Advanced Sensing and Control Techniques to Facilitate Semi-Autonomous Decommissioning of Hazardous Sites

Lead Principal Investigator:
Robert J. Schalkoff
rjschal@clemson.edu
(864)-656-5913

Co-Investigators:
Robert M. Geist
Darren M. Dawson
Electrical and Computer Engr.
Computer Science
Clemson University, Clemson, SC 29634-0915

DOE Project Monitors:
Dr. Robert Price and Dr. Ramoncita Massey

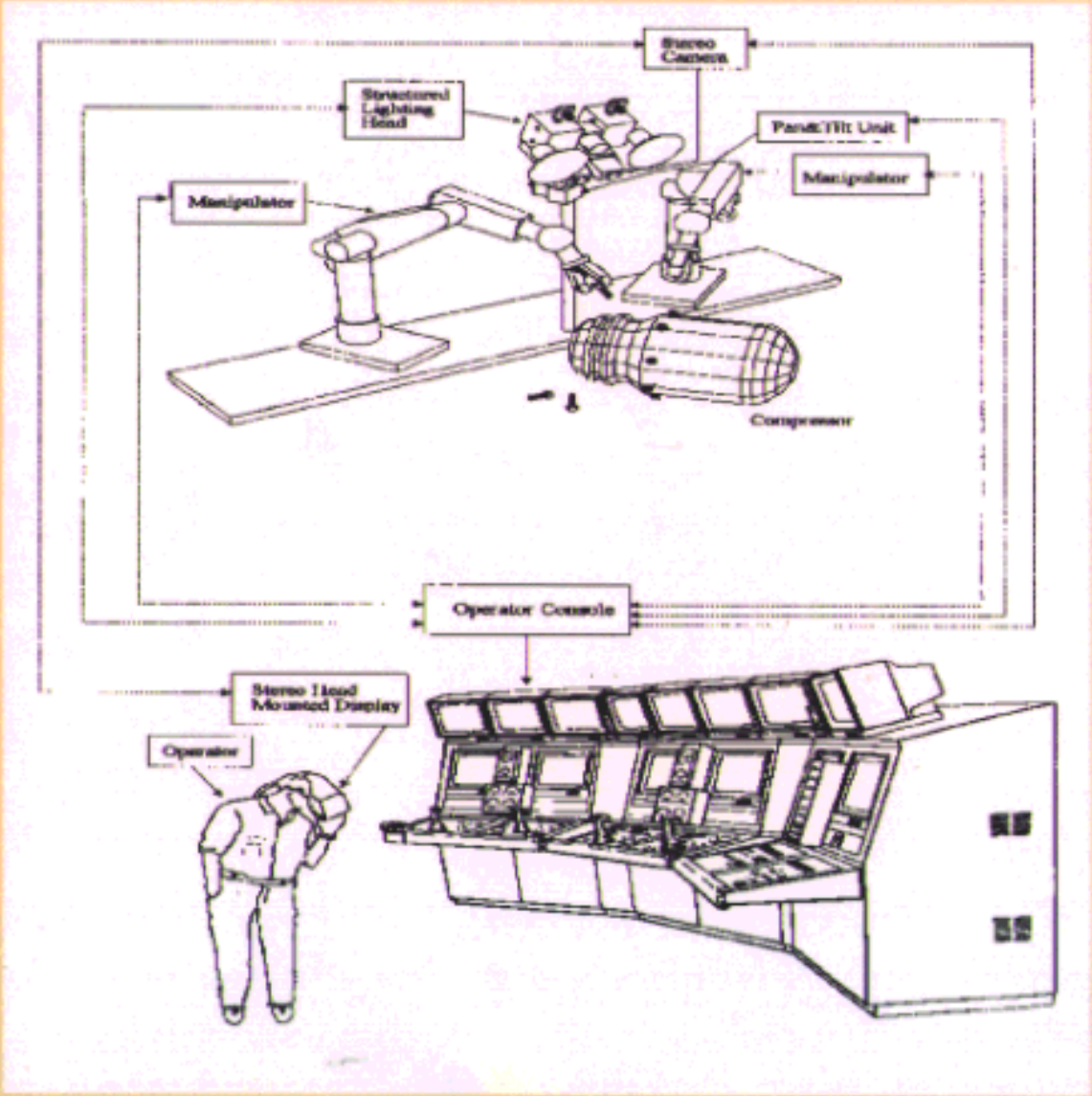
Overview

This research is intended to advance the technology of semi-autonomous teleoperated robotics as applied to Decontamination and Decommissioning (D&D) tasks.

Specifically, research leading to a prototype dual-manipulator mobile work cell is underway. This cell is supported and enhanced by computer vision, virtual reality and advanced robotics technology.

3 Components of Effort

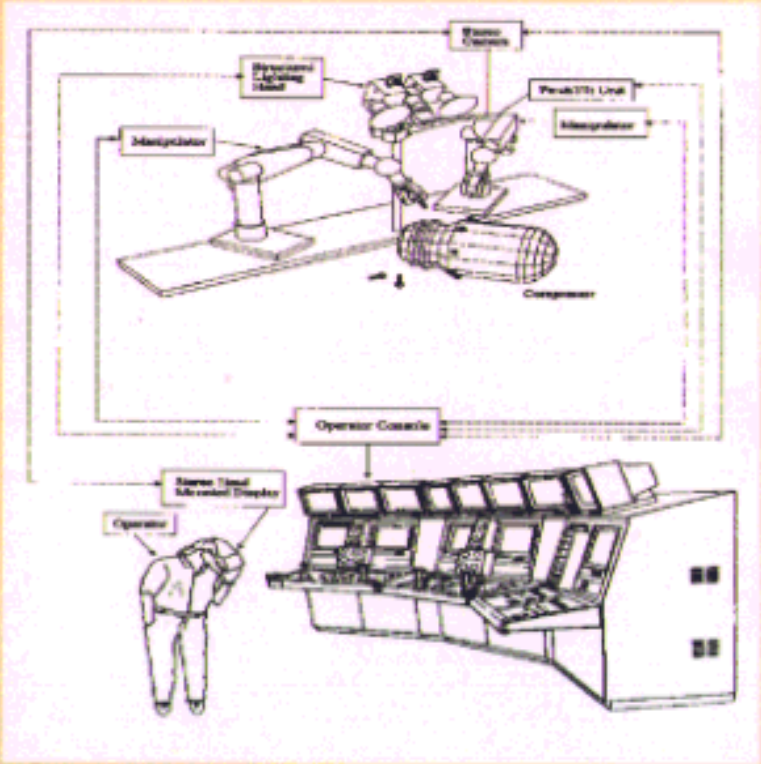
- Characterize Environment
  (Geometric and Radiometric) → Dr. Schalkoff
- Develop VR Rendition of Environment → Dr. Geist
- Control of D&D from Virtual World → Dr. Dawson



- D&D Using VR-Enabled Telepresence (Requires Site *Virtualization*)

Motivation for Virtualization

- *Virtualization* is a technique for the autonomous, non-contact creation of a virtual environment from an existing, real environment.
- Provides a new approach to remote sensing and for providing telepresence in previously unmodeled environments, e.g., waste disposal and storage areas.



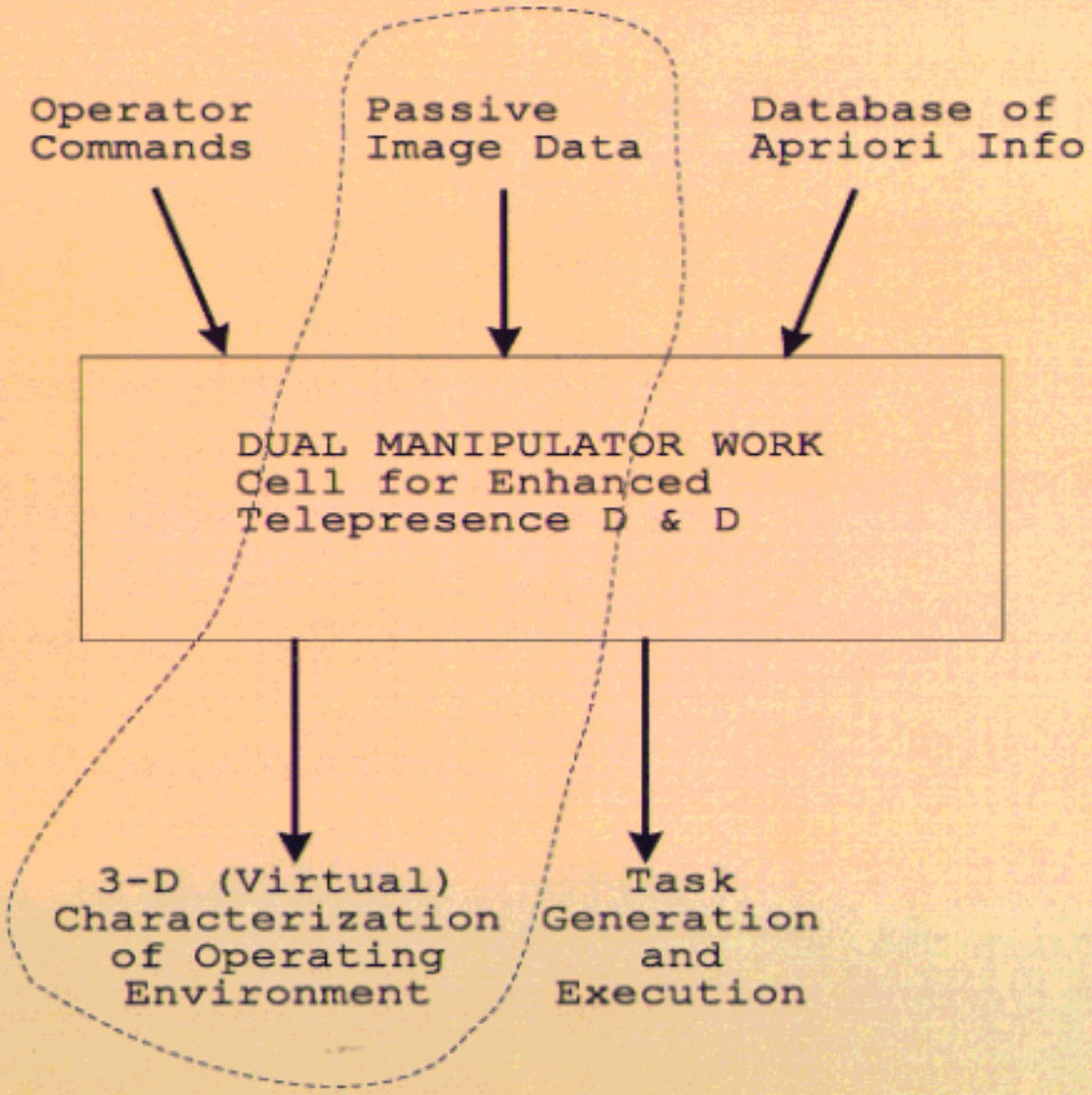


Figure 1: Characterization Emphasis

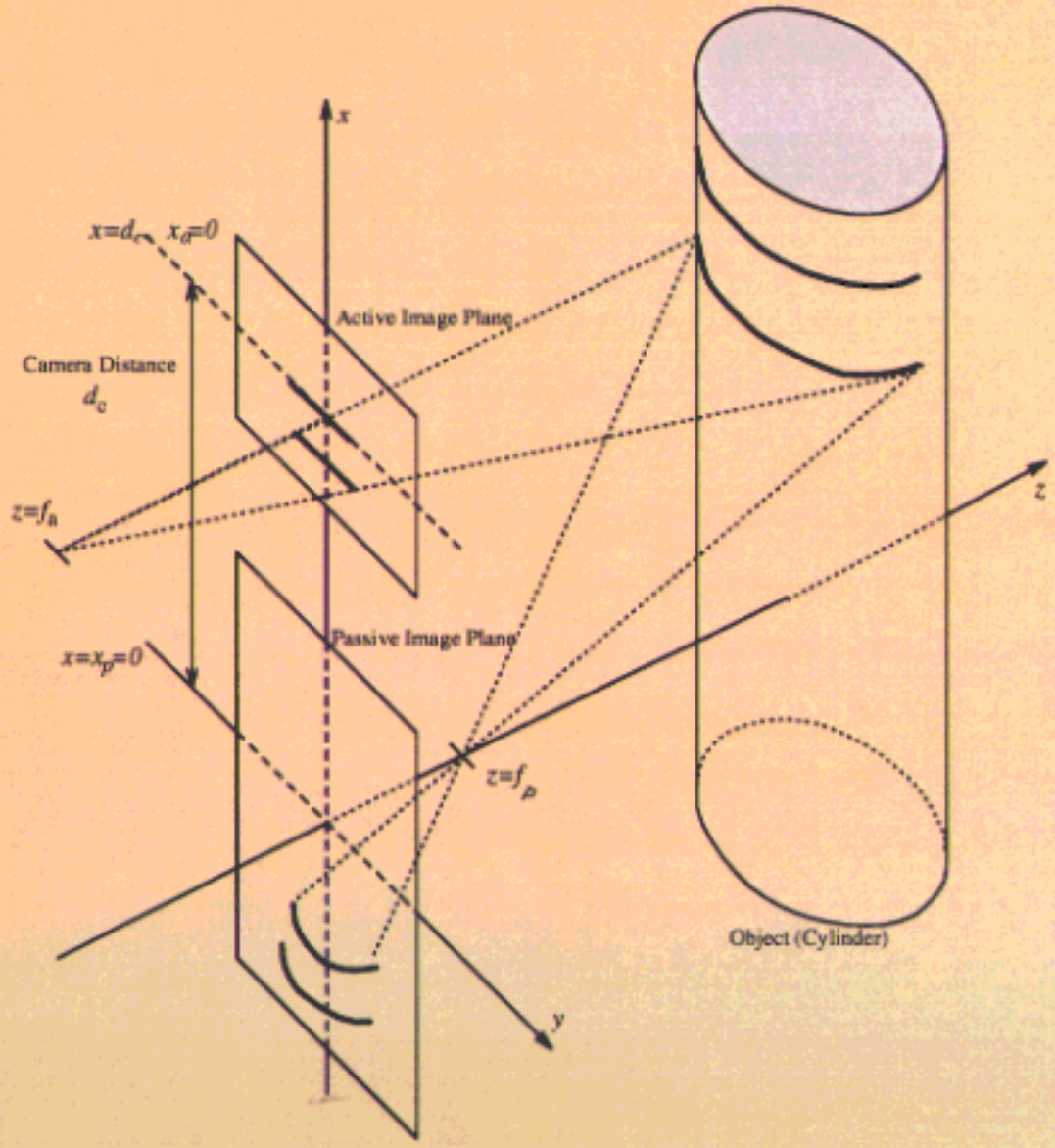


Figure 2: Determining Geometry (llustration Only)

Virtualization Data Head

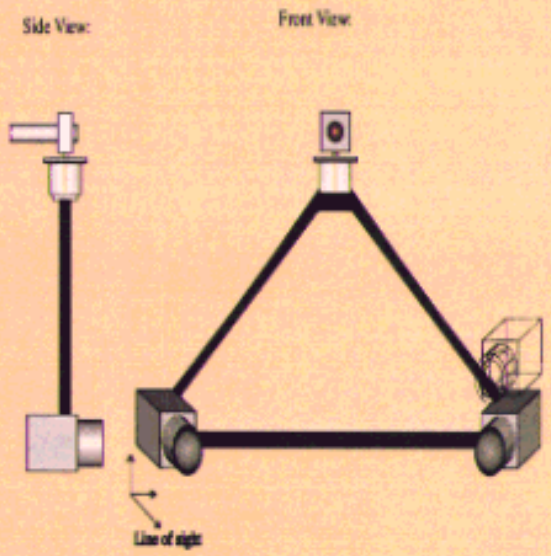


Figure 3: Vision 'Head' for Geometric Characterization Algorithm

- Pulnix TM-7CN monochrome cameras for geometric characterization
- Pulnix TMC-7 color cameras for color and texture rendering
- laser line projector (670 or 630 nM)
- Directed Perception PTU-46-70 high resolution pan-tilt unit

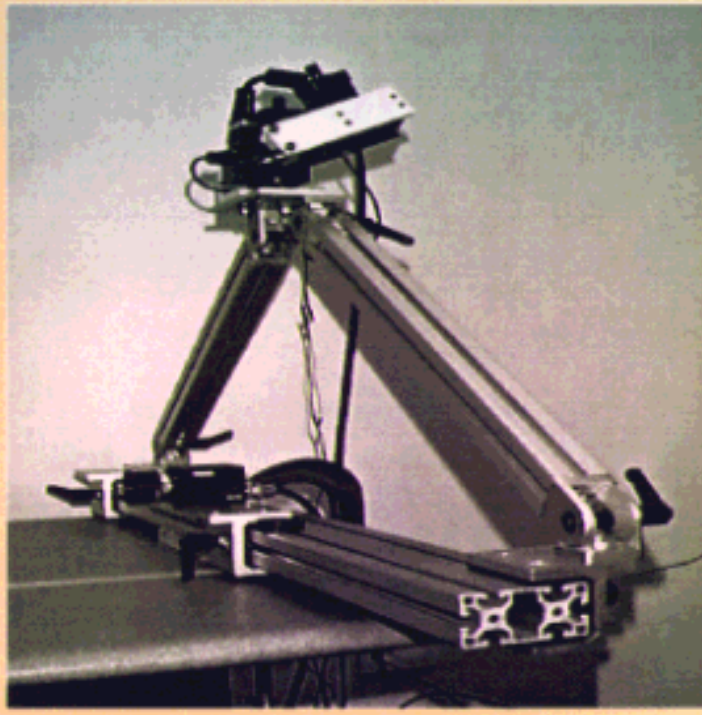


Figure 4: Prototype Vision 'Head

Fitted Passive Image Data

Figure 5: Fitted Passive Stripes on Ball

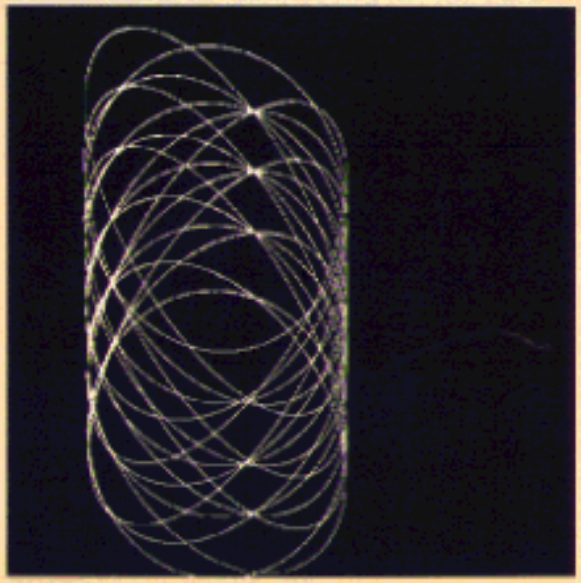


Figure 6: Passive Stripes on Cylinder

Correspondence Problem

- Determine correspondence between the passive image curve set and quadric surface patches.
- This is challenging in complex scenes.

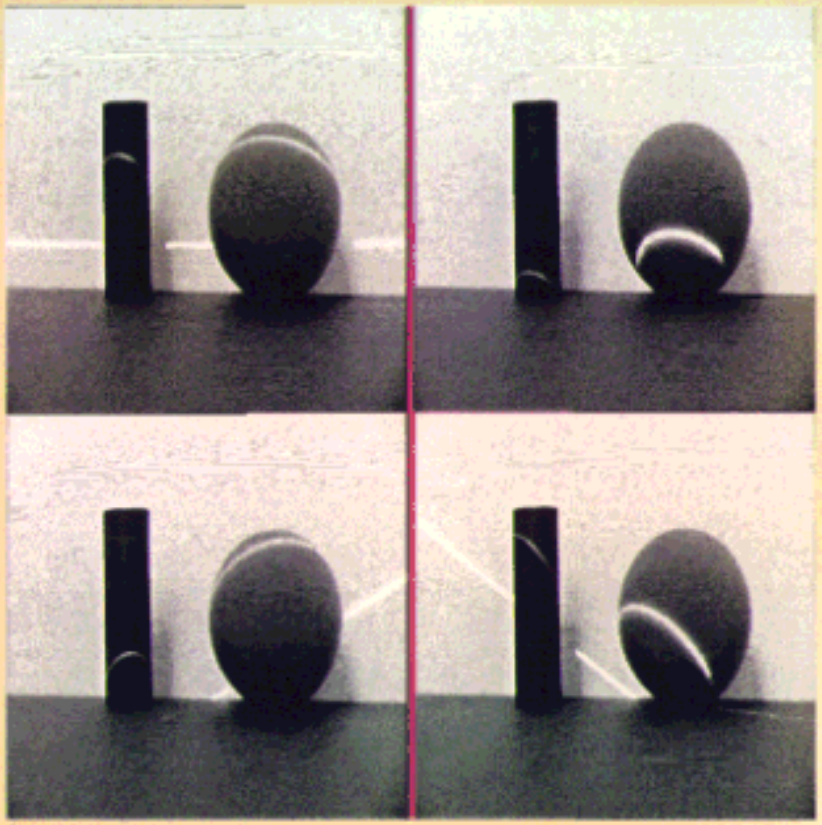


Figure 7: Segmentation Example

Additional Resources

Extensive coverage of our efforts are available on the Web. Relevant pages are:

`http://ece.clemson.edu/iaal/doeweb/doeweb.htm` (project home page). This includes a MPEG Clip of a 'Virtualization' Example (Sphere) Using the LCM Algorithm (`vzation.mpg`).

`http://chip.eng.clemson.edu/vr/cpsc/cpsc.html` (virtual reality),

`http://ece.clemson.edu/crb/research/robotics/doevr/index.html` (robotics)

Rendering Virtual Environments

- sophisticated diffuse lighting effects off-line
- real-time display precludes sophisticated specular treatment
- suitable rendering equation (in radiance):

$$c = B/\pi + k_s(H \bullet N)^m$$

where:

- B = patch radiosity (exiting irradiance)
- N = unit-length normal
- H = unit-length $(V + L)$

Extracting Color/Texture

Assume rendering equation:

$$c = B/\pi + k_s(H \bullet N)^m$$

describes real scene. Use:

- deduced quadric geometry
- stereo (left-eye, right-eye) camera images

to estimate:

- surface specular parameters k_s, m,
- patch reflectivity, ρ_i, that determines patch radiosity, B_i

Extracting Color/Texture

Step 1. Sample raw texture using "rainbow" mask.

- a "rainbow" texture has a unique color in each pixel
- render virtual scene; use:
  - known geometry
  - known camera position
  - "rainbow" textures mapped to all surfaces
- overlay virtual image on camera image; extract texture map from camera image into corresponding pixels

Extracting Color/Texture

Step 2: specular correction

- sample pixels with underlying vertices
- each vertex (stereo) yields sample pair whose difference eliminates radiosity:

$$c_l - c_r = k_s((H_l \bullet N)^m - (H_r \bullet N)^m)$$

- fit k_s and m to samples
- remove specular effects (per-pixel interpolation)

Extracting Color/Texture

Step 3: diffuse correction (MUCH more difficult)

- direct illumination varies with $L \bullet N$
- significant indirect illumination in real scenes
- we seek an inversion of the standard radiosity equation:

$$\rho_i = \frac{B_i - E_i}{\sum_{j=1}^{n} B_j \frac{A_j}{A_i} F_{ji}}$$

- assume missing indirect illumination is linearly scaled value of known illumination
- interpolate pixel correction factors from vertices

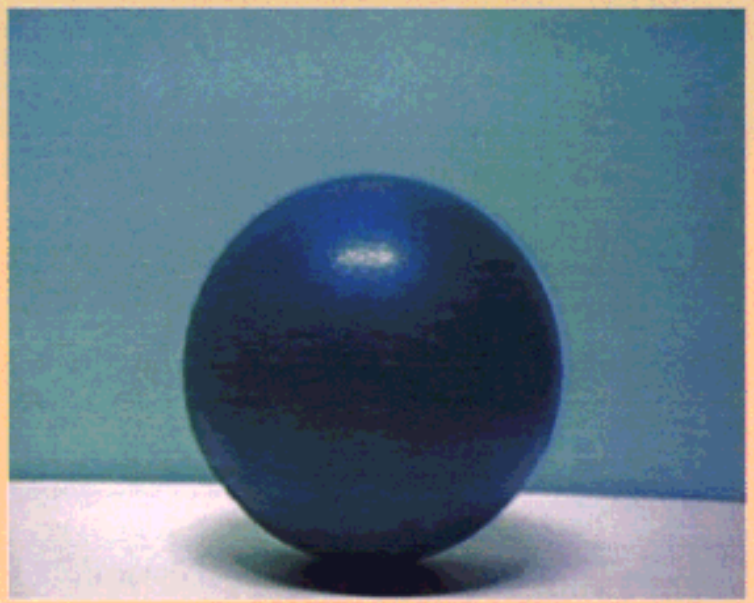
Extracting Color/Texture

Step 4: extrapolation

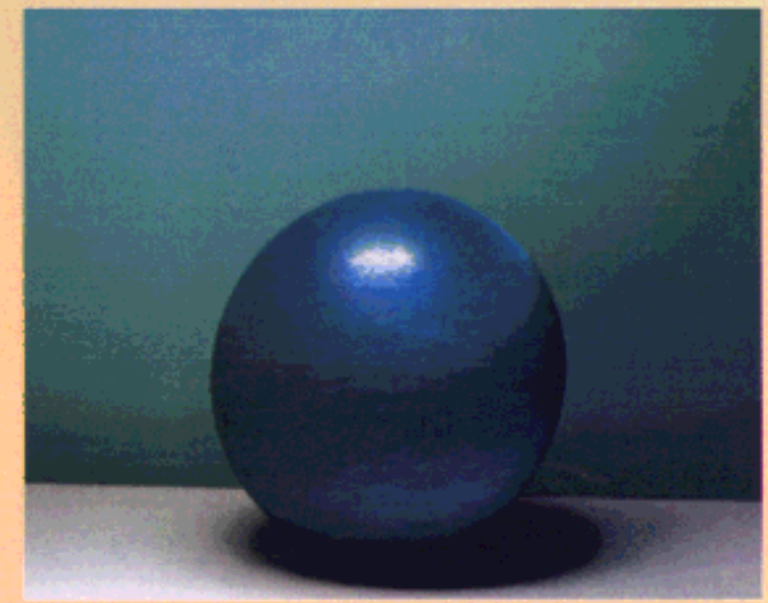
- assume texture patterns "continue"
- avoid blurring
- avoid boundary artifacts

solution:

- apply discrete cosine transform to largest rectangle
- inverse transform reconstructs rectangle exactly
- domain of inverse transform extends arbitrarily

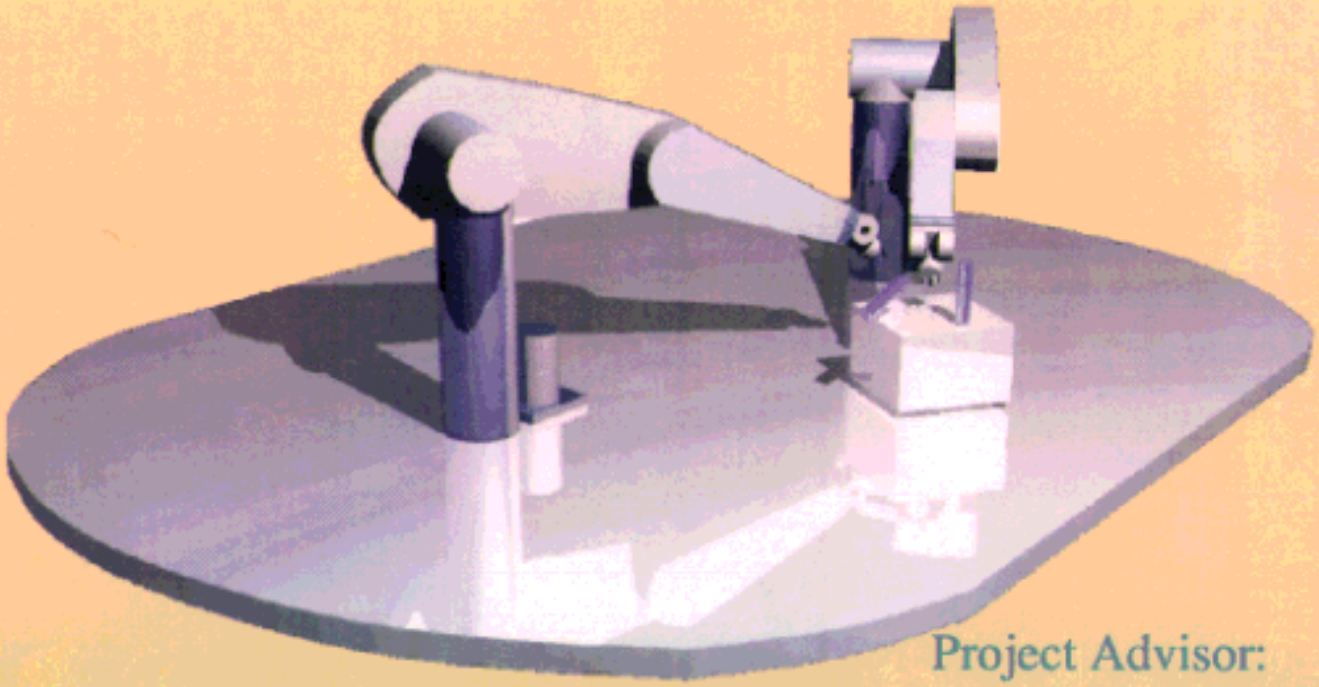


photograph of real scene



automatic virtual replica

Robotics Effort



Project Advisor:

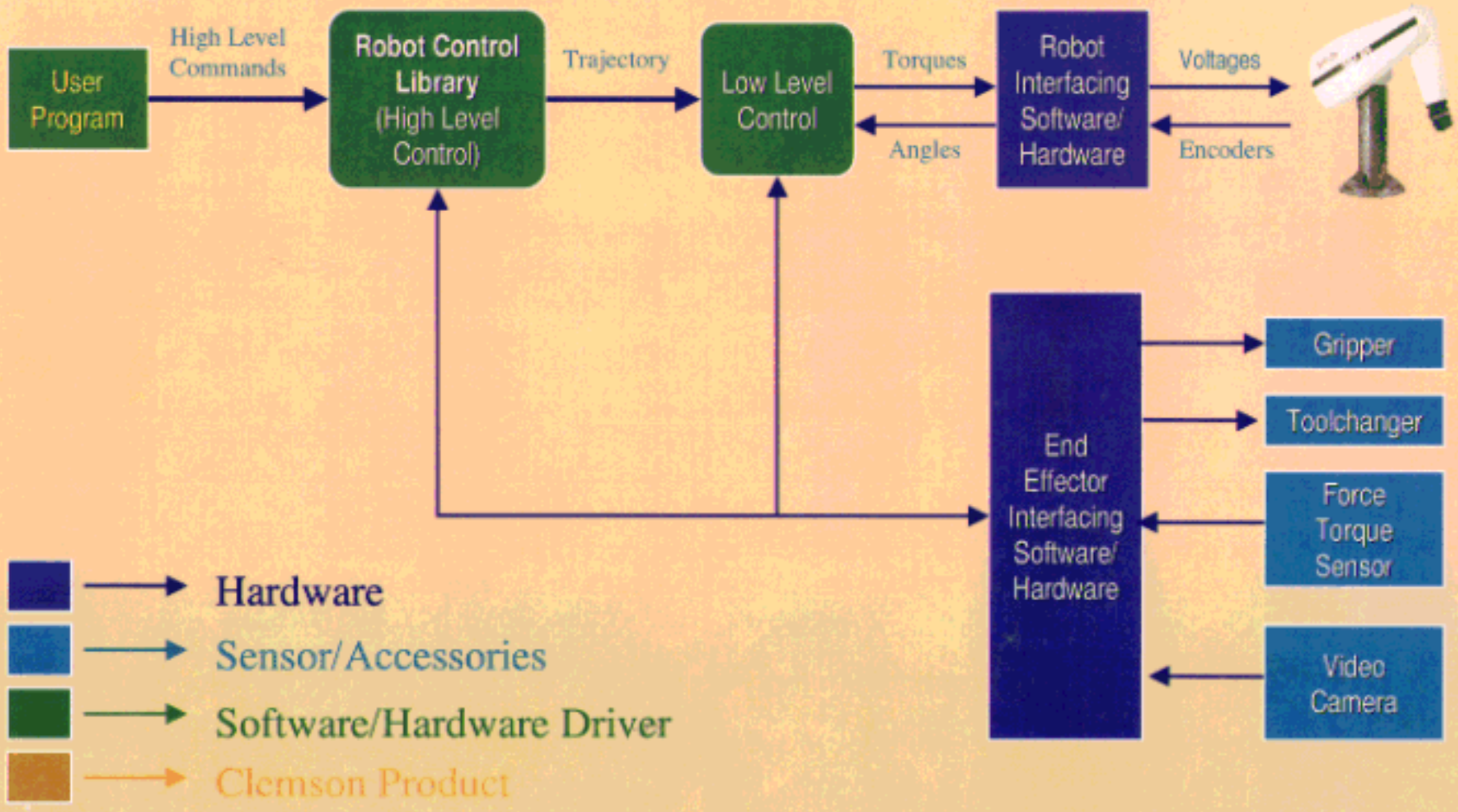
D. M. Dawson

Robotics Team Members

N. Costescu, M. Loffler, E. Zergeroglu, B.Costic

Standard Robot Control System

Background



Hardware

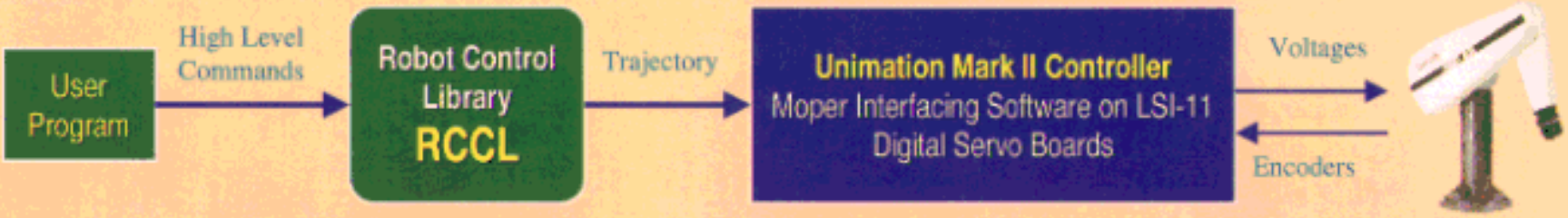
Sensor/Accessories

Software/Hardware Driver

Clemson Product

Standard RCCL Hardware/Software Configuration

Background



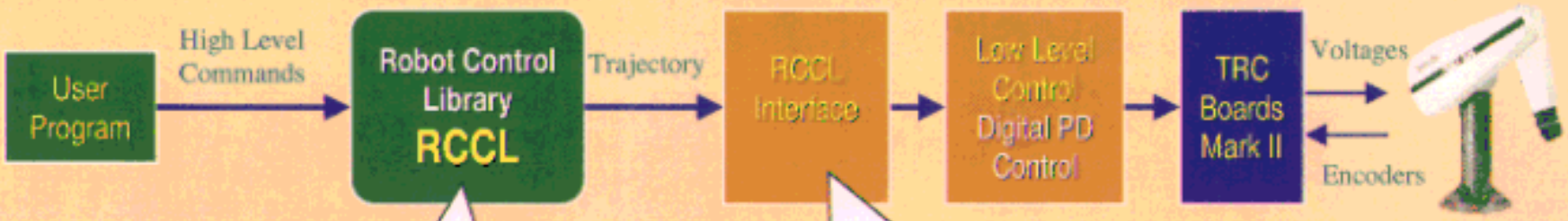
Robot Control C Library (RCCL)

Disadvantages/Problems

- Mark II Controller does not let the user interact directly with the low level controller
- Only a limited number of sensors can be utilized (via RCCL)
- RCCL can be ported to other types of robots but it requires extra hardware and some software changes (not cost efficient)
- RCCL only runs with Unix based systems, and the Unix kernel has to be changed to enable the near-real time properties of RCCL
- An RCCL based system is inflexible and is not user friendly

RCCL-Based Hardware/Software Modifications

Phase 1



RCCL disadvantages:

- No longer supported
- No comments
- Complicated programming style
- User programs are complicated
- Requires a separate workstation

Disadvantages:

- No real-time capabilities over the net
- RCCL behaves differently in simulator mode (Some of the commands do not work in simulator mode)
- TRC Boards can only be used for the Puma Robot; hence the system can not be used with other robot types

ARCL- Based Hardware/Software Modifications

Phase 2

Notes on ARCL:

- Commented
- Smaller than RCCL

- Not supported
- Complicated programming style
- Not as powerful as RCCL
- User programs are easier but still too complicated
- Open loop trajectory generation is utilized

Advanced Robot Control Library (ARCL)

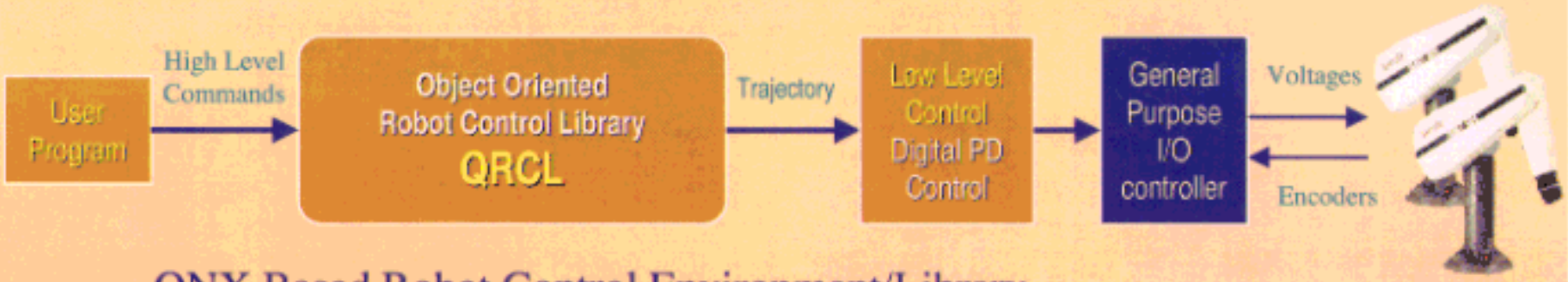
Advantages:

- System can be completely run on a PC
- Integration of sensors is easy
- Easy to modify

Disadvantages/Problems

- Porting of ARCL to QNX is difficult
- ARCL interfacing is difficult to use
- System can be used only with Puma Robots

Final Hardware/Software Configuration



QNX Based Robot Control Environment/Library

Advantages of Proposed System

- Object Oriented Programming is ideal for big and extendable systems
- Less code, easier to read
- Makes full use of QNX's real-time and multitasking capabilities
- Scalable system via client/server techniques
- Allows the low-level control algorithm to be modified easily
- Can be used with many different types of robots
- Easier to integrate with sensors and other telerobotic hardware (Cameras, Joysticks, VR Tools ...)
- User programs will be much simpler