Proc. 2002 Australasian Conference on Robotics and Automation
Auckland, 27-29 November 2002

# Grasping Unknown Objects with a Humanoid Robot

**Geoffrey Taylor and Lindsay Kleeman**
Intelligent Robotics Research Centre
Department of Electrical and Computer Systems Engineering
Monash University, Clayton, VIC 3168, Australia
{Geoffrey.Taylor | Lindsay.Kleeman}@eng.monash.edu.au

**Abstract**

This paper combines the authors' previous work on a self-calibrated, position based visual servoing framework for a humanoid robot, with a robust laser stripe scanner that can capture registered colour/range measurements of arbitrary objects in ambient indoor light. Using stereo measurements for validation, the laser scanner is robust against sensor noise, spurious reflections and cross talk from other robots. Range data is processed to identify objects of interest in the workspace, which are modelled using simple geometric primitives. The resulting textured 3D models can be used for recognition, tracking and grasp planning. Finally, we present a simple grasp planner that guides visual servoing in the task of grasping a modelled object. The fusion of these components allows the humanoid robot to locate and grasp a class of *a priori* unknown objects in its workspace. The effectiveness of these techniques are demonstrated on an experimental humanoid platform.

## 1 Introduction

In previous work the authors developed a domestic humanoid robot (see Figure 1) which used self-calibrated, position based visual servoing to perform complex manipulation tasks [Taylor and Kleeman, 2001]. However, the robot was only able to recognize and locate simple, artificially marked blocks. The motivation for this work is to develop the sensing and visual processing techniques that will allow the robot to classify and locate a broad class of arbitrary objects, taking us one step closer to the ultimate goal of developing a humanoid robot that can perform *ad hoc* tasks in a domestic/office environment.

Visual servoing with domestic robots and real objects has been demonstrated in other research [Kragić, 2001]. However, in this and similar systems the objects are geometrically modelled prior to the manipulation task. In contrast, we believe it is important for the robot to create geometric models autonomously, providing greatest flexibility in performing arbitrary tasks. In our framework, objects will be approximated with geometric primitives such as cubes, spheres and cylinders that are fitted to sensor data during the manipulation task. The use of geometric primitives rather than non-parameterized models (as in [Müller and Wörn, 2000]) has the advantage of allowing us to make reasonable assumptions about hidden regions.

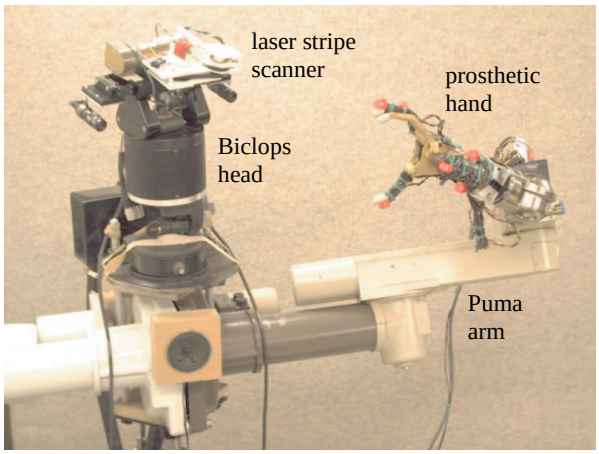


Figure 1. An experimental humanoid robot with stereo stripe scanner, for locating and manipulating arbitrary objects.

A crucial aspect of this approach is the acquisition of dense and reliable colour/range measurements. Passive stereo is usually associated with humanoid sensing, but the accuracy and reliability of current techniques often depend on the content of the scene. Laser stripe range sensing offers a computationally efficient alternative, but also presents unique challenges when used on a domestic robot; the sensor must operate in normal ambient light and must be capable of rejecting sensor noise, spurious reflections and cross talk from other robots. A number of methods for robust laser scanning using measurement validation have been proposed [Haverinen and Röning, 1998; Magee *et al.*, 1994; Nygårds and Wernersson, 1994; Trucco *et al.*, 1998], but these suffer from issues including assumed scene structure, inability to capture colour, and lack of error recovery. The authors have developed a robust stereo laser stripe scanner which addresses these issues and provides dense, registered colour/range measurements [Taylor *et al.*, 2002].

Once the measurements have been acquired, the robot must extract and localize objects of interest and plan the manipulation task. Various methods for segmenting range data have been proposed, and the techniques used in this work are based on 3D connectivity [Stamos and Allen, 2000] and planar region merging [Cobzas and Zhang, 2001; Goldschneider and Li, 2001]. The parameterized data is then fitted with a geometric primitive to produce a textured, 3D polygonal model that can be used for recognition, tracking and grasp planning. While the first two tasks are not addressed in this paper, we present a simpli-

fied grasp planner using known properties of the geometric primitive. As before, the grasp planning component is developed from the general principles upon which similar works are based [Ferrari and Canny, 1992].

The following section describes the hardware configuration of the experimental humanoid platform. Section 3 provides an overview of the visual servoing component, and section 4 describes the framework for robust stereo laser stripe scanning. In section 5 we describe how raw laser data is processed to localize generic objects and plan a grasp. Finally, Section 6 provides experimental validation of the techniques described in this work, and is followed with concluding remarks.

## 2 Experimental Humanoid Platform

Figure 1 shows the mechanical configuration of our experimental upper-torso humanoid robot. The arms are approximately anthropomorphic in configuration and scale, and consist of two 6-DOF Puma 260 robots with 1-DOF Otto Bock prosthetic hands. Vision is provided by a pair of PAL cameras on a Biclops pan/tilt/verge robotic head. The cameras capture stereo images at 320×240 pixel resolution, and image processing occurs at PAL frame rate (25 Hz). The hands, head and Puma controllers are all coordinated via serial links from a 450 MHz Pentium III PC.

The laser stripe generator is mounted on the Biclops head above the cameras. The generator consists of a 5 mW laser diode module with a cylindrical lens to produce a vertical light plane. A DC motor drives the laser about a vertical axis to scan the stripe across a scene, and the rotation angle is measured via an optical encoder. Motor control and optical encoder measurements are implemented on a PIC microcontroller, which communicates with the host PC via a serial link.

## 3 Visual Servoing Framework

The visual servoing component of the humanoid robot is based on the work presented in [Taylor and Kleeman, 2001]. In this work, we developed a position based method that allows visual servoing to be performed without knowledge of the transformation between the camera frame and robot base frame, provided the kinematics of the robot arm are reasonable well known. This is achieved by measuring the frame of the gripper directly, and formulating the pose error in the gripper frame. Figure 2 illustrates the basic visual servoing task.

Artificial cues in the form of red LEDs are attached to the gripper to simplify image processing and increase tracking robustness. The position of the LEDs in the gripper frame *G* are manually calibrated and form an internal model which is used to determine the pose of the gripper. At each iteration of the control loop, the position of the visible LEDs are measured in the camera frame *C* and processed using a Kalman filter to provide an optimal estimate of the pose. The filter employs a constant velocity dynamics model, assuming the motion of the gripper to be smooth. A simple proportional control law is used for visual servoing, based on the translation and rotation required to move the gripper to the target pose *T*, expressed in the current frame of the gripper *G*. These differential pose parameters are passed to the Puma controller, which calculates the appropriate joint motions.

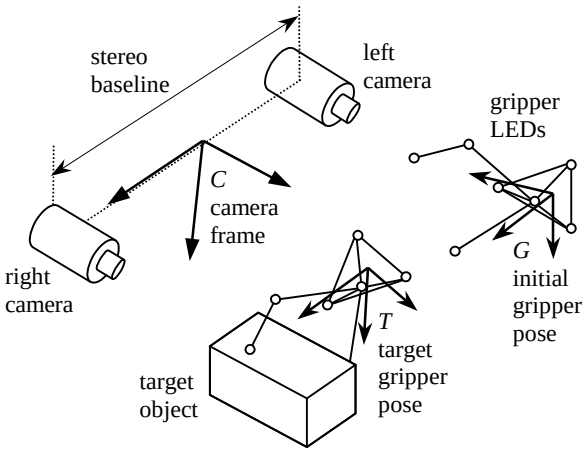


Figure 2. Visual servoing framework.

At the commencement of a new servoing task, the initial state of the gripper is determined using an automated procedure. Firstly, the cameras scan the workspace to find the LEDs, and the Kalman filter is initialised with the average measured position. The LEDs are then flashed individually to provide unambiguous localization (against an arbitrary background). These measurements are processed by the Kalman filter to estimate the initial pose of the gripper, and the cycle is repeated until the filter converges to the initial state. During servoing, loss of visual feedback is minimized by actively tracking the motion of the gripper with the robotic head. However, if tracking is lost, the initialisation procedure also provides an automatic recovery mechanism.

## 4 Robust Laser Stripe Scanning

The robust laser stripe scanner used in this system is based on work published in [Taylor *et al.*, 2002]. The sensor uses stereo measurements of the stripe to achieve a number of advantages over a single-camera configuration. These include robustness to spurious reflections and sensor noise, rejection of cross-talk from other stripe sensors and improved ranging accuracy. The noise rejection properties allow objects to be scanned in ambient indoor light, enabling the cameras to capture registered range and depth. Furthermore, the noise rejection condition is formulated as a linear relationship in image plane coordinates. Thus a simple framework exists for on-line calibration of the relative position between the laser plane and cameras, using measurements of an arbitrary non-planar target.

The basis for formulating a validation condition is that valid measurements must correspond to a point coincident with the laser plane. This may be formulated in a number of ways, but the condition derived below is preferred as it results in a linear relationship between image plane coordinates for the case of rectilinear stereo (parallel optical axes and coplanar image planes). It should be noted that requiring rectilinear stereo does not necessarily constrain the camera configuration; *projective rectification* allows the effective rectilinear coordinates to be recovered for any camera configuration (see [Hartley, 1999]).

In Figure 3, a vertical stripe is projected onto the target object, and we observe point **X** at image plane coordinates $\mathbf{x}_L = (x_L, y)^T$ and $\mathbf{x}_R = (x_R, y)^T$, noting that the rectilinear

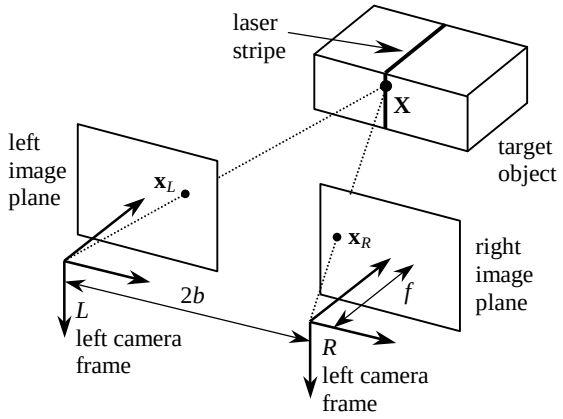

Figure 3. Stereo laser stripe validation framework.

configuration allows us to use the notation $y_L = y_R \equiv y$. Let $\mathbf{\Omega} = (A, B, C, D)^T$ represent the parameters of the laser plane in the world frame. We wish to confirm that **X** is coincident with this laser plane, in which case **X** must satisfy the following plane equations in *L* and *R*:

$$\begin{aligned} A(X + b) + BY + CZ + (D - Ab) &= 0 \\ A(X - b) + BY + CZ + (D + Ab) &= 0 \end{aligned} \quad (1)$$

The projective transformation due to the cameras is described using a pin hole model with focal length *f* (equal for identical cameras), and stereo baseline 2*b*:

$$\begin{aligned} (x_L, y)^T &= (X + b, Y)^T \cdot f / Z \\ (x_R, y)^T &= (X - b, Y)^T \cdot f / Z \end{aligned} \quad (2)$$

Substituting (2) into (1) we have

$$\begin{aligned} (Ax_L + By + Cf)Z + (D - Ab) f &= 0 \\ (Ax_R + By + Cf)Z + (D + Ab) f &= 0 \end{aligned}$$

Finally we can eliminate the unknown depth *Z* to give

$$\begin{aligned} &(D + Ab)(A_L x_L + By + Cf) \\ &\quad - (D - Ab)(Ax_R + By + Cf) = 0 \end{aligned} \quad (3)$$

By rearranging (3) and making the change of variables

$$\begin{aligned} \alpha &= (Ab - D)/(Ab + D) \\ \beta &= 2Bb/(Ab + D) \\ \gamma &= 2Cb/(Ab + D) \end{aligned}$$

we can simplify (3) even further to

$$x_L + \alpha x_R + \beta y + \gamma f = 0 \quad (4)$$

Now, image plane coordinates $\mathbf{x}_L$ and $\mathbf{x}_R$ have two degrees of freedom that must be constrained for validation, and the laser plane test (4) provides one constraint. A second is the epipolar constraint, which is used when searching the image for candidate matches. As noted earlier, the rectilinear stereo epipolar constraint is simply $y_L = y_R \equiv y$.

Equation (4) is linear in image plane coordinates, with the three coefficients $\alpha$, $\beta$, and $\gamma$ related to the laser plane parameters. This suggests a simple mechanism for automatically calibrating the laser plane position from image plane measurements. In this method we simply project the laser onto an arbitrary non-planar target and measure the location of the stripe on each of *n* epipolar lines to produce *n* measurement pairs, $\mathbf{x}_{L,i}$ and $\mathbf{x}_{R,i}$, $i = 1 \ldots n$. From (4), we form the residue

$$r_i = x_{L,i} + \alpha x_{R,i} + \beta y_i + \gamma f \ , \ i = 1 \ldots n \quad (5)$$

and solve the linear regression problem

$$\frac{\partial}{\partial j}\left( \sum_{i=1}^{n} r_i^2 \right) = 0 \ , \ j = \alpha, \beta, \gamma \quad (6)$$

The result gives the coefficients $\alpha$, $\beta$, and $\gamma$ at a particular encoder value. The calibration process is completed by scanning the laser across the target and solving (6) at regular intervals of the encoder value. Finally, parameterised models are fitted to the coefficients, which allows them to be calculated for any encoder value.

When an object is scanned, an edge detection filter identifies candidate positions for the light stripe on a given epipolar line. Candidates are matched by minimizing the residue in (5) for all possible edge pairs on the epipolar line, and the result is validated by requiring that the minimum residue is below a fixed threshold error. Details of the calibration, implementation and image processing are given in [Taylor *et al.*, 2002].

## 5 Grasping an Unknown Object

In this work, grasping is executed using a *look-then-move* approach. First, a range/colour image of the workspace is acquired and the object of interest is identified and localized. The current implementation is able to fit rectangular prisms to the range data, allowing the robot to model unknown boxes. In addition to shape, the robot may distinguish boxes by colour or texture to determine which objects are applicable to a given task. A grasp planning algorithm is then applied to determine the pose of the robot for grasping the target box. Finally, the hand is visually servoed to the desired pose and the object is grasped. The system presented here does not track objects during servoing, but we intend to add this capability in future work to improve servoing accuracy. We also intend to include geometric primitives such as cylinders and spheres to increase the generality of the system.

Figure 4 shows the raw range/colour scan of a typical scene, which will be used as an example for the analysis and grasp planning algorithms described in the following sections.

### 5.1 Range Data Segmentation

The first step in scene analysis is to segment and parameterise the raw range data into planar regions. A surface normal **n** is determined for each square patch of *N* points $\mathbf{m}_i$ in the range image, using least squares plane fitting [Faugeras, 1993]. The normal is calculated as the eigenvector associated with the smallest eigenvalue $\lambda$ of the covariance matrix $\Lambda$ for the range data $\mathbf{m}_i$, where

$$\Lambda = \tfrac{1}{N} \sum_{i=0}^{N} (\mathbf{m}_i - \overline{\mathbf{m}})(\mathbf{m}_i - \overline{\mathbf{m}})^T \ , \quad \overline{\mathbf{m}} = \tfrac{1}{N} \sum_{i=0}^{N} \mathbf{m}_i$$

The smallest eigenvalue $\lambda$ is also the residual error from the least squares fit. Patches for which $\lambda$ is greater than a fixed error threshold (such as at depth discontinuities) are invalidated and discarded. Segmentation of the data into planar regions is a two step process: initial segmentation, followed by region merging. The initial segmentation is similar to the technique presented in [Stamos and Allen, 2000], and based on the familiar sequential binary connec-

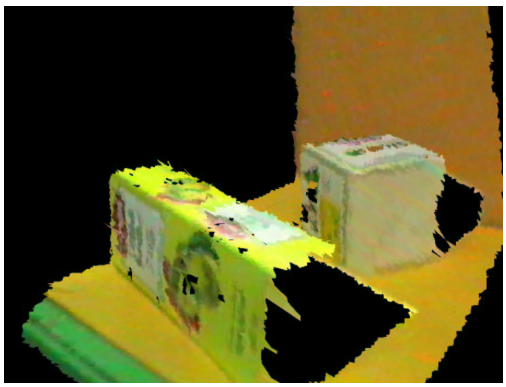

Figure 4. Raw colour/range scan of a typical scene.

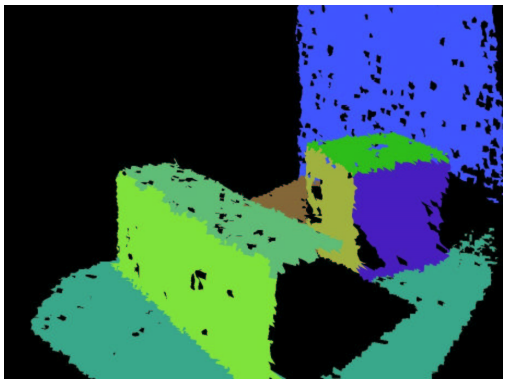

Figure 5. Planar segmentation: extracted planes are labelled with uniform colour.

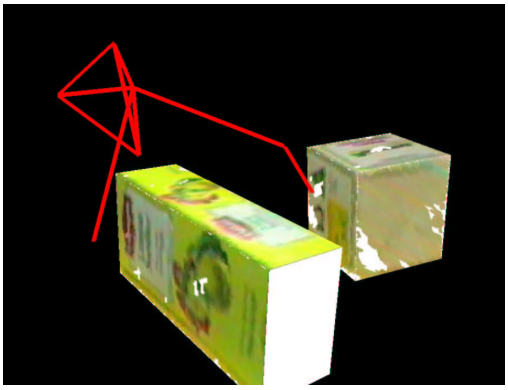

Figure 6. Extracted box models and grasp planning result.

tivity algorithm. For each neighbouring pair $\mathbf{m}_1$ and $\mathbf{m}_2$ with valid normals $\mathbf{n}_1$ and $\mathbf{n}_2$, coplanarity is tested by requiring $\mathbf{n}_1 \cdot \mathbf{n}_2 > \alpha_1$ and $\max(|\mathbf{r}_{12} \cdot \mathbf{n}_1|, |\mathbf{r}_{12} \cdot \mathbf{n}_2|) < \alpha_2$, where $\mathbf{r}_{12}$ is the normalized in-plane vector given by $\mathbf{r}_{12} = (\mathbf{m}_1 - \mathbf{m}_2)/|\mathbf{m}_1 - \mathbf{m}_2|$. Coplanar pairs are assigned identical region labels. Typical thresholds $\alpha_1$ = 0.9 and $\alpha_2$ = 0.1 result in a large number of small regions to be merged later. After connectivity is applied, measurements without a valid normal are simply added to the neighbouring region for which the plane equation is satisfied.

Region merging is based on an iterative boundary cost minimization approach [Goldschneider and Li, 2001; Cobzas and Zhang, 2001]. For each pair of regions $R_i$ and $R_j$ sharing a common boundary, a cost is calculated as the residual error $\lambda_{ij}$ of a plane fitted to the merged region $R_i \cup R_j$. At each iteration of the merging algorithm, the regions which result in the minimum boundary cost $\lambda_{min} = \min\lambda_{ij}$ are merged, and the region list and boundary errors are updated. Merging only a single pair of regions at each iteration ensures a controlled growth of the total error for all regions. The process is repeated until the minimum error grows beyond a set threshold, $\lambda_{min} > \lambda_{th}$. Figure 5 demonstrates the result of applying the segmentation algorithm to the range data shown in Figure 4.

Planes describing distinct objects may now be grouped using the following assumptions. A horizontal plane below all other planes in the scene is identified as a *floor*, and similarly a vertical plane behind all other planes is a *wall*. These are removed and the scene is checked again for floors and walls until none remain. Assuming objects do not touch, the remaining planar regions are segmented into connected groups associated with distinct boxes to be modelled using the technique described next.

## 5.2 Fitting a Generic Box Model

A generic box is modelled as three pairs of parallel planes, with each pair approximately orthogonal to the others. Parallel plane pairs are labelled $A_i$ and $B_i$, $i$ = 1, 2, 3, where $A_i$ is the surface closest to the cameras (the origin of the world frame). Each pair is parameterised by a common normal $\mathbf{n}_i$ and perpendicular distance to the origin, $a_i$ and $b_i$, where $a_i < b_i$. We assume the top of the box is visible, and the system must be able to view at least one other side to determine all parameters of the model. At most, three sides of the box will be visible, these being plane $A_i$ in the three plain pairs. The top surface is identified as the region parallel to the floor, and assigned to $A_1$. The remaining visible regions are assigned to the frontal sides of the box, $A_2$ and $A_3$. If only one side is visible, we assign this to $A_2$ and calculate the normal vector of $A_3$ as the cross product of the visible normals, $\mathbf{n}_3 = \mathbf{n}_1 \times \mathbf{n}_2$.

Now all that remains is to find the distances $b_i$ for the hidden rear faces (and $a_3$ for the case of only two visible front faces), which are calculated by fitting planes to the edges of the visible surfaces. For example, $B_3$ shares an edge with $A_1$ and $A_2$, so the first step in determining $b_3$ is to identify the range points $\mathbf{e}_i$ at the edge of surfaces $A_1$ and $A_2$. The basis of our method is the assumption that $B_3$ will be coincident with one of these edge points. We perform a numerical fit similar to a Hough transform; each edge point $\mathbf{e}_i$ is associated with a likelihood function $c_i$ that measures the support of all edge points for a plane passing through $\mathbf{e}_i$:

$$c_i = \sum_j (d_j + 1)^{-2}, \quad d_j = \mathbf{n}_3 \cdot \mathbf{e}_j$$

The location of $B_3$ is found by searching for the edge point that maximizes this likelihood. Finally we calculate $b_3 = \mathbf{e}_{max} \cdot \mathbf{n}_3$. The box model is completed by calculating the distance parameters for the other hidden sides.

The final step in model construction is to extract surface textures from the colour component of the laser scan. Each visible face of the box is assigned a texture image of fixed size. To render the texture for a particular surface, the range data is projected onto the associated plane and transformed into texture coordinates. These are then tessellated into triangles and rendered with the associated colour data to form the texture image. Figure 6 shows the final 3D rendered models of the boxes extracted from the scan in Figure 4.

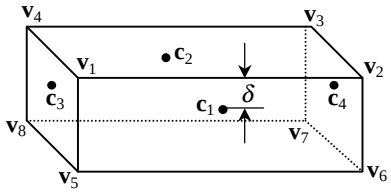

Figure 7. Candidate contact points $\mathbf{c}_i$ for grasping a box.

### 5.3 Grasp Planning

After the object has been modelled and localized, a grasp planner calculates the pose of the gripper for a stable grasp. Alternatively, the grasp planner may determine that the object cannot be grasped at its current location. In this work we present a simple grasp planner that considers only precision grasps of boxes, and does not check for collisions with other objects.

Calculation of candidate grasps is partly based on the criteria developed in [Smith *et al.*, 1999]. As the object is *a priori* unknown, the grip should be chosen to have minimum dependence on the unknown surface friction. Thus, the force applied by the fingers should be normal to the gripped surface, and the object should be grasped near the centre of mass to minimize load torque when lifted. These rules are easily applied to a box under the reasonable assumption of a uniformly distributed mass.

The procedure followed by most grasp planning algorithms is to apply heuristics to determine a set of candidate contact points on the object, then select the best candidate based on a suitable cost function. Our simple, two-fingered grasp planner considers only the two candidate contact point pairs shown in Figure 7: ($\mathbf{c}_1$, $\mathbf{c}_2$) and ($\mathbf{c}_3$, $\mathbf{c}_4$), where $\mathbf{c}_1$ and $\mathbf{c}_3$ are on the visible faces. The axis joining contact pairs is perpendicular to the surface of the box and above the centre of mass, satisfying the rules for minimizing the effect of surface friction. To ensure the grasp is reachable, the contacts are placed a fixed distance $\delta$ below the top surface (typically $\delta$ = 1 cm). The contact points are calculated from the corner vertices $\mathbf{v}_1 \ldots \mathbf{v}_8$; thus $\mathbf{c}_1$ is

$$\mathbf{c}_1 = \mathbf{v}_1 + \tfrac{1}{2}(\mathbf{v}_2 - \mathbf{v}_1) + \delta(\mathbf{v}_5 - \mathbf{v}_1)/\left|\mathbf{v}_5 - \mathbf{v}_1\right|$$

with the other contact points calculated similarly. In the following we consider only the pair ($\mathbf{c}_1$, $\mathbf{c}_2$), and the same calculations are applied to ($\mathbf{c}_3$, $\mathbf{c}_4$).

The first requirement for a suitable grasp is that the contact points are sufficiently close. Let $\mathbf{f}_t$ and $\mathbf{f}_i$ represent the location of the thumb and index fingertips in the gripper frame, with the gripper opened to the maximum grasp; we require $\left|\mathbf{c}_1 - \mathbf{c}_2\right| < \left|\mathbf{f}_t - \mathbf{f}_i\right|$. If this is satisfied, we calculate the transformation that will align the fingertips with the contact points, such that $\mathbf{f}_t$ contacts $\mathbf{c}_1$ and $\mathbf{f}_i$ with $\mathbf{c}_2$ (note that the thumb contacts the surface closer to the robot, as occurs in human grasping). Expressing the rotational component of this transformation as $\mathrm{R}_1(\mathbf{a}_1,\theta_1)$, we calculate the rotation axis $\mathbf{a}_1$ as

$$\mathbf{a}_1 = (\mathbf{f}_t - \mathbf{f}_i) \times (\mathbf{c}_2 - \mathbf{c}_1)$$

and the rotation angle $\theta_1$ as

$$\cos\theta_1 = (\mathbf{f}_t - \mathbf{f}_i)^T (\mathbf{c}_2 - \mathbf{c}_1)/(\left|\mathbf{f}_t - \mathbf{f}_i\right|\left|\mathbf{c}_2 - \mathbf{c}_1\right|)$$

We then apply a second rotation $\mathrm{R}_2(\mathbf{a}_2,\theta_2)$ about $\mathbf{a}_2 = \mathbf{c}_2 - \mathbf{c}_1$ to ensure that the palm of the hand is above the top surface of the box (ie. $\delta$ above the fingertips) and a collision is avoided. Currently, rotation angle $\theta_2$ is calculated using a simple numerical search over all angles. These two rotations combine to give the rotational component of the target gripper pose as $\mathrm{R}_g = \mathrm{R}_2\mathrm{R}_1$. The translational component $\mathbf{t}_g$ of the target pose simply aligns the midpoint between the fingertips above the centre of mass:

$$\mathbf{t}_g = \tfrac{1}{2}(\mathbf{c}_1 + \mathbf{c}_2) - \tfrac{1}{2}\mathrm{R}_g(\mathbf{f}_t + \mathbf{f}_i)$$

Finally we check that the target pose ($\mathrm{R}_g$, $\mathbf{t}_g$) is reachable by ensuring that it is not beyond the length of the arm.

A candidate grasp is calculated for both contact pairs ($\mathbf{c}_1$, $\mathbf{c}_2$) and ($\mathbf{c}_3$, $\mathbf{c}_4$). When both candidates are reachable, a mechanism must be established for selecting the preferred grasp. Here we apply a simple heuristic: the target pose that minimizes the angle between wrist and forearm is chosen as being the most *comfortable* grasp.

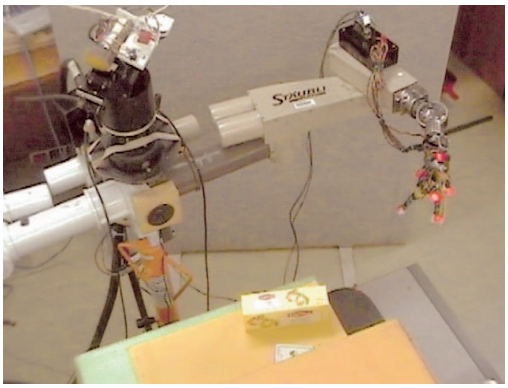
Figure 8. Initial pose of humanoid and target box.

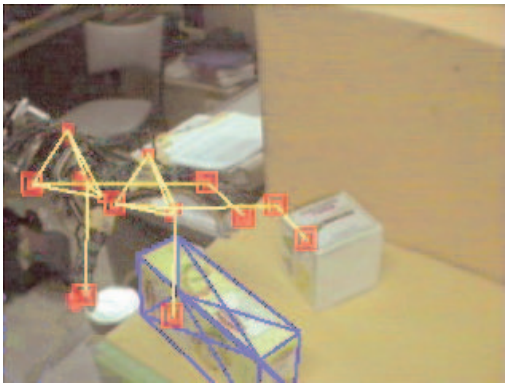
Figure 9. Gripper tracking and box location while servoing.

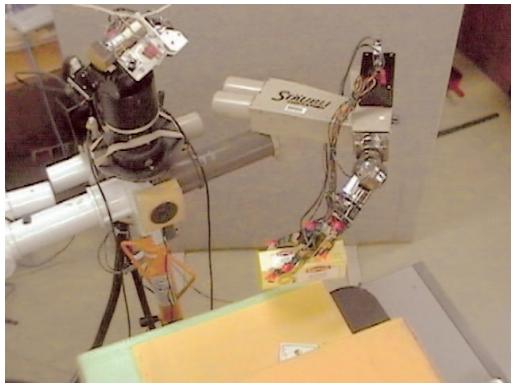
Figure 10. Successful completion of grasping task.

## 6 Experimental Results

The techniques described in this paper have been implemented on the experimental humanoid robot shown in Figure 1. The humanoid is presented with two *a priori* unknown boxes, and given the task of grasping the first yellow box found in the workspace. The initial position of the robot and the boxes are given in Figure 8. Figure 4 shows the raw colour/range scan of the workspace with the candidate boxes, and the extracted models are rendered in Figure 6. The system identifies the target box by examining the colour histrogram of the texture images for each model. Figure 6 also shows the planned grasp for the correctly identified box, indicated by the red wireframe hand model. Colour/range data acquisition, modelling and grasp planning require about 60 s of processing time.

The planned grasp is used as the target pose in the visual servoing task. The image shown in Figure 9 was captured by the robot about halfway through the servoing task; the measured position of the box and the current and target positions of the hand are overlaid as wireframe models. The total time for visual servoing was about 30 s. Finally, Figure 10 shows the position of the robot at the successful completion of the grasping task.

## 7 Conclusions and Future Work

We have presented a fusion of self-calibrated visual servoing, robust laser stripe scanning and object modelling to enable our experimental humanoid robot to locate and grasp boxes in its workspace. The significance of our system is that target objects may be *a priori* unknown, and are parameterised during the grasping task. Range data from the scanner is segmented and fitted with planes using established techniques, and we develop a novel algorithm for fitting a rectangular prism to the segmented data. Finally, we develop a grasp planner that exploits known geometric properties of the box to simplify planning.

The system is currently restricted to modelling rectangular prisms. In future work we intend to include primitives such as cylinders, spheres and even compound objects such as cups and bottles, each with specialized grasp planners. The issue of obstacle avoidance must be addressed by implementing a motion planner to ensure that the arm avoids any collisions while moving from the initial position to the target grip site. Finally, the current implementation only uses a *dead reckoning* servoing strategy; once the object is localized from the range data it is no longer tracked. In future work we intend to exploit the colour and geometric information in the textured model to track the target while servoing, and thus improve grasping robustness and accuracy.

## Acknowledgement

This project was funded by the Strategic Monash University Research Fund for the *Humanoid Robotics: Perception, Intelligence and Control* project at IRRC.

## References

[Cobzas and Zhang, 2001] D. Cobzas and H. Zhang. Planar Patch Extraction with Noisy Depth Data. In *Proc. Third International Conference on 3D Digital Imaging and Modeling*, 2001, pp. 240-5.

[Faugeras, 1993] O. D. Faugeras. *Three Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Boston, 1993.

[Ferrari and Canny, 1992] C. Ferrari and J. Canny. Planning Optimal Grasps. In *Proc. 1992 IEEE International Conference on Robotics and Automation*, 1992, pp. 2290-95.

[Goldschneider and Li, 2001] J. R. Goldschneider and A. Q. Li. Variational Segmentation by Piecewise Facet Models with Application to Range Imagery. In *Proc. 2001 IEEE International Conference on Image Processing*, vol. 1, 2001, pp. 812-15.

[Hartley, 1999] R.I. Hartley, "Theory and Practice of Projective Rectification", *International Journal of Computer Vision*, vol. 35, no. 2, 1999, pp. 115-127.

[Haverinen and Röning, 1998] J. Haverinen and J. Röning. An Obstacle Detection System Using a Light Stripe Identification Based Method. In *Proc. IEEE International Joint Symposium on Intelligence and Systems*, 1998, pp. 232-236.

[Kragić, 2001] D. Kragić. *Visual Servoing for Manipulation: Robustness and Integration Issues*. PhD Thesis, Royal Institute of Technology, Stockholm, 2001.

[Magee *et al.*, 1994] M. Magee, R. Weniger and E. A. Franke. Location of features of known height in the presence of reflective and refractive noise using a stereoscopic light-striping approach. *Optical Engineering*, vol. 33, no. 4, April 1994, pp. 1092-1098.

[Müller and Wörn, 2000] M. Müller and H. Wörn. Planning of Rapid Grasp Operations in Unstructured Scenes. In *Proc. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 2000, pp. 1975-80.

[Nygårds and Wernersson, 1994] J. Nygårds and Å. Wernersson. Specular Objects in Range Cameras: Reducing Ambiguities by Motion. In *Proc. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 1994, pp. 320-328.

[Smith *et al.*, 1999] G. Smith, E. Lee, K. Goldberg, K. Böhringer, J. Craig. Computing Parallel-Jaw Grips. In *Proc. 1999 IEEE International Conference on Robotics and Automation*, 1999, pp. 1897-1903.

[Stamos and Allen, 2000] I. Stamos and P. K. Allen. 3-D Model Construction Using Range and Image Data. In *Proc. International Conference on Computer Vision and Pattern Recognition*, vol. 1, 2000, pp. 513-516.

[Taylor and Kleeman, 2001] G. Taylor and L. Kleeman. Flexible Self-Calibrated Visual Servoing for a Humanoid Robot. In *Proc. Australian Conference on Robotics and Automation*, 2001.

[Taylor *et al.*, 2002] G. Taylor, L. Kleeman and Å. Wernersson. Robust Colour and Range Sensing for Robotic Applications Using a Stereoscopic Light Stripe Scanner. In *Proc. 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, to be published.

[Trucco *et al.*, 1998] E. Trucco *et al*. Calibration, Data Consistency and Model Acquisition with a 3-D Laser Striper. *International Journal of Computer Integrated Manufacturing*, vol. 11, no. 4, 1998, pp. 292-310.