An Approach to the Integration of Vision and Touch for Robot Control

R. E. Ellis

COINS Technical Report 85-20

Laboratory for Perceptual Robotics
Department of Computer and Information Science
University of Massachusetts, Amherst MA 01003

ABSTRACT

Advanced, flexible robotic systems require multiple sensors to successfully perform tasks in changing or unfamiliar environments. The data from these sensors must be integrated dynamically, in a coherent fashion, if the robot is to respond promptly and appropriately and complete its assigned task.

This paper describes a methodology for the integration of two robotic senses – vision and touch. Integration is performed by augmenting the usual *structural* models of the object in the robot's world with *functional* information. These functional attributes, which describe how an object relates to other objects in the world, facilitates the process of inferring from multiple sensors and thus in controlling the robot's motions.

This report is a paper to be presented at the IFAC Symposium on Robot Control in Barcelona, Spain, November 6-8, 1985. Publication as a COINS Technical Report was recommended by Prof. Michael Arbib.

§ 1.0 Introduction

Our research on the dynamic integration of data from multiple sensors into the control of a complex robot manipulator is based upon three principles: that systematic integration of multiple sensors can be achieved by dealing with task-related features extracted from the sensors, rather than with the raw sensor data; that this integration requires object models of a novel sort, incorporating functional knowledge of the way the objects in the robot's world (including the manipulator) can interact with each other, as well as standard structural (geometric) knowledge; and that incorporation of multiple sensors proceeds best by using focus-of-attention strategies to accomplish the robotic task.

The methodology described below is applicable to integration of many different kinds of sensors; its feasibility is being tested by concentrating upon visual, tactile, and force sensors. These sensors are chosen because they are sufficiently distinct as to present interesting problems in integration, and yet techniques for dealing with them separately are already well established. Described in turn are our general approach to the problem of multisensor integration, the issues which must be addressed by any system that integrates multiple sensors, and a brief analysis of some of the problems which arise in multisensor integration.

§ 2.0 Integration of Sensory Information

Multisensor integration is the result of inferring a low-level feature, or an item of higher-level knowledge, which cannot be reliably and efficiently inferred from the data provided by a single sensor. Multisensor integration thus includes low-level combination of information from sensors; using several sensors simultaneously to achieve a goal; and the successive use of several sensors to achieve a goal.

Two dimensions of the problem are the level of abstraction at which the inference is conducted (in the present case, "low" and high" are all that are distinguished), and whether sensors are used simultaneously or successively in various combinations. Four simple examples which display various kinds of integration are:

- Low-level, simultaneous use of joint-angle position and tactile analysis to locate a tactile feature, e.g., an object edge, at a particular position and orientation in three-dimensional coordinates.
- Low-level, successive use of vision and touch to roughly locate an edge and verify its properties, e.g., finding the radius of the edge in addition to locating it.
- High-level, simultaneous use of vision and touch to detect slippage of a part during a transfer operation.
- High-level, successive use of vision and touch to locate a hole and then compliantly insert a peg into it.

The distinct issues to be addressed depend upon the level of integration proposed.

In general, low-level (or feature-level) integration requires special-purpose algorithms that often must be sensor-dependent. In particular, successive feature-level integration may require knowledge not only of the sensors but of the manipulator or even of the object being sensed. In the above examples, simultaneous integration requires a model of the manipulator and of the tactile sensor, while successive integration requires that the kind of information (in this case, an edge) be definable and verifiable in both sensor domains. Such requirements are not always easy to meet.

High-level (or knowledge-level) integration employs methods that are of more general use than does feature-level integration. In order for information from several sensors to be successfully integrated, it must be available to algorithms in a form that can be readily accessed. The present research concentrates upon knowledge-level sensor integration, and a modified object model is proposed as a data structure that facilitates the transfer and integration of multisensor information.

In order to accomplish a robotics task, models of the world and the objects in it must be used. What are proposed here are unified models: they involve not only structural information (e.g., the planar faces of an object's surface), but also functional information (e.g., how the object may be picked up and fit into another object). An example of the overlap and interchangeability of the two representations is a threaded fastener. A bolt may be described as a solid helix with an octahedral head, or as an object which may be put in a wrench, fit in a hole, twisted, and is then not extractable by translatory motion. Exactly which representation is used is a pragmatic decision, because a sufficiently powerful reasoning system might be able deduce one from the other, given certain facts about forces and shapes; but once provided with a unified object model, it becomes very easy to monitor the execution of many tasks.

Unified models are especially useful in multisensor integration because they are a place in which all of the attributes of an object – structural, sensor-based, action-oriented, procedural, or otherwise – can be represented in a readily accessible manner. For example, the final torque to be applied to a bolt in a fastening procedure is both an attribute needed to act (twist), and a feature to sense for purposes of feedback; the unified model provides a mechanism for expressing the disparate pieces of information in a manner useful to other parts of the system.

Described below are examples of sensors,, features, and actions; what a task plan looks like; what an object model is; and a few example task plans to indicate the power and limitations of this approach to integration of multiple sensors into the control of robot motion.

§ 3.0 Sensors, Features, Actions, and Tasks

During task execution, features and actions are the basic units by which information about the environment comes to the control system, and by which the environment is changed by the control system. These are abstract units in that they are ways of describing what is happening, but are not physical devices (as are sensors and actuators).

A **sensor system**, usually abbreviated as just 'sensor', is the hardware needed to transduce from an environmental property to computer-readable form. A sensor is thus a combination of a transducer and some sort of hardware interface. Examples of sensors are interfaced cameras, rangefinders, tactile and force sensors, thermal sensors, arm position and velocity sensors, and even simple limit switches on a gripper.

A **feature** is a property inferred from the information provided by one or more sensors that may be used to reason about the environment. (In order to reduce ambiguity, 'extract' will be used heareafter in place of 'infer'; the latter term will be used in talking about models and how one may reason about them.) Examples of features are reflectivity of a patch, a visual edge, a tactile corner, the radius of a tactile edge, the barycentre of an object that is being gripped, or the position and orientation of a link of a robot arm. Some of these, such as the edge radius, can be best acquired from dynamic sensing by multiple sensors; others, such as the visual edge, can be extracted from a single sensing incident by a single sensor. Features thus may be specifically associated with a particular sensor or sensor type; or they may be the result of integration of several sensors, or of the data from a single sensor over time.

An **actuator** is a physical, computer-controllable device which can produce some change in the environment. Obvious examples are the motors or hydraulic cylinders on a robot arm; less obvious ones are the mechanisms used to focus or move a camera. The robot itself can be viewed as an object in the environment that can be sensed and moved, and that has further effects on the environment. An **action** is an operation that may be performed on the environment. These can map in a simple manner onto actuator changes, e.g. movement of a single link of an arm; or they can be quite complex, e.g. insertion of a peg into a hole, where the peg is held in a gripper via a remote-centre compliance wrist.

Features, then, are useful ways of dealing with information from multiple sensors, especially when the properties of some *particular* sensor are irrelevant to the accomplishing of the task. The "logical sensors" of [Henderson *et al.*, 1984] are similar in concept; indeed, both approaches are basically aspects of sound systems design, in which clean interfacing of modules is an aid to abstraction, and eases production and maintenance of software.

§ 3.1 Task Plans

A task plan is a set of instructions (including those for sensory monitoring of the environment) for accomplishing a higher-level goal. When multiple sensors are available in a robot system, a framework for integrating them is needed. Task plans provide such a framework by specifying when and how the information derived from a sensor is needed, and how it is to be used.

Execution of a task plan occurs within some limited context. For example, if the task is to move the robot arm from one configuration to another, the execution module does not necessarily know what else is in the workspace; if a failure occurs, the current task plan may no longer be applicable, and some error-recovery procedure (such as re-planning the task) will have to be performed. This view is proposed in order to circumscribe the problem of monitoring the execution of the task plan; the planner has considerable knowledge of the environment, and attempting to control the execution and simultaneously attend to great detail is computationally expensive. Tasks can often be broken into distinct, clean sub-tasks, and the execution stage is a convenient place to implement such a breakdown.

The applicability of a task plan is delimited by **parameters** and **envelopes**. A parameter is simply something which varies, and the variations of which can be sensed. Examples are the position or velocity of a robot arm joint; the presence of a tactile feature; and the spatial distance between the centroid of two world objects. Associated with the parameters are envelopes, which are abstract surfaces that partition the parameter space into two equivalence classes: acceptable and unacceptable. Examples are positional limits of the robot joints, or the visually-extracted distance between a part and the gripper.

Task plans are goal-directed to a significant degree. Many robot tasks are well-constrained, with reasonably accurate models of parts and their approximate place in the environment. This permits execution of the task to be interspersed with sensing checkpoints, at which various features are expected and whose absence indicates deviation from the plan. Provided that the parameter envelope has not been exceeded, the task then specifies actions that may be performed to ensure proper execution.

§ 4.0 Object Models

A broad distinction in object modelling can be made between *structural* models, which describe the object's physical structure, and *functional* models, which describe how the object interacts with the environment (including a robot system). Our system combines these structural and functional descriptions into a single, *unified* object model. Unified models simplify the integration of information from multiple sensors by providing an intermediate representation, in which multisensor reasoning pertaining to a particular object in the world can be readily expressed.

§ 4.1 Structural Models

Structural models are those which provide a description of the geometry of the object. [Requicha, 1980] gives an excellent discussion of solid modelling methods for rigid objects. Some of the methodologies that might be used to represent objects in robotics are: constructive solid geometry; boundary methods, such as surface patches [Faugeras, 1984]; sweep methods, including generalised cones [Brooks, 1981]; and spatial occupancy enumeration, particularly octrees [Connolly, 1984].

The preferred structural modelling methodology depends upon the task domain, and upon how the model will be used. Our initial requirements have been rapid and convenient identification of surface and volume properties (local surface normals, axes of symmetry, etc.), and ease of matching incoming features to the model. Uniqueness, verification, and 3-D rotation for graphics display purposes are of much less importance. After evaluating various structural modelling styles by these criteria, a simple boundary model was selected as best meeting our needs.

Our structural model of an object is a set of polygonal faces, edge segments, and vertices. The main advantage of this representation, for our purposes, is that an interpretation/verification scheme can be used with tactile, visual, or direct range data. This facilitates testing of all combinations of low-level and high-level sensor integration, both simultaneous and successive.

Our system is an extension of [Grimson and Lozano-Perez, 1984]. Their system represents objects as a set of convex polygonal faces, and recognizes objects by employing 3-D sense data and applying simple, local geometric constraints. We have extended their work by representing objects in terms of linear edges, vertices, and texture patches in addition to polygonal faces. Edges and vertices provide elegant and powerful geometric constraints to the recognition and localization procedures; because our tactile algorithms extract the 3-D features by distinguishing among planar, edge, and point contact, little additional computational overhead is required. Preliminary analysis indicates that the combinatorics of recognition are not worse than, and are usually much better than, those of a model employing only faces.

The model can be readily modified to represent other sorts of data. For example, a patch of visually or tactilely observable texture can be hierarchically attached to a face, or to a set of faces, and thus be used to either verify that a given face is in a particular position and orientation, or to recognise that several points must be constrained to lie on the same face.

This model does not require that the boundary primitives form a closed surface. For instance, if a face is not observable – such as the interior face of an open but opaque bottle – then there might be no need to include it in the model instance. It is not possible to deduce volumetric information from this kind of model, so such information would have to be separately; but parsimony is, in this case, secondary to convenient access to the information.

§ 4.2 Functional Models

Functional models represent the object in a very different way, and are much less explored. An example from psychology is Piaget's description of children's models as *practical*; in these, the object in question seems to be known more by how it may be felt and moved than by how it appears. Computational examples of functional modelling include those of [Winston *et al.*, 1983], where 'cup' is partly defined as a liftable thing that holds certain other things; [Weymouth, 1985] also has a partly-functional model in that part of the definition of 'house' includes procedures for recognising a house, i.e. the way that a house interacts with a computer vision system.

It might be argued that many robotics systems already have limited, implicit functional models of objects. For example, a description of how to torque a bolt into a casting is also a description of one aspect of how a bolt can interact with certain other objects (in this case, appropriately threaded holes). The limitation of this is that generality and systematisation of the representation are lacking, and so it is hard to build up a coherent description. Some schemes are slightly more explicit, e.g., in [Arbib *et al.*, 1984], the grasp is shaped according to the type of handle of a mug, so that their schemas include both sensory parameters passed by vision and touch and task-related parameters for the control of movement.

§ 4.3 Unified Object Models

A unified object model is one in which both structural and functional attributes are present. It is entirely possible that a system with sufficient reasoning power and a sufficiently large knowledge base could deduce structural information from functional, and *vice versa*. This, however, seems impractical. A unified model is proposed not on the criterion of conceptual parsimony, but rather on the criterion of pragmatism. Deductions from thread descriptions to torques happen to be well-explored; such is not the case for grasping, which is still an open area of research.

The instantiated object model serves as a local blackboard for the integration of features extracted from the multiple sensors. Some processes perform the specialized reasoning from existing object attributes and incoming sensed features, and add or modify the object attributes dynamically. These are complemented by other processes which reason from existing attributes to robot actions. The functional attributes of the object model provide a clean interface between the sensor-based features and the robot actions, facilitating the integration process. Invocation of the processes in this system is guided by the goal-directed task plan.

Unified models have the advantages of generality and utility in control and assembly regimes, and of providing a systematic way of adding new sensors or feature extractors into a high-level controller. Their disadvantages are redundancy of representation (with the consequent problems of maintaining consistency), and difficulty in reasoning about functional aspects.

§ 5.0 Using Multiple Sensors in Robot Systems

The basic problem which we address is how visual, tactile, and other sensors can be dynamically integrated into a robot system so that the system can perceive its environment, and thus accomplish a task in the face of incomplete knowledge of, or unexpected changes in, the environment. Our claim is that by employing features extracted from the multiple sensors in a goal-directed manner, and using these features in conjunction with unified object models, a robot system can dynamically update its world model and respond to the new information so as to accomplish its task.

The focus of our research is on the problems of object identification, object acquisition, slip detection, mating of parts, and recovery from detected errors. Initial investigation of this approach has been done mostly with a fairly simple experimental setup in which the parameters can be widely varied. Variations of "peg-in-hole" problems can serve, if they include identifying, grasping, and (of course) mating the parts. Although the peg-in-hole problem is an apparently simple one, it embodies the important problems encountered in many industrial tasks, and the results from our research into this problem should be directly applicable to more complex assembly problems.

Our experimental setup consists of a plate with various sizes of square, round, and hexagonal holes, and pegs of appropriate shapes, sizes, and fitting tolerances. Varying the parameters of the setup permits investigation of a wide range of problems, e.g., similarity in the pegs tests identification and acquisition techniques, fitting tolerances of thepegs in the holes tests the insertion procedure, and changing the surfaces of the gripper and pegs changes the nature of slip and its detection.

The research issues which we are addressing can be illuminated by examination of a simple example. Consider the task of fitting a peg into, say, an hexagonal hole of known dimension, location, and orientation. The basic steps of the task might be:

1) Find all pegs which might fit the hole, using stereo vision.

2) If more than one target remains, examine them in turn using touch until a suitable one is found.

3) Grasp target peg for transfer and insertion, and verify the grasp.

4) Monitor the transfer for peg slippage or collision with obstacles, using vision, touch and force.

5) Insert peg in hole; monitor using touch and force, according to the closeness of fit of the peg.

This apparently modest example provides an excellent illustration of the above issues. These issues are pervasive, and can best be examined by elaboration of each step of our example (keeping in mind the relation between the steps of the example and the issues).

1) *Find all pegs which might fit the hole, using stereo vision.* An algorithm for scanning the workspace, segmenting the depth map into objects (which can be simplified by knowing in advance what objects are being sought), and finding the location and orientation, or pose, of each peg. The pegs must be measured (within stated error bounds) to eliminate inappropriate sizes; measurement is a function of sensor accuracy, the algorithm, and environmental parameters such as distance from the sensor. This step is deemed to have failed if no pegs are visible, or if none have been found to be the correct size (within the error bounds). In the latter case, recovery could be either to re-scan with another algorithm (a peg may have been missed), or to attempt to measure the pegs with another algorithm.

2) *If more than one target remains, examine them in turn using touch until a suitable one is found.* Execution of this step is performed only if: vision has been unable to measure and classify pegs with sufficient accuracy; the pose is known and can be achieved by the robot arm; and a sensing location for distinguishing the object is known or calculable. An excellent approach would be to identify, in the object model, areas that are useful in such a task (such as the peg end, which is unique for each peg type). There is a trade-off in the quality of the sensor, the accuracy of the algorithm, and the part of the peg to be touched; e.g., if the sensor is very dense and sensitive, many more distinguishing points might be available. Another trade-off is generality of the algorithm, in which tactile features are matched against the instantiated attributes of the object model, against a specific algorithm designed to accomplish this particular task with these particular objects.

3) *Grasp target peg for transfer and insertion, and verify the grasp.* The grasp points may already be known; however, if the model is only partially instantiated and no grasp points have yet been identified, they could be calculated and inserted during this step. The model would contain a pre-defined parametrised procedure for grasping that leads to transport and insertion. The grasp point is a functional attribute of the model which can be inferred from

the model's structure, the attributes of the gripper, and which can be examined and verified by multiple sensors. Grasp can be verified by using touch and vision concurrently; this is an instance of the perception being highly focussed and goal-directed, Using the wrist sensor, a simple manipulation can establish the mass and first moments of the mass distribution. Failure to grasp properly might result either in another attempt to grasp or, if repeated often enough, to aborting the task. Advanced error analysis could involve determining if the grasp point was poor for this peg in this pose, and hence the selection of an alternative grasp point.

4) *Monitor the transfer for peg slippage or collision with obstacles, using vision, touch and force.* Detection of slippage is, in itself, a major topic for the proposed research; rather than attempting to design special-purpose slip sensors, we will take a multi-sensor approach to its solution. We believe that slippage of the peg can be deduced from orientation and position changes of the peg by touch; sudden acceleration detected by a force sensor can indicate slippage or collision; and a highly-focussed vision procedure can observe slippage and collision, because the manipulator path is known and the object well modelled. (Real-time implementation of these monitoring techniques is dependent upon the hardware available.) If the slippage is minor, the task can continue; if it is moderate, improving the grasp (see step 3) may be the answer; and if slippage is severe, the peg must be located and grasped again (steps 1 and 3, modified by the knowledge that this is the correct peg).

5) *Insert peg in hole; monitor using touch and force, according to the closeness of fit of the peg.* The insertion procedure (parameterised by the closeness of fit) will be a special-purpose procedure attached to the object model; the features it uses are dependent on its algorithm. The insertion is not a general-purpose one because object shape has an effect, e.g., the causes and cures for jamming of round pegs and hexagonal pegs are different. There are significant problems with using vision or direct-ranging sensors in this task; occlusion and accuracy given likely values of distance and perspective parameters must be dealt with for these sensors to be useful in controlling approach and insertion.

Initial testing of multisensor integration, using this experimental setup, has been with a combination of our existing equipment and simulation.

The location and orientation of the peg on the worktable is found from stereo vision, using the method described in [Ellis, 1985]. This method produces a sparse depth map of points on the object; the map is sufficient for acquisition by a simple gripper, and can be used to determine that the object is elongated, but in typical viewing scenarios is not a dense enough map for identifying which of several pegs has been located. Hence, the object is grasped with a parallel-jaw gripper, which is instrumented with a tactile array sensor developed in our Laboratory [Begej, 1985].

Once the object is touched, a great deal of information is available. As described in [Ellis, 1984], we can extract vertices, lines, and face features (as well as texture features, if the peg is in any way textured). Once the position of these features on the array is known, their three-dimensional position and orientation can be deduced from the kinematic structure of the gripper and arm.

These elements can then be used to deduce the type, location, and orientation of the peg (within symmetry classes) given errorful data. The pegs are simple in geometry, so deducing an insertion grasp is not difficult; the grasp is verified by extracting the orientation of the major axis of contact on the tactile array.

The transfer procedure is monitored by continuing to extract the major tactile axis, and by watching the worktable to determine if a new object, i.e. the peg, has been introduced. The insertion procedure is elementary – the principal problem has been lack of adequate force sensors, so insertion is simulated rather than real.

This partly simulated system has shown that for simple tasks, multisensor integration can be achieved with a unified object model. The task plan indicates which features are necessary at each stage, which sensory processes are needed to deduce structural or functional attributes of this instance of the object, and which motor processes are required to produce the desired physical actions. Monitoring of error conditions occurs throughout.

These experiments have shown that attachment of simple procedures and functional features to a structural model permit ready development of goal-directed, multisensor robot tasks. Our future work will be the improvement of these methods, extension of the system to include other sensors, and further analysing the ways in which high-level integration of multiple sensors can be achieved in robotics.

§ 6.0 Acknowledgements

This research was supported by the National Science Foundation under Grant ECS-8108818, and by the Office of Naval Research under contract N00014-84-K-0564. Leslie Kitchen, Stefan Begej, George Reynolds, and Michael Arbib provided valuable comments on this paper.

§ 7.0 References

Arbib, M.A., Iberall, T., and Lyons, D. 1984: "Coordinated control programs for movements of the hand", *Exp. Brain Res. Suppl.* (in press).

Begej, 1985: "A tactile sensing system and optical tactile-sensor array for robotic applications", Technical Report 85-06, Department of Computer and Information Science, University of Massachusetts.

Brooks, R. A. 1981: "Symbolic reasoning among 3-D models and 2-D images", *Artificial Intelligence* (17)185-348.

Connolly, C.I. 1984: "Cumulative generation of octree models from range data", *Proceedings of the IEEE Conference on Robotics*, pp. 25-32.

Ellis, R.E. 1984: "Extraction of tactile features by passive and active sensing", Proceedings of the SPIE Intelligent Robots and Computer Vision Conference, pp. 289-295.

Ellis, R.E. 1985: "Locating and object in a robot workspace from multiple stereo images", Proceedings of the SPIE Intelligent Robots and Computer Vision Conference, in press.

Faugeras, O.D. 1984: "New steps toward a 3-D vision system for robots", *Proceedings of the IEEE International Conference on Pattern Recognition*, pp. 796-805.

Grimson, W.E.L., and Lozano-Perez, T. 1984: "Model-based recognition and localization from tactile data", *Proceedings of the IEEE Conference on Robotics*, pp. 248-255.

Henderson, T.C., Fai, W.S., and Hansen, C. 1984: "MKS: a multisensor kernel system", *IEEE Transactions on Systems, Man, and Cybernetics*, 14(5)784-791.

Requicha, A.A.G. 1980: "Representations for rigid solids: theory, methods, and systems", *Computing Surveys* 12(4)437-460.

Weymouth, T. 1985: "Using object descriptions in a schema network for machine vision", Ph.D. thesis, COINS Department, University of Massachusetts (in preparation).

Winston, P.H., Binford, T.O., Katz, B., and Lowry, M. 1983: "Learning physical descriptions from functional definitions, examples, and precedents", *Proceedings AAAI*, pp. 433-439.