

Planning for Optimal Robot Trajectory for Visual Inspection

Ahmad Yusairi Bani Hashim*, Nur Suffiah Akmala Ramdan, Radin Puteri Hamizah Radin Monawir, Mohd Nazrin Muhammad

Department of Robotics and Automation, Faculty of Manufacturing Engineering, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Melaka, Malaysia

*Corresponding author: yusairi@utem.edu.my

Received October 26, 2013; Revised December 30, 2013; Accepted January 24, 2014

Abstract Automatic inspection is common in mass production inspections where robot manipulators are chosen to perform visual inspection to avoid inconsistency in manual inspection. The purpose of this work is to estimate the optimum workspace where a robot manipulator could perform a visual inspection tasks onto a work piece where a camera is attached to the end effector. While maneuvering through the programmed path, the robot will stop at a predefined point so that an image could be captured where the ideal parameter for the coefficient correlation (CC) template matching was computed. The template distance selection, the threshold adjustment and the average image processing speed for CC were determined. The pseudo codes for the planned path are derived from the number of tool transit points, the delay time to the transit points, the process cycle time, and the configuration space that the distance between the tool and the work piece. It was observed that express start and swift end were acceptable in a robot program because applicable works usually in existence during these moments. However, during the mid-range cycle, there are always practical tasks programmed to be executed. For that reason, it is acceptable to program the robot such as that speedy alteration of actuator displacement is avoided. A dynamic visual inspection system using a robot manipulator seems practical where an optimal distance between an object face and a camera is determined in priori using the CC method.

Keywords: robot manipulator, visual inspection, coefficient correlation

Cite This Article: Ahmad Yusairi Bani Hashim, Nur Suffiah Akmala Ramdan, Radin Puteri Hamizah Radin Monawir, and Mohd Nazrin Muhammad, "Planning for Optimal Robot Trajectory for Visual Inspection." *Journal of Automation and Control*, vol. 2, no. 1 (2014): 15-20. doi: 10.12691/automation-2-1-3.

1. Introduction

Manual inspection is still the choice for final inspection in the quality controls. However, inconsistency in human inspection is real because humans can get tired after some time. Human is incapable of focusing on the repeating work. It is difficult and costly to hire and train human experts. It is fair to claim that human cannot achieve consistency as compared to automatic inspection, such as using a vision system. In fact, there are also cases where inspection tends to be tedious or difficult, even for the best-trained expert [1].

Recent trends show that automatic inspection is more appealing for mass production inspections. In such as a case, a robot manipulator seems the best candidate to run a dynamic visual inspection. Robot manipulator is a machine formed by a mechanism, including a number of degrees of freedom, often having the appearance of one or some arm ends in a wrist capable of holding a tool, a work piece or an inspection device [2]. A manipulator is a mechanical unit that provides motions or trajectories similar to those of a human arm and hand. A robot manipulator, on the other hand, provides trajectories by programmed instructions.

On a manipulator, the end of the wrist can reach a fixed point having a particular set of coordinates and in a specific orientation. This is realized by the application of an end effector where the end of the wrist in a robot is equipped with an end-of-arm tooling. An end effector may be equipped with a special tooling [3].

Robot manipulator can be classified by two basic structural types: the parallel structure and the serial structure [4]. For a serial structure, it is constructed in such a way as to form the shape of an elbow, wrist and shoulder [5]. One of the major advantages of the serial type over the parallel type is its workspace that is larger than that of the parallel type.

The purpose of this work is to estimate the optimal workspace where a robot manipulator performs a visual inspection process onto a work piece where a camera is attached to the end effector. While planning for the trajectory, the joints' position, velocity, and acceleration are carefully planned that the obstacles are to be avoided, and the tool is to travel along the shortest path, and the cycle time is within a desired one.

A vision system that is fixed onto a location has a limited image capturing area. The idea is to attach a camera onto the end effector. For the robot while in motion, the camera will have a wider capturing area and at

an ideal angle with respect to a complex shaped work piece.

2. Methodology

This work carries out the following sequence of activities: identification of the method in path and trajectory planning; designing the program structure; developing the robot program; and assessment the robot's paths and trajectories.

2.1. Path and Trajectory Planning

A robot manipulator performs pre-planned tasks by controlling the rate of movement of its actuators. This process is known as the trajectory planning. The rates and sequences of actuator movement are programmed so that a desired path is obtained. A continuous map, with $\gamma[0,1] \rightarrow \Theta$ with $\gamma(0)=q$ has a path that begins at a start point (q_o) to final point (q_f) in a configuration space. A trajectory is a function of time $q(t)$ such that $q(t_o)=q_o$ and $q(t_f)=q_f$. The difference $t_f - t_o$ is the time taken to execute the trajectory [6]. The trajectory itself explains the rate of actuator rotation on individual manipulator joint. Therefore, there are six of this functions that define for the trajectories of six actuators.

Consider a cubic trajectory based on a single joint trajectory. For a scalar joint variable, $q(t)$ at time t_o the joint variable satisfies constraint (1). A cubic trajectory and its derivative are defined in (2), whereas (3) is the algebraic representation of the trajectory. The constraints and definitions stated in (1), (2), and (3) are used to plan for robot's tool path where the actuator's angles of rotation, velocities and accelerations are realized through robot's program.

$$\left. \begin{aligned} q(t_o) = q_o, \dot{q}(t_o) = v_o, q(t_f) = q_f, \dot{q}(t_f) = v_f, \\ \ddot{q}(t_o) = \alpha_o, \ddot{q}(t_f) = \alpha_f. \end{aligned} \right\} \quad (1)$$

$$\left. \begin{aligned} q(t) &= a_0 + a_1t + a_2t^2 + a_3t^3 \\ \dot{q}(t) &= a_1 + 2a_2t + 3a_3t^2 \end{aligned} \right\} \quad (2)$$

$$\begin{bmatrix} 1 & t_o & t_o^2 & t_o^3 \\ 0 & 1 & 2t_o & 3t_o^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_o \\ \dot{q}_o \\ q_f \\ \dot{q}_f \end{bmatrix} \quad (3)$$

2.2. Robot Program Structure

The pseudo codes for a planned path are derived from the number of tool transit points, the delay time at the transit points, the process cycle time, and the configuration space that the robot makes. Table 1 shows the transit points (vertex-position), the assigned coordinates, the path that passes through the transit points. There are eight vertices within the path that the tool has to travel along.

Table 1. On the planned trajectory where there are eight designated points where the tool has to transit

Vertex	Position	X	Y	Z	Path	Remark
0	Home	x_0	y_0	z_0	7-1	Robot home
1	Initial	x_1	y_1	z_1	0-1	Ready to initiate path
2	P1	x_2	y_2	z_2	1-2	Defect area 1
3	P2	x_3	y_3	z_3	2-3	Defect area 2
4	P3	x_4	y_4	z_4	3-4	Defect area 3
5	P4	x_5	y_5	z_5	4-5	Defect area 4
6	P5	x_6	y_6	z_6	5-6	Defect area 5
7	End	x_7	y_7	z_7	6-7	Ready to return home

2.3. Experiment Setup



Figure 1. The car door (works piece) is shown stood up, and the end effector has a camera installed. The front view of the end effector is marked in the box within the figure

The robot manipulator was used to realize the automatic visual inspection where the manipulator's function was to

bring the camera to the pre-defined locations within the configuration space. The camera shootings were realized

by another separate system. Figure 1 exhibits the experiment setup that consists of the robot itself, a work piece that is a car door, and a camera that is attached onto the end effector. The white markings were plastic straws that functioned as indicators that limit the configuration space. They provided the limit within which the camera should reach but to avoid any obstacles. Prior to the

realization of the inspection process, the setup was simulated in Workspace. Figure 2 shows the simulated environment of the setup. The results obtained from the simulation would explain the behavior of the individual actuator and this would reflect to equations (1), (2), and (3).

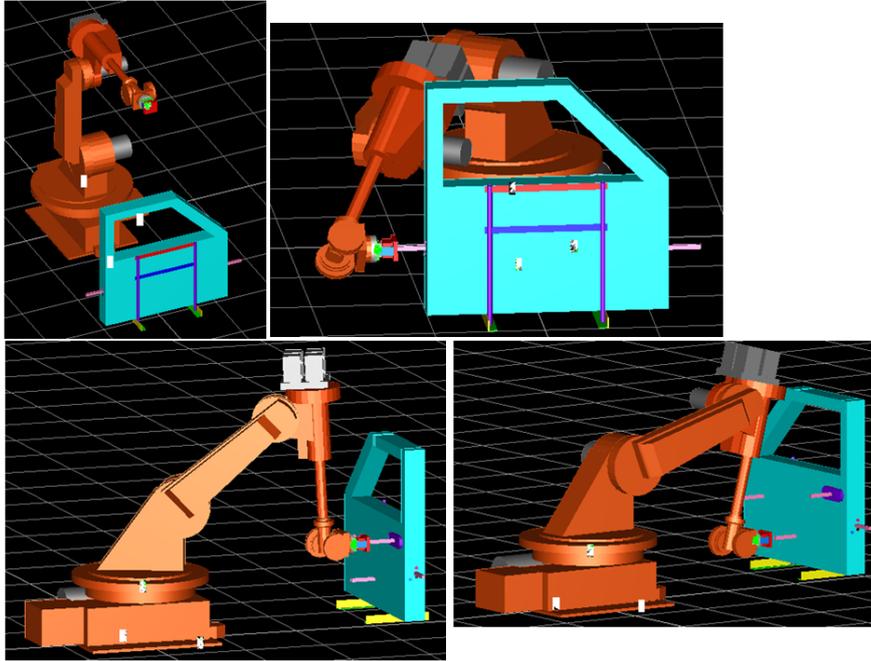


Figure 2. Prior to setting up the experiment, a number of simulation exercises were performed. The results from the exercise were used to justify the set up exhibited in Figure1 above

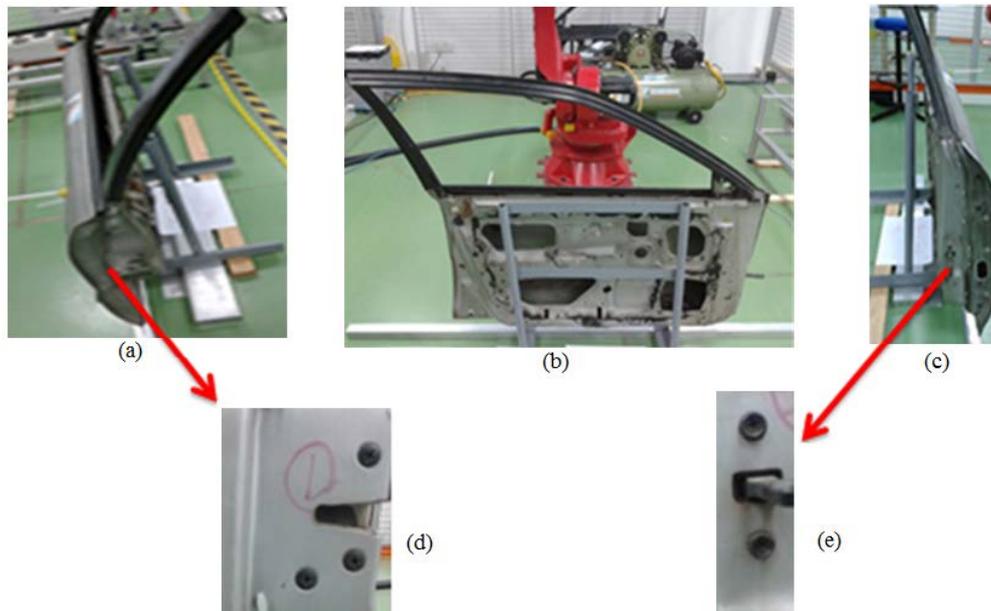


Figure 3. The selected inspection locations are shown in the figure where figures (d) and (e) are the images of transit spots such that the robot was programmed to have the end effector stop momentarily and at proximity to the spot for inspection. Figures (a) and (c) are the views of the work piece, whereas figure (b) shows the rear view

While maneuvering through the programmed path, the robot will stop at the predefined point so that an image could be captured. Figure 3 shows example of the points where image areas are selected such as shown in Figure 3d and Figure 3e. This sub process is conducted to find the optimum parameter for the coefficient correlation (CC) template matching. The template distance selection, the

threshold adjustment and the average image processing speed for CC are determined. The sample images are tested while manipulating threshold value (α) and the distance between camera and object (L). The sample of images captured within the range $15cm \leq L \leq 35cm$.

3. Results and Discussion

The realization of the experimental setup that is based on the simulation environment depicted Figure 2. Figure 4, Figure 6, and Figure 8 describe the joint behaviors in terms of positions, velocities, and accelerations resulted from simulation exercises set to run within 80 seconds. The simulation was performed on Workspace software. The robot would have to initiate an inspection task such that the robot joints would rotate at prescribed angles. In Figure 4, all joints are active during the inspection process from start to end where joint-4 has the greatest angle deviation, whereas joint-6 has the slightest. In actual implementation (Figure 5), however, joint-4 and joint-6 have the greatest angle deviations, whereas joint-2 has the least.

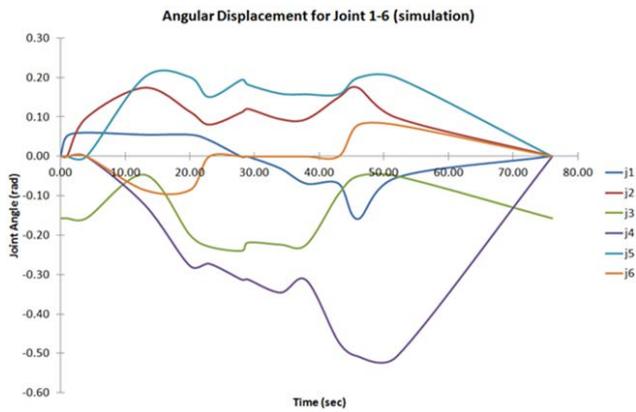


Figure 4. The displacement curves that elucidate every joint’s angular position with respect to time obtained from a simulation exercise. The figure shows that joint-4 has the largest angle deviation, whereas joint-6 has the least

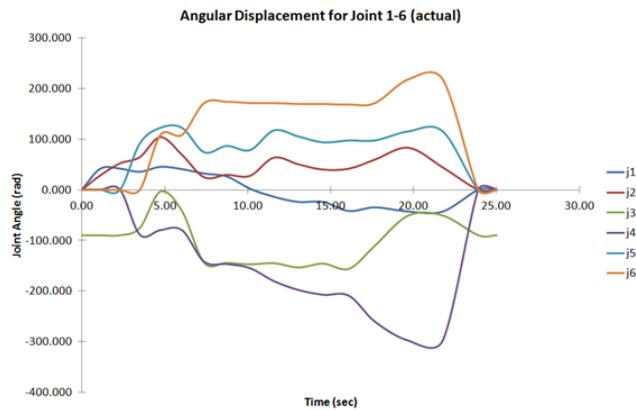


Figure 5. The displacement curves that clarifies each joint’s angular position with respect to time obtained from an actual implementation on the robot arm. The figure shows that joint-4 and joint-6 have the largest angle deviations, whereas joint-2 has the least

In Figure 6, joint-4 is seen to consume energy the most, whereas joint-6 was the least. It shows that joint-1 and joint-4 would have the most rapid angular change, especially towards the end to the cycle, whereas the remaining joints would seem to slow down towards the end. In actual implementation (Figure 7), however, joint-1 and joint-6 would have the most rapid angular change, especially towards the end to the cycle, whereas the remaining joints would seem to slow down towards the end.

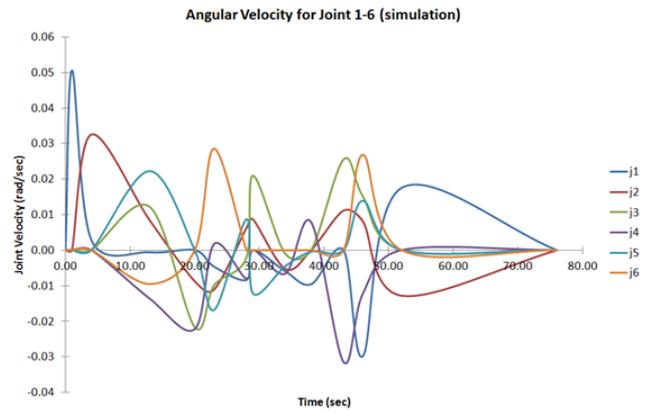


Figure 6. The velocity curves that explain individual joint’s velocity with respect to time obtained from a simulation exercise. The figure shows that joint-1 and joint-4 has the most rapid angular change especially towards the end of the cycle, whereas the remaining joints seem to slow down towards the end

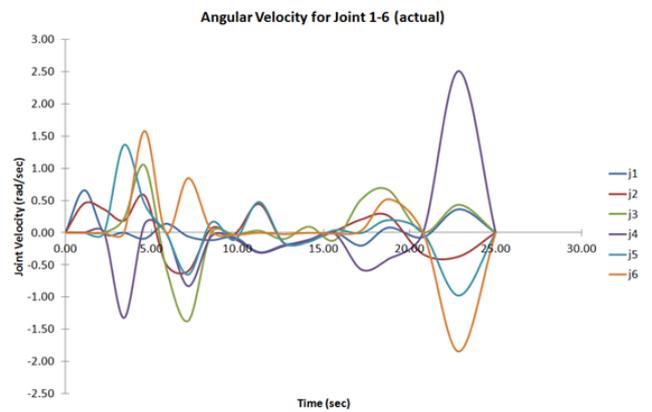


Figure 7. The velocity curves that explain individual joint’s velocity with respect to time obtained from an actual implementation. The figure shows that joint-1 and joint-6 has the most rapid angular change especially towards the end of the cycle, whereas the remaining joints seem to slow down towards the end

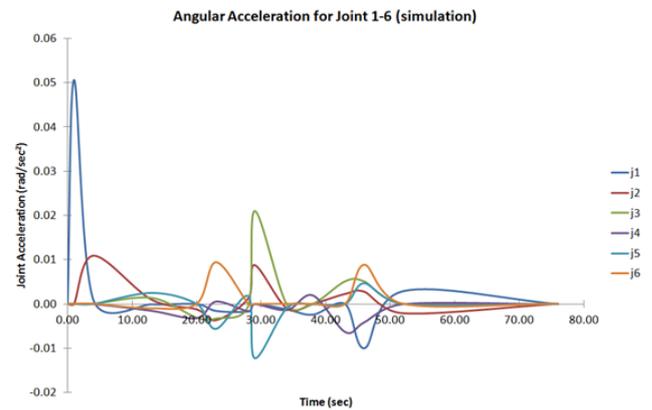


Figure 8. The acceleration curves that explain individual joint’s acceleration with respect to time obtained from a simulation exercise. The figure shows that joint-1 has the most rapid initial acceleration, whereas the remaining joints seem to slow down towards the end but joint-3 and joint-5 exhibit rapid change at mid cycle

In Figure 8, joint-1 would have the most rapid initial acceleration, whereas the remaining joints would seem to slow down towards the end but joint-3, and joint-5 would exhibit rapid changes at mid cycle. In the actual implementation (Figure 9), however, all joints would have rapid initial acceleration, and they would seem to slow

down at mid cycle Towards the end, however, joint-4 and joint-6 would exhibit rapid changes in acceleration.

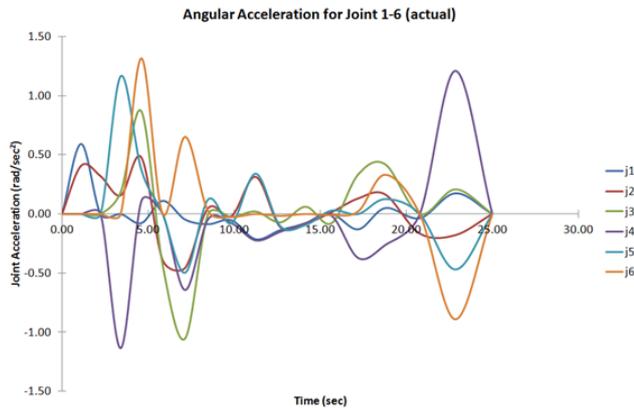


Figure 9. The acceleration curves that explain individual joint's acceleration with respect to time obtained from a simulation exercise. The figure shows that all joints have rapid initial acceleration and they seem to slow down at mid cycle. Towards the end, however, joint-4 and joint-6 exhibit rapid change

Based on the simulation results, it was observed that not all joints needed to be controlled in order to achieve the optimum configuration space following the specifications listed in Table 1. Figure 5 shows the actual joint displacement behavior within a cycle time of 25.02 seconds as compare to 80 seconds when run on software.

It is less than that the simulation cycle time because of the improved configuration space. In addition, it is observed that displacement for all joints was moderated where each displaced accordingly. All joints were functioning while the tool in motion.

There were rapid speed changes where all joints were functioning throughout the cycle (see Figure 6), whereas in the actual setup, most of the joints are somewhat steady during the mid-range of the cycle (see Figure 7). The mid-range of the cycle was where the inspection was carried out. During simulation, the robot had a swift start and a quick end. Similarly, in the actual implementation, the trajectories were fast start and rapid end. This is further explained in Figure 8 and Figure 9 where the robot had slight decelerations in simulation environment, but with considerable rapid decelerations in the actual setup.

The images shown in Figure 10 are the actual implementations of the inspection program based on the justification of the joint angle behavior concluded from

the simulation exercises. The distance limits (tool to spot) were marked with plastic ties so that the tool would not collide with the work piece.

The image of motion frames seen in Figure 10 are the four selected robot motion images during the inspection process. The frames show the transit points upon which the camera captured images of areas that were likely to have defects. There are five points (Table 1) where the tool needed to stop for a moment during which the camera would capture images of the defect areas. The images were then uploaded for image processing to determine the degrees of defects.

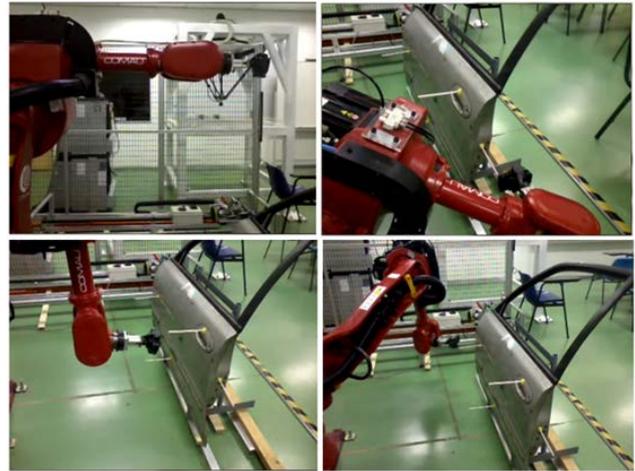


Figure 10. The images show the actual implementation of the inspection program. The distance limits were marked with a plastic ties so that the tool would not collide with the work piece

In general, a template of an image at distance $L = X$ is suitable to be match with image taken through a template at the same distance. A template of image at $= Y$, however, could match a template of image at $L = X$ [7]. Therefore, this work implemented the template of image as such where the first parameter of the algorithm that was the template distance was found. Template size of $0.25cm \times 0.25cm$ for images at distance range from $15cm \rightarrow 30cm$ was tested. Template of $L = 35$, however, was not taken as a sample template because the distance was too far. In fact, an image captured through the template could result in loss of image features, hence poor accuracy.

Table 2. Range of template matching when template from input at different L

Template from distance, L (cm)	Number of points detected (overall point detect, side 1=3 at L)					Distance of detection (cm)	Percent detection (%)
	15	20	25	30	35		
15	3/3	1/3	-	-	-	15,20	26.67
20	1/3	3/3	2/3	2/3	-	15,20,25,30,35	53.33
25	1/3	1/3	3/3	1/3	1/3	15,20,25,30,35	46.67
30	-	1/3	1/3	3/3	3/3	20,25,30,35	53.33

The results are listed in Table 2. For $L = 15$ cm, the image retains most of its features. The value 3/3 dictates that 3 out of 3 points were detected. The template cropped from $L = 15$ cm was found to be acceptable for short ranges: of 15cm and of 20 cm, respectively. The degree of detection, however, was the lowest—26.67%. Conversely, there were two cases where the degree of detection was the highest: at $L = 20$ cm and at $L = 30$ cm, respectively. These templates could detect more points at different

distances. Nevertheless, the template with $L = 20$ cm was selected to reduce the risk of losing image features. Figure 11 depicts the templates that match the points based on the CC method. The points are shown within the boxes in Figure 11b. In addition, Figure 12 explains, by the correlation measurement space, how the points were detected by chromatic scale such that the equivalent value of 0.8 represented the detected region.

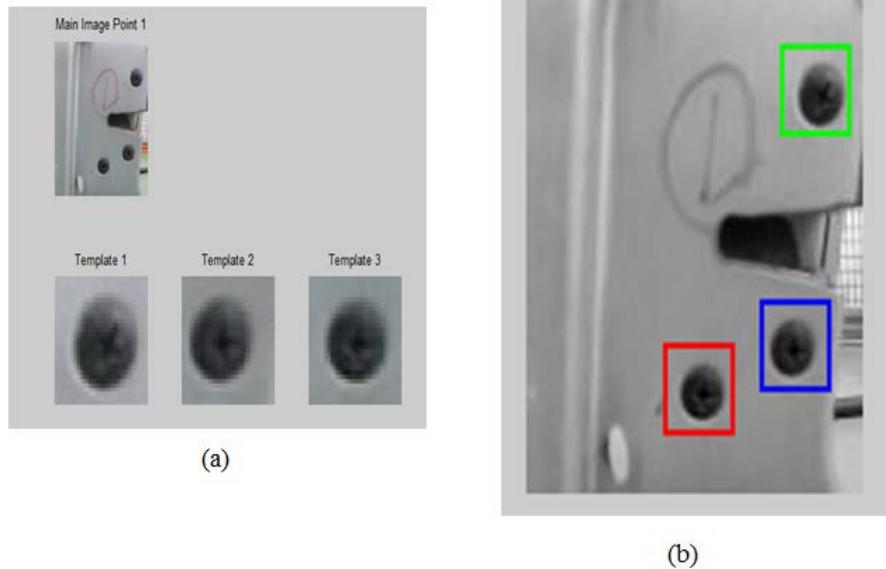


Figure 11. The template matching based on the correlation coefficient approach

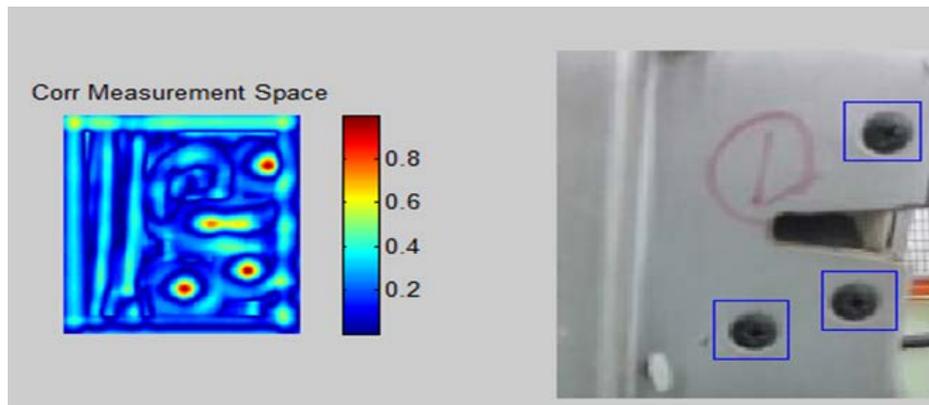


Figure 12. The correlation measurement space exhibits chromatic degree of detection where the equivalent value of 0.8 represents the detected region

4. Conclusion

A trajectory of an individual joint may be observed from the plots of respective curves. From the curves, the planner may program the robot such that the joints will be instructed to rotate by means of actuators that intake some amount of electric current from the driver circuits. This work provides a solution to automatic visual inspection where a camera is attached onto the end effector of a manipulator. The camera will capture images at pre-defined stopover and at a preset distance between the camera and the object's face. A special configuration space is designed so that the cycle time is optimal where the camera will be at the right pose at every transit point. An express start and a swift end are acceptable in a robot program because useful works usually do not exist during these moments. However, during the mid-range cycle, there are always useful tasks to be executed. For that reason, it is acceptable to program a robot such as that speedy alteration of actuator displacement is avoided. In short, a dynamic visual inspection system using a robot manipulator seems practical where an optimal distance

between an object face and a camera is determined in priori using the CC method.

References

- [1] E. N. Malamas, E. G. Petrakis, M. Zervakis, L. Petit, and J.-D. Legat, "A survey on industrial vision systems, applications and tools," *Image and Vision Computing*, vol. 21, no. 2, pp. 171-188, Feb. 2003.
- [2] S. Kalpakjian and S. Schmid, *Manufacturing Engineering & Technology* (7th Edition). Prentice Hall, 2013, p. 1224.
- [3] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot Manipulator Control: Theory and Practice*. CRC Press, 2002, p. 430.
- [4] Y. Li, W. Wang, H. Li, and Y. Ding, "Feedback method from inspection to process plan based on feature mapping for aircraft structural parts," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 3, pp. 294-302, Jun. 2012.
- [5] C. D. Crane III and J. Duffy, "Kinematic Analysis of Robot Manipulators," Jan. 1998.
- [6] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley, 2005, p. 496. M.
- [7] Nixon and A. S. Aguado, *Feature Extraction & Image Processing* (Google eBook). Academic Press, 2008, p. 424.