28

Vision-Guided Robot Control for 3D Object Recognition and Manipulation

S. Q. Xie, E. Haemmerle, Y. Cheng and P. Gamage
Mechatronics Engineering Group, Department of Mechanical Engineering,
The University of Auckland
New Zealand

1. Introduction

Vision-guided robotics has been one of the major research areas in the mechatronics community in recent years. The aim is to emulate the visual system of humans and allow intelligent machines to be developed. With higher intelligence, complex tasks that require the capability of human vision can be performed and replaced by machines. The applications of visually guided systems are many, from automatic manufacturing (Krar and Gill 2003), product inspection (Abdullah, Guan et al. 2004; Brosnan and Sun 2004), counting and measuring (Billingsley and Dunn 2005) to medical surgery (Burschka, Li et al. 2004; Yaniv and Joskowicz 2005; Graham, Xie et al. 2007). They are often found in tasks that demand high accuracy and consistent quality which are hard to achieve with manual labour. Tedious, repetitive and dangerous tasks, which are not suited for humans, are now performed by robots. Using visual feedback to control a robot has shown distinctive advantages over traditional methods, and is commonly termed visual servoing (Hutchinson et al. 1996). Visual features such as points, lines, and regions can be used, for example, to enable the alignment of a manipulator with an object. Hence, vision is a part of a robot control system providing feedback about the state of the interacting object.

The development of new methods and algorithms for object tracking and robot control has gained particular interest in industry recently since the world has stepped into the century of automation. Research has been focused primarily on two intertwined aspects: tracking and control. Tracking provides a continuous estimation and update of features during robot/object motion. Based on this sensory input, a control sequence is generated for the robot. More recently, the area has attracted significant attention as computational resources have made real-time deployment of vision-guided robot control possible. However, there are still many issues to be resolved in areas such as camera calibration, image processing, coordinate transformation, as well as real time control of robot for complicated tasks.

This chapter presents a vision-guided robot control system that is capable of recognising and manipulating general 2D and 3D objects using an industrial charge-coupled device camera. Object recognition algorithms are developed to recognize 2D and 3D objects of different geometry. The objects are then reconstructed and integrated with the robot controller to enable a fully vision-guided robot control system. The focus of the chapter is placed on new methods and technologies for extracting image information and controlling a serial robot. They are developed to recognize an object to be manipulated by matching image features to a geometrical model of the object and compute its position and orientation (pose) relative to the robot coordinate system. This absolute pose and cartesian-space information is used to move the robot to the desired pose relative to the object. To estimate the pose of the object, the model of the object needs to be established. To control the robot based on visual information extracted in the camera frame, the camera has to be calibrated with respect to the robot. In addition, the robot direct and inverse kinematic models have to be established to convert cartesian-space robot positions into joint-space configurations. The robot can then execute the task by performing the movements in joint-space.

The overall system structure used in this research is illustrated in Figure 1. It consists of a number of hardware units, including the vision system, the CCD video camera, the robot controller and the ASEA robot. The ASEA robot used in this research is a very flexible device which has five degrees of freedoms (DOF). Such a robot has been widely used in industry automation and medical applications. Each of the units carries out a unique set of inter-related functions.

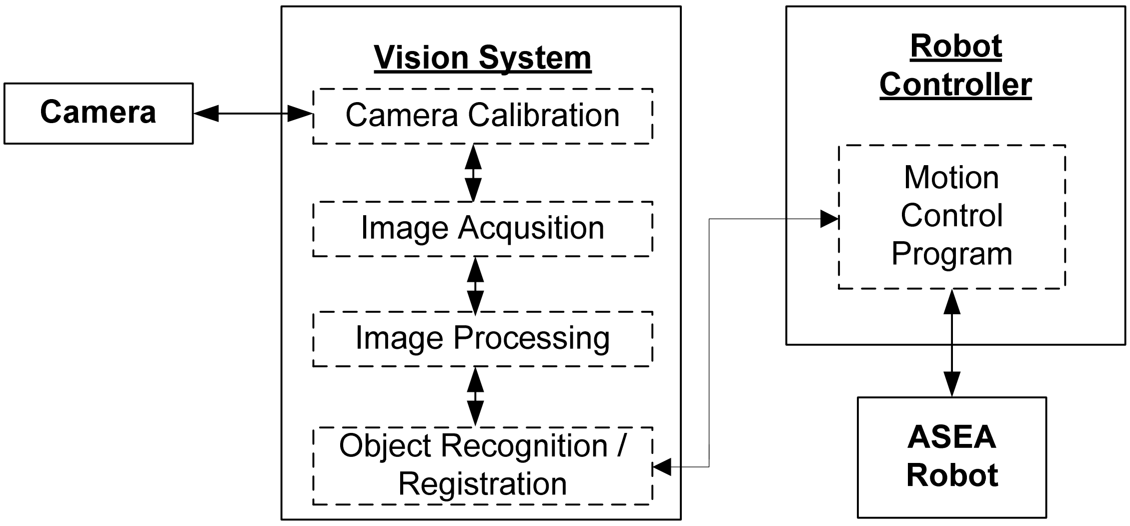


Figure 1. The overall system structure

The vision system includes the required hardware and software components for collecting useful information for the object recognition process. The object recognition process undergoes five main stages: (1) camera calibration; (2) image acquisition; (3) image and information processing; (4) object recognition; and (5) results output to the motion control program.

2. Literature Review

2D object recognition has been well researched, developed and successfully applied in many applications in industry. 3D object recognition however, is relatively new. The main issue involved in 3D recognition is the large amount of information which needs to be dealt with. 3D recognition systems have an infinite number of possible viewpoints, making it difficult to match information of the object obtained by the sensors to the database (Wong, Rong et al. 1998).

Research has been carried out to develop algorithms in image segmentation and registration to successfully perform object recognition and tracking. Image segmentation, defined as the separation of the image into regions, is the first step leading to image analysis and interpretation. The goal is to separate the image into regions that are meaningful for the specific task. Segmentation techniques utilised can be classified into one of five groups (Fu and Mui 1981), threshold based, edge based, region based, classification (or clustering) based and deformable model based. Image registration techniques can be divided into two types of approaches, area-based and feature-based (ZitovZitova and Flusser 2003). Area-based methods compare two images by directly comparing the pixel intensities of different regions in the image, while feature-based methods first extract a set of features (points, lines, or regions) from the images and then compare the features. Area-based methods are often a good approach when there are no distinct features in the images, rendering feature-based methods useless as they cannot extract useful information from the images for registration. Feature-based methods on the other hand, are often faster since less information is used for comparison. Also, feature-based methods are more robust against viewpoint changes (Denavit and Hartenberg 1995; Hartley and Zisserman 2003), which is often experienced in vision-based robot control systems.

Object recognition approaches can be divided into two categories. The first approach utilises appearance features of objects such as the colour and intensity. The second approach utilises features extracted from the object and only matches the features of the object of interest with the features in the database. An advantage of the feature-based approaches is their ability to recognise objects in the presence of lighting, translation, rotation and scale changes (Brunelli and Poggio 1993; ZitovZitova and Flusser 2003). This is the type of approach used by the PatMax algorithm in our proposed system.

With the advancement of robots and vision systems, object recognition which utilises robots is no longer limited to manufacturing environments. Wong et al. (Wong, Rong et al. 1998) developed a system which uses spatial and topological features to automatically recognise 3D objects. A hypothesis-based approach is used for the recognition of 3D objects. The system does not take into account the possibility that an image may not have a corresponding model in the database since the best matching score is always used to determine the correct match, and thus is prone to false matches if the object in an image is not present in the database. Büker et al. (BBuker, Drue et al. 2001) presented a system where an industrial robot system was used for the autonomous disassembly of used cars, in particular, the wheels. The system utilised a combination of contour, grey values and knowledge-based recognition techniques. Principal component analysis (PCA) was used to accurately locate the nuts of the wheels which were used for localisation purposes. A coarse-to-fine approach was used to improve the performance of the system. The vision system was integrated with a force torque sensor, a task planning module and an unscrewing tool for the nuts to form the complete disassembly system.

Researchers have also used methods which are invariant to scale, rotation, translation and partially invariant to affine transformation to simplify the recognition task. These methods allow the object to be placed in an arbitrary pose. Jeong et al. (Jeong, Chung et al. 2005) proposed a method for robot localisation and spatial context recognition. The Harris detector (Harris and Stephens 1988) and the pyramid Lucas-Kanade optical flow methods were used to localise the robot end-effector. To recognise spatial context, the Harris detector and scale invariant feature transform (SIFT) descriptor (Lowe 2004) were employed. Peña-Cabrera et al. (Pena-Cabrera, Lopez-Juarez et al. 2005) presented a system to improve the performance of industrial robots working in unstructured environments. An artificial neural network (ANN) was used to train and recognise objects in the manufacturing cell. The object recognition process utilises image histogram and image moments which are fed into the ANN to determine what the object is. In Abdullah et al.'s work (Abdullah, Bharmal et al. 2005), a robot vision system was successfully used for sorting meat patties. A modified Hough transform (Hough 1962) was used to detect the centroid of the meat patties, which was used to guide the robot to pick up individual meat patties. The image processing was embedded in a field programmable gate array (FPGA) for online processing.

Even though components such as camera calibration, image segmentation, image registration and robotic kinematics have been extensively researched, they exhibit shortcomings when used in highly dynamic environments. With camera calibration in real time self-calibration is a vital requirement of the system while with image segmentation and registration it is crucial that the new algorithms will be able to operate under the presence of changes in lighting conditions, scale and with blurred images, e.g. due to robotic vibrations. Hence, new robust image processing algorithms will need to be developed. Similarly there are several approaches available to solve the inverse kinematics problem, mainly Newton-Raphson and Neural Network algorithms. However, they are hindered by accuracy and time inefficiency respectively. Thus it is vital to develop a solution that is able to provide both high accuracy and a high degree of time efficiency at the same time.

3. Methods

The methods developed in our group are mainly new image processing techniques and robot control methods in order to develop a fully vision-guided robot control system. This consists of camera calibration, image segmentation, image registration, object recognition, as well as the forward and inverse kinematics for robot control. The focus is placed on improving the robustness and accuracy of the robot system.

3.1 Camera Calibration

Camera calibration is the process of determining the intrinsic parameters and/or the extrinsic parameters of the camera. This is a crucial process for: (1) determining the location of the object or scene; (2) determining the location and orientation of the camera; and (3) 3D reconstruction of the object or scene (Tsai 1987).

A camera has two sets of parameters, intrinsic parameters which describe the internal properties of the camera, and extrinsic parameters which describe the location and orientation of the camera with respect to some coordinate system. The camera model utilised is a pinhole camera model and is used to relate 3D world points to 2D image projections. While this is not a perfect model of cameras used in machine vision systems, it gives a very good approximation and when lens distortion is taken into account, the model is sufficient for the most common machine vision applications. A pinhole camera is modelled by:

$$x = K[R \mid t]X \tag{1}$$

Where X is the 3D point coordinates and x the image projection of X. (R,t) are the extrinsic parameters where R is the 3×3 rotation matrix and t the 3×1 translation vector. K is the camera intrinsic matrix, or simply the camera matrix, and it describes the intrinsic parameters of cameras:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where (f_x, f_y) are the focal lengths and (c_x, c_y) the coordinates of the principal point along the major axes x and y, respectively. The principal point is the point at which the light that passes through the image is perpendicular to the image, and this is often, but not always, at the centre of an image.

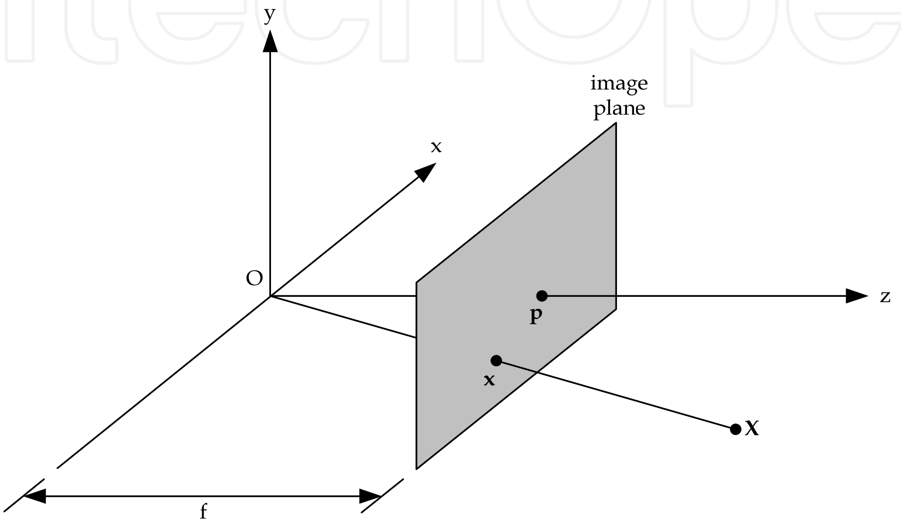


Figure 2. Pinhole camera showing: camera centre (O), principal point (p), focal length (f), 3D point (X) and its image projection (x)

As mentioned previously, lens distortion needs to be taken into account for pinhole camera models. Two types of distortion exist, radial and tangential. An infinite series is required to model the two types of distortions, however, it has been shown that tangential distortions can often be ignored, in particular for machine vision application. It is often best to limit the number of terms for the distortion coefficient for radial distortion for stability reasons (Tsai 1987). Below is an example of how to model lens distortion, taking into account both tangential and radial distortions, using two distortion coefficients for both:

$$\tilde{x} = x + x\left[k_1 r^2 + k_2 r^4\right] + \left[2p_1 xy + p_2\left(r^2 + 2x^2\right)\right] \tag{3}$$

$$\tilde{y} = y + y\left[k_1 r^2 + k_2 r^4\right] + \left[2p_1 xy + p_2\left(r^2 + 2y^2\right)\right] \tag{4}$$

$$r = \sqrt{x^2 + y^2} \tag{5}$$

where (x, y) and $(\tilde{x}, \tilde{y})$ are the ideal (without distortion) and real (distorted) image physical coordinates, respectively. r is the distorted radius.

3.2 Object Segmentation

Image segmentation is the first step leading to image analysis. In evaluation of the five methodologies (Fu and Mui 1981), the deformable model based segmentation holds a distinct advantage over the others when dealing with image-guided robotic systems. Many visually-guided systems use feature based approaches for image registration. Such systems may have difficulty handling cases in which object features become occluded or deformation alters the feature beyond recognition. For instance, systems that define a feature as a template of pixels can fail when a feature rotates relative to the template used to match it. To overcome these difficulties, the vision system presented in this chapter incorporates contour tracking techniques as an added input along with the feature based registration to provide better information to the vision-guided robot control system. Hence, when a contour corresponding to the object boundary is extracted from the image, it provides information about the object location in the environment. If prior information about the set of objects that may appear in the environment is available to the system, the contour is used to recognise the object or to determine its distance from the camera. If additional, prior information about object shape and size will be combined with the contour information, the system could be extended to respond to object rotations and changes in depth.

3.2.1 Active Contours Segmentation

The active contours segmentation methodology implemented is a variation of the one proposed by (Kass, Witkin et al. 1988).

The implemented methodology begins by defining a contour parameterized by arc length *s* as

$$C(s) \equiv \{(x(s), y(s)) : 0 \le s \le L\} : \Re \to \Omega \tag{6}$$

where *L* denotes the length of the contour *C*, and Ω denotes the entire domain of an image *I(x, y)*. The corresponding expression in a discrete domain approximates the continuous expression as

$$C(s) \approx C(n) = \{(x(s), y(s)) : 0 \le n \le N, s = 0 + n\Delta s\} \tag{7}$$

where $L = N\Delta s$. An energy function *E(C)* can be defined on the contour such as

$$E(C) = E_{\text{int}} + E_{\text{ext}} \tag{8}$$

where E_{int} and E_{ext} respectively denote the internal energy and external energy functions. The internal energy function determines the regularity and smooth shape, of the contour. The implemented choice for the internal energy is a quadratic functional given by

$$E_{\text{int}} = \sum_{n=0}^{N} \left(a|C'(n)|^2 + \beta |C''(n)|^2 \right) \Delta s \tag{9}$$

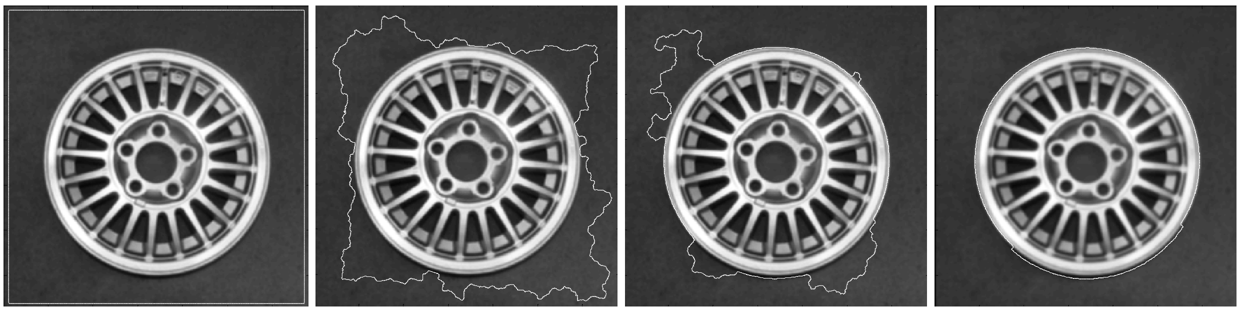
Here a controls the tension of the contour, and β controls the rigidity of the contour. The external energy term determines the criteria of contour evolution depending on the image *I(x, y)*, and can be defined as

$$E_{ext} = \sum_{n=0}^{N} E_{img}(c(n)) \Delta s \tag{10}$$

where E_{img} denotes a scalar function defined on the image plane, so the local minimum of E_{img} attracts the active contour to the edges. An implemented edge attraction function is a function of image gradient, given by

$$E_{\mathrm{img}}(x,y) = \frac{1}{\lambda \left| \nabla G_{\sigma} \times I(x,y) \right|} \tag{11}$$

where G_{σ} denotes a Gaussian smoothing filter with the standard deviation σ, and λ is a suitably chosen constant. Solving the problem of active contours is to find the contour C that minimises the total energy term E with the given set of weights a, β and λ. The contour points residing on the image plane are defined in the initial stage, and then the next position of those snake points are determined by the local minimum E. The connected form of those points are considered as the contour to proceed with. Figure 3 shows an example of the above method in operation over a series of iterations.



(a) (b) (c) (d)

Figure 3. (a) Original level set contour, (b) Level set contour after 20 iterations, (c) Level set contour after 50 iterations, (d) Level set contour after 70 iterations

3.3 Image Registration and Tracking

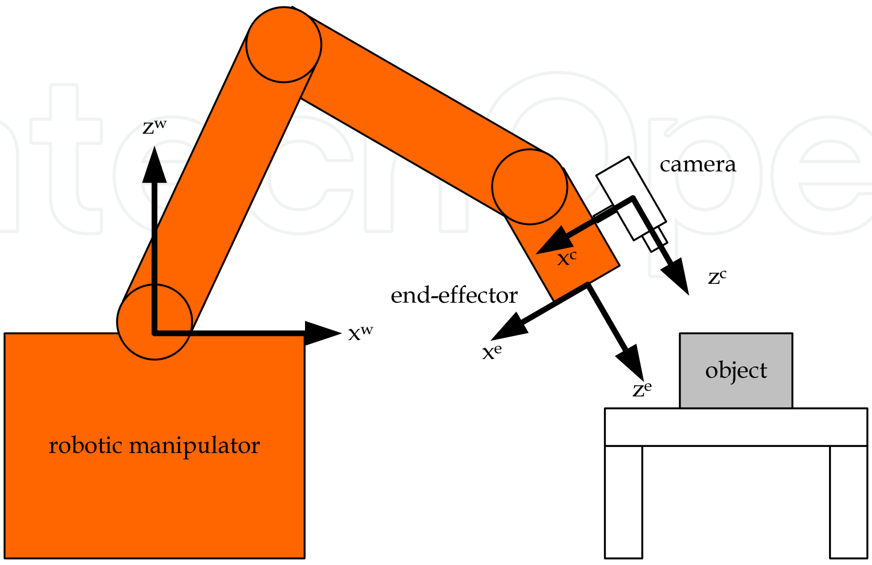


Figure 4. Eye-in-hand vision system

An important role of machine vision in robotic manipulators is the ability to provide feedback to the robot controller by analysing the environment around the manipulator within the workspace. Thus a vision-guided robotic system is more suitable for tasks such as grasping or aligning objects in the workspace compared to conventional feedback controls, for example force feedback controllers. The main challenge in vision-based robot control systems is to extract a set of robust and useful feature points or regions and using these features to control the motion of the robotic manipulator in real-time.

To accurately provide feedback information from the vision system, the camera first needs to be aligned with the robotic manipulator. Figure 4 shows an eye-in-hand configuration (Spong, Hutchinson et al. 2006), which has the camera mounted on the end-effector. The 'eye-in-hand' setup is often the preferred configuration in machine vision applications since it provides a better view of the object. The base of the robotic manipulator contains the world coordinate system(x^w, y^w, z^w), which is used as the reference for all the other coordinate systems. The end-effector's coordinate system (x^e, y^e, z^e) has a known transformation with respect to the world coordinate. The camera's coordinate system is aligned to the end-effector by a number of methods, one of which is based on the geometric relationship between the origin of the end-effector and the camera coordinate. By knowing the geometric relationship of the end-effector and the camera, the rigid transformation or homogeneous transformation of the two components can be defined:

$$P^{camera} = H^{camera}_{end\,effector} P^{end\,effector} \tag{12}$$

where P^{camera} is the homogeneous representation of a 3D point in the camera coordinate,

$$P^{camera} = \begin{bmatrix} p^{camera} \\ 1 \end{bmatrix} \tag{13}$$

Using the homogeneous transformation, the information obtained by the vision system can be easily converted to suit the robotic manipulator's needs. In order for the vision system to track an object in the workspace, either continuous or discontinuous images can be captured. Continuous images (videos) provide more information about the workspace using techniques such as optical flow (Lucas and Kanade 1981), however require significantly more computation power in some applications, which is undesirable. Discontinuous images provide less information but can be more difficult to track an object, especially if both the object and the robotic manipulator are moving. However, the main advantage is that they require less computation as images are not being constantly compared and objects tracked. One method for tracking an object is by using image registration techniques, which aims at identifying the same features of an object from different images taken at different times and viewpoints.

Image registration techniques can be divided into two types of approaches, area-based and feature-based (Zitova and Flusser 2003). Area-based methods compare two images by directly comparing the pixel intensities of different regions in the image, while feature-based methods first extract a set of features (points, lines, or regions) from the images, the features are then compared. Area-based methods are often a good approach when there are no distinct features in the images, rendering feature-based methods useless as they cannot extract useful information from the images for registration. Feature-based methods on the other hand, are often faster since less information is used for comparison. Also, feature-based methods are more robust against viewpoint changes (Denavit and Hartenberg 1955; Hartley and Zisserman 2003), which is often experienced in vision-based robot control systems.

3.3.1 Feature-Based Registration Algorithms

To successfully register two images taken from different time frames, a number of feature-based image registration methods, in particular, those based on the local descriptor method have been studied. In order for the algorithms to be effective in registering images of an object over a period of time, two types of rigid transformations have been analysed: (1) change of scale and (2) change of viewpoint. A change of scale implies that the camera has moved towards or away from the camera, hence the size of the object in the image has changed. A change in viewpoint implies that the camera has moved around in the environment and that the object is being viewed from a different location. Four algorithms were compared, namely: Scaled-Invariant Feature Transform (SIFT) (Lowe 2004), Principal Component Analysis SIFT (PCA-SIFT) (Ke and Sukthankar 2004), Gradient Location and Orientation Histogram (GLOH) (Mikolajczyk and Schmid 2004), and Speeded Up Robust Features (SURF) (Bay, Tuytelaars et al. 2006). The scale of the image is defined as the ratio of the size of the scene in an image with respect to a reference image:

$$s = \frac{h_{I,s}}{h_{I,r}} \tag{14}$$

where $h_{I,r}$ is the height of the scene in the reference image, and $h_{I,s}$ is the height of the scene in the image of scale s . From this equation it is clear that the scale is proportional to the height of the scene in the image. This is best illustrated in Figure 5, where Figure 5(a) shows the image used as the reference and has a scale of one by definition, and Figure 5(b) shows a new image with scale s .

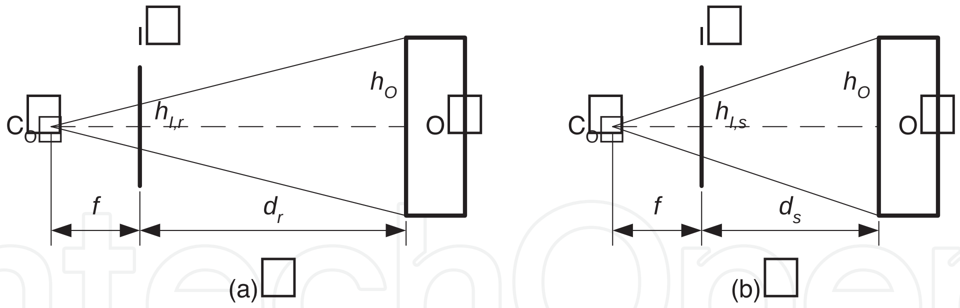


Figure 5. Change of scale of objects in images

3.4 Improved Registration and Tracking Using the Colour Information of Images

3.4.1 Colour model

One area of improvement over existing methods is the use of the information available from the images. Methods such as SIFT or SURF make use of greyscale images, however many images which need to be registered are in colour. By reducing a colour image to a greyscale one, a large proportion of information are effectively lost in the conversion process. To overcome this issue, it is proposed that colour images are used as inputs for the registration process, thus providing more unique information for the formation of descriptors, increasing the uniqueness of descriptors and enhancing the robustness of the registration process.

While it is possible to simply use the RGB values of the images, it is often not the desirable approach, since factors such as the viewing orientation and location of the light source affect these values. Many colour invariant models exist in an attempt to rectify this issue, and the $m_1m_2m_3$ model (Weijer and Schmid 2006) is utilised here. The advantage of the chosen model is that it does not require *a priori* information about the scene or object, as the model is illumination invariant, and is based on the ratio of surface albedos rather than the surface albedo itself. The model is a three component model, and can be defined as:

$$m_1 = \frac{R^{x_1} G^{x_2}}{R^{x_2} G^{x_1}},\ \ m_2 = \frac{R^{x_1} B^{x_2}}{R^{x_2} B^{x_1}},\ \ m_3 = \frac{G^{x_1} B^{x_2}}{G^{x_2} B^{x_1}} \tag{15}$$

where $\mathbf{x}_1, \mathbf{x}_2$ are the neighbouring pixels. Without loss of generality, m_1 is used to derive the results, by taking the logarithm on both sides:

$$\begin{aligned} \log\left(m_1\left(R^{\mathbf{x}_1} R^{\mathbf{x}_2} G^{\mathbf{x}_1} G^{\mathbf{x}_2}\right)\right) &= \log\left(\frac{R^{\mathbf{x}_1} G^{\mathbf{x}_2}}{R^{\mathbf{x}_2} G^{\mathbf{x}_1}}\right) \\ &= \log R^{\mathbf{x}_1} + \log G^{\mathbf{x}_2} - \log R^{\mathbf{x}_2} - \log G^{\mathbf{x}_1} \\ &= \log\left(\frac{R^{\mathbf{x}_1}}{G^{\mathbf{x}_1}}\right) - \log\left(\frac{R^{\mathbf{x}_2}}{G^{\mathbf{x}_2}}\right) \end{aligned} \tag{16}$$

From this, it can be shown that the colour ratios can be represented as the difference of the two neighbouring pixels:

$$d_{m_1}(\mathbf{x}_1, \mathbf{x}_2) = \left(log\left(\frac{R}{G}\right)\right)^{\mathbf{x}_1} - \left(log\left(\frac{R}{G}\right)\right)^{\mathbf{x}_2} \tag{17}$$

3.4.2 Dimension Reduction

It should be noted that one main issue which arises from using the colour information of images is the increase in the amount of data which needs to be dealt with. By computing descriptors for the model described above, the dimension of each descriptor is increased three-fold. For example, in the case of the SURF descriptor, the dimension of each descriptor increases from 64 to 192. While this increase in dimension often aids in improving the robustness, a significant drop in computational speed is noted. To overcome this issue, Principal Component Analysis (PCA) (Pearson 1991) is applied. PCA is a statistical method for vector space transform, often used to reduce data to lower dimensions for analysis. The original data is transformed as follows:

$$Y = UX \tag{18}$$

where Y is the new data, based on the original data X, and the eigenvectors of the covariant matrix of X, U. The covariance matrix can be computed by first mean-shifting the data, then estimating:

$$\operatorname{cov}(X) = \frac{1}{n-1} \sum_{i=1}^{n} \left(X_i - \overline{X}\right)\left(X_i - \overline{X}\right)^T \tag{19}$$

where, $\overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$ is the sample mean, and n is the number of data entries. Approximately 20,000 image patches have been used for estimating the covariance matrix. Here, the SURF descriptor has been used to generate a set of descriptors for each image patch.

3.5 Object Recognition

The algorithm for object recognition is one of the most important parts of the developed vision system. This module is used to make decisions for the object recognition process and command different parts of the system to function.

3.5.1 Algorithm for 2D Object Recognition

Recognising 2D objects is a relatively simple task, since only the top view of the objects need to be considered. One simple algorithm is described below. Firstly, the program commands the VisionServer, part of the vision system, to take an image of the object from the top and match it against the images of all the models in the database. The VisionServer will then compute a score for each model to indicate how close each model matches with the object, with 1.0 being the highest and 0.0 being the lowest. Preceding this, the program can identify what the object is by comparing the scores of the various models.

One simple methodology is to compare the scores with a threshold value. The model is reported as the object only if the score is above the threshold value. Sometimes there will be more than one model getting a score that is bigger than the threshold value due to the similarity of the models in the database. In this case, the model with the highest score will be reported as the object. On the other hand, if none of the scores is above the threshold value, the program will report an error. After the object is recognised, the next 2D object can be processed using the same procedure. Since this algorithm is simple and only one image needs to be checked against other trained images in the system at each time, problems such as long computational time and view planning, which are common in 3D object recognition systems, do not exist.

3.5.2 Algorithm for 3D Object Recognition

Three-dimensional object recognition is much more complex compared to 2D cases. This is because a 3D object can have many different faces and sitting bases. The appearance of the object can be very different from one viewpoint to another. Moreover, an object recognition system often needs to identify more than one object. If the camera is only fixed in one place to take images, it may not have enough information to identify an object if it looks similar to more than one model in the database. It may also report the wrong model name when cluster or occlusion exists. Therefore, taking multiple images from different angles is a common approach for 3D object recognition systems.

Consider the case where there are several 3D objects to be recognised and there is no restriction on the number of sitting bases each object can have. One approach to solving this problem is to obtain and train several views of each object that can be seen by the camera and store the results in the system. In run-time, if any one of the images in the database looks similar to the object being recognised, the system will report the name of the model that this image belongs to. Otherwise, it will go to another viewpoint to take a new image and repeat the same process until it finds a trained image that looks similar to the new view of the run-time object. Sometimes, there may be two or more images from different models which appear similar to the view of the object due to the similarity of the models. In this case, the system will need to take another image from a new position and repeat the process again until there is only one image in the database which matches the run-time image. Figure 6 shows the flowchart of this algorithm. This algorithm is easy to implement but it has many drawbacks. One of the major problems is the long computational time needed. If there is only one model in the database, this algorithm can perform reasonably fast. However, this is not a practical setup. If there are more models to be recognised or if the models have more complex shapes and more faces, there will be a lot more trained images in the database. For example, if there are n models to be identified and each has m faces, there will then be m×n trained images in the database of the system. Since there is no restriction on the number of sitting bases each object can have, the system has no idea which view of the object will face the camera. As a result, every time an image is taken, the system needs to compare the image with all the trained images in the database. This process is extremely time consuming and occupies a lot of computer resource. This problem is magnified when there are similar models in the database or when the system fails to find a trained image which looks similar to the run-time image because, as shown in the flowchart in Figure 6, the system will simply repeat the same process until it finds one model that matches. Another problem in this algorithm is if an object that has not been trained is put in the system for recognising, then the system will simply go into an endless loop and never finish since it can never find a model that matches the image of the new object. These disadvantages make it impractical to be used in real life applications and therefore, an improved method is derived.

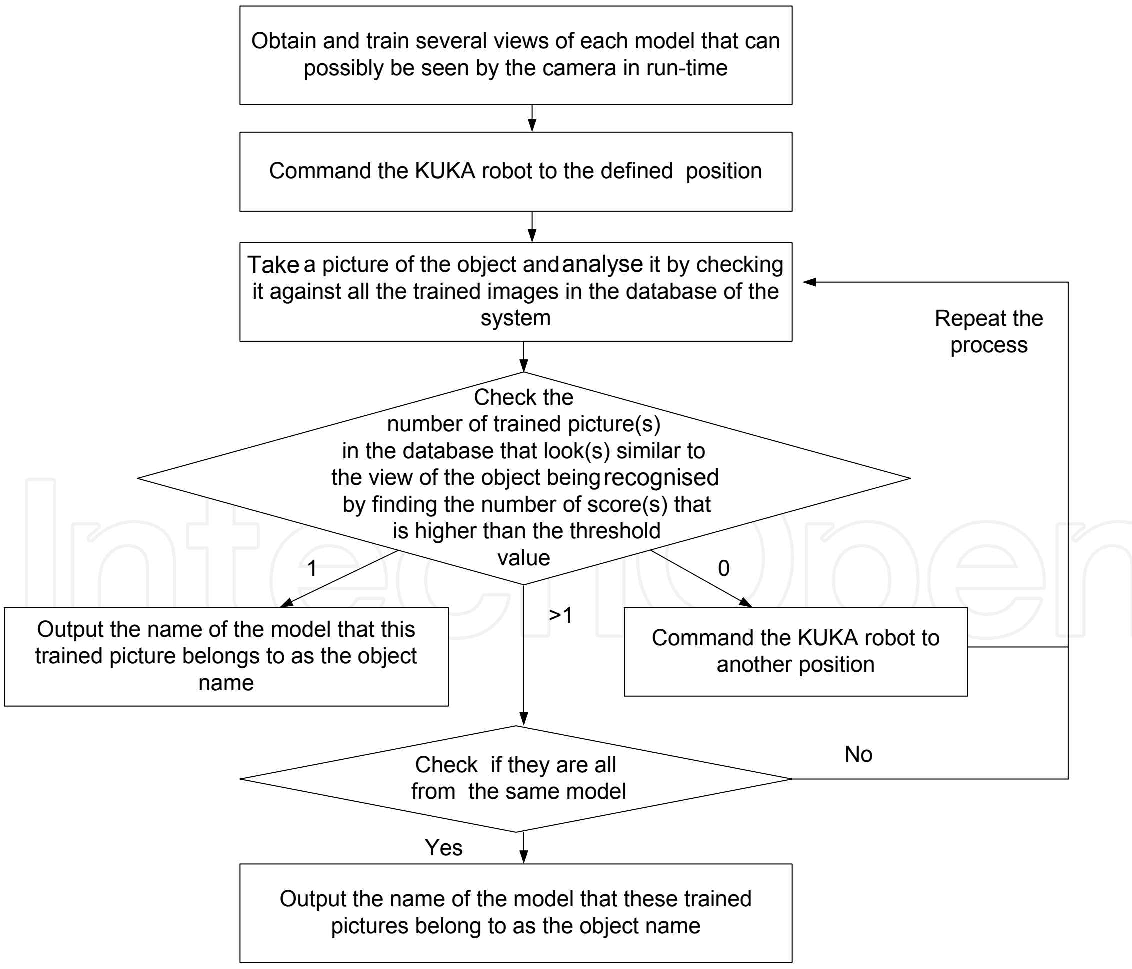

Obtain and train several views of each model that can possibly be seen by the camera in run-time
Command the KUKA robot to the defined position
Take a picture of the object and analyse it by checking it against all the trained images in the database of the system
Repeat the process
Check the number of trained picture(s) in the database that look(s) similar to the view of the object being recognised by finding the number of score(s) that is higher than the threshold value
1
0
>1
Output the name of the model that this trained picture belongs to as the object name
Command the KUKA robot to another position
Check if they are all from the same model
No
Yes
Output the name of the model that these trained pictures belong to as the object name

Figure 6. Flowchart of a simple algorithm for 3D object recognition

3.5.3 Improved Algorithm for 3D Object Recognition

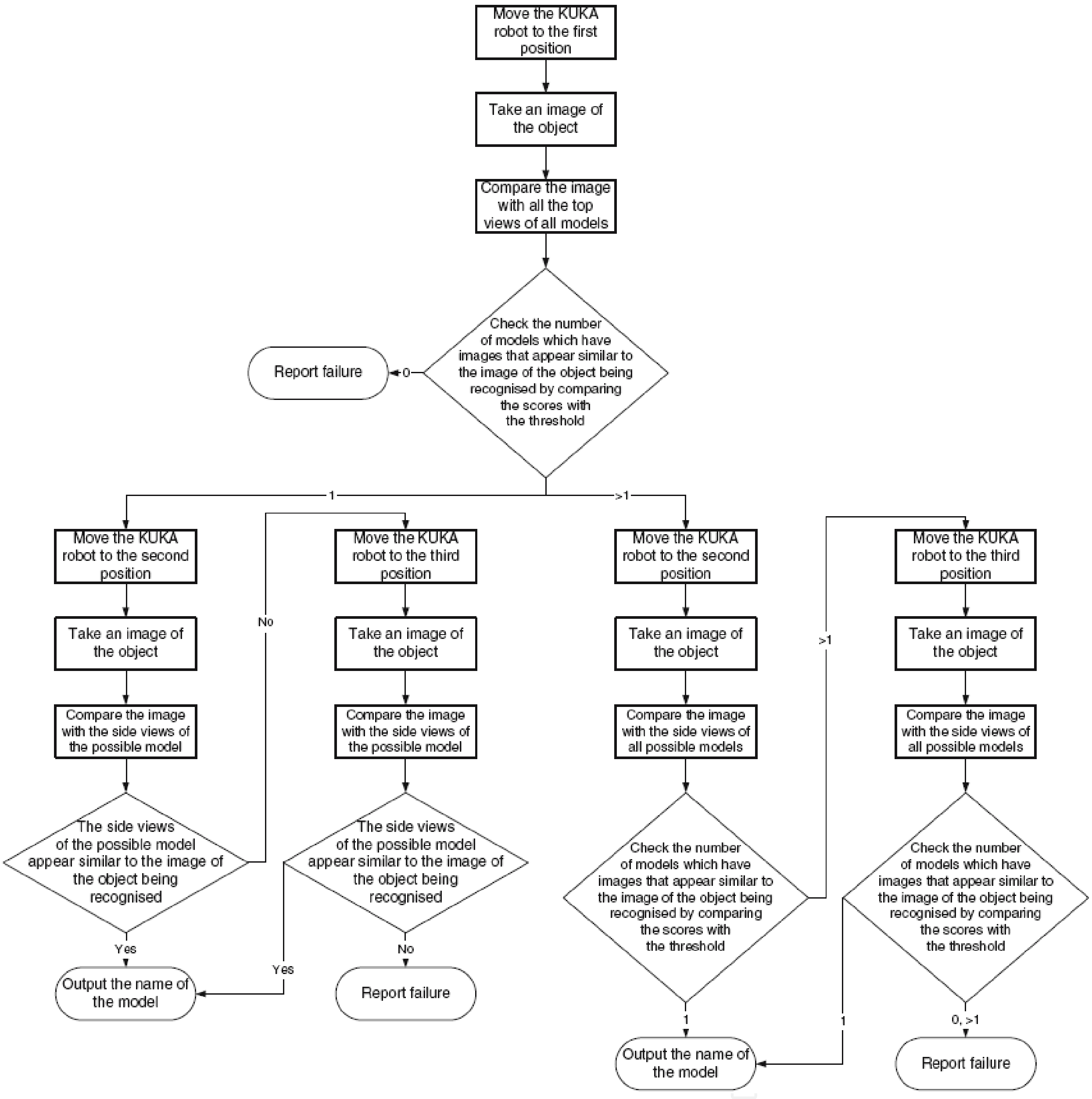

Move the KUKA robot to the first position
Take an image of the object
Compare the image with all the top views of all models
Check the number of models which have images that appear similar to the image of the object being recognised by comparing the scores with the threshold
0
Report failure
1
>1
Move the KUKA robot to the second position
Take an image of the object
Compare the image with the side views of the possible model
The side views of the possible model appear similar to the image of the object being recognised
Yes
Output the name of the model
No
Move the KUKA robot to the third position
Take an image of the object
Compare the image with the side views of the possible model
The side views of the possible model appear similar to the image of the object being recognised
Yes
No
Report failure
Move the KUKA robot to the second position
Take an image of the object
Compare the image with the side views of all possible models
Check the number of models which have images that appear similar to the image of the object being recognised by comparing the scores with the threshold
1
Output the name of the model
>1
Move the KUKA robot to the third position
Take an image of the object
Compare the image with the side views of all possible models
Check the number of models which have images that appear similar to the image of the object being recognised by comparing the scores with the threshold
1
0, >1
Report failure

Figure 7. Flowchart of an improved algorithm for 3D recognition

To improve the above algorithm, one method which is used is to restrict all the 3D objects to one sitting base only. By placing this restriction, any 3D object will always have the same face showing up no matter what pose it is placed under the camera. This means that every time a new object recognition process starts, the robot can always go to the top (position 1) to take an image of the object and compare it with the top views of all other models in the database. After the images have been compared, the system will have an idea of which model the object being processed is likely to be by comparing the scores. It outputs what object it is based on the top view. If the system is uncertain what the object is, it goes to the side positions (either position 2 or 3) to check for the object's side views to further verify whether or not it is the one that the system initially determines before the system outputs the result. By allowing the system to output the result based on the top view, it can make the process performs faster, however, it also decreases the accuracy. On the other hand, if the system checks for at least two views of the object before it outputs the result, the system will perform slower, but will have a much higher accuracy. There is a trade-off between speed and accuracy and this decision should be made based on the situation. In this work, it was chosen to check for two views in for 3D object recognition to enhance the accuracy of the system.

Using the proposed approach, if the run-time object looks similar to two or more models from the top, the ASEA robot will analyse the side view of the run-time object from position 2 by checking it against all the side views of all the possible models. If the system can identify which model the object is, it will stop and output the result, otherwise, it will go to position 3 and repeat the same process. If the system still cannot determine what this object is, it will output a failure message and stop the object recognition process at this point. This is to prevent the system from running for a long period of time to handle complex or new objects.

This improved algorithm is fast and accurate compared with the simple 3D object recognition algorithm described in Figure 6. For every object recognition process, only a small portion of the trained images in the database needs to be looked at. This minimises the time required and the workload of the vision computer. In addition, when there is not sufficient information to identify the object due to the presence of similar models in the database, the system will automatically go to a new position to take another image of the object and perform the analysis again. If the system still cannot identify the object after three runs, it will stop and report a failure. This algorithm can also be used in cases where objects have two or more sitting bases. This is achieved by treating the object sitting on the other base as a separate object.

3.6 ASEA Robot Modelling and Simulation

The control of the ASEA robot directly affects the performance of the system. The processing time is one of the key factors to determining the performance. For example, if the robot moves too slowly, it will reduce its efficiency. However if the robot moves too fast, vibrations will affect the quality of the images taken. Therefore, the control of the ASEA robot has to be carefully considered.

One of the problems is to find a method to control the end-effector of the robot manipulator to follow a pre-defined trajectory. Normally a desired pose of the end-effector with respect to the base is given as well as the torque required to move a particular object. For serial robots, inverse kinematics algorithm is used to determine the desired rotational angle of each joint, given the desire position of end-effector. The corresponding torque of each motor needs be computed as well using inverse kinematics. In the kinematics model, the torques provided by the motors are the inputs into the plant, and the outputs are the resulting rotational angles and the angular velocities of the links. The operation is accomplished by actuating the five joint motors, by sensing the joint angles and angular velocities, and by comparing the desired angular positions with the actual ones. Figure 8 below shows a simulation model established to validate the control and kinematics of the ASEA robot.

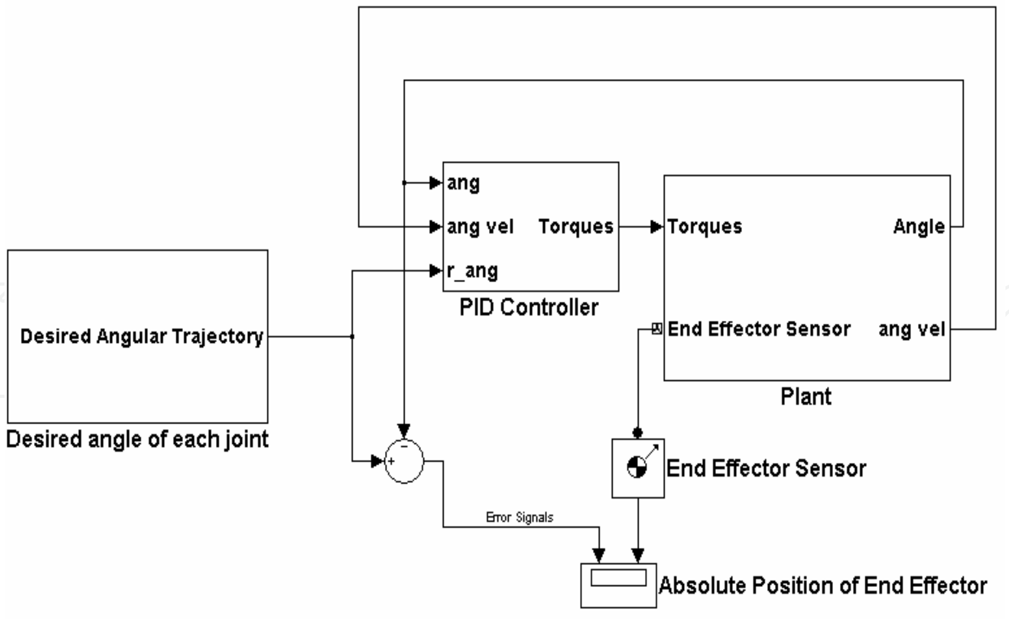


Figure 8. Overall control system modelled in Simulink

3.6.1 Kinematics Analysis

The ASEA robot manipulator has five serial links, which allows the robot to move with five degrees of freedom (DOF). The dimension of the robot manipulator is shown in Table 1. L1-L5 represent the five links that make up the serial robot while q1-q5 represent the five joints that form the serial structure. For the serial robot, a new inverse kinematics algorithm is developed using a combination of the numerical and analytic methods. The forward kinematics was implemented in combination with inverse kinematics in order to minimise the error of inverse kinematics solution. The basic principle is that the solution of inverse kinematics is fed to forward kinematics so that the desired position derived from calculated joint angles are computed. The difference between real desired position and calculated desired position is then calculated, which is added to inverse kinematics function. By repeating the loop a few times, the solution of inverse kinematics should be optimal.

Link	Length(mm)	Joint	Type of Rotation	Angle of Rotation(deg)
L1	730	q1	axis	360
L2	690	q2	vector	82.5
L3	670	q3	vector	63
L4	73	q4	vector	208.5
L5	20	q5	axis	331.5

Table 1. The ASEA robot parameters

3.6.2 Forward Kinematics

The end-effector position and orientation can be solved link by link from the base to the end-effector. If x, y, z, u and w are defined as instantaneous cartesian coordinates then the end-effector orientation can be solved using the following relations,

$$\begin{aligned}
x &= \cos(q_1) \times (L_2 \times \cos(q_2) + L_3 \times \cos(q_2 + q_3) + (L_4 + L_5) \times \cos(q_2 + q_3 + q_4)) \\
y &= \sin(q_1) \times (L_2 \times \cos(q_2) + L_3 \times \cos(q_2 + q_3) + (L_4 + L_5) \times \cos(q_2 + q_3 + q_4)) \\
z &= L_2 \times \sin(q_2) + L_3 \times \sin(q_2 + q_3) + L_1 + (L_4 + L_5) \times \sin(q_2 + q_3 + q_4)) \\
u &= \cos(q_2 + q_3 + q_4) \\
w &= \sin(q_2 + q_3 + q_4)
\end{aligned} \tag{20}$$

3.6.3 Inverse Kinematics: Sugeno Fuzzy Inference System (SFIS)

To achieve a mapping between the cartesian variables and joint variables of the ASEA robot, a zero-order Sugeno Fuzzy Inference System (SFIS) was developed. The Sugeno system is based on the concepts of fuzzy set theory, fuzzy if-then rules, and fuzzy reasoning. Essentially the fuzzy system has three main components: a rule-base, which contains fuzzy if-then rules; a database defining input and output fuzzy membership functions (MF) used in the rule-base and an inferencing mechanism to perform fuzzy reasoning. While carrying out the inverse kinematic analysis it is required to find a set of joint variables (q_1, q_2,......q_5) to achieve the specified end effector pose of the robot. The cartesian variables of the end effector are chosen to be fuzzy antecedent variables and each variable is represented by two gaussian fuzzy MF with linguistic terms such as Low and High. Similarly, constant functions are intuitively chosen for the output joint variables and a rule-base containing a total of 32 rules is generated based on the following rule (Jamwal and Shan 2005):

$$R = \prod_{j=1}^{n} m_j \tag{21}$$

where R represents the total number of rules**;** m_j is the number of linguistic terms of the j^{th} linguistic variable and n is the total number of input variables. Thus in the present case the total number of rules becomes 2^5 or 32. A Sugeno inferencing (Jang, Sun et al. 1997) is selected due to its advantages over the Mamdani method. The rule-base defines the relationship between inputs and outputs and a typical Sugeno fuzzy rule has the following structure:

$$If \quad \begin{matrix} x \\ y \\ z \\ u \\ w \end{matrix} \quad are \quad \left[Low \; High \right] \quad Then \quad \begin{matrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{matrix} \quad = \quad \begin{matrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{matrix} \tag{22}$$

where x, y, z, u and w are instantaneous input cartesian coordinates and r_1, r_2, r_3, r_4 and r_5 are the crisp output values for the joint variables of ASEA robot. The fuzzy system so developed is not accurate (Jang, Sun et al. 1997) until the fuzzy parameters of antecedent and consequent variables are adjusted or tuned.

3.6.3.1 Consequent MF tuning

Once a database containing a set of inputs and outputs is available from the forward kinematic analysis, the consequent MFs are tuned to produce the desired model accuracy. To achieve this, a gradient-descent method is used to minimise an error function. Outputs from the SFIS are compared from the corresponding values available from forward kinematic analysis to find the error. The learning rule for the consequent crisp MF (r_i's) can be expressed as below:

$$r_i(t+1) = r_i(t) \quad \alpha \frac{\partial E}{\partial_i} \tag{23}$$

where E is Sum of Squared Errors (SSE) and α is the learning rate which decides the quantum of change in the parameters after each iteration.

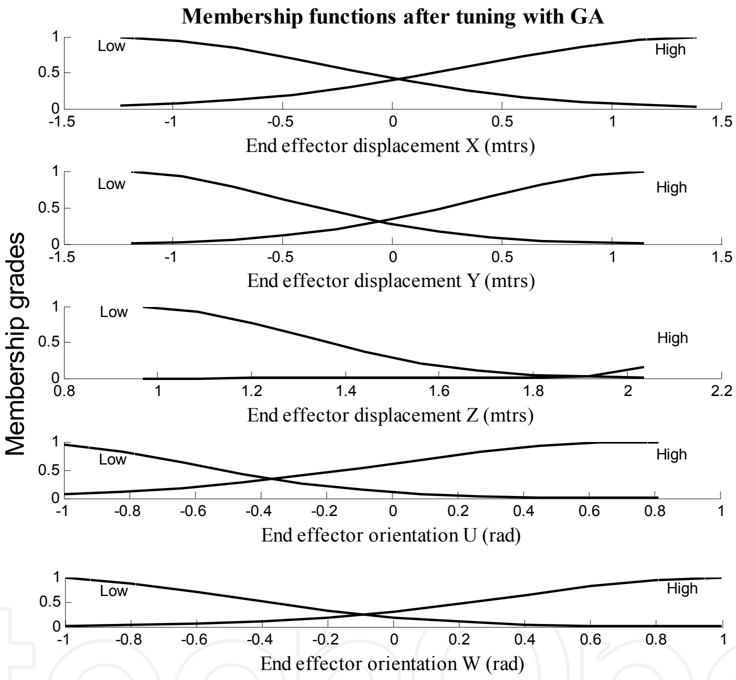


Figure 9. Antecedent membership functions after tuning with genetic algorithms

3.6.3.2 Antecedent MF tuning

Initially the antecedent variables (cartesian coordinates) have been fuzzified and two gaussian MF linguistically termed as Low (L) and High (H) are placed equally dividing their respective universe of discourses. The position of the point of minimum fuzziness or the mean of the gaussian MF and its standard deviation are the important parameters, which influence accurate inferencing. There exists an optimal set of mean and standard deviation of all the membership functions for which the fuzzy model shall be given a maximum accuracy. In the present application these values of MF are varied within ±33% of their values. For example, mean of the 'Low' membership function of end effector displacement 'X' has been varied between -1.7373 and -0.8686 metres and the standard deviation is varied between 0.66 and 1.33. Thus for five input variables consisting two MF each, there are 20 points of mean and standard deviation which require tuning. In the present work genetic algorithms (GA) have been used (Jamwal and Shan 2005) for this tuning and to represent 20 tuning points a binary string of 200 bits is used, wherein 10 bits are devoted to each tuning point. Initial population of 20 binary coded strings of 200 bits was generated using Knuth's random number generator. The roulette wheel selection method (Jang, Sun et al. 1997) is used for selection of binary strings in the mating pool from the initial population and the probabilities of crossover and mutation operators were kept as 0.95 and 0.01. A Matlab code has been written to implement a genetic algorithm to tune the fuzzy system.

3.6.4 Simulation Model

A simulation model is established using SimMechanics to validate the developed control and kinematic algorithms. The model is shown in Figure 10. The model directly communicates with the control algorithms developed in Simulink, hence the control algorithms are firstly tested before they are implemented. The simulation model also provides visual feedback on how the robot tracks predefined trajectories.

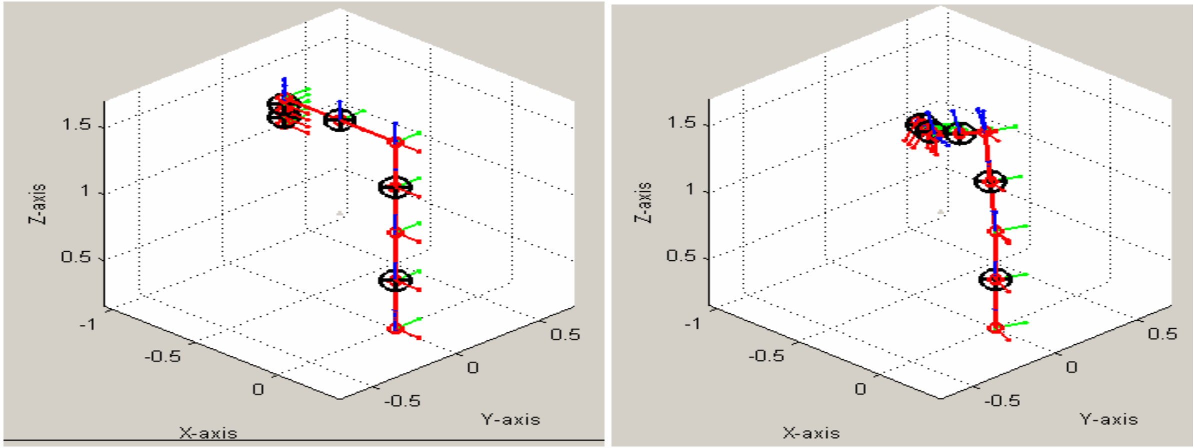


Figure 10. Simulation results of the simple integrated control system – starting point of the robot (left) and final position of the robot (right)

4. Results

In this section the results that have been achieved to date are presented. These are from the vision system, modelling and simulation of the robot, and the test results of the vision-guided robot.

4.1 Experimental Setup

The experimental setup is shown in Figure 11. The experiment consists of an ASEA robot and a small scale assembly line setup as shown. The ASEA robot is shown in its default position and carries a CCD camera on its end-effector for visual feedback. The assembly line runs at a constant speed, and carries a number of objects (car wheels). The experiment is set up to automatically inspect car wheels.

In the robot control system such as this, it is important to keep track of the camera poses for taking images. An appropriate viewpoint planning algorithm is required to decide the camera viewpoint to get the next best view of an object, if the first image does not provide sufficient information to identify what the object is. This prevents the camera from taking images from an inadequate viewpoint. In this research, the ASEA robot was only allowed to go to one of three pre-defined viewpoints to capture images. Three fixed viewpoints exist, one on top and two on the sides of the object. There are several software packages used in this system. Controlling the operations of the whole process, enabling communication between different parts and carrying out object recognition tasks are the main roles of the developed software system. The software used to facilitate the construction of the system include the VisionPro software package, LabVIEW Programming Language, and the ASEA Robot Programming Language. VisionPro is a very comprehensive software package for machine vision applications. It contains a collection of Microsoft's Component Object Model (COM) components and ActiveX controls that allow the development of vision applications in Visual Basic and Visual C++ programming environments with great flexibility.

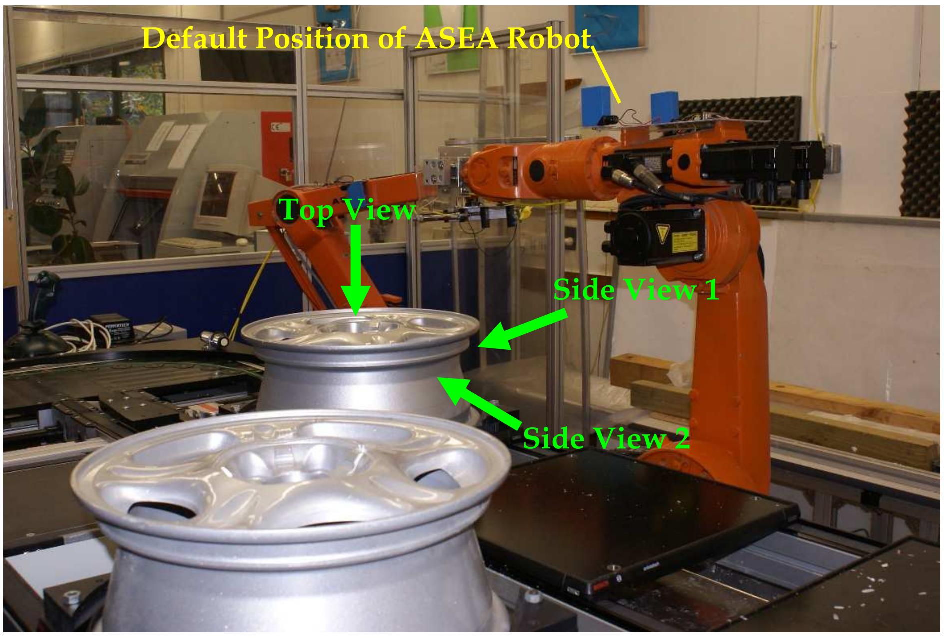


Figure 11. Experiment setup

4.2 Results: Feature Identification

The performance of the local descriptors are evaluated by using the criterion described in (Mikolajczyk and Schmid) which is based on the number of correct matches and number of false matches obtained for any image pair. A plot of the recall versus the 1-precision is computed for each image pair. In addition, the recall value for each image pair is computed and this is plotted against the viewpoint changes. Recall is a measure of how well the local descriptors performed, based on the ratio of the number of correct matches and the total number of corresponding regions, determined with the overlap error (Kadir, Zisserman et al. 2004):

$$\text{recall} = \frac{\#\text{ of correct matches}}{\#\text{ of correspondences}} \tag{24}$$

An overlap error of 50% is used. The 1-precision is a measure of the accuracy of the local descriptors:

$$1 \text{ - precision} = \frac{\#\text{ of false matches}}{\#\text{ of matches}} \tag{25}$$

The performance of the descriptors for the two different transformation types are discussed below.

4.2.1 Scale Changes

The recall versus 1-precision plot for scale changes is shown in Figure 12(a). This curve is near straight, indicating that the performance of the descriptors do not change with the change in scale (Mikolajczyk and Schmid). Figure 12(b) shows the recall plot for scale changes. The first section of the plot shows the performance is not affected by the change in scale. The performance however decreases rapidly for scale values greater than two. Further study of the local descriptors suggests that the drop in the recall value is due to the loss of information in the images due to the images being taken from further distances away from the objects.

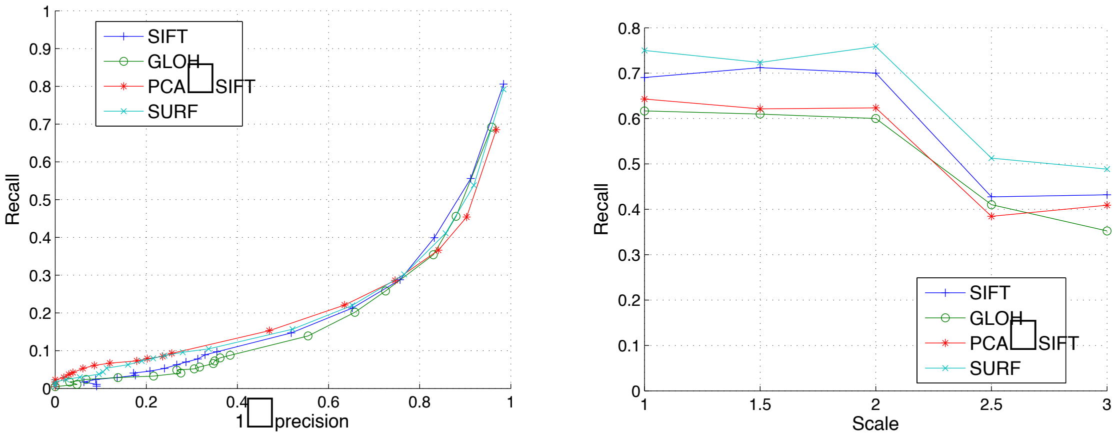


Figure 12. Recall v.s. 1-precision for a scale change of two (a) and recall for different scales (b)

The good performance for scale change is in agreement with previous research (Mikolajczyk and Schmid 2004) which indicates that the local descriptors are relatively robust to the above-mentioned transformations regardless of the type of scene used. As the descriptors are normalised for scale, good results for the scale changes are expected and this is true for scales of up to two for the carved flute. The poor performance of the techniques for scales of higher values is possibly caused by the removal of the background in all images and the scene of interest not covering the entire reference image. Therefore less information from the images is available compared to the evaluation studies in previous works (Mikolajczyk and Schmid 2004).

4.2.2 Viewpoint Changes

Viewpoint changes often pose the greatest challenge for image registration and this is reflected in the results obtained. Figure 13 shows the recall plotted against the 1-precision for different viewpoint changes. A slowly increasing curve in Figure 13(a) indicates that the performance of the local descriptors is influenced by the degradation of images, in particular viewpoint changes. The near-horizontal curve in Figure 13(b) indicates that the performance of the local descriptors is limited by the similarity of the structures of the scenes, and the descriptors cannot distinguish between the structures (Mikolajczyk and Schmid 2004).

Figure 14 shows the best recall values for the different viewpoint changes. The best recall value is defined as the value which has the maximum number of correct matches for the given number of correspondences, obtained by using different threshold values for identifying a pair of descriptors. As can be shown, the recall values degrade rapidly as the viewpoint angle increases, indicating the poor performance of the descriptors for dealing with viewpoint changes when using complex 3D scenes or scenes without distinctive features. Note that the recall value for a zero viewpoint angle is computed from two images of the scene taken from very close, but not identical, viewpoints due to the errors which exist in the turntable. The reason for a relatively low recall value despite the small difference in viewpoints is due to the way descriptors are matched. For low matching threshold values, not all the descriptors from the same region in the scene can be matched correctly, however for high threshold values mismatches occur, which decreases the recall value.

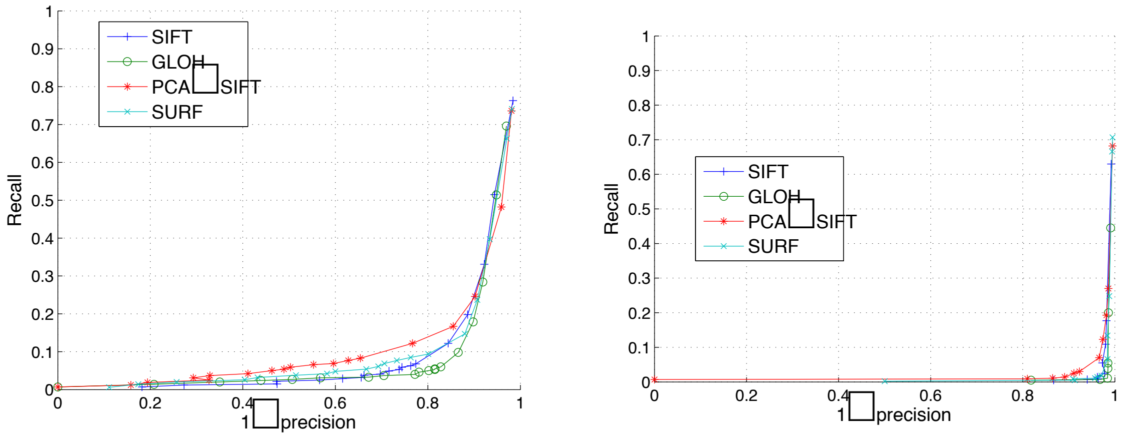


Figure 13. Recall v.s. 1-precision for a 5° viewpoint change (a), and 10° viewpoint change (b)

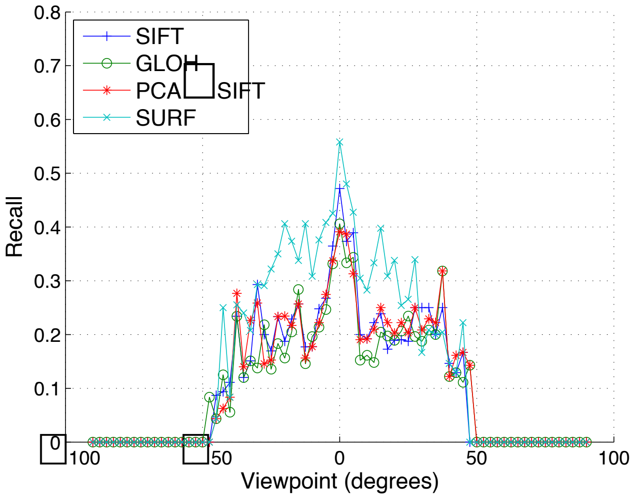
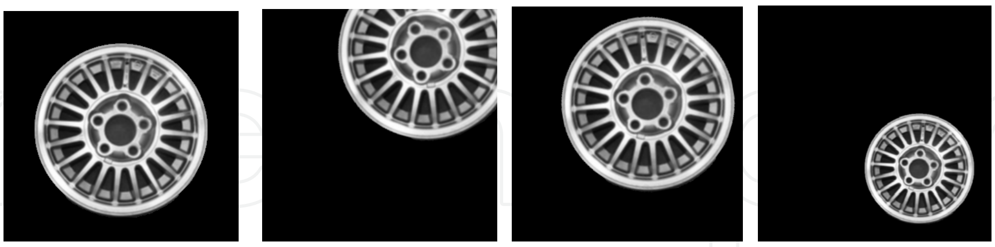


Figure 14. Recall values for different viewpoint changes

4.3 Results: Improved Feature Identification Using Colour Information of Images

The performances of several of the local descriptors were once again evaluated to quantify the benefit of utilising colour information during the feature identification process. The top two images shown in Figure 15 were taken from the active contours segmented tyre images of Figure 3.



(a) (b) (c) (d)

Figure 15. Images used for comparison of the performance of the improved feature identification approach. (a) is the original control image while (b), (c) and (d) have been rotated, translated and scaled respectively. The results of the registration process between (a) and (b) are given below

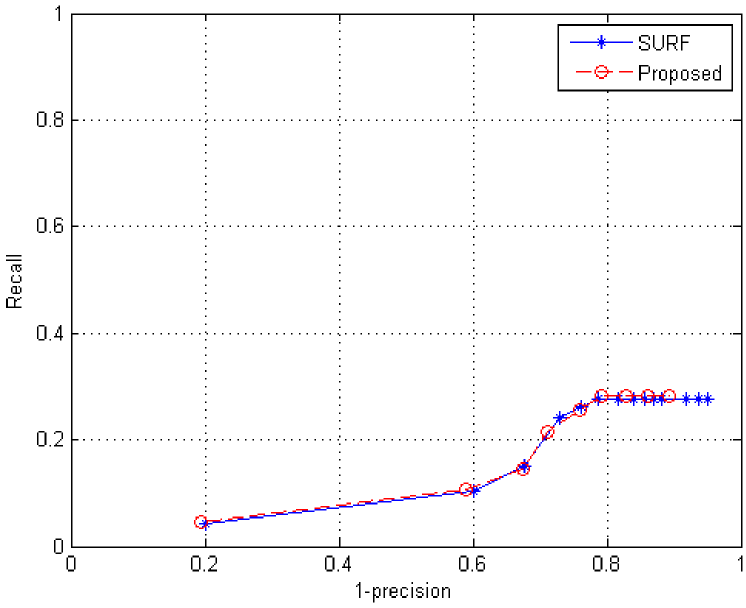


Figure 16. Results for the improved approach. The graphs show the recall against 1-precision values for the tyre images

Once again the recall versus 1-precision plot is used, which were discussed previously. As it can be shown in Figure 16 the improved descriptor performs slightly better than the SURF descriptors, indicating the robustness of the algorithm against changes in the image conditions (for example, illumination). This is due to the lack of sufficient colour information in the test images. Significant improvements are observed when colour images are used.

4.4 Results: SFIS

The inverse kinematics SFIS system was tuned using genetic algorithm for 50 cycles and the average sum of squared errors in prediction of joint variables was dropped from 2.339 to

0.000193 for a set of 100 random end-effector poses. The SFIS, after tuning of antecedent and consequent variables is now accurate enough and is available for implementation in SIMULINK for simulations and online use. In order to check the system accuracy, 50 random sets of Cartesian variables were given to the system as inputs and the results obtained were plotted in Figure 17. It can be observed that the sum of squared errors in joint variables is very small and falls in the range of $0-2\times10^{-3}$ radians. Further, the inference system so developed can be used for online control application, since it is time efficient and can provide results in less than 0.8 milliseconds.

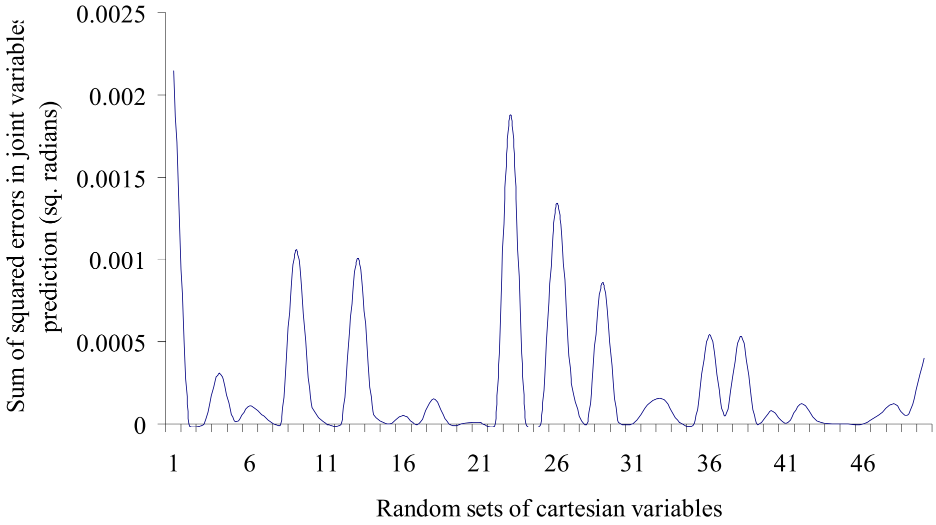


Figure 17. Sum of squared errors in joint variables for random sets of cartesian variables

5. Conclusions

Research into a fully automated vision-guided robot for identifying, visualising and manipulating 3D objects with complicated shapes is still undergoing major development world wide. The current trend is toward the development of more robust, intelligent and flexible vision-guided robot systems to operate in highly dynamic environments.

The theoretical basis of image plane dynamics and robust image-based robot systems capable of manipulating moving objects still need further research. Research carried out in our group has been focused on the development of more robust image processing methods and adaptive control of robot. Our research has been focused on manufacturing automation and medical surgery (Graham, Xie et al. 2007), with the main issues being the visual feature tracking, object recognition and adaptive control of the robot that requires both position and force feedback.

The developed vision-guided robot system is able to recognise general 2D and 3D objects using a CCD camera and the VisionPro software package. Object recognition algorithms were developed to improve the performance of the system and a user interface has been successfully developed to allow easy operation by users. Intensive testing was performed to verify the performance of the system and the results show that the system is able to recognise 3D objects and the restructured model is precise for controlling the robot. The kinematics model of the robot has also been validated together with the robot control algorithms for the control of the robot.

6. Future Research

Future work includes a number of improvements which will enhance the robustness and efficiency of the system. The first is the use of affine invariant methods for 3D object recognition. This method utilises local descriptors for feature matching and it has been shown to have higher success and accuracy rates for complex objects, cluster and occlusion problems (Bay, Tuytelaars et al. 2006). An automatic threshold selection process can also be implemented. This will eliminate the need to manually select a threshold for determining if an object is correctly recognised or not.

The current threshold value was determined empirically and is not guaranteed to work in every possible case. The system needs to be optimised to reduce the processing time required for recognising objects. This requires the development of more efficient image processing algorithms that are also robust and accurate.

Much work still needs to be done into the processing of images in real time, especially when objects become more complicated geometrically. A more robust algorithm for camera self-calibration that works for any robot set-up is also under development in our group.

The control of a serial robot still needs to be investigated for manipulating a wider range of 3D objects. New hybrid algorithms need to be developed for applications that require multi-objective control tasks, e.g. trajectory and force. Efficient algorithms also need to be developed for the inverse kinematics of a serial robot. In this regard, we will further improve the Sugeno Fuzzy Inference System for the inverse kinematics by optimising the number of fuzzy rules. The rule-base of the fuzzy system is equally important and has larger effect on time efficiency of the system.

7. References

Abdullah, M. Z., M. A. Bharmal, et al. (2005). High speed robot vision system with flexible end effector for handling and sorting of meat patties. *9th International Conference on Mechatronics Technology.*

Abdullah, M. Z., L. C. Guan, et al. (2004). The applications of computer vision system and tomographic radar imaging for assessing physical properties of food. *Journal of Food Engineering* 61: 125-135.

Bay, H., T. Tuytelaars, et al. (2006). SURF: Speeded up robust features. *Lecture Notes in Computer Science* 3951 NCS: 404-417.

Billingsley, J. and M. Dunn (2005). Unusual vision - machine vision applications at the NCEA. *Sensor Review* 25(3): 202-208.

Brosnan, T. and D.-W. Sun (2004). Improving quality inspection of food products by computer vision - A review. *Journal of Food Engineering* 61: 3-16.

Brunelli, R. and T. Poggio (1993). Face recognition: features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(10): 1042-1052.

Buker, U., S. Drue, et al. (2001). Vision-based control of an autonomous disassembly station. *Robotics and Autonomous Systems* 35(3-4): 179-189.

Burschka, D., M. Li, et al. (2004). Scale-invariant registration of monocular endoscopic images to CT-scans for sinus surgery. *Lecture Notes in Computer Science* 3217(1): 413-421.

Denavit, J. and R. S. Hartenberg (1955). A kinematic notation for lower pair mechanisms. *Journal of Applied Mechanics* 23: 215-221.

Fu, K. S. and J. K. Mui (1981). A survey on image segmentation. *Pattern Recognition* 13: 3-16.

Graham, A. E., S. Q. Xie, et al. (2007). Robotic Long Bone Fracture Reduction. Medical Robotics. V. Bozovic, *i-Tech Education and Publishing*: 85-102

Harris, C. and M. Stephens (1988). A combined corner and edge detector. *Proceedings of the 4th Alvey Vision Conference*: 147-151.

Hartley, R. and A. Zisserman (2003). *Multiple View Geometry in Computer Vision*, Cambridge University Press.

Hough, P. (1962, December). *Method and means for recognizing complex patterns*.

Jamwal, P. K. and H. S. Shan (2005). Prediction of Performance Characteristics of the ECM Process Using GA Optimized Fuzzy Logic Controller. *Proceedings of Indian International Conference on Artificial Intelligence*.

Jang, J. S. R., C. T. Sun, et al. (1997). *Neuro-Fuzzy and Soft computing*. NJ, Prentice Hall.

Jeong, S., J. Chung, et al. (2005). Design of a simultaneous mobile robot localization and spatial context recognition system. *Lecture Notes in Computer Science* 3683 NAI: 945-952.

Kadir, T., A. Zisserman, et al. (2004). An affine invariant salient region detector. *Lecture Notes in Computer Science* 3021: 228-241.

Kass, M., A. Witkin, et al. (1988). Snakes, active contour model. *International Journal of Computer Vision*: 321–331.

Ke, Y. and R. Sukthankar (2004). PCA-SIFT: A more distinctive representation for local image descriptors. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2: 506-513.

Krar, S. and A. Gill (2003). *Exploring Advanced Manufacturing Technologies*. 200 Madison Avenue, New York., Industrial Press Inc.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2): 91-110.

Lucas, B. D. and T. Kanade (1981). An iterative image registration technique with an application to stereo vision. *Proceedings of Imaging understanding workshop,*: 121-130.

Mikolajczyk, K. and C. Schmid (2004). Comparison of affine-invariant local detectors and descriptors. *Proceedings of the 12th European Signal Processing Conference*: 1729-1732.

Mikolajczyk, K. and C. Schmid (2005). A performance evaluation of local descriptors. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2: 257-263.

Pearson, K. (1991). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine* 2(6): 559-572.

Pena-Cabrera, M., I. Lopez-Juarez, et al. (2005). Machine vision approach for robotic assembly. *Assembly Automation* 25(3): 204-216.

Spong, M. W., S. Hutchinson, et al. (2006). *Robot Modeling and Control*, John Wiley & Sons, Inc.

Tsai, R. Y. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation* RA-3(4): 323-344.

Weijer, J. V. D. and C. Schmid (2006). Coloring Local Feature Extraction. *Lecture Notes in Computer Science* 3952: 334-348.

Wong, A. K. C., L. Rong, et al. (1998). Robotic vision: 3D object recognition and pose determination. *IEEE International Conference on Intelligent Robots and Systems* 2: 1202-1209.

Yaniv, Z. and L. Joskowicz (2005). Precise Robot-Assisted Guide Positioning for Distal Locking of Intramedullary Nails. *IEEE Transactions on Medical Imaging* 24(5): 624-625.

Zitova, B. and J. Flusser (2003). Image registration methods: A survey. *Image and Vision Computing* 21(11): 977-1000.