If the correlation threshold for the white sub-template is set such that it allows one noise pixel and the threshold for the black sub-template for four noise pixels, then again five noise pixels are allowed in total, but constant black areas or isolated white pixels will not give rise to false alarms any more.

(a) d_{tl}

1	1	0	1	0	1	1
1	1	0	1	0	1	1
1	1	0	1	0	1	1
1	1	0	1	0	1	1
1	1	0	1	0	1	1

w_{tl}

0	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	0	1	0	0	0

(b) complete template

0	0	X	1	X	0	0
0	0	X	1	X	0	0
0	0	X	1	X	0	0
0	0	X	1	X	0	0
0	0	X	1	X	0	0

Figure 33. *Thin line detection, the number of black do care elements is not in the same order as the number of white do care elements.*

7.5 Multi-level input binary template matching

Binary template matching but also split binary template matching have certain inadequacies when applied on real world images. They both require binary images as an input image. This can be achieved by performing a thresholding operation upon the original grey-level input image.

Threshold levels can be pre-determined globally, i.e. can be applied for the whole image. This works well if the lighting of the scene is sufficiently even and the objects have a sufficiently high contrast with the background. If the scenes to be imaged meet these requirements, then it is usually possible to set a static video threshold in advance and then there is no problem obtaining suitable binary images. Setting the video threshold in advance of course requires additional processing.

Most real world images however, are not that stable, so they can not be processed using a static threshold. These images often require their threshold to be determined locally, i.e. the threshold for a pixel at location (x,y) is determined by its grey-level and by the grey levels of its surrounding pixels. For example, the threshold is being determined by taking the mean of all the pixel values in a window. The method however has some drawbacks due to the fact that the window chosen is limited in size yielding a limited view of the scene. For example a staircase intensity profile will result into occurrences of false edges. See Figure 34.a. At so called "three-land points", it is almost impossible to transform the input image into a proper binary image. See Figure 34.b.

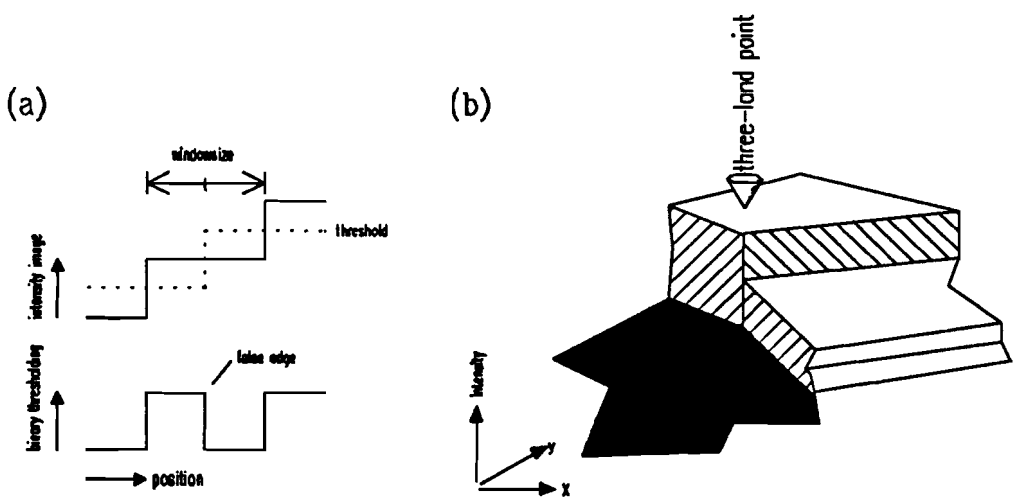


Figure 34. *(a) Staircase intensity profile*
(b) Three land points

In order to overcome these problems, thresholding the input image with every static video threshold possible (in our case 256) and to process each of these images with binary correlation, could be a solution. This however either takes a lot of time if this is done serially per video threshold, or it takes a lot of hardware, if all the possible binary images are processed in parallel.

At Philips Nat. Lab. and Philips CFT, a patented method [6] has been developed which performs the equivalent of binary template matching using every threshold possible, in real time and with rather modest amount of hardware. The method is called multi-level input binary template matching, the hard-ware implementation of the operator dynamic range correlation (DRC).

7.5.1 Principles of multi-level input binary template matching

Multi-level input binary template matching is based on split-template binary template matching using the sub-templates w_w and w_b. These are matching templates for white and black pixels respectively. Both kernels have the same dimensions, M by N pixels and a "grey-level" set L which consists of two elements, L= {1,0}. The ones represent white or black do care elements in template w_w and w_b respectively, the zeros don't care elements. Figure 35. shows for example a template for recognizing an upper left-hand corner of a light object. The various steps to be performed are schematically drawn in Figure 36.

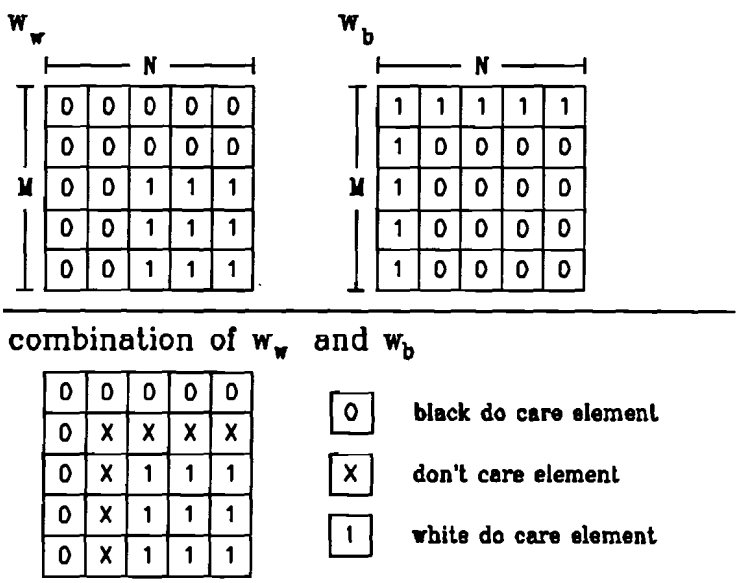


Figure 35. *Mask for multi-level input binary template matching*

An input image i_g is thresholded using the video threshold t_{vw} in order to obtain a binary input image i_w for white pixels template matching using sub-template w_w.

The input image i_g is also thresholded using the video threshold t_{vb} in order to obtain a binary input image i_b for black pixels template matching using sub-template w_b.

Shifting template w_w pixel by pixel over binary image i_w, and counting the pixels corresponding with do care elements, results in the white count image m_w.

Shifting template w_b pixel by pixel over binary image i_b, and counting the pixels corresponding with do care elements, results in the black count image m_b.

The count images m_w and m_b are thresholded by correlation thresholds t_{cb} and t_{cw} respectively, and obtained are the recognition images o_{bw} and o_{bb}.

The correlation thresholds t_{cb} and t_{cw} have to be set to reasonable values. This is somewhere between one (at least one matching element required) and the value required for a complete match, i.e. the number of do care elements contained in the corresponding templates. In order to obtain a match at position (x,y), both counting images must have pixel values which exceed their correlation thresholds.

In formula:

Binary images obtained by thresholding:

$$i_w(x,y) = T_{t_{vw}}(i_g(x,y))$$

$$i_b(x,y) = T_{t_{vb}}(i_g(x,y))$$

Convolution of the sub-templates results into the counting images:

$$m_w(x,y) = \sum_{k=0}^{M_{w_w}} \sum_{l=0}^{N_{w_w}} t_w(k,l) \cdot i_w(x+k,y+l)$$

$$m_b(x,y) = \sum_{k=0}^{M_{w_b}} \sum_{l=0}^{N_{w_b}} t_b(k,l) \cdot \bar{i}_b(x+k,y+l)$$

In order to obtain a match at position (x,y), $m_w(x,y)$ and $m_b(x,y)$ must exceed their correlation thresholds.

$$m(x,y) = T_{t_{cw}}(m_w(x,y)) \wedge T_{t_{cb}}(m_b(x,y))$$

$$o_b(x,y) = o_{bw}(x,y) \wedge o_{bb}(x,y)$$

The steps presented in schematic form yields:

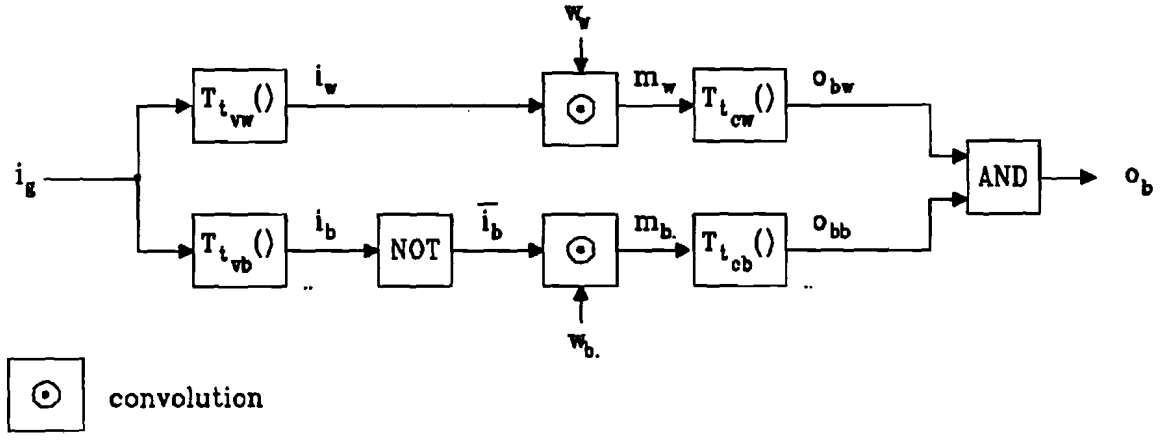


Figure 36. *Split binary template matching*

As already mentioned, the problem of template matching of a scene with non-even and non-stable foreground and background levels can be solved by thresholding the input image with every video threshold possible. The video threshold t_{vw} and t_{vb} must be set to every value contained in the grey-level, and the obtained binary images must be processed according previously given formulas. This however requires a lot of hardware and/or processing time, which is of course not feasible in practical situations.

Instead of thresholding the input images for all possible threshold values and process the obtained binary images, we can also search for the thresholds for which the input image matches the separate sub-templates.

7.5.2 Multi-level input matching for a complete match

Suppose the input image for white pixels template matching is obtained by performing a thresholding operation using the lowest value for threshold t_{vw} which is globally valid. The obtained binary image i_w will now only contain ones (and no zeros). This means that the white sub-template w_w, which can be filled with any particular pattern, will match everywhere in this binary image. By increasing the video threshold t_{vw}, the binary image will contain more and more zeros, however, once a pixel has become zero it remains zero when increasing the global video threshold t_{vw}. This means that there exists a local video threshold $t_{lvw}(x,y)$ for each location (x,y) for which the sub-template w_w will match at that particular location, but for $t_{lvw}(x,y)+1$ not any more. This means that sub-template w_w will match at location (x,y) in all the binary images resulting from the application of video thresholds in the interval $[0,t_{lvw}(x,y)]$, but it will not match at that location if a video threshold is applied ranging from $[t_{lvw}(x,y)+1,L-1]$. The grey-level set of i_g consists of L elements.

With a similar explanation it can be clarified that the sub-template for black pixels w_b will match on binary images $\bar{i}_b(x,y)$ resulting from the application of all video thresholds in the range of $[t_{ivb}(x,y), 255]$, but it will not match on those locations where a video threshold is applied in the range of $[0,t_{ivb}(x,y)-1]$.

If there is an overlap between the intervals $[0,t_{ivw}(x,y)]$ and $[t_{ivb}(x,y),255]$, then it can be said that the combination of sub-templates w_w and w_b will match at location (x,y) for those binary images i_w and i_b which have been derived using global threshold t_{vw} and t_{vb} respectively and t_{vw} and t_{vb} lie within the interval $[t_{ivb}(x,y),t_{ivw}(x,y)]$. Thus, there are $t_{ivw}(x,y)-t_{ivb}(x,y)+1$ thresholds t which yield a match of $T_t(i_g))$ at location (x,y) for both sub-templates w_w and w_b. The number of thresholds t is denoted $N_{tv}(x,y)$ and is called the *range*. $N_{tv}(x,y)$ is location dependent.

Multi-level input binary template matching adopts $N_{tv}(x,y)$ to be the measure of similarity at location (x,y). In order to have a match of the total template at location (x,y), $N_{tv}(x,y)$ should be greater than zero or a presetted positive value. In order to derive $N_{tv}(x,y)$, the local thresholds $t_{ivw}(x,y)$ and $t_{ivb}(x,y)$ have to be determined.

For determining the so called range efficiently, rank-value filters are applied in the DRC operator. Rank-value filters perform real-time ranking of pixel values which are inside a window with a maximum size of 8x8 pixels[2]. The operation of the rank-value filter is such that it replaces the don't care elements of the sub-templates by zero and sorts the pixel values of the remaining elements in ascending order; counting starts at zero.

The local thresholds $t_{ivw}(x,y)$ and $t_{ivb}(x,y)$ can be determined as follows. Suppose we are using a black sub-template w_b with correlation threshold t_{cb} and a white sub-template w_w with correlation threshold t_{cw}. The correlation thresholds are set to a value which requires a complete match for both sub-template matching, i.e. they are equal to the number of do care elements in the corresponding sub-templates.

Template matching for 2D images will now be explained. For ease of drawing, a 1D edge transition is presented as example in Figure 37.a. It can be used without detracting the validity of the 2D explanation.

Figure 37.a. shows an edge transition, and the template to be applied. Two sets, a set of black pixels and a set of white pixels are distinguished. These sets correspond with the black respectively white do care pixels of the applied template.Applying a video threshold $t_{ivb}(x,y)$ which is one higher than the one yielding the maximum of a set of pixels, results in a binary image that is black for all of these pixels. If this set of pixels corresponds to the set of the do care pixels of the black sub-template w_b (the elements which are set to one in the template) positioned at (x,y), then we have a complete match using this video threshold. See Figure 37.b. for a 1D situation.

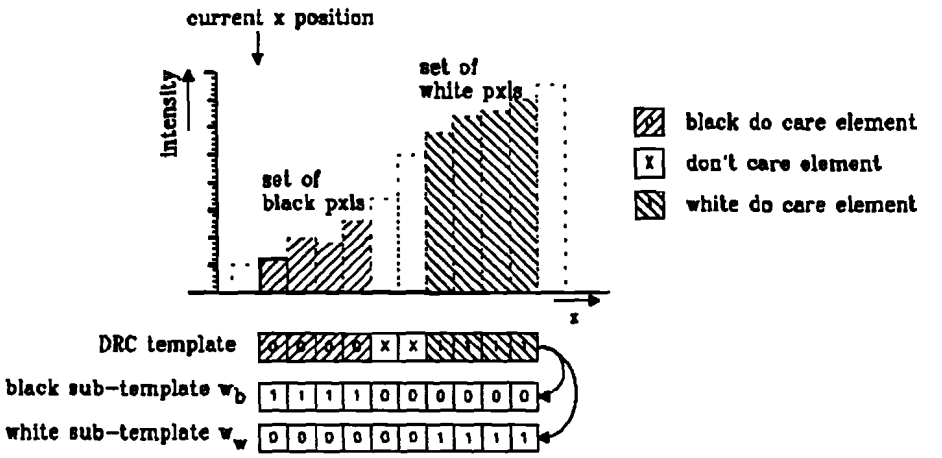


Figure 37.a. *Multi-level input binary template matching at location x.*

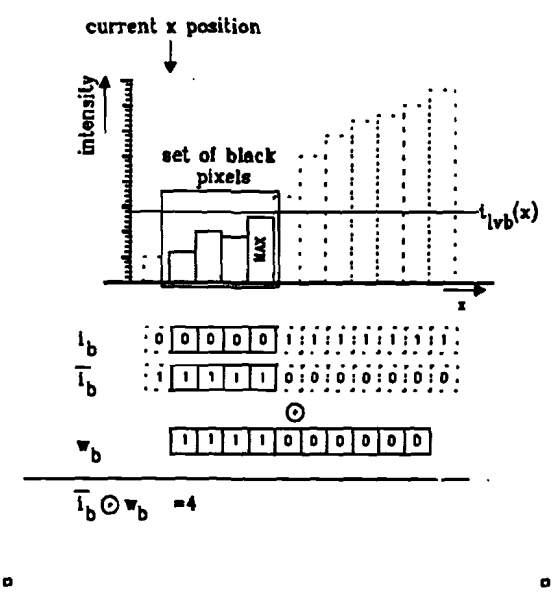


Figure 37.b. *$T_{lvb}(x)$ is one higher than the maximum of the set of black pixels*

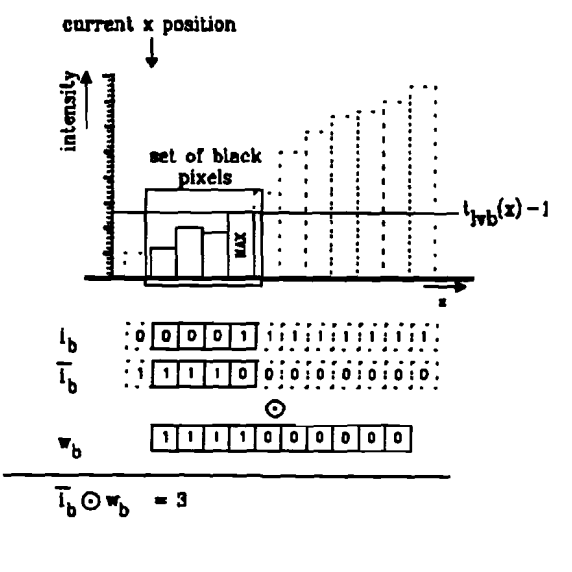


Figure 37.c. *Lowering $T_{lvb}(x)$ yields non-complete matching*

If the video threshold is lowered with one, then at least the pixel which has the maximum values will become white in the resulting binary image. This means that we do not have a complete match any more and that therefore the maximum of the input pixels corresponding with the do care pixels of the black sub-template is just one lower than the $t_{lvb}(x,y)$ we are looking for. See Figure 37.c.

For similar reasons, applying a video threshold $t_{lvw}(x,y)$ which is the minimum grey value of a set of pixels results into a complete match of the white sub-template w_w. See Figure 37.d. Increasing the threshold $t_{lvw}(x,y)$ by one results into a binary image that is white for all of these pixels, except for the pixels which have the minimum value. See Figure 37.e. Therefore, the minimum of the input pixels corresponding with the do care pixels of the white sub-template is the $t_{lvw}(x,y)$ we are looking for.

In formula:

$$t_{vb}(x,y) = 1 + \underset{\forall_{(k,l)\,:\,w_b(k,l)=1}}{\mathrm{MAX}} \{i_g(x+k, y+l)\}$$

$$t_{vw}(x,y) = \underset{\forall_{(k,l)\,:\,w_w(k,l)=1}}{\mathrm{MIN}} \{i_g(x+k, y+l)\}$$

The range $N_{tv}(x,y)$ or the number of video thresholds for which both sub-templates match completely equals:

$$N_{t_v}(x,y) = t_{vw}(x,y) - t_{vb}(x,y) + 1$$

In order to have a match of the total template at location (x,y), $N_{tv}(x,y)$ should be greater or equal to a presetted positive number N_{tv} greater than one.

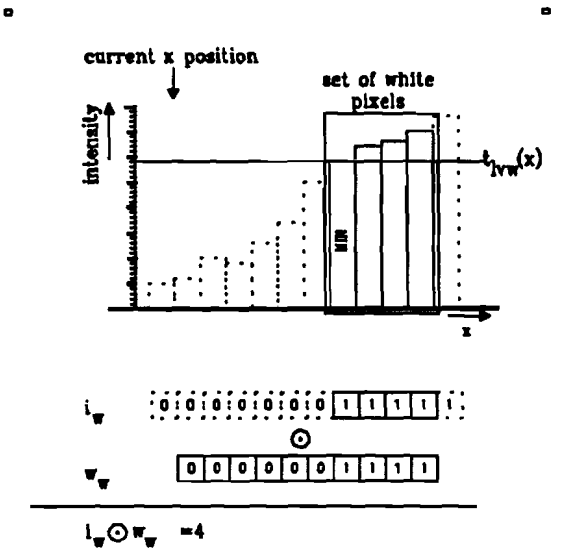


Figure 37.d. *$T_{lvw}(x)$ is the minimum of the set of white pixels*

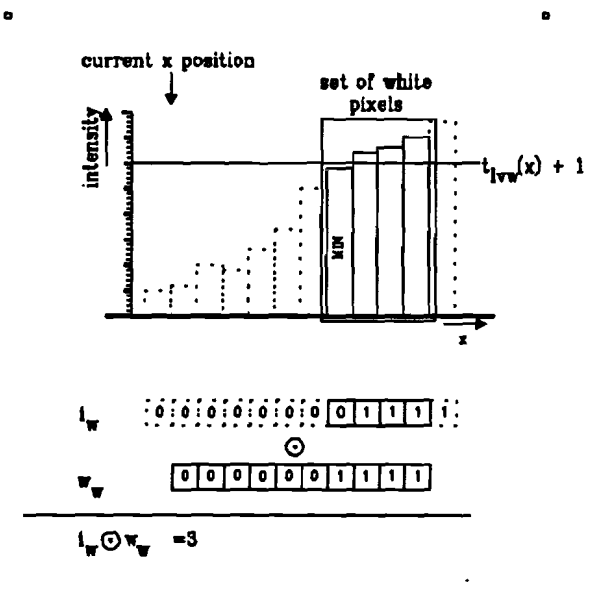


Figure 37.e. *Increase of $T_{lvw}(x)$ yields none-complete matching*

7.5.3 Multi-level input matching for a non-complete match

Up till now, a complete match for the separate sub-templates was required in order to obtain a match. This constraint may be relaxed in order to cope with noise and deformation. By allowing m_b mismatches for the black sub-template and m_w for the white sub-template, the constraints on the correlation thresholds t_{cb} and t_{cw} are being relaxed.

In order to do so, first the value *maxbut n* and *minbut n* are being defined. The operation of the rank-value filter is such that it replaces the don't care elements of the sub-templates by zero and sorts the elements in ascending order; counting starts at zero. Introduced are now:

- $R_b(x,y)$ the set consisting of the pixels $i_g(x+i,y+j)$ for which $w_b(i,j)=1$.
- $R_w(x,y)$ the set consisting of the pixels $i_g(x+i,y+j)$ for which $w_w(i,j)=1$.
- $N\{S\}$ the number of elements of the set S, and
- RANK(n,S) the n^{th} element of the set S sorted in ascending order. Indexing starts with 0.

The *maxbut n* value of a set S equals RANK(N{S}-1-n,S); the *minbut n* value of a set S equals RANK(n,S).
In order to allow m_w noise pixels for the white sub-template w_w, $t_{lvw}(x,y)$ should be the *minbut n* element of $R_w(x,y)$. This means that there can be zero to m_w pixels with a value lower than or equal to this threshold, but certainly not more than m_w pixels. How much lower is not defined. In this way it is possible to cope with dips, i.e. noise pixels with extreme lower grey-level compared to its neighbouring pixels. See Figure 37.f. Increase of $t_{lvw}(x,y)$ yields a larger range $N_{lv}(x,y)$.

A similar explanation can be given for $t_{lvb}(x,y)$ when m_b mismatches are allowed. See Figure 37.g.

In formula:

$$t_{lvb}(x,y) = 1 + RANK((N\{R_b(x,y)\} - 1) - m_b, R_b(x,y))$$

$$t_{vw}(x,y) = RANK(m_w, R_w(x,y))$$

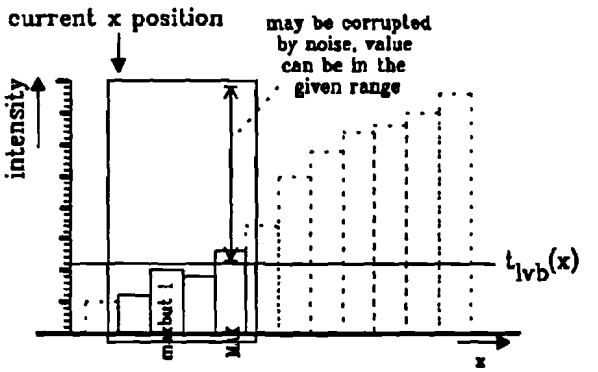


Figure 37.f. *Non-complete matching by setting $T_{lvb}(x)$=one higher than maxbut one*

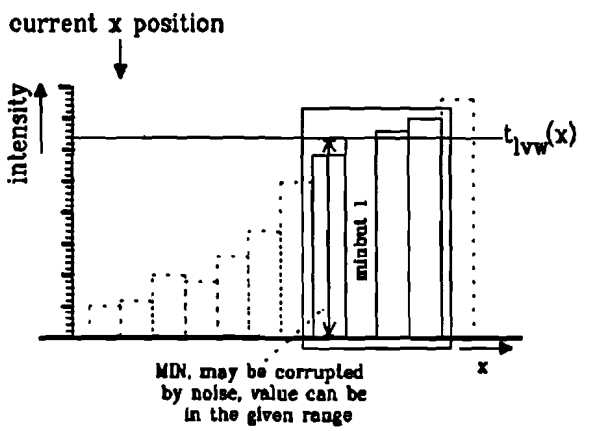


Figure 37.g. *Non-complete matching by setting Tlvw(x) minbut one*

7.6 Hardware implementation as a real-time operator

The hardware implementation of the multi-level input binary template matching makes use of a programmable rank-value filter (L64220 of LSI). This filter sorts all the values within a window and outputs the value of interest as determined by the RANK selector. Commonly used rank values are the maximum or minimum value within a set of inputs; i.e. $t_{vw}(x,y)$ and $t_{vb}(x,y)$-1 for complete template matching at location (x,y). Any element in the window can be masked by setting any of the 64 masks bits LOW. Masking enables the programming of any shape of the sub-templates in the processing window. The N unmasked data values in the window are sent to the RANK selector. The RANK selector outputs the value with the rank that is stored in a particular control register. The maximum value always has a rank of 63, while the minimum has a value of 64-N, N is the number of valid input values as specified by the mask.

Basically, the implementation of multi-input binary template matching in hardware consists of two rank-value filters in order to determine the proper video thresholds $t_{vw}(x,y)$ and $t_{vb}(x,y)$-1, line delay circuitry in order to provide for the input, and summation and comparison circuitry for processing the output and determination of the recognition result. See Figure 38.

Applying a RANK selector which is able to perform full ranking, i.e. every rank in the sort can be selected to be the output, enables both complete matching as non-complete matching. By allowing only a restricted range of ranks, considerable hardware savings is achieved. In order to perform complete multi-input level binary template matching, only the minimum or the maximum of a set of specified input values in the window has to be passed. Multi-level input binary template matching requiring a complete match result of the two sub-templates however effects the method's insensitiveness towards noise. The requirement of a complete match however is expected to yield a higher position measuring accuracy. The exact relations of one to another have to be evaluated.

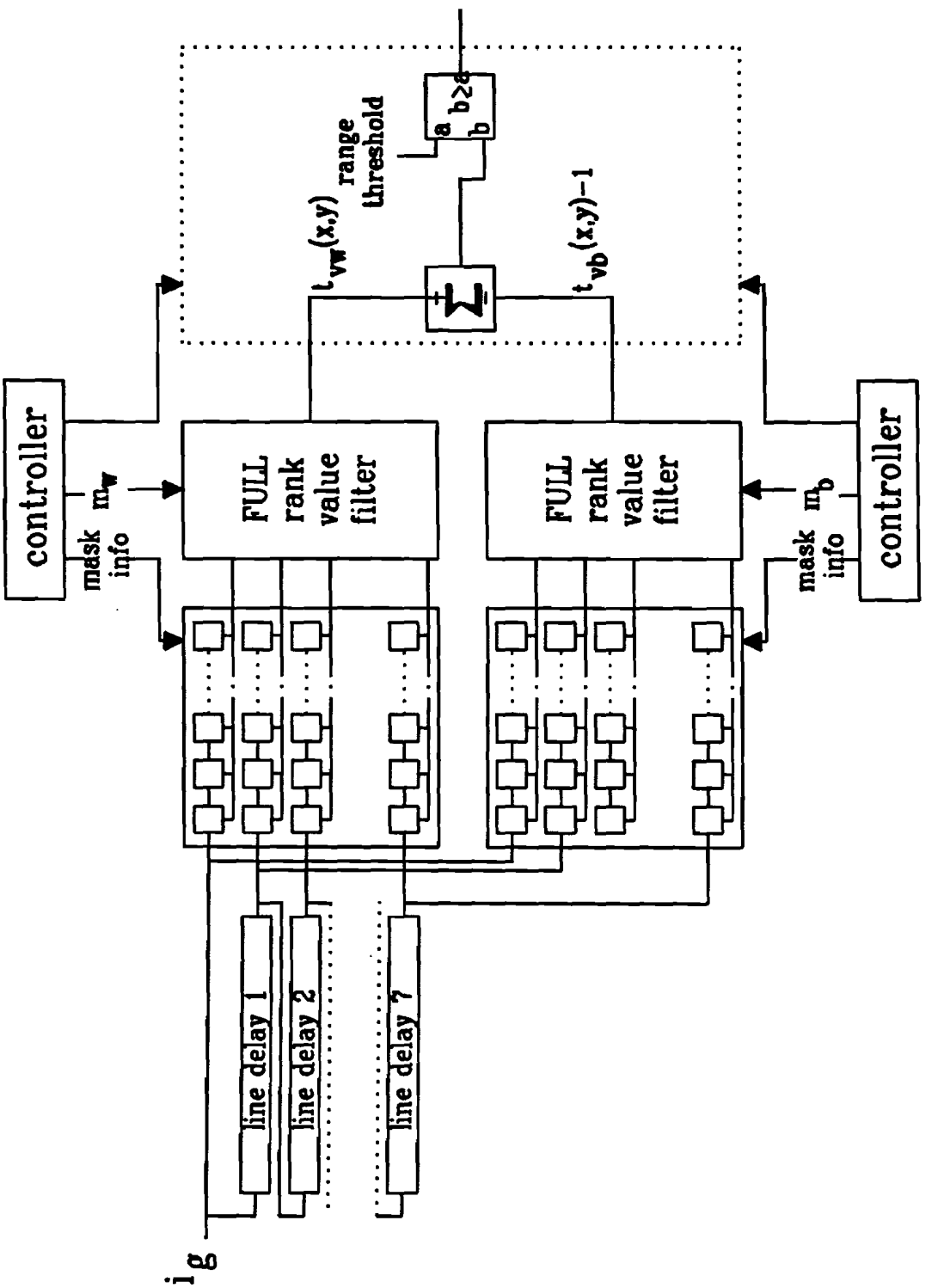


Figure 38. *Hardware implementation of Dynamic Range Correlation*

8 DRC evaluation

8.1 Introduction

Alignment features basically comprise patterns contained in the PCB under test. The position of these patterns are determined by applying multi-level input binary template matching. In Chapter 7 multi-level input binary template matching has been discussed. The presented method overcomes certain drawbacks of binary template matching with a pre-determined static (video) threshold, or other derived, more sophisticated approaches.

Multi-level input binary template matching performs binary template matching on grey-level images. This is done by virtually thresholding the input image using every (video) threshold possible and summing the detection results for each pixel. An evaluation has to be performed whether this method is suitable for pattern recognition purposes in view of dynamic alignment of CAD with respect to image data. The ultimate goal of the feasibility analysis should be an evaluation scheme presenting the necessary requirements and criteria in order to conclude whether multi-level input binary template matching is suitable or not.

8.2 Requirements for dynamic alignment

The requirements for template matching methods and object recognition methods in general are quite obvious:

a) maximize the recognitions rate.
b) minimize the false alarms rate.
c) minimize the missed recognitions rate.

If position information is important as is the case with template matching for alignment purposes:

d) obtain high position accuracy.

If real-time dedicated hardware is involved we also have to consider:

e) the minimization of cost and amount of hardware.

Applied to multi-level input binary template matching, most of these requirements are however in contradiction with each other. In order to implement the method for alignment purposes, we want to evaluate certain properties of the template matching method. The alignment measurement unit will rely on these properties.

8.2.1 Property I

Multi-level input binary template matching is to be used to measure the location of feature patterns contained in the image of the PCB under test. In order to evaluate whether the method is suitable to be applied for dynamic alignment, an estimation of its measuring accuracy must be provided for.

In ideal situations, the templates to be used do not contain any don't care elements at all, and recognition requires a complete match of the applied sub-templates w_w and w_b. Due to noise, deformation and resolution limitations, don't care elements have to be introduced in order to still detect patterns which resemble to a certain extent.

Another possibility to deal with noise, deformation and resolution limitations, is by allowing non-complete matches of the separate sub-templates, i.e. the correlation thresholds are set $m_w>0$ or $m_b>0$.

Both provisions effect the matching result compared to ideal template matching. Instead of obtaining a single recognition point, we now may obtain a recognition blob. This may affect the accuracy of position measurement. See Figure 39.a.

The linear scan pattern of the 3D laser scanner yields the acquired pixels to be offered to the data processing unit in a data stream of 15 Mpixels/s. Due to the huge amount of picture information, data processing is basically performed pixel by pixel in real time. This data stream can be considered as the input image of the dynamic range correlator, which can be considered as a picture function. The output image may now contain recognition blobs. A recognition blob however contains multiple matches referring to the same entity. Without proper processing, these matches are considered to be separate occurrences of entities. A recognition blob has therefore to be labelled to one entity.

In order to solve this problem, a picture function r(x,y) is introduced, which transforms an image containing recognition blobs into a picture in which the recognition blobs are replaced by recognition points (r stands for reduce recognition blob to recognition point). The way in which this reduction is performed, will effect the measuring accuracy.

Considered is a recognition blob as presented in Figure 39.a. Image pixels are offered sequentially according the linear scan pattern adopted by the 3D laser scanner. Images as presented in this report are "scanned" line by line from left to right, and the scanline's progression is downwards.

Perhaps the most common approach to perform such a reduction is implemented as picture function $r_1(x,y)$. Picture function $r_1(x,y)$ considers the position of the first match of a recognition blob to be encountered during scan, as the location at which recognition occurred. To obtain the recognition image r_1, the match first encountered has to be passed to the next

stage in the pipeline, the remaining matches contained in the recognition blob have to be removed. See Figure 39.b.

Higher measuring accuracy can be obtained by considering the centre of gravity of a recognition blob to be the measured position. Determining the centre of gravity of recognition blobs on the other hand puts impracticable computational demands. It is however of no importance at all to determine exactly the centre of gravity of a recognition blob, an approximation within a pixels of it suffices.

An estimation of the centre of gravity of a recognition blob can be obtained by applying the picture processing operations dilate and erosion with stop criteria. This will be discussed in Appendix 2.

The picture function $r_2(x,y)$ transforms an image containing recognition blobs into an image in which the blobs are replaced by an estimation of their centre of gravity points. See Figure 39.c.

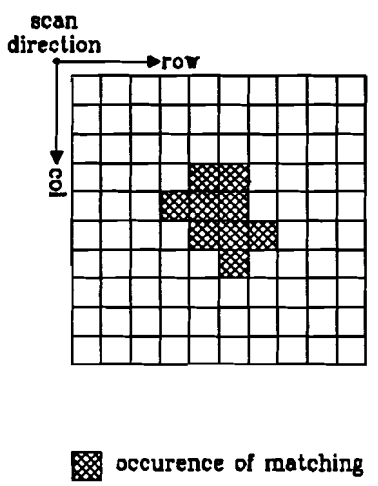


Figure 39.a. *Recognition blob to be labelled to one entity*

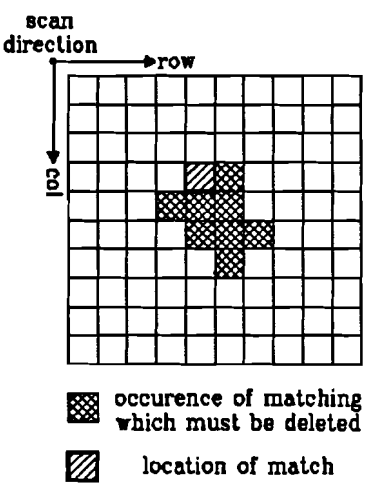


Figure 39.b. *Labelling to one entity by $r_1(x,y)$*

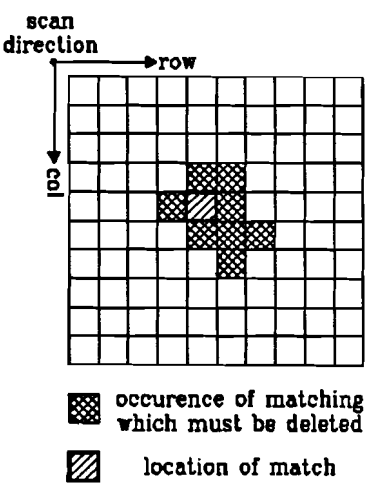


Figure 39.c. *Labelling to one entity by $r_2(x,y)$*

The concepts for implementing the picture functions $r_1(x,y)$ and $r_2(x,y)$ are discussed in Appendix 2. It is quite obvious that the implementation of $r_1(x,y)$ requires less hardware than the implementation of $r_2(x,y)$. Determining the centre of gravity of recognition blobs requires additional hardware. Expected is however a gain in position measuring accuracy. An indication in what proportion these two matters stand one to another is therefore needed.

Also necessary is an estimation of the measuring accuracy which can be achieved by multi-level input binary template matching

8.2.2 property II

As described in previous chapter, multi-level input binary template matching is able to cope with noise and deformation, by allowing non-complete matches and introducing a small band of don't cares. See Figure 40.b.

Template matching is then such that white sub-template matching using template w_w allows m_w pixels to be corrupted, and black sub-template matching using template w_b allows m_b pixels to be corrupted. Non-complete matching however requires the RANK selector to be equipped with the possibility of selecting the *maxbut* n_b or *minbut* n_w value among the ranking of the selected input values as specified by the corresponding sub-templates w_b and w_w; n_b= m_b and n_w= m_w.

Complete matching on the other hand, only requires the RANK selector to be equipped with the possibility of selecting the maximum and minimum value out of the ranking of the selected input values specified by the sub-templates. This results in a considerable decrease of hardware requirements.

The hardware savings introduced result in a certain inflexibility towards the ability to cope with noise and deformation. It is expected that this can be compensated for using a wider band containing don't care elements in between the white do care elements and the black do care elements. See Figure 40.c. A wider band of don't care elements is not expected to inflict the rejection ability of the applied template for other objects than the intended one. Especially not if the number of remaining do care elements is quite large, and the threshold for the range is set quite high.

It has to be verified whether such a band indeed compensates for the "inflexibility" of the complete match requirements of both sub-templates with respect towards noise and deformation. It has also to be verified whether the introduction of a wider band does not influence the position measuring accuracy of the template matching method.

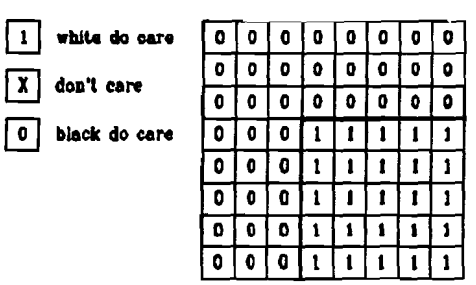


Figure 40.a. *Ir-realistic template for ideal situations*

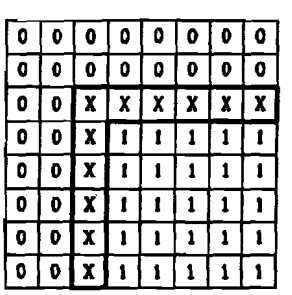


Figure 40.b. *Small band*

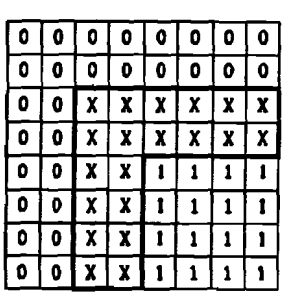


Figure 40.c. *Wide band*

8.2.3 Property III

For reasons which will be explained in later sections, a certain tolerance is allowed in the recognition rate of multi-level input binary template matching as applied in the alignment measurement unit. It is not strived for a 100% recognition rate, rather than it is aimed at avoiding false alarms. Requiring complete match results of the separate sub-templates contributes to this objective. So does the increase of the threshold setting for the range.

It has to be verified that, when the input image is of rather reasonable image quality, performing template matching according above settings still yields acceptable performance.

8.2.4 Crucial question

Are we able to obtain reasonable position measuring accuracy, with reasonable recognition rating, e.g. higher than 90%, when a minimal hardware configuration is applied.

8.3 Evaluation approach

In order to study the behaviour of multi-level input binary template matching, the availability of a well chosen and defined quality criteria, is of importance. This enables the comparison of the matching operator's behaviour under different conditions and parameter settings. The defined quality criteria should act as some suitable chosen distance measure between the actual response of the matching operator applied on real world images, and its ideal response. It is however difficult to define such an ideal response, moreover, in our case, no real world images exists, mere expectations. For these reasons, the need for artificially generated test images, resembling real world images to a certain extent emerges. These test images should be accompanied by reference images showing the response of an ideally operating pattern recogniser (see Figure 41.). Naturally the test images should also reflect the influences of the imaging devices (e.g. noise and blur).

8.4 Test image generation

Images of PCBs can be subdivided into regions representing the appearance of copper material, and regions representing the appearance of substrate material. These regions obtain a foreground grey level respectively a background grey level. When detecting the appearance of copper material, we make use of the characteristic that boundaries between copper and substrate material, show up as grey level in-continuities in the image. It is expected that these boundaries show up as rather steep grey level in-continuities, i.e. a high image quality is obtained.

In order to evaluate multi-level input binary template matching for pattern recognition purposes, a test image i_t comprising above mentioned image characteristics is to be generated together with its corresponding reference image i_r. See Figure 41.

Evaluation concerns the recognition of a particular shaped pattern with mean grey level μ_2, which is surrounded by substrate material with mean grey level μ_1. As in real world images, edge heights are liable to variance. Test images must therefore comprise this edge height variability expressed by means of the signal parameter σ_h. When a particular pattern is placed at equidistant places in the image, the resulting foreground and background grey levels are not constant over the test image. Obtained now is the test scene.

When acquiring the test scene, we have to deal with resolution limitations and spot aberrations of the 3D laser scanner causing image degradation. Also to be considered is noise introduced by the acquisition unit's electronics and noise due to speckles of copper material and dust. These properties build up the image formation model and must be comprised in the test image.

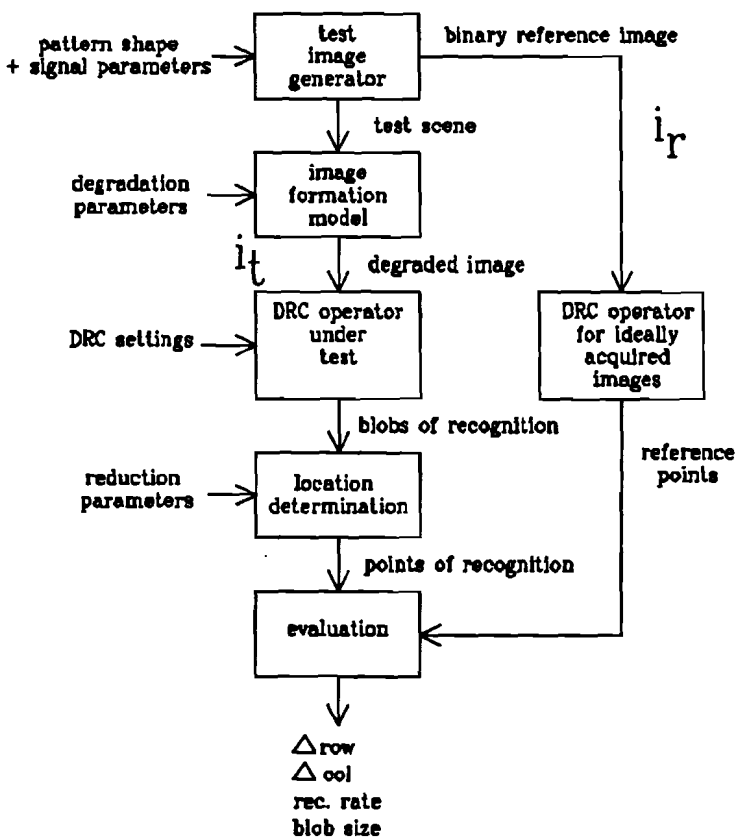


Figure 41. *Evaluation of the DRC operator*

8.4.1 Reference image

The generation of the reference image i_r is started by defining equally sized rectangular black areas, which practically fills the entire screen. The areas are denoted reg(i,j), and are called background regions. The grey value zero is assigned to these background regions. The pattern to be recognised is being drawn in the centre of all of these regions. The grey level assigned is one. The shapes drawn in the background regions reg(i,j) are denoted shp(i,j). The grey value set of the reference image i_r equals $L_{ir}= \{0,1\}$.

We have now obtained a binary reference image i_r which comprises a set of patterns which locations can be incontrovertibly determined using a template which does not contain any don't care elements. The template matching results of this set of patterns serve as reference positions. See Figure 42.

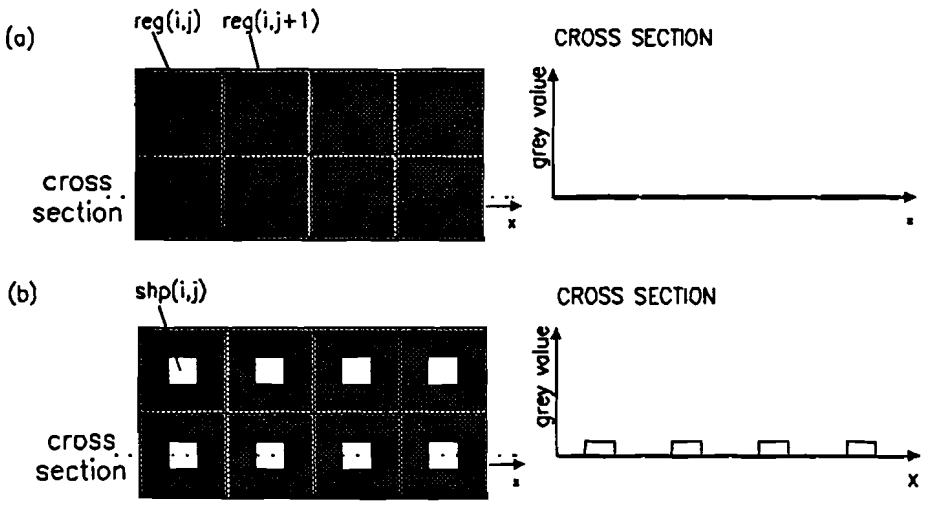


Figure 42. *(a) labelling background region reg(i,j)*
(b) adding patterns

8.4.2 Test image

The actual test image i_t initially equals its reference image i_r; test image i_t is also subdivided into background regions reg(i,j), each containing a pattern shp(i,j).

μ_l σ_h The variance in background grey level as has been brought forward in a previous section, is emulated by assigning random grey levels to each background region reg(i,j). The grey levels are drawn from a population with a prescribed probability distribution, with mean μ_l and variance σ_h . The evaluation procedure performed, adopted a gaussian distribution. See Figure 43.a.

In order to avoid false alarms caused by the edges at boundaries of adjacent background regions, pixels laying in the vicinity of a boundary get a grey level resulting from the interpolation within the 5x5 neighbourhood of that pixel, i.e. averaging using a 5x5 window is performed. See Figure 43.b.

μ_2 σ_h The patterns to be recognised shp(i,j) obtain grey levels drawn from a population with mean μ_2 and variance σ_h (gaussian distributed). See Figure 43.c.

By performing above procedure it is assured that the mean edge height measured between copper and substrate material is μ_2-μ_1, and the variance of the edge height is $2\sigma_h$. So σ_h describes the edge height variability. Obtained is now the test scene. See Figure 43.c.

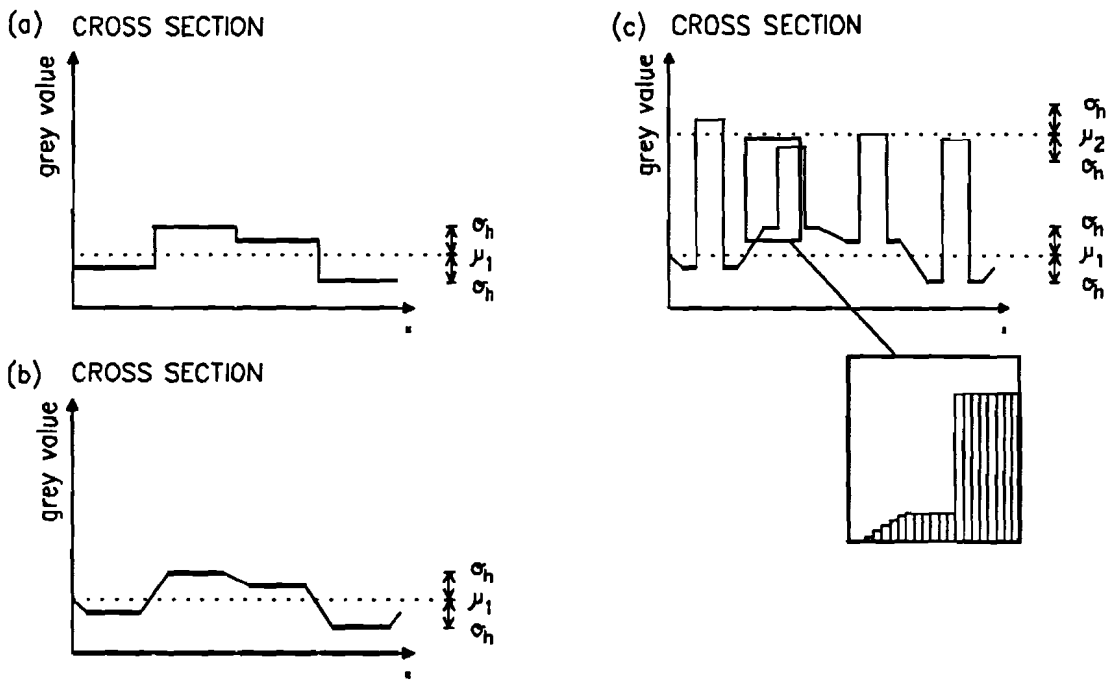


Figure 43. *(a) Background regions*
(b) Blurred boundaries
(c) Patterns

Next step in the test image generation is defining a model representing the image formation system. The image formation system comprises the scan spot projection on the surface to be inspected. As discussed in a previous chapter, the ability to project the scan spot at a particular position with a spot size and positiion deviation as small as possible, yields the highest resolution. The main factors determining the performance of the image formation system, are the 3D laser scanner's spot aberration, the spot size, and its position non-linearity.

Concerning position non-linearity, this factor does not effect the ability to measure the template position, because spot deviation over one pixel is negligibly small (<0.2%).

σ_p Spot size and spot aberration however, causes the projection of the scan spot to be scattered over one-three adjacent scan elements. This effect manifest itself by yielding a resolution deterioration. In order to reflect these effects a low pass filter with a gaussian psf having a standard deviation of σ_p is assumed. So σ_p describes the image formation system's resolution.

σ_n The image formation system is also subject to noise. Gaussian stationary white noise with standard deviation σ_n is added to the test image i_t so far obtained.

d_{sp} h_{sp} Finally speckles on the surface due to copper spots or dust, must be contained in the test image i_t. This can be described by an parameter d_{sp}, giving the appearance rate of such spots in comparison to regular pixels. For ease of simulation the speckles are assumed to be of fixed height h_{sp}. The degradation effects are schematically presented in Figure 44.

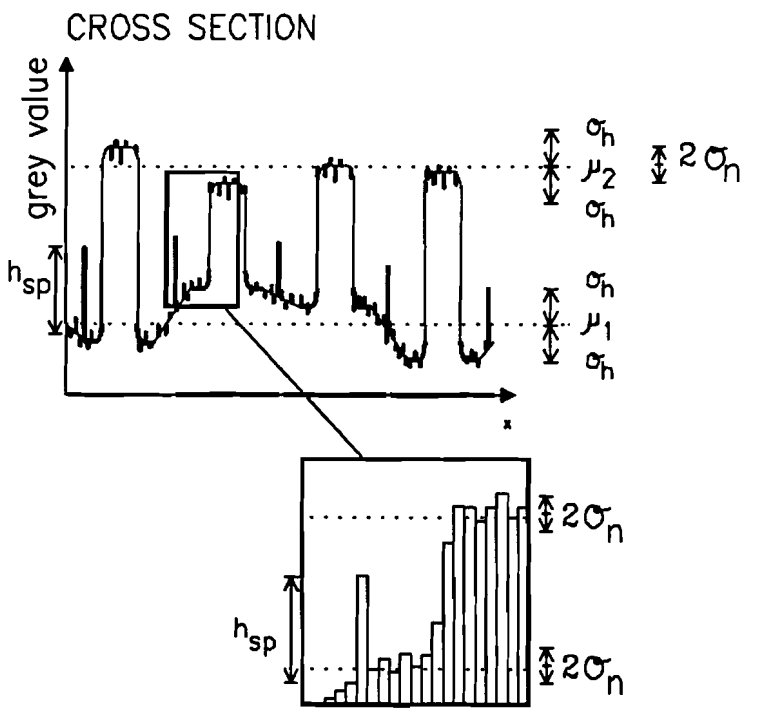


Figure 44. *Image formation model: low pass filter (σ_p), additional noise (σ_n), speckles (h_{sp})*

In summary the parameter σ_h describes the randomness in height measurement, σ_p and σ_n characterizes the image formation system (blur and noise), d_{sp} and h_{sp} models the appearance of small spots of copper material or dust. The parameters μ_1 and μ_2 can be set to arbitrary

levels if σ_h and σ_n are normalized to $|\mu_1-\mu_2|$. The steps performed are schematically presented in Figure 45.

8.5 Evaluation criteria

Multi-level input binary template matching will be applied for position measuring purposes. For this reason we are mainly interested in the following aspects:

1. The displacement along the scanline between the actual position of the pattern and the position of the recognition, Δcol.
2. The displacement perpendicular to the scanline between the actual position of the pattern and the position of the recognition, Δrow.
3. The probability of missed recognitions
4. The number of matches contained in the recognition clouds.

Of course we are also interested in the sensitivity of multi-level input binary template matching to other objects than the intended one. Usually, a system that is able to recognise a certain object is optimised such that it is able to discriminate this object from noise. However, its response to other objects is quite often not taken into account in the optimisation. It is however very difficult to take unknown, but non-stochastic signals into account in the evaluation scheme. Based on knowledge of 2D laser scanning, we are however allowed to restrict the domain of interfering objects to the simplest objects possible: small clusters of noise pixels; with small a maximum of 2x2 pixels is meant. These objects represent copper speckles and dust.

8.6 Evaluation

The test image is first being filtered by means of the Edge Preserving Smoothing Filter (DP.3) in order to remove salt and pepper noise due to dust and/or copper speckles. This filter is discussed in Appendix 3. The obtained image is being passed through the DRC yielding blobs of recognition of the patterns to be recognized. A blob to point reduction function operates on the obtained recognition image, which results into an image containing the locations of the entities recognised.

These positions are being compared with their corresponding reference points. These points are obtained by performing the DRC operation upon the reference image using an ideal template. The number of matches occurred in a recognition blob, and the recognition percentage are also being registered. See Figure 46. on page 72.

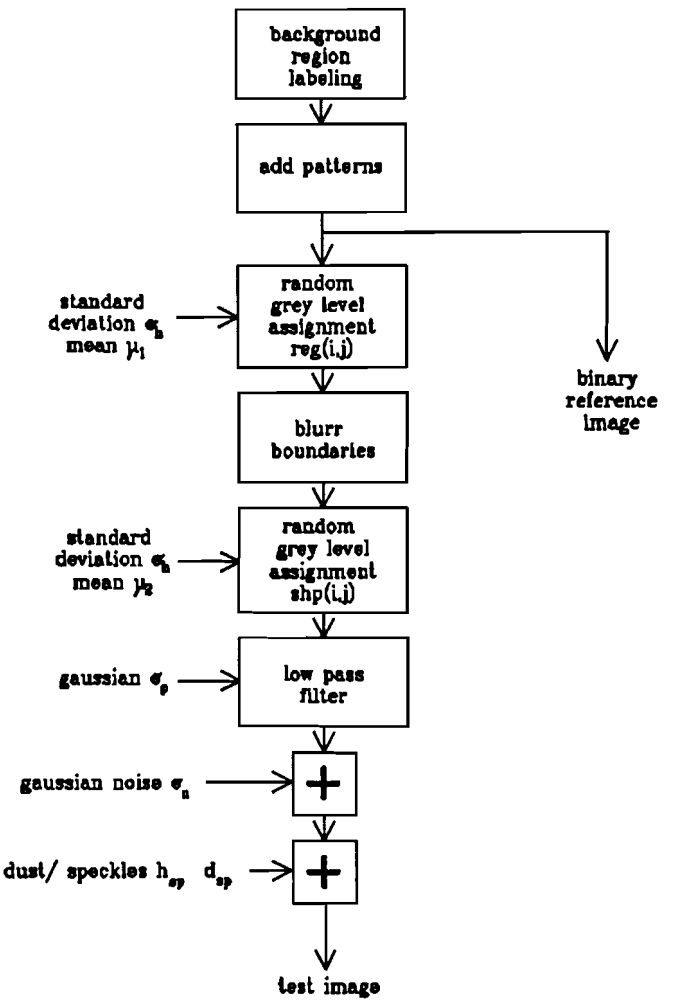


Figure 45. *Generation of test images and related reference images*

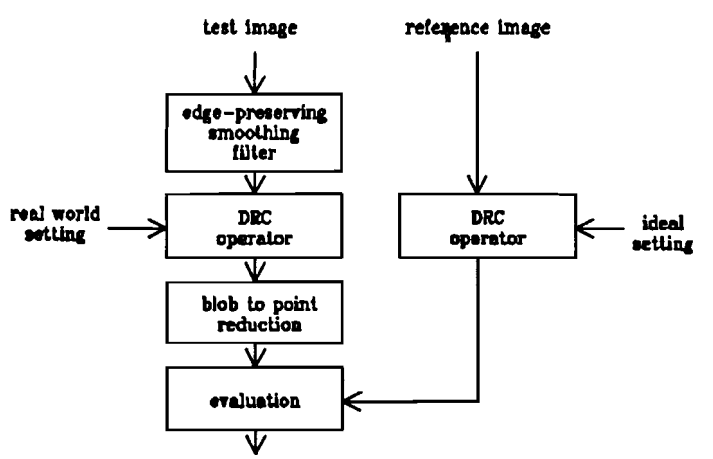


Figure 46. *Evaluation of the DRC*

8.7 Simulation arrangements

The evaluation of multi-level input binary template matching is limited to corner shaped templates. Corners of rectangulars are most suitable to be used as patterns for position measurement, because they contain edges which are perpendicular to each other and therefore provide for location information in both row as column direction.

The simulations performed can be basically distinguished into template matching concerning templates using a small band of don't care elements and templates using a wide band of don't care elements. See Figure 47. and Figure 48. respectively. Presented are templates for upper left-hand corner recognition and their corresponding ideal templates. The templates for upper right-hand corners are vertically mirrored versions of the templates presented below.

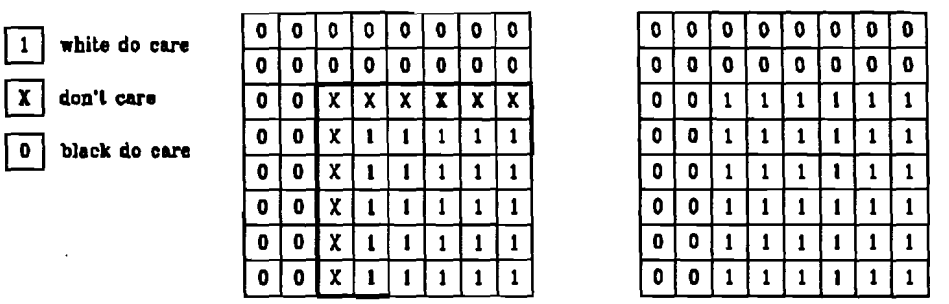


Figure 47. *Template for upper left-hand corners, using a small band of don't cares.*

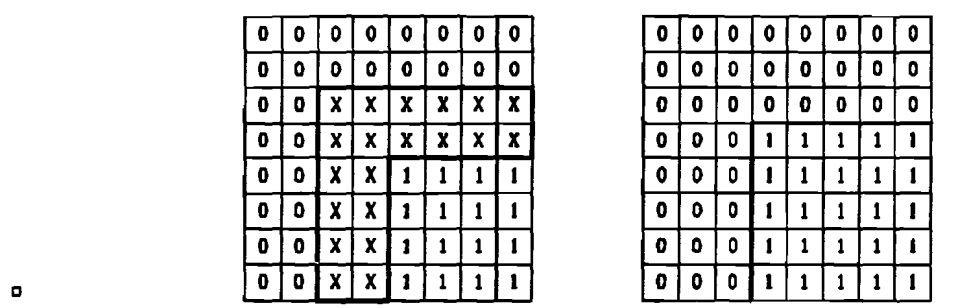


Figure 48. *Template for upper left-hand corners using a wide band of don't cares*

Simulations concerning upper left-hand side corners are denoted LNRRW or LWIDE, dependent whether a small or wide band of don't care elements is being used. Simulations concerning upper right-hand side corners are denoted RNRRW or RWIDE. Both left and right-hand corner template matching have to be evaluated due to the difference in their resulting recognition blob's shape. See Figure 49.

Figure 49. *Expected shapes of recognition blobs for left and right corners, (a) resp. (b)*

Simulations concerning templates with a small band of don't care elements are denoted LNRRW or RNRRW, dependent whether left-hand corners or right-hand corners are involved. Simulations concerning templates with a wide band of don't care elements are denoted LWIDE or RWIDE.

Template matching has been carried into effect for test images comprising the following properties:

- A moderate and large distinction between foreground and background grey level.

  moderate distinction $\mu_1 = 40, \mu_2 = 100, |\mu_1 - \mu_2| = 60$
  large distinction $\mu_1 = 30, \mu_2 = 130, |\mu_1 - \mu_2| = 100$

- a foreground and background grey level having small and large variance, resulting a small and large edge variability.

  small edge variability $\sigma_h = 10$
  large edge variability $\sigma_h = 30$

- The test images were also subject to little additional noise, moderate and a lot of additional noise.

  little noise $\sigma_n = 5$
  moderate noise $\sigma_n = 10$
  a lot of noise $\sigma_n = 25$

- All test image contained speckles with $d_{sp} = 0.001$ and $h_{sp} = 30$

Test images were acquired using

- high resolution and moderate resolution.

  high resolution $\sigma_p = 0.5$
  moderate resolution $\sigma_p = 2.0$

Template matching has been performed

- requiring complete matching and allowing non-complete matching

  complete matching $m_w = m_b = 0$
  non-complete matching $m_w = m_b = 1$

Template matching has been performed setting

- a minimal range threshold and a large range threshold N_{tv}.

  minimal range threshold N_{tv}= 1
  large range threshold N_{tv}= 10

Position measuring has been performed using

- the first match encountered of a recognition blob or the centre of gravity of a recognition blob.

The test results are given in Appendix 7. They present:

- the mean Δrow and its standard deviation
- the mean Δcol and its standard deviation
- the number of matches occurring in a recognition blob in average and its standard deviation
- number of patterns recognised out of the 360 contained in the test image

8.8 Experimental results

8.8.1 Property I

The application of the blob to point reduction function $r_2(x,y)$ instead of $r_1(x,y)$ is expected to yield a higher measuring accuracy. This is especially the case for template matching which:

- allows non-complete matching of the sub-templates;
- applies a "wide" band of don't care elements;
- applies a low range threshold.

Simulations according above properties have been performed for the recognition of upper-left hand corners and upper-right hand corners. A non-complete matching of maximal one pixel for both sub-templates is allowed, $m_w=m_b=1$, and the range threshold is set to $N_{tv}=1$. Simulations performed yielding these properties are enlisted in the table below.

left-hand	right-hand	$\mu_2-\mu_1$	σ_h
LWIDE1.b	RWIDE1.b	60	10
LWIDE3.b	RWIDE3.b	100	10
LWIDE4.b	RWIDE4.b	100	30

The gain in position measuring accuracy which can be achieved may be considerable. Appendix 5. shows which gains can be achieved. Here only simulations yielding high recognition rates were taken into account. It can be distracted from Appendix 5. that a gain

of about 3 pixels can be achieved by applying blob to point reduction function $r_2(x,y)$ instead of $r_1(x,y)$.

Strived for is however the application of minimal hardware. Therefore complete matching of both sub-templates, i.e. $m_w=m_b=0$ is required. In order to obtain guaranteed pattern resemblance to a high extent the range threshold is set to a distinct value, $N_{tv}=10$. Considered are now simulations according these settings which are enlisted in the table below.

left-hand	right-hand	$\mu_2-\mu_1$	σ_h
LNRRW1.c	RNRRW1.c	60	10
LNRRW3.c	RNRRW3.c	100	10
LNRRW4.c	RNRRW4.c	100	30

Appendix 6. shows which gains can be achieved. The increase which can be achieved in position measuring accuracy as can be distracted from the results of above simulations, is now not so considerable. About one pixel gain can be obtained.

Appendix 6. also shows that with multi-level input binary template matching requiring complete matching of both sub-templates, setting a distinct threshold (N_{tv}= 10) and applying function $r_1(x,y)$, a position measuring accuracy of within two pixels can be achieved. A more extensive verification can be made by considering the simulations enlisted in the following table. These simulations are also being performed according above mentioned specifications.

simulation	$\mu_2-\mu_1$	σ_h
LNRRW1.c	60	10
LNRRW2.c	60	30
LNRRW3.c	100	10
LNRRW4.c	100	30
RNRRW1.c	60	10
RNRRW2.c	60	30
RNRRW3.c	100	10
RNRRW4.c	100	30
LWIDE1.c	60	10
LWIDE2.c	60	30
LWIDE3.c	100	10
LWIDE4.c	100	30
RWIDE1.c	60	10
RWIDE2.c	60	30
RWIDE3.c	100	10
RWIDE4.c	100	30

8.8.2 Property II

The effect of a wider band of don't care elements, can be clearly shown on basis of simulations' results. Considered are the following simulations:

simulation	μ_2-μ_1	σ_b	band width	m_w, m_b
LNRRW4.c	100	30	1	0
LNRRW4.d	100	30	1	1
LWIDE4.c	100	30	2	0
LWIDE4.d	100	30	2	1

Compared are the simulations results of the test image acquired at low resolution, and high resolution and their relation to one another.

Acquisition at high resolution yields a higher measuring accuracy, as can be expected. For the performed simulations, the gain is quite small (less than a pixel), and measuring accuracy is, as already mentioned, within two pixels.

Acquisition at low resolution however effects the recognition rate as can be clearly shown by the diagram presented in Figure 50. Presented are the recognition rates of the test scene as used in LNRRW4.c, acquired with high resolution vs. moderate resolution for several noise levels.

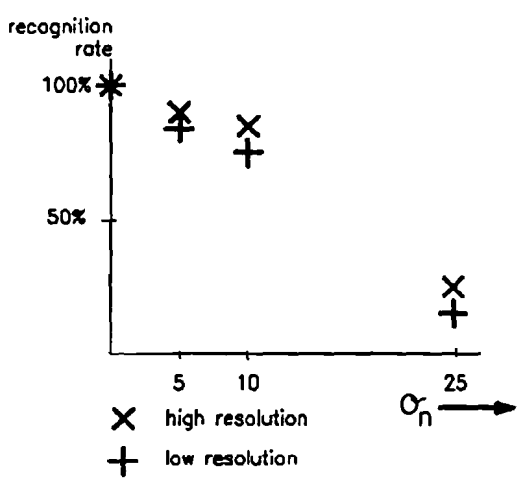


Figure 50. *Comparison of recognition rate for test scenes acquired with high resolution vs. moderate resolution.*

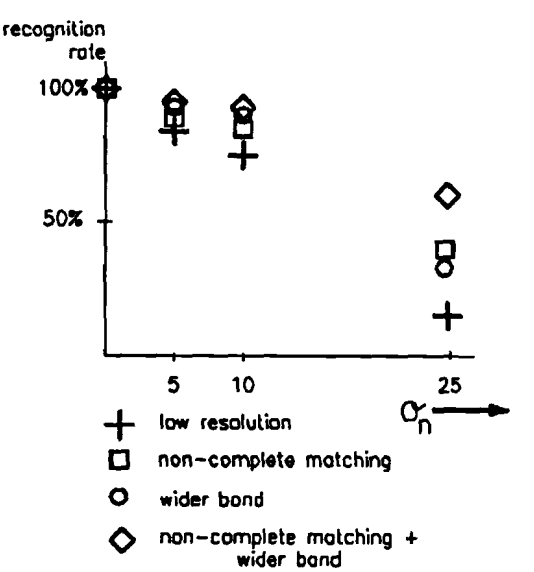


Figure 51. *Comparison of recognition rate for various DRC settings, low resolution has been applied.*

The recognition rate can be increased by adding a wider band of don't care elements, allowing non-complete matching of the sub-templates, or a combination of both. Similar simulations as LNRRW4.c have been performed but now above mentioned properties are being implemented. This result into the simulations:

- LNRRW4.d: $m_w=m_b=1$, allowing non-complete matching
- LWIDE4.c: wide band of don't cares
- LWIDE4.d: $m_w=m_b=1$ and a wide band of don't cares

The results are presented in Figure 51.

Although quantative measures can not be given, because we're dealing with generated test images, a first estimation and some qualitative measures can be given.

If there is a distinct gap between foreground and background grey level, and noise is considerably lower than the average edge height, unsharp edge transitions can be dealt with by introducing a wider band of don't care. This will increase the recognition rate without effecting the position measuring accuracy too much. It is not necessary to allow non-complete matching of the sub-templates. For low noise levels compared to the edge height, the effect of non-complete matching with respect to recognition rate is rather small. This is not the case when the noise level increases. Then considerable increase of recognition rate can be achieved by allowing non-complete matching.

8.8.3 Property III

By setting a higher range threshold, the recognition rate will be decreased, however the measuring accuracy is expected to increase. This can be shown for example on basis of simulations LWIDE3.a and LWIDE3.c, for low resolution. For low noise levels, $\sigma_n= 5$ and $\sigma_n= 10$, increase of N_{tv} yields higher position measuring accuracy however without effecting the recognition rate. For noise levels which are considerably lower than the edge height, the range threshold can be increased yielding higher position measuring accuracy, but without effecting the recognition rate. See Figure 52.

Thus, when there exists a distinct gap between foreground and background level, N_{tv} has to be chosen much larger than one in order to effect the position measuring accuracy of multi-level input binary template matching in a positive way.

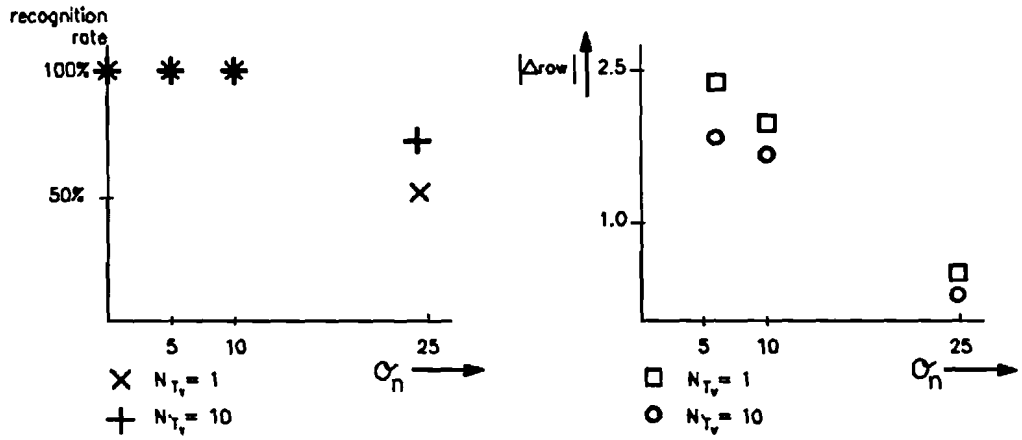


Figure 52. *Increase of Ntv yields higher measuring accuracy*

8.8.4 Crucial question

Without additional hardware, and applying the blob to point reduction function $r_1(x,y)$, a first estimation shows that a measuring accuracy which lies within two pixels can be achieved. This however requires the noise level to be considerably lower than the edge height, and a distinct gap between foreground and background grey level. In this way the size of blobs of recognition can be limited yielding sufficient measuring accuracy and recognition rate. A recognition rate of 90% under previously mentioned conditions seems then to be more than feasible.

9 Dynamic alignment

9.1 Introduction

To determine the scan spot position on the PCB under test with respect to the corresponding position in the CAD data file, the PCB must contain a set of alignment positions in order to measure the misalignment ($\Delta X,\Delta Y$), which are vectors between measured and expected alignment positions. The measured vectors ($\Delta X,\Delta Y$) should change very smoothly over the board. The correction vector ($\Delta x,\Delta y$) by which the reference data will be aligned with respect to the acquired image data of the PCB under test, is to follow the tendency of the change of the measured misalignment ($\Delta X,\Delta Y$) by a kind of interpolation. The correction vector ($\Delta x,\Delta y$) will be updated regularly. Alignment will be performed for reference data which is needed by various units. The exact reason for performing the interpolation will be discussed later.

The stepwise refinement performed on the basic concepts of interpolation in order to elaborate an alignment algorithm which can cope with the misalignment specification, will be discussed. Elaboration has been based on software simulations in order to judge whether satisfying results were obtained or further refinements needed.

9.2 Alignment features

The alignment features which will serve as reference points, consist of one or two small pattern features. These are included in a regular PCB pattern and are to be recognised (basically) by the application of multi-level input binary template matching. The alignment measurement unit can be instructed where to expect an alignment feature by the CAD-PAPS interface unit, because of the implementation of CAD-driven measurements. This area is specified by the points r_{start} and r_{end} which are the upper left-hand and bottom right-hand corners of the square with the reference alignment position r located at its centre, and size (2Δ, 2Δ). See Figure 53.

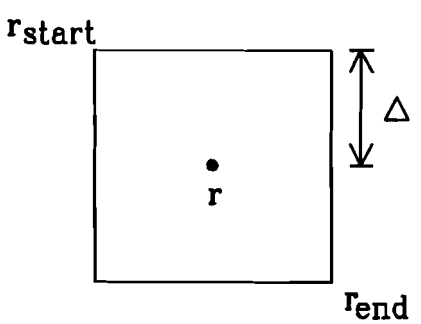


Figure 53. *AOI*

This square is called *area of interest* (AOI), and its size is inherently due to the fact that misalignment may yield a maximum position deviation of an alignment feature of Δ in all directions with respect to the reference alignment position. An alignment feature should therefore be unique within such an *area of interest* (AOI). So an alignment feature should be unique within at least twice the maximum misalignment which can occur.

9.2.1 Paired template matching

Because the requirement of uniqueness is difficult to meet for alignment features consisting of single templates such as corners, they will preferably consist of a composition of two patterns, a *master* and a *slave* pattern. The pattern to be encountered first, when scanning the PCB is called the master. Alignment features preferably consist of a pair of pattern features with a (one-dimensional) fixed distance between them. The pattern features are to be positioned at a maximum distance of 127 pixels horizontally or 63 pixels vertically from each other (because of hardware limitations). The unique combination of two templates and the distance between them will fulfil the requirement of uniqueness within an area of interest.

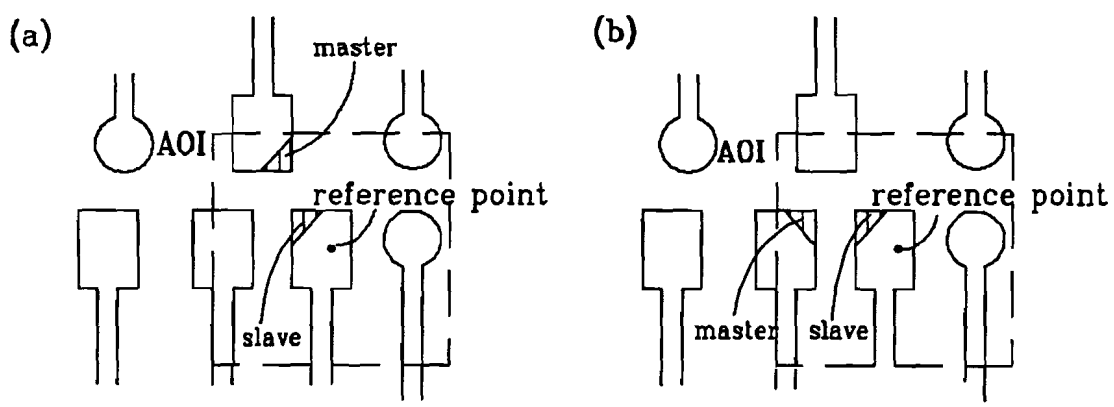


Figure 54. *Improvement of uniqueness within an AOI by combining master and slave recognition, and their distance*

Right angled shapes are preferred, because they consist of edges in both directions and therefore give alignment information in both x and y direction. When angles are not available in the scene under test, other elementary patterns such as straight line segments can be used. These, however, only give one-dimensional information. This might be sufficient for certain measuring purposes if they do not have to be combined with templates yielding information for the other dimension. For two-dimensional measuring purposes it is better to avoid the use of alignment features consisting merely of curved shapes. The obtained measuring accuracy is then not optimum. Preferably, feature templates which contain orthogonal boundaries should be chosen. This is because pixels as delivered by the image sensor are square, and therefore have orthogonal edges.

So, there are three kinds of recognitions:

i) recognitions based on a single template;
ii) recognitions based on two templates which are horizontally separated by distance *dist*;
iii) recognitions based on two templates which are vertically separated by distance *dist*.

9.2.2 Hardware concept for paired template matching

In order to combine master and slave recognitions, we make use of the hardware concept presented in Figure 56. The concept is suitable for linear scanned patterns.

Master recognitions are written into a 64 lines delay buffer. The lines are implemented as separate read-modify-write RAMs. The lines are addressed by the column counter. At each pixel clock pulse, the newly acquired pixel is stored at location y, which is specified by the column counter in line buffer 0. Suppose position (x,y) is scanned, then the following situation is obtained. Line buffer 0 is overwritten by the newly scanned value pixel (x,y). Line buffer n+1 is overwritten by the current value of line buffer n ($0 \leq n < 63$). The overwrite action is thus to be performed after the contents of line buffer n is read. See Figure 55.

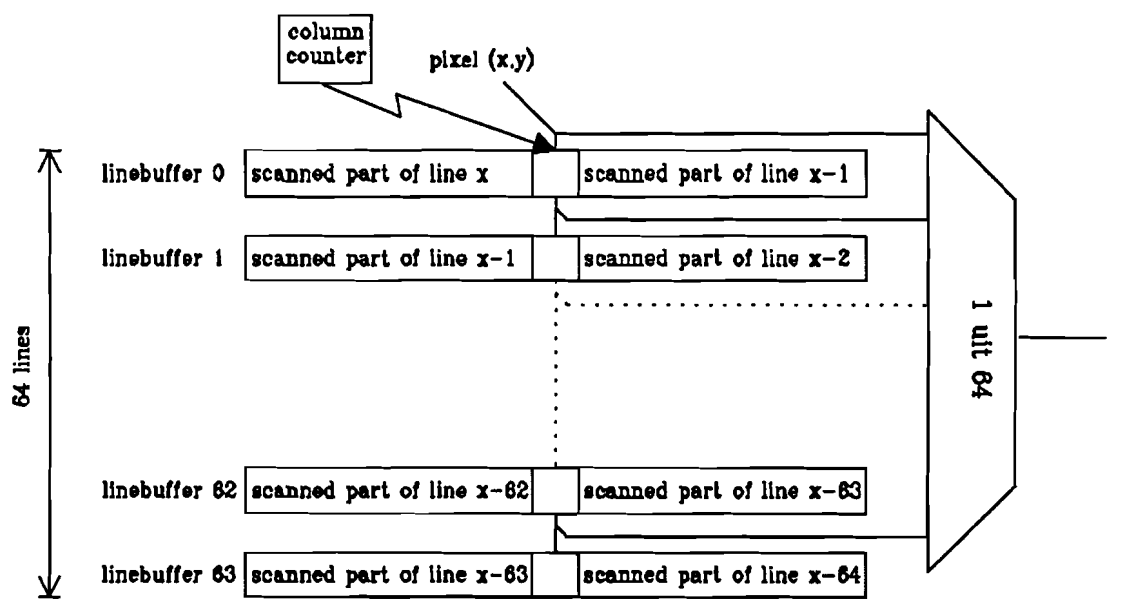


Figure 55. *Line delay buffer, allowing the current column of the last 64 li3es to be accessed*

By adding a 64:1 multiplexer as presented in Figure 55., the last 63 pixels of the current column can be selected. They can be passed to the hardware section implementing a horizontal delay.

Vertically delayed master recognitions are written at memory location a_0=addr$_{mst}$ in a cyclic addressed dual ported RAM. This contains the 128 entries which are as much as the maximum horizontal delay allowed. See Figure 56. At the same time, slave recognition is **AND**ed with the content of memory location a_1=addr$_{mst}$-d. Value d is derived from the horizontal distance between master and slave template. Addr$_{mst}$ is a cyclic counter.

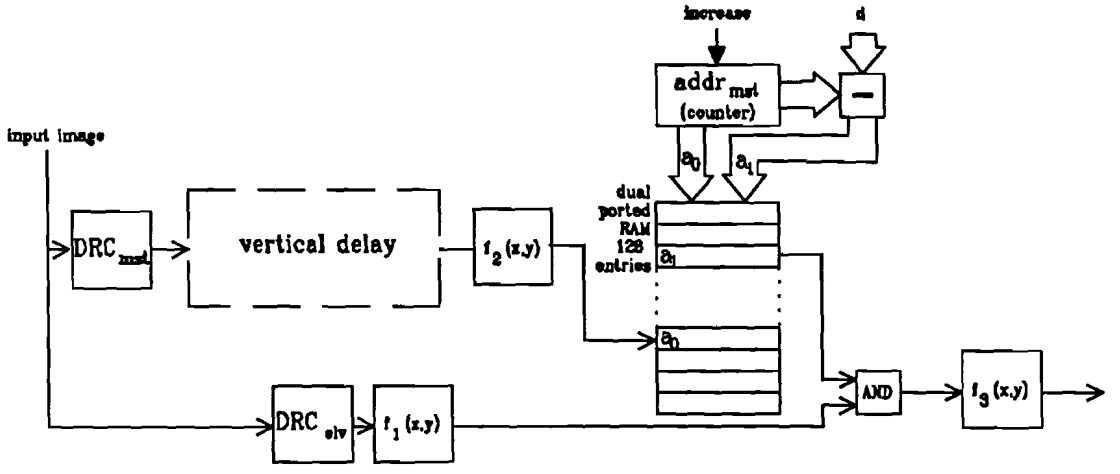


Figure 56. *Paired template matching by means of a (cyclic) dual ported RAM*

In general the DRC operator yields recognition blobs. The evaluation of the measuring characteristics of the DRC applied to angled shapes in a previous chapter, showed us the feasibility to achieve a measuring accuracy within two pixels with a recognition rate of more than 95%. It is desired that paired DRC template matching approaches this performance. Therefore picture functions $f_1(x,y)$, $f_2(x,y)$ and $f_3(x,y)$ are introduced. See Figure 56.

With respect to measuring accuracy, best performance can be obtained when one picture function, for instance $f_1(x,y)$ performs a "blob to centre of gravity point reduction function" while $f_2(x,y)$ performs a dilate operation and $f_3(x,y)$ is transparent. The dilate operation is performed in order to reduce the chance of points obtained by $f_1(x,y)$ missing points obtained by $f_2(x,y)$, in spite of the fact that both $f_1(x,y)$ as $f_2(x,y)$ yielded recognition clouds. Combined operation of $f_1(x,y)$ and $f_2(x,y)$ achieve slightly reduced recognition rate (90% when single DRC template matching yields 95% recognition rate) but the same measuring accuracy can be guaranteed. In addition the image coming from the DRC slave on which picture function $f_2(x,y)$ is performed, may apply non-angled shaped templates without affecting the measuring accuracy. See Figure 57.

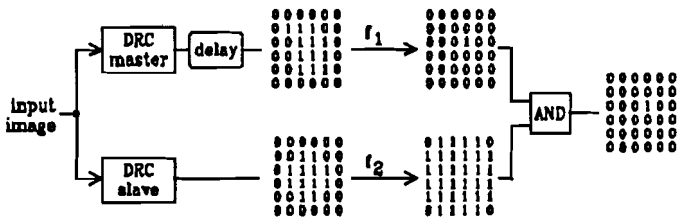


Figure 57. *$f_1(x,y)$ blob to centre of gravity point operation; $f_2(x,y)$ dilate operation*

DRC evaluation showed that "blob to centre of gravity point reduction" is not really necessary. For image featuring a rather high image quality, matching points are within 2 pixels of their reference point. Therefore, setting $f_1(x,y)$, $f_2(x,y)$ and $f_3(x,y)$ transparent will even yield smaller recognition blobs and higher measuring accuracy. See Figure 58.

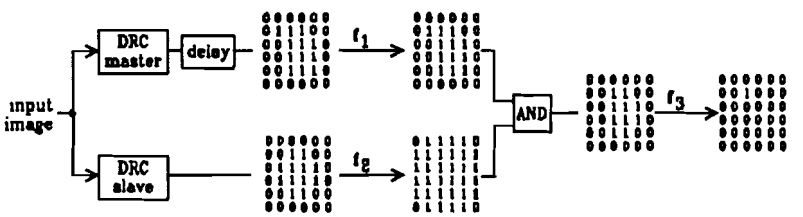


Figure 58. *$f_1(x,y)$ transparent; $f_2(x,y)$ dilate operation*

To be able to handle DRC templates yielding one dimensional position information, the operation of picture function $f_1(x,y)$ and $f_3(x,y)$ must be switched. So picture function $f_1(x,y)$ and $f_2(x,y)$ are transparent, and $f_3(x,y)$ performs a blob-to-point reduction by taking the first match it encounters. In this way measuring accuracy within two pixels at recognition rate of 90% can be achieved.

The output of picture function $f_3(x,y)$ is an image comprising indications of those locations where alignment features have been detected, and it is to be used as input for the stage at which misalignment vectors are calculated. This principle can be demonstrated by means of software simulation. See Appendix 9.2.

Along a scanline, the templates to be used for master or slave recognition will not remain the same. Multiple templates may be required. Loading templates can only be done during initialization of the system and not at time of inspection, due to strict time restrictions. Therefore the following hardware implementation has been adopted. A special IC is being developed. It features the possibility of four templates being loaded at initialization time. By enabling one of the four templates by means of a controller, that particular template is selected. See Figure 59. Templates can thus be changed without offending the time restriction. If four templates do not suffice, then multiple ICs can be applied, and their output multiplexed.

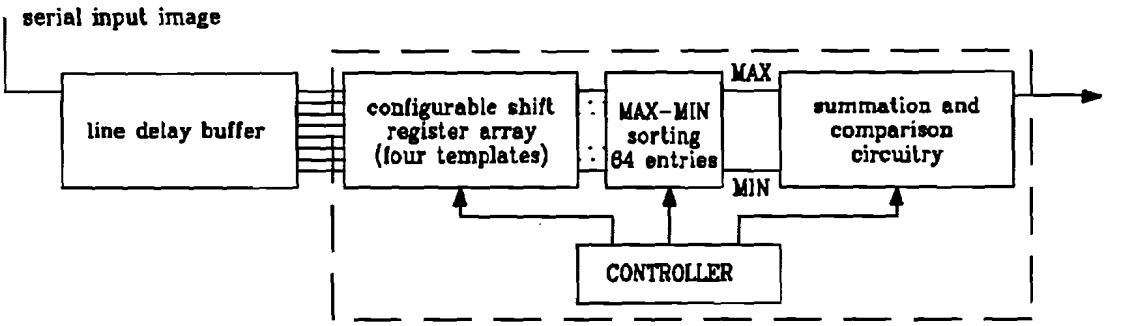


Figure 59. *Changing DRC templates by enabling one out of four templates loaded during initialization of the system*

9.3 Hardware concept for misalignment vector calculation

Because of time restrictions, and the adopted linear scan pattern of the 3D laser scanner, alignment positions have to be determined in a rather complex way in order to obtain enough processing time to perform the necessary calculations in a hardware implementation.

Whenever an alignment position r is to be expected at scan position (x,y) according the CAD-data reference file, the alignment measurement unit (AMU) is to be notified of point r_{start} which has coordinates (x-Δ, y-Δ). The exact reason has been explained earlier. From position r_{start} on, an alignment feature recognition is to be expected to occur within a window of 2Δx2Δ pixels^2 with its upper left-hand side corner at point r_{start}.

Opening a window sized 2Δx2Δ pixels^2 at any particular point r_{start} for linear scanned patterns is, however, quite a difficult task to perform in real time. To overcome this problem, a grid of cells originating from scan position (0,0) and which lines are spaced 2Δ pixels from each other in both vertical and horizontal directions, is being introduced.

Instead of opening a window at point r_{start} with size 2Δx2Δ pixels^2, the window boundaries in which recognition is to occur is being relaxed, and considered as four neighbouring cells, $c_{i,j}$,$c_{i,j+1}$, $c_{i+1,j}$, $c_{i+1,j+1}$, comprising that particular area of interest. See Figure 60.

A so-called scratch memory *SMem*, which is used to keep track of all the information necessary to calculate the misalignment vector prevailing in the area of interest, is also introduced. *SMem* contains the records S_j, $0 \leq j < N$, with N being the number of cells which can be fitted in the horizontal direction. So uniqueness is now required within 4Δx4Δ pixels^2!

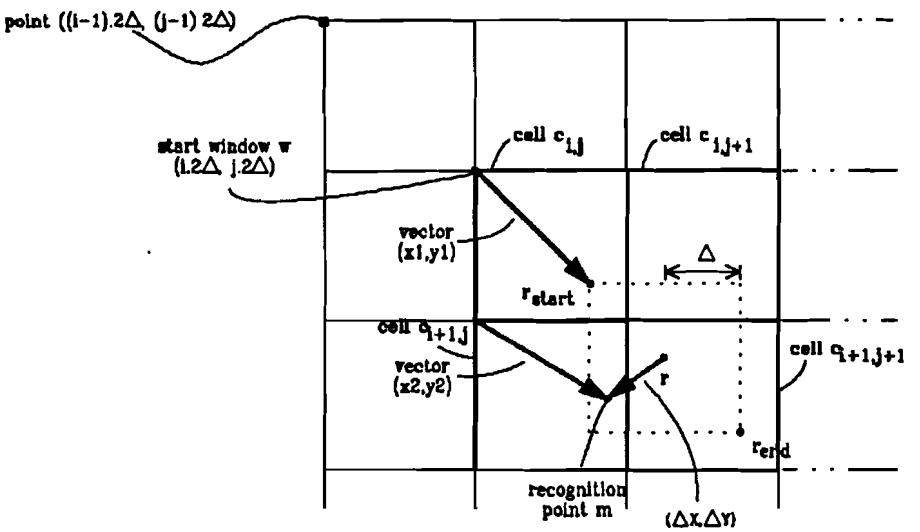


Figure 60. *Windowing mechanism*

When the scan spot reaches point w, i.e. cell $c_{i,j}$ is entered, vector (x_1,y_1) is tracked by the last N bits of the row and column counter of the scan spot. As soon as r_{start} is encountered, which is given by a control signal, record S_j has to include an indication that a recognition is to be expected within cell $c_{i,j}$, that cell $c_{i,j}$ is of cell type *c0*, an upper left-hand side cell, and vector (x_1,y_1), i.e. the window, is actually open.

As the scan spot progresses and no recognition occurred when entering cell $c_{i,j+1}$, then an indication that a recognition is to be expected within cell $c_{i,j+1}$, that cell $c_{i,j+1}$ is of cell type *c1*, an upper right-hand side cell, and vector (x_1,y_1) has to be stored in record S_{j+1}. The situation is presented in Figure 61. The different kind of cell types and legends of the scratch memory are presented in Figure 62.

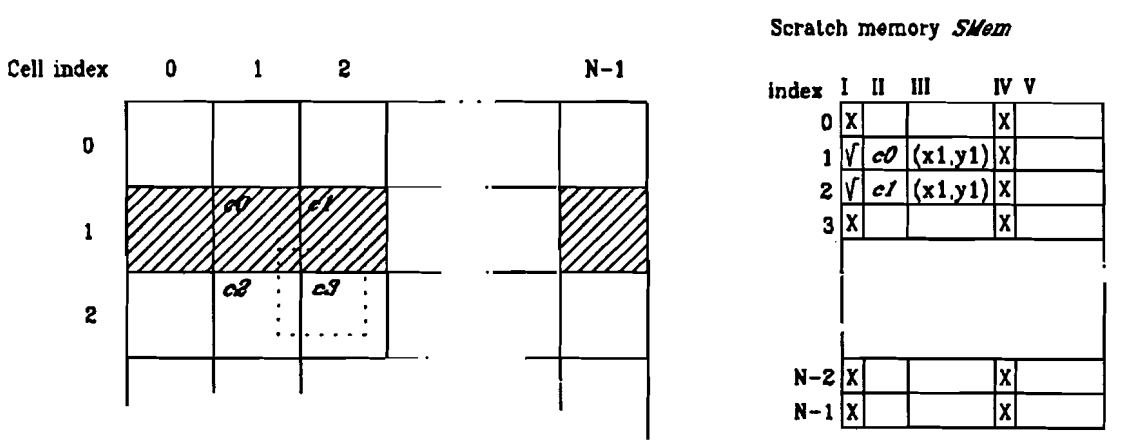


Figure 61. *Windowing mechanism using a scratch memory, scan is within the two upper cells*

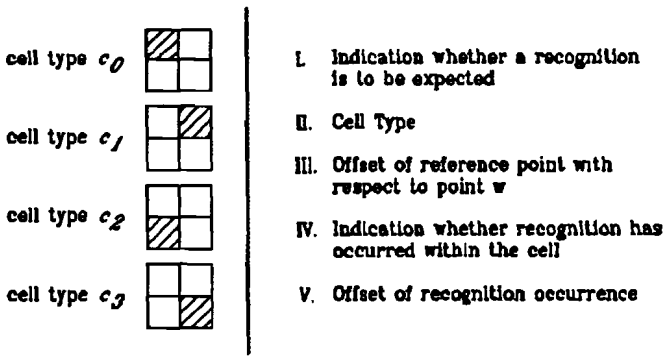


Figure 62. *The different cell types and SMem's legenda*

Assume recognition m is to occur as in Figure 60. No recognition has occurred when cell $c_{i+1,j}$ is being entered, so record S_j has to be updated; the cell type indication must be changed into *c2*, a bottom left-hand side cell, instead of *c0*. The remaining information must be kept unchanged.

When no recognition has occurred when entering cell $c_{i+1,j+1}$, the cell type indication of record S_{j+1} is updated to be cell type *c3*, a bottom right-hand side cell, instead of *c1*, the remaining information must be kept unchanged. The situation is presented in Figure 63.

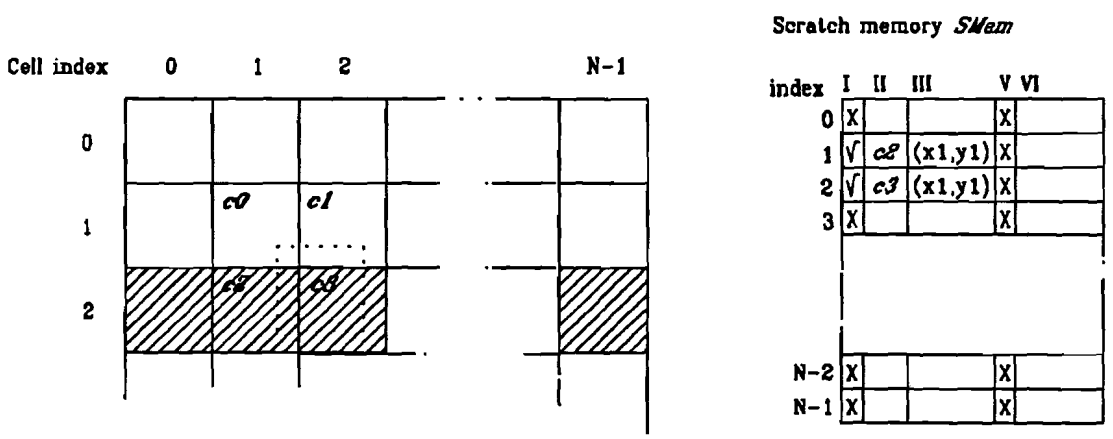


Figure 63. *Windowing mechanism using a scratch memory, scan is within the two lower cells*

An indication of recognition occurrence, together with vector (x_2,y_2), is to be stored in the scratch memory in order to process the obtained information in following steps. The indication that a recognition is expected has to be reset for records S_j and S_{j+1}. The situation is presented in Figure 64.

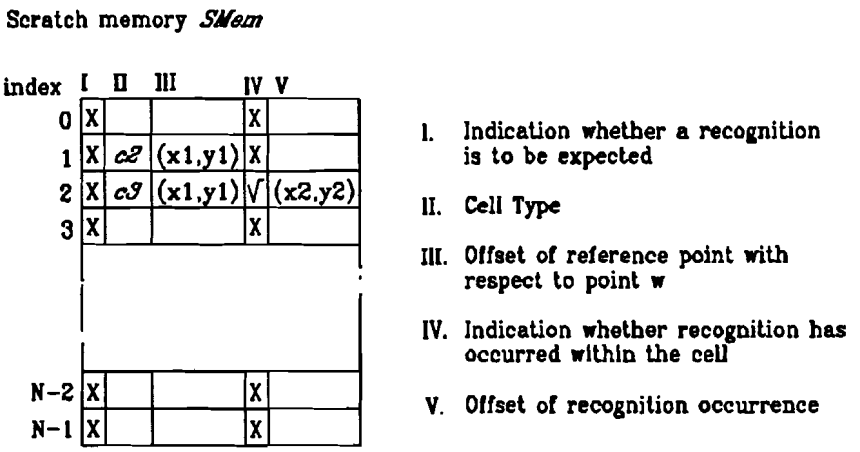


Figure 64. *Situation after recognition*

As will be made clear in later sections, the obtained misalignment vectors must be passed to the next stages of the alignment measurement unit as soon as possible. This is in order to minimize the delay between occurrence and processing of misalignment vectors, and therefore deviation. As soon as possible after recognition m has occurred, the misalignment vector (ΔX,ΔY) has to be calculated.

This is possible because vector (x_1,y_1) is known, the cell type is known, and the scan position at which recognition occurred is known. This location of recognition m is expressed by vector (x_2,y_2) giving the coordinates with respect to the upper left-hand side corner of the cell containing recognition m. Misalignment vector components ΔX and ΔY are calculated as follows:

$$\Delta X = x_2 + x_{offset} - (x_1 + \Delta);$$

$$\Delta Y = y_2 + y_{offset} - (y_1 + \Delta);$$

With

$$\textit{if } c_0 \textit{ then } \{x_{offset} = 0;\ y_{offset} = 0;\}$$

$$\textit{else if } c_1 \textit{ then } \{x_{offset} = 2\Delta;\ y_{offset} = 0;\}$$

$$\textit{else if } c_2 \textit{ then } \{x_{offset} = 0;\ y_{offset} = 2\Delta;\}$$

$$\textit{else if } c_3 \textit{ then } \{x_{offset} = 2\Delta;\ y_{offset} = 2\Delta;\}$$

9.4 The concerted action of the CAD-PAPS interface and the AMU

9.4.1 Alignment regions

As already discussed in the Chapter 6, misalignment is mainly caused by translation, rotation and deliberate product deformation. The last two causes yield a gradually changing misalignment which must be dynamically corrected. To capture the misalignment, a density of about one alignment feature for each 5x5 cm^2 of the PCB under test is expected to be suitable. Alignment features are, however, required to be recognised very accurately with respect to their specified shape and location. A density of one alignment feature each 2.5x2.5 cm^2 will therefore be applied, i.e. four times as much that required, which allows the rejection of less accurate recognitions.

In order to capture the misalignment, a grid of cells sized 2048x2048 $pixels^2$ which corresponds to 2.56x2.56 cm^2 when 12.5μm resolution is being introduced, is virtually superimposed on the PCB under test. These cells are called *alignment regions* and denoted $AR_{m,n}$, $0 \le m < M$, $0 \le n < N$. On average, each *alignment region* $AR_{m,n}$, $0 \le m < M$, $0 \le n < N$, will comprise a misalignment vector $(\Delta X, \Delta Y)_{m,n}$ prevailing in that particular region. Misalignment vectors are to be used for computing correction vectors. The CAD reference data stream is aligned with respect to the acquired image data for these correction vectors. A subdivision into alignment regions for a PCB is presented in Figure 65. The figure shown represents an average layer of a multi-layer PCB to be inspected. The patterns drawn are not to scale.

In Figure 65., we see that the grid is superimposed at a specific start position, which is detected in order to start the x and y counters. Determining the start position can, however, be inaccurate, yielding deviation. This has to be corrected. We also see stripes of regular patterns at the edges of the PCB. These patterns can be recognised accurately and used for initialisation, run-in, or outlet purposes of the alignment algorithm. The upper stroke is used for initialisation purposes, the left-hand side stroke for run-in purposes, and the right-hand side stroke for outlet purposes. A full discussion will be given later.

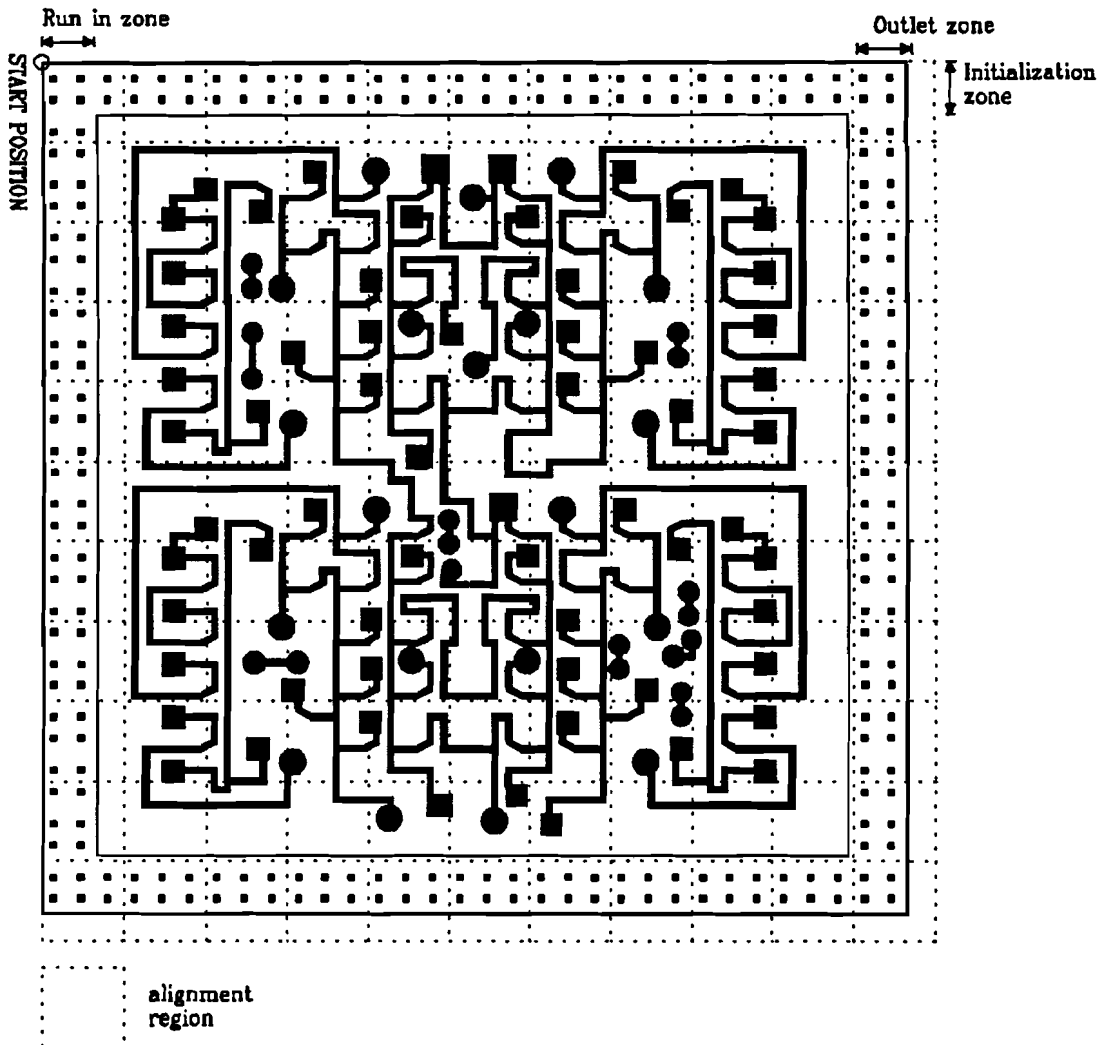


Figure 65. *Subdivision of a schematically drawn layer into alignment regions, and various zones*

9.4.2 Correction vector (Δx,Δy) specification

The information for the CAD-PAPS interface is provided in compressed form and ordered by lines. For each unit to be controlled, a separate picture (PAPS II picture format: 15 MHz data stream) containing control code must be generated.

To output relevant information, i.e. control code corresponding to the region of the PCB at that particular moment being scanned, the correct line, and within the line, the correct pixel representing control code, must be put on the designated bus.

The AMU is designed to generate a correction vector ($\Delta x,\Delta y$) for every line segment. The length is initially chosen to be 1024 pixels. The alignment information coming from the AMU is sent in PAPS picture format. Updating the correction vector ($\Delta x,\Delta y$) once every segment in both x and y directions is not expected to yield alarming deviations, because misalignment is expected to change very gradually over the board.

The CAD-PAPS interface unit stores the incoming data from a hard disk unit in a circular RAM buffer configured such that it can be simultaneously read from and written to. A RAM buffer is required because the CAD-PAPS interface must be able to randomly access reference information of the PCB regions which are in the vicinity of the current scan position.

Two address pointers are distinguished, a read pointer and a write pointer. The write pointer is always ahead of the read pointer. The read pointer is set to the scan position at that moment, compensated by the correction vector ($\Delta x,\Delta y$) prevailing for that segment.

Ooverall it is estimated that a correction of one pixel per segment in both x and y (or column and row) directions is enough. So correction vectors will be computed per segment in both x and y directions.

The 3D laser scanner has a scan width of 25.10^3 pixels. About 30 lines of CAD reference data should always be available in the CAD-PAPS interface unit buffer. According to estimations made in the Chapter 6, the maximum dynamic misalignment to be compensated for may amount up to 40 pixels along the scanline, the value of 30 pixels is being considered to be the average case.

9.4.3 Update regions

The concept of providing a correction vector once per segment in both x and y, or column and row, directions, virtually superimposes a grid with cells having sides as long as a segment over the PCB under test. These cells are called *update regions* and denoted $UR_{k,l}$, $0 \le k < K$, $0 \le l < L$. For each update region $UR_{k,l}$, a certain correction vector $(\Delta x,\Delta y)_{k,l}$ prevails.

In principle, the CAD reference data may comprise a control code at each location (x,y). The resulting coordinates (x,y) are derived by correcting the scan spot position $p_{scan} = (x_{count}, y_{count})$ with the prevailing correction vector $(\Delta x,\Delta y)_{k,l}$. The scan position $p_{scan} = (x_{count}, y_{count})$ is tracked by means of a row and col counter, synchronized by the pixel clock coming from the 3D laser scanner. This may cause certain coordinates to be skipped when Δx or Δy correction increases. To overcome this, Δx or Δy of adjacent update regions are only allowed to change one pixel. Still, certain arrangements have to be put into effect so as not to miss any measurement.

Correction vectors of update regions sharing a side are not allowed to differ by more than one pixel for vector components Δx or Δy. This arrangement yields the resulting x and y coordinate to exhibit a track which comprises increases by one, increases by two or which remains the same. See the table below. The first possibility occurs when the correction vector component of two adjacent update regions remains the same along the track of the scan spot. The second possibility occurs when the correction vector component of two adjacent update regions increases by one, and the last possibility occurs when the correction vector component of two adjacent update regions decreases by one along the track of the scan spot.

Because of the Δx correction, it is possible that a measurement is not performed at all if Δx increases by one. This is the case for measurements which occur at the beginning or end of a segment as in the table below for i=1024. To overcome this shortcoming, measurements are not allowed to be initiated at the beginning or end of a segment, but have to be performed just after the beginning or just before the end of a segment to ensure their occurrence. This arrangement is covered for by the CAD-PAPS interface unit.

Δx	10					11					11					10				
i	0	1	..	1022	1023	1024	1025	..	2046	2047	2048	2049	..	3070	3071	3072	3073	..	4094	4095
x=Δx+i	10	11	..	1032	1033	1035	1036	..	2057	2058	2059	2060	..	3081	3082	3082	3083	..	5004	5005
x(i)-x(i-1)	n.v.t.	+1	..+1..	+1	+1	+2	+1	..+1..	+1	+1	+1	+1	..+1..	+1	+1	+0	+1	+1	+1	+1

Because of the Δy correction, it is possible that the data of some segment is obtained twice (if the Δy decreases by one). This means that some of the measurement values can occur twice at the output. It is also possible that the data of some sections may be skipped. The solution is to repeat these measurements in the next line (so initiation of measurements is supplied twice).

Thus a framework is adopted which superimposes a coarse grid of alignment cells and a finer grid of update cells on the PCB under test. Capturing the misalignment over the board by a number of equidistantly placed correction vectors, each prevailing in an update region, must be such that resulting deviations are within 3-5 pixels. This margin is within the order of magnitude of fine line patterns such as copper tracks. The amount of change in correction vectors has to exhibit a certain gradualness in order to guarantee occurrence of initiations of measurements which can be specified by the CAD reference data to be at any location.

9.5 Correction vectors

For acquisition units that exhibit sufficient time to gather data of objects to be inspected, store all information, and subsequently perform software processing, misalignment can be easily

corrected for. This is because all image information can be randomly accessed, and computation of correction vectors can be iteratively performed using multiple steps.

To meet the specifications and objectives required for the correction vectors for ESPRIT 2017, we have to face difficulties resulting from:

- the use of a flying spot 3D laser scanner yielding a 15 MHz full grey level pixels data stream;
- the real-time aspect adopted for data processing in order to meet the inspection time constraints;

These set

- the requirement of on-the-fly correction;
- the limited or, practically speaking, non-existing possibility of storing input image data in a delay line buffer; storage of a single scanline would already require memory in the order of 30 kB;
- measuring inaccuracy introduced by the DRC.

Real time requirements only permit the appliance of simple arithmetic and logical operations such as increase, decrease, shift by n etc. Performing more complex operations such as correlation in real time asks for extensive pipeline processing.

Dynamic misalignment may not be constant for several inspection runs. This is due to the causes mentioned in Chapter 6. The requirement of on-the-fly correction, asks for correction vector computations to be performed simultaneously with image acquisition and measuring of the PCB under test. Thus the PCB under test is being scanned only once. On-the-fly correction, together with the adopted linear scan pattern of the 3D laser scanner, enlarges the problem of dealing with the framework of alignment and update regions. The update regions will be called in the order presented in Figure 66. The linear scan pattern implies that the scan of an update region is not completed before the next update region is entered.

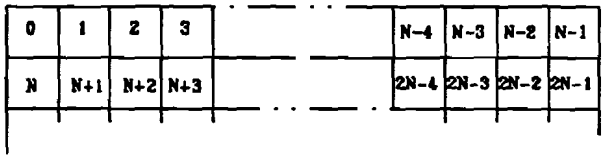


Figure 66. *Order in which update and alignment regions are called at*

Preferably, on-the-fly correction requires the correction vectors computation to be based on misalignment vectors still to come, so as to anticipate misalignment trend. Only then can correction vectors be computed properly without being behind with respect to the misalignment prevailing in the vicinity. See Figure 67.

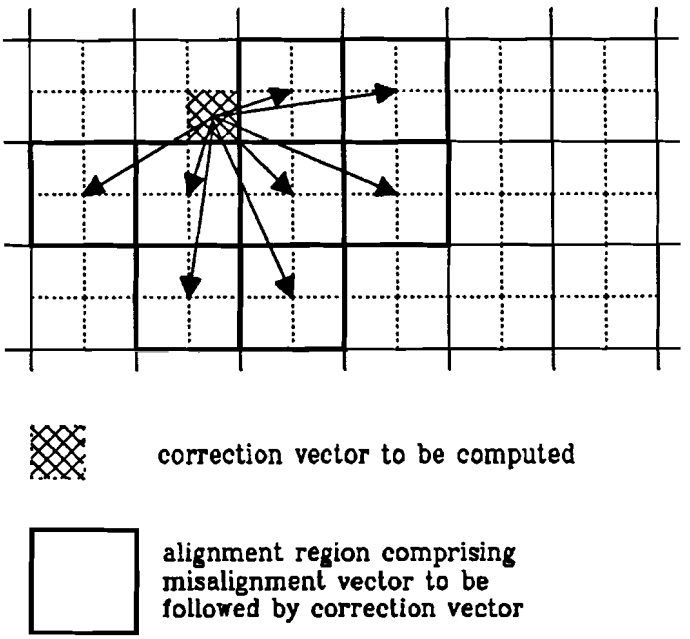


Figure 67. *On the fly correction requires to look ahead for vectors*

Due to hardware implementation considerations however, a delay line buffer of a thousand lines cannot be introduced. So, data acquired by the 3D laser scanner will serve as input data for the various measurement units without being delayed first with respect to the input image of the measurement alignment unit. Computation of correction vectors can therefore only be based on (derived) information acquired up to the scan position. However, it is expected that misalignment changes very gradually over the board, so differences between misalignment vector components of adjacent alignment regions are small. Correction may therefore be delayed with respect to the trend of misalignment. This of course causes deviations with respect to the measuring accuracy which can be achieved by paired DRC template matching. The objective is now to elaborate an appropriate correction vector computation algorithm which yields alignment within a margin of 3-5 pixels, which is about 60 μm when scanning has been performed at highest resolution of 12.5 μm.

9.6 Interpolation algorithm

Subdivision into N alignment regions along the scanline is considered. An alignment region is in turn subdivided into r_{ua}^2 update regions. And r_{ua} represents the ratio between the number of update regions which can be fitted along the scanline with respect to the number of alignment regions. The order in which the alignment or update regions are called is presented in Figure 66.

Before continuing, first the relationship between n and n' is being defined.

$$n' = \lfloor n / r_{ua} \rfloor;$$

So, if n refers to an index of an update region, then n' refers to the index of the alignment region comprising that update region.

The underlying thought for the alignment algorithm is to capture the initial misalignment and continuously adjust the obtained correction by accommodating the previously obtained horizontal stroke of update regions to the most recent trend of misalignment.

9.6.1 1D interpolation

Basically, the computation of correction vector $v_{corr}(i,j)$ of update region $UR_{i,j}$ is implemented by adjusting the correction vector of its preceding update region $UR_{i,j-1}$. The adjustment to be made is accommodated to the misalignment vector of alignment region $AR_{i',j'+1}$, which is the alignment region succeeding the one comprising the correction vector to be computed. This misalignment vector may, however, not always be available, so the misalignment vector prevailing in alignment region $AR_{i'-1,j'+1}$, which is the alignment region above, is taken. This can be justified because the course of misalignment is expected to exhibit a certain gradualness. The adjustment to be made may comprise Δx or Δy of the previous correction vector to be copied to be decreased by one, or to be increased by one. The interpolation for $(\Delta x_{i,j}, \Delta y_{i,j})$ can be completely presented in terms of formulas as given below. Similar formulas can be drawn for Δy:

$$\Delta x_{i,j} = \Delta x_{i,j-1} + x_{trend};$$

And x_{trend} is being computed as follows

$$\textit{if } \Delta x_{i,j} < \Delta X_{i'-1,j'+1} \textit{ then } x_{trend} := 1;$$

$$\textit{else if } \Delta x_{i,j} = \Delta X_{i'-1,j'+1} \textit{ then } x_{trend} := 0;$$

$$\textit{else if } \Delta x_{i,j} > \Delta X_{i'-1,j'+1} \textit{ then } x_{trend} := -1;$$

9.6.2 2D interpolation

The previously presented algorithm only suffices for a one-dimensional update of correction vectors; upstairs neighbouring update regions are not taken into consideration. To obtain two dimensional gradualness for which update regions sharing a side do not differ more than one, the function $x_{update}(x_{trend},\Delta x_{left},\Delta x_{up}) \longrightarrow \{-1,0,1\}$ needs to be introduced. Three parameters are to be specified. Parameter x_{trend} is the adjustment to be made without considering the neighbouring update region at the left or above the update region being considered. The involved correction vector component of these update regions are specified by Δx_{left} and Δx_{up}. See Figure 68. Function $x_{update}()$ must exhibit the following relation:

$x_{update}()$		x_{trend}		
		-1	0	1
$\Delta x_{up}-\Delta x_{left}$	-2	-1	-1	-1
	-1	-1	0	0
	0	-1	0	1
	1	0	0	1
	2	1	1	1

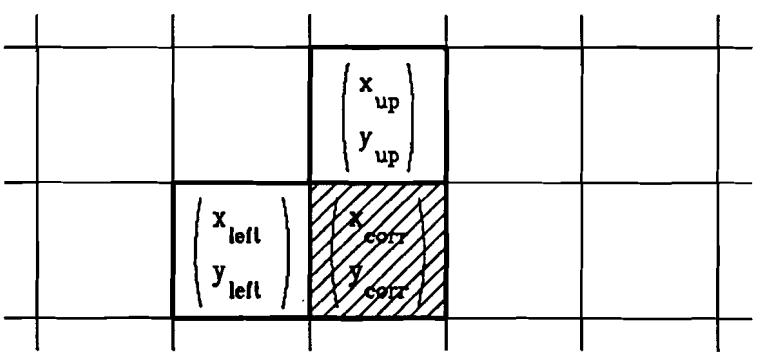


Figure 68. *Positions of the update regions involved*

For some update regions Δx_{left} or/and Δx_{up} cannot be specified. For these cases the parameters specified must be the same in order to obtain $\Delta x_{up}-\Delta x_{left}= 0$ which yields $x_{update}(x_{trend},\Delta x_{left},\Delta x_{up})= x_{trend}$.

Correction vectors are updated as follows:

First x_{trend} has to be computed:

$$\Delta x = \begin{cases} \Delta x_{i-1,j} & \text{if } j=0 \\ \Delta x_{i,j-1} & \text{if } j \geq 1 \end{cases}$$

if $\Delta x < \Delta X_{i'-1,j'+1}$ *then* $x_{trend} := 1;$

else if $\Delta x = \Delta X_{i'-1,j'+1}$ *then* $x_{trend} := 0;$

else if $\Delta x > \Delta X_{i'-1,j'+1}$ *then* $x_{trend} := -1;$

Then the proper adjustments have to be made:

$$\Delta x_{i,j} = \Delta x + x_{update}(x_{trend}, \Delta x, \Delta x_{i-1,j});$$

Notice that an outlet zone is required.

9.6.3 Overcoming delayed processing of misalignment vectors

The dynamic alignment measurement unit is expected to correct for a maximum dynamic misalignment of about 44 pixels along the scanline. Suppose misalignment features are provided for each 2.5x2.5 cm^2, which results in about 12 alignment positions along the scanline. This yields a maximum difference of 3.7 pixels in average between misalignment vector components of adjacent alignment regions. Together with measuring inaccuracy, we may have to deal with 5.7 pixels uncertainty, causing failure with respect to our desiderata of 3-5 pixels.

Alignment features location is, however, not fixed within its alignment region. Misalignment vectors are randomly distributed over the board, with an average of one per alignment region. A rigid subdivision into alignment regions and update regions can therefore not be made. This property can, however, be used to anticipate for a change in correction at an earlier moment.

The opted solution for solving the problem of misalignment vectors exhibiting deviations due to the elapsed time between occurrence and application is twofold. First, adjustment is accommodated to the most recent misalignment vector available, so misalignment vectors contained in the current horizontal stroke of alignment regions ($AR_{i',.}$) or the stroke above it ($AR_{i'-1,.}$) are used. Second, update of correction vectors is accommodated to multiple misalignment vectors, so it is more likely that a change in trend is noticed sooner, resulting

in corrections exhibiting smaller delays with respect to the trend of misalignment. The number of misalignment vectors involved has to be limited because a particular misalignment vector only prevails in its nearby vicinity. Three misalignment vectors will be considered. The one the furthest from the alignment region comprising the correction currently to be updated, has the least influence on the update.

Now the misalignment vector memory $VMem_{j'}$, $0 \leq j' < N$, which stores the most recent misalignment vector within a vertical stroke as wide as a side of an alignment region is introduced.

Correction vectors are being updated as follows:

First x_{trend} has to be computed:

$$\Delta x = \begin{cases} \Delta x_{i-1,j} & \text{if } j=0 \\ \Delta x_{i,j-1} & \text{if } j \geq 1 \end{cases}$$

$$\textit{if } (\Delta x - VMem_{j'+1}) \quad > 0 \textit{ then } x_{trend} := -1;$$

$$\textit{else if } (\Delta x - VMem_{j'+1}) \quad < 0 \textit{ then } x_{trend} := 1;$$

$$\textit{else if } \lfloor (\Delta x - VMem_{j'+2}) / r_{ua} \rfloor > 0 \textit{ then } x_{trend} := -1;$$

$$\textit{else if } \lfloor (\Delta x - VMem_{j'+2}) / r_{ua} \rfloor < 0 \textit{ then } x_{trend} := 1;$$

$$\textit{else if } \lfloor (\Delta x - VMem_{j'+3}) / 2r_{ua} \rfloor > 0 \textit{ then } x_{trend} := -1;$$

$$\textit{else if } \lfloor (\Delta x - VMem_{j'+3}) / 2r_{ua} \rfloor < 0 \textit{ then } x_{trend} := 1;$$

$$\textit{else } x_{trend} := 0;$$

Then, the proper adjustments can be made:

$$\Delta x_{i,j} = \Delta x + x_{update}(x_{trend}, \Delta x, \Delta x_{i-1,j});$$

Notice that an outlet zone is required. This can be implemented by reserving three more places in vector memory *VMem*, which will obtain the value of $VMem_{N-1}$.

Once correction vectors have been pulled towards the most recent misalignment vectors, they may become outdated because neighbouring alignment regions could exhibit more up-to-date vectors which may give rise to a changing trend. Correction vectors are now to follow the

new trend but are, however, at the same time retained by the outdated vectors. To overcome this shortcoming, a tolerance margin δ is introduced. Correction vectors accommodated a misalignment vector within a tolerance δ. This causes correction vectors not to tag along after new trends immediately but only to pursue them if the difference between correction vector and misalignment vector is distinctive. This will introduce some deviation. On the other hand, tolerance margin δ prevents correction vectors from being retained to outdated misalignment vectors. Correction vectors are updated as follows:

First x_{trend} has to be computed:

$$\Delta x = \begin{cases} \Delta x_{i-1,j} & \text{if } j=0 \\ \Delta x_{i,j-1} & \text{if } j \geq 1 \end{cases}$$

$$\textit{if } (\Delta x - VMem_{j'+1}) > \delta \textit{ then } x_{trend} := -1;$$

$$\textit{else if } (\Delta x - VMem_{j'+1}) < -\delta \textit{ then } x_{trend} := 1;$$

$$\textit{else if } \lfloor (\Delta x - VMem_{j'+2})/ r_{ua} \rfloor > \delta \textit{ then } x_{trend} := -1;$$

$$\textit{else if } \lfloor (\Delta x - VMem_{j'+2})/ r_{ua} \rfloor < -\delta \textit{ then } x_{trend} := 1;$$

$$\textit{else if } \lfloor (\Delta x - VMem_{j'+3})/2r_{ua} \rfloor > \delta \textit{ then } x_{trend} := -1;$$

$$\textit{else if } \lfloor (\Delta x - VMem_{j'+3})/2r_{ua} \rfloor < -\delta \textit{ then } x_{trend} := 1;$$

$$\textit{else } x_{trend} := 0;$$

Then the proper adjustment can be made:

$$\Delta x_{i,j} = \Delta x + x_{update}(x_{trend}, \Delta x, \Delta x_{i-1,j});$$

9.6.4 Improving the algorithm robustness

Up till now it has been implicitly assumed that each alignment region comprises an alignment feature. It is also assumed that recognition of these alignment features is 100%. Real world situation is however different. Alignment features are provided for, one per each alignment region *on average*. A PCB may contain areas which do not comprise suitable alignment features at all. Template matching is also subject to missed recognitions. These matters cause the vector memory *VMem* to contain misalignment vectors which are seriously outdated compared to neighbouring vectors.

Besides vectors being outdated, we also have to face the possible occurrence of false alignment feature recognitions causing the vector memory to contain corrupted values.

To prevent the vector memory *VMem* frome containing outdated and/or corrupted misalignment vectors to a certain extent, the following measures are being undertaken.

i) Alignment vectors contained in the vector memory *VMem* will obtain a flag indicating whether they can be considered to be valid or not. The status memory *SMem* is used in order to store these indications. A misalignment vector may be rendered invalid when its life-time has elapsed, i.e. when no misalignment vector prevailing in this vertical stroke of PCB has been measured for a certain time. To keep track of a misalignment vector's remaining life-time, life-time memory *LMem* is introduced. Initially, misalignment vectors' life-time are set to τ.

ii) To deal with false recognitions, misalignment vectors contained in the vector memory *VMem* can only be updated by vectors which are within a certain margin η. When the vector contained in *VMem* is invalid, then it will be automatically updated when more up to date vectors are available.

The alignment algorithm's function can be completely depicted on operations performed for each update region $UR_{i,j}$. So the order in which the update regions are called, i.e. the update region indices, serve as unity of progression. In hardware, the three memories *VMem*, *LMem* and *SMem* are implemented as one dimensional arrays. Values contained in these memories are being overwritten by an updated version. To perform such an update, the currently stored values have to be provided for. To distinguish old and new values, the presented formulas make use of two kinds of denotations. For example $VMem_{j'}$ denotes the misalignment vector currently stored in the vector memory and corresponding to the update region $UR_{i,j}$, that is if the stroke of update regions with row index i is being scanned at the moment. $VMem_{i,j'}$ denotes the misalignment vector stored in the vector memory at time update region $UR_{i,j}$ was/is being scanned.

Update of correction vector with row index i only requires that correction vectors with row index i-1 are available. It therefore suffices if we store correction vectors in a one dimensional array.

For each update region:

First the life-time memory has to be updated

$$LMem_{i,j'} = LMem_{i-1,j'} - 1;$$

Next step is to check whether the misalignment vector's life-time has elapsed

if $LMem_{ij'} \leq 0$ *then* $SMem_{ij'}$-*invalid*; *else* $SMem_{ij'}$-*valid*;

Correction vectors are now updated as follows:

First x_{trend} has to be computed:

$$\Delta x - \begin{cases} \Delta x_{i-1j} & \text{if } j=0 \\ \Delta x_{ij-1} & \text{if } j \geq 1 \end{cases}$$

if $SMem_{j'+1} \wedge (\Delta x - VMem_{j'+1}) > \delta$ *then* $x_{trend} := -1$;

else if $SMem_{j'+1} \wedge (\Delta x - VMem_{j'+1}) < -\delta$ *then* $x_{trend} := 1$;

else if $SMem_{j'+2} \wedge |(\Delta x - VMem_{j'+2})/ r_{ua}| > \delta$ *then* $x_{trend} := -1$;

else if $SMem_{j'+2} \wedge |(\Delta x - VMem_{j'+2})/ r_{ua}| < -\delta$ *then* $x_{trend} := 1$;

else if $SMem_{j'+3} \wedge |(\Delta x - VMem_{j'+3})/2r_{ua}| > \delta$ *then* $x_{trend} := -1$;

else if $SMem_{j'+3} \wedge |(\Delta x - VMem_{j'+3})/2r_{ua}| < -\delta$ *then* $x_{trend} := 1$;

else $x_{trend} := 0$;

Then proper adjustments are to be made:

$$\Delta x_{ij} = \Delta x + x_{update}(x_{trend}, \Delta x, \Delta x_{i-1j});$$

Misalignment vector calculation provides for newly measured vectors v, which must be booked in the three memories. Flag v_{store} indicates that a new misalignment is available and has to be booked in. The booking must be performed independently of the computation of correction vectors. The storage of newly measured misalignment vectors proceeds as follows:

if $v_{store} \wedge (|v - VMem_{i-1j'}| < \eta \vee SMem_{i-1j'}$-*invalid*) *then* $VMem_{ij'} = v$;
$SMem_{ij'}$-*valid* ;
$LMem_{ij'} = \tau$;

9.7 Concepts for real-time implementation

To convince ourselves of the algorithm's feasibility in hardware, some concepts have been drawn.

i) Misalignment vector calculation provides, for a signal v_{store}, that a misalignment vector has been encountered during scan and is to be stored. It also provides for the location for storage within vector memory *VMem* by means of v_{addr}. This address can be derived from the column position of the occurrence of recognition. Of course calculation also provides for the misalignment vector v itself. The occurrence of recognition can temporarily be dispatched by booking it into a dual-ported RAM, *VRAM*, accompanied by a flag indicating that the regarded vector has not yet been processed further. By adopting this mechanism, the provision of misalignment vectors can be virtually separated from its booking into the memories *SMem*, *LMem* and *VMem* and further processing.

ii) The booking and actual computation of correction vectors can be performed in stages, i.e. processing being spread over several line segments contained by a particular update region. This, however, requires the temporary storage of intermediate results, such as for Δx and x_{trend}. For this purpose again one dimensional arrays Δx*Mem* and x_{trend}*Mem* are introduced. Which stage is to be performed can be derived from the row counter, which update region is actually being considered can be derived from the column counter. These two counters yield the current indices (i,j) of the update region currently being considered. Note that the initialization stroke get row index 0, so $1 \leq i < K$. Column index range from $0 \leq j < L$. Memories, however, contain three more places in order to achieve outlet of the algorithm. The stages presented can be handled within the scan of one line segment.

The following concept for hardware implementation is presented:

Stage 1: booking of the misalignment vector

if $v_{store} \wedge (| VMem_{i-1j'} - v | < \eta \vee SMem_{i-1j'} = invalid)$ *then*

{
$VMem_{j'} = VRAM_{j'}.v$;
$SMem_{j'} = valid$;
$LMem_{j'} = \tau$;
$VRAM_{j'}.flag = invalid$;
}

Stage 2: update life time memory

$LMem_{ij'} = LMem_{i-1j'} - 1$;

Stage 3: check misalignment vector's life time

if $LMem_{ij'} \leq 0$ *then* $SMem_{ij'} = invalid$; *else* $SMem_{ij'} = valid$;

Stage 4: preparing the outlet zone (time in which a line is being scanned is available)

$VMem_N = VMem_{(N-1)}$;
$SMem_N = SMem_{(N-1)}$;
$LMem_N = LMem_{(N-1)}$;

$VMem_{(N+1)} = VMem_{(N-1)}$;
$SMem_{(N+1)} = SMem_{(N-1)}$;
$LMem_{(N+1)} = LMem_{(N-1)}$;

$VMem_{(N+2)} = VMem_{(N-1)}$;
$SMem_{(N+2)} = SMem_{(N-1)}$;
$LMem_{(N+2)} = LMem_{(N-1)}$;

Stage 5: preparation for computing x_{trend}

$$\Delta xMem_{j'} = \begin{cases} \Delta x_{i-1,j} & \text{if } j=0 \\ \Delta x_{i,j-1} & \text{if } j \geq 1 \end{cases}$$

$$x_{trend}Mem = 0;$$

Stage 6: sub-computation of x_{trend} with lowest weight factor

$$\text{if } SMem_{j'+3} \wedge \lfloor (\Delta xMem_{j'} - VMem_{j'+3})/2r_{ua} \rfloor > \delta \text{ then } x_{trend}Mem_{j'} := -1;$$

$$\text{if } SMem_{j'+3} \wedge \lfloor (\Delta xMem_{j'} - VMem_{j'+3})/2r_{ua} \rfloor < -\delta \text{ then } x_{trend}Mem_{j'} := 1;$$

Stage 7: sub-computation of x_{trend} with middle weight factor

$$\text{if } SMem_{j'+2} \wedge \lfloor (\Delta xMem_{j'} - VMem_{j'+2})/ r_{ua} \rfloor > \delta \text{ then } x_{trend}Mem_{j'} := -1;$$

$$\text{if } SMem_{j'+2} \wedge \lfloor (\Delta xMem_{j'} - VMem_{j'+2})/ r_{ua} \rfloor < -\delta \text{ then } x_{trend}Mem_{j'} := 1;$$

Stage 8: sub-computation of x_{trend} with highest weight factor

$$\text{if } SMem_{j'+1} \wedge \lfloor (\Delta xMem_{j'} - VMem_{j'+1}) \rfloor > \delta \text{ then } x_{trend}Mem_{j'} := -1;$$

$$\text{if } SMem_{j'+1} \wedge \lfloor (\Delta xMem_{j'} - VMem_{j'+1}) \rfloor < -\delta \text{ then } x_{trend}Mem_{j'} := 1;$$

Stage 9: finally the proper adjustments can be made:

$$\Delta x_{i,j} = \Delta xMem_{j'} + x_{update}(x_{trend}Mem_{j'}, \Delta xMem_{j'}, \Delta x_{i-1,j});$$

9.8 Initialization of the alignment algorithm

The alignment algorithm requires the initial tendency of misalignment to be captured first, i.e vector memory *VMem* together with *SMem* and *LMem* have to contain relevant values and a horizontal stroke of correction vectors must be available. For this purpose, an initialization stroke on th PCB is used. These strips are already present in the production mask of the PCB for handling the PCBs. They can be used for providing the initial alignment marks.

The initialization stroke contains a regular pattern of alignment features which occur two by two. See Figure 69.

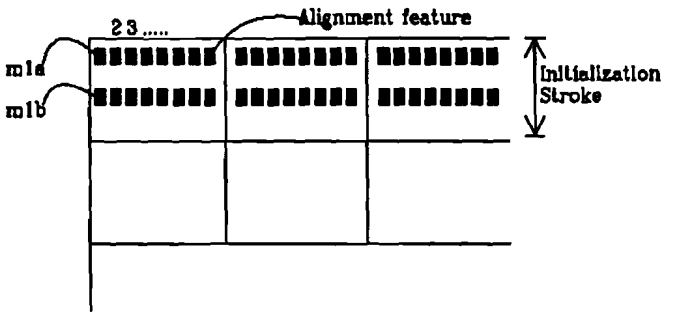


Figure 69. *Initialization stroke*

On the board the alignment features for initialization are recognised using the DRC recognition algorithm. The density of alignment features is eight times that used in the regular case. The initial tendency is calculated as follows:

For each pair of alignment features (these are no paired template) the (ΔX,ΔY) is measured and stored in the vector memory. The calculation of the correction vector (Δx,Δy) is according the principle discussed in previous sections, and stored after averaging.

At the time that all the features of the initialization stroke are measured the mean value is determined in the following way:

ΔX_{m1a}-ΔX_{m1b} < threshold
ΔY_{m1a}-ΔY_{m1b} < threshold

When satisfying the above conditions, the value is valid. The valid values are used to calculate the mean vector of an alignment region. There have to be four valid values for each such alignment region.

ΔX_i= (four valid) (ΔX_{m1a} + ... + ΔX_{m1a}) / 4
ΔY_i= (four valid) (ΔY_{m1a} + ... + ΔY_{m1a}) / 4

Each set of eight entry contains such a mean value. The curve formed by these mean values represents the initial tendency.

10 Evaluation of the alignment algorithm

10.1 Introduction

The alignment algorithm has been drawn by means of stepwise refinement based on results of computer simulations. A distinct margin in which the final corrections' deviation will range is hard to derive. Several reasons can be given. First of all, the course and magnitude of misalignment are mere expectations, real world misalignments are not available. Second, the distribution of alignment features over the board comprise uncertainty. Third, the measuring performance of DRC (paired) template matching has been derived using computer simulations applied on test images. Therefore only an indication can be given. That is an indication that the alignment algorithm is actual capable to correct for misalignments as they are expected to occur. The verification for misalignment which is assumed to be rather representative will therefore be presented.

10.2 Misalignment test vectors generation

It is expected that rotation and wanton product deformation are the main causes for the need for dynamic alignment. In order to verify the alignment algorithm's performance it suffices to consider the correction of dynamic misalignment in one direction only.

Drawn is a 3D surface in order to emulate such a misalignment. The surface's x and y dimensions corresponds to the column and row dimension of the 3D laser scanner and the z dimension represents the misalignment in one direction.

The 3D surface has to exhibit a certain smoothness. During elaboration of the alignment algorithm several aspects of misalignment have to be taken into account. It must therefore be possible to easily mould the 3D surface.

Bezier surfaces exhibit the mentioned properties. For an input set of control points, an approximating Bezier surface is formed by adding a sequence of polynomial functions formed from the coordinates of the control points. Programming has been such that control points can be interactively specified by cursor-keys and that the cross sections of the surface are displayed instantly so the surface generation over the complete board can be completely controlled. Bezier surfaces are to be used to emulate misalignment due to product deformation. Misalignment due to rotation causes the obtained Bezier surface to be tilted. So, modelling the 3D surface representing misalignment, consists of modelling a Bezier surface, and introducing a tilt to it.

The alignment algorithm's unity of progression is an update region. Superimposed on the obtained 3D surface will therefore be a grid of update regions, yielding the values $Ref_{i,j}$.

$0 \le i < K$, $0 \le j < L$, and K x L is the number of update regions. Specified is also N, which is the number of alignment regions along the scanline. The actual misalignment is being modelled, thus reference correction vectors are available.

10.3 Measuring scenario

In order to actually perform the alignment algorithm, first a *measuring scenario* has to be specified. The measuring scenario determines at what stage of the scan a misalignment vector recognition is to occur and to be booked into the vector memory *VMem*, with corresponding settings of *SMem* and *LMem*. By adapting a measuring scenario we can emulate different distributions of alignment features over the PCB. There are two kinds of bookings:

Initialize : $VMem_{i,j} = Ref_{i,j}$;
Previous : $VMem_{i,j} = Ref_{i-1,j}$;

X : $VMem_{i,j} = VMem_{i-1,j}$;

As will turn out later on, the measuring scenario is of importance for the alignment algorithm's performance.

10.4 Setting δ,τ and η

The alignment algorithm requires the setting of parameters for

- tolerance δ;
- initial life time τ;
- and false alarm margin η.

Maximum deviation of the alignment algorithm will always be larger than the inaccuracy introduced by DRC (paired) template matching method and tolerance δ. DRC inaccuracy lies within two pixels. In order to prevent deviation to exceed 3-5 pixels, tolerance can has be chosen to be $\delta=1$ or $\delta=2$. Misalignment in one direction may differ 3.7 pixels between adjacent alignment regions, in order not to retain correction to outdated vectors tolerance is best chosen to be $\delta=2$.

Measured misalignment vectors obtain a limited life time in order to prevent correction vectors' retainment. Life time should be no longer than the time required to scan two alignment regions in vertical direction. This could already introduce a deviation of about 10 pixels (2 times 3.7 pixels and measuring inaccuracy).

If life time is chosen as has been previously specified, then false alarm margin should also be chosen in that range also: $\eta = 10$ pixels.

10.5 An experimental result

Assumed are 16 x 16 alignment regions, and corrected has to be for the following misalignment. The meaning of the following two 3D surfaces can be derived from Figure 70.:

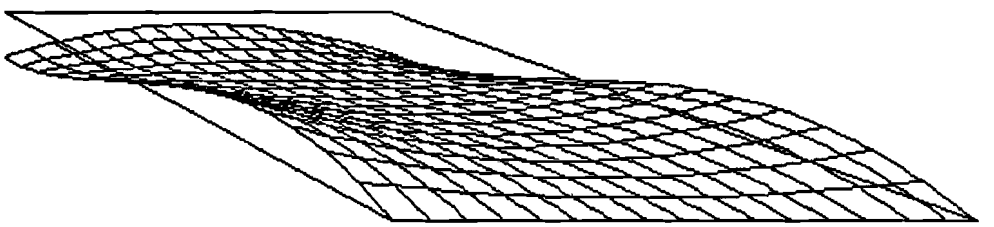


Figure 70.a *Misalignment caused by product deformation, modelled by Bezier surface*

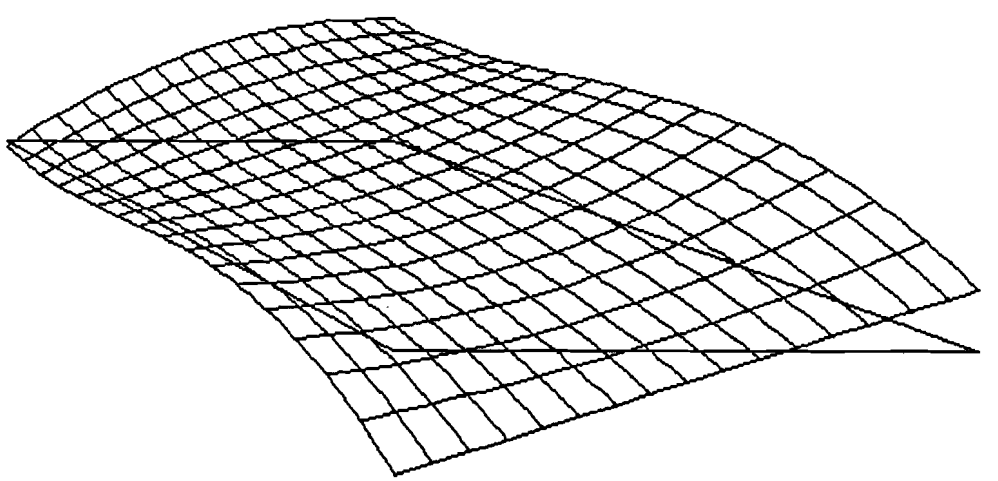


Figure 70.b *Misalignment in one direction to be corrected for (tilted Bezier surface)*

Scan direction and boundaries are as presented in Figure 70.

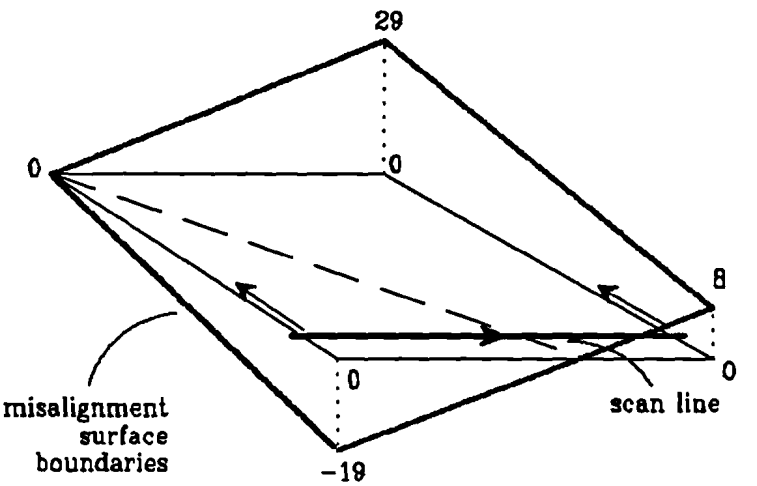


Figure 70.c *Meaning of the previously presented 3D surfaces.*

Specified are:

- number of alignment regions (in one direction) : 16
- number of update regions (in one direction) : 4 x 16
- thus, r_{ua}=4

- run in zone : 1 alignment region
- outlet zone : 3 alignment regions

- δ= 2 pixels
- τ= 8 update regions
- η= 10 pixels

- measuring inaccuracy: 2 pixels

With the above settings three simulations with different measuring scenarios are being considered. The first measuring scenario exhibits a regular pattern of alignment feature occurrences. The second measuring scenario exhibits randomly distributed occurrences of alignment features, with each alignment region comprising a alignment feature in average. The last scenario exhibits 50% recognition rate. The applied measuring scenarios are presented in Appendix 8.

Simulations shows that all three scenarios do yield satisfying deviations which are within 5 pixels. It can however be clearly seen by comparing mean and S.D. of the simulations, that randomly distributed alignment features with recognition rate of 100 % will yield the best correction results.

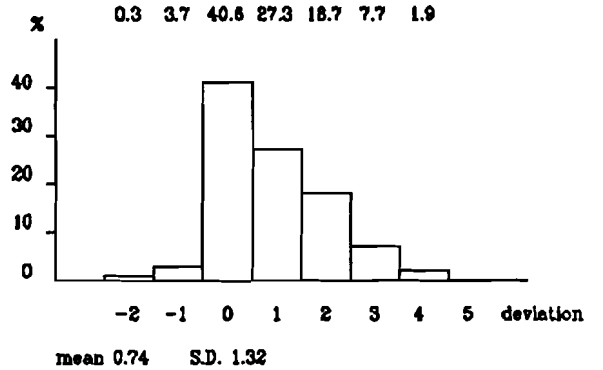


Figure 71.a *Correction deviation for regular distributed alignment features*

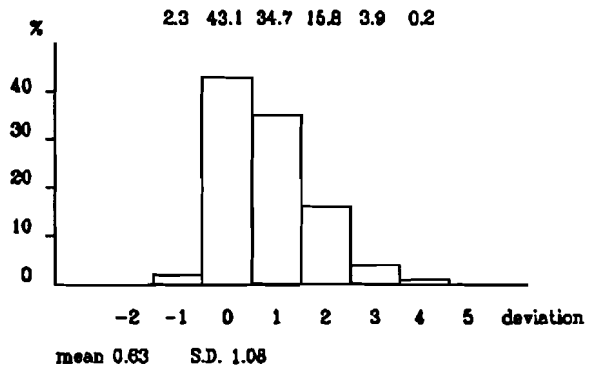


Figure 71.b *Correction deviation for randomly distributed alignment features (recognition rate: 100%)*

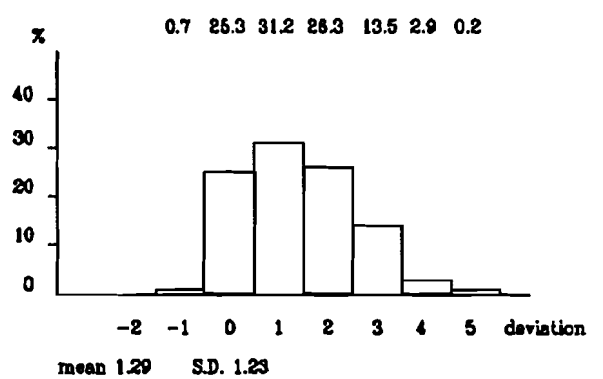


Figure 71.c *Correction deviation for randomly distributed alignment features (recognition rate: 50%)*

o valid measured misalignment vectors stored in $VMem_{32,\cdot}$.
——— smooth continuous line representing reference misalignment
⌐¬ uncontinous line representing computed correction vectors

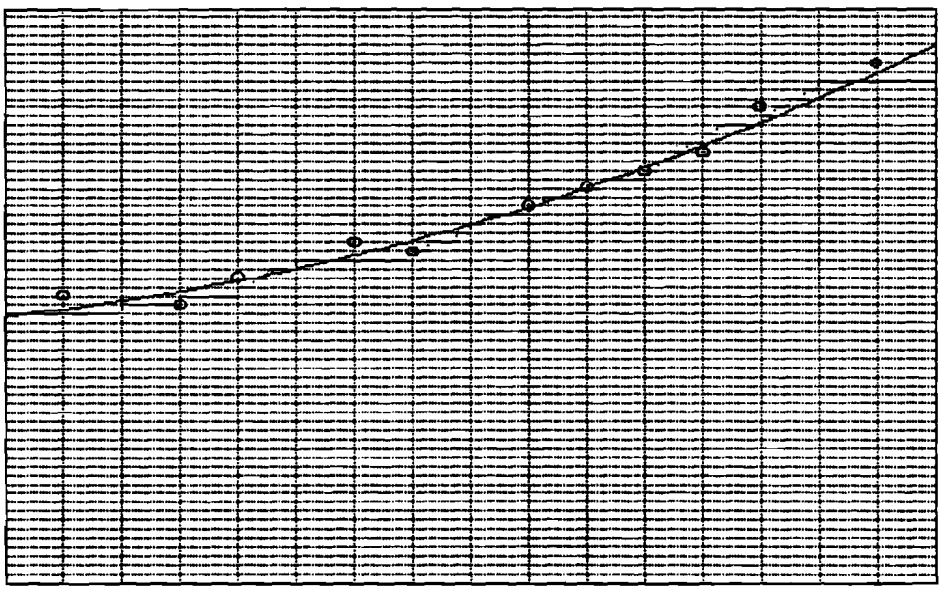


Figure 72. *Cross section along scanline at row index 32 (mid)*

\+ valid measured misalignment vectors stored in $VMem_{\cdot,32}$

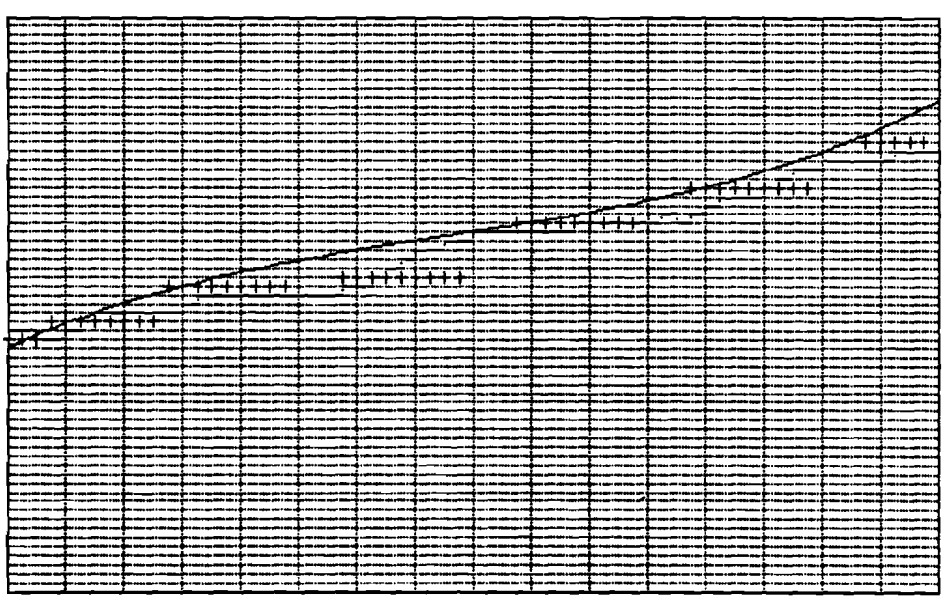


Figure 73. *Cross section perpendicular to scanline at col index 32 (mid)*

10.6 Conclusion

If alignment features are being randomly provided for with each alignment region comprising one in average, then misalignment can be captured with sufficient accuracy of about 5 pixels. That is if recognition is such that measuring inaccuracy is within 2-3 pixels and alignment features are detected within 50% of the alignment regions.

The alignment algorithm is capable of correcting for misalignment exhibiting a certain gradualness. Considered as measure of gradualness has been a tilted 3D Bezier surface comprising less than 3-4 pixels height difference between adjacent alignment regions. It is assumed that such a 3D misalignment surface mimic real world misalignment sufficiently.

Appendix 1 Hardware configuration

The hardware configuration which has been used to perform the necessary simulations consists of two Philips' Single Board Image Processors (SBIP) which are controlled by a Philips AT by means of a VME bus. The two SBIP cards will be referred to as SBIP0 and SBIP1.

The SBIP is built around the TMS34010 graphics processor by Texas Instruments, and is capable of performing a wide range of tasks involving image processing. The board contains a 512 kByte system memory to store programs, text fonts templates and can also be used to store images. The separate video memory is 256 kByte in size and can store one interlaced image of 512x512x8 pixels or 2 non-interlaced images.

The video-input of the SBIP-card consists of a video multiplexer with 4 composite video inputs and a analog to digital converter.

In front of the actual video-input of SBIP0, a DRC board is being mounted. The DRC settings concerning mask registers for the sub-templates w_w and w_b, and the range treshold are done by programming.

The output for a video monitor has an output look up table capable of supplying the red, green and blue output with 256 grey levels or 256 colours.

SBIP1's video memory will be used to store acquired or generated test images. These images are to be processed by the Dynamic Range Correlator (DRC) mounted on SBIP0. Image transfer is however being performed in analogous way. This requires side-effects to be accounted for.

First of all, deviation in synchronization requires images to be transferred in non-interlaced mode. Thus, upper and lower half of the image have to be transferred separately. Otherwise pixels transferred become blurred.

Transfer causes a delay of 20 pixels in column direction. The software simulation has to account for this side-effect.

Transfer causes pixels' grey level to be attenuated by factor 1.67. This can be remedied for by setting the output look-up-table. Grey level is now however limited to 0 .. 150.

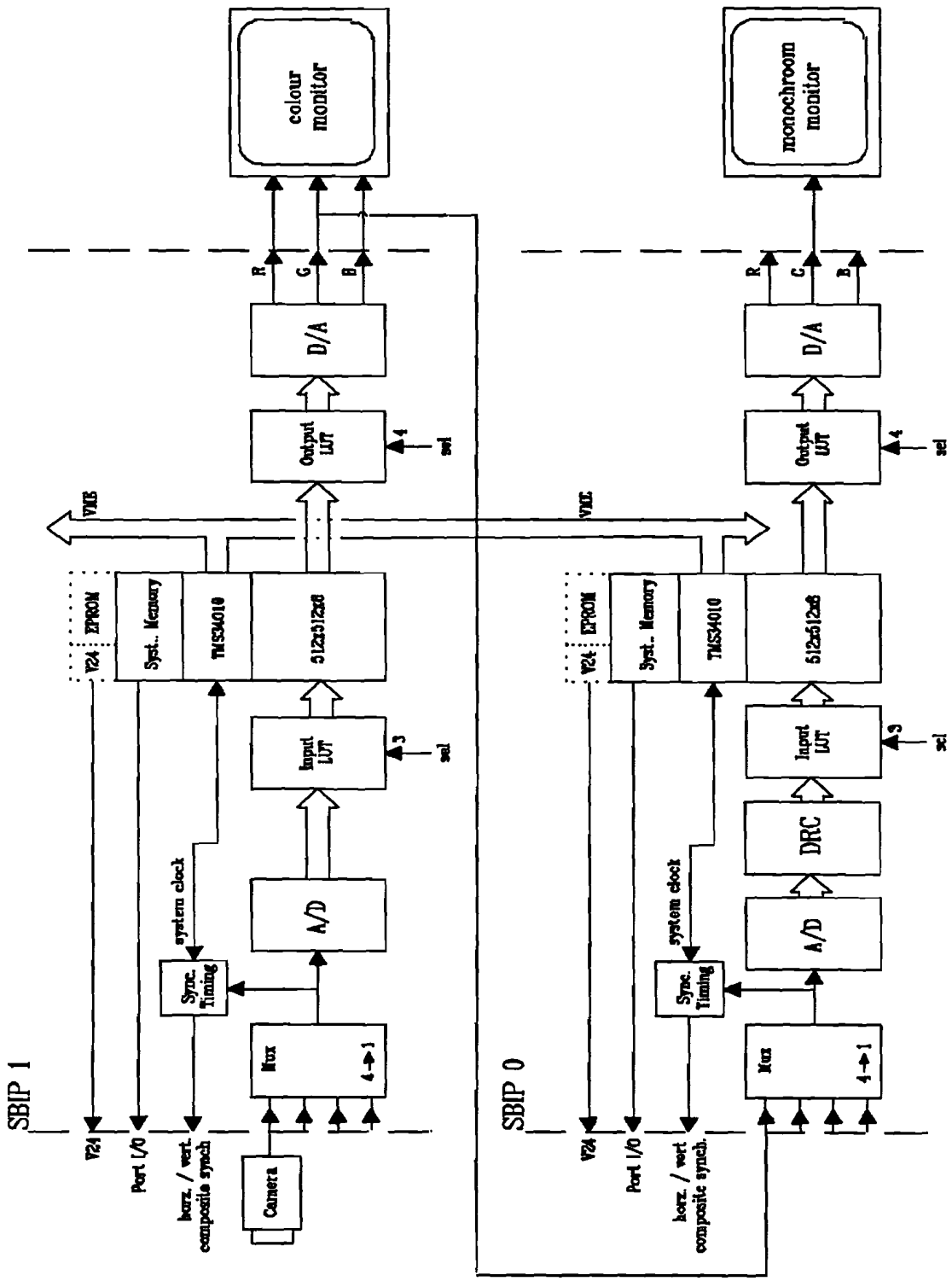


Figure 74 *Hardware configuration used for software simulations*

Appendix 2 Cellular-logic operations applying neighbourhood LUTs

2.1 Basic principle

Cellular-logic operations on binary images such as erosion, dilation, skeletonization etc., can be efficiently performed in hardware and software by applying neighbourhood table lookups.

The basic principle comprise the following. Considered is a 3x3 window w which is moved over the binary input image. At one position of the window let the pixels contained in the window be

$$w = \begin{matrix} P_3 & P_2 & P_1 \\ P_4 & P_8 & P_0 \\ P_5 & P_6 & P_7 \end{matrix}$$

Each P_k is either 0 or 1 and may be seen as a separate bit. The bits are concatenated into a string $P_8\ P_7\ P_6\ P_5\ P_4\ P_3\ P_2\ P_1\ P_0$. The string is then interpreted as the address A in a 512-entry 1-bit-wide table. The content at address A is the new value of the central pixel.

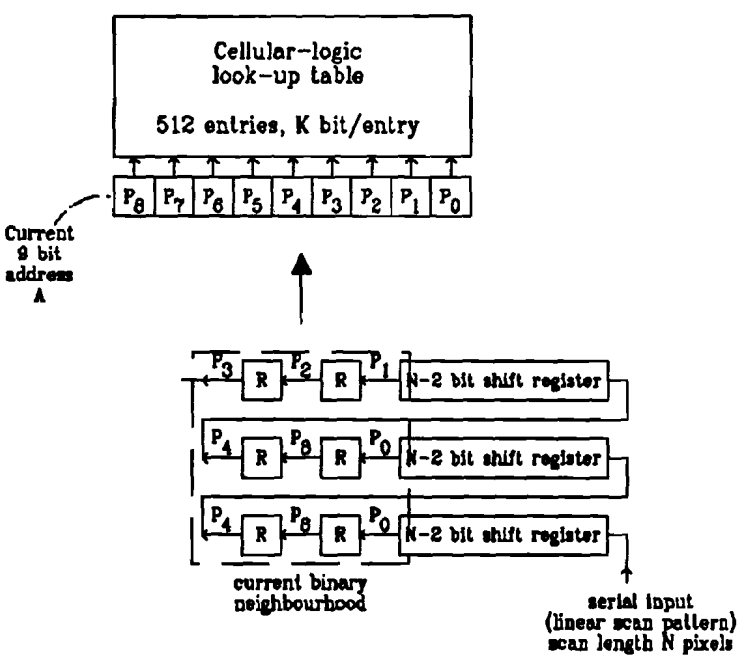


Figure 75. *Delay line buffers (shift registers) are used to assemble and update the 9 bits of the current binary neighbourhood, which forms address A.*

Instead of concatenating, address A can also be calculated by

$$A = \sum_{k=0}^{8} 2^k P_k$$

I.e., A can be calculated trough linear convolution for the coefficient scheme

8	4	2
16	256	1
32	64	128

Cellular-logic operator can thus be described as convolution with a special coefficient scheme, followed by a table lookup.

Before a cellular-logic operator f(x,y) can be applied, its table has to be generated first. This has of course to be done off-line. Address A will then go through its addressing range, and a corresponding window w will be derived. Cellular-logic operation f(x,y) is now being performed on the centre of window w yielding a result to be stored in the table lookup at address A.

Software simulation makes use of a raster scan pattern. The neighbourhood table lookup approach can now also be implemented. Here considerable gain of efficiency can be achieved if we adopt a "linear scan pattern", i.e. the convolution kernel is being moved as is the case in linear scanned patterns.

Introduced is a line buffer containing the addresses formed by 3x3 windows of a horizontal stroke of the windows' central pixel. When scanline progresses downwards, update of the line buffer does not comprise 3x3xN operations, i.e. reading all the pixels of the 3x3 neighbourhood of the central pixels in order to form new addresses. Increase of efficiency is obtained by introducing a table lookup with 512 entries and which specifies the relationship between the current address formed by the contents of a window (there are 512 possible window contents/addresses) and when this window is moved downwards one line, i.e. when binary pixels comprised at the places $P_0\ P_8\ P_4\ P_5\ P_6\ P_7$ will now be comprised at places $P_1\ P_2\ P_3\ P_4\ P_8\ P_0$. Places $P_5\ P_6\ P_7$ will initially contain zero and will be taken into consideration to form a new address A when the last acquired scanline is being read. See Figure 76.

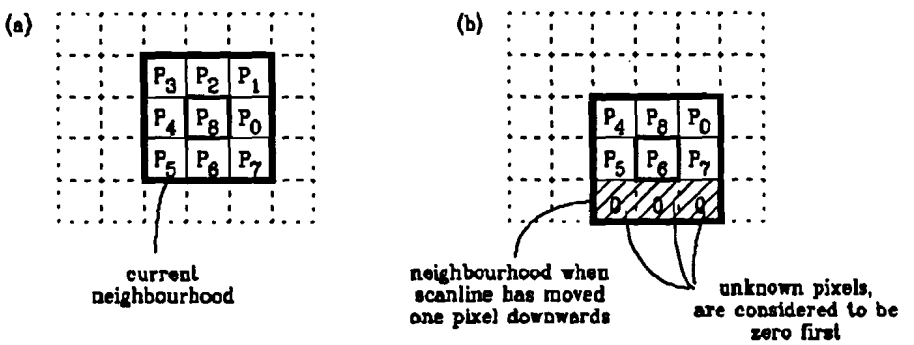


Figure 76. *When moving the 3x3 window forming address A downwards one pixel, then the new address A can be derived by table lookup*

2.2 Dilate operation

Both 4 connectivity and 8 connectivity dilate operation can be implemented using a table lookup.

2.3 Blob to centre of gravity point reduction

An estimation for the centre of gravity point of a recognition blob can be obtained by performing:

1) a n-iteration dilate operation in order to close possibly existing gaps in the recognition blob;
2) contour pixels removal until the final recognition point remains in order to perform a blob-to-point reduction.

Contour points removal of objects in binary images is being performed following the method of Stefanelli and Rosenfeld [14] with some adaptations. Originally this method is meant for skeletonization operations. By applying a different set of templates, the method is able to perform blob-to-point reduction.

It must be stated that there exists more efficient ways to perform blob-to-point reduction in hardware, however, in order to evaluate in what order measuring accuracy can be increased by applying blob-to-point reduction, the method is most suitable.

The method is based on successive deletion of contour point. Each cycle, contour points are being removed, one layer thick. In order to prevent complete objects to be removed, each cycle is subdivided into four subcycles, numbered 0,1,2,3; at each subcycle only a part of the contour is removed namely those points satisfying just one of the conditions in presented in Figure 77.

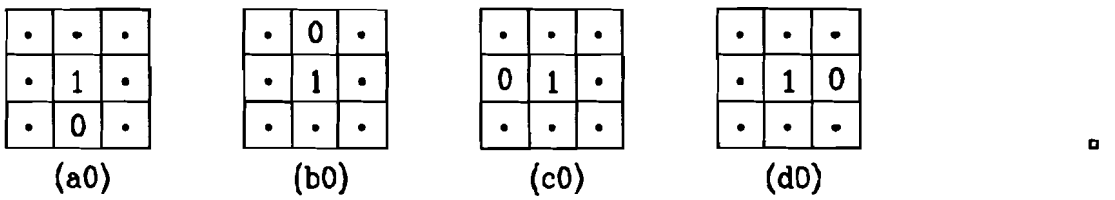


Figure 77. *Contour point condition: •'s represent don't care conditions. Lower (a0), upper (b0), left (c0), and right (d0) contour points.*

The final point conditions are those presented in Figure 78.

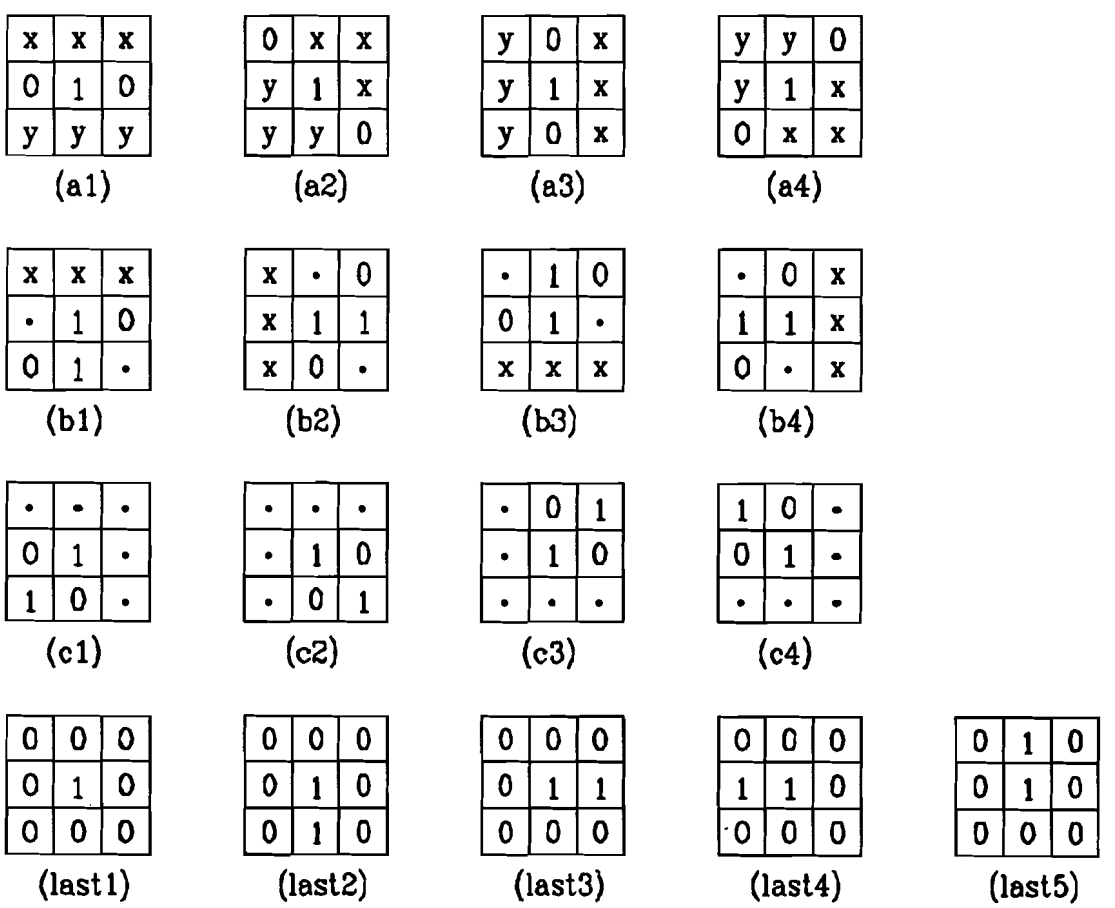


Figure 78. *Final point conditions. At least one x and one y must be non-zero (cleaning condition)*

The criteria for the various subcycles under which the centre point will be considered to be a contour point and removed is that the contour point condition must hold and that none of the final points conditions are valid as enumerated in the following table.

Subcycle	Type of contour points removed	Contour point condition	Final point condition
0	Lower	a0	a0 a1 a2 a3 a4 b1 b2 last1 last2 last3
1	Upper	b0	b0 a1 a2 a3 a4 b3 b4 last1 last3 last4
2	Left	c0	c0 a1 a2 a3 a4 b1 b4 last1 last2 last4 last5
3	Right	d0	d0 a1 a2 a3 a4 b1 b4 last1 last2 last5

For each subcycle a table can be derived. Obtained are function pxlclrA, pxlclrB, pxlclrC, pxlclrD, with a 3x3 window as input and specifying whether the centre pixel of window w has to be removed. For each cycle the subcycles are called at the following order: 0 1 3 2.

Appendix 3 Edge-preserving smoothing filter (DP.3)

The edge-preserving smoothing filter (DP.3) is designed to remove "salt and pepper noise" without removing fine lines representing cracks in copper tracks our their interspacings.

The filter can best be compared with a median filter. Both filters consider the outmost pixels within the window which is being moved over the input image. See Figure 79. For each location the evaluation set consisting of pixels referred to by a "1" in the window is being processed. If the difference between the highest and the lowest values within the set is smaller than a threshold value, then all the centre pixels are set to the average of the maximum and minimum of the set. Otherwise no action is undertaken.

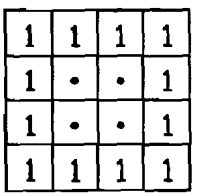


Figure 79. *Edge-preserving smoothing filter*
• *centre pixel*
1 evaluation set pixel

This means that dots small enough to fit into the central area will be removed, but lines or edges that cross the borders of the square and have a contrast, relative to the background higher than the threshold value will retain their original grey value.

Appendix 4 Bézier surfaces

Before dealing with Bézier surface generation, first Bézier curves have to be discussed.

An Bézier curve is being formed by specifying an input set of control points. The Bézier curve is then formed by adding a sequence of polynomial functions formed from the coordinates of the control points.

Suppose n+1 control points are input and designated as the vectors $p_k=(x_k,y_k,z_k)$ for k varying form 0 to n. From these coordinate points, we calculate an approximating Bézier vector function $P(u)$, which represents the three parametric equations for the curve that fits the input control points p_k. This Bézier coordinate function is calculated as

$$P(u)=\sum_{k=0}^{n} p_k.B_{k,n}(u)$$

Each $B_{k,n}(u)$ is a polynomial function defined as

$$B_{k,n}(u)=C(n,k).u^k(1-u)^{n-k}$$

and the C(n,k) represent the binomial coefficients

$$C(n,k)=\frac{n!}{k!(n-k)!}$$

P(u) can be written in explicit form as a set of parametric equations for the individual curve coordinates

$$x(u)=\sum_{k=0}^{n} x_k.B_{k,n}(u)$$

$$y(u)=\sum_{k=0}^{n} y_k.B_{k,n}(u)$$

$$z(u)=\sum_{k=0}^{n} z_k.B_{k,n}(u)$$

The polynomials $B_{k,n}(u)$ are called *blending functions* because they blend the control points to form a composite function describing the curve. This composite function is a polynomial of degree one less than the number of control points used. an important property of any Bézier curve is that it lies within the convex hull (polygon boundary) of the control points. This ensures that the curve smoothly follows the control points.

Two sets of Bézier curves can be used to represent surfaces of objects specified by input control points. The parametric vector function for the Bézier surface is formed as the Cartesian product of Bézier blending functions

$$P(u,v) = \sum_{j=0}^{m} \sum_{k=0}^{n} p_{j,k} . B_{j,m}(u) . B_{k,n}(v)$$

with $p_{j,k}$ specifying the location of the (m+1) by (n+1) control points.

Bézier surfaces have the same properties as Bézier curves, and they provide a convenient method for interactive design applications.

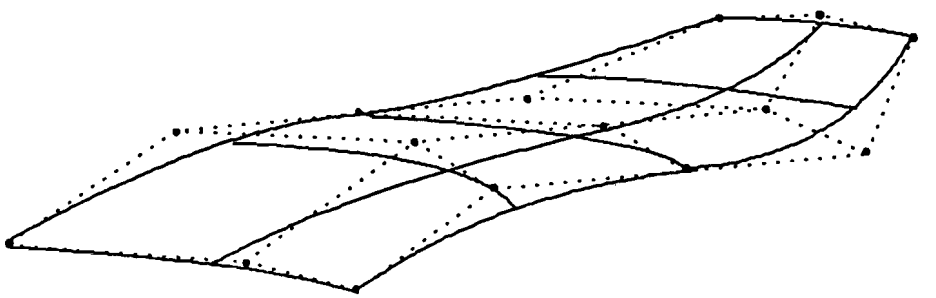


Figure 80. *Example of a 3D Bézier surface*

Appendix 5 Gain in measuring accuracy for non-complete matching

sim.	r_1 Δrow		r_2 Δrow		#match		%	σ_p	σ_n
LWIDE1.b	-3.42	0.84	-0.76	0.69	23.38	4.65	100	2.0	5.0
LWIDE3.b	-3.23	0.96	-0.65	0.68	22.58	4.20	100	0.5	5.0
LWIDE4.b	-3.01	0.96	-0.55	0.85	22.24	4.51	98.6	2.0	5.0

sim.	r_1 Δcol		r_2 Δcol		#match		%	σ_p	σ_n
LWIDE1.b	-2.51	1.08	-0.71	0.72	23.38	4.65	100	2.0	5.0
LWIDE3.b	-2.43	0.89	-0.39	0.71	23.00	3.23	100	2.0	5.0
LWIDE4.b	-2.36	1.07	-0.50	0.83	22.24	4.51	98.6	2.0	5.0

sim.	r_1 Δrow		r_2 Δrow		#match		%	σ_p	σ_n
RWIDE1.b	-3.09	0.95	-0.43	0.57	21.21	3.06	100	2.0	5.0
RWIDE3.b	-2.82	0.96	-0.45	0.64	19.59	2.89	100	2.0	5.0
RWIDE4.b	-2.76	0.93	-0.44	0.68	19.42	3.61	98.7	2.0	5.0

sim.	r_1 Δcol		r_2 Δcol		#match		%	σ_p	σ_n
RWIDE1.b	1.02	0.99	0.12	0.66	21.21	3.06	100	2.0	5.0
RWIDE3.b	1.09	0.63	0.23	0.62	19.59	2.89	100	2.0	5.0
RWIDE4.b	0.99	0.94	0.12	0.65	19.42	3.61	98.7	2.0	5.0

Appendix 6 Gain in measuring accuracy for minimal sized recognition blobs

sim.	r_1 Δrow		r_2 Δrow		#match		%	σ_p	σ_n
LNRRW1.c	-0.92	0.27	-0.81	0.27	3.15	1.02	86	0.5	10.0
LNRRW3.c	-1.32	0.48	-0.27	0.44	6.35	1.99	99	2.0	5.0
LNRRW4.c	-1.16	0.53	-0.35	0.48	5.38	2.57	85	2.0	5.0

sim.	r_1 Δcol		r_2 Δcol		#match		%	σ_p	σ_n
LNRRW1.c	-0.81	0.40	-0.52	0.50	2.79	1.17	84	2.0	5.0
LNRRW3.c	-1.14	0.54	-0.63	0.48	4.34	1.77	99	2.0	10.0
LNRRW4.c	-1.06	0.56	-0.61	0.51	3.94	2.13	76	2.0	10.0

sim.	r_1 Δrow		r_2 Δrow		#match		%	σ_p	σ_n
RNRRW1.c	-0.81	0.40	-0.60	0.49	2.67	1.19	86	2.0	5.0
RNRRW3.c	-1.27	0.45	-0.33	0.48	6.18	1.91	100	2.0	5.0
RNRRW4.c	-1.18	0.51	-0.41	0.49	5.25	2.49	86	2.0	5.0

sim.	r_1 Δcol		r_2 Δcol		#match		%	σ_p	σ_n
RNRRW1.c	0.46	0.51	0.26	0.45	1.85	0.99	57	2.0	10.0
RNRRW3.c	0.32	0.49	-0.15	0.45	6.18	1.91	100	2.0	5.0
RNRRW4.c	-0.34	0.49	-0.05	0.44	5.25	2.49	87	2.0	5.0

LNRRW1.a $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{tr}=1$ $m_w=m_b=0$

	σ_w	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	357		356		348		327		13		5	
	Δrow	-1.16	0.50	-1.84	1.03	-1.01	0.30	-1.07	0.63	-0.54	0.50	-0.20	0.40
	Δcol	-1.17	0.49	-1.73	1.05	-0.98	0.28	-1.04	0.64	-0.62	0.49	-0.60	0.49
	#match	4.30	0.81	5.77	2.18	3.75	0.91	3.33	1.67	1.54	0.63	1.20	0.40
r_2	%x360	357		355		346		323		17		5	
	Δrow	-0.87	0.45	-0.74	0.74	-0.84	0.37	-0.61	0.56	-0.47	0.50	-0.20	0.40
	Δcol	-1.01	0.31	-1.03	0.60	-0.88	0.33	-0.61	0.57	-0.71	0.46	-0.40	0.49
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LNRRW1.b $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{tr}=1$ $m_w=m_b=1$

	σ_w	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		359		358		353		74		43	
	Δrow	-1.63	0.75	-2.87	1.02	-1.28	0.54	-1.66	0.66	-0.61	0.54	-0.51	0.59
	Δcol	-1.50	0.74	-2.48	1.01	-1.23	0.58	-1.46	0.70	-0.66	0.50	-0.33	0.64
	#match	6.21	1.78	9.77	2.81	5.23	1.45	6.93	2.27	1.88	0.99	1.60	0.92
r_2	%x360	360		359		358		353		77		47	
	Δrow	-0.70	0.61	-0.93	0.68	-0.67	0.48	-0.42	0.54	-0.52	0.55	-0.32	0.51
	Δcol	-1.04	0.51	-1.31	0.62	-0.95	0.35	-0.84	0.44	-0.60	0.49	-0.40	0.70
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LNRRW1.c $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{tr}=10$ $m_w=m_b=0$

	σ_w	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	343		303		313		220		3		1	
	Δrow	-0.93	0.25	-0.83	0.39	-0.92	0.27	-0.75	0.53	-0.33	0.47	0.00	0.00
	Δcol	-0.93	0.26	-0.81	0.40	-0.93	0.26	-0.72	0.49	-0.33	0.47	-1.00	0.00
	#match	3.67	0.83	2.79	1.17	3.15	1.02	1.84	0.91	1.00	0.00	1.00	0.00
r_2	%x360	344		302		310		218		4		2	
	Δrow	-0.91	0.29	-0.65	0.48	-0.81	0.39	-0.64	0.50	-0.25	0.43	0.00	0.00
	Δcol	-0.90	0.30	-0.52	0.50	-0.75	0.44	-0.36	0.51	-0.50	0.50	-0.50	0.50
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LNRRW1.d $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{tr}=10$ $m_w=m_b=1$

	σ_w	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	350		346		346		322		33		15	
	Δrow	-0.97	0.20	-0.98	0.36	-0.98	0.21	-1.05	0.50	-0.55	0.56	-0.20	0.40
	Δcol	-0.94	0.24	-0.97	0.38	-0.98	0.25	-1.00	0.49	-0.61	0.49	-0.60	0.49
	#match	3.84	0.65	3.94	1.29	3.73	0.80	3.55	1.68	1.76	0.95	1.27	0.57
r_2	%x360	351		343		345		321		33		14	
	Δrow	-0.95	0.22	-0.65	0.48	-0.92	0.26	-0.57	0.51	-0.45	0.56	-0.14	0.35
	Δcol	-0.93	0.26	-0.85	0.35	-0.92	0.27	-0.66	0.49	-0.58	0.49	-0.57	0.49
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LNRRW2.a $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=1$ $m_w=m_b=0$

	σ_e	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	288		287		262		244		56		29	
	Δrow	-1.26	0.64	-1.90	0.96	-1.03	0.39	-1.20	0.66	-0.77	0.46	-0.31	0.46
	Δcol	-1.24	0.74	-1.71	1.02	-1.01	0.41	-1.19	0.70	-0.73	0.44	-0.59	0.56
	#match	4.35	1.41	6.24	2.79	3.90	1.16	4.46	2.53	2.16	1.21	1.38	0.67
r_2	%x360	288		283		261		245		57		30	
	Δrow	-0.91	0.68	-0.66	0.75	-0.78	0.44	-0.42	0.55	-0.70	0.46	-0.30	0.46
	Δcol	-1.09	0.56	-1.13	0.60	-0.92	0.38	-0.72	0.49	-0.75	0.43	-0.60	0.55
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LNRRW2.b $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=1$ $m_w=m_b=1$

	σ_e	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	301		297		284		281		109		82	
	Δrow	-1.88	0.93	-2.69	0.96	-1.38	0.67	-1.70	0.69	-0.93	0.44	-0.72	0.63
	Δcol	-1.78	1.02	-2.36	1.00	-1.28	0.69	-1.51	0.75	-0.81	0.50	-0.87	0.60
	#match	6.59	2.72	9.48	3.15	5.46	1.95	7.32	3.11	2.91	1.36	2.80	1.63
r_2	%x360	300		301		285		278		108		87	
	Δrow	-0.83	0.73	-0.95	0.68	-0.62	0.55	-0.46	0.54	-0.78	0.48	-0.52	0.56
	Δcol	-1.16	0.65	-1.24	0.63	-0.93	0.43	-0.85	0.49	-0.78	0.42	-0.69	0.57
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LNRRW2.c $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=10$ $m_w=m_b=0$

	σ_e	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	243		225		233		180		36		17	
	Δrow	-0.95	0.24	-0.96	0.47	-0.92	0.30	-0.92	0.60	-0.67	0.47	-0.18	0.38
	Δcol	-0.93	0.26	-0.92	0.46	-0.94	0.25	-0.86	0.55	-0.61	0.49	-0.53	0.61
	#match	3.76	0.76	3.98	2.23	3.38	1.02	2.99	1.74	1.97	1.07	1.29	0.46
r_2	%x360	243		227		229		178		35		16	
	Δrow	-0.91	0.29	-0.46	0.50	-0.85	0.36	-0.49	0.51	-0.57	0.49	-0.19	0.39
	Δcol	-0.91	0.29	-0.68	0.47	-0.85	0.36	-0.50	0.51	-0.63	0.48	-0.56	0.61
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LNRRW2.d $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=10$ $m_w=m_b=1$

	σ_e	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	250		245		243		230		86		53	
	Δrow	-0.96	0.26	-1.22	0.54	-1.02	0.25	-1.27	0.59	-0.80	0.48	-0.55	0.53
	Δcol	-0.97	0.26	-1.12	0.55	-1.01	0.28	-1.17	0.57	-0.72	0.45	-0.64	0.62
	#match	3.90	0.77	5.31	2.56	4.00	0.80	5.07	2.76	2.65	1.22	2.19	1.35
r_2	%x360	251		242		245		228		84		54	
	Δrow	-0.90	0.31	-0.48	0.51	-0.85	0.35	-0.39	0.49	-0.70	0.46	-0.35	0.48
	Δcol	-0.94	0.30	-0.90	0.31	-0.93	0.25	-0.79	0.43	-0.74	0.44	-0.54	0.53
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LNRRW3.a $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=1$ $m_w=m_b=0$

	σ_a	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		359		194		101	
	Δrow	-1.31	0.63	-1.89	0.62	-1.20	0.43	-1.60	0.52	-0.74	0.46	-0.43	0.51
	Δcol	-1.31	0.67	-1.53	0.69	-1.13	0.37	-1.51	0.52	-0.72	0.45	-0.51	0.61
	#match	5.31	1.44	8.68	1.46	4.64	0.84	6.94	1.84	2.21	1.09	1.41	0.63
r_2	%x360	360		360		360		359		185		97	
	Δrow	-0.65	0.53	-0.15	0.43	-0.78	0.42	-0.16	0.38	-0.68	0.47	-0.44	0.50
	Δcol	-1.03	0.32	-0.99	0.27	-1.00	0.17	-0.92	0.34	-0.70	0.46	-0.53	0.58
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LNRRW3.b $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=1$ $m_w=m_b=1$

	σ_a	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		360		318		286	
	Δrow	-2.33	0.99	-2.45	0.81	-1.62	0.61	-2.02	0.35	-0.93	0.33	-0.83	0.54
	Δcol	-2.08	1.09	-2.18	0.73	-1.50	0.65	-1.86	0.43	-0.89	0.35	-0.82	0.57
	#match	8.86	2.49	11.01	1.57	7.07	1.87	10.11	1.23	3.40	1.05	2.93	1.45
r_2	%x360	360		360		360		360		320		279	
	Δrow	-0.72	0.73	-0.54	0.65	-0.44	0.51	-0.32	0.50	-0.78	0.44	-0.58	0.54
	Δcol	-1.16	0.62	-1.07	0.46	-0.94	0.41	-0.92	0.30	-0.85	0.39	-0.66	0.51
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LNRRW3.c $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=10$ $m_w=m_b=0$

	σ_a	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		358		359		357		112		52	
	Δrow	-1.00	0.05	-1.32	0.48	-1.01	0.12	-1.20	0.56	-0.70	0.46	-0.37	0.48
	Δcol	-1.00	0.00	-1.13	0.38	-1.00	0.11	-1.14	0.54	-0.71	0.46	-0.46	0.50
	#match	4.01	0.12	6.35	1.99	3.97	0.27	4.34	1.77	2.07	0.98	1.21	0.45
r_2	%x360	360		358		359		357		111		50	
	Δrow	-0.99	0.07	-0.27	0.44	-0.98	0.14	-0.45	0.51	-0.60	0.49	-0.32	0.47
	Δcol	-1.00	0.00	-0.92	0.27	-0.98	0.15	-0.63	0.48	-0.68	0.47	-0.48	0.50
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LNRRW3.d $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=10$ $m_w=m_b=1$

	σ_a	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		359		282		207	
	Δrow	-1.02	0.15	-1.73	0.45	-1.10	0.30	-1.74	0.44	-0.87	0.40	-0.68	0.53
	Δcol	-1.02	0.15	-1.44	0.50	-1.09	0.29	-1.47	0.50	-0.79	0.43	-0.72	0.55
	#match	4.12	0.36	8.11	1.86	4.40	0.65	7.72	1.83	2.88	1.16	2.07	1.11
r_2	%x360	360		360		360		359		283		203	
	Δrow	-0.94	0.24	-0.10	0.30	-0.83	0.37	-0.13	0.34	-0.77	0.44	-0.55	0.54
	Δcol	-1.00	0.05	-1.00	0.11	-1.00	0.12	-0.95	0.28	-0.72	0.45	-0.61	0.52
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LNRRW4.a $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{rv}=1$ $m_w=m_b=0$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	348		343		334		322		131		77	
	Δrow	-1.34	0.68	-1.87	0.83	-1.14	0.45	-1.45	0.63	-0.82	0.41	-0.57	0.52
	Δcol	-1.34	0.75	-1.55	0.83	-1.06	0.39	-1.28	0.62	-0.81	0.39	-0.62	0.54
	#match	5.10	1.76	7.41	2.59	4.31	1.17	5.75	2.61	2.55	1.17	1.65	0.82
r_2	%x360	348		342		335		323		125		74	
	Δrow	-0.73	0.58	-0.36	0.62	-0.73	0.45	-0.31	0.50	-0.74	0.44	-0.43	0.50
	Δcol	-1.03	0.42	-1.04	0.43	-0.94	0.31	-0.86	0.46	-0.78	0.42	-0.49	0.55
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LNRRW4.b $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{rv}=1$ $m_w=m_b=1$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	352		354		346		346		216		187	
	Δrow	-2.17	0.98	-2.49	0.87	-1.49	0.57	-1.83	0.59	-0.92	0.39	-0.91	0.59
	Δcol	-2.04	1.02	-2.25	0.84	-1.39	0.63	-1.66	0.62	-0.88	0.44	-0.91	0.66
	#match	7.95	2.79	9.93	2.73	6.36	2.09	8.53	2.72	3.38	1.30	3.25	1.78
r_2	%x360	353		354		348		345		214		186	
	Δrow	-0.78	0.69	-0.75	0.71	-0.54	0.59	-0.33	0.55	-0.80	0.43	-0.59	0.53
	Δcol	-1.16	0.68	-1.17	0.55	-0.97	0.48	-0.91	0.45	-0.83	0.38	-0.74	0.54
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LNRRW4.c $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{rv}=10$ $m_w=m_b=0$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	323		306		311		274		94		45	
	Δrow	-0.97	0.20	-1.16	0.53	-1.00	0.19	-1.12	0.57	0.79	0.41	-0.44	0.50
	Δcol	-0.97	0.17	-1.04	0.42	-0.98	0.19	-1.06	0.56	-0.77	0.42	-0.53	0.54
	#match	3.89	0.61	5.38	2.57	3.68	0.82	3.94	2.13	2.29	1.08	1.42	0.65
r_2	%x360	323		305		308		266		94		49	
	Δrow	-0.94	0.24	-0.35	0.48	-0.94	0.23	-0.47	0.51	0.72	0.45	-0.41	0.49
	Δcol	-0.97	0.18	-0.77	0.42	-0.91	0.28	-0.61	0.51	-0.71	0.45	-0.55	0.54
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LNRRW4.d $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{rv}=10$ $m_w=m_b=1$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	329		321		327		320		181		135	
	Δrow	-1.00	0.23	-1.45	0.56	-1.05	0.33	-1.45	0.62	-0.90	0.39	-0.80	0.57
	Δcol	-1.01	0.25	-1.33	0.56	-1.04	0.31	-1.33	0.58	-0.84	0.38	-0.81	0.58
	#match	4.12	0.80	6.71	2.73	4.19	0.95	6.32	2.90	3.06	1.20	2.52	1.41
r_2	%x360	330		322		326		318		180		140	
	Δrow	-0.84	0.36	-0.30	0.46	-0.80	0.41	-0.28	0.45	-0.74	0.46	-0.60	0.56
	Δcol	-0.96	0.21	-0.93	0.26	-0.97	0.19	-0.84	0.40	-0.83	0.38	-0.66	0.52
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW1.a $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{tv}=1$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	358		359		348		327		17		5	
	Δrow	-1.14	0.45	-1.90	1.01	-1.00	0.28	-1.03	0.64	-0.53	0.50	-0.40	0.49
	Δcol	0.14	0.42	0.83	0.88	0.11	0.33	0.49	0.62	0.24	0.42	0.00	0.00
	#match	4.27	1.02	5.92	2.09	3.64	1.00	3.20	1.65	1.35	0.59	1.00	0.00
r_2	%x360	358		358		346		321		19		6	
	Δrow	-0.88	0.40	-0.84	0.63	-0.85	0.35	-0.62	0.54	-0.58	0.49	-0.33	0.47
	Δcol	0.07	0.26	0.27	0.56	0.09	0.30	0.28	0.52	0.32	0.46	0.17	0.37
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW1.b $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{tv}=1$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		357		357		70		38	
	Δrow	-1.76	0.79	-2.88	1.02	-1.31	0.51	-1.69	0.72	-0.67	0.50	-0.58	0.59
	Δcol	0.54	0.74	1.28	0.79	0.22	0.55	0.59	0.73	0.30	0.49	0.24	0.58
	#match	6.25	1.81	9.78	2.86	5.14	1.45	6.72	2.40	2.06	1.21	1.84	1.09
r_2	%x360	360		360		358		359		63		39	
	Δrow	-0.86	0.55	-0.97	0.63	-0.77	0.44	-0.66	0.55	-0.62	0.52	-0.46	0.50
	Δcol	0.24	0.50	0.62	0.60	0.09	0.35	0.17	0.52	0.29	0.49	0.33	0.47
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW1.c $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{tv}=10$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	333		305		314		206		6		3	
	Δrow	-0.97	0.18	-0.83	0.40	-0.93	0.26	-0.68	0.51	-0.50	0.50	-0.33	0.47
	Δcol	0.03	0.16	0.32	0.47	0.24	0.42	0.46	0.51	0.17	0.37	0.00	0.00
	#match	3.68	0.79	2.67	1.19	3.05	1.00	1.85	0.99	1.00	0.00	1.00	0.00
r_2	%x360	332		310		311		211		7		3	
	Δrow	-0.95	0.22	-0.60	0.49	-0.77	0.42	-0.54	0.51	-0.43	0.49	-0.33	0.47
	Δcol	0.03	0.16	0.16	0.37	0.18	0.38	0.26	0.45	0.29	0.45	0.00	0.00
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW1.d $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{tv}=10$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	345		339		344		327		31		16	
	Δrow	-0.97	0.18	-1.03	0.35	-0.98	0.17	-1.03	0.51	-0.65	0.48	-0.56	0.50
	Δcol	0.02	0.14	0.19	0.39	0.05	0.21	0.42	0.51	0.26	0.44	0.25	0.43
	#match	3.77	0.70	3.93	1.30	3.75	0.82	3.53	1.71	1.94	0.91	1.63	0.70
r_2	%x360	345		338		347		322		35		16	
	Δrow	-0.94	0.23	-0.74	0.44	-0.91	0.30	-0.58	0.51	-0.63	0.48	-0.50	0.50
	Δcol	0.01	0.14	0.10	0.32	0.04	0.20	0.23	0.48	0.20	0.47	0.19	0.39
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW2.a $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=1$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	287		289		258		246		50		33	
	Δrow	-1.37	0.71	-1.93	0.93	-1.04	0.34	-1.20	0.68	-0.86	0.35	-0.67	0.47
	Δcol	0.31	0.63	0.84	0.82	0.16	0.41	0.52	0.60	0.32	0.47	0.36	0.48
	#match	4.50	1.51	6.29	2.74	3.79	1.16	4.17	2.50	2.16	1.01	1.61	0.78
r_2	%x360	286		288		256		245		53		32	
	Δrow	-0.94	0.57	-0.74	0.70	-0.85	0.39	-0.62	0.59	-0.75	0.43	-0.59	0.49
	Δcol	0.07	0.42	0.20	0.53	0.09	0.30	0.26	0.55	0.32	0.47	0.38	0.48
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW2.b $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=1$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	306		305		286		281		101		83	
	Δrow	-1.98	0.96	-2.63	1.00	-1.40	0.66	-1.73	0.66	-0.89	0.42	-0.82	0.60
	Δcol	0.63	0.79	1.03	0.91	0.30	0.60	0.58	0.69	0.18	0.43	0.34	0.54
	#match	6.59	2.50	9.20	3.29	5.29	1.99	7.01	2.94	2.94	1.28	2.60	1.57
r_2	%x360	305		303		284		278		105		76	
	Δrow	-0.91	0.63	-0.90	0.65	-0.79	0.53	-0.57	0.60	-0.76	0.45	-0.58	0.52
	Δcol	0.26	0.57	0.37	0.66	0.12	0.41	0.18	0.50	0.25	0.43	0.32	0.49
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW2.c $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=10$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	235		216		223		170		36		19	
	Δrow	-0.96	0.19	-0.94	0.45	-0.92	0.27	-0.86	0.61	-0.86	0.35	-0.58	0.49
	Δcol	0.02	0.14	0.27	0.45	0.17	0.37	0.43	0.53	0.28	0.45	0.47	0.50
	#match	3.72	0.76	4.02	2.20	3.32	1.00	2.91	1.84	2.14	1.08	1.32	0.57
r_2	%x360	236		221		225		173		35		18	
	Δrow	-0.95	0.21	-0.49	0.51	-0.82	0.38	-0.49	0.50	-0.77	0.42	-0.56	0.50
	Δcol	0.03	0.16	0.04	0.49	0.13	0.34	0.18	0.47	0.29	0.45	0.33	0.47
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW2.d $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=10$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	243		241		251		236		79		48	
	Δrow	-1.00	0.19	-1.25	0.54	-1.02	0.25	-1.19	0.59	-0.86	0.38	-0.83	0.51
	Δcol	0.03	0.17	0.38	0.53	0.08	0.27	0.48	0.52	0.23	0.45	0.44	0.54
	#match	3.91	0.76	5.33	2.45	3.84	0.94	4.65	2.62	2.72	1.25	2.31	1.19
r_2	%x360	242		240		252		228		78		51	
	Δrow	-0.93	0.25	-0.57	0.50	-0.86	0.35	-0.53	0.51	-0.76	0.43	-0.55	0.60
	Δcol	0.01	0.16	0.02	0.35	0.05	0.23	0.10	0.47	0.24	0.43	0.37	0.48
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW3.a $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=1$ $m_w=m_b=0$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		359		187		117	
	Δrow	-1.47	0.80	-2.00	0.74	-1.17	0.42	-1.62	0.54	-0.76	0.43	-0.60	0.56
	Δcol	0.42	0.71	0.93	0.61	0.17	0.44	0.72	0.58	0.33	0.47	0.36	0.53
	#match	5.50	1.47	8.82	1.60	4.46	1.01	6.78	1.91	2.19	0.99	1.41	0.66
r_2	%x360	360		360		360		359		192		112	
	Δrow	-0.81	0.54	-0.29	0.52	-0.85	0.36	-0.50	0.50	-0.71	0.45	-0.55	0.55
	Δcol	0.11	0.42	0.07	0.49	0.02	0.17	0.02	0.39	0.31	0.46	0.38	0.55
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW3.b $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=1$ $m_w=m_b=1$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		360		318		275	
	Δrow	-2.49	0.97	-2.38	0.77	-1.68	0.58	-2.00	0.39	-0.97	0.33	-0.88	0.56
	Δcol	1.07	0.79	1.02	0.78	0.54	0.67	0.79	0.51	0.13	0.38	0.22	0.56
	#match	9.16	2.45	11.01	1.90	6.89	1.79	9.88	1.51	3.38	1.13	2.89	1.50
r_2	%x360	360		360		360		360		319		276	
	Δrow	-0.87	0.62	-0.54	0.58	-0.69	0.52	-0.34	0.50	-0.80	0.40	-0.61	0.53
	Δcol	0.47	0.61	0.26	0.51	0.17	0.49	0.17	0.47	0.17	0.39	0.25	0.53
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW3.c $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=10$ $m_w=m_b=0$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		359		127		61	
	Δrow	-1.00	0.00	-1.27	0.45	-1.00	0.05	-1.18	0.55	-0.70	0.46	-0.52	0.50
	Δcol	0.00	0.00	0.32	0.49	0.01	0.12	0.50	0.60	0.35	0.48	0.41	0.52
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
r_2	%x360	360		360		360		359		129		64	
	Δrow	-0.98	0.13	-0.33	0.48	-0.97	0.16	-0.56	0.50	-0.67	0.47	-0.52	0.50
	Δcol	0.00	0.00	-0.15	0.45	0.01	0.10	0.19	0.48	0.37	0.48	0.36	0.51
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW3.d $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=10$ $m_w=m_b=1$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		360		273		200	
	Δrow	-1.04	0.19	-1.75	0.45	-1.11	0.31	-1.67	0.47	-0.85	0.37	-0.73	0.56
	Δcol	0.04	0.19	0.78	0.50	0.11	0.32	0.69	0.51	0.19	0.40	0.34	0.52
	#match	4.16	0.76	8.17	1.98	4.38	0.85	7.39	1.86	2.77	1.24	2.04	1.11
r_2	%x360	360		360		360		360		273		203	
	Δrow	-0.91	0.29	-0.29	0.45	-0.83	0.38	-0.45	0.50	-0.76	0.43	-0.61	0.53
	Δcol	-0.01	0.07	-0.02	0.33	0.00	0.00	-0.00	0.35	0.20	0.40	0.31.	0.51
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW4.a $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{tv}=1$ $m_w=m_b=0$

	σ_e	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	347		346		337		346		140		83	
	Δrow	-1.42	0.74	-1.99	0.86	-1.12	0.45	-1.99	0.86	-0.82	0.38	-0.64	0.53
	Δcol	0.37	0.65	0.87	0.74	0.18	0.44	0.87	0.74	0.29	0.45	0.39	0.51
	#match	5.05	1.70	7.60	2.51	4.19	1.21	7.60	2.51	2.35	1.08	1.57	0.79
r_2	%x360	347		346		335		346		135		82	
	Δrow	-0.83	0.51	-0.50	0.66	-0.82	0.41	-0.50	0.66	-0.74	0.44	-0.60	0.51
	Δcol	0.09	0.42	0.14	0.55	0.05	0.31	0.14	0.55	0.27	0.44	0.40	0.54
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW4.b $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{tv}=1$ $m_w=m_b=1$

	σ_e	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	353		353		345		353		217		178	
	Δrow	-2.30	0.97	-2.56	0.92	-1.59	0.68	-2.56	0.92	-0.95	0.40	-0.99	0.61
	Δcol	0.92	0.82	1.09	0.80	0.45	0.67	1.09	0.80	0.18	0.43	0.28	0.62
	#match	8.04	2.68	10.29	2.71	6.26	1.99	10.29	2.71	3.29	1.33	3.32	1.68
r_2	%x360	354		351		347		351		215		177	
	Δrow	-0.89	0.65	-0.74	0.69	-0.69	0.50	-0.74	0.69	-0.79	0.42	-0.64	0.55
	Δcol	0.34	0.60	0.36	0.60	0.16	0.45	0.36	0.60	0.18	0.41	0.23	0.53
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW4.c $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{tv}=10$ $m_w=m_b=0$

	σ_e	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	322		311		318		311		98		52	
	Δrow	-0.98	0.12	-1.18	0.51	-0.97	0.19	-1.18	0.51	-0.80	0.40	-0.62	0.49
	Δcol	0.02	0.15	0.34	0.49	0.08	0.27	0.34	0.49	0.31	0.46	0.44	0.53
	#match	3.83	0.69	5.25	2.49	3.61	0.89	5.25	2.49	2.20	1.11	1.40	0.63
r_2	%x360	319		312		315		312		95		49	
	Δrow	-0.94	0.24	-0.41	0.49	-0.90	0.29	-0.41	0.49	-0.71	0.46	-0.55	0.50
	Δcol	0.03	0.18	-0.05	0.44	0.06	0.24	-0.05	0.44	0.32	0.46	0.45	0.54
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RNRRW4.d $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{tv}=10$ $m_w=m_b=1$

	σ_e	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	327		322		330		322		172		138	
	Δrow	-1.03	0.25	-1.53	0.52	-1.07	0.34	-1.53	0.52	-0.90	0.32	-0.83	0.57
	Δcol	0.06	0.25	0.59	0.55	0.11	0.32	0.59	0.55	0.13	0.36	0.39	0.53
	#match	4.13	0.90	6.83	2.65	4.22	1.12	6.83	2.65	3.07	1.19	2.44	1.31
r_2	%x360	328		324		329		324		174		139	
	Δrow	-0.89	0.34	-0.42	0.49	-0.81	0.40	-0.42	0.49	-0.78	0.41	-0.63	0.55
	Δcol	0.01	0.15	0.02	0.30	0.02	0.21	0.02	0.30	0.18	0.38	0.31	0.48
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE1.a $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{tv}=1$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	358		359		358		357		54		40	
	Δrow	-1.15	0.44	-2.69	1.03	-1.15	0.44	-1.52	0.71	-0.04	0.82	0.05	0.80
	Δcol	-0.99	0.47	-1.99	1.12	-0.99	0.47	-1.04	0.79	-0.06	0.80	-0.08	0.75
	#match	9.31	1.83	16.13	4.26	9.31	1.83	10.24	3.34	2.20	1.74	2.08	1.42
r_2	%x360	357		359		357		358		49		40	
	Δrow	-0.17	0.41	-0.75	0.59	-0.17	0.41	-0.28	0.56	0.14	0.76	0.20	0.71
	Δcol	-0.06	0.31	-0.36	0.74	-0.06	0.31	-0.11	0.53	-0.16	0.74	-0.10	0.66
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE1.b $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{tv}=1$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		360		167		142	
	Δrow	-1.73	0.66	-3.42	0.84	-1.73	0.66	-2.18	0.77	-0.48	0.83	-0.34	0.73
	Δcol	-1.12	0.90	-2.51	1.08	-1.12	0.90	-1.41	0.99	-0.32	0.80	-0.23	0.83
	#match	13.42	2.79	23.38	4.65	13.42	2.79	17.37	3.89	4.10	2.82	3.50	2.47
r_2	%x360	360		360		360		360		166		145	
	Δrow	-0.38	0.50	-0.76	0.69	-0.38	0.50	-0.36	0.60	0.07	0.70	0.10	0.65
	Δcol	-0.18	0.52	-0.71	0.72	-0.18	0.52	-0.21	0.62	-0.13	0.67	-0.10	0.71
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE1.c $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{tv}=10$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	350		351		350		344		24		24	
	Δrow	-0.95	0.27	-1.04	0.41	-0.95	0.27	-0.93	0.54	-0.21	0.82	0.17	0.80
	Δcol	-0.89	0.33	-0.85	0.54	-0.89	0.33	-0.67	0.60	0.00	0.82	-0.17	0.75
	#match	7.79	1.95	7.91	2.35	7.79	1.95	6.15	2.70	2.33	1.60	1.54	0.82
r_2	%x360	350		351		350		346		26		20	
	Δrow	-0.05	0.26	-0.06	0.32	-0.05	0.26	-0.06	0.46	0.00	0.68	0.35	0.73
	Δcol	-0.08	0.31	-0.12	0.40	-0.08	0.31	-0.23	0.49	-0.23	0.75	0.05	0.67
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE1.d $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{tv}=10$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	356		358		356		356		98		70	
	Δrow	-1.09	0.31	-1.45	0.54	-1.09	0.31	-1.43	0.56	-0.24	0.81	-0.26	0.73
	Δcol	-0.97	0.39	-1.25	0.65	-0.97	0.39	-0.99	0.70	-0.20	0.76	-0.01	0.84
	#match	9.16	1.40	10.96	3.03	9.16	1.40	10.66	3.52	3.01	2.08	2.77	1.96
r_2	%x360	356		357		356		355		97		76	
	Δrow	-0.15	0.37	-0.47	0.51	-0.15	0.37	-0.30	0.49	0.11	0.74	0.12	0.78
	Δcol	-0.03	0.17	-0.03	0.31	-0.03	0.17	-0.08	0.44	-0.10	0.65	-0.03	0.74
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE2.a $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=1$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	303		305		283		287		93		79	
	Δrow	-1.81	1.00	-2.51	1.06	-1.20	0.55	-1.49	0.79	-0.57	0.68	-0.41	0.65
	Δcol	-1.44	1.05	-1.89	1.09	-1.03	0.64	-1.02	0.87	-0.61	0.57	-0.32	0.61
	#match	11.64	3.65	15.67	5.24	9.57	2.70	10.96	4.82	4.63	2.80	3.59	2.25
r_2	%x360	304		303		288		288		92		74	
	Δrow	-0.63	0.75	-0.64	0.69	-0.24	0.48	-0.34	0.58	-0.08	0.52	0.03	0.46
	Δcol	-0.29	0.70	-0.26	0.78	-0.04	0.38	0.01	0.57	-0.30	0.55	-0.16	0.49
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE2.b $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=1$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	308		310		304		304		161		147	
	Δrow	-2.55	1.04	-3.07	1.05	-1.75	0.76	-1.99	0.78	-0.82	0.75	-0.81	0.80
	Δcol	-1.87	1.24	-2.49	1.10	-1.20	0.93	-1.51	1.08	-0.52	0.77	-0.44	0.85
	#match	17.34	5.41	21.32	5.99	13.48	4.37	16.30	5.72	6.30	3.60	6.50	4.30
r_2	%x360	311		309		307		302		160		146	
	Δrow	-0.66	0.68	-0.70	0.84	-0.32	0.65	-0.30	0.62	-0.03	0.63	-0.05	0.60
	Δcol	-0.50	0.75	-0.63	0.80	-0.14	0.63	-0.22	0.70	-0.13	0.54	-0.13	0.62
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE2.c $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=10$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	265		261		259		250		67		56	
	Δrow	-0.98	0.31	-1.25	0.59	-0.95	0.32	-1.12	0.68	-0.46	0.72	-0.23	0.57
	Δcol	-0.91	0.41	-0.82	0.63	-0.88	0.44	-0.70	0.65	-0.42	0.78	-0.25	0.66
	#match	8.31	2.02	9.77	4.43	7.93	2.23	7.99	4.13	4.33	2.59	2.91	1.83
r_2	%x360	266		262		257		251		67		58	
	Δrow	-0.02	0.20	-0.14	0.36	-0.08	0.27	-0.13	0.46	-0.01	0.56	0.00	0.49
	Δcol	-0.02	0.35	0.07	0.53	-0.05	0.25	-0.10	0.50	-0.19	0.60	-0.14	0.54
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE2.d $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=10$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	273		274		271		268		127		115	
	Δrow	-1.10	0.44	-1.56	0.66	-1.17	0.44	-1.52	0.60	-0.65	0.70	-0.61	0.80
	Δcol	-1.04	0.49	-1.39	0.69	-0.95	0.50	-0.97	0.71	-0.59	0.67	-0.37	0.82
	#match	8.97	2.35	12.30	4.99	9.41	2.67	11.92	5.08	5.69	3.37	5.17	3.52
r_2	%x360	271		274		267		268		126		119	
	Δrow	-0.21	0.46	-0.40	0.52	-0.23	0.44	-0.33	0.49	-0.01	0.50	-0.01	0.60
	Δcol	-0.04	0.38	-0.03	0.51	-0.02	0.24	0.04	0.49	-0.12	0.51	-0.18	0.66
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE3.a $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=1$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_g	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		360		290		260	
	Δrow	-2.27	1.04	-2.41	0.84	-1.52	0.64	-1.94	0.33	-0.64	0.63	-0.47	0.63
	Δcol	-1.57	1.18	-1.87	0.82	-1.12	0.66	-1.46	0.64	-0.53	0.66	-0.32	0.64
	#match	14.85	3.51	18.98	2.29	11.50	1.94	16.06	2.39	5.04	2.51	3.63	2.13
r_2	%x360	360		360		360		360		291		260	
	Δrow	-0.68	0.54	-0.26	0.47	-0.49	0.51	-0.34	0.50	-0.01	0.53	-0.08	0.53
	Δcol	-0.26	0.73	0.11	0.56	0.02	0.26	0.26	0.53	-0.20	0.54	-0.18	0.55
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE3.b $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=1$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_g	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		360		349		344	
	Δrow	-3.23	0.96	-2.86	0.92	-2.14	0.68	-2.15	0.45	-1.02	0.49	-1.07	0.64
	Δcol	-2.12	1.25	-2.43	0.89	-1.36	0.92	-1.90	0.62	-0.77	0.63	-0.62	0.81
	#match	22.58	4.20	23.00	3.23	17.24	2.86	20.54	1.52	8.54	2.34	8.65	3.41
r_2	%x360	360		360		360		360		353		345	
	Δrow	-0.65	0.68	-0.29	0.65	-0.37	0.58	-0.09	0.44	-0.11	0.44	-0.06	0.55
	Δcol	-0.65	0.71	-0.39	0.71	-0.21	0.61	-0.18	0.49	-0.06	0.37	-0.12	0.59
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE3.c $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=10$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_g	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		360		232		189	
	Δrow	-1.04	0.21	-1.80	0.40	-1.09	0.28	-1.60	0.49	-0.56	0.66	-0.26	0.64
	Δcol	-0.99	0.26	-0.98	0.68	-1.01	0.32	-1.01	0.62	-0.47	0.68	-0.29	0.58
	#match	9.25	0.59	14.84	3.19	9.32	0.78	12.12	2.89	4.08	2.42	2.82	1.82
r_2	%x360	360		360		360		359		233		190	
	Δrow	-0.04	0.20	-0.21	0.41	-0.13	0.33	-0.34	0.52	-0.06	0.58	-0.02	0.55
	Δcol	0.00	0.00	0.43	0.51	-0.00	0.05	0.11	0.45	-0.21	0.60	-0.16	0.46
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE3.d $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=10$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_g	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		360		336		327	
	Δrow	-1.35	0.48	-2.00	0.32	-1.46	0.50	-1.95	0.22	-0.86	0.51	-0.80	0.67
	Δcol	-1.24	0.56	-1.79	0.52	-1.06	0.61	-1.45	0.59	-0.68	0.63	-0.48	0.74
	#match	10.62	1.61	17.66	2.18	11.43	1.93	17.27	2.16	7.09	2.51	6.17	2.92
r_2	%x360	360		360		360		360		333		322	
	Δrow	-0.46	0.50	-0.25	0.43	-0.48	0.51	-0.27	0.44	-0.02	0.41	-0.02	0.49
	Δcol	0.01	0.14	0.12	0.35	0.00	0.19	0.20	0.52	-0.14	0.46	-0.16	0.57
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE4.a $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{tv}=1$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	355		355		351		349		186		167	
	Δrow	-2.03	1.01	-2.42	0.90	-1.35	0.59	-1.71	0.61	-0.71	0.61	-0.61	0.67
	Δcol	-1.50	1.03	-1.89	0.95	-1.11	0.65	-1.31	0.80	-0.63	0.60	-0.50	0.63
	#match	13.42	4.12	17.28	4.33	10.50	2.76	13.57	4.61	5.56	2.93	4.37	2.77
r_2	%x360	355		354		351		350		182		170	
	Δrow	-0.64	0.65	-0.44	0.65	-0.36	0.53	-0.29	0.51	-0.04	0.49	-0.13	0.52
	Δcol	-0.23	0.73	-0.16	0.72	-0.06	0.49	0.11	0.62	-0.14	0.49	-0.27	0.53
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE4.b $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{tv}=1$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	355		356		353		353		267		260	
	Δrow	-2.87	1.04	-3.01	0.96	-1.95	0.77	-2.12	0.62	-0.97	0.64	-0.99	0.80
	Δcol	-1.98	1.19	-2.36	1.07	-1.33	0.96	-1.65	0.88	-0.72	0.70	-0.60	0.88
	#match	19.92	5.28	22.24	4.51	15.69	4.06	18.51	4.04	7.76	3.26	8.16	4.55
r_2	%x360	355		355		353		354		267		258	
	Δrow	-0.68	0.73	-0.55	0.85	-0.37	0.61	-0.16	0.42	-0.10	0.52	-0.10	0.59
	Δcol	-0.52	0.73	-0.50	0.83	-0.21	0.64	-0.16	0.57	-0.09	0.45	-0.13	0.62
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE4.c $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{tv}=10$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	335		336		333		326		154		132	
	Δrow	-1.01	0.25	-1.52	0.59	-1.03	0.35	-1.36	0.61	-0.66	0.64	-0.45	0.63
	Δcol	-0.96	0.27	-0.94	0.69	-0.96	0.38	-0.90	0.67	-0.63	0.56	-0.37	0.53
	#match	8.89	1.51	12.43	5.02	8.68	1.94	10.44	4.41	5.03	2.70	3.67	2.36
r_2	%x360	335		333		334		329		151		133	
	Δrow	-0.04	0.22	-0.14	0.37	-0.07	0.30	-0.22	0.53	0.00	0.49	-0.05	0.46
	Δcol	-0.01	0.15	0.25	0.58	-0.05	0.27	0.02	0.53	-0.25	0.53	-0.19	0.46
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

LWIDE4.d $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{tv}=10$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	338		341		340		336		232		223	
	Δrow	-1.31	0.50	-1.78	0.55	-1.33	0.52	-1.73	0.50	-0.91	0.56	-0.83	0.75
	Δcol	-1.19	0.62	-1.58	0.60	-0.98	0.57	-1.29	0.69	-0.65	0.63	-0.53	0.74
	#match	10.32	2.27	14.82	4.69	10.62	2.59	14.76	4.68	6.95	3.01	6.54	3.81
r_2	%x360	338		340		341		340		228		219	
	Δrow	-0.36	0.49	-0.31	0.48	-0.34	0.50	-0.29	0.51	-0.08	0.47	-0.06	0.55
	Δcol	-0.01	0.21	0.03	0.32	-0.01	0.25	0.07	0.46	-0.11	0.45	-0.20	0.53
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE1.a $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{iv}=1$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		359		359		51		46	
	Δrow	-1.25	0.56	-2.18	0.97	-1.06	0.28	-1.19	0.47	-0.33	0.81	-0.09	0.80
	Δcol	-0.96	1.30	0.85	1.71	-0.98	0.71	-0.79	1.35	-0.43	0.66	-0.37	0.70
	#match	11.41	2.07	14.76	2.52	9.40	1.66	10.26	2.73	2.47	1.91	2.02	1.71
r_2	%x360	360		360		359		359		58		45	
	Δrow	-0.33	0.57	-0.43	0.52	-0.18	0.39	-0.27	0.46	-0.21	0.74	-0.11	0.67
	Δcol	-0.22	0.51	-0.13	0.54	-0.09	0.32	-0.25	0.55	-0.28	0.55	-0.31	0.69
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE1.b $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{iv}=1$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		359		359		166		159	
	Δrow	-1.95	0.88	-3.09	0.95	-1.43	0.61	-1.84	0.74	-0.58	0.70	-0.47	0.77
	Δcol	0.00	1.56	1.02	0.99	-0.86	1.29	-0.03	1.48	-0.59	0.79	-0.56	0.93
	#match	16.13	2.89	21.21	3.06	12.97	2.47	15.55	2.84	4.55	2.76	3.95	2.65
r_2	%x360	360		360		359		359		167		151	
	Δrow	-0.32	0.53	-0.43	0.57	-0.33	0.51	-0.23	0.51	-0.07	0.63	-0.11	0.69
	Δcol	-0.19	0.60	0.12	0.66	-0.14	0.54	-0.11	0.47	-0.26	0.67	-0.35	0.70
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE1.c $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{iv}=10$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	355		355		349		346		22		20	
	Δrow	-1.00	0.08	-1.01	0.18	-0.95	0.23	-0.87	0.37	0.00	0.80	-0.05	0.80
	Δcol	-0.97	0.30	-1.05	0.82	-0.82	0.46	-0.73	0.81	-0.18	0.57	-0.30	0.64
	#match	8.73	1.21	8.78	2.39	7.99	1.80	6.81	2.71	2.32	1.89	1.80	1.33
r_2	%x360	355		353		352		350		23		17	
	Δrow	-0.04	0.20	-0.21	0.43	-0.05	0.26	-0.13	0.39	0.04	0.75	-0.18	0.78
	Δcol	-0.05	0.22	-0.26	0.49	-0.11	0.34	-0.43	0.54	-0.04	0.46	-0.24	0.55
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE1.d $|\mu_1-\mu_2| = 60$ $\sigma_b = 10$ $N_{iv}=10$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	357		357		358		355		104		91	
	Δrow	-1.04	0.22	-1.32	0.49	-1.04	0.21	-1.15	0.37	-0.44	0.81	-0.37	0.79
	Δcol	-1.14	0.64	-0.82	1.39	-1.09	0.57	-1.06	1.06	-0.48	0.73	-0.53	0.72
	#match	9.48	1.35	11.73	2.43	9.30	1.57	10.70	2.74	3.56	2.50	2.96	2.06
r_2	%x360	357		359		358		357		102		88	
	Δrow	-0.23	0.42	-0.35	0.48	-0.23	0.42	-0.28	0.46	-0.08	0.68	-0.07	0.75
	Δcol	-0.07	0.29	-0.25	0.47	-0.05	0.27	-0.30	0.51	-0.32	0.64	-0.40	0.70
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE2.a $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=1$ $m_w=m_b=0$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	308		309		296		295		99		89	
	Δrow	-1.53	0.84	-2.06	1.01	-1.00	0.41	-1.16	0.55	-0.66	0.68	-0.57	0.70
	Δcol	-0.59	1.55	0.53	1.69	-1.01	0.82	-0.66	1.42	-0.62	0.66	-0.55	0.73
	#match	11.47	3.40	13.74	4.01	9.03	2.85	10.20	3.93	4.81	2.90	3.65	2.50
r_2	%x360	308		305		291		293		95		91	
	Δrow	-0.38	0.63	-0.45	0.64	-0.20	0.47	-0.21	0.50	-0.15	0.60	-0.15	0.63
	Δcol	-0.27	0.61	-0.15	0.56	-0.18	0.47	-0.27	0.56	-0.26	0.51	-0.38	0.64
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE2.b $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=1$ $m_w=m_b=1$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	314		315		305		307		151		141	
	Δrow	-2.16	1.00	-2.86	1.02	-1.51	0.79	-1.77	0.77	-0.85	0.56	-0.90	0.52
	Δcol	0.06	1.49	0.81	1.14	-0.70	1.39	-0.07	1.49	-0.75	0.81	-0.91	0.97
	#match	16.37	4.62	18.84	5.31	12.50	3.73	14.55	4.54	6.81	3.16	6.86	3.47
r_2	%x360	316		314		302		305		149		145	
	Δrow	-0.45	0.76	-0.49	0.75	-0.32	0.58	-0.17	0.48	-0.10	0.53	-0.18	0.56
	Δcol	-0.21	0.71	0.04	0.69	-0.17	0.65	-0.19	0.58	-0.13	0.51	-0.34	0.63
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE2.c $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=10$ $m_w=m_b=1$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	261		261		256		252		73		64	
	Δrow	-0.97	0.22	-1.04	0.37	-0.94	0.29	-0.91	0.43	-0.60	0.68	-0.41	0.76
	Δcol	-1.02	0.37	-0.96	1.11	-0.89	0.45	-0.90	0.95	-0.51	0.62	-0.33	0.73
	#match	8.62	1.87	9.58	3.26	8.17	2.02	8.08	3.49	4.07	2.80	2.75	2.13
r_2	%x360	261		260		254		255		76		61	
	Δrow	-0.10	0.33	-0.20	0.43	-0.06	0.29	-0.15	0.44	-0.09	0.65	-0.10	0.59
	Δcol	-0.07	0.30	-0.33	0.50	-0.10	0.33	-0.43	0.53	-0.22	0.55	-0.26	0.60
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE2.d $|\mu_1-\mu_2| = 60$ $\sigma_b = 30$ $N_{tv}=10$ $m_w=m_b=1$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	270		268		271		267		121		113	
	Δrow	-1.07	0.36	-1.46	0.57	-1.00	0.34	-1.21	0.52	-0.79	0.61	-0.79	0.57
	Δcol	-1.05	0.86	-0.34	1.43	-1.16	0.65	-0.85	1.21	-0.77	0.65	-0.73	0.82
	#match	9.60	2.50	12.27	3.67	9.39	2.46	11.26	3.72	6.26	3.02	5.65	3.13
r_2	%x360	268		269		269		269		121		112	
	Δrow	-0.24	0.44	-0.28	0.46	-0.23	0.45	-0.21	0.42	-0.10	0.52	-0.12	0.55
	Δcol	-0.14	0.36	-0.16	0.38	-0.14	0.37	-0.26	0.50	-0.17	0.52	-0.32	0.62
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE3.a $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=1$ $m_w=m_b=0$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		360		295		275	
	Δrow	-1.65	0.89	-2.12	0.84	-1.15	0.41	-1.47	0.53	-0.69	0.60	-0.60	0.61
	Δcol	-0.26	1.92	1.15	1.53	-1.04	1.16	-0.17	1.82	-0.60	0.67	-0.60	0.68
	#match	14.00	2.28	15.43	1.55	11.23	1.69	13.68	1.62	5.20	2.62	4.07	2.39
r_2	%x360	360		360		360		360		296		278	
	Δrow	-0.35	0.51	-0.25	0.43	-0.27	0.48	-0.15	0.37	-0.04	0.52	-0.12	0.53
	Δcol	-0.15	0.65	0.08	0.35	-0.18	0.43	-0.12	0.36	-0.20	0.52	-0.36	0.54
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE3.b $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=1$ $m_w=m_b=1$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		360		350		345	
	Δrow	-2.75	1.04	-2.82	0.96	-1.79	0.65	-2.03	0.41	-1.00	0.29	-0.99	0.37
	Δcol	0.83	1.03	1.09	0.63	-0.10	1.48	0.62	0.86	-0.99	0.68	-1.08	0.87
	#match	20.28	3.06	19.59	2.89	15.69	2.37	17.54	1.62	8.48	2.07	8.45	2.82
r_2	%x360	360		360		360		360		351		344	
	Δrow	-0.36	0.63	-0.45	0.64	-0.18	0.39	-0.16	0.41	-0.13	0.41	-0.13	0.51
	Δcol	0.04	0.67	0.23	0.62	-0.14	0.43	-0.03	0.37	-0.12	0.36	-0.29	0.50
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE3.c $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=10$ $m_w=m_b=0$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		360		242		211	
	Δrow	-1.00	0.00	-1.15	0.36	-1.00	0.00	-1.12	0.32	-0.57	0.68	-0.46	0.71
	Δcol	-1.23	0.43	-1.31	1.39	-1.12	0.33	-1.09	1.28	-0.56	0.64	-0.52	0.64
	#match	9.62	1.01	12.78	1.52	9.43	0.92	11.41	2.10	4.24	2.50	3.27	2.00
r_2	%x360	360		360		360		360		240		213	
	Δrow	-0.17	0.38	-0.13	0.34	-0.09	0.28	-0.21	0.42	-0.05	0.59	-0.11	0.59
	Δcol	-0.06	0.24	-0.27	0.46	-0.06	0.23	-0.32	0.50	-0.26	0.55	-0.34	0.58
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE3.d $|\mu_1-\mu_2| = 100$ $\sigma_b = 10$ $N_{tv}=10$ $m_w=m_b=1$

	σ_w	5.0				10.0				25.0			
	σ_n	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	360		360		360		360		339		325	
	Δrow	-1.21	0.41	-1.87	0.40	-1.15	0.36	-1.67	0.47	-0.90	0.40	-0.86	0.44
	Δcol	-1.07	1.22	0.56	1.12	-1.21	0.98	0.01	1.42	-0.81	0.60	-0.90	0.72
	#match	11.62	1.67	15.38	1.11	11.28	1.68	14.81	1.55	7.24	2.53	6.68	2.79
r_2	%x360	360		360		360		360		339		330	
	Δrow	-0.32	0.47	-0.07	0.25	-0.32	0.47	-0.08	0.28	-0.05	0.37	-0.11	0.48
	Δcol	-0.24	0.43	-0.04	0.22	-0.19	0.39	-0.08	0.29	-0.10	0.38	-0.28	0.55
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE4.a $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{tv}=1$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	354		353		347		349		193		181	
	Δrow	-1.53	0.89	-2.10	0.90	-1.11	0.40	-1.35	0.58	-0.72	0.59	-0.64	0.61
	Δcol	-0.51	1.69	0.91	1.65	-0.94	1.08	-0.36	1.68	-0.67	0.64	-0.66	0.68
	#match	12.70	2.94	14.86	2.67	10.39	2.35	12.02	3.16	5.66	2.85	4.63	2.68
r_2	%x360	355		354		348		350		192		182	
	Δrow	-0.37	0.57	-0.33	0.54	-0.23	0.44	-0.19	0.43	-0.05	0.49	-0.10	0.55
	Δcol	-0.25	0.58	-0.03	0.51	-0.15	0.40	-0.17	0.48	-0.15	0.47	-0.34	0.58
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE4.b $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{tv}=1$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	356		356		353		354		269		259	
	Δrow	-2.58	1.07	-2.76	0.93	-1.65	0.70	-1.96	0.65	-0.88	0.57	-0.94	0.57
	Δcol	0.54	1.36	0.99	0.94	-0.41	1.51	0.34	1.23	-0.83	0.75	-0.95	0.97
	#match	18.51	4.17	19.42	3.61	14.37	3.02	16.27	3.27	7.59	3.09	7.98	3.64
r_2	%x360	356		356		352		354		270		262	
	Δrow	-0.40	0.60	-0.44	0.68	-0.25	0.57	-0.19	0.45	-0.13	0.53	-0.18	0.56
	Δcol	-0.08	0.62	0.12	0.65	-0.15	0.51	-0.08	0.49	-0.10	0.39	-0.28	0.56
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE4.c $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{tv}=10$ $m_w=m_b=0$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	333		332		334		332		156		144	
	Δrow	-1.00	0.05	-1.11	0.35	-0.97	0.19	-1.00	0.37	-0.68	0.61	-0.62	0.61
	Δcol	-1.12	0.44	-1.11	1.34	-1.01	0.47	-1.03	1.09	-0.65	0.61	-0.57	0.70
	#match	9.24	1.45	11.48	2.96	8.84	1.84	9.81	3.45	5.26	2.88	3.89	2.31
r_2	%x360	333		333		334		330		155		145	
	Δrow	-0.11	0.32	-0.15	0.38	-0.07	0.31	-0.18	0.40	-0.08	0.55	-0.15	0.53
	Δcol	-0.09	0.29	-0.32	0.49	-0.05	0.24	-0.36	0.51	-0.17	0.50	-0.38	0.59
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

RWIDE4.d $|\mu_1-\mu_2| = 100$ $\sigma_b = 30$ $N_{tv}=10$ $m_w=m_b=1$

	σ_n	5.0				10.0				25.0			
	σ_p	0.5		2.0		0.5		2.0		0.5		2.0	
r_1	%x360	334		336		340		337		240		231	
	Δrow	-1.17	0.39	-1.65	0.52	-1.11	0.37	-1.45	0.53	-0.84	0.53	-0.81	0.58
	Δcol	-1.06	1.08	0.07	1.36	-1.11	0.97	-0.42	1.42	-0.80	0.66	-0.80	0.83
	#match	10.98	2.08	14.00	2.81	10.59	2.33	13.15	3.18	6.95	2.92	6.66	3.47
r_2	%x360	337		339		339		338		240		227	
	Δrow	-0.28	0.45	-0.16	0.39	-0.28	0.46	-0.14	0.35	-0.08	0.49	-0.15	0.57
	Δcol	-0.18	0.40	-0.11	0.34	-0.15	0.37	-0.18	0.42	-0.12	0.43	-0.29	0.55
	#match	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00

Appendix 8 Applied measuring scenarios

Scenario with regular pattern.

update region row index	alignment region col index 0										1					
	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*	*1*	*2*	*3*	*4*	*5*
0 *0*	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
5	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
8	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
9	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1 *0*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
3	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
6	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
8	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
9	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2 *0*	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
5	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
8	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
9	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3 *0*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
3	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
6	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
8	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
9	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4 *0*	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
5	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
8	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
9	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
5 *0*	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
3	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
5	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
6	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
8	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
9	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
6 *0*	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Scenario with alignment features being randomly distributed. Each alignment region will comprise an alignment feature in average.

update region row index	alignment region col index 0										1					
	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*	*1*	*2*	*3*	*4*	*5*
0	*0*	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
	2	X	P	X	P	X	P	X	P	X	P	X	P	X	P	X	P
	3	P	X	P	X	P	X	P	X	P	X	P	X	P	X	P	X
	4	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
	5	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
	6	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
	7	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
	8	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
	9	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
1	*0*	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
	1	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
	2	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
	3	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
	4	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
	5	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
	6	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
	7	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
	8	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
	9	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
2	*0*	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
	1	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
	2	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
	3	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
	4	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
	5	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
	6	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
	7	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
	8	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
	9	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
3	*0*	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
	1	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
	2	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
	3	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
	4	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
	5	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
	6	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
	7	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
	8	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
	9	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
4	*0*	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
	1	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
	2	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
	3	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
	4	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
	5	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
	6	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
	7	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
	8	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
	9	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
5	*0*	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
	1	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
	2	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
	3	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
	4	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
	5	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
	6	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
	7	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
	8	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
	9	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
6	*0*	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
	1	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
	2	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
	3	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X

Scenario with alignment features being randomly distributed, recognition rate is 50%.

update region row index	alignment region col index 0										1					
	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*	*1*	*2*	*3*	*4*	*5*
0*0*	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2	X	X	X	P	X	P	X	P	X	P	X	P	X	P	X	P
3	P	X	P	X	P	X	P	X	P	X	P	X	P	X	P	X
4	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
5	X	X	X	X	X	X	P	X	X	X	P	X	X	X	P	X
6	P	X	X	X	X	X	X	P	X	X	X	X	X	X	X	P
7	X	X	X	X	P	X	X	X	X	X	X	X	P	X	X	X
8	X	P	X	X	X	P	X	X	X	X	X	X	X	P	X	X
9	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
1*0*	P	X	X	P	X	X	X	X	X	X	X	P	X	X	X	P
1	X	X	X	X	X	X	X	X	X	X	X	X	P	X	X	X
2	X	P	X	X	X	X	X	X	X	P	X	X	X	P	X	X
3	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
4	P	X	X	P	X	X	X	X	X	X	X	P	X	X	X	P
5	X	X	X	X	P	X	X	X	X	X	X	X	P	X	X	X
6	X	P	X	X	X	P	X	X	X	X	X	X	X	P	X	X
7	X	X	P	X	X	X	P	X	X	X	X	X	X	X	P	X
8	P	X	X	P	X	X	X	P	X	X	X	X	X	X	X	P
9	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
2*0*	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
1	X	X	P	X	X	X	X	X	X	X	P	X	X	X	P	X
2	P	X	X	X	X	X	X	P	X	X	X	P	X	X	X	P
3	X	X	X	X	X	X	X	X	P	X	X	X	P	X	X	X
4	X	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X
5	X	X	X	X	X	X	P	X	X	X	P	X	X	X	P	X
6	P	X	X	X	X	X	X	X	X	X	X	P	X	X	X	P
7	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
8	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
9	X	X	P	X	X	X	P	X	X	X	X	X	X	X	P	X
3*0*	P	X	X	X	X	X	X	P	X	X	X	P	X	X	X	P
1	X	X	X	X	P	X	X	X	X	X	X	X	P	X	X	X
2	X	X	X	X	X	X	X	X	X	P	X	X	X	P	X	X
3	X	X	X	X	X	X	X	X	X	X	P	X	X	X	P	X
4	P	X	X	X	X	X	X	P	X	X	X	P	X	X	X	P
5	X	X	X	X	P	X	X	X	X	X	X	X	P	X	X	X
6	X	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X
7	X	X	P	X	X	X	X	P	X	X	X	P	X	X	P	X
8	X	X	X	P	X	X	X	X	X	X	X	P	X	X	X	P
9	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
4*0*	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
1	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
2	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
3	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
4	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
5	X	X	X	X	X	X	P	X	X	X	P	X	X	X	P	X
6	P	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P
7	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
8	X	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X
9	X	X	X	X	X	X	X	X	X	X	P	X	X	X	P	X
5*0*	P	X	X	X	X	X	X	P	X	X	X	P	X	X	X	P
1	X	X	X	X	X	X	X	X	X	X	X	X	P	X	X	X
2	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X	X
3	X	X	P	X	X	X	X	X	X	X	P	X	X	X	P	X
4	X	X	X	X	X	X	X	P	X	X	X	P	X	X	X	P
5	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X
6	X	X	X	X	X	X	X	X	X	P	X	X	X	P	X	X
7	X	X	P	X	X	X	P	X	X	X	P	X	X	X	P	X
8	X	X	X	X	X	X	X	X	X	X	X	P	X	X	X	P
9	X	X	X	X	X	X	X	X	P	X	X	X	P	X	X	X
6*0*	X	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X
1	X	X	X	X	X	X	P	X	X	X	P	X	X	X	P	X
2	X	X	X	X	X	X	X	P	X	X	X	P	X	X	X	P
3	X	X	X	X	P	X	X	X	P	X	X	X	P	X	X	X

Appendix 9 User's guide

9.1 SBIPCNFG

DRC evaluation requires the possibility to process generated test images by a DRC filter. The SBIP image processing boards contains two double rows of connector pins. In this way connections with the most vital signals on the SBIP can be made. This possibility has been made for implementing functions on a plug-in board as option, such as the DRC filter. The DRC filter is mounted in such a way that the output of the A/D converter for analog video signals serve as input for the filter. The SBIP in which the DRC filter is plugged in, will be referred to as SBIP0. In order to provide for an input image for the DRC filter, an analog video signal of the test image to be evaluated is needed. Test images are therefore drawn in the frame buffer of a second SBIP image processing board, which is referred to as SBIP1. Pixels contained in frame buffer of SBIP1 and which are acquired in the frame buffer of SBIP0, are desired to exhibit the following relationship (the DRC filter is set transparent):

$$p_0(x,y) = p_1(x,y)$$

However, due to the analog path interconnecting the SBIPs' frame buffers, the following relationship exists

$$p_0(x,y) = a.p_1(x-x_{delay}, y-y_{delay}) + \eta$$

SBIPCNFG determines the unknown offset parameters x_{delay} and y_{delay} and attenuation factor a. The noise component η is relatively small, that is $\eta<3$ and for ease of simplicity not taken into account. It must be explicitly stated that video mode of both the SBIPs have to be set in non-interlaced mode. If not, pixels become blurred.

9.2 DUAL_DRC

Dual_DRC has been written in order to demonstrate paired template matching. Distinguished are master recognitions, slave recognitions, and the combination of both of them. Each of them can be represented by a picture function, $f_1(x,y)$, $f_2(x,y)$ and $f_3(x,y)$ respectively. See Figure 81.

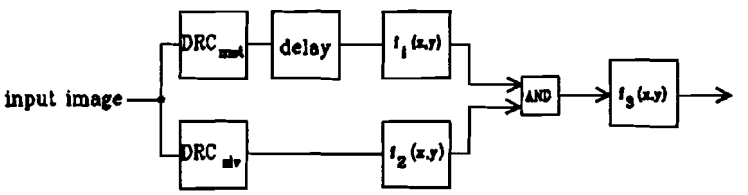


Figure 81. *Paired template matching*

Each picture function may comprise dilate, and/or erode (with stop condition). For the software environment, the input image may entail images acquired by CCD camera, or stored images. The operations can be invoked by following the menu structure.

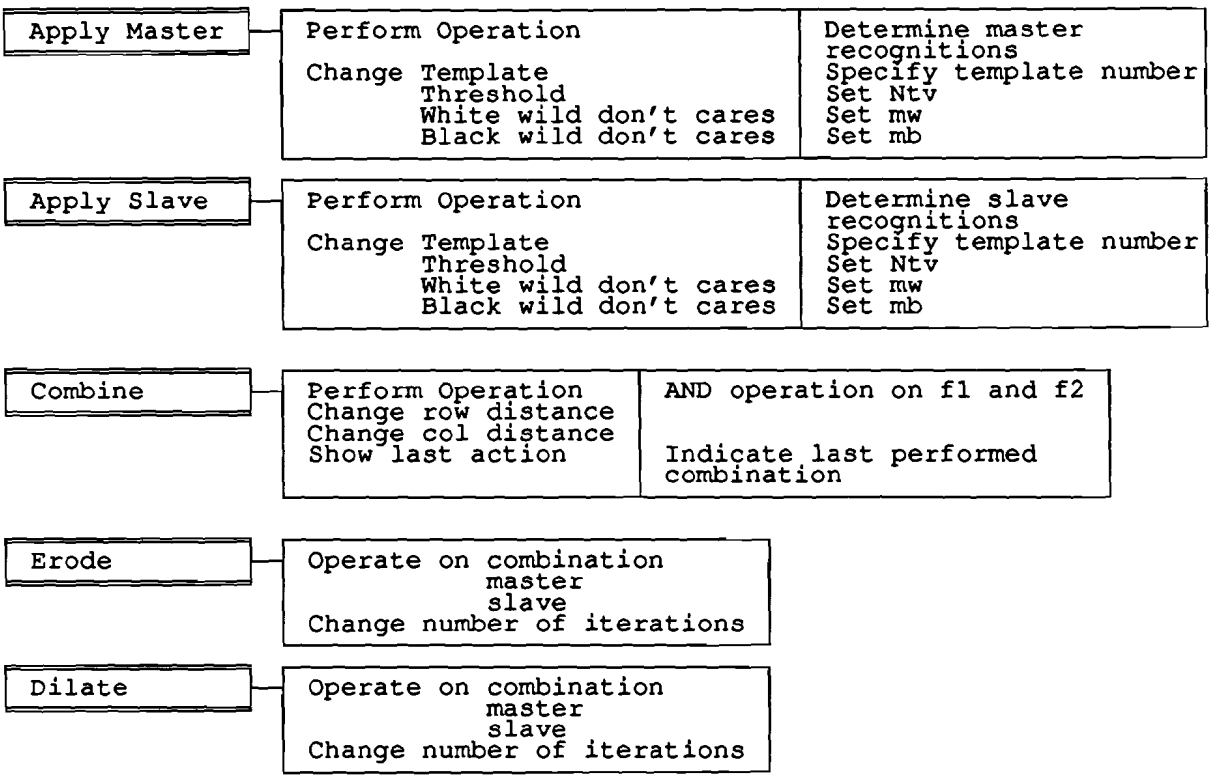
Main menu items:

1. Acquisition
2. Dual DRC
3. DRC

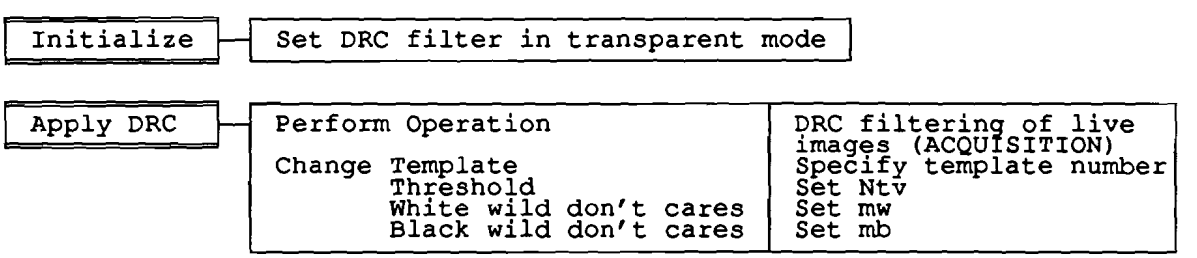
1. Acquisition

Display 0..255		Toggles between displaying the upper or lower half of the screen
View 0..511	Display	displays entire screen
	Magnify SBIP0	magnifies a part of the frame buffer
	Magnify SBIP1	similar, but now applied on SBIP1
Live Image		Passes live images from input 3 of SBIP0 directly via the DRC filter
Freeze Image		Acquires image from input 0 from SBIP1 and freezes it
Retrieve Image		Fill SBIP1's frame buffer with a stored image
Save Image		Saves the frame buffer's contents of SBIP0 or SBIP1
Mark Matchings		Indicates the recognitions points in the input image (master, slave, combined, final point)

2. Dual DRC



3. DRC



nr.0
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx

nr.1
xxxxxxxx
x111111x
x1000000
x10xxxxx
x10xxxxx
x10xxxxx
x10xxxxx
xx0xxxxx

nr.2
xx0xxxxx
x10xxxxx
x10xxxxx
x10xxxxx
x10xxxxx
x1000000
x111111x
xxxxxxxx

nr.3
xxxxx0xx
xxxxx01x
xxxxx01x
xxxxx01x
xxxxx01x
0000001x
x111111x
xxxxxxxx

nr.4
xxxxxxxx
x111111x
0000001x
xxxxx01x
xxxxx01x
xxxxx01x
xxxxx01x
xxxxx0xx

nr.5
xxxxxxxx
x000000x
x0111111
x01xxxxx
x01xxxxx
x01xxxxx
x01xxxxx
xx1xxxxx

nr.6
xx1xxxxx
x01xxxxx
x01xxxxx
x01xxxxx
x01xxxxx
x0111111
x000000x
xxxxxxxx

nr.7
xxxxx1xx
xxxxx10x
xxxxx10x
xxxxx10x
xxxxx10x
1111110x
x000000x
xxxxxxxx

nr.8
xxxxxxxx
x000000x
1111110x
xxxxx10x
xxxxx10x
xxxxx10x
xxxxx10x
xxxxx1xx

nr.9
xxx10xxx
xxx10xxx
xxx10xxx
xxx10xxx
xxx10xxx
xxx10xxx
xxx10xxx
xxx10xxx

nr.10
xxx01xxx
xxx01xxx
xxx01xxx
xxx01xxx
xxx01xxx
xxx01xxx
xxx01xxx
xxx01xxx

nr.11
xxxxxxxx
xxxxxxxx
xxxxxxxx
11111111
00000000
xxxxxxxx
xxxxxxxx
xxxxxxxx

nr.12
xxxxxxxx
xxxxxxxx
xxxxxxxx
00000000
11111111
xxxxxxxx
xxxxxxxx
xxxxxxxx

nr.13
xxx1x0xx
xx1x0xxx
xx1x0xxx
xx1x0xxx
xx1x0xxx
xx1x0xxx
xx1x0xxx
xxx1x0xx

nr.14
xx0x1xxx
xxx0x1xx
xxx0x1xx
xxx0x1xx
xxx0x1xx
xxx0x1xx
xxx0x1xx
xx0x1xxx

nr.15
xxx0x1xx
xx0x1xxx
xx0x1xxx
xx0x1xxx
xx0x1xxx
xx0x1xxx
xx0x1xxx
xxx0x1xx

nr.16
xx1x0xxx
xxx1x0xx
xxx1x0xx
xxx1x0xx
xxx1x0xx
xxx1x0xx
xxx1x0xx
xx1x0xxx

nr.17
xxxxxxxx
xxxxxxxx
0xxxxxx0
x000000x
1xxxxxx1
x111111x
xxxxxxxx
xxxxxxxx

nr.18
xxxxxxxx
xxxxxxxx
x111111x
1xxxxxx1
x000000x
0xxxxxx0
xxxxxxxx
xxxxxxxx

nr.19
xxxxxxxx
xxxxxxxx
1xxxxxx1
x111111x
0xxxxxx0
x000000x
xxxxxxxx
xxxxxxxx

nr.20
xxxxxxxx
xxxxxxxx
x000000x
0xxxxxx0
x111111x
1xxxxxx1
xxxxxxxx
xxxxxxxx

nr.21
xxxxxxxx
xx00000x
xx01110x
xx01110x
xx01110x
xx01110x
xx01110x
xx01110x

nr.22
xxxxxxxx
x000000x
x011110x
x011110x
x011110x
x011110x
x011110x
x0xxxx0x

nr.23
xxxxxxxx
xxxxxxxx
x0000000
x0111110
x01xxx10
x01xxx10
x01xxx10
x0111110

nr.24
xxxxxxxx
xxxxxxxx
00000000
01111110
01111110
011xx110
011xx110
011xx110

nr.25
xxxxxxxx
xx11111x
xx10001x
xx10001x
xx10001x
xx10001x
xx10001x
xx10001x

nr.26
xxxxxxxx
x111111x
x100001x
x100001x
x100001x
x100001x
x100001x
x1xxxx1x

nr.27
xxxxxxxx
xxxxxxxx
x1111111
x1000001
x10xxx01
x10xxx01
x10xxx01
x1000001

nr.28
xxxxxxxx
xxxxxxxx
11111111
10000001
10000001
100xx001
100xx001
100xx001

nr.29
xx01110x
xx01110x
xx01110x
xx01110x
xx01110x
xx01110x
xx00000x
xxxxxxxx

nr.30
x0xxxx0x
x011110x
x011110x
x011110x
x011110x
x011110x
x000000x
xxxxxxxx

nr.31
x0111110
x01xxx10
x01xxx10
x01xxx10
x0111110
x0000000
xxxxxxxx
xxxxxxxx

nr.32
011xx110
011xx110
011xx110
011xx110
01111110
00000000
xxxxxxxx
xxxxxxxx

nr.33
xx10001x
xx10001x
xx10001x
xx10001x
xx10001x
xx10001x
xx11111x
xxxxxxxx

nr.34
x1xxxx1x
x100001x
x100001x
x100001x
x100001x
x100001x
x111111x
xxxxxxxx

nr.35
x1000001
x10xxx01
x10xxx01
x10xxx01
x1000001
x1111111
xxxxxxxx
xxxxxxxx

nr.36
100xx001
100xx001
100xx001
100xx001
10000001
11111111
xxxxxxxx
xxxxxxxx

nr.37
xxxxxxxx
xxxxxxxx
0000000x
1111110x
1111110x
1111110x
0000000x
xxxxxxxx

nr.38
xxxxxxxx
0000000x
x111110x
x111110x
x111110x
x111110x
0000000x
xxxxxxxx

nr.39
xxxxxxxx
000000xx
111110xx
1xxx10xx
1xxx10xx
1xxx10xx
111110xx
000000xx

nr.40
000000xx
111111xx
111110xx
xxxx10xx
xxxx10xx
111110xx
111110xx
000000xx

nr.41
xxxxxxxx
xxxxxxxx
1111111x
0000001x
0000001x
0000001x
1111111x
xxxxxxxx

nr.42
xxxxxxxx
1111111x
x000001x
x000001x
x000001x
x000001x
1111111x
xxxxxxxx

nr.43
xxxxxxxx
111111xx
000001xx
0xxx01xx
0xxx01xx
0xxx01xx
000001xx
111111xx

nr.44
111111xx
000001xx
000001xx
xxxx01xx
xxxx01xx
000001xx
000001xx
111111xx

nr.45
xxxxxxxx
xxxxxxxx
x0000000
x0111111
x0111111
x0111111
x0000000
xxxxxxxx

nr.46
xxxxxxxx
x0000000
x011111x
x011111x
x011111x
x011111x
x0000000
xxxxxxxx

nr.47
xxxxxxxx
xx000000
xx011111
xx01xxx1
xx01xxx1
xx01xxx1
xx011111
xx000000

nr.48
xx000000
xx011111
xx011111
xx01xxxx
xx01xxxx
xx011111
xx011111
xx000000

nr.49
xxxxxxxx
xxxxxxxx
x1111111
x1000000
x1000000
x1000000
x1111111
xxxxxxxx

nr.50
xxxxxxxx
x1111111
x100000x
x100000x
x100000x
x100000x
x1111111
xxxxxxxx

nr.51
xxxxxxxx
xx111111
xx100000
xx10xxx0
xx10xxx0
xx10xxx0
xx100000
xx111111

nr.52
xx111111
xx100000
xx100000
xx10xxxx
xx10xxxx
xx100000
xx100000
xx111111

nr.53
xx01110x
xx01110x
x0011100
x0011100
x0011100
x0011100
xx01110x
xx01110x

nr.54
00111100
00111100
00111100
00111100
00111100
00111100
00111100
00111100

nr.55
0011xx10
0011xx10
0011xx10
0011xx10
0011xx10
0011xx10
0011xx10
0011xx10

nr.56
01xxxx10
01xxxx10
01xxxx10
01xxxx10
01xxxx10
01xxxx10
01xxxx10
01xxxx10

nr.57
xx10001x
xx10001x
x1100011
x1100011
x1100011
x1100011
xx10001x
xx10001x

nr.58
11000011
11000011
11000011
11000011
11000011
11000011
11000011
11000011

nr.59
1100xx01
1100xx01
1100xx01
1100xx01
1100xx01
1100xx01
1100xx01
1100xx01

nr.60
10xxxx01
10xxxx01
10xxxx01
10xxxx01
10xxxx01
10xxxx01
10xxxx01
10xxxx01

nr.61
xxxxxxxx
xx0000xx
00000000
11111111
11111111
11111111
00000000
xx0000xx

nr.62
00000000
00000000
11111111
11111111
11111111
11111111
00000000
00000000

nr.63
00000000
00000000
11111111
11111111
xxxxxxxx
xxxxxxxx
11111111
00000000

nr.64
00000000
11111111
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
11111111
00000000

nr.65
xxxxxxxx
xx1111xx
11111111
00000000
00000000
00000000
11111111
xx1111xx

nr.66
11111111
11111111
00000000
00000000
00000000
00000000
11111111
11111111

nr.67
11111111
11111111
00000000
00000000
xxxxxxxx
xxxxxxxx
00000000
11111111

nr.68
11111111
00000000
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
00000000
11111111

nr.69
xxxxxxxx
x0000000
x0xxxxxx
x0x11111
x0x1xxxx
x0x1xxxx
x0x1xxxx
x0x1xxxx

nr.70
xxxxxxxx
0000000x
xxxxxx0x
11111x0x
xxxx1x0x
xxxx1x0x
xxxx1x0x
xxxx1x0x

nr.71
xxxx1x0x
xxxx1x0x
xxxx1x0x
xxxx1x0x
11111x0x
xxxxxx0x
0000000x
xxxxxxxx

nr.72
x0x1xxxx
x0x1xxxx
x0x1xxxx
x0x1xxxx
x0x11111
x0xxxxxx
x0000000
xxxxxxxx

nr.73
xxxxxxxx
x1111111
x1xxxxxx
x1x00000
x1x0xxxx
x1x0xxxx
x1x0xxxx
x1x0xxxx

nr.74
xxxxxxxx
1111111x
xxxxxx1x
00000x1x
xxxx0x1x
xxxx0x1x
xxxx0x1x
xxxx0x1x

nr.75
xxxx0x1x
xxxx0x1x
xxxx0x1x
xxxx0x1x
00000x1x
xxxxxx1x
1111111x
xxxxxxxx

nr.76
x1x0xxxx
x1x0xxxx
x1x0xxxx
x1x0xxxx
x1x00000
x1xxxxxx
x1111111
xxxxxxxx

nr.77
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx

nr.78
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx

nr.79
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx
xxxxxxxx

9.3 EVALUATE

The procedure evaluates the performance of the DRC template matching by comparing its appliance on ideal and real world situations

First simulation parameters have to be specified, then an ideal test image with corresponding reference recognition points have to be derived. The test image is then deteriorated and performed is DRC template matching is being applied. Recognitions are being evaluated yielding estimation for deviations in row and column direction and size of recognition blobs (mean and standard deviations).

The simulation parameters to be specified are:

- Template number ideal template:
  Specified has to be the template number to be applied on the ideal test image.

  Templates have been declared in program text as an array, and template shapes have been restricted to corners only. Templates' shapes can be changed by altering program text.

nr. 8	*nr. 9*	*nr. 10*	*nr. 11*
00000000	00000000	11111100	00111111
00000000	00000000	11111100	00111111
00111111	11111100	11111100	00111111
00111111	11111100	11111100	00111111
00111111	11111100	11111100	00111111
00111111	11111100	11111100	00111111
00111111	11111100	00000000	00000000
00111111	11111100	00000000	00000000

nr. 12	*nr. 13*	*nr. 14*	*nr. 15*
00000000	00000000	11111000	00011111
00000000	00000000	11111000	00011111
00000000	00000000	11111000	00011111
00011111	11111000	11111000	00011111
00011111	11111000	11111000	00011111
00011111	11111000	00000000	00000000
00011111	11111000	00000000	00000000
00011111	11111000	00000000	00000000

- Template number real world image:
  Specified has to be the template number to be applied on the deteriorated test image.

nr. 0	*nr. 1*	*nr. 2*	*nr. 3*
00000000	00000000	11111x00	00x11111
00000000	00000000	11111x00	00x11111
00xxxxxx	xxxxxx00	11111x00	00x11111
00x11111	11111x00	11111x00	00x11111
00x11111	11111x00	11111x00	00x11111
00x11111	11111x00	xxxxxx00	00xxxxxx
00x11111	11111x00	00000000	00000000
00x11111	11111x00	00000000	00000000

nr. 4	*nr. 5*	*nr. 6*	*nr. 7*

```
00000000    00000000    1111xx00    00xx1111
00000000    00000000    1111xx00    00xx1111
00xxxxxx    xxxxxx00    1111xx00    00xx1111
00xxxxxx    xxxxxx00    1111xx00    00xx1111
00xx1111    1111xx00    xxxxxx00    00xxxxxx
00xx1111    1111xx00    xxxxxx00    00xxxxxx
00xx1111    1111xx00    00000000    00000000
00xx1111    1111xx00    00000000    00000000
```

- μ_1 mean background level (0..150)
- μ_2 mean foreground level (0..150)
- σ_h S.D. background and foreground level (e.g. 10, 20)
- σ_p S.D. of Gaussian low pass filter emulating resolution (e.g. 0.5, 1.0)
- σ_n S.D. of additional Gaussian noise. Various noise levels (5,10,25) have been generated and stored in data files. Generation can be performed by means of program NOIS_IMG.
- adding speckles, height h_{sp}. Speckles are being added according a pattern stored in the image file generated by SPCK_IMG. The density d_{sp} in which speckles occur is specified within this program.
- removing speckles. Specified has to be the threshold to be applied for the edge-preserving noise removing filter.
- file where to store evaluation results in.

Next step is ideal test image generation which serves as reference image.

- First, and ideal test image is being generated.
- DRC template matching applying the template for the ideal test image is being applied, yielding reference points.

Next step is to deteriorate the reference image in order to let it mimic real world situations.

- define background levels
- deform boundaries of background regions in order to avoid recognition
- blur background in order to avoid recognition
- draw test patterns
- blur edges, emulation resolution properties
- add Gaussian noise.

Next step is to acquire the recognitions of DRC template matching applied on the real world situation:

- specify whether blob to centre of gravity point reduction has to be performed.
- specify the number of white wild don't cares allowed
- specify the number of black wild don't cares allowed
- specify the range threshold

Finally

- apply the DRC template for real world situations
- acquire the recognitions
- compute statistics

In order to perform simulations, several auxiliary files have to be generated first.

- The file containing various Gaussian PDF's, PDF.DTA, is generated by invoking CALC_PDF.C.
- The file containing locations where to add speckles, SPECKLES.IMG, is generated by SPCK_IMG. The speckles' density is comprised within this program
- Files containing the grey values to be added to or to be subtracted from the original image in order to add gaussian noise, NOIS??A.IMG resp. NOISE??S.IMG, are generated by GAUS_IMG.

In order to perform simulations, several TMS routines are invoked. These routines are presented in a separate subdirectory.

9.4 ANAALIGN

ANAALIGN evaluates the alignment measurement unit's performance. Misalignment in one direction which serves as reference is being introduced by means of a tilted Bezier surface. DRC evaluation showed that DRC template matching is liable to a certain measuring inaccuracy. This can be accounted for, and the alignment algorithm with various parameter settings can be performed in order analyze their effects. When invoking ANAALIGN, the following menu structure is being presented.

Main menu items:

0. Quit
1. Create misalignment due to product deformation (3D Bezier surface)
2. Monitor misalignment
3. Perform alignment
4. Monitor alignment

1. Create misalignment due to product deformation (3D Bezier surface)

This routine is invoked in order to generate a 3D Bezier surface and which mimics real world misalignment due to product deformation in one direction.

In order to do so we first have to specify

- the target file
- the number of alignment regions (in one direction)
- the number of control points (in one direction)

Then the first row of control points has to be specified. This can be done by means of the cursor keys. Press enter when the desired location for a control point has been reached. Note that the first and last control point of a row of control points must be at the left most and right most position of the grid.

The control points of the successing rows are specified by means of the following keys.

ArrUp, ArrDn move current selected control point (in vertical direction)

PgUp, PgDn change control point selected

Ctrl-PgUp, Ctrl-PgDn change row

Pressing [Enter] starts the Bezier surface computation and storage on disk.

2. Monitor Bezier surface

Specified has to be the name of the source file on disk storing a 3D Bezier surface. This surface can now be displayed by 3D representation or 2D representation of its cross section(s). The magnitude of rotation and global offset can also be taken into consideration.

The sub-menu presented is as follows:

0. Quit
1. 3D display
2. horizontal cross section (2D representation)
3. vertical cross section (2D representation)
4. set rotation (horizontal misalignment, vertical misalignment, global offset).

3. Perform Alignment

Before actually performing the alignment and evaluation, the following properties have to be specified first.

First a reference misalignment in one direction has to be specified:

- enter rotation (horizontal misalignment, vertical misalignment, global offset)
- cushion effect filename

Next step is to account for measuring inaccuracy introduced by DRC template matching.

- distortion file, emulating measuring inaccuracy.

Then various alignment parameters are to be specified

- number of interpolations
- delta (tolerance margin δ, expressed in pixels)
- tau (initial life-time τ, expressed in update regions)
- eta (maximal validation vector distance η, expressed in pixels)

Distribution of alignment features is specified by means of a so called measuring scenario file which is an ASCII file with NxM' elements. N is the number of update regions in vertical direction and M' is the number of alignment regions in horizontal direction.

The elements comprised in the measuring scenario file have the following meaning:

I the misalignment vector prevailing in the update region (at the beginning of the alignment region) is booked into VMem, SMem, and LMem;

P the misalignment vector prevailing the in the corresponding update region (at the beginning of the alignment region) in the row above is booked into VMem, SMem, and LMem;

X no booking is being performed.

Last sub-routine is the evaluation of the performed correction for misalignment by means of statistical properties which are stored in a target file which name is to be specified, as are the update regions involved.

Also generated are

CORRECT.TMP (NxM)
LMEM.TMP (dimension NxM')
VMEM.TMP (dimension NxM')
SMEM.TMP (dimension NxM')

These files presents the computed corrections and contents of the memories involved. The location of the presented elements within the files corresponds to the progression of scan.

The elements comprised in SMEM.TMP have the following meaning:

V valid, the misalignment vector contained in VMem is valid;

E elapsed, the misalignment vector has elapsed;

I initialization, a misalignment vector is being booked which prevails at the location of scan;

P previous, a misalignment vector is being booked which prevailed in an update region contained in a previous horizontal stroke of update regions;

F the misalignment vector to be booked exceeds the validation margin η.

9.5 AMU

AMU demonstrates the alignment measurement unit. DRC template matching is being performed on real world images acquired by means of a CCD camera attached to video input 1 of SBIP1. Upper left-hand corners of black objects on white background are being considered to be alignment features. Operations can be invoked by the following menu structure.

Main menu items:

0. Quit
1. Edit number of samples
2. Edit number of interpolations
3. Acquire golden plate
4. Acquire moved object
5. Display alignment
6. Edit tolerance margin
7. Edit template nr.

1. Edit number of samples

Asked is to specify the number of alignment regions (in both row as column direction), and the number update regions along the side of an alignment region. The numbers should preferably be a power of two.

2. Edit number of interpolations

Asked is to specify the number of update regions along the side of an alignment region.

3. Acquire golden plate

The object currently under camera, is considered to be the golden plate. The start position of the grid of alignment and update regions has first to be specified by means of the cursor keys and enter key. By pressing the space bar, the step size can be altered. Once an offset has been specified, the object can be positioned correctly by moving it. Recognitions to be found can be displayed by toggling the space bar. The golden plate is frozen by pressing the enter key. The actions are being displayed on SBIP0's monitor.

4. Acquire moved object

Misalignment can be wantonly introduced by moving the object under camera. The maximum misalignment set for the demonstration is 16 pixels in all directions. This area is being indicated by windows and emulates the area of interest. Reference points have to be within these windows. By pressing the space bar, SBIP0's monitor output is toggled to be the scene under camera or its resulting recognitions.

5. Display alignment

Invoking this operations present the following sub-menu:

0. Quit
1. Alignment regions along scanline (row direction)
2. Alignment regions perpendicular to scanline (column direction)
3. Update regions along scanline (row direction)
4. Update regions perpendicular to scanline (column direction)
5. Graphics display

ad 1. Displayed on SBIP1's monitor is now the coloured representation of the measured misalignment in row direction. By means of cursor keys, a window can be moved over the image, comprising several alignment regions. Simultaneously the misalignment values of these alignment regions are being displayed on the computer monitor. Press [esc] in order to return.

ad 2. Similar to 1. but now measured misalignment in column direction is being considered.

ad 3. Similar to 1. but now computed correction in row direction is being considered.

ad 4. Similar to 1. but now computed correction in column direction is being considered.

ad 5. Displayed is a horizontal or vertical stroke of update regions. The direction can be toggled by pressing the space bar. Point of pivoting of the stroke of update regions is indicated as a solid update region. Which update region is to become solid can be determined by means of the cursor keys.

The following five options are now available. By choosing options 1,2,3 or 4, SBIP1's monitor displays the coloured representation of

1. Measured misalignment in row direction
2. Measured misalignment in column direction
3. Correction of misalignment in row direction
4. Correction of misalignment in col direction

Option 5 yields a window which can be moved over the image and comprise alignment regions or update regions. It presents the values of the regions currently being displayed.

6. Edit tolerance margin

Set the tolerance margin δ.

7. Edit template nr.

Set the number of the template to be applied. The list of templates applied is the same as the one given for DUAL_DRC.

Related documents

[1] Vision and Information Processing for Automation
A. Browne and L.N. Norton-Wayne
Plenum Press, New York and London 1986

[2] Automated Visual Inspection
B.G. Batchelor, D.A. Hill and D.C. Hodgson
IFS (Publications) Ltd. UK, 1985

[3] ESPRIT 2017, Automated Process and Assembly Inspection by 3D Vision
Technical report TR3, 1990

[4] SAD specification of a 3D inspection system for fine line PCBs
M.F.W. Pechler, Philips CFT, 1991

[5] Dynamic Thresholding of Grey-level images
ir. J.A.C. Bernsen, Philips Nat.Lab.
Nat.Lab. report no. 6144

[6] Multi-level Input Binary Template Matching,
Theory and real-time implementation using rank-value filter
ir. J.A.C. Bernsen, Philips Nat.Lab.
Nat.Lab. technical note nr. 075/89

[7] L64220 Rank-Value Filter (RVF)
LSI Logic (Acrobel, Oss, The Netherlands), 1987.

[8] A pipelined MAX-MIN filter image processor
doc.no. 88i06.n.1
S3 Silicon and Software Systems, 28.09.90

[9] Single Board Image Processor SBIP
Philips Centre for Manufacturing Technology
Report 17/89EN

[10] The SBIP DRC-module User Guide
E.F.J. Claessens, Philips CFT, september 1989

[11] Templates for alignment and inspection
Olof Ekman, Philips CFT, 1990

[12] Evaluation of Edge Detection Algorithms
IEEE Conference proceedings on Image Processing and its Applications 1989
F. van der Heyden, University of Twente, The Netherlands

[13] Implementation of Cellular-logic Operators using 3x3 Convolutions and Table Lookup Hardware
F.A. Gerritsen (NLR, The Netherlands), P.W. Verbeek (TUD), 1983
Publications 1984 TH Delft, Department of Applied Physics, Pattern Recognition Group

[14] Some Parallel Thinning Algorithms for Digital Pictures
R. Stefanelli and A. Rosenfeld
Journal of the Association for Computing Machinery Inc., Vol. 18, No. 2, April 1971, pp. 255-264

[15] Computer Graphics, College Syllabus 2K900
Eindhoven University of Technology 1988.

[16] Computer Graphics
D. Hearn and M.P. Baker
Prentice Hall International, 1986.