Behaviour Based Assembly Experiments using Vision Sensing

Prabhas Chongstitvatana
Alistair Conkie
Department of Artificial Intelligence
University of Edinburgh
5 Forrest Hill, Edinburgh, EH1 2QL, Scotland

Abstract

We discuss design and experimentation with vision sensing in robotic assembly, in the context of a behaviour based approach. This approach leads to an elegant way of incorporating sensing into an assembly system. The method used does not need a coordinate system that is common to the subcomponents of the assembly system, and has minimal representation. It has strong coupling between sensing and action. A complete assembly task was tested to show the robustness of the system.

Keywords: Robotic Assembly, Behaviour Based System, Vision

Introduction

The motivation for the work described here comes from dissatisfaction with today's assembly robot systems which cannot cope with uncertainties nor handle sensing adequately, and are in general hard to program.

We have been pursuing the behaviour based approach to robotic assembly in which planning is done in terms of behavioural modules [12], [13]. Behavioural modules, amongst other things, are an abstraction mechanism allowing assembly planning without detailed consideration of the real world. Plans are specified in terms of behavioural modules. Each behavioural module is designed to achieve a particular task in a robust manner, at run time, in the robot cell. Behaviours can interact with each other via the real world. For example, one behaviour could pick up a part then put it down at a predetermined location for another behaviour to reorient it.

Malcolm[11] has demonstrated a successful automatic assembly system for the Soma world using this approach. Soma is a set of parts made by considering all the different ways in which 4 or less cubes of the same size can be abutted face-to-face so as to form irregular solids. There are seven such parts [4], (see Fig. 1). The Soma assembly planner (Somass) is an automated assembly planning and execution system. It plans and assembles the Soma parts according to a specified shape of final assembly. The plan specifies ideal motions of parts. The behavioural modules, at run time, execute the robot motions to achieve the required part motions.

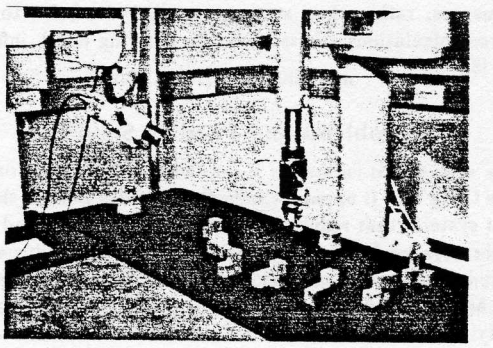

Figure 1: The Assembly Cell and Soma Parts

In the Somass system, the implementation of behaviours uses no sensing. Instead it employs constrained motion to locate the parts. This paper describes extensions to the Somass system, using vision sensing for the part-acquisition behaviour. This extension avoids the use of a global coordinate system for using the vision data and employs a self-calibration procedure to help achieve robustness. The position of the camera does not need to be known.

As we shall see, the experiments also show that two robots, one holding a camera the other performing the assembly, can cooperate with each other in a very natural way without either having to know exactly where the other is.

Strategy

Review of Previous Work

Visual feedback has been used to guide robots in "hand-eye" coordination tasks since the early days of robotic research. Jones [8] used a simple tracking method to control a camera's aim and derived useful information from the image in terms of pixel coordinates. To increase assembly robot capabilities, a geometrical model system was introduced. This was used to reason about spatial occupancy, for collision avoidance, and to plan robot actions. The use of geometric modelling together with the control of robot hand in a cartesian coordinate system gives rise to the need of mapping objects into a common coordinate frame. Most robotic assembly systems build this map [7], [14], [15], [9], [6].

Vision systems have been used to build up 3D models of objects by various techniques, for example binocular stereo vision [7], laser range finder systems with model-based object localisation [9]. These systems incorporate explicit geometric models of the environment, the robot, and the camera. The success of operation relies heavily on the accuracy of these models and calibration of the vision system with respect to robot arm and environment.

In dealing with a dynamic world, Andersson [1] combines both real-time performance of visual feedback and manipulation that demands high speed response. He uses special hardware and sophisticated calibration techniques to achieve the accuracy required. It is worth noting that the technique of using image sequences, rather than static scene analysis, helps to simplify stereo calculations and is useful for deriving depth information [2], [5].

Problems and Possible Solutions

One thing that is apparent from the previous work in this area is that systems that require an accurate world model and accurate calibration tend to be large and complicated, possibly unweildy. This is a situation we would prefer to avoid. Various principles are adopted to help in making things simpler. The general underlying motivation is to reduce the cost of representation and increase robustness.

The first principle is to try and replace, as far as is possible, calculations by sensing. This equates to preferring perception over representation. Several techniques are used that contribute to this end. Frequent vision sensing, for example, allows motions in the world to be viewed as linear approximations, and only short term predictions of motion need be made for following an object. The tracking algorithm adopted is similar to that used by others [8], [2].

The system is purposely made as calibration free as possible. Relative quantities are used and self-calibration is done while carrying out tasks. This approach is possible because adequate sensing allows self-calibration. The need for calibration to establish the mapping of image data to the world coordinate is absent from our system. The camera position need not be known.

The idea of having an explicit world model is rejected in favour of using heuristics to guide the robot hand, in contrast to much previous work. Contextual information is also exploited in the same spirit as others have used it [7], [8]. The system does not rely on explicit models of the objects nor does each assembly agent needs to know about the other(s). The system does not maintain representations of the Soma pieces, the camera, the robots nor of possible uncertainties due to inaccuracies of the model with respect to the real world, unlike other systems.

This philosophy of avoiding using a world model and instead relying directly on vision to pick up the Soma pieces has benefit as regards uncertainties. Uncertainties are dealt with directly, rather than being treated as an afterthought that would fall outwith the formalism of the system. Direct observation and feedback allow the possibility of removing many uncertainties without resorting to complex analysis.

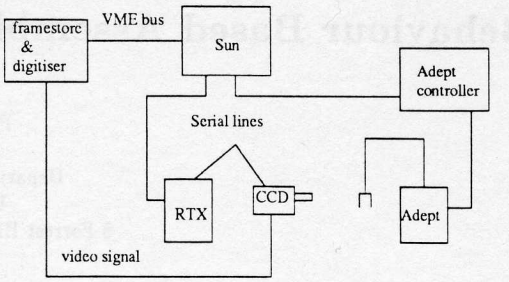

Figure 2: System Architecture

The Experiments

System Architecture and Setup

Our system consists of two robots and a Sun workstation with some vision processing hardware (see Fig. 2).

One robot is an Adept industrial robot, with 5 degrees of freedom. This robot carries out the assembly task. It has a simple pneumatic close-open parallel gripper. The finger are painted white and the rest of the hand painted black.

A second robot arm (an RTX made by UMI Ltd.) is used to hold a small camera in the second experiment which is moved to track the movements of the Adept arm.

The Sun machine contains two MaxVideo boards, an image digitiser and a framestore. A subset of the data in the framestore is selected and analysed on the Sun. The hardware is driven in interrupt mode. Both robots are controlled via serial lines from the Sun. The software is written in C and Prolog.

The camera is a CCD video camera with automatic gain control, Sony DXC-101P with 16 mm. lens. The work cell is lit by a white light lamp, and the surface of the table is covered by a black cloth, which gives adequate contrast.

The Soma set we used is made of wood and is not painted. It is in its natural texture and colour. The plan for the complete assembly is generated by Soma Planner [11]. Only the pick-up behavioural module is substituted into the plan.

Two experiments were carried out. In the first a behaviour was devised to pick up a Soma part from the work area. Starting with the robot hand above the nominal (predefined) position of the part, and the part in approximately its nominal position, a limited amount of image data combined with a simple strategy to move the hand can guide the hand to pick up the part. This was done without referring to robot, part, or camera coordinates.

The second experiment supplements the first. A behaviour capable of finding the robot hand is used. The camera follows the hand, it moves to keep the hand in its field of view. This tracking behaviour thus aims the camera at the right place for the pick-up behaviour to start, replacing the manual camera movements of the first experiment.

In the context of a complete assembly, the part-acquiring behavioural module is used to locate each part. The vision system does not have to recognise the individual part. Before we pick up a part we move the hand to directly above the nominal position of the part. Because we can locate the fingertips we know where the hand is in the image. We search for the part in the area below the hand. The camera is moved to aim at the correct part each time the new part is to be acquired.

First Experiment: Picking up a Soma Part

In this experiment the camera is moved manually, for each part, so that both the robot hand and the part to be picked up are within the field of view. The part has a nominal position with the translation variance of the part being about twice its cube size and the rotation variance being plus or minus 30 degree.

System Architecture and Setup

The camera looks down at an angle that can vary between approximately 30 and 60 degrees, and approximately plus or minus 15 degrees horizontally, from the ideal "head-on to the robot fingers" position. A wide range of positions are possible. The goal is to use the information derived from the image data to guide the hand to align the finger tips with one edge of the top cube of the part. We assume that the height of the part is known, and the part lies flat on the working surface with the cube to be grasped pointing up. The technique used requires that both the fingers and the specified edge are in clear view.

The strategy is as follows (see Fig. 3): We define a plane parallel to the work surface at a known height, and above the tops of the parts, which we call the "approach plane". There are two points of interest that we then project onto the "approach plane". The first is the midpoint between the two fingers of the gripper. The second is the midpoint of the top edge (in the image) of the part. The projection of this point we call the "approach point". We drive the robot in the "approach plane" so as to minimise the distance between these two points.

There is a continuous 1-to-1 mapping from the image plane to the approach plane. This means that, in practice, we can work with the image data only. The robot hand is driven vertically a known distance to lie in the "approach plane". The resulting pixel displacement of the hand allows us to deduce, the "approach point" in the image. Successive estimates are made of how the robot should move which are refined based on the actual movements (again in the image plane), until the error is reduced to an acceptable level. This process is repeated twice.

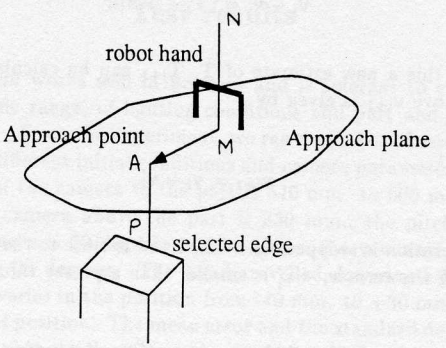

Figure 3: Strategy Diagram

Finally the hand is rotated so that it is parallel to the edge and then it can move down to the final position.

We show here two parts of the analysis, the first part treats the method of visual servoing the hand and the second part treats the analysis of the perspective distortion of projecting the image into the camera.

Visual Servoing

We consider the mapping of the motion in a plane in the robot coordinate frame to the image plane.

$$\mathbf{v} = \mathbf{T}\,\mathbf{m} \tag{1}$$

where $\mathbf{m}$ is a vector $\overrightarrow{PQ}$ joining the point P to the point Q in the robot coordinate frame which lies on the approach plane. $\mathbf{v}$ is the corresponding vector in the image plane formed by projecting P, Q into p, q. $\mathbf{v} = \overrightarrow{pq}$. $\mathbf{T}$ is the transformation of the points in the approach plane into the image plane.

The robot hand has two translational and one rotational degrees of freedom. Only the translational components are considered here. The transformation $\mathbf{T}$ is estimated iteratively. The first estimate of $\mathbf{T}$, $\mathbf{T}_1$, comes from observing the vector between the fingertips, with *a priori* knowledge about their position in the robot coordinate frame. In general, $\mathbf{T}_i$ can be used to estimate $\mathbf{m}_i$.

$$\mathbf{m}_i = \mathbf{T}_i^{-1}\,\mathbf{v}_i \tag{2}$$

Then specify the robot to move by $\mathbf{m}_i$, The observed movement is $\mathbf{v}_i'$

$$\mathbf{v}_i' = \mathbf{T}\,\mathbf{m}_i \tag{3}$$

$$\mathbf{v}_i - \mathbf{v}'_i = (\mathbf{T}_i - \mathbf{T})\mathbf{m}_i \quad (4)$$

From this a new estimate of $\mathbf{T}$, $\mathbf{T}_{i+1}$ can be calculated. The next move $\mathbf{v}_{i+1}$ is given by

$$\mathbf{v}_{i+1} = \mathbf{v}_i - \mathbf{v}'_i \quad (5)$$

The iteration is stopped when the hand is close to the goal, that is when the term $\mathbf{v}_i - \mathbf{v}'_i$ is smaller than a preset value.

Analysis of the perspective distortion

Our method finds the approach point by moving the hand downward by half of the vertical distance from the top of the object thus making the line PA parallel and equal to MN. (see Fig. 3) Because of the perspective distortion, there will be a discrepancy QA, (see Fig. 4) which will create an error AA' in determining the approach point.

We determine QA as follows, d_1, d_2 are measured along the optical axis of the camera and f is the focal length of the camera.

From the triangle OMN,

$$\frac{ON}{\sin\beta_1} = \frac{MN}{\sin\gamma} \quad (6)$$

$$\frac{OC}{\sin\alpha_1} = \frac{BC}{\sin\gamma} \quad (7)$$

$$BC = h\,\frac{OC}{ON}\,\frac{\sin\beta_1}{\sin\alpha_1} \quad (8)$$

and similary the triangle OPQ, where $BC = DE$

$$DE = PQ\,\frac{OE}{OQ}\,\frac{\sin\beta_2}{\sin\alpha_2} \quad (9)$$

from $\frac{OC}{ON} = \frac{f}{d_1}$; $\frac{OE}{OQ} = \frac{f}{d_2}$

$$PQ = h\,\frac{d_2}{d_1}\,\frac{\sin\beta_1}{\sin\beta_2}\,\frac{\sin\alpha_2}{\sin\alpha_1} \quad (10)$$

$$QA = h - PQ \quad (11)$$

$$QA = h\left(1 - \frac{d_2}{d_1}\,\frac{\sin\beta_1}{\sin\beta_2}\,\frac{\sin\alpha_2}{\sin\alpha_1}\right) \quad (12)$$

We can arrange the camera position such that QA is small. This does not mean we have to know the exact position of the camera.

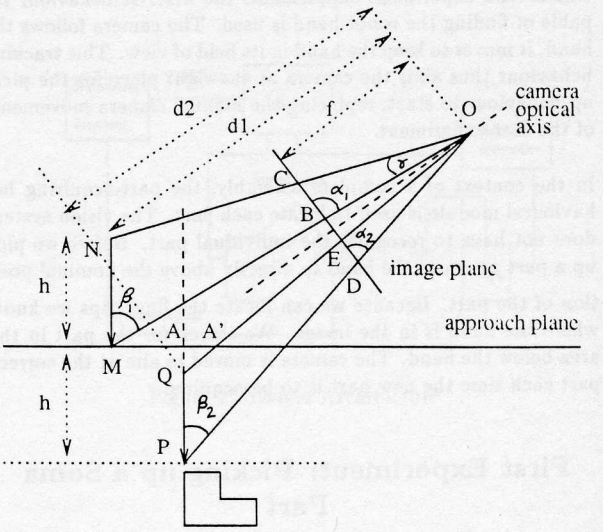

Figure 4: Determination of the perspective distortion

The camera position is not known but bounded. The permissible range is large enough such that in practice the camera can be positioned without any measurement tools.

Vision processing

The information we needed from the image is the boundary of the Soma part, and the pixel coordinates of the fingertips.

In order to find in the first instance where the fingertips are, we difference two pictures with only the fingers in different positions. Thus we can locate the initial finger positions and select two small windows enclosing the tips and keep these windows tracking each fingertip throughout the hand motion by moving the windows to keep the finger centroids in the middle. The fingertips are painted white, so they are the brightest objects in the scene. This makes their identifications more robust against the background.

The background is the easiest and most invariant thing to identify in our image. We make use of the knowledge about the image histogram (see Fig. 5), that it is composed of three distributions; from the part, which has low peak and large variance, from the background which has high peak and small variance and from the shadow which is darker than the background. The histogram is first smoothed. To separate the background we find the maximum peak, which is the mean of the background distribution, analyse the variance of this distribution and select the threshold. This algorithm works well even if shadows are present.

The system does not recognise the part in the usual sense. It follows the hand, using it as a pointer. A binary representation of the area image below the hand is made using thresholding as described above. The outline of the part is traced and the polygon approximation of that shape derived [10]. Then we select

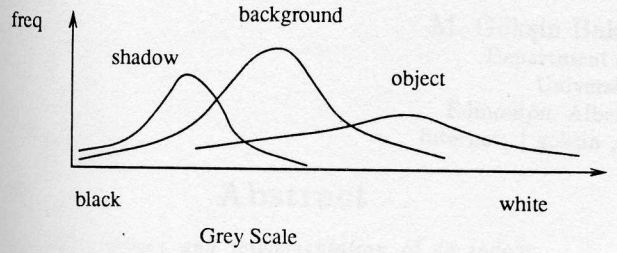

Figure 5: Components of the Histogram

the most horizontal top edge. It is the least obscured against the background and other objects, and the most reliable feature to look for.

Second Experiment: Tracking the Robot Hand

The manual positioning of the camera in the first experiment is replaced by automatic positioning using tracking. To establish the correct preconditions for the pick-up behaviour we use the second robot (the RTX) to hold the camera and to aim it correctly for each part to be picked up. We do this by tracking the hand. The method of tracking does not require any knowledge about the camera position. It starts with seeing the hand move, and follows it until it goes to the ready position to pick up a part. When the hand is there, the camera is already aiming at about the right place. We then stop the camera motion and let the pick-up behaviour carry on until it is finished then follow the hand to the next acquisition.

The tracking system is composed of two behaviours, one for following the moving hand, and the second for locating the hand.

The following the moving hand behaviour does not assume any knowledge of the shape of the hand. It is based on finding the difference between two successive images. The moving camera will be instructed to centre its view on the part of the image that has changed. This in itself does not give accurately the location of the hand in motion and cannot be used when the hand does not move. In conjunction with this behaviour, the second behaviour was used to locate the hand.

The behaviour to locate the hand looks for a special mark on the robot hand. This mark is brightly coloured. We use a simple spectrum classification scheme based on intensity values of {Red,Green,Blue} to separate the mark from the background and calculate its centroid. Finally, the camera is view-centred on this point. The two behaviours are simple and complement each other well, and are sufficient to establish the proper precondition for the pick-up behaviour.

Test results

The system works well in practice and is tolerant to changes, over a wide range, of lighting conditions and part and camera positions. ¿From the experiment, we ran a number of assemblies with the different initial conditions and camera parameters. The distance of the camera to the part is 540 mm. to 600 mm., the height of camera above the part is 330 mm., the pitch angle of the camera is 20 degrees to 25 degrees, the yaw angle is -10 degrees to 20 degrees. The hand travelled downward 40 mm. The part varies in the position from -40 mm. to +30 mm. from its nominal position. The mean error and the standard deviation in determining the position and orientation of the part are 2.10 mm., SD 0.95 mm., 2.03 degrees, SD 1.47 degrees respectively. The hand reaches the desired position within 3 to 5 moves in average.

The system will fail, as we would expect, if certain conditions are not met. If the contrast is sufficiently low the thresholding will fail. Tracking the fingers will fail if they appear too small in the image, for example if a wide angle lens is used. The angle of the camera with respect to the plane of the table is not critical, but near the horizontal our strategy gives less accurate results and near the vertical (top view) the fingers of the hand are more likely to be obscured.

We were able to arrange the environment so that all these failure modes were avoided without difficulty.

Conclusions

Behavioural modules can make use of vision sensing to achieve robust reliable operations in the assembly cell. Programming using behavioural modules is a good method of problem subdivision, and of dealing with uncertainty.

The system we have developed works without a common coordinate system for the components of the assembly system. Vision processing is simpler and robots can cooperare in a simpler way than would result from using such a coordinate system. In addition, the system is generally robust because it does not depend on calibration of subsystems.

By observing rather than reasoning or predicting we have been able to tightly couple vision sensing and action within the system. This is achieved without having to represent the world and its changes explicitly, thereby avoiding the danger of constructing Dennett's imaginary robot [3], which, as he describes it, is

> ... sitting, Hamlet-like, outside the room containing the ticking bomb, ... "Do something!" they yelled at it. "I am," it retorted. "I'm busily ignoring some thousands of implications I have determined to be irrelevant. Just as soon as I find an irrelevant implication, I put it on the list of those I must ignore, and..." the bomb went off.

References

[1] Andersson, R.L., "A Robot Ping-Pong Player". MIT Press, 1988.

[2] Brown, C.M., Ed., "Rochester Robot", technical report 257, University of Rochester, Computer Science, 1988.

[3] Dennett, D.C., "Cognitive Wheels: The Frame Problem of AI", in The Robot's Dilemma, Pylyshyn, Z.W., ed., Ablex Publishing, 1988.

[4] Gardner M., "Pleasurable Problems with Polycubes", Scientific American, Sept. 1972, p176.

[5] Hayes, G.M., "A Real Time Kinematic Depth System", MSc Dissertation, Edinburgh University, 1989.

[6] Heikkila, Matsushita and Sato, "Planning of Visual Feedback with Robot-Sensor Co-operation", Proc. 1988 IEEE Inter. Workshop on Intelligent Robots and Systems ,Tokyo, 1988.

[7] Inoue, I., "Hand Eye Coordination in Rope Handling", 1st Inter. Symp. on Robotics Research. Bretten Woods USA, 1983.

[8] Jones, V.C., "Tracking: An Approach to Dynamic Vision and Hand-Eye Coordination", PhD Thesis in Electrical Engineering, University of Illinois, Urbana Champaign, 1974.

[9] Lozano-Perez et al, "Handeye : a robot system that recognises, plans, and manipulates", Proc. 1987 IEEE Inter. Conf. on Robotic and Automations vol.2 ,1987, pp. 843-849.

[10] Malcolm, C.A., "The Outline Corner Filter", in Proc. of the 3rd Inter. Conf. on Robot Vision and Sensory Controls, Nov. 1983, pp 61-68.

[11] Malcolm, C.A., "Planning and Performing the Robotic Assembly of Soma Cube Constructions", MSc Dissertation, Edinburgh University, 1987.

[12] Malcolm, C.A., Smithers, T., "Programming Assembly Robots in terms of Task Achieving Behavioural Modules: First Experimental Results", In Proc. of the Inter. Advanced Robotics Program: Second Workshop on Manipulators, Sensors and Steps Towards Mobility, 1988. Also available as research report 410, Department of Artificial Intelligence, Edinburgh University.

[13] Malcolm, C.A., Smithers, T., "Symbol Grounding via a Hybrid Architecture in an Autonomous Assembly System", Workshop on Knowledge Representation and Learning in an autonomous agent, 1988. Also available as research report 420, Department of Artificial Intelligence, Edinburgh University.

[14] Popplestone, R.J., Ambler, A.P. and Bellos, I., "An Interpreter for a Language for Describing Assemblies", Artificial Intelligence 14, 1 (1980), pp. 79-107.

[15] Yin, B., "Using Vision Data in an Object-Level Robot Language RAPT", Inter. Jour. Robotics Research vol 6 no1, 1987, pp. 43-58.