

# VISUAL ALIGNMENT ROBOT SYSTEM: KINEMATICS, PATTERN RECOGNITION, AND CONTROL

SangJoo Kwon and Chansik Park

*School of Aerospace and Mechanical Engineering, Korea Aerospace University, Goyang-city, 412-791, Korea*

**Keywords:** Visual alignment, robotics, parallel mechanism, precision control, pattern recognition.

**Abstract:** The visual alignment robot system for display and semiconductor fabrication process largely consists of multi-axes precision stage and vision peripherals. One of the central issues in a display or semiconductor mass production line is how to reduce the overall tact time by making a progress in the alignment technology between the mask and panel. In this paper, we suggest the kinematics of the 4PPR parallel alignment mechanism with four limbs unlike usual three limb cases and an effective pattern recognition algorithm for alignment mark recognition. The inverse kinematic solution determines the moving distances of joint actuators for an identified mask-panel misalignment. Also, the proposed alignment mark detection method enables considerable reduction in computation time compared with well-known pattern matching algorithms.

## 1 INTRODUCTION

In the flat panel display and semiconductor industry, the alignment process between mask and panel is considered as a core technology which determines the quality of products and the productivity of a manufacturing line. As the sizes of panel and mask in the next generation products increases but the dot pitch becomes smaller, the alignment systems must fulfill more strict requirements in load capacity and control precision. The alignment system largely has two subsystems. One is the multi-axes robotic stage to move the mask in a desired location with acceptable alignment errors and the other one is the vision system to recognize the alignment marks printed in mask and panel surfaces. In a display or semiconductor production line, the alignment systems are laid out in series as subsystems of pre-processing and post-processing equipments such as evaporation, lithography, and joining processes.

The alignment stage has at least three active joints to determine planar three degrees of freedom alignment motions. It usually adopts a parallel mechanism, specifically when it is used for display panel alignment, since it has the advantage of high stiffness and high load capacity. In this paper, we are to discuss the inverse kinematics of the parallel stage which has four prismatic-prismatic-revolute (4PPR) limbs where all the base revolute joints are active ones. For the same-sized moving platform, the four-limb mechanism brings higher stiffness and load capacity com-

paring with normal three-limb stages but inevitably it leads to a more difficult control problem. Although a commercial alignment stage with four driving axes was announced in (Hephaist Ltd., 2004), reports on the kinematics and control can be rarely found.

The next issue is the vision algorithm to extract the position of alignment marks. In many machine vision systems, the normalized gray-scale correlation (NGC) method (Manickam et al., 2000) has been used as a representative template matching algorithm. However, it requires long computation time since all pixels in the template image are compared in the matching process. An alternative to reduce the computation time is the point correlation (PC) algorithm (Krattenthaler et al., 1994) but it is still weak to the rotation of object and the change of illumination condition. As another, the edge-based point correlation (Kang and Lho, 2003) was proposed to mitigate the effect of illumination change. In fact, commercial vision libraries, e.g., (Cognex Ltd., 2004), are adopted in many visual alignment systems considering the stability of the vision algorithm. However, they are computationally inefficient from the point of view that they have overspec for the monotonous vision environment of alignment systems (e.g., simple mark shape, fine light, and uniform background). In this paper, by incorporating the binarization and labeling algorithm (Gonzalez and Wood, 2002) together and designing a geometric template matching scheme instead of the general methods: NGC (Manickam et al., 2000) and PC (Krattenthaler et al., 1994), an efficient

pattern recognition algorithm for alignment marks is suggested, which can greatly reduce vision processing time comparing with commercial products.

Related to the autonomous alignment system, several articles can be found. A two step alignment algorithm was suggested for wafer dicing process (Kim et al., 2004) and a self-alignment method for wafers was investigated using capillary forces (Martin, 2001). A visionless alignment method was also studied (Kanjilal, 1995). As a trial to improve the alignment speed, an additional sensor was integrated (Umminger and Sodini, 1995) and a modified template matching algorithm was presented (Lai and Fang, 2002).

## 2 VISUAL ALIGNMENT SYSTEM

### 2.1 System Configuration

Figure 1 shows the schematic of a PC-based experimental setup for the control of visual alignment system. Broadly speaking, it consists of the vision system to detect alignment marks on mask and panel and the stage control system to compensate misalignments. The vision system has normally two CCD cameras and optical lenses, illumination equipment, and frame grabber board to capture mark images. In real production lines, prior to the visual alignment, the pre-alignment process puts the marks of mask and panel into the field-of-view of CCD Cameras. Then, the vision system determines the mark positions in the global frame and the misalignment distance in camera coordinate can be converted into the moving distances of joint actuators through the inverse kinematic solution of alignment stage.

As denoted in Fig 2, the feedback control system in the alignment stage has the hierarchical feedback loops, where the outer visual servoing loop determines the misalignment quantity between mask and panel and the inner joint control loop actively compensates it. Due to system uncertainties such as friction and backlash, the alignment process is usually not completed by one visual feedback but a few cycles are repeated.

### 2.2 Parallel Alignment Stage

As the size of flat panel displays including TFT/LCD, PDP, and OLED becomes larger and larger, the alignment stage is required to have higher load capacity and wider moving platform. In this regard, the motion control performance of alignment stage is directly related to the productivity of the manufacturing process.

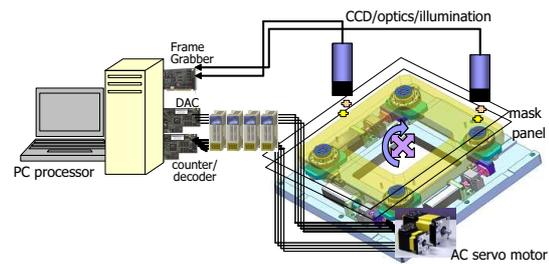


Figure 1: Schematic of visual alignment control system.

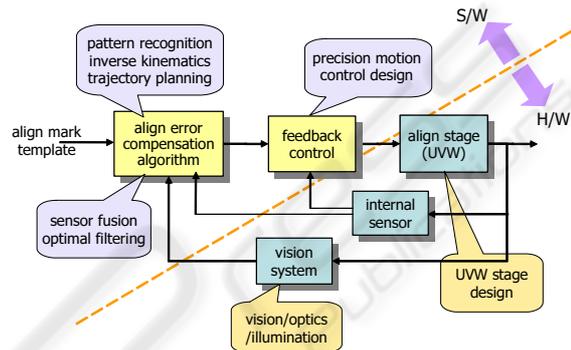


Figure 2: S/W and H/W components of visual alignment system.

The advantages of parallel manipulation mechanisms comparing with serial ones are i) high payload-to-weight ratio, ii) high structural rigidity since the payload is carried by several limbs, iii) high manipulation accuracy since active joints are distributed in parallel and the joint errors are non-cumulative, and iv) simple inverse kinematic solution. But, they suffer from smaller workspace and singular configurations. Although the movable range of the parallel stage is very small (usually a few *mms*), it is actually enough to compensate misalignments between mask and panel.

To position the moving platform (in which the mask is mounted) in a planar location, the task space of alignment stage must have at least three degrees of freedom and then it requires at least three active joints. Hence, it is common for the parallel alignment stage to have three active limbs. However, if an extra driving limb is added to support the moving platform, the stiffness and load capacity can be much increased. The visual alignment testbed with four driving axes is shown in Fig. 3, where the motion of the rotary motor in each limb is transferred to the moving platform through a ballscrew, cross-roller guide (for planar translational motion) and cross-roller ring (for rotational motion).

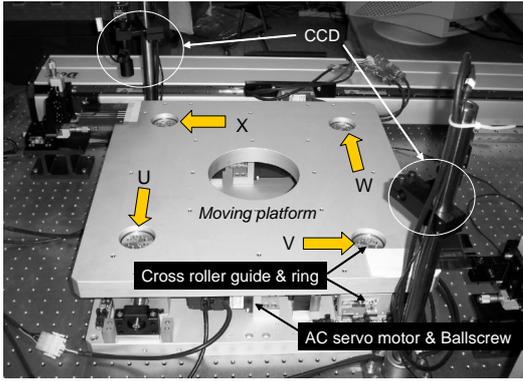


Figure 3: Visual alignment stage with four active joints.

### 3 KINEMATICS OF 4PPR PARALLEL MECHANISM

#### 3.1 Determination of Misalignments

When the centroids of alignment marks of panel and mask have been obtained through the image processing, it is trivial to get the misaligned posture between panel and mask. In Fig. 4, let the two align marks in the panel from the respective CCD image have the centroids of  $C_1 = (x_{C_1}, y_{C_1})$  and  $C_2 = (x_{C_2}, y_{C_2})$  and those in the mask have  $L_1 = (x_{L_1}, y_{L_1})$  and  $L_2 = (x_{L_2}, y_{L_2})$  in the global frame. Then, the center of line connecting two centroids can be written as

$$p_x = (x_{C_1} + x_{C_2})/2, \quad p_y = (y_{C_1} + y_{C_2})/2 \quad (1)$$

for the panel marks and also

$$m_x = (x_{L_1} + x_{L_2})/2, \quad m_y = (y_{L_1} + y_{L_2})/2 \quad (2)$$

for the mask ones. If the mask is to be aligned to the panel, the misaligned distance between mask and panel is given by

$$\Delta x = p_x - m_x = \frac{x_{C_1} - x_{L_1} + x_{C_2} - x_{L_2}}{2} \quad (3)$$

$$\Delta y = p_y - m_y = \frac{y_{C_1} - y_{L_1} + y_{C_2} - y_{L_2}}{2} \quad (4)$$

$$\Delta \phi = \tan^{-1} \left[ \frac{y_{C_2} - y_{C_1}}{x_{C_2} - x_{C_1}} \right] - \tan^{-1} \left[ \frac{y_{L_2} - y_{L_1}}{x_{L_2} - x_{L_1}} \right] \quad (5)$$

in  $(x, y)$  directions and orientation, respectively.

#### 3.2 Mobility of 4ppr Alignment Stage

The degrees of freedom of a mechanism can be found by the Grubler criterion (Tsai, 1999):

$$F = \lambda(n - j - 1) + \sum_i f_i - f_p, \quad (6)$$

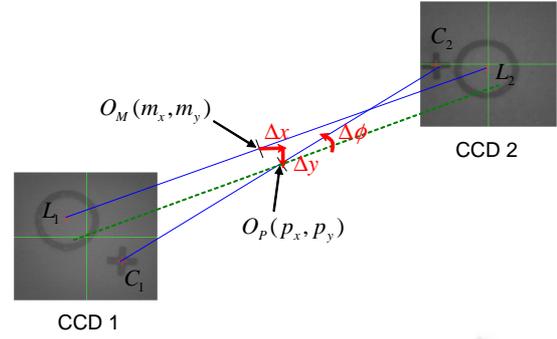


Figure 4: Determination of misaligned posture between mask and panel (circle: mask marks, cross: panel marks).

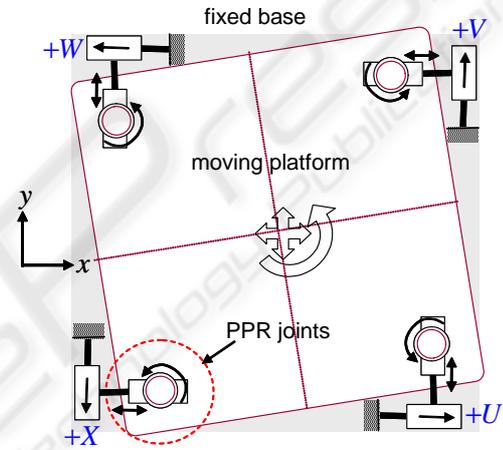


Figure 5: Planar 3-DOF, 4PPR parallel mechanism.

where  $\lambda$  is the dimension of the space in which the mechanism is intended to function,  $n$  the number of links,  $j$  the number of joints,  $f_i$  degrees of relative motion permitted by joint  $i$ , and  $f_p$  the passive degrees of freedom. In the case of a planar 4PPR parallel mechanism shown in Fig. 5, in which the moving platform is supported by four limbs with prismatic-prismatic-revolute joints, we have  $\lambda = 3$ ,  $n = 2 \times 4 + 1 + 1$  including fixed base and moving platform,  $j = 4 \times 3$ ,  $\sum f_i = 12 \times 1$  and  $f_p = 0$ . Hence, the degrees of freedom of the 4PPR mechanism is  $F = 3$  as is expected.

In parallel manipulators, every limb forms closed-loop chain and the number of active limbs is typically equal to the number of degrees of freedom of the moving platform. Moreover, each limb has the constraint of having more joints than the degrees of freedom. Hence, at least three limbs must have an active joint (the first P in this case) to achieve the 3-DOF motion of moving platform and the remaining limb becomes passive. On the other hand, if all the four limbs are actuated to increase the rigidity of motion, the actuation redundancy problem is present and a sophisti-

cated control logic is required to avoid mechanical singularity.

### 3.3 Inverse Kinematics

The inverse kinematic problem is to seek the moving ranges of input joints which correspond to the end-effector movement of a mechanism, i.e., moving platform in this case. In Fig. 6, the square of a fixed base is defined by the four fixed points  $(P, Q, R, S)$  and the moving platform is connected to the limbs at the four points  $(A, B, C, D)$ . We assume that the two squares have the same side length of  $h$  and the global coordinates system is located at the center  $(O_1)$  of the fixed base.

Then, the positions of fixed points  $(P, Q, R, S)$  are given by  $(x_P, y_P) = (-\frac{1}{2}h, -\frac{1}{2}h)$ ,  $(x_Q, y_Q) = (\frac{1}{2}h, -\frac{1}{2}h)$ ,  $(x_R, y_R) = (\frac{1}{2}h, \frac{1}{2}h)$ , and  $(x_S, y_S) = (-\frac{1}{2}h, \frac{1}{2}h)$ . Moreover, assuming that the position of point  $A$  in the moving platform is known, the positions of the other three connecting points can be expressed as

$$(x_B, y_B) = (x_A + h \cos \Delta\phi, y_A + h \sin \Delta\phi) \quad (7)$$

$$(x_C, y_C) = (x_B - h \sin \Delta\phi, y_B + h \cos \Delta\phi) \quad (8)$$

$$(x_D, y_D) = (x_A - h \sin \Delta\phi, y_A + h \cos \Delta\phi) \quad (9)$$

In Fig. 6, we have  $(x_A, y_A) = \overline{O_1A} = \overline{O_1O_2} + \overline{O_2A}$  where  $\overline{O_1O_2} = (\Delta x, \Delta y)$  and

$$\overline{O_2A} = R(\Delta\phi)\overline{O_1P} = \begin{bmatrix} \cos \Delta\phi & -\sin \Delta\phi \\ \sin \Delta\phi & \cos \Delta\phi \end{bmatrix} \begin{bmatrix} -\frac{1}{2}h \\ -\frac{1}{2}h \end{bmatrix} \quad (10)$$

Hence, the position of point  $A$  can be written by the misalignment variables as

$$x_A = \Delta x - h(\cos \Delta\phi - \sin \Delta\phi)/2 \quad (11)$$

$$y_A = \Delta y - h(\sin \Delta\phi + \cos \Delta\phi)/2 \quad (12)$$

As denoted in Fig. 6, the moving distances of input prismatic joints have the following relationships (note the positive directions defined in Fig. 5):  $U = x_B - x_Q$ ,  $V = y_C - y_R$ ,  $W = x_S - x_D$ , and  $X = y_P - y_A$ . Now, by substituting (7)–(12) into the above expressions, we finally have

$$U = \Delta x + h(\cos \Delta\phi + \sin \Delta\phi - 1)/2 \quad (13)$$

$$V = \Delta y + h(\cos \Delta\phi + \sin \Delta\phi - 1)/2 \quad (14)$$

$$W = -\Delta x + h(\cos \Delta\phi + \sin \Delta\phi - 1)/2 \quad (15)$$

$$X = -\Delta y + h(\cos \Delta\phi + \sin \Delta\phi - 1)/2 \quad (16)$$

### 3.4 Forward Kinematics

In serial mechanisms, the inverse kinematic solution is generally hard to find, but the direct kinematics is

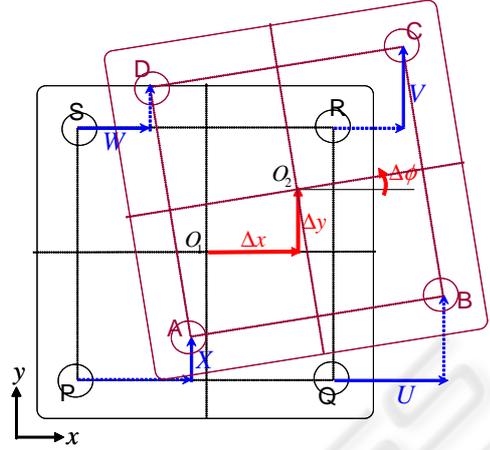


Figure 6: Moving distances of active (solid line) and passive (dotted line) joints for a misalignment posture.

straightforward. However, the situation is reversed in parallel mechanisms, where the inverse kinematics is rather simple as shown in the former section but the direct kinematics is more complicated. The inverse kinematics is important for the control purpose while the direct kinematics is also required for the kinematic analysis of end-point.

Although finding out the direct kinematics in position level could be a long-time procedure, it becomes an easy task in velocity level. By differentiating (13)–(16) with respect to the time, the following Jacobian relationship between joint space velocity and task space one is given.

$$\begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \\ \dot{X} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{1}{2}h(\cos \Delta\phi - \sin \Delta\phi) \\ 0 & 1 & \frac{1}{2}h(\cos \Delta\phi - \sin \Delta\phi) \\ -1 & 0 & \frac{1}{2}h(\cos \Delta\phi - \sin \Delta\phi) \\ 0 & -1 & \frac{1}{2}h(\cos \Delta\phi - \sin \Delta\phi) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} \quad (17)$$

which can be simply represented by

$$\dot{q}(t) = J(p)\dot{p}(t) \quad (18)$$

Considering the Jacobian  $J \in \mathbb{R}^{4 \times 3}$  in (18), the columns are linearly independent ( $\text{rank}(J) = 3$ ). Hence,  $J^T J \in \mathbb{R}^{3 \times 3}$  is invertible and there exists a left-inverse  $J^+ \in \mathbb{R}^{3 \times 4}$  such that  $J^+ = (J^T J)^{-1} J^T$  and the linear system (18) has a unique solution  $\dot{p}$  for every  $\dot{q}$  (Strang, 1988). In the sequel, the direct kinematic solution at current time can be determined by

$$\dot{p}(t) = J^+ \dot{q}(t) \quad (19)$$

$$\rightarrow p(t) = \int_{t_0}^t (J^T J)^{-1} J^T \dot{q}(t) dt \quad (20)$$

## 4 PATTERN RECOGNITION

A typical visual alignment system provides a structured vision environment. For example, CCD camera and illumination is under static condition and the objects to be recognized are just two kinds of alignment mark for mask and panel. Moreover, the shape of marks are usually symmetric ones such as circle, cross, rectangle, and diamond, in which case the feature points of an object can be easily determined. We consider just circle and cross shaped marks because they are most common in display and semiconductor industry. In this section, an efficient alignment mark recognition algorithm in Fig 10 is suggested by combining the conventional labeling technique and a geometric template matching method which is designed by analyzing the characteristics of alignment marks. Basically, it is assumed that the alignment marks are in the field-of-view (FOV) of cameras after the pre-alignment process in the industrial visual alignment system.

### 4.1 Preprocessing

**Bottom-Hat Transform:** Before applying the labeling algorithm, the gray image from the CCD camera is converted into the binary image. However, a proper binarization is impossible when the illumination is nonlinear or non-steady due to the limit of light source or other unexpected light disturbances. In order to eliminate these effects, we can use the bottom-hat transform which is a morphology method (Gonzalez and Wood, 2002), where the transformed image  $h$  can be calculated by

$$h = (f \cdot b) - f \quad (21)$$

where  $f$  represents the original image,  $b$  the circular morphology, and  $(f \cdot b)$  the closing operation between  $f$  and  $b$ . By the closing operation, objects smaller than the size of  $b$  are eliminated and the rest of background is extracted. The size of  $b$  is usually determined experimentally. The figure 7 shows an example.

**Dynamic Thresholding and Noise Filtering:** To segregate marks from the background in the Bottom-Hat transformed image, we apply the binarization algorithm. Among several methods to determine best threshold value, the following repetition method can be used: 1) Set the initial threshold  $T$ , which is the average between the maximum and the minimum of brightness from the binary image. 2) Divide the image into class  $G_1$  and  $G_2$  by  $T$  and calculate the averages  $m_1$  and  $m_2$  of them. 3) Calculate  $T = (m_1 + m_2)/2$ . 4) Repeat step 2 and step 3 until  $m_1$  and  $m_2$  are not changed.

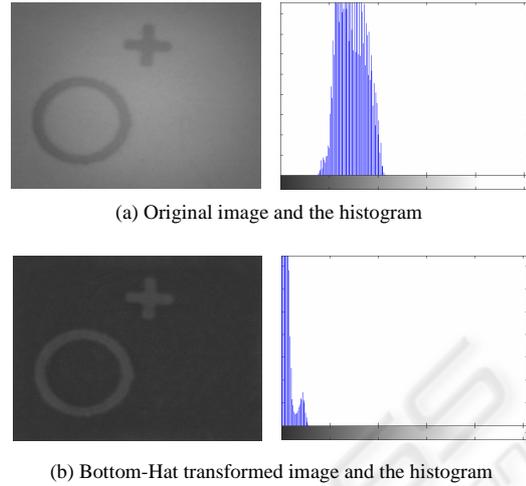


Figure 7: Comparison of original and transformed image and the histograms ( $b = 20$  pixels).

Small noise patterns in the binary image can be recognized as independent areas during the labeling process. They can be eliminated by the opening and closing method:

$$\hat{f} = (f \circ b) \cdot b \quad (22)$$

where the morphology  $b$  is a small (e.g., 3 by 3) square matrix whose components are all 1,  $(f \circ b)$  an opening operator, and  $\hat{f}$  the filtered image.

**Labeling:** If the objects in a captured image are not overlapped, an easily accessible algorithm to extract them is the labeling technique. To separate marks from the noise-filtered image, we first apply the labeling algorithm to the area chained by eight-connectivity. Once the labeling process is finished, a number of areas including the marks will be labeled. As explained in Fig 10, if the labeled area is not counted, a new image should be captured by repeating the pre-alignment process. If the labeled area is only one, it is highly possible that two marks are overlapped and each mark should be extracted by applying another pattern recognition scheme. If there are 2 labeled areas, the marks of mask and panel are not overlapped. Then, the centroids of marks can be simply calculated by the center of area method:

$$X_c = \frac{1}{n} \sum_{i=0}^{n-1} X_i, Y_c = \frac{1}{n} \sum_{i=0}^{n-1} Y_i \quad (23)$$

where  $X_c$  and  $Y_c$  represent the central point of labeled image and  $X_i$  and  $Y_i$  the horizontal and vertical positions of the pixels. Finally, if there are more than 3 labeled areas, as denoted in Fig 10, the areas which have less pixels than a specified number must be eliminated through an extra filtering process.

## 4.2 Geometric Template Matching

When the two marks of mask and panel are overlapped, i.e., when the number of labeled area is only one in Fig 10, the labeling algorithm alone is not enough but the marks can be separated in terms of any pattern matching method.

Since the alignment marks used in the display or semiconductor masks are very simple, their templates can be readily characterized by a few feature points. First, for the circular mark in Fig. 8(a) where the radius ( $r$ ) is a unique trait, for example, the eight pixels along the circumference can be selected as the feature points. All the pixels in the matching area can be scanned by assuming them as the center of circle. Then, the centroid of circular mark can be found when all the feature pixels have the same brightness of 255 (white) in the binarized image. In reality, since the actual circular mark has a thickness, every pixel that fulfills this condition must be stored in the memory stack and the final centroid can be calculated by averaging the coordinate values. Similarly, five feature points can be chosen for the cross mark with length  $l$  as in Fig. 8(b). However, differently from the circular mark, we have to consider the rotation factor. Hence, the matching process should be performed by rotating the template from 1 to 90 degrees for all pixels in the matching area.

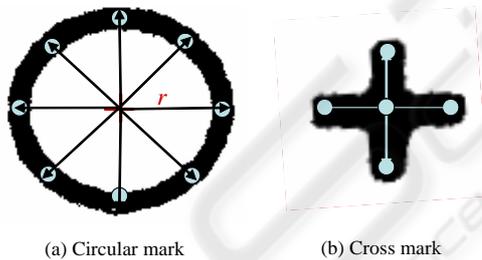


Figure 8: Feature pixels of circular mark (radius = 2 mm) and cross mark (length = 1 mm).



Figure 9: Matching areas for (a) circular mark and (b) cross mark.

The matching area for the circular mark is given

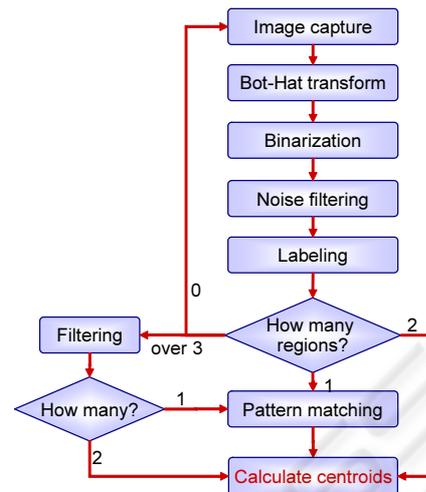


Figure 10: Overall flow of alignment mark recognition.

by the rectangle in Fig. 9(a) with the size of  $(M - 2r) \times (N - 2r)$  pixels in the  $M \times N$  pixel image. Under the assumption that the exact central point of the circular mark has been found, the matching area for the cross mark is equal to  $2(r + l) \times 2(r + l)$  pixels as shown in Fig. 9(b) regarding the farthest positions of the cross mark in the overlapped image.

The normalized correlation (NC) (Manickam et al., 2000) and the point correlation (PC) algorithm (Krattenthaler et al., 1994) are most widely used template matching methods in machine vision systems. However, since the NC requires a vector operation for all pixels in the image and template to determine the correlation coefficient, it is too time consuming. Although the PC algorithm reduces the dimension of computation time, it is still not appropriate for real-time applications because it includes feature points extraction process. On the contrary, in our matching algorithm, the feature points of alignment marks are geometrically extracted based on the analysis of mark shape and the individual feature points and the image pixels at the same coordinates are directly compared without any vector operations.

The overall sequence of the suggested algorithm for alignment mark recognition is described in Fig 10, where the feature pixels of marks can be determined in advance of labeling and the computation will be finished at the labeling process when the marks are not overlapped. As far as the mark shape is geometrically simple, as is the case in semiconductor and display industry, the combined algorithm of labeling and geometric pattern matching can be considered as a reasonable way to reduce the overall tact time.

## 5 CONTROL

The first step in the visual alignment process is to detect the centroids of alignment marks from the raw images. The visual processing can be divided into pre-processing and image analysis. As described in the former section 4, the pre-processing includes binarization, edge detection, and noise filtering etc. In the image analysis procedure, a few object recognition techniques can be applied such as labeling and template matching. Once the centroids of marks are determined, the misalignment distance and angle between mask and panel, which is usually less than hundreds of microns, can be readily determined using geometric relationships. Given the misalignment quantity for the current image, the driving distances of joint actuators can be produced by the inverse kinematic solution for a specific parallel mechanism as shown in Section 3. Finally, the misalignment can be compensated by the joint controller as in Fig. 11, where the outer visual feedback loop should be repeated until it meets permissible alignment errors.

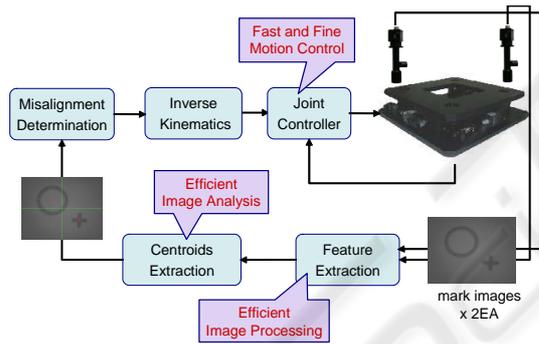


Figure 11: Vision-based look and move motion control.

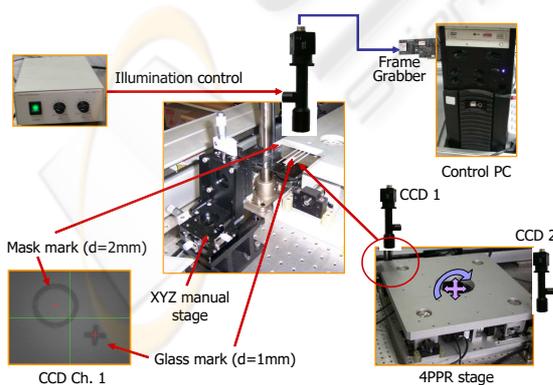


Figure 12: Experimental setup for visual alignment.

Figure 12 shows the experimental setup for the mask and panel alignment, where the CCD image

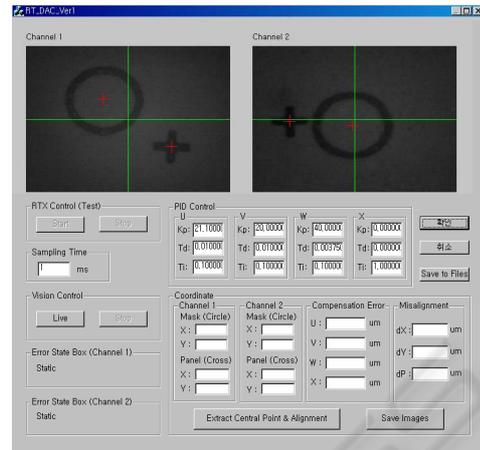


Figure 13: Graphic User Interface of operating software.

has the size of  $640 \times 480$  pixels and the field of view (FOV) is  $6.4 \times 4.8$  mm and the rate of capture in the frame grabber is 15 fps. Circular marks have a diameter of 2 mm and thickness of  $200 \mu\text{m}$  and the cross marks have the length of 1 mm. The graphic user interface shown Fig. 13 was developed to integrate all the functions to achieve the autonomous visual alignment.

First of all, we compared the performance of the alignment mark recognition algorithm in Fig. 10 with the popular NC (Manickam et al., 2000) and PC method (Krattenthaler et al., 1994). For the captured images in Fig. 13, the proposed method takes 37 msec on average to find the centroids of all marks including pre-processing time, while the NC and PC required 661 msec and 197 msec, respectively. As a result, the proposed method reduces one order of recognition time. As explained before, this is mainly because in the geometric template matching the vector operations for all pixels are not necessary unlike the NC and the feature points of objects are selected in advance unlike the PC. Although the geometric pattern matching is confined to simple objects, it is actually enough to extract alignment marks.

Figure 14 shows the visual alignment process for a given misaligned posture between mask and panel. As explained in Fig. 11, the inverse kinematic solutions are cast into the joint controllers as a reference input. To avoid excessive chattering, at every alignment cycle, we have applied the polynomial trajectory with rise time 0.6 sec for the reference values. In Fig. 14, the mask and panel were almost aligned after the 1st cycle and the 2nd cycle was activated since the misalignments in U-axis and W-axis are still over the tolerance. Considering the joint control error in Fig. 15 for the 2nd cycle, the controlled motion of V-axis

is not smooth. When the reference input is too small, a stick-slip motion may occur at low velocities and this makes precision control very difficult.

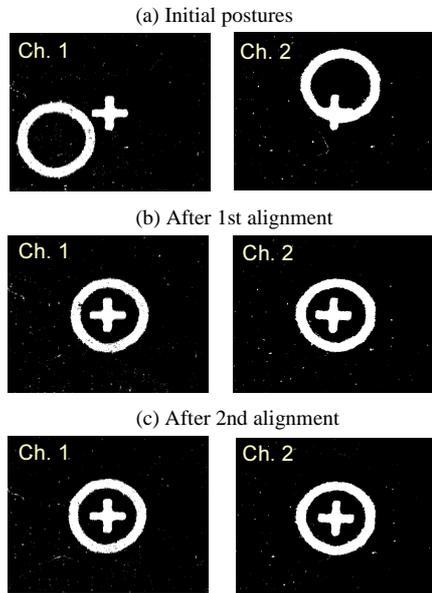


Figure 14: Visual alignment experiment: the inverse kinematic solution is  $(U, V, W) = (132.4, -592.6, -1367.6)\mu\text{m}$  at initial posture and  $(U, V, W) = (-73.4, -3.5, 66.5)\mu\text{m}$  after the 1st alignment.

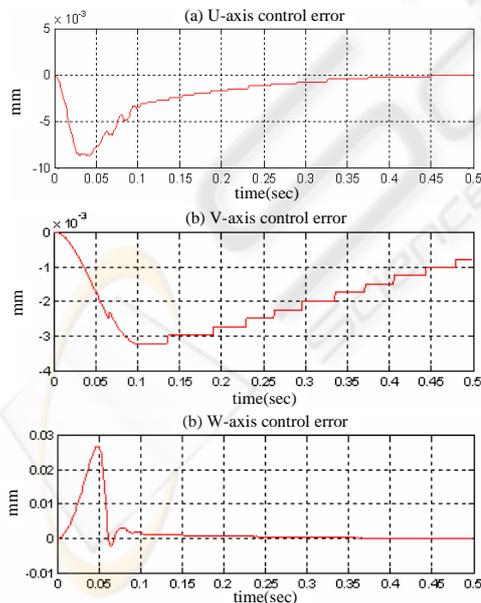


Figure 15: Joint control errors during the 2nd alignment (the 4th X-axis was not active).

## 6 CONCLUSION

In this paper, we investigated the visual alignment problem which is considered as a core requirement in flat panel display and semiconductor fabrication process. The kinematics of the 4PPR parallel mechanism was given and an efficient vision algorithm in terms of geometric template matching was developed for fast recognition of alignment marks. Through the control experiment, the proposed method was proven to be very effective.

## REFERENCES

- Cognex Ltd. (2004). In <http://www.cognex.com/products/VisionTools/PatMax.asp>.
- Gonzalez, R. C. and Wood, R. E. (2002). *Digital Image Processing*. Prentice Hall.
- Hephaist Ltd. (2004). In [http://www.hephaist.co.jp/pro/n\\_4stage.html](http://www.hephaist.co.jp/pro/n_4stage.html).
- Kang, D. J. and Lho, T. J. (2003). Development of an edge-based point correlation algorithm for fast and stable visual inspection system. In *Journal of Control, Automation and System Engineering*. Vol. 9, No. 2, August, 2003 (in Korean).
- Kanjilal, A. K. (1995). Automatic mask alignment without a microscope. In *IEEE Trans. on Instrumentation and Measurement*. Vol. 44, pp. 806-809, 1995.
- Kim, H. T., Song, C. S., and Yang, H. J. (2004). 2-step algorithm of automatic alignment in wafer dicing process. In *Microelectronics Reliability*. Vol. 44, pp. 1165-1179, 2004.
- Krattenthaler, W., Mayer, K. J., and Zeiller, M. (1994). Point correlation: a reduced-cost template matching technique. In *IEEE Int. Conf. on Image Processing*. pp. 208-212, 1994.
- Lai, S.-H. and Fang, M. (2002). A hybrid image alignment system for fast and precise pattern localization. In *Real-Time Imaging*. Vol. 8, pp. 23-33, 2002.
- Manickam, S., Roth, S. D., and Bushman, T. (2000). Intelligent and optimal normalized correlation for high-speed pattern matching. In *Datacube Technical Paper*. Datacube Incorporation.
- Martin, B. R. (2001). Self-alignment of patterned wafers using capillary forces at a wafer-air interface. In *Advanced Functional Materials*. Vol. 11, pp.381-386, 2001.
- Strang, G. (1988). *Linear Algebra and Its Applications*. College Publishers, 3rd edition.
- Tsai, L.-W. (1999). *Robot Analysis: The mechanics of serial and parallel manipulators*. Wiley-Interscience.
- Umminger, C. B. and Sodini, C. G. (1995). An integrated analog sensor for automatic alignment. In *IEEE Trans. on Solid-State Circuits*. Vol. 30, pp. 1382-1390, 1995.