Y.F. Li[1] and M.H. Lee[2]

Applying Vision Guidance in Robotic Food Handling

This paper presents the development of a visually guided robotic system for handling food products that are presented in an unstructured manner. Vision guided grasping is described in the context of vector manipulations. The grasp vector is formulated using both on-line visual information and off-line data. Problems due to the constraints imposed in food handling applications are addressed.

Food products normally exhibit a degree of variability and are often presented at the packing sites by means of systems which do not control their position or orientation [5, 6]. Most of today's assembly robots however are not directly applicable to such processes due to the stringent requirement of a precisely known robot environment for both motion and grasp planning. As a result, the assembly lines of today's food manufacturing industry are still highly labor intensive although the tasks consist of simple and repetitive operations. Developing flexible robotic handling systems to tackle these uncertainties and accommodate task changes with minimum operator interference is of practical importance [4, 3].

The key to the automation of such processes lies in introducing external sensory capabilities and integrating them with robot manipulation. The past few years have seen research efforts made towards this end. These include work in designing specialized grippers and using a vision system for product identification and inspection in poultry product handling [5], visually guided straightening of deformed natural products [4], as well as the use of knowledge to overcome the limitations of visual information in processing products of high variability [6]. However, severe difficulties arise in adapting these systems to applications with different environments due to the special nature of the hardware and software. In food handling applications, this is significant as the reconfigurability of a robotic system is of crucial importance for a flexible manufacturing system. It is also of practical interest to be able to make effective use of sensory information in the handling strategy generation, assessment and modification within a handling system.

In an effort to fill these gaps, this paper presents our work towards building such a system for food handling tasks. We use a Sun Sparc station as the host controller running the Poplog AI software environment. Integrating the host controller, a vision system and an industrial robot controller with different programming languages and working environments presents a communication handling problem. This problem is solved here by the development of control commands at different levels within the host programming environment. This has also provided a degree of flexibility unavailable within traditional robot programming [1]. In order to achieve a successful, robust and efficient grasp manipulation without introducing expensive hardware a vector formulation technique is adopted, and on-line visual information is integrated with target features extracted off-line to generate the actions required by the robot to accomplish the food handling task.

For food handling applications, the products to be handled and the grippers to be used constitute a special case

[1]Department of Manufacturing Engineering, City University of Hong Kong, Kowloon, Hong Kong
[2]Center for Intelligent Systems, Department of Computer Science, The University of Wales, Aberystwyth, UK

1070-9932/96/$5.00©1996 IEEE

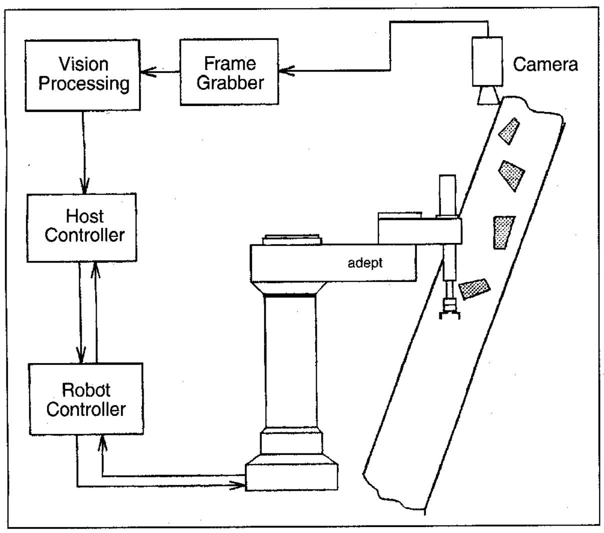


Figure 1. Schematic representation of the handling system

of constrained grasping. The problems associated with grasp deformations are presented. In order to be able to measure the quality of a grasp and predict its performance, grasp strategies are simulated and the resulting forces and torques studied. Experimental grasp trials are then carried out on the robot system to determine the effect of grasp deformations on the success/failure rate of a grasp strategy.

SYSTEM DESCRIPTION

System Components

A schematic representation of the visually guided robotic system for food handling is depicted in Fig. 1.

The work cell consists of the following parts:

- *Host controller*. The host computer is a Sun 4 Sparc station.
- *Robot system*. An Adept One robot is used as the manipulator for the food handling task.
- *Vision system*. The vision system used here is composed of some inexpensive general-purpose hardware and software components. This is not only for cost considerations but also to achieve the functionality and flexibility required which is not readily available with a standard robot vision system, e.g. Adept vision.

  The vision system consists of three parts:

1. Camera. The camera is a JVC color camera with 500 × 582 effective pixels of solid-state CCD elements. The camera is fixed 0.9 meters above the conveyer belt and is separate from the pick up location so that visual processing and robot actions can be carried out simultaneously.

2. Frame grabber. The frame grabber is an SBus board for use with Sun Sparc stations. It allows applications to be developed in an X-Window environment. Its 1.75Mb on-board memory enables images to be captured in real time with full processor access during the operation.

3. Image processing software. The image processing is carried out in C on the Sparc station. Use is made of an image analysis package, Visilog, which is compatible with C programs and allows easy development of new image processing functions as well as use of existing library functions. The vision algorithms developed include image processing functions needed for product identification, vector manipulation and grasp formulation. The software is also responsible for configuring and initializing the image devices concerned.

- *Grasping device*. The details of the gripper in use are given in a later section.
- *Conveyer system*. The system is designed for use with an industrial standard moving belt. In the present implementation, a constant and known conveyer speed is assumed.

The host computer is responsible for the high level control of the robot task. It receives information from both the vision system and robot controller. The vision programming is developed independently and the compiled vision codes are then loaded and linked to the main control program to be called as external procedures during run time. After working out the task requirements and extracting the environmental information via the vision algorithms, the high level commands are generated and sent along with the necessary data to the low level robot system. Upon receiving the high level commands, the robot controller formulates its own commands which are sent to the robot actuators to drive the manipulator. For most handling tasks, the goal locations are normally well defined. Therefore, a nominal robot path can be specified for part of the operation in the same fashion as in conventional robot motion by "teaching" or by explicitly specifying the robot end effector position and orientation at the destinations. Thus placing products at their goal locations consists of pre-defined operations using data generated off-line. For a known type of product, suitable grasp sites can also be derived off-line in the object space. However, products are normally presented to the robot in an uncertain manner and their positions and orientations in the robot space are rather random and need to be measured on-line. Therefore, the food handling operation consists of a mixture of on-line and off-line derived commands, which is facilitated via the flexible host control provided by the communication handling scheme.

Communication Handling

The communication handling scheme is aimed at establishing efficient communication between a host and robot controller. It should also facilitate the application of AI techniques and languages (e.g. Pop11) to robot control and allow easy interfacing of external sensing systems (e.g. vision) and signal processing modules to the system. For the current application, the host control is Poplog-based and must communicate with the VAL-based robot controller. This communication is achieved via a "coding and interpreting" scheme. However, an extra portion of time is spent on this process prior to sending executable commands to the robot joint controllers. In a typical "pick and place" operation when the robot speed was set to half of its rated maximum and the work span averaged about 116cm, the time spent on the communication handling ran to as much as 23.4% of the total time (2.52 seconds in this example) in executing the task (communication handling and robot action). This effect will become more significant in the following circumstances:

- *Smaller segments of motion.* In the above example, the span of the robot motion is relatively large. Practically, the smaller the motion segments, the less time is spent on robot action with more time spent on communication handling.
- *Higher motion speed.* The maximum speed in the above task was set to half of the rated maximum.
- *More degrees of freedom of motion.* The above task consisted of four-DOF motions. An additional half of the above communication time will be needed in the data transfer if a six DOF motion is involved.

To solve the above problem, a multi-level command structure has been developed, which permits optimized coordination between the high and low level actions by defining a communications protocol and applying control actions at different levels. We have defined and implemented the following levels for host control:

- *The primitive level.* The commands at this level can be regarded as equivalent to the original VAL commands of the robot controller which include:
  — User-program input and output directed to the controller terminal.
  — Disk commands and data transfers.
  — Robot status information.
  However, the *primitive level* is not limited to the existing VAL commands. In fact, it is the capability to allow more flexible host commands to be developed at the users' discretion that makes the communication handling strategy worth developing. For motion commands, the destination data as well as a command identifier have to be given and communicated to the robot controller. This level provides maximum flexibility for host control but at the cost of efficiency.
- *The intermediate level.* The commands at this level correspond to part of a complete robot action as dictated by some collection of VAL commands. The motion commands, for example, can have a fixed increment of translation and/or rotation in world coordinates. Thus only a command identifier needs to be specified and this provides more efficient communication.
- *The operational level.* By incorporating a set of instructions into a subroutine in a VAL program, a complete robot action for some simple task can be accomplished upon receipt of a single command from the host controller. Thus this level is the most efficient from the communication point of view, but it provides the least degree of flexibility for host programs.

Each layer supports a different range of functions. The functions of a higher layer can be performed by some lower level functions without affecting the nature of the action. The lower the level, the more flexibility is provided, but the less efficient is the overall operation. In this way the communications can be tailored to the needs of specific tasks, and the robot controller can receive the results of off-line computation undertaken on the host. The handling of the layered communication strategy is symmetric on each side of the system, as shown in Fig. 2.

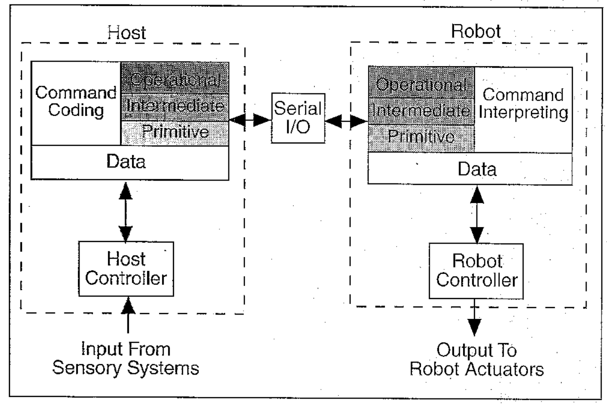


Figure 2. The communication scheme

For the same "pick and place" task as mentioned above, the time spent on the communication handling was reduced to 16.4% of the total time when the *intermediate level* was used and 9.6% when the *operational level* was used respectively. For a highly interactive robot task, when frequent information exchanges between the robot and the host control are required, a higher level communication handling is desirable. The communication handling strategy developed here is aimed at facilitating robot control by the host. Therefore the bulk of the commands flows from the host to the robot controller. The serial port, however, supports communications in both directions: to and from the robot side. For example, a number of robot status variables can be defined. On completing an action, the status variable can be assigned a value and sent back to the host. Thus the host can keep track of the task sequences carried out. In our implementation, however, we eliminated such back communications to reduce the handling cycle time.

VISUAL PROCESSING

Binary Imaging

Food products normally have very irregular surfaces, and variations of light conditions in manufacturing environments make grey level image processing difficult. To reduce complexity in processing the product images, binary imaging was used. The double-bound thresholding technique was adopted to transform a grey level image into a binary one. Thus an image pixel is given the value of 1 if its intensity lies between an upper and a lower threshold bound and 0 otherwise. The threshold bounds can be chosen by studying the intensity histograms for the products under consideration.

Figs. 3 and 4 show some examples of intensity profiles of a fish portion under different lighting conditions. The intensity profiles were obtained by choosing two points on each side of the image of the object and scanning along the line joining the two points and reading the grey level intensities of the pixels ("index" represents the counts of pixels starting from the first point to the second along the line). The first test was obtained using a special light table where the illumination is beneath the object. The second consisted of using normal fluorescent ambient lighting with the object on a light colored

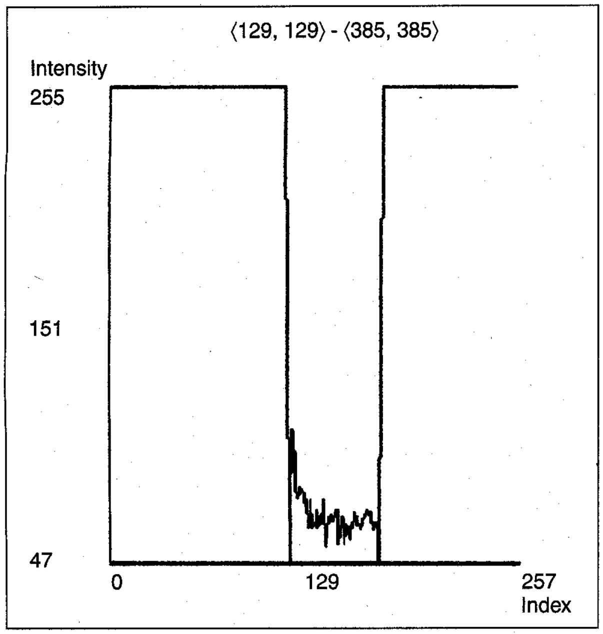


Figure 3. Intensity profile of a fish product using special light table

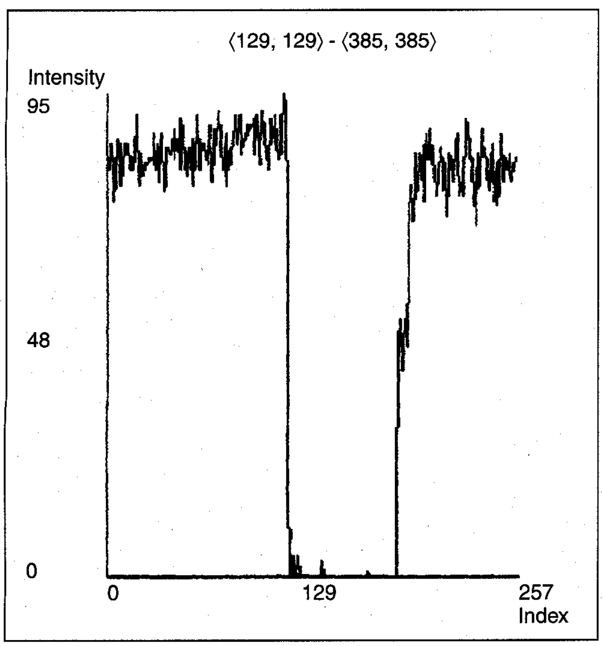


Figure 4. Intensity profile of a fish product using ambient light

background. Although the contrast in the latter case is not as good as in the former, a clear difference is still observed between the object of interest and the background. The results indicate that binary image processing can be used for the products concerned even without using any special lighting systems. The histogram in Fig. 5 shows clear distinctions of the intensities for the food product under normal fluorescent light.

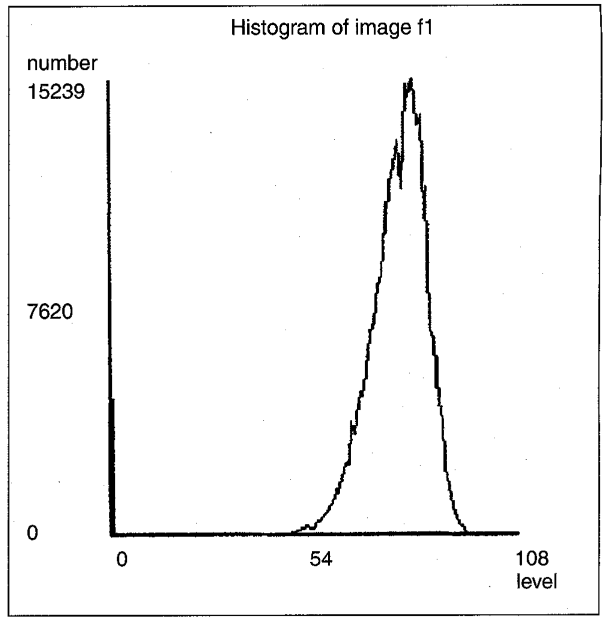


Figure 5. Histogram of a food product under normal lighting

Inertia Vector Measurement

Feature extraction methods for object identification and manipulation are often too complex and sensitive to measurement inaccuracies [2]. Food products normally exhibit high degrees of variability and irregularity at the surfaces and edges which we did not attempt to model in our product library. Figs. 6 and 7 show typical grey level images of a chicken and a fish product.

The requirements for reliability in manufacturing environments entail a robust approach in visual processing. A vector measurement technique was adopted to achieve this. We first define a vector to be measured — the Inertia vector $\mathbf{V}_I$. The origin of $\mathbf{V}_I$ is defined as the center of mass of an object and its direction is taken to be that of the minimum inertia axis of the object. Therefore an Inertia vector serves as a representation of an object's position and orientation. The visual measurement of the components of the vector is achieved using

Figure 6. The grey level image of a chicken product

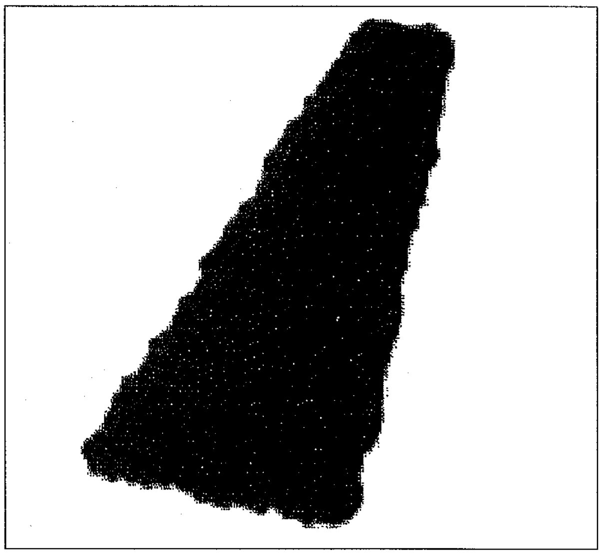


Figure 7. The grey level image of a fish fillet

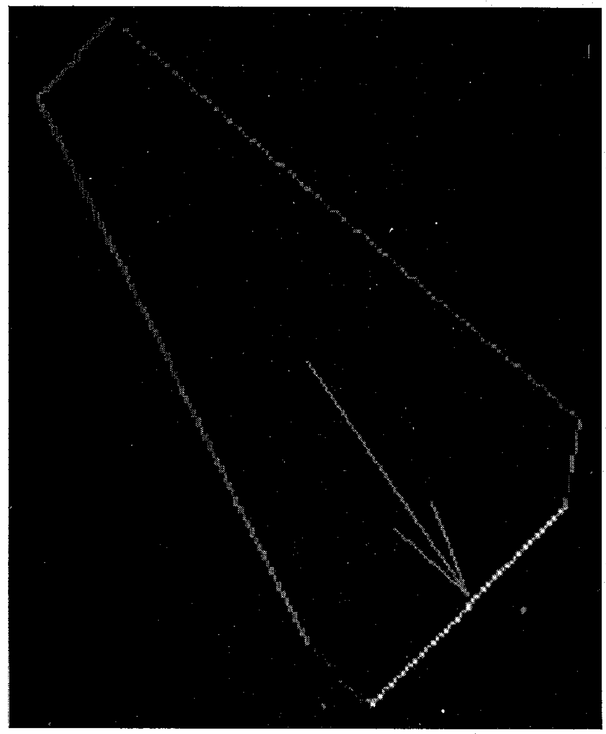


Figure 8. Inertia vector of an oval object

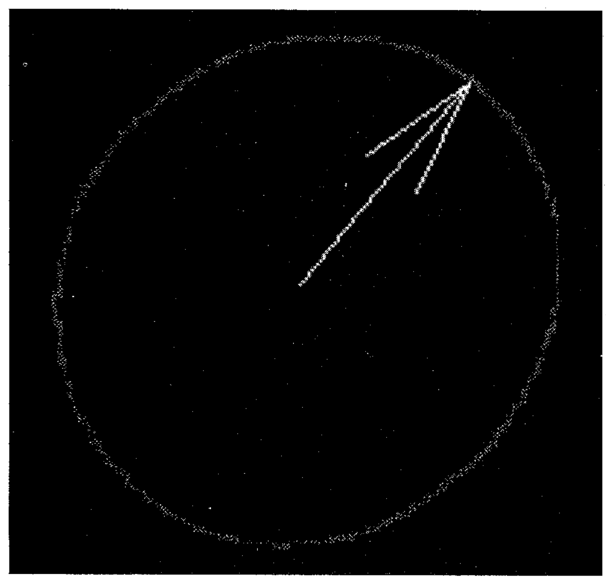


Figure 9. Inertia vector of a trapezoid object

the concept of moment of an object image [11]. The quantities derived on-line from vision, the center of mass and the direction of axis about which the object image has a minimum moment of inertia, determine the origin and the direction of an object's Inertia vector. As the vision algorithm integrates, on a pixel basis, over the whole region of an object image, it promises robustness to noise at individual points. Figs. 8 and 9 show the visually measured inertia vectors for an oval and a trapezoidal product. The Inertia vector is the basis upon which the grasp strategies are formulated.

A VECTOR APPROACH TO MANIPULATION

Definitions

In order to use vectors in the formulation of a grasping task, the following are defined:

- *Grasp polygon*. A grasp polygon is defined as the polygon whose vertices are the contacting points between the object and the gripper fingers. Point contact is assumed here. With a two-fingered gripper, the *grasp polygon* degenerates to a line which can be regarded as a rectangle of zero width.
- *Grasp vector* $\mathbf{V}_g$. The origin of $\mathbf{V}_g$. is the center of an object's grasp polygon. The direction of $\mathbf{V}_g$. points from this origin to one of the contact points. For asymmetric grasp polygons, this needs to be chosen by the operator, whereas in symmetric cases, the choice can be made arbitrarily as this does not affect the subsequent formulation.
- *Gripper vector* $\mathbf{V}_r$. This is the same as the grasp vector but is associated with the robot gripper instead of the object. Examples of the gripper vector for two, three, and four fingered grippers are shown in Fig. 10.
- *Deviation vector* $\mathbf{V}_d$. The origin of $\mathbf{V}_d$ coincides with that of the inertia vector $\mathbf{V}_I$ and it points to the origin of the grasp vector $\mathbf{V}_g$. Thus the length of the deviation vector is defined as the distance between the origin of the inertia vector $\mathbf{V}_I$ and the origin of the grasp vector $\mathbf{V}_g$.

Determination of Initial Inertia Vector

The initial inertia vector is needed to derive the deviation vector in the object target space. The formulation of the inertia

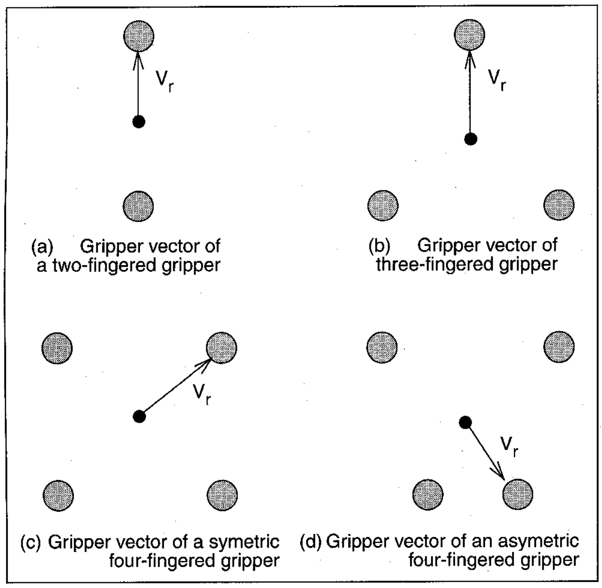


Figure 10. Examples of gripper vectors

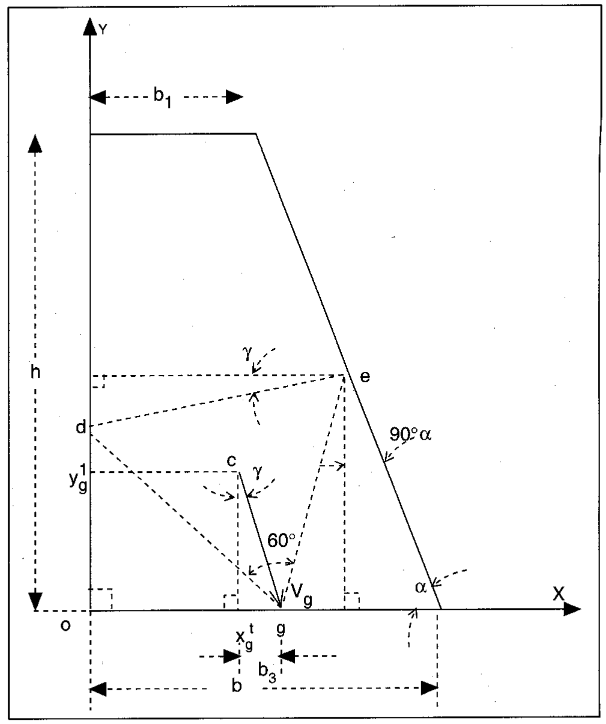


Figure 11. A trapezoidal object in its target space

vector can be accomplished off-line through analysis of product geometry and other known properties available. Given an object geometry, the center of mass and axis of minimum inertia can be computed with reference to a frame attached to it.

Initial Grasp Vector Derivation

The problems of finding optimized grasp polygons have been addressed by others [7, 10]. Both geometric and analytic approaches can be employed to achieve a stable grasp in terms of force or form closure [9]. This grasp location is the initial grasp vector in the target space. For a trapezoidal object, the initial grasp vector can be derived from the object geometric properties taking into account the gripper constraints to arrive at a sub-optimal grasp. The origin of the gripper vector is set at the center of the triangle of the gripper tips with one of the tips as the end point of the vector as shown in Fig. 11.

The grasp polygon and the grasp vector can also be obtained by taking geometric constraints into account [9]. In this case, the grasp vector in target space $\mathbf{V}_g^t$ can be given by $\{x_g^t, y_g^t, \gamma_g^t\}$ and the grasp polygon turns out to be an equilateral triangle as given by d – e – g. The formulation of the grasp vector in target space $\mathbf{V}_g^t$ is based on product model parameters and is carried out off-line.

Grasp Formulation

The task of grasping is to control the robot so that its gripper vector coincides with the grasp vector of the target. The objective of vector manipulation then is to estimate the vector error between the grasp vector $\mathbf{V}_g$ of an object and the gripper vector $\mathbf{V}_r$ of the end effector, and to control the robot in order to eliminate it.

Based on the above definitions, once an object is given, its inertia vector can be calculated in the target space as $\mathbf{V}_I^T$. When the gripper type is specified, the grasp vector can also be obtained in the target space as $\mathbf{V}_g^T$. Then the deviation vector can be derived in the target space, i.e. $\mathbf{V}_d^T = \mathbf{V}_I^T - \mathbf{V}_g^T$.

In actual implementation, the inertia vector and grasp vector are given in camera space as $\mathbf{V}_I^C$ and $\mathbf{V}_g^C$. By definition, the deviation vector $\mathbf{V}_d$, however, will remain the same regardless of the reference space. The invariability of the deviation vector gives

$$\mathbf{V}_g^C = \mathbf{V}_I^C - \mathbf{V}_d^T \, . \qquad (1)$$

This is the equation for deriving the grasp vector given the inertia vector from vision processing and the object and gripper parameters. Note that apart from the inertia vector which has to be generated on-line, the formulation above is dependent on the product only and can therefore be carried out off-line. A homogeneous transformation can then be applied to map the grasp vector from the camera space to the robot world space. Thus the desired grasp vector can be communicated from the host to the robot via a host control command (with the components of the vector as the destination data) as we have described previously. By comparing the desired grasp vector with its current gripper vector, the robot can formulate an error vector and compensate for it via its joint motion.

Through vector manipulation, part of the visual processing can effectively be carried out in an off-line stage and on-line resources are therefore saved. However, in the present implementation, no special visual processing hardware is used and the speed of the vision algorithm is limited by the host CPU. On average, the total visual processing takes about 0.9 second. In order to increase the throughput of the system without introducing expensive and dedicated hardware, visual processing and robot manipulation are carried out in parallel by separating the imaging and manipulating at different sites. With the Adept's permanent speed and an average work span of 38 cm, 32 products can be handled per minute for one-by-one handling (i.e. pick up and place one product each time). For stacked two-by-two handling (i.e. pick up two by stacking and place two each time), a handling rate of 45 products/minute is achievable.

GRASPING UNDER CONSTRAINTS

A Three-Finger Gripper

Satisfactory manipulations of objects require firm and stable grasp by the grippers. This is reflected in the formulation of force closure grasps whereby any external forces and torques can be balanced out by the grasp. For a 2D object, under the non-zero friction assumption, a three-finger force closure grasp exists. A special case of interest is when the contact forces intersect at a common point, which is the case for our three-finger gripper. This gripper is symmetric and the point of focus coincides with the geometric center of the finger tips as seen in Fig. 12.

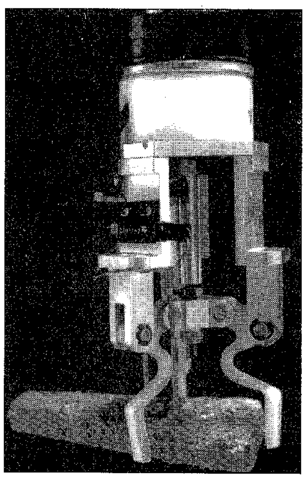


Figure 12. A three-finger gripper for food handling

A gripper designed in this way can achieve an optimal equilateral grip with minimum grasp effort required [8]. However, the gripper fingers cannot be controlled individually and the gripper has a binary open and close mode only. This limits the optimal formulation to the case of standard products. Any inconsistency from the standard case will give rise to deformations in the grasp due to the constraints of the gripper. As illustrated in Fig. 13, the finger tips must lie within a grasp ring, i.e. a circle of a designated radius (r_0) with tolerances Δr_{o1} and Δr_{o2} attached to it. The grasp radius r_o is manually adjustable. The finger orientation must lie within the grasp friction cone, i.e. a cone bounded by the friction coefficients.

Grasp Deformations

In an ideal case, each finger of the above described gripper would push against the object in the direction normal to the local object surface towards the grasp center, with all the finger tips lying equally spaced on the same circle. Practical grasp situations will differ from the idealized case, which will cause grasp deformations. For our three-finger gripper, the following deviations are commonly encountered during a grasp:

- Grasp center deviation from the center of mass of an object. Actual grasps involve deviations not represented by the *deviation vector* given in the previous section.

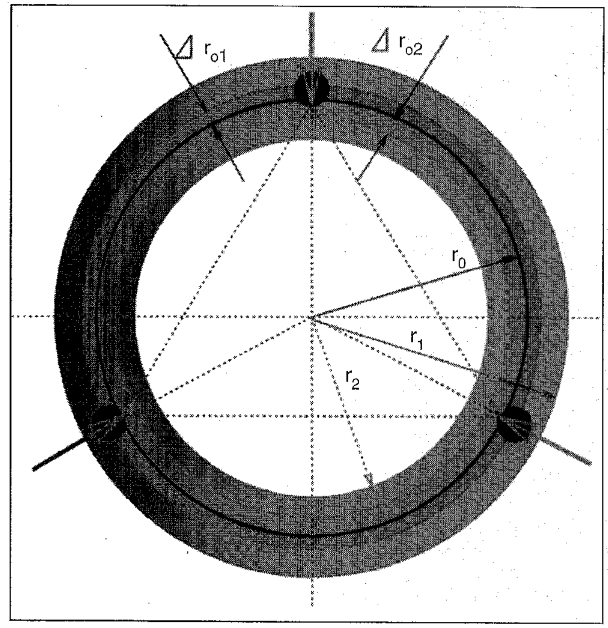


Figure 13. The configuration of three-finger gripper

- Grasp radius deviation (d_r) from the desired radius of aperture when the gripper is closed.
- Gripper finger deflections ($d\theta_v$) in tangential directions.
- Local deviations of the orientation of the finger tips to the contact normals of the object surface ($d\theta_e$). This is a constraint imposed by the object and it dictates the contact deformation from the ideal case when the finger pushes perpendicularly inward.

The last three kinds of grasp deviations are illustrated in Fig. 14 for a single finger.

Grasp Performance Evaluation in the Presence of Deformations

Actual grasps during food product handling will involve one or more of the deformations we have described. For a given object and gripper, the *grasp vector* is derived based on some theoretical optimal criterion. Gross approximations often have to be adopted in both the object and the gripper models and the theoretical criteria tend to be subjective. Actual grasp configurations are bound to deviate more or less from the modeled case because food products normally exhibit a high degree of variability. These variations cannot be taken into account in the current implementation using the vector formulation as a fixed *deviation vector* is assumed based on standard product geometry during grasp formulation at the off-line stage. The grasp deformations will result in force and torque acting on the object and disturbing the grasp, giving rise to grasp stability problems.

Although an obvious solution to the grasp stability problem lies in sensing and controlling the grasp force, hygiene requirements and cost considerations imply that using force sensing/control in the grasping devices will not be readily accepted by the food industry. It is therefore of practical inter-

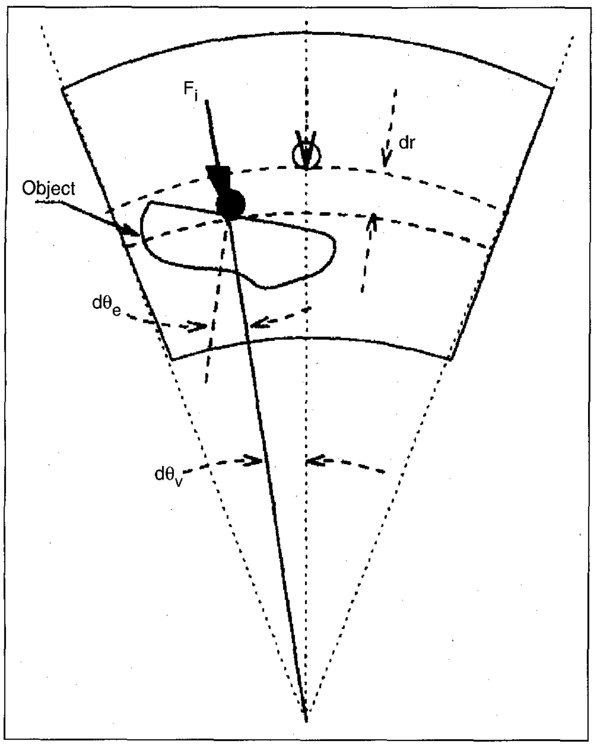


Figure 14. Contact deformations from the ideal case

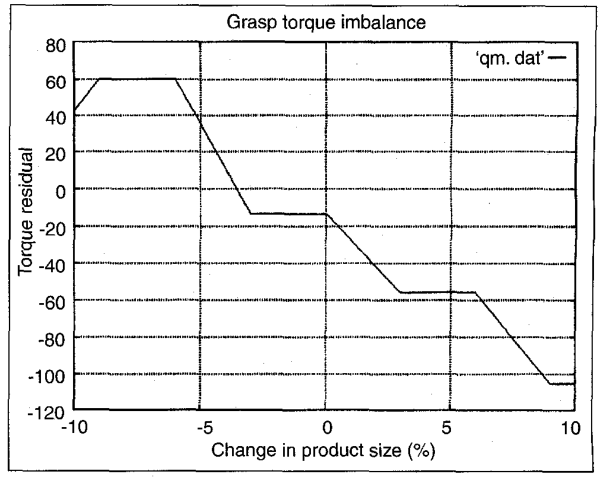


Figure 15. Imbalanced torques

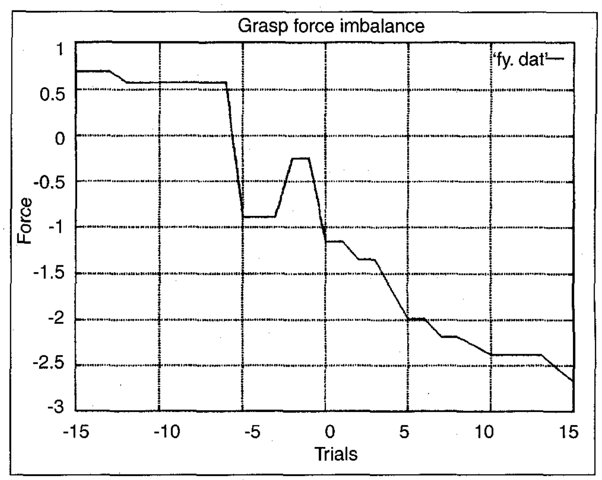


Figure 16. Imbalanced grasp force

est to be able to predict grasp deformations using vision sensing alone to gain some insight into the performance of a deformed grasp prior to its execution. This is provided by the system's programming environment which allows simulations to be performed using real product images, "testing" grasps using the gripper model and outputting a performance index which is to be associated with the success/failure rate during subsequent real tests.

An example of the theoretically derived grasp vector for a trapezoidal product has been given in Fig. 11 as $\mathbf{V}_g$. As product size and/or shape vary, its center of mass and orientation will change accordingly. A grasp configuration is therefore forced to deform due to the changes in the on-line measured inertia vector of the object. Imbalanced torques will then result due to such changes as shown in Fig. 15 when the shortest side b_1 of a trapezoidal shape product varied by $\pm$ 10% of its standard length of 3cm. The other two sides were fixed at h = 13.15 cm and b = 7.8 cm. The level of the resulting grasp moment imbalance increased almost linearly with the size variation.

Similarly, imbalanced grasp forces will result when product size varies. Fig. 16 shows the imbalanced force in the y direction when the short axis of an oval shaped product (chicken-burger) varied from -15% to +15% of the standard size of 7.5 cm in the short axis and a fixed length of 11.2cm in the long axis. The result of the imbalanced force/moment will disturb the object position and orientation and therefore the planned grasp configuration. A transition stage results that will bring the grasp to a new configuration. This constitutes a cause of grasp failures due to the simple driving mechanisms that are in wide-spread use for industrial grippers in manufacturing. Due to the binary driving mode of the current gripper, there is no active control over the speed or position of the fingers and each finger simply strikes the object at full speed until balanced by the counter actions of the other fingers. The imbalanced forces and torques therefore provide an indication of the grasp performance and the corresponding success/failure rate of a deformed grasp.

Experiments were performed using the binary driven gripper without force sensing but with a presence/absence detection sensor mounted on the gripper to record the success/failure rate automatically during batch trials. Each batch in the experiment consisted of 500 trials and was carried out by program control which automatically repeated the visual extraction, grasp formulation and simulation, and robot action cycles and recorded any failures during the runs. For products of standard sizes and shapes, the achievable failure rates were better than 0.1% for optimal grasp strategies derived off-line. As an example, implementation tests showed that an imbalanced force of 2.3N in the y direction of the oval product would increase the failure rate to as much as 0.6%, suggesting that a change of 7% in the product size should lead to its rejection in product handling operation.

Similarly, the grasp simulation is useful in providing information on the possible damage to a product that may result from an improper grasp or product variation. Steady state contact force between each individual finger as well as the initial contact force can be simulated in this respect. Based on the simulated forces and the vulnerability of the product concerned, a risk index can be designed by which grasps are judged and those with unacceptable risk index are prevented from being executed.

Theoretically the solution to the problem of grasp deformation lies in modifying the deviation vector and therefore the grasp vector using the on-line visual information and incorporating force and moment closure conditions into the grasp formulation. However, this is a computationally expensive approach. Furthermore, there are other factors apart from force/torque that are difficult to account for in a theoretical formulation. A compromise approach is being identified through a learning process, which is one of the objectives of our future work.

CONCLUSION

In this paper, we described our work in developing a visually guided robotic system for food handling applications. A communication handling scheme was developed to provide the flexibility unavailable with traditional robot programming. Special attention was paid to robustness considerations in the visual processing component and a vector notation and manipulation method has been adopted for the grasp formulation. By the vector approach, visual information extracted on-line can be combined with off-line generated initial vectors to formulate a grasp for the product handling. The manipulation scheme proved suitable for handling large batch products with limited variations. For products with variations exceeding these limits as is often the case with food products, the effect of product variations on the success/failure rate of handling needs to be studied. This is carried out by investigating constrained grasping through simulation and experimentation in an off-line stage. The insight gained during off-line performance evaluation can be used as a basis for predicting the success/failure rate of a grasp strategy in consequent on-line stages for the product under study. This approach will also benefit automatic reconfiguration of the system for handling different types of product

ACKNOWLEDGMENTS

This research is supported by the Engineering and Physical Science Research Council through grant number GR/H 85397. The authors are grateful for the support of Unilever Research Colworth Laboratory and their colleagues J.J. Rowland and M.A. Rodrigues.

REFERENCES

[1] V. Hayward and R. P. Paul. Robot manipulator control under Unix RCCL: A robot control C library. *Int. J. Robotics Research*, 5(4):94-111, Winter 1986.

[2] N. Hyache and O. D. Faugeras. Hyper: A new approach for the recognition of two-dimensional objects. *IEEE Truns. on Patten Analysis and Machine Intelligence*, PAMI-8(1) :44-.04, Jan. 1986.

[3] T. D. Jerney. Replacing touch labour with robots in food packaging. In *Proc. Int. Conf. Robots and Vision Automat.* pages 81-88, Detroit, USA, Apr. 1993.

[4] M. Kassler, P. I. Corke, and P. C. Wong. Automatic grading and packing of prawns. In *Proc. of Int. Symp. on Industrial Robots*, pages 9.27-9.32, Detroit, USA, Oct. 1990.

[5] K. Khodabandehloo. Robot handling and packaging of poultry products. *Robotica*. 8:285-997, 1990.

[6] D.E. Malone, W.E. Friedrich, N.R. Spooner, and P.P.K. Lim. Knowledge based control in the processing of highly varying products. In Proc. *IEEE Int. Conf. Robotics Automat.*, pages 2903-2908, San Diego, USA, May 1994.

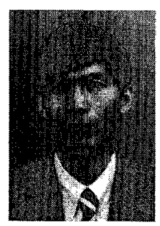
[7] X. Markenscoff and C.H. Papadimitriou. Optimum grip of a polygon. *Int. J. Robotics Research*, 8(2):17-29, April 1989.

[8] B. Mirtich and J. Canny. Easily computable optimum grasps in 2-D and :3-D. In *Proc. IEEE Int. Conf. Robotics Automat.*, pages 739-744, San Diego, USA, May 1994.

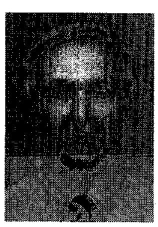
[9] V.D. Nguyen. Constructing force closure grasp. *Int. J. Robotics Research*, 7(3):3-16, June 1988.

[10] J. Ponce, D. Stam and B. Faverjon. On computing two-finger force-closure grasps of curved 2D objects. *Int. J. Robotics Research*, 12(3):263-273, June 1993.

[11] A. Rosenfeld and A.C. Kak. *Digital Picture Processing*. Academic Press, 1982.



Youfu Li received the B.S. and M.S. degrees in electrical engineering from Harbin Institute of Technology, China, and The PhD degree in engineering science from the University of Oxford in 1982, 1986, and 1993 respectively. From 1993 to 1995 he was a postdoctoral research associate in the AI and Robotics Research Group in the Department of Computer Science at the University of Wales, Aberystwyth. He is currently an assistant professor in the Department of Manufacturing Engineering at City University of Hong Kong. His research interests include real-time sensor-based control, vision guided manipulation, virtual reality, flexible assembly systems and intelligent manufacturing.



Professor Mark H. Lee is Director of the Centre for Intelligent Systems in the Department of Computer Science, University of Wales, Aberystwyth, UK. Previously a Lecturer and Senior Lecturer in Computer Science, Prof. Lee was awarded a Personal Chair in 1987. He has spent sabbaticals at the University of Massachusetts, Amherst, USA, and the University of Auckland, New Zealand. He is BSc, MSc (University of Wales) and PhD (University of Nottingham). He is a European and Chartered Engineer, a Fellow of the Royal Society of Arts, a Fellow of IEE and Member of ACM. Prof. Lee is also a member of many editorial and advisory boards and has published over 80 technical papers and a book on robotics. The Centre for Intelligent Systems researches into advanced intelligent software to support engineering. Funding currently is mainly from UK Research Councils, with heavy industrial collaboration from automotive, software and engineering companies.