Appearing in the International Conference on Recent Advances in 3-D Digital Imaging and Modeling, Ottawa, Ontario, May 12-15, 1997.

A System for Semi-automatic Modeling of Complex Environments

Andrew E. Johnson, Regis Hoffman, Jim Osborn and Martial Hebert
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We present a perception system, called Artisan, that semi-automatically builds 3-D models of a robot's workspace. Range images are acquired with a scanning laser rangefinder and then processed, based on a systematic sensor characterization, to remove noise and artifacts. Complex 3-D objects represented as surface meshes are subsequently recognized in the range images and inserted into a virtual workspace. This graphical virtual workspace is then used to by human operators to plan and execute remote robotic operations.

1. Introduction

Artisan is a system that combines 3-D sensors, object modeling and analysis software, and an operator interface to create a 3-D model of a robot's work area. This paradigm, known as task space scene analysis, provides a much richer understanding of complex, interior work environments than that gleaned from conventional 2-D camera images, allowing an operator to view the work space from inaccessible angles and also to plan and simulate robotic actions within the virtual world model [1] [6]. Through object recognition, Artisan assigns semantic meaning to objects in the scene, facilitating execution of robotic commands, drastically simplifying operator interaction and setting the stage for automatic task execution.

World modeling with Artisan proceeds as follows. From a remote workstation, the operator directs a scanning laser rangefinder to acquire images. Positioned at the work site by a mobile worksystem, the rangefinder sends range and intensity images to the operator for display. The operator then defines a desired region of interest and selects from a CAD model database objects that appear in the region of interest. From these clues, Artisan recognizes and locates the objects selected by the operator. Finally, the operator accepts or rejects models recognized by Artisan. Once accepted, the objects appear in a virtual world model at the calculated location. This process of range data collection, 3-D task space model in place, the operator can now access automatic controls such as trajectory planning, collision avoidance, and scripted motion sequences to execute tasks automatically. Figure 3 shows the Artisan system user interface which is composed of a main menu for controlling perception tasks, a range and intensity image display tool, a 3-D viewing tool and virtual robot workspace used to plan and execute robotic actions. Figure 2 shows all of the components of the Artisan system with data flow between them and necessary user interaction.

This research was performed as part of the Artisan Project in the Field Robotics Center and was funded by the Department of Energy under contract DE-AC21-92MC2910.

2. Scene sensing and processing

During world modeling, an operator commands Artisan to take an image of the scene with a Perceptron Model 5000 laser rangefinder. The Perceptron 5000 returns a 256x256x12bit range image. Its field of view is 30˚, its ambiguity interval is 10 meters, and it has a nominal standoff distance of 1.5 meters. To eliminate some of the noise in the range data, multiple range images (~10) are taken and the median range value for each pixel is returned. Range data acquisition and temporal filtering takes roughly 10 seconds. After the scene data is acquired, the range and intensity images are displayed to the operator.

Next, in the display of the range image (Figure 1), the operator draws a rectangular region of interest around the

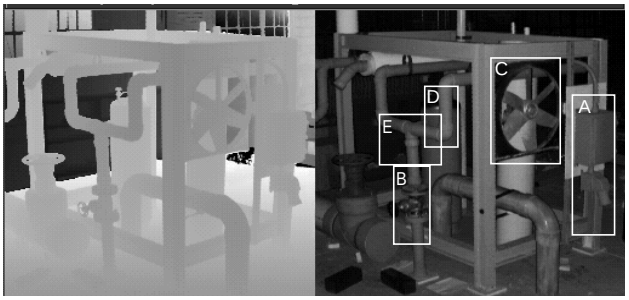
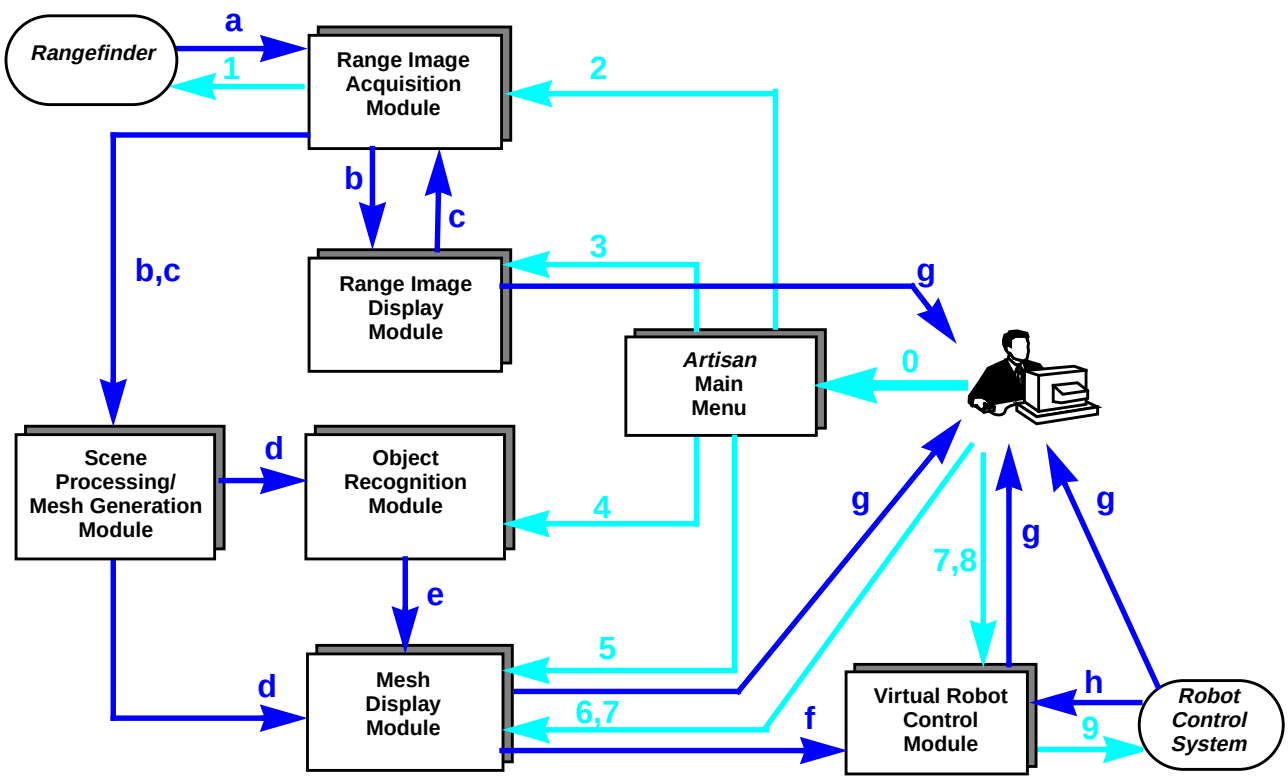


Figure 1. Range and intensity image with marked regions of interest. The first step in modeling the environment is to acquire a range image of the scene. Next, regions of interest outlining the location of interesting objects are drawn by the user.

Artisan system block diagram



control flow

data flow

0	button presses & menu selections
1	scanner control signals
2	range image acquisition
3	region of interest selection
4	recognition initiation
5	object match acceptance/rejection
6	object pose/dimension adjustments
7	viewpoint changes
8	simulated robot commands
9	real robot commands

a	raw range images
b	filtered range data
c	range of interest
d	scene surface mesh
e	matched object models
f	accepted, adjusted object models
g	visual feedback to operator
h	robot joint angles & status

Figure 2. The Artisan scene modeling system consists of several modules for 3-D data acquisition, scene data processing, object recognition, data display and robot control. The diagram details the data flow and the necessary human interaction to make Artisan work.

objects to be modeled. This reduces the data to be processed to that which is important to the task at hand. Next a scene surface mesh is created from the range image in the region of interest by making each pixel a vertex and connecting vertices across rows and columns. If the operator chooses, the amount of scene data to be processed can be reduced by sub-sampling the range image during mesh creation. Sub-sampling is appropriate when recognizing large objects without large amounts of surface detail.

Previous work with the Perceptron laser rangefinder [7] has noted problems with laser scanning technology with regard to intrinsic scanner characteristics, surface material properties and surface incidence angle. As part of the development of the Artisan system, an effort was made to characterize and understand the 3-D laser rangefinder, and to use these results to improve the scene sensing capability of the system. For example, the graph in Figure 4 shows the effect of surface incidence angle on the range measurement. A polished aluminum surface at fixed distances from the sensor is scanned by the laser as the surface incidence angle increases. As the incidence angle goes beyond 20 degrees, the range measurement is no longer constant. A practical implication of this result is that shiny, cylindrical objects (such as metal pipes frequently found in industrial facilities) are difficult to model correctly.

We incorporate the results of this characterization into scene data processing by removing vertices whose incidence angles (measured as the angle between mesh normal and viewing direction) are greater than a fixed threshold. Eliminating vertices with large incidence angles has the added benefit of eliminating mixed pixels, pixels that lie on range discontinuities. Mixed pixels do not correspond to the points on the surfaces of objects and should be eliminated.

When a pixel is eliminated, all of the edges in the mesh that are connected to it are also eliminated. Removing edges can result in the creation of small isolated patches of pixels. Since these patches generally do not contribute to recognition, they are eliminated from the mesh to reduce the amount of scene processing for recognition.

The final step in scene data processing is the application of a low-pass filter to the mesh [10]. This filter smooths without shrinking, so it removes spurious noise from the mesh while still preserving the shape of objects. Since the

Figure 3. User interface for the Artisan system. Artisan consists of a control panel (upper right), a range image display module (upper left), a 3-D mesh display module (lower right) and an interface to a virtual robot world.

filter operates on a 3-D surface mesh, and not on the range image, the smoothing provided will not contain artifacts caused by the parameterization of the range image.

Figure 1 shows a shaded view of a scene surface mesh from a region of interest encompassing the entire range image shown in Figure 1 before and after processing. Scene processing for a typical region of interest of 5000 points takes approximately 20 seconds on an 100MHZ Silicon Graphics Indigo2.

3. Model representation

Prior to object recognition, models of objects that are to be recognized must be created. Since we are interested in modeling complex scenes, the representation we choose must be flexible. A polygonal surface mesh is an established way to describe the shape of complex objects in computer graphics, and it is amenable to our recognition algorithm which requires a surface represented as oriented points (3-D points with associated surface normal). The vertices of a surface mesh correspond to points on the surface of the object. The normals at the vertices can be calculated by fitting a plane to the vertex and all of the neighboring vertices in the mesh. Using surface meshes places very few restrictions on the shape of objects that can be represented, making our recognition system extremely flexible.

An important requirement of any recognition system is the ability to generate the model representations used in recognition with relatively little effort. Given a CAD drawing of an object to be recognized, surface mesh generation is simple. The CAD drawing is imported into a CAD package with finite element capabilities (e.g., ProEngineer). The finite element software is then used to automatically tessellate the surface of the object into triangular faces given some user defined constraints on minimum and maximum edge lengths. Figure 6 shows an example of a CAD model of a fan transformed into a surface mesh and Figure 7 shows some of the surface mesh models created from CAD drawings used in Artisan's model library. Since models for recognition can be easily generated from CAD drawings, Artisan can be quickly modified to recognize objects in any man-made environment, provided the CAD drawings of those objects are available.

Modeling objects for which a CAD drawing cannot be generated is not as trivial, but is still feasible if the object is available for imaging. Modeling from range images proceeds as follows: With a rangefinder, a complete model of the object is imaged from many different viewpoints. The multiple surface mesh views are then registered to find the correct location of the data sets in a single world coordinate frame [4]. Fortunately, the matching techniques used in our recognition system are also useful when registering multiple views of an object. Next, using a volumetric

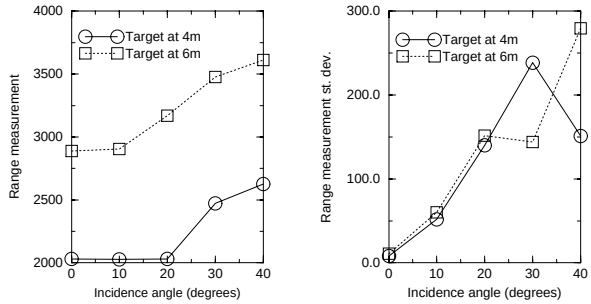


Figure 4. Range measurement mean (left) and standard deviation (right) for a sequence of 100 measurements of a polished aluminum target (at 4m and 6m distances) as the incidence angle is increased.

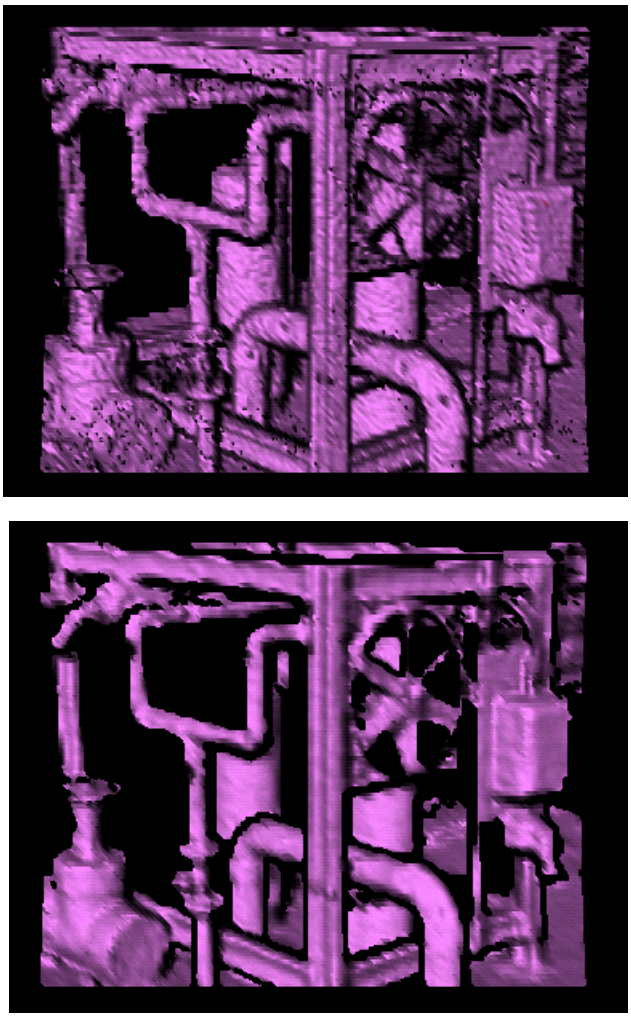


Figure 5. Shaded views of a scene surface mesh before (top) and after (bottom) processing. Processing removes pixels on range discontinuities, removes small patches and low-pass filters the data.

surface mesh integration algorithm [5], the surface meshes are integrated into a single seamless surface mesh model.

The ability to generate models of objects from range images has some useful consequences when modeling scenes with unexpected or unknown objects. If an unknown object that is important, but is not in the model library is encountered when modeling, the operator can make a model of the object simply by taking an image of the object and selecting a region of interest that contains only object data. The model representation (albeit one view) can then be created from the scene data. The model is then inserted into the model library for recognition at a later time. If a complete model of the object is desired, then multiple views of the object can be taken and registered and integrated as explained above. This ability to generate models at run-time is important when modeling interiors as built.

The representation we use for recognition assumes that the vertices in the surface mesh adequately describe the shape of the object. Implicit in this is the assumption that the vertices of the surface mesh are evenly distributed over the surface of the object. Occasionally, the model surface mesh generation process (FEM or Multi-view) will distribute vertices that are either too coarse or too fine for recognition. To solve this problem we use a mesh simplification algorithm that normalizes the lengths of edges in a mesh while preserving the shape of the object represented [3]. The end result is a surface mesh representation of appropriate resolution that has an even distribution of points over the surface and from which our recognition representation can be made.

4. Object recognition

Artisan recognizes objects by establishing correspondences between oriented points (surface mesh vertices with normals) on the surface of a model and oriented points in the scene. Every oriented point on an object has an associated *spin-image* that describes the shape of the object with respect to the oriented point; the procedure for spin-image generation for surface registration

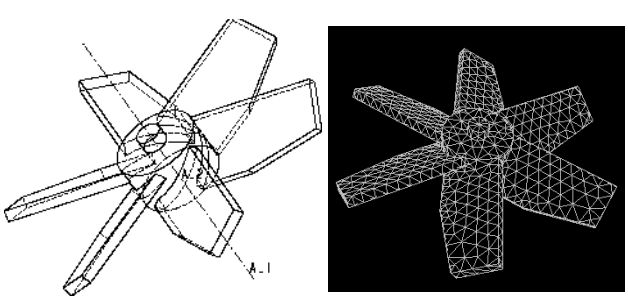


Figure 6. The generality of our recognition system is demonstrated by its ability to use surface meshes generated from CAD models using finite-element surface tessellation software.

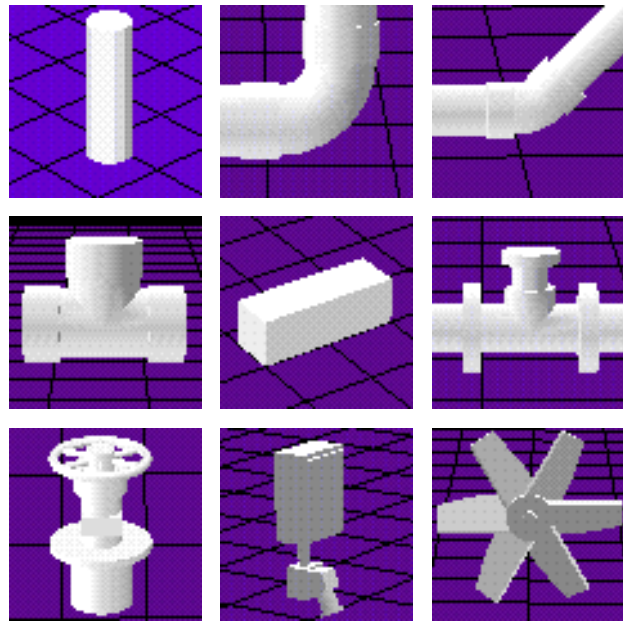


Figure 7. Models generated from CAD drawings used in Artisan.

is detailed in [4]. Not surprisingly, spin-images used for surface registration can also be used for object recognition.

A spin image for an oriented point on a surface mesh is created as follows. For all points on the surface mesh, the 2-D oriented point coordinates (α,β) as shown in Figure 8 (distance to the line passing through the oriented point and oriented along the normal and the distance to the tangent plane of the oriented point) are calculated. Next, the pixel in the spin image that is indexed by the oriented point coordinates is incremented by one. Some examples of spin images for a CAD surface mesh of a valve are given in Figure 9. In the spin-images, darker pixels correspond to pixels in the image that have more points projected into them. Spin-images should not be confused with camera images of the models; spin-image generation depends on a cylindrical projection of points on the surface of the model, not a perspective projection. The procedure for matching points based on spin-images lies somewhere between geometric hashing [8] and structural indexing [9], two forms of model based object recognition.

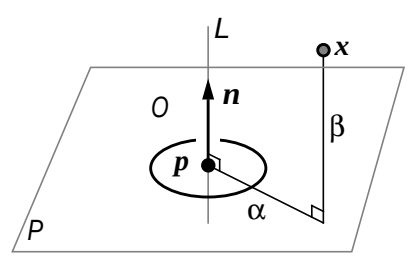


Figure 8. Oriented point basis.

Spin images of corresponding oriented points will be similar because the construction of spin images is based only on the shape of the object, not on the pose of the object. Therefore, a comparison of spin-images suffices to establish oriented point correspondence.

The models in the model library are processed for recognition, prior to world modeling. For each model, the spin-images for each oriented point on the model are created. The spin-images for each model are then stored in a stack of images for later use in recognition. Each model in the model library has its own spin-image stack. Processing the models off-line saves time during recognition.

After selection of the region of interest, the operator selects a model(s) from the model library to be recognized and localized in the scene. Next, an oriented point is randomly selected from the scene. The spin-image of this oriented point is created using the scene data. This scene spin-image is then compared to all of the spin-images in the stack(s) of the model(s) selected. If the scene spin-image and a model spin-image have enough similarity, a point correspondence is established between model and scene points. This process is repeated for a random selection of scene points (~1/10 the total number of scene points), to establish many model/scene point correspondences.

Because clutter in the scene and symmetries in the models may cause incorrect correspondences, correspondences are filtered by grouping them into sets that are geometrically consistent. From these sets of correspondences, plausible rigid transformations are calculated; these transformations align the model(s) with the scene data.

The plausible transformations are verified and refined with a modified iterative closest point algorithm [2]. The verification algorithm iteratively spreads point correspondences from those already established by spin-image comparisons over the surface of the model and the scene. At each iteration, it recalculates the transformation from model to scene. If the model and scene are well aligned and have similar shapes, then the number of correspondences created will increase dramatically. Models that have a large fraction of their points in correspondence with scene data are reported as recognized in the scene.

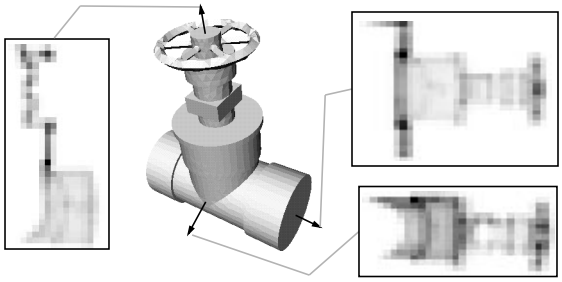


Figure 9. Some example spin images generated from a surface mesh generated from a CAD drawing of a valve. Each spin-image is connected to its associated oriented point.

5. World modeling

Recognition results are presented to the operator in a 3-D viewer window that provides rotation, translation and zoom capabilities as well as several modes of rendering. Figure 10 shows wire frame representations of the recognized models for the five regions of interest selected in Figure 1, along with the original scene data shown as shaded surfaces. The user makes the determination of whether the recognized objects are localized well enough to be inserted into the model of the robot's world. This confirmation step affords a final human check on the validity of the object recognition.

Once the user accepts Artisan's results, the objects are

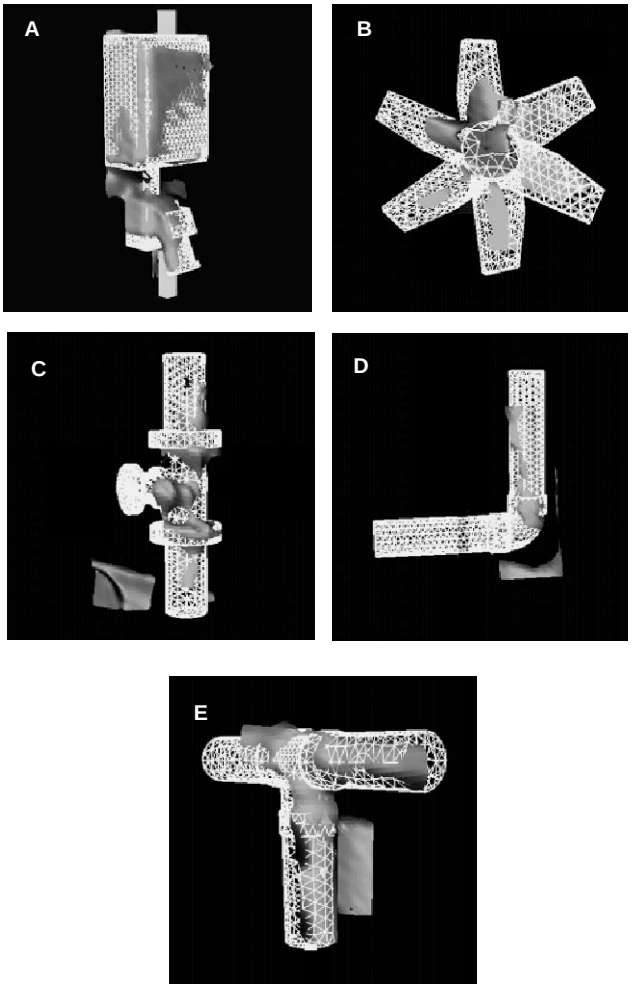


Figure 10. Recognized objects (wireframe) superimposed on scene data (shaded) for five models selected from the model library.

inserted into the global 3-D world model that is captured using TeleGRIP (by Deneb Robotics, Inc.) Figure 10 shows the objects of Figure 7 as they appear in the scene relative to our Cincinnati Milacron T3 industrial manipulator.

Once the objects are placed in the workspace, robotic tasks can be planned and executed virtually before being executed by the real robot. This provides safer, and more reliable execution of remote tasks than is possible in teleoperation using conventional 2-D cameras. Furthermore, because a semantic meaning is attached to objects in the virtual world high level commands to the robot can be used to perform tasks (e.g. “cut that pipe”, “turn that valve”).

The Artisan system has been demonstrated in multiple dismantlement and decommissioning contexts. In a decontamination task using the T3 manipulator, for example, Artisan correctly identified and located objects in the manipulator's work space, and a simple trajectory to follow surfaces of pipes and vessels in the workcell was computed using TeleGRIP’s motion planning features. The operation was executed first in simulation; then the proper commands were downloaded to the real manipulator’s controller, which then performed the wash-down operation. In this manner, locations of the real objects could be compared to the locations of the recognized objects. Observed deviations were on the order of 1-4 cm.

Artisan has also been integrated into the Rosie mobile worksystem to enable semi-automatic size reduction of the lower shield plug of the Argonne National Lab CP-5 experimental reactor. Using the observed motions of the real Rosie relative to the shield plug (actually a mock-up), overall task performance in this case had lower fidelity due, in part, to the scale of the object (approximately ten feet in diameter) relative to the laser rangefinder’s field of view. The lower fidelity is also attributable to the somewhat lower stiffness of Rosie’s heavy manipulator & controller with respect to the T3 manipulator.

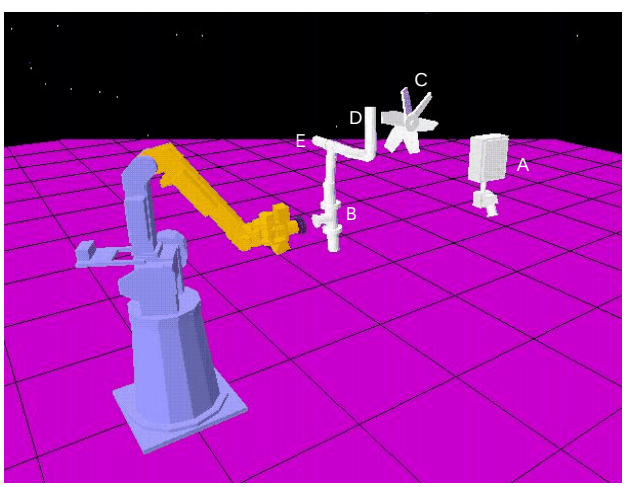


Figure 11. Virtual robot workspace showing T3 robot and five recognize objects. The virtual workspace is used to plan robotic tasks in simulation before they robotic control commands are down loaded to the real robot.

6. Discussion

Our approach to object recognition offers substantial improvements in performance over previous systems. First, the approach does not make any assumptions on the sensor used for acquiring the range data. In fact, we have demonstrated the system both with an imaging laser rangefinder and with a light stripe rangefinder. In contrast, systems based on interactive stereo [11] or line segmentation make strong assumptions on the geometry of the sensor and do not generalize nearly as well as does the point matching algorithm at the heart of Artisan.

Another drawback of competing techniques is that they tend to degrade the input range data by forcing it into one of the primitive classes. Our approach is general in that, by working directly with surface meshes, it does not make assumptions of the shape of the objects that can be recognized. Our earlier system [6] restricted recognition to objects mostly composed of planar and quadric surfaces. This constraint relegated Artisan to use in environments comprised only of objects with simple geometries, such as cylinders and boxes. Further, our earlier system’s ability to recognize was much more dependent on the amount of range data available for a given object in the scene. This dependency made the system more susceptible to the occlusions that are prevalent in industrial process complexes; the earlier Artisan could not determine that two segments of a horizontal pipe lying behind a vertical pipe were actually the same object. In addition, the 3-D segmentation step is computationally expensive. The newer Artisan can do everything its predecessor did in about half the time.

7. Conclusions and future work

Artisan’s ability to model objects, even when surfaces are partially occluded or the sensing viewpoint changes, makes it a valuable tool for modeling complex environments and planning remote robotic tasks. Artisan is effective because user input is used to make high level decisions for the system while low level recognition and registration are done automatically. Furthermore, Artisan was designed with generality in mind: the system can work with any sensor that acquires 3-D surface data; object models can be generated easily from CAD drawings; and models of new unknown objects can be built and incorporated into the system at run-time.

We are working on a number of extensions to Artisan.

First, we will need to deal with large library of objects in real applications. This is not only because of the complexity of the environments, but also because allowing large sets of models will alleviate the burden on the operator of selecting models from the library. We are working on an approach in which all the spin-images from all the models are combined into a single stack. At recognition time, the spin image from the scene are compared to all the spin images in the combined stack. Because this approach is computationally expensive, we are working on an algorithm for reducing the number of spin images representing the models using the principal components of the space spanned by the spin-images.

One weakness of the current recognition algorithm is that it takes more time to recognize symmetric objects as opposed to asymmetric objects because there are many different ways of pairing the spin images. The technique outlined above for multiple objects will also work for identifying symmetries in objects and for reducing the recognition time.

Another limitation is that the object models are represented at a single scale. For example, we can recognize a pipe and its position in the image but we cannot compute its radius if it deviates too much from the model's radius. One approach is to parameterize the spin images as functions of the scale of the objects. We are investigating this possibility.

References

[1] Barry, R.E., Little, C., and Burks, B. Requirements and Design Concept for a Facility Mapping System. *Proc. ANS 6th Topical Meeting on Robotics and Remote Systems* (ANS '95), pp. 775-783, February 1995.

[2] P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):239-256, 1992.

[3] A. Johnson and M. Hebert, "Control of mesh resolution for 3-D object recognition," CMU Robotics Institute TR, CMU-RI-TR-96-20, October 1996.

[4] A. Johnson and M. Hebert, "Surface registration by matching oriented points." *Proc. Int'l Conf. on 3-D Digital Imaging and Modeling*, May 1997.

[5] A. Johnson and S. Kang, "Registration and integration of textured 3-D data." *Proc. Int'l Conf. on 3-D Digital Imaging and Modeling*, May 1997.

[6] A. Johnson, P. Leger, R. Hoffman, M. Hebert, J. Osborn, "3-D object modeling and recognition for telerobotic manipulation," *Intelligent Robots and Systems 1995 (IROS '95)*, pp. 103-110, 1995.

[7] Kweon, I.S., Hoffman, R., and Krotkov, E. *Experimental Characterization of the Perceptron Laser Rangefinder*. Carnegie Mellon University Robotics Institute Technical Report CMU-RI-TR-91-1. January 1991

[8] Y. Lamdan and H. Wolfson, "Geometric Hashing: a general and efficient model-based recognition scheme," *Proc. Second Int'l Conf. Computer Vision (ICCV '88)*, pp. 238-249, 1988.

[9] YF. Stein and G. Medioni, "Structural Indexing: efficient 3-D object recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, (14)2:125-145, 1992.

[10] G. Taubin, "A Signal processing approach to fair surface design," *Proc. Computer Graphics (SIGGRAPH '95)*, pp. 351-358, 1995.

[11] S. Thayer, S., C. Gourley, M. Trivedi, C. Chen, S. Marapane, P. Butler and H Costello, On-line stereo vision and graphical interface for decontamination and decommissioning applications using the advanced servo manipulator. *Proc. ANS 5th Topical Meeting on Robotics and Remote Systems (ANS '95)*, pp. 287-294, April 1993.

[12] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int'l J. Computer Vision*, 13(2):119-152, 1994.

Acknowledgments

We would like to thank our contract officer representative, Vijay Kothari, for his unselfish support of our efforts. We also appreciate the technical guidance and programmatic insights of our peers and partners: Bill Hamel, Dennis Haley, Mark Noakes, Bob Barry, Roger Bradley and the rest of the Oak Ridge National Lab Robotics and Remote Systems Division; Lin Yarbrough of the Robotics Technology Development Program; Mike McDonald and Charles Little of Sandia National Labs; and RedZone Robotics. The work would not have been possible without the technical support of Sachin Chheda, Jim Frazier, Cecelia Shepherd and Thomas Tadlock and Marie Elm.