Robotics and Computer Graphics

C.E.Queiros

Dept. Computer Science
Faculdade de Ciencias e Tecnologia
Quinta da Torre
2825 MONTE DA CAPARICA
PORTUGAL

Abstract:

This paper focuses on a number of areas in which computer graphics has a direct and important relation to developments in robotics. It is not attempted to cover the whole field of computer graphics and robotics but instead to concentrate on a given number of areas in which the author has been involved.

1. Introduction.

Robotics is a multidisciplinary science. Its importance stems from the fact that robotics is considered to have a major impact in the factories of the future.

There are several key aspects in robotics of which two will interest us here: computer vision and programming. Both of them require advances which, in our opinion, partially fall within the area of computer graphics.

In the following two sections, a particular emphasis is given to the role of computer graphics in the two aspects just referred to. Also, some of the work done at our Department will be presented. This paper concludes with some final conclusions.

2. Computer vision.

In order to be able to interact with the environment, robots need sensors. Among these, vision sensors are by far the most important. By means of TV cameras and dedicated hardware, images of scenes may be acquired and stored in a computer memory where further processing may take place.

An image in a computer memory is a matrix of raw numbers. Further processing is needed so that objects on the scene are recognized and their location is computed. If the identification and location steps can be reliably performed, it is possible to plan and manipulate objects randomly located and oriented in a given environment. This is a key paradigm in the development of truly flexible robot manipulators.

The task of recognizing objects requires that a recognition strategy is established beforehand. That is, the salient features that most distinguish different types of objects under study, have to be found and an appropriate algorithm or decision process that makes use of these salient features, must be designed. This process is known as learning.

Basically, there are two general approaches to learning. The first one may be called learning by showing, and it essentially works as follows. Several images of a physical object in different positions are shown to the vision system, and the relevant parameters for recognition are extracted. This is the most common approach.

More recently, efforts have been directed towards what may be called automatic learning. Before analyzing automatic learning, we shall briefly comment on industrial scenes and CAD systems.

Industrial scenes are very much predictable. A scene consists of a given background (e.g. a conveyor, a known surface) on top of which one or more of a finite set of known objects lay. In order to recognize and manipulate objects, we only have to concentrate on a given number of expected objects which are known before hand and which normally are rigid objects. This is an important aspect which should be explored.

Also, more and more, CAD systems are being used to model industrial objects. This implies that all the geometrical information concerning an object can be expected to be available in its CAD representation. Other aspects which are important for vision can be easily integrated into all presently used CAD systems (e.g. color, reflectance properties of the materials that made up the object, etc.).

A real breakthrough in production engineering will occur when technology may enable low batch series of products to be produced as if they were mass production products. In other words, the economics of mass production be also present at the level of small series production. The main point here is that the factory of the future should be such that changes from product to product, i.e. from production cycle to production cycle, should be easily and economically implemented. This requires flexible manufacturing systems (FMS) possibly using (among others) flexible robots equipped with suitable sensors able to adapt from task to task in such way as to be sufficiently efficient and not imposing additional constraints on the overall manufacturing process. It stems from this, the importance of automatic learning processes for training recognition procedures required by vision processes.

The first point to be well understood is that the first stages leading up to the final manufacturing, include the design of the object (and we should expect it to be entered into a CAD system), and the planning and lay-out of the manufacturing process. If FMS is seen as the way the factories of the future will be, then robots should be expected to play a role. If flexible robots are to be used, this means that sensors have a role to play and recognition processes should use make full use of the CAD representations available. The whole process of retooling the assembly line should be implemented off-line and expensive and time consuming tries can not be realisfically considered (the fundamental point is that products are to be manufactured in small series, and the product that is manufactured in a given day might be completely different from the product to be manufactured thereafter).

Therefore, instead of having a physical object to show to the various sensors (among them vision sensors), it is much more interesting to use the information available in a CAD system.

This brings us to the implementation done at our department. Our CAD system (or more precisely our system modeler) is PADL2 (see [1]). The so called CSG (Constructive Solid Geometry) is used in order to model more or less complex solids. This form of representation has interesting properties. Among others it may uniquely represent a given object. Also, it is a very user friendly way of representing more or less complex solids.

PADL2 can produce shaded images by using a ray tracing approach with some limitations. Among others, it only assumes a given light source (at infinity) and it only considers the first intersection of a ray with an object. This implies that the shadows from solids onto other solids or shadows from parts of a solid onto other parts of the same solid are not considered. Also it is a time consuming technique (in view of the CSG form of representation) since boundary elements have to be obtained from volumetric representations in order to obtain synthetic images.

Nevertheless, it has been interconnected into a vision system (MAGISCAN), and the so called training by showing can be done in an interactive way. That is, different images of an object may be formed and processed by the vision system as if they were 'real' images and the salient discriminatory features be identified.

Also, and since an object on a planar surface (be it a conveyor or something else) will lie in one of its stable positions, an algorithm has been developed which given a polyhedrical object (convex or not convex) can compute its stable positions. This algorithm works by computing the convex hull of a 3D polyhedral, its center of gravity, and from there arrives at the stable positions (see [2]).

This is a brute force approach on which computer graphics has a role. Nevertheless, brute force approaches are always unreasonable solutions in the sense that they imply that more efficient and elegant solutions are missing.

Other commonly used CAD representations are more efficient (e.g. boundary representations) but nevertheless the gap between CAD and vision systems is still wide. This arises mainly from the fact that CAD representations are not intended nor devised as being also suitable to vision representations.

Research is being going on new approaches that can be efficiently used by both CAD systems as well as by vision representations schemes (see e.g. [3], [4], [5]). So far, no efficient solution has been found.

I finish this section by making an appeal to the computer graphics community. Since the simulation of real scenes is one of the topics of Computer Graphics, since these scenes are composed of objects and since a representation has to be available form these objects, can the gap between CAD and vision representation schemes be solved? This is the key to major developments both in the area of computer graphics as well as in the area of robotic vision.

3. Off-line programming.

A key paradigm with robot programming is the so called off-line programming. This is a way of programming a manipulator by means of simulation. Its importance stems from the fact the programming of manipulators imply interactions with the physical environment where the robot is positioned. These interactions may be in the form of acquisition of sensorial information, movements of a gripper from position X to position Y, grasping of objects, assembly of different elements into a single object, etc.. The differences between conventional programming and manipulator programming are self evident.

Physical action is involved with the programming of a manipulator. Therefore, and since error free programs are not likely to be achieved at the first step, it is clear that the simulation of sets of instructions can avoid some unpleasant effects which may occur as consequences of erroneous software. They might be detected and solved at an early stage and physical consequences may be overcome. This is one of the reasons why off-line programming is such an important paradigm in robotics.

Fine tuning is a task that is also frequently required. If quality and reliability are advantages to be expected from the factories of the future, and if manipulators are to have a role there, than precision movements and positioning have to be considered both in the design of manipulators and controllers as well as in the design of programming facilities.

From here, we may directly go to aspects related both to robotics and computer graphics: simulation and fine precision.

If the movement of a manipulator given a set of instructions has to be analyzed before real action, then simulation of its movements on a computer screen is a viable and user friendly way of helping the programmer. A manipulator can be either geometrically, kinematically or dynamically simulated. At our department software has been developed which allows the simulation (geometric) of various manipulators (see [6] for a description of a 3D graphic package used by the simulation package). The software is done in such a way as to allow for an user friendly way of introducing 'new' manipulators into the system.

Fine motions are difficult motions. Also, future programming languages for manipulators should not ignore that major breakthroughs can only be achieved if languages are sufficiently simple so that medium skilled personnel can operate with them. Interactive programming is one of the approaches that may achieve the goals which gather a large consensus. Since manipulators move in 3D coordinates systems, it is clear that editing facilities such as a 3D mouse, are a major help in the construction of manipulator programs. A 3D mouse has been implemented at our department.

4. Conclusions.

Two areas where there are subjects common both to robotics and computer graphics, have just been presented. The subjects analyzed do not cover all areas where robotics and computer graphics have a common interest. It is our experience which has just been presented.

It is clear from the contents of this paper that the point where solutions are most needed is in the question of finding representation schemes suitable both to CAD systems as well as to vision systems. At present, proposed solutions on the side of computer vision are too much restricted.

References.

[1] - Solid modelling: current status and research directions, Requicha A.A.G., Voelcker H.B., IEEE Computer Graphics and Applications, Oct. 1983, pp. 25-37, 1983.

[2] - Determinação das posições estaveis de um solido, Queiros C.E., Santos J.A., Steiger Garção A., Internal Report GR RT-PS-14-87, 1987.

[3] - CAD-based robot vision, IEEE Computer, August, 1987.

[4] - CSG-EESI: A new solid representation scheme and a conversion expert system, Xie S., Calvert T.W., IEEE Transactions on PAMI, vol. 10, pp. 221-234, 1988.

[5] - Generating an interpretation tree from a CAD model for 3D-object recognition in bin picking tasks, Ikeuchi K., International Journal of Computer Vision, pp. 145-165, 1987.

[6] - Ambiente multiwindow 3D, Santos J.A., Internal Report GR RT-PS-30-87, 1987.