**ORE TECHNOLOGY & EXPERIMENTAL FACILITIES**

- **Project #:** E-25-X48
- **Center #:** 10/24-6-R7615-0A0
- **Contract #:** 92006
- **Prime #:** AA46420T
- **Subprojects ?:** Y
- **Main project #:** 
- **Project unit:** MECH ENGR
- **Project director(s):** BOOK W J
- **Unit code:** 02.010.126
- **Mod #:** ADM. REVISION
- **Document type:** CONT
- **Contract entity:** GTRC

**Sponsor/division names:** ERDA

**Sponsor/division codes:** 500

**Award period:** 921001 to 940331 (performance) 940331 (reports)

**Sponsor amount**

<table>
<thead>
<tr>
<th>Item</th>
<th>Amount</th>
<th>New this change</th>
<th>Total to date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contract value</td>
<td>180,673.00</td>
<td>0.00</td>
<td>180,673.00</td>
</tr>
<tr>
<td>Funded</td>
<td>180,673.00</td>
<td>0.00</td>
<td>180,673.00</td>
</tr>
<tr>
<td>Cost sharing amount</td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
</tbody>
</table>

**Title:** NAVIGATION & COLLISION AVOIDANCE SYSTEM FOR A MOBILE ROBOT

**PROJECT ADMINISTRATION DATA**

- **OCA contact:** Jacquelyn L. Tyndall 894-4820
- **Sponsor technical contact:** DR. RATIB A. KARAM (404)894-3600
- **Sponsor issuing office:** BUSINESS MANAGER (404)894-3600
- **ERDA:** 900 ATLANTIC DRIVE ATLANTA, GA 30332-0425
- **Security class (U,C,S,TS):** U
- **Defense priority rating:** N/A
- **Equipment title vests with:** Sponsor X GIT

**Administrative comments:**

- ISSUED TO EXTEND PROJECT TERMINATION DATE TO MARCH 31, 1994. DELIVERABLE SCHEDULE REVISED TO REFLECT EXTENSION.
GEORGIA INSTITUTE OF TECHNOLOGY
OFFICE OF CONTRACT ADMINISTRATION

NOTICE OF PROJECT CLOSEOUT

---

Closeout Notice Date 04/20/94

<table>
<thead>
<tr>
<th>Project No.</th>
<th>E-25-X48</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center No.</td>
<td>10/24-6-R7615-0A0</td>
</tr>
</tbody>
</table>

| Project Director | BOOK W J |

| Sponsor | ERDA/ATLANTA, GA |

| Contract/Grant No. | 92006 |
| Contract Entity | GTRC |

| Prime Contract No. | AA46420T |

| Title | NAVIGATION & COLLISION AVOIDANCE SYSTEM FOR A MOBILE ROBOT |

| Effective Completion Date | 940331 (Performance) 940331 (Reports) |

**Closeout Actions Required:**

<table>
<thead>
<tr>
<th>Description</th>
<th>Y/N</th>
<th>Date Submitted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final Invoice or Copy of Final Invoice</td>
<td>Y</td>
<td>___</td>
</tr>
<tr>
<td>Final Report of Inventions and/or Subcontracts</td>
<td>Y</td>
<td>___</td>
</tr>
<tr>
<td>Government Property Inventory &amp; Related Certificate</td>
<td>Y</td>
<td>___</td>
</tr>
<tr>
<td>Classified Material Certificate</td>
<td>N</td>
<td>___</td>
</tr>
<tr>
<td>Release and Assignment</td>
<td>Y</td>
<td>___</td>
</tr>
<tr>
<td>Other</td>
<td>N</td>
<td>___</td>
</tr>
</tbody>
</table>

**Comments**

Subproject Under Main Project No.  
Continues Project No.  

**Distribution Required:**

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Y/N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Director</td>
<td>Y</td>
</tr>
<tr>
<td>Administrative Network Representative</td>
<td>Y</td>
</tr>
<tr>
<td>GTRI Accounting/Grants and Contracts</td>
<td>Y</td>
</tr>
<tr>
<td>Procurement/Supply Services</td>
<td>Y</td>
</tr>
<tr>
<td>Research Property Management</td>
<td>Y</td>
</tr>
<tr>
<td>Research Security Services</td>
<td>N</td>
</tr>
<tr>
<td>Reports Coordinator (OCA)</td>
<td>Y</td>
</tr>
<tr>
<td>GTRC</td>
<td>Y</td>
</tr>
<tr>
<td>Project File</td>
<td>Y</td>
</tr>
<tr>
<td>Other</td>
<td>N</td>
</tr>
</tbody>
</table>

**NOTE:** Final Patent Questionnaire sent to PDPI.
A. Project Accomplishments:

The contract was initiated and the five faculty researchers on the four subtasks agreed on a spend down plan that was consistent with the wishes of WSRC. The plan is attached. Subaccount will be established for each subtask, with some adjustments made to accommodate payment of students in each of the departments.

A visit to WSRC to attend the demonstrations of the waste operations research and development was made on 10/24. All faculty researchers and one student attended. In addition to attending the demonstrations in the morning, a meeting with individuals involved in the drum inspection problem to which this project pertains was held in the afternoon. In addition to Georgia Tech and WSRC personnel, researchers from Transitions Research Corporation and the University of South Carolina were present.

B. Milestones Achieved (Based on those identified in the Task Order):

No specific deliverables or milestones were defined for this period. The following activities are identified in the statement of work. WSRC facility was visited and purchase of the Dickerson Video Technologies camera for Subtask III was initiated.

C. Problems Encountered

In view of the desired distribution of payments into two fiscal years, the expenditures will need to be extended into the first quarter of calendar year 1994. A request for a no cost extension has been submitted.
### Subtask 1: Ultrasonic Obstacle Avoidance

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faculty</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6,556</td>
</tr>
<tr>
<td>Secretary</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1,000</td>
</tr>
<tr>
<td>Students</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>0</td>
<td>14,000</td>
</tr>
<tr>
<td>Total personal services</td>
<td>0</td>
<td>0</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>21,556</td>
</tr>
<tr>
<td>Benefits</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2,578</td>
</tr>
<tr>
<td>Computing cost recovery</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1,051</td>
</tr>
<tr>
<td>Travel</td>
<td>75</td>
<td>1,450</td>
<td>0</td>
<td>0</td>
<td>75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2,150</td>
</tr>
<tr>
<td>Materials, supplies, reports</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2,000</td>
</tr>
<tr>
<td>Total burdened direct</td>
<td>75</td>
<td>1,450</td>
<td>250</td>
<td>1,504</td>
<td>1,579</td>
<td>1,254</td>
<td>1,254</td>
<td>1,254</td>
<td>1,254</td>
<td>1,254</td>
<td>1,254</td>
<td>1,254</td>
<td>1,254</td>
<td>1,254</td>
<td>1,254</td>
<td>1,254</td>
<td>1,254</td>
<td>1,254</td>
<td>28,812</td>
</tr>
<tr>
<td>Indirect</td>
<td>34</td>
<td>651</td>
<td>112</td>
<td>675</td>
<td>709</td>
<td>563</td>
<td>563</td>
<td>563</td>
<td>563</td>
<td>1,811</td>
<td>1,811</td>
<td>1,811</td>
<td>1,811</td>
<td>1,811</td>
<td>1,811</td>
<td>1,811</td>
<td>1,811</td>
<td>1,811</td>
<td>12,937</td>
</tr>
<tr>
<td>Capital Equipment</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total for subtask</td>
<td>109</td>
<td>2,101</td>
<td>362</td>
<td>2,180</td>
<td>2,288</td>
<td>1,817</td>
<td>1,817</td>
<td>1,817</td>
<td>1,817</td>
<td>1,817</td>
<td>1,817</td>
<td>1,817</td>
<td>1,817</td>
<td>1,817</td>
<td>1,817</td>
<td>1,817</td>
<td>1,817</td>
<td>1,817</td>
<td>41,749</td>
</tr>
<tr>
<td>Cumulative for subtask</td>
<td>109</td>
<td>2,210</td>
<td>2,572</td>
<td>4,752</td>
<td>7,040</td>
<td>8,857</td>
<td>10,675</td>
<td>12,492</td>
<td>14,310</td>
<td>20,155</td>
<td>26,000</td>
<td>31,845</td>
<td>35,581</td>
<td>38,846</td>
<td>41,640</td>
<td>41,749</td>
<td>41,749</td>
<td>41,749</td>
<td>41,749</td>
</tr>
</tbody>
</table>

### Actual

- Encumbered
- Spent
- Cumulative
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal services</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7,983</td>
</tr>
<tr>
<td>Faculty</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1,100</td>
</tr>
<tr>
<td>Secretary</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2,471</td>
</tr>
<tr>
<td>Students</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>14,000</td>
</tr>
<tr>
<td>Total personal services</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>23,083</td>
</tr>
<tr>
<td>Benefits</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2,150</td>
</tr>
<tr>
<td>Computing cost recovery</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Travel</td>
<td>75</td>
<td>1,450</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2,150</td>
</tr>
<tr>
<td>Materials, supplies, reports</td>
<td>0</td>
<td>250</td>
<td>250</td>
<td>250</td>
<td>250</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2,150</td>
</tr>
<tr>
<td>Total burdened direct</td>
<td>75</td>
<td>1,700</td>
<td>250</td>
<td>250</td>
<td>325</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2,150</td>
</tr>
<tr>
<td>Indirect</td>
<td>34</td>
<td>763</td>
<td>112</td>
<td>112</td>
<td>146</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13,427</td>
</tr>
<tr>
<td>Capital Equipment</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total for subtask</td>
<td>109</td>
<td>2,463</td>
<td>362</td>
<td>362</td>
<td>471</td>
<td>0</td>
<td>1,691</td>
<td>1,691</td>
<td>1,691</td>
<td>6,595</td>
<td>6,595</td>
<td>6,595</td>
<td>3,236</td>
<td>2,765</td>
<td>2,656</td>
<td>2,089</td>
<td>1,980</td>
<td>1,980</td>
<td>43,330</td>
</tr>
</tbody>
</table>
## Spend Down for ERDA Project E25-X48

### Subtask III: Vision (Landmark Tracking) Positioning System

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Personal services</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faculty</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7,278</td>
</tr>
<tr>
<td>Secretary</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Students</td>
<td>0</td>
<td>0</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>14,000</td>
</tr>
<tr>
<td>Total personal services</td>
<td>0</td>
<td>0</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>3,426</td>
<td>3,426</td>
<td>3,426</td>
<td>3,426</td>
<td>3,426</td>
<td>3,426</td>
<td>3,426</td>
<td>21,278</td>
</tr>
<tr>
<td>Benefits</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1,980</td>
</tr>
<tr>
<td>Computing cost recovery</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Travel</td>
<td>75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>75</td>
</tr>
<tr>
<td>Materials and supplies</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>700</td>
</tr>
<tr>
<td>Total burdened direct</td>
<td>175</td>
<td>100</td>
<td>100</td>
<td>1,100</td>
<td>1,175</td>
<td>1,100</td>
<td>1,100</td>
<td>1,100</td>
<td>4,186</td>
<td>4,086</td>
<td>4,086</td>
<td>4,086</td>
<td>4,086</td>
<td>4,086</td>
<td>4,086</td>
<td>25,458</td>
</tr>
<tr>
<td>Indirect</td>
<td>79</td>
<td>45</td>
<td>45</td>
<td>494</td>
<td>528</td>
<td>494</td>
<td>494</td>
<td>494</td>
<td>1,879</td>
<td>1,835</td>
<td>1,835</td>
<td>1,835</td>
<td>1,835</td>
<td>1,835</td>
<td>1,835</td>
<td>11,430</td>
</tr>
<tr>
<td>Capital Equipment</td>
<td>2,500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3,000</td>
</tr>
<tr>
<td><strong>Total for subtask</strong></td>
<td>2,754</td>
<td>145</td>
<td>145</td>
<td>1,594</td>
<td>1,703</td>
<td>1,594</td>
<td>1,594</td>
<td>1,594</td>
<td>6,065</td>
<td>5,920</td>
<td>5,920</td>
<td>2,674</td>
<td>1,703</td>
<td>1,594</td>
<td>1,594</td>
<td>39,888</td>
</tr>
<tr>
<td><strong>Cumulative for subtask</strong></td>
<td>2,754</td>
<td>2,898</td>
<td>3,043</td>
<td>4,637</td>
<td>6,340</td>
<td>7,934</td>
<td>9,528</td>
<td>11,122</td>
<td>12,715</td>
<td>18,781</td>
<td>24,701</td>
<td>30,622</td>
<td>33,295</td>
<td>34,998</td>
<td>36,592</td>
<td>38,294</td>
</tr>
<tr>
<td><strong>Totals</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>39,888</td>
</tr>
</tbody>
</table>
## Spend Down for ERDA Project E25-X48

### Subtask IV: Transputer Control System (for Visual Navigation)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Totals</td>
<td></td>
</tr>
<tr>
<td>Faculty</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1,879</td>
<td>1,879</td>
<td>1,879</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1,879</td>
<td>1,879</td>
<td>1,879</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15,966</td>
</tr>
<tr>
<td>Secretary</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>14,000</td>
</tr>
<tr>
<td>Students</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total personal services</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3,045</td>
<td>3,045</td>
<td>3,045</td>
<td>1,167</td>
<td>1,167</td>
<td>1,167</td>
<td>0</td>
<td>3,045</td>
<td>3,045</td>
<td>3,045</td>
<td>2,731</td>
<td>2,731</td>
<td>2,731</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Benefits</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>511</td>
<td>511</td>
<td>511</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>511</td>
<td>511</td>
<td>511</td>
<td>426</td>
<td>426</td>
<td>426</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Computing cost recovery</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>325</td>
<td>325</td>
<td>325</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Travel</td>
<td>150</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>150</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Materials and supplies</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>2,000</td>
</tr>
<tr>
<td>Total burdened direct</td>
<td>250</td>
<td>100</td>
<td>100</td>
<td>3,674</td>
<td>3,824</td>
<td>3,674</td>
<td>1,284</td>
<td>1,284</td>
<td>1,284</td>
<td>1,284</td>
<td>3,674</td>
<td>3,674</td>
<td>3,674</td>
<td>4,182</td>
<td>3,732</td>
<td>3,582</td>
<td>250</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Indirect</td>
<td>112</td>
<td>45</td>
<td>45</td>
<td>1,650</td>
<td>1,717</td>
<td>1,650</td>
<td>577</td>
<td>577</td>
<td>577</td>
<td>577</td>
<td>1,650</td>
<td>1,650</td>
<td>1,650</td>
<td>1,878</td>
<td>1,676</td>
<td>1,608</td>
<td>112</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>Capital Equipment</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total for subtask</td>
<td>362</td>
<td>145</td>
<td>145</td>
<td>5,324</td>
<td>5,541</td>
<td>5,324</td>
<td>1,861</td>
<td>1,861</td>
<td>1,861</td>
<td>1,861</td>
<td>5,324</td>
<td>5,324</td>
<td>5,324</td>
<td>6,060</td>
<td>5,408</td>
<td>5,191</td>
<td>362</td>
<td>145</td>
<td>145</td>
</tr>
</tbody>
</table>

### Cumulative for subtask

<table>
<thead>
<tr>
<th>Actual</th>
<th>Encumbered</th>
<th>Spent</th>
<th>Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>3,333</td>
<td>8,187</td>
<td>9,202</td>
<td>18,661</td>
</tr>
<tr>
<td>26,664</td>
<td>37,399</td>
<td>44,362</td>
<td>51,324</td>
</tr>
<tr>
<td>58,287</td>
<td>82,116</td>
<td>105,801</td>
<td>129,485</td>
</tr>
<tr>
<td>145,190</td>
<td>158,331</td>
<td>170,566</td>
<td>174,828</td>
</tr>
<tr>
<td>178,547</td>
<td>180,672</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Cumulative for contract | 3,333 | 8,187 | 9,202 | 18,661 | 26,664 | 37,399 | 44,362 | 51,324 | 58,287 | 82,116 | 105,801 | 129,485 | 145,190 | 158,331 | 170,566 | 174,828 | 178,547 | 180,672 |
A. Project Accomplishments:

Student work on the simulator for vehicle motion and obstacle avoidance was initiated. John Hogan is working with students supported by the NSF project to extend that simulation to the needs of the contract for Subtask II. Fred Anders has started work on the vision system characterization of Subtask III.

B. Milestones Achieved (Based on those identified in the Task Order):

No specific deliverables or milestones were defined for this period. The following activities are identified in the statement of work: Purchase of the Dickerson Video Technologies camera for Subtask III was processed based on a sole source justification. It was delivered early in December.

C. Problems Encountered

The establishment of Georgia Tech subaccounts for subtasks has been slow. These need to be in place for hiring additional students in January. There is no apparent obstacle to establishing these accounts in the near future.
We have begun to configure a transputer development station consisting of a host PC, attached transputer processing modules, and a transputer-based digitizer with camera. The FLF code has been studied, and we are about to begin porting it to the transputer system for off-line development while Bill continues to use the robot for algorithm tuning. The onboard transputer host, with five processors, is available for use at any time.

A module has been designed to ease the interface between the onboard host and the newer Denning robots. This module will use transputer modules and embedded firmware to handle the dual serial ports of the robots while presenting both a serial port for workstations like the Sun and MicroVax, as well as transputer links for the onboard host. Either offboard, onboard, or simultaneous control will be possible. Most of the parts for this module are already available. Purchase procedures are underway for a quadruple serial port transputer board.

B. Milestones Achieved (Based on those identified in the Task Order):

No specific milestones or deliverables were planned for this period. We have begun using logbooks and conducting weekly subtask meetings in addition to scheduled monthly project (task) meetings.

C. Problems Encountered

Task 4

No visual data sets available yet for testing algorithms. We need videotape / photos / floor plans of sites for inspection. This is important for accurate code and parameter development. We do not have a point-of-contact for this data.
**Subtask 4 : Transputer Control System**

We have completed hardware design of the transputer-based interface module between the Sun, PC host, and Denning robot. It will reside within the robot's I/O card cage on an STD bus card, which is currently being wire-wrapped. Only one component was purchased -- the quadruple serial port, which has not yet been delivered.

Rather than port the Fast Line Finder code in its present state, we have determined that it would be better to convert it to ANSI C. Bill Wester is working on that task. Bill also purchased the empty drums for the research, and he has set them up in an appropriate room.

**B. Milestones Achieved (Based on those identified in the Task Order):**

No specific milestones are identified in the Task Order.

The level of effort has been according to plan, with four funded Graduate Research Assistant slots (five students involved since two working on Subtask 2 are jointly funded with M.E. School funds). About $2400 were also spent on faculty personal services and directly related expenses on Subtask 4. Also purchased from operating supplies was 40 used drums for experimentation.

**C. Problems Encountered**

No problems were encountered
A. Project Accomplishments:

Project accomplishments were minimal due to Christmas holidays and end of quarter disruptions. Work on the simulation for Subtask II and on the camera programming for Subtask III was continued. Plans were made for meetings in January when additional tasks will be launched and the level of effort will be substantially increased. Subaccounts were created for all subtasks, enabling the faculty and students to be paid in the normal manner.

B. Milestones Achieved (Based on those identified in the Task Order):

No specific deliverables or milestones were defined for this period. The following activities are identified in the statement of work: We received the camera for Subtask III and the student, Fred Anders, met with technical staff from Dickerson Video Technologies regarding current capabilities and programming features.

C. Problems Encountered

No unexpected obstacles were encountered. Holidays reduced the number of work days in the period by over one week.
ERDA MONTHLY STATUS REPORT

University: Georgia Institute of Technology
PI: Wayne J. Book

Project Title: Navigation and Collision Avoidance System for a Mobile Robot

Period of Performance: From 1/1/93 to 1/31/93

WSRC Technical Representative: Clyde Ward

A. Project Accomplishments:

Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning

Task specifications for visualization code for drum testing made. The GRA developed preliminary communications link with robot for code acquisition. Drum suppliers were checked and a supplier located. The drums were inspected and found to be satisfactory for our project. Space was located for testing and drum storage in the 1st floor of MaRC. The GRA is prototyping the visualization code for ultrasound.

Subtask 2: Vehicle Positioning and Obstacle Avoidance

A thorough outline of software for simulation of the vehicle moving in the vicinity of obstacles has been created. The simulator will be programmed in C++ with an object oriented approach. Leveraging to produce the simulator will include GRAs supported by the NSF project on mobile manipulation which has need for similar software. To facilitate the programming tutorials on the basics of robotic kinematics and dynamics have been held once or twice per week during the last month. These tutorials are now conducted on request or during the weekly group meetings about this and related projects.

Subtask 3: Vision (Landmark Tracking) Positioning System

GRA Fred Anders is continuing investigation into the optics, lighting, and landmark recognition algorithms necessary to navigate a mobile robot within a storage warehouse using the DVT Integrated Vision System. Retroreflective landmarks in conjunction with incandescent lights has proven ideal for such indoor tracking, and further exploration into strobe lighting will hopefully extend the range and sampling rate and reduce the power consumption of the system. A servomotor-based gimbal platform has been constructed to augment the field of view of the cameras, and the next step is to verify the resolution and positioning accuracy of the system. A mock up of the SRP storage facility has been created in one of the MaRC research labs to facilitate testing both this general navigation as well as the close maneuvering/barrel inspection tasks.

Subtask 4: Transputer Control System

The GRA for this task, Bill Wester, began working with the Fast Line Finder (FLF) algorithm, to be used in identifying drum edges for corridor-following. He has successfully used the algorithm on the Denning robot, "George," to traverse short hallways, remaining near the center line. The FLF is running on a Microvax system with an attached Gould digitizer. Bill has also located a source for empty drums, and purchase procedures are underway. He has started to study the Fast Region Segmenter algorithm, which will be used also.
University: Georgia Institute of Technology

PI: Wayne J. Book (coordinator)
Co-PI: Ronald Arkin
Thomas Collins
Steve Dickerson
Andrew Henshaw

Project Title: Navigation and Collision Avoidance System for a Mobile Robot

Period of Performance: From 2/1/93 to 2/28/93

WSRC Technical Representative: Clyde Ward

A. Project Accomplishments:

The project received a visit from the Quality Audit Team consisting of Bob Young and Don Strosnyder on 2/25/93. Useful suggestions were made in the area of keeping technical records and maintaining software revisions.

Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning

The robot is now equipped for data gathering and the preliminary visualization environment for interpreting sonar results is in place. The system is based on X-windows and has a variety of options for replaying sonar data. Preliminary test runs have been made in the mobile robot laboratory to validate the code. Our objectives for next month include moving the robot to the downstairs lab and gathering data from the actual rows of drums. This is a bit of a logistical problem requiring us to relocate our computer work stations on a temporary basis, but should not prove overly difficult.

Subtask 2: Vehicle Positioning and Obstacle Avoidance

To facilitate coding of the simulator a special account has been established to maintain the latest encoded simulator module versions. Rough originals will be developed and tested on GRAs accounts. The simulator account contains History and Checkout files, to be updated as new versions of the software are moved between accounts. Back-ups of the simulator account will be updated weekly. Test programs in the simulator account have been developed to ensure changes to modules do not affect older version's functionality and compatibility. The output from these test programs are compared and the differences alerted to the user.

Recent additions to the simulator account are basic math modules needed to describe vehicle's kinematics (Vector, Matrix, coordinate Rotation, coordinate Transformation, LU Decomposition, Back Substitution and their associated functions).

Subtask 3: Vision (Landmark Tracking) Positioning System

A complete 3 camera system with servo controlled gimbaled vision heads has been demonstrated. This integrates 3 vision systems, 3 microprocessor controlled head drives, and one 386 PC. Current accuracy over a roughly 100 foot by 50 space is about 1 foot. No subpixel algorithms are being used. The head drives have backlash, which is the greatest source of error. Current efforts are on removing the backlash, installing subpixel algorithms, building and testing a high power halogen lighting system, and testing xenon strobe illumination.
A. Project Accomplishments:

A written report on the Quality Audit conducted in February was received. In response a statement was submitted outlining how the suggestions would be implemented to produce labbooks that improved documentation of the project.

Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning

Two programs are finished and working: SAMPLER and TOOL. SAMPLER is used to get sensor data from the robot and TOOL is used to visualize sensor data. SAMPLER has the option to output actual data gathered directly from the robot or old data gathered from a file. It also has the option to store the actual data into a file for later use. SAMPLER can accept a script of robot actions specified in a command file. These actions can be executed at run-time on user demand. While running, SAMPLER accepts commands from keyboard to get ultrasound sensor samples, get robot status, and execute the next action of the command file.

TOOL is used to visualize ultrasonic data previously gathered by the SAMPLER program. It has a control panel that operators use to select the samples and sensors they want to visualize. The program displays for each sample, the position and orientation of the robot, the "raw" model of each sensor (arcs), the segments induced by the sensor models, the projected drums positions, and the world configuration as it is supposed to be. The line segments are constructed using temporal sensor sampling extrapolation. Drum position and size estimation are constructed using "tangent fitting" (fit a circle within near segments were the segments are considered as tangents to the circle).

Subtask 2: Vehicle Positioning and Obstacle Avoidance

Several steps have been taken to protect the integrity of the simulator code which is being developed by three people. A special account has been established to maintain the latest encoded simulator module versions. Rough originals will be developed and tested on Graduate Research Assistant accounts. The simulator account contains History and Checkout files, to be updated as new versions of the software are moved between accounts. Back-ups of the simulator account will be updated weekly. Test programs in the simulator account have been developed to ensure changes to modules do not affect older version's functionality and compatibility. The output from these test programs are compared and the differences alerted to the user.

Recent additions to the simulator account are basic math modules needed to describe vehicle's kinematics (Vector, Matrix, coordinate Rotation, coordinate Transformation, LU Decomposition, Back Substitution and their associated functions). The graphics display modules are also under development. Our goal is to have a demonstration of the basic simulator capabilities by May 4.

Subtask 3: Vision (Landmark Tracking) Positioning System

Work continues on construction of the camera systems. Better servomotors and a different
control scheme should improve camera accuracy and repeatability. Debugging of the triangulation and camera integrating software continues as more cameras are introduced into the system. Once the system has been proven in an off-board configuration, we plan to fix the targets, which are now moving, and move the cameras, which are now stationary, with the vehicle. This allows the passive targets to be left behind in the storage building and requires the minimum of more expensive and less reliable active cameras. We have begun to solicit information about the vehicle from that group, but many questions about the working environment itself (warehouse arrangement, ambient lighting, etc.) must be answered before several critical design decisions can be made.

**Subtask 4: Transputer Control System**

It was necessary to set up stand-alone Sun workstations in order to be able to conduct research in the room containing the aisles of drums. The code for the fast line finding algorithm (fl.c) was converted to ANSI C, so that it is now fully compatible with the Sun workstations. The method for capturing images was changed, and they are now are captured and displayed more quickly.

The quadruple serial port for the transputer subsystem has been delivered, and the hardware is being assembled.

**B. Milestones Achieved (Based on those identified in the Task Order):**

No milestones were identified in the task order for this reporting period.

**C. Problems Encountered**

A problem previously noted was the difficulty in getting a data sets (images) for testing the vision algorithms for identifying and moving down the aisle of stored drums. The impact of this problem is growing. We have been told that it will take "months" to get approval to receive images of the actual sites to be inspected. This will run late into the contract period. It will seriously impact Subtask #1 by not having specifics regarding drum location and will force Subtask #4 to use images from simulated sites. While the significance of this is unknown, it clearly reduces the impact of the results. If simulated sites are to be used, it would be very desirable that these sites be created by WSRC personnel who have seen the conditions at the actual sites. Conditions include, but are not limited to, lighting, surface finish and arrangement of drums forming the aisles, floor texture, and colors of all objects.
A. Project Accomplishments:

In addition to the normal stimulus in the subtasks described below, Atlanta was the site for the 1993 IEEE International Conference on Robotics and Automation. The students and faculty were able to attend many technical sessions related to this project. Tours on campus provided incentive to bring demonstrations of the capabilities on-line. Unfortunately, WSRC personnel at the conference could not adjust their schedule to visit the laboratories.

Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning

We performed extensive experiments using different ultrasonic firing parameters and got successful results. We used four different environments in the tests: aisles with widths of 1 foot, 2 feet, 3 feet, and 4 feet of free space surrounding the robot were used, and four different combinations of ultrasonic firing parameters. This produced a total of 16 data files with 35 samples each providing enough data to determine the behavior of the ultrasonic ranging in this kind of environment. We augmented the visualization program with an algorithm to recognize salient features that are reliable to count the drums along the aisle. We designed a relaxation-like process to perform this task. Other alternatives are being considered, these include mean square fitting of slopes and determination of inflection points, and statistical pattern recognition using Bayesian methods. None of these alternatives has been completed yet.

The goals for this month are to analyze the data and determine which firing parameter sequence is the best for detecting salient features to perform robotic navigation. Also, we will compare different methodologies in detecting these salient features. To accomplish these goals we are going to implement and run different algorithms for feature detection using all the data files and gathered from the experiments. We are going to test different configurations of the relaxation method to determine the feasibility of this approach for drum counting.

Subtask 2: Vehicle Positioning and Obstacle Avoidance

Many accomplishments have taken place in the development of the simulator graphics. The C graphics library SPHIGS has been encapsulated in C++ code. This allows us easy access to SPHIGS commands for the development of graphic primitives and objects.

Any mechanisms involving implemented links can be created and manipulated directly. A mechanism is defined as a list of links with their associated graphical description. Kinematics associated with the links are complete. Currently Denavit-Hartenberg links (revolute, prismatic, unactuated) as well as nonholonomic links (steerable knife and tricycle) are modeled. Uniting these accomplishments has resulted in rapid development of mechanism animation. Our code now animates a Denning-like vehicle, with an attached 5 DOF manipulator, moving along an aisle of barrel pallets.

Subtask 3: Vision (Landmark Tracking) Positioning System
We are now testing the accuracy of the landmark tracking system in the offboard configuration. Statically (or with a slow moving vehicle), the system currently exhibits accuracies of 1.5 feet at a distance of 120 feet. To achieve the targeted 1" resolution, it is clear that an improved, feedback resolver-equipped camera pan mechanism will have to be employed, as the 1 degree backlash on the servos now being used open-loop is by far the primary source of error. Once this error has been reduced, investigation into finer tuning (such as through camera subpixelization) will be applicable. Since the system as a whole is currently providing position results at rates exceeding 36 Hz, we are looking into ways to apply some of the extra computational power toward collecting raw data more intelligently and smoothing the data points to eliminate noise spikes.

Subtask 4 : Transputer Control System

Bill Wester has continued to port the flf (fast line finder) code to an ANSI C version which will be compatible with the Sun host or the Transputer-based host. The algorithm is now partly working on the Sun, and it should be ready for porting to the Transputer soon. Since we will soon lose access to the large room where we have set up the metal drums, Bill has also been gathering data to be used in subsequent vision processing trials.

The Transputer board has been completely wire-wrapped and tested. We will have to replicate it so that we will have a total of eight Transputer module "sites," half of which will be occupied by the Transputer-based frame-grabber. These Transputer boards have been made in the STD-bus form factor so that they can be installed in the I/O card cage of the Denning MRV-2 robots or in an externally-mounted card cage on the older robot, "George." Some of the onboard robot code (to handle the serial communications with the robot and the optional remote host) is under development and will be tested shortly.

B. Milestones Achieved (Based on those identified in the Task Order): No specific milestones or deliverables were planned for this period.

C. Problems Encountered

Mechanism animation is quite slow. The utility of a graphics accelerator card is being explored.

There have been unexpected delays in transferring the task subaccount from the School of Electrical Engineering to the Georgia Tech Research Institute, where the task co-investigators are now based. This does not affect the eventual completion of the task on schedule, but it has limited the amount of effort which could be allocated over the past two months. It is expected to be resolved this month.

The drums used for experimentation will have to be moved from their current location on the first floor of the Manufacturing Research Center Building because the space is committed to other purposes. The new location will be on the third floor in the high bay area. This space has will not be dedicated to a single purpose, will have different lighting conditions and floor surface, but should cause no substantial problems in the research.
A. Project Accomplishments:

All of the subtasks were involved with demonstrations to the visitors to campus who were attending the 1993 IEEE International Conference on Robotics and Automation. Much useful feedback was obtained from the visitors as well as visibility for the project.

Plans were made for Tom Collins and Ron Arkin to attend accompany Clyde Ward to the Fernald site to see first hand the inspection problem that the research and development should be applicable to.

Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning

Different configurations of the relaxation algorithm for drum recognition were considered. The earlier versions were not successful because they did not consider all the data. The later versions use new formulas and were developed to overcome this situation. Now the algorithm is more reliable. There are still some problems with local minima. The alternative of mean square fitting resulted in more complications than anticipated since a directed sorting process must be performed before analyzing the data. A preprocessing algorithm was also considered to simplify the drum detection algorithms and to enhance the quality of the data. The Bayesian alternative was discarded since its incorporation would require an expensive change in the data representation structures. The goal for this month is to finish the preprocessing algorithm and use it with the two alternatives for drum detection. Also, we will run comparative experiments to evaluate the robustness and quality of the two alternatives considered for drum detection.

Subtask 2: Vehicle Positioning and Obstacle Avoidance

The development of the vehicle/attached arm simulator is now capable of displaying the moving system on the screen according to predefined motion. Present work is focused on realistically representing dynamics of system motion and response of sensors that would be useful in obstacle avoidance. The dynamics will be important in realistically predicting the ability to change directions or stop to avoid an obstacle.

The inverse dynamics (computing joint and wheel torques given position, velocity, acceleration) with associated properties (such as inertia matrices and mass) of multibody mechanisms is complete. We can now calculate the needed torque to carry out a maneuver prescribed in terms of of system motion. The inverse dynamics utilizes an iterative Newton-Euler routine. The next step will be predicting motion given the joint torques, such as would be commanded by an obstacle avoidance scheme and control algorithm.

A C++ object has been developed to simulate data from user placed sensors. The object quickly scans all graphics elements within the simulated environment and returns a distance to the nearest element that lies in the direction of the sensor. In this respect, the sensor functions similar to a range finder. The program allows the sensor to be placed anywhere on the vehicle and articulated relative to the vehicle.

Plans have been made to implement the TRC type vehicle into the set of available joints. The implementation will be used as a test case to evaluate the ease of implementing a new joint type with the object oriented form of the program.
One student, John Hogan, will be working in a summer job beginning in June. Paul Schmitt will continue to work on the project.

**Subtask 3: Vision (Landmark Tracking) Positioning System**

Mr. Fred Anders completed his M.S. thesis on the topic of vehicle tracking using the integrated vision system and both retroreflective and standard targets. In the case of the thesis the primary vehicle of interest was a model helicopter which flies outdoors. Most of the thesis research is directly applicable to the measurement of AGV position. A separate report of Mr. Anders has been drafted for submission to WSRC. Mr. Anders expects to complete this report in June while still at Georgia Tech. Experiments at 75 foot range indicate that linearity of measurement down to less than 0.1 pixel is readily achieved. In most situations this would correspond to 0.03 to 0.2 inches.

**Subtask 4: Transputer Control System**

During this month, we have built and tested a Transputer motherboard that fits the STD form factor used by the Denning robots. This motherboard holds 4 Transputer modules and will be replicated to provide a total of 8 processors and/or I/O modules. The 4-port Transputer-based RS-232 module has been tested and communication software using this module has been written. This software/hardware combination allows a network of Transputers to control, or be controlled by, as many as 4 separate RS-232 devices. The communication software transparently routes packets between all of the connected devices. A monolithic programmable logic device has been designed and programmed as part of a switch-sensing system that can be read through a Transputer link. The circuit board to contain this chip has not yet been fabricated. The Fast-Line-Finding routines have been completely ported to ANSI C and are running on a Sun workstation. A simple menu system has been added as a front end for the software.

**B. Milestones Achieved (Based on those identified in the Task Order):**

No milestones were identified in the Task Order.

**C. Problems Encountered**

No problems were encountered.
A. Project Accomplishments:

Ron Arkin and Tom Collins accompanied WSRC personnel to visit the Fernald site to see first hand the inspection task at that location. This was very helpful and the information has been transferred to the other project investigators in an oral and a written presentation.

Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning

The preprocessing algorithm is finished and tested. Preliminary experiments show that it did not improve significantly the performance of the relaxation algorithm but it did improve the performance of the mean square fitting algorithm.

The goal for this month is to prepare a demonstration program that consists of identifying a specific drum while navigating an unknown hallway as well as performing more experiments to compare the relaxation and mean square fitting algorithms working in cooperation with the preprocessing algorithm.

Subtask 2: Vehicle Positioning and Obstacle Avoidance

Present work continues on dynamics and sensors. The inverse dynamics (computing torques from specified acceleration, velocity and position) has been coded and verified with several published and internal cross checks. Currently we are encoding forward dynamics (computing position, velocity and acceleration given joint torques). We are currently testing the Sensor C++ object. Efforts are aimed at placing a sensor at any desired location on a moving multibody. The current version displays the distance to the surface of a sphere that encloses an obstacle. We are also currently brainstorming ideas to shape our User Interface.

A laboratory videotape has been produced that will be sent to the technical personnel at WSRC to further describe the status.

Subtask 3: Vision (Landmark Tracking) Positioning System

A demonstrable system for measuring vehicle positions has been constructed. Although still running basically the same vision algorithms, the cameras are now physically mounted on the "vehicle", while the supervisory PC software for triangulation has been modified slightly to reflect this change. A workspace of the approximate dimensions as the storage warehouses (180' long aisles) was measured out and used to empirically confirm the system's accuracy. Using a retroreflective bar code to gauge the offset from the center of the aisle as well as the distance to the ends, preliminary experiments indicate that the system is able to resolve lateral deviations (from the aisle centerline) to +/- 0.5", which was the targeted accuracy. Although not as important of a requirement, longitudinal positioning was accurate only to +/- 10 feet. The use of simplified test landmarks or potential software bugs are suspected to be causing the error in this latter case.
As these tests were conducted using a 300 watt incandescent bulb, the next goal will be to evaluate the effectiveness of using lower power xenon strobes or IR LED banks for illumination. Additionally, the accuracy data needs to be further substantiated through more testing under various conditions.

**Subtask 4: Transputer Control System**

A Transputer development system has been set up in the Mobile Robotics Laboratory. This includes the ANSI C compiler which is now available to use on the fast-line-finder code which has already been rewritten to be ANSI C compatible.

A second Transputer motherboard has been completely wire-wrapped and tested, now providing a total of eight Transputer module "sites." The low-level serial communications drivers for the robot were tested, and intermittent character loss was discovered. The problem was eventually traced to the RS232 interface board which had been purchased for this work. The manufacturer was contacted, and they are supplying us with two dual-RS232 boards to replace the single quad-RS232 board. Early indications are that these boards will work fine and that the robot will be running autonomously under Transputer control within a few days. The system will then be ready for the integration of the visual navigation component.

The meeting at the Fernald site provided answers to all questions pertaining to drum configuration. One unexpected feature was that many of the drums are larger than 55 gallons (usually 85-gallon or 110-gallon), resulting from successive repackaging. From a visual navigation standpoint, this means that the aisle edges will sometimes appear as alternating "in- and-out" drums, rather than uniform scalloped lines.

It was also pointed out that the robot runs will normally be made at night. Since the lighting is inconsistent, this will probably necessitate a robot-mounted headlight, especially in the large fabric "sprung structures." Fernald has been provided with a scenario for a video which approximates the view seen by a robot travelling through their drum storage areas. This video will then be used to provide test images for the vision algorithms.

**B. Milestones Achieved (Based on those identified in the Task Order):**

No milestones were specified in the Task Order.

**C. Problems Encountered**

Some delay resulted from the time spent isolating the RS232 problem and waiting for replacement hardware.
University: Georgia Institute of Technology  
PI: Wayne J. Book

Project Title: Navigation and Collision Avoidance System for a Mobile Robot

Period of Performance: From 7/1/93 to 7/31/93  
WSRC Technical Representative: Clyde Ward

A. Project Accomplishments:

All four subtasks were represented in a trip to WSRC on July 27. The new mobile robot from Transitions Research Corporations, code named SWAMI, was viewed and discussed. Additional information was collected and requested for modeling the vehicle in future work. In the afternoon the directions of research at Georgia Tech were discussed.

A visit by WSRC personnel to Georgia Tech to view the status of the work was scheduled for August 24.

Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning

The robot interface program is capable of performing all the basic command functions for the Denning MRV-2 robot: home, initialize, move forward, rotate, start/stop ultrasonic, change ultrasonic parameters. The graphical interface program is able to gather (and save on demand) telemetry data and plot it in an on-line fashion. With one hour of work the graphical interface will have a robot control panel with all the basic commands. This will make a complete operational interface for robot teleoperation. I expect to complete the autonomous interface the first week of August. The autonomous interface will consist on only three motor schemas (move-to-goal, avoid-obstacle, and wander).

There are two working versions for the drum finding algorithm. The first one consists of spawning two drum-shapes at either side of the robot at fixed distance periods. The drum-shapes move themselves until they match with the segments constructed so far. The second one runs a nearest-neighbor clustering algorithm on the segments constructed so far and places the drum-shapes in the center of mass of each cluster, then the drum-shapes move themselves until they stabilize.

Subtask 2: Vehicle Positioning and Obstacle Avoidance

We are currently in the process of upgrading the Mechanism and Link objects to accept a sensor description script file. Once this is accomplished, the user will be able to position a number of sensors at desired locations along a specified link. Each link has an optional list of associated sensors. The number and positions of these sensors can be changed without recompilation. For easy command reference each sensor is required to have an identification number. The graphics associated with each declared sensor is added to the link graphics description. Redefining the link graphics description in this way automatically utilizes the link graphics functions for sensor graphics.

The preliminary data available on the TRC vehicle was used to program the graphics file for SWAMI. This required less than 5 minutes, showing the capability of the simulation package to be customized for new robots. Additional detail will be corrected based on new data, and the kinematics and dynamics of the vehicle will be incorporated in the coming month. The proposed addition of sensors to SWAMI can then be studied.
A preliminary scheme for incorporation of obstacle avoidance when maneuvering in close quarters has been proposed. This needs to be compatible with Subtask 1 while exploring the alternative provided by explicit consideration of vehicle kinematics and dynamics.

**Subtask 3: Vision (Landmark Tracking) Positioning System**

During the month of July, Mike Schreiber took over from Fred Anders as the principle graduate research assistant on the Vision Positioning System Task. He built a strobe based retro-reflection test apparatus and was able to verify that distances of 180 feet were readily achieved for an indoor environment. That is, a standard, photographic strobe commonly used with 35 mm cameras, has sufficient power to nearly saturate the vision detector at 180 feet. The exposure time used was 0.002 seconds. He is now turning to the task of constructing a rapid fire strobe (up to 10 per second) with computer controlled intensity. Tasks for August also include design and analysis work with fiducials for ranges between 3 feet and 180 feet.

**Subtask 4: Transputer Control System**

The fast line finder (fl.c) has been successfully compiled on the Transputer system. This verifies that the code was completely converted to ANSI C specifications. The next step is to develop graphic display routines to run on a display Transputer. The system will then be used to process sample images. The Sun-hosted version of the fast line finder was used to process several images of drums that were set up in our laboratory.

Four videotapes of sample images were supplied by Fernald, based on suggestions supplied by Collins and Arkin. They visited the Fernald facility the previous month and became familiar with the drum storage areas. These videotapes have been scanned, and they appear to be suitable for processing. Several variations in lighting, location, camera angle, and motion path are available.

The Transputer-based controller was thoroughly tested in a robot configured for the AAAI (American Association for Artificial Intelligence) Conference in Washington, DC. A similar configuration will be used, with vision added, for the vision-guided tests on an actual robot using the fast line finder.

**B. Milestones Achieved (Based on those identified in the Task Order):**

One of the trips to WSRC as identified in the Task Order was completed.

**C. Problems Encountered**

The information on the hardware and algorithms of the TRC vehicle that is needed for prediction of its performance is not readily available. Dave Wagner said he would try to provide the information, but he will be leaving WSRC in August.
A. Project Accomplishments:

The project investigators hosted a team of technical WSRC personnel consisting of Clyde Ward, Larry Harpring, Randy Singer and Kurt Peterson on August 24 from 10 am to 4 pm. All subtasks were discussed and the software and hardware was demonstrated. As one byproduct of the meeting, helpful written comments from Clyde Ward were received and distributed to the investigators. Action items from the meeting include (a) Georgia Tech commits to generate a three to five minute videotape to be shown at the November 2 to 4 demonstrations at WSRC and (b) Clyde Ward will produce a job description for student (e.g. Co-Qp) temporary jobs at WSRC.

Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning

The demonstration program is finished and tested. It consists of a navigation platform for reactive navigation developed under C++ integrated with the visualization program developed earlier (TOOL). The navigation platform can perform on the real robot as well as on simulated worlds. The goal of this month is to fine-tune the drum detection algorithm based on clustering+relaxation methods. We are going to try different shape models to cope with drums that are too close and that are currently being detected by the algorithm as a single object. We also are going to extend the program so it can work with only pre-specified ultrasonic sensors. Then, we will perform experiments and determine the impact on the number of sensors and sensor distribution on the performance of drum detection algorithm.

Subtask 2: Vehicle Positioning and Obstacle Avoidance

Mechanism and Link objects have been upgraded to accept sensor description script files. The user is able to position a number of sensors at desired locations along a link. The number and positions of these sensors can be changed without modifying or recompiling code.

The following steps were taken to simulate SWAMI, the Westinghouse Savannah River Site testbed vehicle. The vehicle's kinematics (i.e., variables and equations that describe the motion of the link) were incorporated into a new class derived from the parent Link class. 50 minutes were needed to encode this DualWheel class, indicating the versatility of the object oriented programming approach that has been taken. SWAMI took shape as its dimensions were set into a graphics description file. Approximating SWAMI with 4 rectangular prisms was a 28 minute task. Next, sensors were added to SWAMI through a sensor description file. Positioning and orienting SWAMI's 24 sensors took 58 minutes. Another hour was needed to develop a routine to test the accuracy of the above files and code.

To drive SWAMI, the user specifies angular velocities to the motor-driven wheels. For example, if both velocities are of the same magnitude and positive, SWAMI moves forward along a straight line. If the signs are opposite, SWAMI spins in place.

Work has begun on a "Map" object. Such an object will accept and store sensor input,
manipulating the data appropriately as the vehicle moves.

**Subtask 3: Vision (Landmark Tracking) Positioning System**

Mike Schreiber has completed a working Xenon strobe system for the tracking system developed by Fred Anders. Current rate is about 10 per second (vision system itself is faster). He has completed a preliminary stepper motor drive as well and is in the process of integrating into the system to replace the analog scan system used by Anders. A more "industrial grade" stepper system has been ordered. The hope is to bring the scanning drive errors down closer to the level of the vision systems errors. The drive Anders's system had about 100 times the error that was in the vision system with subpixelization. However, this error will be eliminated for most of the anticipated applications by fixing two heads together. Then the error of the drive will only contribute (in most situations) to the knowledge of the vehicle heading.

Work is progressing on more elaborate algorithms and fiducial geometries. Progress on this front is the main objective for September. Focusing and dynamic control of Xenon strobe is also an objective.

**Subtask 4: Transputer Control System**

A visit was made to WSRC last month to provide feedback and to learn more about the intended application on SWAMI. It was pointed out that many of the components of the GISC controller are in place and that the TRC machine has performed well in initial tests. Considering this, it appears that the best use of Transputers (and the fast-line-finder) is to incorporate them within a dedicated sensor system that is compatible with the GISC architecture. Several problems remain, including backing out of aisles and integrating a large amount of image processing hardware for drum image acquisition and storage.

The fast-line-finder core routines (fl.c) were successfully compiled and tested (with very few changes) on the Transputer system, using test data which could be compared with results from the Sun version. Initially, a simple menu-driven calling program was used, but later the complete driving program was used, which includes visual serving and parameter setting based on expected corridor location. This program (flf.c) also ported with relatively few changes.

In order to more directly address WSRC's needs, the Transputer-based fast-line-finder was restructured as a minimal, stand alone system, and a demonstration was developed that used actual images from Fernald to visually servo a pan-tilt mechanism. (It made no sense to servo a robot in this open-loop situation.) Although no study has yet been performed to determine optimal parameter settings for the fast-line-finder over the range of expected images, the demonstration system showed that approximate corridor position was usually determined correctly. In particular, the larger drums (and the correspondingly-wider spacing) presented no difficulties.

This demonstration was presented to WSRC during a visit to Georgia Tech, along with a demonstration of the Sun-based fast-line-finder serving an actual robot between drums. WSRC indicated that they are concerned about accuracy of serving (within 4 inches of center line). This can only be addressed with additional testing.

Upcoming work will specifically address
- GISC compatibility of the Transputer subsystem,
- potential use of same hardware for drum image acquisition, compression, and storage, and
- improved performance and accuracy of the fast-line-finder.
B. Milestones Achieved (Based on those identified in the Task Order):

Specific milestones for this period were not identified.

C. Problems Encountered

No problems were encountered.
A. Project Accomplishments:

Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning

The drum detection algorithm was fine-tuned and now can perform with different polygons one at a time. Polygons with 'V' and 'W' shape where tested to verify if the detection algorithm can find the point where two drums are touching. A qualitative comparison of the two polygon shape resulted in the 'W' polygon producing in better performance than the 'V' polygon and only when there the drums in the environment are very close to each other. For drums that are far apart the standard 'semi-circle' polygon produced the best performance. The goal for this month is to extend the detection algorithm to process different polygons simultaneously an pick the one that results the best. With this capability, the detection algorithm will be able to select automatically which polygon is the best for describing the outline of a physical object.

Subtask 2: Vehicle Positioning and Obstacle Avoidance

John Hogan, GRA, returned from his summer co-op assignment at Ford to join Paul Schmitt working on the project. We set up the quarter's main priorities. The main priorities will be to work on kinematics, dynamics, sensor, and map.

The kinematics will be integrated from either velocities or accelerations. The velocities or accelerations could be driven by information from the map or read from an input file. These input files could be created from off-line control sources or previously calculated simulations. The inverse dynamics output a list of torques and will plot to a real-time graph.

The new Sensor_Data class holds information specific to each sensor. Information.
Sensor_Data transforms sensor generated output to useful map input such as the sensor's position, orientation, maximum range, and angular spread.

The map will record in 2-D the results retrieved from Sensor_Data and the sensor readings.

Subtask 3: Vision (Landmark Tracking) Positioning System

A strobe illumination system with indoor range of more than 200 feet was achieved with average power level of 1.5 joules per strobe. This means that at 5 strobes per second, the system would draw approximately 10 watts per vision head AT 200 FEET. Power levels would be lower at shorter distances by essentially the square of the distance. Further research could probably further reduce power by greater focus of the energy. Work is underway on the design of a computer controlled strobe intensity. The other primary progress in September was the implementation of a stepper motor drive for the vision heads. No data on repeatability has yet been taken.

Next month (October) our primary goals are a dual head system with integrated strobe and stepper motor control, and associated data on accuracy/repeatability.
Subtask 4: Transputer Control System

No activity took place on this task this month.

B. Milestones Achieved (Based on those identified in the Task Order):

No milestones were identified in the task order for this period.

C. Problems Encountered

No notable problems occurred during this period. The fall quarter began and progress was somewhat interrupted by the usual academic activities during these periods.
A. Project Accomplishments:

Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning

The goal for extending the multiple shape detection algorithm was accomplished only partially. The were two main reasons for this. First, during testing it was discovered that the velocity of robot influences the performance of the segment matching algorithm. When the robot moves the outline segments are short or long whether the robot moves slow or fast respectively. The segment matching algorithm works best when the length of the outline segment is approximately the same as the length of the segments in the model. This means that the performance of the matching algorithm is directly dependent on the velocity of the robot. Second, considerable time to produce the video tape that demonstrates the research we have done so far. The solution for the problem stated above was to use orthogonal distances instead center-of-mass distances in the segment matching algorithm. In this way the dependence in robot velocity (i.e. outline segment length) is avoided.

The goal for this month is to finish the multiple shapes detection algorithm. For this the algorithm will try each of the models at each position and keep only that one that reduces a dissimilarity measure or cost. The cost will depend on how many model segments are paired with outline segments and what is the total area between them.

Subtask 2: Vehicle Positioning and Obstacle Avoidance

Methods have been added to retrieve mechanism range data (Sensor_Data) from a user-specified sensor. However, the current method that we are using is tedious, and a new, cleaner approach is being implemented. Also, to aid in debugging sensor code, an activated sensor draws a line along its line of sight to the "sensed" object. Kinematics are now incorporated into each Link. The kinematics can be integrated from velocities. The inverse dynamics are calculated and output to a file. An off-line plotting program is used to display joint values (torques, position, velocity). This tool will be modified for real-time use.

We edited and produced a video summarizing MBSim's motivation, objectives, and abilities.

Subtask 3: Vision (Landmark Tracking) Positioning System

During this month two different stepper motor panning systems were built and tested. However, data on accuracy and speed has not been acquired as planned. Work has progressed on a variable intensity Xenon strobe for the purpose of saving energy and exposure control. Design work on a dual head system has been started but not completed. Targets for November are (1) dual head design and construction, (2) completion of variable intensity strobe, and (3) data on accuracy/repeatability of panning device.
Subtask 4: Transputer Control System

Development work has continued on the fast region segmenter algorithm, which should improve the accuracy and performance of the visual navigation module, particularly in environments where sharp boundary lines are not apparent. The effort is concentrated now on rewriting the code as a more portable version, suitable for both the Sun host and the Transputer modules. A videotape segment was produced for the November demonstrations at WSRC. The segment illustrated the performance of the fast line finder on images from Fernald, as well as showing the Denning robot running in our mock-up drum storage environment.

B. Milestones Achieved (Based on those identified in the Task Order):

A video tape was prepared and presented at the integrated demos at WSRC.

C. Problems Encountered

Funding will not be sufficient to continue efforts until the end of the contract period. If funds are not received before March there will be a disruption of the graduate students committed to work on the project.
ERDA MONTHLY STATUS REPORT

University: Georgia Institute of Technology  PI: Wayne J. Book

Project Title: Navigation and Collision Avoidance System for a Mobile Robot

Period of Performance: From 11/1/93 to 11/30/93  WSRC Technical Representative: Clyde Ward

A. Project Accomplishments:

A delegation attended the Integrated Project Demonstrations at the Savannah River Site. The video tape on this work was completed and shown at the demonstrations.

Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning

The proposed solution to drum localization using orthogonal distances produced good results. The algorithm performs drum detection and localization within ± 1/10 feet accuracy and it is independent (in some reasonably range) of the robot velocity. Several experiments were performed using the real and simulated robot to verify this result.

The goal for this month is to extend the algorithm to process multiple shapes simultaneously (this was the goal before the problem with the segment lengths were detected). For each of the segment clusters, a polygon shape will be tested in turn. The one that matches the best will be the candidate that describes the obstacle that produce those readings. After this we can use the algorithm to make the robot perform a simple task such as "go to the third drum on your left".

Subtask 2: Vehicle Positioning and Obstacle Avoidance

A rough ultrasonic sensor model has been added to MBSim. The sensor ping method returns a "hit" when it detects an intersection between the ultrasonic cone and the sensed object's surrounding sphere. If such an intersection is detected, a cone is drawn from the sensor to the "sensed" object. This work also uncovered a new method in Rangefinder to yield more accurate range data.

We are in the process of learning the GL graphics library. This library will replace SPHIGS, our existing source for three-dimensional plane generating routines. SPHIGS has been a good introductory tool, however we've now out-grown its limited capabilities.

We have also created a plot utility to enable joint values to be plotted in real time. And we are currently learning more of the X windows system graphics for plotting and user interfaces.

Subtask 3: Vision (Landmark Tracking) Positioning System

In November, prototype strobe systems and stepper motor drives were completed and partially tested. The strobe has computer controlled variable light output. Current tests indicate that we can get a 100:1 ratio in light output. The variable intensity strobe will be useful in dynamically controlling exposure and in conserving electric energy. We anticipate the need to make measurements at a rate of less than 10 per second.

A stepper motor panning drive was constructed with direct connection to a vision head. A standard commercial motor with 1.8 degree steps is used. The indicated repeatability of head
position, based in vision measurements, is less than 0.1 pixel in standard deviation. This converts to 0.006 degrees of rotation, or 22 seconds of arc. Because two heads will be tied together, back to back, for observations down an aisle, this accuracy is important only in terms of vehicle heading (and is overkill). However, the same drive is likely to be needed on cornering triangulation. At 50 feet, this accuracy corresponds to 0.07 inches, standard deviation.

During December, we hope to complete all tests and final design of the strobe circuitry and to do more definitive tests of vision accuracy.

**Subtask 4: Transputer Control System**

Bill Wester has finished his work on this project. No other work has been performed this month.

**B. Milestones Achieved (Based on those identified in the Task Order):**

No milestones were identified in the task order.

**C. Problems Encountered**

The video tape quality was lower than desired. Expected equipment was not available and a substitution had to be made at the last minute.
A. Project Accomplishments:

Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning

A multiple shape detection algorithm that processes multiple shapes simultaneously was completed. The algorithm was tested using two semi-circles, one that opens to the left and one that opens to the right, to detect drums at either side of the path of the robot. In this way, the algorithm is able to detect automatically which of the shapes should be used for each cluster of segments. An experiment to test detection of drums and boxes mixed together in the environment is underway. A paper “Model-Based Detection of Environmental Objects” was written and submitted to the International Conference of Robots and Systems - IROS '94.

The goals for this and next month is to incorporate the result of the detection algorithm in the schema-based navigation system of a Denning robot and perform experiments. With this incorporation the robot will be able to accept a command like "move through the hallway and stop at the nth drum to your right". These goals will be accomplished in three phases: Phase I consists in incorporating the drum detection algorithm into the perception mechanism of Driver++ (a program used to control the Denning). Phase II consists on incorporating motor schemas associated with the drum of interest so that the robot can move to it. Phase III consists on performing fine-tuning to the platform and elaborating experiments.

Subtask 2: Vehicle Positioning and Obstacle Avoidance

To further study MBSim's practicality, we modeled an additional WSRC vehicle, MoRT. MoRT's base vehicle is modeled as a tricycle. On top of this base is a five degree of freedom Shilling arm. Robert Witherspoon provided the vehicle's dimensions, while its mass properties were approximated. In all, it took 25 hours of parameter determinations and graphics descriptions, with a majority of time from the latter. As a result we're considering ways to make the graphics easier to describe.

Work was done to prepare for the conversion of MBSim to the coming Silicon Graphics computer, such as loading necessary compilers and utilities. A potential user-interface was reviewed with another student. This interface would make MBSim easier to use. Substantial "house cleaning" of MBSim was performed. MBSim now utilizes separate environment and graphics view script files for reduced program overhead, ease of use, and increased flexibility. To help MBSim users develop a simulation, a help manual was begun.

Subtask 3: Vision (Landmark Tracking) Positioning System

Additional work on strobe and accuracy measurements were conducted. Component selection/ vendors were examined. The problem of driving the strobe from lower DC voltage (rather than 120 VAC) was examined and a few alternatives identified. Additional stepper motors and stepper drives ordered.
The primary focus for January will be establishing a straw man design of a final product for SWAMI II and the interface characteristics, both geometric and electronic.

**Subtask 4: Transputer Control System**

No effort was expended on the project during this month

**B. Milestones Achieved (Based on those identified in the Task Order):**

No milestones were identified in the Task Order.

**C. Problems Encountered**

No problems were encountered, but holidays slowed progress.
A. Project Accomplishments:

Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning

Phase I has been completed successfully. In other words, the drum detection algorithm has been incorporated into Driver++ and the robot (both in simulation and real modes) can detect drums in the environment. One detail came up in the simulation mode which is that a bug is causing some of the sensor readings to be invalid. This however does not deteriorate the accuracy in the estimation of the position of the drums.

The goal for this month is to go on with phase II, which consists on incorporating motor schemas associated with the drum of interest so that the robot can move to it. Also, there is the possibility of getting the sensor configuration of the HelpMate so that we can study the performance of the detection algorithm in that platform.

Subtask 2: Vehicle Positioning and Obstacle Avoidance

The new Silicon Graphics workstation arrived 1-5-94. The system is setup and is fully operational. All compilers and libraries have been loaded and are working properly. The MBSim graphics have been converted from SPHIGS to Silicon Graphics GL language. The graphics are completed to an operational point. Further fine tuning and optimization remains. Significant speed gains have resulted as expected.

A paper describing the object oriented approach for programming MBSim was completed and will be forwarded directly to the Technical monitor.

Subtask 3: Vision (Landmark Tracking) Positioning System

The strobe work has been documented in a short report on the ability to vary the energy of the strobe over a 50:1 range (0.052 joule to 2.21 joule) with a TTL pulse from the vision system (1.1 ms to 72 ms). Thus we think all feasibility questions have been addressed. In follow on work we expect to minimize the energy required to run the strobe and to operate from available DC voltage. The current version is run from 120 VAC which will not be available on the vehicle.

A draft of volumetric requirements and possible single unit construction for a SWAMI II system has been completed for discussion purposes with SRP. Most of the electronic items for multiple copies of the strobes and stepper motor drives have been acquired. These are actually purchased for the autonomous helicopter project, but will allow us to test systems readily.
Subtask 4: Transputer Control System

Work has resumed on assessing the usefulness of the Fast Region Segmenter in aisle navigation. At this time the code has not been ported to ANSI C on the Sun, and a significant amount of debugging remains. The goal during the coming month will be to develop a version which is suitable for a qualitative evaluation of the Fast Region Segmenter similar to that already performed for the Fast Line Finder.

B. Milestones Achieved (Based on those identified in the Task Order):

No milestones were identified in the Task Order.

C. Problems Encountered

No Problems were encountered.
A. Project Accomplishments:

Plans for the final report are underway. It will consist of an integrated executive summary followed by sections oriented to subtasks.

Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning

The detail about having some invalid ultrasonic readings when running under the simulation mode was corrected. The work during this month was concentrated on designing and implementing state diagrams to handle plans. A two state plan will direct the robot to position itself in front of a target drum. The first state will direct the robot to move forward along the hallway until it has passed beyond the position of the target drum. Then, the second state will direct the robot to move and position itself in front of the target drum as estimated by the model.

The goal for this month is to continue with this work and finish it by the end of the month. If there is time, we could prepare a demo having the robot move around and position itself in front of a pre-specified drum.

Subtask 2: Vehicle Positioning and Obstacle Avoidance

The MBSim code was refined, eliminating a memory allocation problem that hampered performance. The initial performance gain in converting workstations yielded a reduction in time by a factor of seven for motion only. Further refinements were performed on many of the classes such as the Sensor class. Simulations utilizing Rangefinder and Ultrasonic sensors showed a dramatic reduction in run time on the order of minutes to seconds. The Graphics_View class allows MBSim to run without an open graphics window, enabling the code to be run on a text terminal.

A plot utility which displays a real-time graph of any joint variable of interest vs. time. For example, the wheel torque on SWAMI given velocity commands can be scrutinized. This will enable the user to determine when unrealistic motions are being commanded.

Work on nonholonomic path planning and the resulting trajectory is incorporating a basic motion consisting of straight and circular line segments. The nonholonomic links that we are considering are Knife, Tricycle, and Dual Wheel.

A report detailing how one creates and modifies a new mechanism with MBSim was created. A draft of this user’s guide will be sent to the Technical Monitor before the final version is completed. Rough video tapes of the current operation of the animation were also sent to the monitor to apprise him of the current status of the display speed. An abstract was submitted to the Spectrum ’94 conference in Atlanta was submitted. The conference is a topical meeting on issues relating to the nuclear industry.

The intelligent sensor data storage array, Map, was linked to MBSim. Map updates itself by pinging all user-specified sensors on a mechanism, and storing the returned range data in its array. The array "moves" with the mechanism to minimize memory and calculation time issues. However, Map is currently not robust, and is being improved.
Subtask 3: Vision (Landmark Tracking) Positioning System

An alternative landmark tracking system has been devised and analyzed as requested. Recall the first design was a two IVU system with rotating optical heads mounted back-to-back. The alternative would use four IVU's and rigidly place one optical head at each corner of the vehicle. The two-IVU design has advantages in power conservation, weight, simplicity (integrating two cameras vs. four), cost, and 360 degrees of sight (whereas the four-IVU design would have some blind spots which would require robot rotation for correction). Alternatives for fiducial landmarks are also undergoing further investigation.

The software driving the optical head stepper motor was altered slightly to meet the needs of the robot's environment; optimization of this software will occur during beta testing of the entire system. Standards for software were also devised as requested.

The packaging of the stepper motor, stepper controller, and stepper power supply was incorporated in one sub-enclosure. The strobe unit components were optimized and are currently being packaged in a second sub-enclosure. These will be modual units which simplify repair and replacement of parts. A CAD drawing was made which consists of both units, the two IVUs, and a rotating optical platform, all of which will be placed in the final professional looking main enclosure

Subtask 4: Transputer Control System

Some of the ANSI C portability modifications to the Fast Region Segmenter have been made, but a working version for the Sun has not been fully developed. This continues to be the primary area of effort, since it remains to be seen how well a region-based vision algorithm works with sample aisle images, as compared to the Fast Line Finder.

At the invitation of Fred Sias (in behalf of Randy Singer) we submitted a paper summary for the 1994 Annual Meeting of the American Nuclear Society. This paper, which will discuss general aspects of using the Fast Line Finder for visual navigation in hazardous environments, was accepted.

B. Milestones Achieved (Based on those identified in the Task Order):

No milestones are identified in the Task Order.

C. Problems Encountered

No notable problems were encountered.
Machine Vision Positioning of Mobile Vehicles
Erik Blasch, Stephen Dickerson, Stephen Fuks, Brian Riddle, and Mike Schreiber

Georgia Institute of Technology
George W. Woodruff School of Mechanical Engineering
December 15, 1993

Abstract: Machine vision based position measurement is used to navigate mobile vehicles as demonstrated in three applications: an industrial automated guided vehicle (AGV), an aerial unmanned vehicle (AUV), and a stored waste autonomous mobile investigator (SWAMI). A DVT70C Stinger integrated vision unit (IVU) and software are integrated with optional hardware such as panning platforms and Xenon strobe tubes to provide position feedback for vehicle guidance and control. Machine vision based landmark tracking systems are attractive due to their flexibility, low cost, accurate and repeatable measurements, and update rates up to 40 Hz.

I. Introduction
Position measurement is essential in guiding autonomous vehicles as demonstrated by a research AGV utilizing ceiling mounted fiducials, an aerial unmanned vehicle (AUV), and an industrial AGV for a nuclear waste inspection known as the Stored Waste Autonomous Mobile Investigator (SWAMI). Machine vision, where images are digitized and analyzed by micro-computers, is a promising technology which provides accurate and reliable estimates of position for feedback control of vehicle movement. This paper details three examples of low-cost machine vision systems for use in autonomous vehicle control by presenting equipment characteristics, fiducial-tracking algorithms, and results of position measurement.

2.0 Motivation: Autonomous Vehicles
Autonomous vehicles are appropriate for entrance and observation in areas that are monitored for security or safety. To track and relay the coordinates of the these vehicles, a machine vision system (MVS) can determine relative positions within an operating environment. The vision system may be located either on the vehicle or strategically located close to the vehicles operating environment. MVS outputs can be used to estimate vehicle position and velocity, compensate for disturbance inputs, identify and avoid obstacles, and determine task completion status.

2.1 Autonomous Aerial Vehicles:
Since, a MVS has the capability to sense a minute change with in its field of view, the vision system is excellent for high-risk security surveillance such as aerial observation for the military. During peace-time or war-time operations, the personnel cost to the military is high, and losing a soldier for information gathering far outweighs a lost vehicle.

Another use for autonomous aerial vehicles is the observation, equipment retrieval, and deposit of equipment/resources in hazardous situations. A chemical rail turnover and a contaminated nuclear facility provide two such examples where human exposure is considered unacceptable.

2.2 Automated Guided Vehicles:
A MVS can be used to track and control a factory/warehouse vehicle which would free up an employee for other tasks. Camera units can be placed at ceiling height (inside), at roof level (outside), or on the vehicle itself. In contrast to AGVs that are dependent upon wire-in-floor or surface stripes, these systems provide route flexibility and ease of repair.

Farm equipment provides another potential use
for AGVs. A MVS on the corners of fields in which a vehicle is plowing, planting, fertilizing, or harvesting can relay position coordinates to the navigation controller on the vehicle. By knowing the vehicle status, the farmer and employees can attend to other farm business.

3.0 Integrated Vision Systems

In each of the applications described, the MVS integrates illumination (if used), optics, a CCD array, a photodetector, digitizing electronics, a micro-computer, and communications in a single device. The Integrated Vision Unit (IVU), shown in Figure 1, is a DVT Stinger 70C IVU developed at Georgia Tech. This MVS is now marketed by a spin-off company located in Atlanta, Georgia. The IVU executes vision operations synchronously with the system electronics; for example, the CCD scan is driven by the same oscillator as the microcomputer. The end result is a simple, compact, lightweight, and inexpensive grayscale IVU, that operates on a single external DC power supply.

After locating the fiducial in the image, the IVU communicates the computed fiducial coordinates over an RS232 communication line at 38400 Baud. The serial port also permits instructions to be sent to the vision system for such purposes as downloading images from the IVU to an external computer. A TTL level parallel port could also be used to cue the IVU to take an image or as an output to direct a servo-controlled gimbal.

4.0 Off-Board AGV Navigation

For the purpose of guiding AGVs off-wire, original development work was conducted with an IVU and retro-reflective materials. McKinney and Banta developed an initial system using a binary imaging system; and Lapin recently completed his PhD dissertation work by incorporating a grayscale version of the vision system on-board an AGV.

4.1 AGV System Configuration

A standard DVT 70C Stinger vision system was mounted inside a Litton Series 800 AGV with the optical axis pointing up at the ceiling as shown in Figure 2. A circular array of 40 LEDs was constructed to illuminate retro-reflective fiducial patterns placed on the ceiling of a laboratory. Retro-reflective materials, which are consumed in millions of pounds annually in marking road signs, are inexpensive and provide an apparent brightness of more than a 1000 times that of a diffuse white surface.

As illustrated in Figure 3, the fiducial consists of two circular retro-reflective disks, the larger circle being 2 inches in diameter, the smaller 1.5 inches. The distance between centers of these two circles is 2.5 inches, and the distance between the top of the AGV and the ceiling was...
approximately ten feet. Using a lens with a 8.4mm focal length provided a field of view (FOV) of approximately 3.14 feet square. The IVU and LED configuration determine a relative position and estimate the relative orientation of the vehicle with respect to the fiducials; but, when a fiducial is not present in the Stinger’s FOV, dead reckoning is used to approximate the vehicle position.

\[\text{Figure 3: Ceiling Fiducial.}\]

4.2 AGV System Software

Recognition of the fiducials is accomplished by processing of bright areas in the image (blobs) and verifying the relationships between blobs as shown by the following algorithm:

1. Scan the image, every other row, every other pixel, looking for a pixel above a threshold.

2. Beginning at a bright pixel (above a threshold value), expand the blob with the four-connected rule to include all neighboring pixels. As the expansion takes place, examined pixel values are set to zero(dark) and various properties are calculated. These properties are square pixel area and the row and column center of gravity, where the grayscale value itself is used as the weighting factor for the center of gravity calculation.

3. Using a set of rules (which might be called fuzzy logic today) find a pair of blobs that have the correct properties including relative position. That is the larger area blob must be in a fixed range, the smaller area blob must be in a fixed range, and the centers of gravity must be separated by a distance in a fixed range. These calculations are straight forward since the distance from the IVU to the fiducial is known and fixed, and the fiducial has a precisely controlled geometry.

4. After verifying that a fiducial is present, the two blobs centers of gravity and areas are processed to give a composite center of gravity and orientation angle for the fiducial.

4.3 AGV Results

The IVU computes the position estimate and the averaged final result is passed to the AGV’s central processor. The total time to locate a fiducial after commanded is less than 0.1 second. The repeatability was better than 2.5 mm (3 sigma), given a stationary vehicle taking repeated measurements of the same fiducial. The identification of the fiducial pattern is required because the 40 watt fluorescent lights on the ceiling are approximately the same brightness as the retro-reflective fiducials.

When the vehicle was in motion, Lapin estimates that the accuracy of position location was bounded by 15 mm in both the X and Y directions. Thus, the vision system itself is not a significant source of error. However, the motion of the vehicle, which causes the optical axis to vary in orientation, can produce a relatively large error on a single observation.

5.0 Autonomous Unmanned Aerial Vehicle Control

5.1 AUV Contest

An application which integrates a MVS for guidance and control is the Georgia Tech Aerial Robotics team (GTARS) entry in the Association for Unmanned Vehicle Systems’ (AUVS) International Aerial Robotics Competition. The contest requires students to design and build a completely autonomous flying vehicle capable of finding and transporting a number of small target disks within a tennis-court size arena. Figure 4 shows an artists sketch of the competition.

\[\text{Figure 4: Unmanned Aerial Vehicle Contest.}\]
The GTARS Team chose a commercial radio-controlled helicopter for their platform because of the availability of spare parts.

5.2 AUV Navigational System Constraints
For the helicopter to have autonomous control, a \([X,Y]\) position is necessary to close the position feedback loop. The position navigational system (NavSys), which estimates target position, is subject to these constraints:

1. A 10 Hz update rate to adequately provide information for system response such as avoiding obstacles,
2. 20 to 40 meter visible range specified by the contest,
3. +/-3 m accuracy to establish the position of the helicopter over the 2 m bins,
4. Robust transfer of data,
5. $8000 price tag including donated parts, which is picked up by the student-run project, and
6. < 4 pounds of navigational equipment placed onboard the aircraft.

Given these constraints on the system, machine vision was selected because it offers significant flexibility in both hardware and software and is relatively easy to implement. Using a C++ based program and passive retro-reflective targets, the flexibility of the system rests in the ability to alter these targets at will. Additionally, using a passive target as a fiducial saves weight onboard the helicopter. The only drawback of the IVU/fiducial-tracking approach is the restricted line-of-sight. To overcome this potential problem, the cameras were mounted on a panning platform. As the fiducial nears the edge of the image, the servo advances the CCD head to a new position so that the target is located close to the center of the screen.

Although only two land-based cameras are needed to triangulate the target's position, four cameras are used for redundant recordings and fault tolerance. The fiducial size is limited to one 600mm by 460mm target onboard the helicopter for avoidance of non-collated multiple target recognition and non-obstructed vehicle performance. Since, successful tracking demands a continuous line-of-sight between the cameras and the fiducial, the helicopter should fly at a constant heading with minimum pitch and roll.

5.3 AUV NavSys Configuration
The vision-based tracking system utilizing four grayscale vision systems, each with a one-degree of freedom panning system that expended the 12° FOV of a CCD head into an effective FOV of better than 150°. The 12 mm focal length of the lens allows for a dynamic depth of field of 1.5 to 40 m which results in a minimum span of 2 pixels per feature at a distance of 40 m.

The panning system consists of a CCD head that is integrated with a Motorola 68HC11 microprocessor and a Futaba hobby motor, typically used to servo aerodynamic surfaces on a model airplanes. The commands for the gimbal rotation are sent to the HC11 via the parallel port of the camera. The HC11 converts the digital signal to an output pulse-width modulated signal for the Futaba 9201 servo motor. The servo motor interprets the signal and positions its motor output shaft accordingly.

A photograph of the entry vehicle, Figure 5, shows the retro-reflective "bar-code" that was used for tracking the vehicle. The fiducial was a set of three vertical white bars on a black background. Each bar was 75 mm wide and made of standard traffic control grade of retro-reflective material (donated by 3M Corp.).

A 300 watt incandescent bulb was mounted just above each camera head to enhance the image contrast. At night, this arrangement effectively makes the bar codes the brightest object in the image and superior tracking results are observed. In daylight, the effects of ambient sunlight make it difficult for the IVU to successfully distinguish the intensity pattern of the bar code from other objects in the image.

Figure 5: AUV Entry Vehicle.
To enhance the image quality, light levels can be altered by changing the amount of time the CCD pixel elements are exposed to the scene. The combination of a 300 watt bulb and ambient sunlight resulted in exposure times of about 20 milliseconds per image. This exposure time reduces the amount of observed streaks in the image and enhances the grayscale output. Outdoor operation of the camera produces varied results since the lighting is highly unstructured and unpredictable. To compensate for the ambient sunlight alterations to the CCD image, one IR-blocking filter was used that eliminates IR light from reaching the CCD. Another IR transmitting filter was used that allows only IR light to pass and by changing the orientation of these two filters/polarizers in series, the amount of IR light passed to the CCD can be adjusted to enhance the contrast of the image. See Figure 6 for an illustration. In addition, a computer-controlled variable broadband liquid-crystal light blocker was used to reduce the total amount of light input to the CCD during periods of bright sunlight. These blockers were specifically manufactured by the Kent State Center for Liquid Crystal Research and could be driven by a single modulated signal from the HC11.

Figure 6: Mechanical Variable Polarizer.

5.4 AUV NavSys Software
The NavSys software initializes the cameras and then automatically sets the cameras in a continuous loop scanning for the helicopter fiducial. The software for each of the four vision systems is presented in the flow graph of Figure 7. The NavSys routine scans row by row looking for a particular light(reflective) to dark(absorbing) transition that characterizes the fiducial. The symmetric stripes of the bar-code target produce a distance-normalized pattern. The method the NavSys uses to resolve these features is a gradient threshold edge detection scheme which searches for 6 transitions of light to dark and dark to light between the 3 equally spaced retro-reflective bars on a black background. If all rows in an image have been scanned and the intensity pattern of the fiducial is not recognized, the servo advances the CCD head to a new field of view. Once the target is located, subpixelization is performed to improve the accuracy of the target position. Unfortunately, the benefits of subpixelization were minor because the effects of servo backlash and image blurring were much larger than one pixel.

Figure 7: IVU Flow Chart for AUV Tracking.

The software for the central navigation computer, NavCom, is displayed in Figure 8. The NavCom computer is a standard 486 machine which communicates with the Stingers over 38400 Baud serial RS-232 lines. In the standard running mode, the Stingers continuously reported to the NavCom computer the azimuth angle of the helicopter at the rate of about 36 Hz. The NavCom reports continuously, at about 10 Hz, the best position estimate of the helicopter to the mission control computer.

Figure 8: NavCom Software.
Initial calibration of the software consists of holding a fiducial bar-code in a known location and computing an azimuth reading. Given these measurements and the known Stinger locations, IVU angle calibration could be achieved. During operation, the information from three of the four cameras is used to triangulate the position of the helicopter. The velocity is computed as the average of the difference of position over time for values stored in a circular buffer.

5.5 AUV MVS Results
Because of the size of the playing field, it is common for one or two of the vision systems to "lose" the helicopter as the distance or sight angle becomes unfavorable. However, because the NavCom computer is continuously estimating position at all times, it can issue commands to any "lost" system to help it re-locate the helicopter.

To offset the problem of erroneous signal transmissions, a header and checksum are sent along with the position estimates. Other sources of error included lighting, servo backlash, and pixel resolution which hindered the position accuracy of the system.

It was determined that at minimum, 3 white stripes on a black background are needed to faithfully separate the fiducial from the background environment. The bar code fiducial is recognizable with a maximum tilt angle from the horizontal of 15 degrees. As a surprise to the experimenters, the speed of target motion did not seem to affect the overall accuracy of the triangulation measurements which is +/-500mm.$^5$

At the third AUVS competition held in 1993, the GTAR team was awarded first place for demonstrating a completely autonomous flight including an autonomous takeoff and landing. However, their entry was unable to demonstrate a complete mission with the pick up and displacement of a disc. During the contest, the team demonstrated the capability of autonomous hover and then separately demonstrated the displacement of a disc with a remotely piloted vehicle. The competition GTAR entry is shown in Figure 9.

Figure 9: Georgia Tech's Entry into the '93 Aerial Unmanned Vehicle Contest.

6.0 Description of the SWAMI

An AGV being developed by engineers at the Savannah River Plant, (SWAMI), is designed to inspect different size hazardous-waste containers as well as the floor and area surrounding the containers. SWAMI is based on a production version of a Transition Research Corporation (TRC) autonomous AGV generally intended for navigation in office buildings, homes, and hospitals. One of these vehicles is being modified to meet the special requirements of monitoring low-level radioactive wastes in DOE warehouses. These warehouses characteristically contain large numbers of steel barrels, arranged neatly in rows, several barrels deep and several barrels high as shown in Figure 10.

Figure 10: SWAMI Inspection Task.

6.1 SWAMI Configuration
Equipped with sensors, the vehicle is able to properly position itself, detect defective barrels, and record its position. SWAMI utilizes low-level vision systems for inspection and positioning ultrasonic sensors for obstacle avoidance, and Beta collectors for radioactive measurements.
The vision group at Georgia Tech has the task of implementing the low-level, fiducial-tracking, machine vision system to precisely measure the vehicle's distance from the path's center line to within 5 mm. Also, the vision system is to provide a longitudinal measure of the vehicle distance along the pathway to within 60 mm.

The proposed machine-vision based vehicle position-monitoring system consists of two stand-alone microprocessor vision units with a pair of CCD sensor units mounted back-to-back and gimbaled, as shown in Figure 11. These sensors locate landmarks at each end of an aisle for position information. By mounting their optical axis 180 degrees apart only a single stepper motor is used to drive the two optical heads as a unit and the stepper motor drive angle is used to calculate the vehicle heading and triangulations during turning maneuvers.

Also mounted in line and close to the CCD heads are Xenon strobe tubes. The intensity of the strobes are computer controllable to provide superior image contrast. Fiducial patterns of retro-reflective materials will be positioned at each end of aisles which can be as long 180 feet.

The IVU computes the relative position of SWAMI with respect to the landmark coordinates. To satisfy position accuracy requirements, a subpixelization technique is also incorporated into the positioning algorithm. All image processing as well as fuzzy control is done internal to the vision system, and the position coordinates are periodically downloaded to the SWAMI's main CPU.

Figure 11: Concept of SWAMI Vision System.

6.1 Results from SWAMI Activity
The vision-based vehicle locating system has not yet been placed on the vehicle; however, all hardware has been tested statically. The stepper motor and driver provide panning capability, and a variable intensity Xenon strobe provides light levels needed to find the landmarks at distances of up to 60m. Test results demonstrate that the stepper motor repeatability is better than one tenth of pixel. Subpixelization incorporated in the software algorithm and the stepper motor have resulted 8.0 mm or .017° 3-sigma accuracy.

7.0 System Comparisons
The measurements and characteristics for each mobile vehicle MVS application are presented in Table 1. Resolution is given in terms of 3 standard deviations. In the case of SWAMI, the ability of the stepper motor to return to a reference position is included. By using a stepper motor with improved repeatability, the machine vision accuracy is improved by about two orders of magnitude relative to the AUV case with DC servomotors. Additionally, by use of a Xenon strobe tube, the indoor depth of field is increased by at least 30 m.

Accuracy, expressed in terms of 3 standard deviations is an absolute error measure. These measurements do not include any avoidable errors in calibration or those which result from uncontrolled vehicle motion. For example, the AGV was tested in a room with tiled concrete floors. On an uneven surface, position estimations from observation of ceiling landmarks would be much poorer as cited by Lapin. In the case of the AUV, constrained vehicle dynamics and varying illumination levels are assumed.

7.1 Future Ambitions
Fiducials do not need to be retro-reflective and experiments are underway to evaluate the use of Xenon strobes as fiducials. As demonstrated in everyday 35 mm cameras, these strobes are inexpensive and low in power consumption, and are effective at long ranges even in the presence of ambient sunlight.

Vehicle position estimates depends not only on an adequate vision system but also on the ability to interpret the data. There is the possibility of using a Kalman filter to smooth a number of
observations into a composite position estimation which utilizes information about the relative motion between images. 3.8

By using a ground-based vision tracking system, the weight of the navigational equipment does not hinder the lifting capabilities of the helicopter; however, with a larger helicopter a MVS can be mounted on-board the helicopter in place of or in addition to off-board cameras to improve the operating range of the aerial vehicle.

8.0 Conclusions

Machine vision is an excellent and practical way of measuring the position of mobile vehicles, particularly in those cases where fiducials can be used for reference. By utilizing inexpensive retro-reflective materials in a unique fiducial pattern, an IVU, panning platforms, and appropriate software, position estimation for vehicle guidance and control is demonstrated.

9.0 Acknowledgments

The authors of this paper would like to thank the following people for their support: Dickerson Vision Technologies (DVT); the Georgia Tech Aerial Robotics Team (GTARS) and NSF SUCEED program; the Georgia Tech Material Handling Research Center; the Georgia Tech Manufacturing Research Center (MaRC); and the DOE’s Education, Research and Development Association (ERDA) of Georgia Universities and Westinghouse Savannah River Company.

10. References

5. Association for Unmanned Vehicle Systems’ (AUUVS) International Aerial Robotics Competition Contest Form.
Integrated Modeling, Simulation, and Animation
of
Rigid Arms and Vehicles
through
Object-Oriented Programming

Paul R. Schmitt John E. Hogan Jonathan M. Cameron Wayne J. Book
Georgia Institute of Technology
The George W. Woodruff School
of Mechanical Engineering
Atlanta, GA 30332
USA

Abstract. This paper describes an object-oriented system, "MBSIM" (MultiBody SIMulator), that models, simulates, and animates the kinematics and dynamics of robotic arms and vehicles. This system creates a three-dimensional graphical environment which can be used as a tool in robotic design and control.

1. INTRODUCTION

A motivation for developing MBSIM arises from an inspection task in the nuclear industry. This task requires vehicles to move safely along narrow aisles to inspect drums of low-level radioactive waste. Safe motion requires sensor-based motion control to avoid collisions with walls and obstacles. Given the difficult nature of the environment involved, a simulation system is necessary for the development and testing of motion planning and control algorithms.

We desired MBSIM to be a convenient and flexible tool for studying the motions of various mechanisms including vehicles, robotic arms, and combinations of the two. Our simulation integrates the kinematics, dynamics, sensors, and graphics of mechanisms into a modular environment. Each mechanism consists of a series of links. The mechanism's kinematics and dynamics are determined from the relative kinematics and mass properties encapsulated in each link. Sensors can be attached onto each link object as desired. These sensors are useful for path generation and control routines. Mechanism graphics are constructed from graphics descriptions associated with each link object.

The types of systems that MBSIM can model involve physical objects such as robots, vehicles, and obstacles, objects that share many characteristics. An object-oriented approach is appropriate for modeling these systems. Common characteristics such as position and geometric shape can be developed once and reused appropriately via object inheritance and inclusion. The resulting software objects reflect the modularity of the world and can be dealt with intuitively. In fact, one of the important benefits of the object-oriented approach is the ability to think of software objects in the same way as we do real objects. MBSIM utilizes object-oriented inheritance to construct new types of links from existing links. To illustrate some benefits of the object-oriented approach, this paper includes a case study that outlines the steps taken as well as the time involved in modeling and simulating a mobile platform.

2. SOFTWARE DESIGN

Our system is implemented using the object-oriented C++ language [4]. C++ is an object-oriented programming language that directly supports inclusion (using user-created objects as data) and inheritance.
(designing objects which inherit previously developed data structures and code from more basic object definitions).

In MBSIM, each mechanism is constructed as a series of links. Each link incorporates the joint kinematics that connect it to the preceding link. In other words, each link "knows" how to move itself with respect to the previous link. This general joint paradigm allows holonomic and non-holonomic constraints to be treated similarly. This paradigm also facilitates modeling and simulation of the kinematics and dynamics of multibody mechanisms with a wide variety of link types. The links contain graphic objects which describe the physical shape of the link and sensors which provide feedback of the environment.

3. KINEMATICS

Following the joint paradigm, each link contains functions to compute its angular and linear velocity and acceleration relative to the preceding link. These functions are pure virtual functions in the generic link class and therefore must be defined in every specific link sub-class. This ensures a uniform interface to each specific link that can be used to calculate velocities and accelerations of any link. The velocities or accelerations are calculated by propagating the velocity out from the base link to the link of interest. This propagation from link to link requires the uniform interface so that the velocities or accelerations can be calculated for mechanisms composed from any number and types of links.

Each specific link also contains functions which enable velocity or acceleration joint variables to be integrated to yield the mechanism motion. These velocities or accelerations would be commanded from control schemes such as obstacle avoidance and path planning. These control schemes would generally be implemented in the simulation to make use of sensor feedback and dynamic characteristics of the system. The commanded velocity or acceleration may be output to an external file for further analysis or for future replays. If needed, this also enables data from external control schemes to be read and tested with the simulation. These integration functions in each link, one for velocities and one for accelerations, contain equations which relate joint velocity variables to joint position variables and joint acceleration variables to joint velocity variables, respectively. Inside these functions, constraint or auxiliary equations may be used to determine the specific behavior of the link, as in a non-holonomic link. The mechanism calls the integration function of each link in the same manner, whether the link is holonomic or non-holonomic.

4. INVERSE DYNAMICS

The inverse dynamics of the system is calculated using an iterative Newton-Euler dynamic formulation similar to the one found in [2]. The inverse dynamics yields the forces and torques that each link's actuator would need to generate motion based on each link's positions, velocities, and accelerations. This is useful to determine whether the desired velocity or acceleration commands are feasible for a particular actuator. Many control schemes such as those found on a Denning robot produce velocity or acceleration commands rather than forces or torques. The consistent velocity and acceleration interface contained within each link enables only one function at the mechanism level to calculate the dynamics for each link in any mechanism in any configuration. Once again, by embedding the specific characteristics of a link inside the specific link class and using consistent interfaces to each link, general functions can be used to yield desired quantities regardless of the mechanism being studied.

5. SENSOR MODELING

To model a mechanism in an unknown or partially known environment, we require sensor feedback of the surroundings. Our system currently incorporates an idealized rangefinder model. As previously mentioned the link sensor file holds the positions and rotations of each sensor on the link. This file is flexible, allowing the user to place an arbitrary number of sensors in any position or orientation on any link.

When activated, the sensor currently scans through all objects in the environment to determine whether the objects' enclosing spheres lie on the sensor's line of sight. Once this rough reading has been determined, the
object's actual geometric characteristics are tested for intersection with the sensor's line of sight. This two-stage method reduces computational overhead.

6. IMPLEMENTATION

The MBSIM hierarchy, as developed above, is implemented into four main parent classes: Mechanism, Link, Sensor, and Graphics_Object. The partial class structure in Fig. 1 shows their inheritance and inclusion relationships. These classes are constructed from script files external to the program. These files are read at run-time so additions and modifications of a wide variety of mechanisms can be made without having to re-compile MBSIM.

The Graphics_Object class encapsulates all graphics library function calls. The system initially utilized SPHIGS, a public domain software library for 3-dimensional graphical display [3]. (SPHIGS is a simplified implementation of the well known 3-dimensional graphics protocol, PHIGS.) This encapsulation facilitates improvements to the graphics library.

7. CASE STUDY

A nuclear industry testbed vehicle is driven by two parallel wheels on a fixed axis with independent velocity control. The vehicle, as shown in Fig. 2, is equipped with twenty-four ultrasonic sensors.

The following steps were taken to simulate this vehicle. The manual derivation of the vehicle's kinematic and dynamic link equations required one hour. These equations were incorporated into a new class derived from the parent Link class, requiring a second hour. The vehicle took shape as its dimensions were set into a graphics description file requiring a half hour. Next, sensors were added to the vehicle through a sensor description file. Positioning and orienting the twenty-four sensors took an hour. Another hour was needed to develop a routine to test the accuracy of the above files and code. The total time required to develop this new link and tailor it to a specific vehicle was about four and one half hours.

8. CURRENT WORK

Current work is focused on extending the flexibility of MBSIM. These areas include forward dynamics, sensor modeling, sensor data mapping, and graphics library improvements.

Forward dynamics can be determined by putting special values into the iterative Newton-Euler routine to determine the mass matrix and the non-linear Coriolis, centrifugal, and gravity terms. Once found, one can then
use these matrices to solve for joint accelerations given joint torques. Velocity and position are then obtained by integrating acceleration.

An ultrasonic sensor is being modeled. Where appropriate, noise will be added to the returned data to more accurately simulate real sensor data.

An intelligent sensor data storage array map is being developed to handle sensor input. Histogramic in-motion mapping will be used mainly to record levels of obstacle existence evidence [1]. The map updates data from new readings and will be used to create and evaluate obstacle avoidance paths.

Currently, the graphics routines limit the performance of our system. This is due to our hardware and the simple nature of the SPHIGS graphics library. While excellent for our initial learning phase, SPHIGS does not have the performance of commercial graphics libraries. We are upgrading our hardware and implementing a more advanced and efficient graphics library.

9. CONCLUSION

MBSIM, an object-oriented three-dimensional robotic simulator, has been introduced. This simulation integrates the kinematics, dynamics, sensors, and graphics of mechanisms into a modular environment. A case study that outlines the steps taken and time involved in modeling and simulating a particular mobile was presented to demonstrate the system's flexibility. Current work will enhance the system's performance, flexibility, and sensing the environment.

10. ACKNOWLEDGMENTS

This work was partially supported by subcontract 92006 to the Educational Research and Development Association (ERDA) of Georgia Universities under contract with Westinghouse Savannah River Plant.

11. REFERENCES


Routine monitoring of stored radioactive materials can be performed safely by mobile robots. In order to accomplish this in an environment consisting of aisles of drums, it is necessary to have a reliable means of aisle-following. This work describes the adaptation of successful road-following methods based on visual recognition of road boundaries to the waste storage problem. Since the effort is targeted for near-term usage in normal operating conditions, special emphasis has been given to the implementation of the visual processing on practical (i.e., small, low-cost, and low-power) hardware platforms. The system described here has been implemented onboard a robot in our laboratory.

The environment for the mobile robot application can be generally described as rows of closely spaced pallets, each containing stacks of drums. The rows are separated by aisles that are large enough for the robot to pass through (about 1 m). It is required that the robot be able to travel down these aisles without colliding with any drums or pallets. The drums vary in size, and they may extend past the edge of the pallets. The resulting visual boundary (at the floor) is a series of line segments and curves, sometimes interrupted by gaps. In order to provide steering cues, the visual algorithm must be able to identify the approximate left and right boundaries, in spite of their deviation from straight lines, and project an imaginary center line. The basis for this work is the Fast Line Finder (FLF) developed at the University of Massachusetts.\(^1\) This algorithm has been used successfully on minicomputers and workstations by its authors and others, including some previous work at Georgia Tech, but for this application it was necessary to port it to a system more suitable for actual deployment on a robot.

A modular, flexible architecture called ANIMA (Architecture for Natural Intelligence in Machine Applications) has been developed at Georgia Tech.\(^2\) The initial versions of this architecture have been based on the Inmos Transputer, a microprocessor designed for parallel applications. ANIMA has been tested and demonstrated in public exhibitions along with robots from other institutions.\(^3\) To address this application, an ANIMA-based real-time visual navigation module has been developed. Using commercial off-the-shelf components, the module consists of a frame-
grabber, a dedicated FLF processor, and an optional graphics display controller. The FLF processor is a 30 MHz T805 Transputer, which includes a floating-point processor capable of over 2.5 MFLOPs, but requiring few external components and minimal power. The processor and its four Megabytes of dynamic memory are packaged in the standard Transputer Module, or TRAM, format. Both the frame-grabber and the display controller also utilize T805 processors and are packaged as TRAMs as well.

The FLF algorithm was written in C, but it did not comply with ANSI C requirements and was not directly portable to the Transputer C compiler. To address this, it was first converted to ANSI C and run on a Sun workstation before attempting compilation on the Transputer system. This virtually eliminated portability problems, since only two bugs were found when the code was eventually ported. Both of these were somewhat obscure order-of-evaluation bugs. Using stored images and the workstation version of the FLF, it was possible to determine suitable parameters.

The onboard computer achieved only about 75 percent of the processing speed of the Sun workstation, which was still more than adequate for this application. In order to perform thorough radiation detection, the robot's speed was relatively low, which constrains image processing speed more than the onboard hardware. Since Transputers were used, the architecture can easily be expanded to achieve higher frame rates if required. An example of an obvious parallelization would be to use a different processor for the left and right boundary extractions. Testing of the FLF algorithm with images of stored hazardous waste showed that this implementation is robust and well suited for autonomous navigation.

References


(Extended abstract – to be presented at American Nuclear Society 1994 Annual Meeting)
Integration of Reactive Navigation with a Flexible Parallel Hardware Architecture

Thomas R. Collins, Ronald C. Arkin, Andrew M. Henshaw

School of Electrical Engineering and College of Computing
Georgia Institute of Technology
Atlanta, GA 30332

Abstract

To demonstrate the flexibility and portability of both a schema-based software architecture and a message-passing hardware architecture, the two were integrated within a very short period to be used in a mobile robot competition. The experience confirmed the advantages of onboard computational capability in mobile systems.

1 Introduction

Autonomous machines with sensory, manipulative, and locomotive capabilities are a significant class of intelligent systems holding great promise for performing hazardous or mundane tasks. Although much work has been performed with isolated aspects of intelligent machines, including vision, sonar, manipulator control, and knowledge-based reasoning, the algorithms are often not considered within the context of a complete machine. In many prior efforts, both software architectures and hardware architectures have been developed to meet the requirements of specific projects, with little regard to reusability in other applications. Often, experimental systems are not robust, failing due to relatively minor environmental variations or task redefinitions. This paper describes the integration of two separate efforts to address these problems. One is a reactive software architecture which has been demonstrated to perform a variety of tasks well, and the other is a targeted, yet flexible, computer architecture that provides modularity and expandability.

2 Background

Most of the earliest work with intelligent machines relied on direct programming in declarative languages [22, 23, 24]. This resulted in software architectures that could only accommodate certain situations, and even then only in a "do this, then do this" fashion. There has been a gradual trend toward reactive software architectures, which combine relatively simple behaviors to implement complex tasks. Since these simple components are designed to respond to general stimuli in "common-sense" fashion, greater robustness is achieved, even when a higher level of deliberative behavior is added.

From a hardware architectural standpoint, many early autonomous machines relied on offboard control, since the necessary computers were so large. Then, as microprocessors became widely available to provide onboard intelligence in relatively small packages, many more projects began. Even today, however, many mobile robots depend on detached workstations, because of the increasing computational demands and the need for sophisticated user interfaces. It has become increasingly evident that autonomous machines must incorporate multiprocessor architectures, but it is not quite clear what form the architectures should take.

The majority of these specialized multiprocessor architectures have taken a hierarchical form, often a simple tree [14, 22, 24]. In the context of an intelligent machine, the processing tasks closest to the environment (i.e., the "low-level" tasks) tend to map most obviously into a tree. Higher levels of thought prefer to describe both perceptions and actions in concise symbolic terms, while environmental interactions tend to involve large amounts of data, often to or from loosely-coupled subsystems. It is thus advantageous to have multiple perceptual processes and multiple control processes running simultaneously, each having extensive interaction with the environment, but little interaction with each other. Communication with higher levels is less frequent, and the messages tend to contain condensed information in a symbolic format. Intermediate levels of processing act to integrate data (on the sensory side) or to coordinate simple tasks (on the control side). In this simple paradigm, only one highest-level, or reasoning, task is required. Unfortunately, this process usually forms a computational bottleneck, since it is an integral part of every sensor/control path.

More recently, hierarchical designs have emphasized connection across the hierarchy, as in the NASA/NBS standard reference model for telerobot architectures.
(NASREM) developed by Albus [1] and the Multiresolutional Control Architecture of Meystel [21]. These architectures allow nested control loops via distinct task levels within the hierarchical structure, providing higher bandwidth for low-level tasks that require shorter response times. Each higher level in the hierarchy thus provides for increasing levels of abstraction in perception, reasoning, and control, and each maintains a model of the world appropriate for its purposes. No clear consensus exists, however, for the number and type of levels to use in such architectures.

The hierarchical viewpoint is increasingly being challenged by reactive approaches. The concept of schemas, as typified by Lyons [20] and Arkin [2, 3], implements sensor-effector connections in a more flexible fashion. A schema is a pattern of behavior exhibiting a stimulus-response characteristic. Typically, each schema monitors only a portion of the available input data and produces an output which may have to be combined, superimposed, or otherwise reconciled with other outputs. Schemas may communicate with other schemas or with sensors and effectors. Normally, an intelligent machine would create instances of predefined schemas as necessary to produce more complex behaviors. Such apparent complexity arises both from the ability of schemas to use other schemas and from the parallel actions of independent schemas.

Another significant reactive approach is the subsumption architecture developed by Brooks [9, 10, 11]. Multiple levels of competence are defined, connecting input and output in a layered system. Higher levels of competence inhibit or subsume all lower levels, and the hardware usually provides direct support for this subsumption characteristic by implementing each level within its own processing subsystem. This structure allows a machine to be developed in stages, building each level on top of a machine that already functions at some given degree of competence. Another advantage is that the lower levels still exhibit useful behaviors that are activated in the absence of any inhibition from above. For example, a low-level obstacle avoidance behavior is still useful even when a path planner exists to provide intermittent goals.

In spite of the emergence of these promising approaches, there is still a considerable amount of disarray in the overall architectural scene. The key limitations of much of the previous work, including both hardware and software considerations, have been:

- reliance on offboard computational facilities and radio communication,
- ad hoc, inflexible hardware architectures,
- lack of inherent support for parallelism,
- awkward development environments, and
- lack of portability.

These limitations hamper the development of reusable, modular hardware and software components, and they have thus slowed the development of a significant commercial market in sophisticated autonomous machines.

3 AuRA – the Autonomous Robot Architecture

AuRA is a hybrid architecture encompassing aspects of both deliberative and reactive control. It consists of 5 major subsystems:

- Perception – charged with collecting and filtering all preliminary sensory data.
- Cartographic – concerned with maintaining long-term memory (a priori models of the environment), short-term memory (dynamically acquired world knowledge), and models of spatial uncertainty.
- Planning – consists of a hierarchical planner (the deliberative component) and the motor schema manager (the reactive component).
- Motor – the interface software to the specific robot to be controlled.
- Homeostatic – a component concerned with dynamic replanning in light of available internal resources [7].

The overall architecture has been described in detail elsewhere. The reader is referred to [4, 6] for more information.

The hardware migration to ANIMA thus far has been concerned with the reactive and perceptual components of the system which run within the confines of the motor schema manager. Figure 1 presents the logical relationships between the varying schemas which constitute this portion of AuRA.

Action-oriented perception forms the underlying philosophy for channeling sensory information to the motor schemas (behaviors) [5]. Only the information that is essential to a particular motor behavior is transmitted to it, essentially on a need-to-know basis. The message-passing

![Figure 1. Inter-schema relationships.](image-url)
paradigm found in ANIMA is well-suited for this type of information flow.

Each of the active motor schemas generates a velocity vector in a manner analogous to the potential fields method [18, 19] with the individual results being summed, normalized and transmitted to the robot for execution. The speed of operation is highly dependent on the rate of processing of incoming sensory data. The parallelism found in the transputer implementation described below is a natural match for this aspect of the AuRA architecture.

4 ANIMA hardware architecture

We have developed a flexible, real-time platform for the development of AuRA and other software architectures. The skeleton of our hardware architecture, ANIMA (Architecture for Natural Intelligence in Machine Applications), has been developed from basic principles. It incorporates a triad of basic systems, just as a conventional computing system includes input, output, and processing subsystems. This fundamental triad of subsystems carries over into the architecture of an intelligent machine, but a more general interconnection pattern is required. The addition of a communication channel between the input subsystem and the output subsystem allows the machine to exhibit reflexive behaviors. Such behaviors are analogous to reflexes in biological systems, where the communication channel is implemented by structures within the spinal cord and lower brain. While it would be possible to develop an autonomous machine without such a channel, it would not take advantage of the localized intelligence within the input and output subsystems. The resulting increase in computational load on the processing subsystem would result in slower response time.

Clearly, reflexive behaviors are virtually "hard-wired" into the system, and their implementation is best reserved for behaviors that:

• must be performed reliably and quickly, usually to avoid danger to the machine or to humans,
• require little or no integration of information from multiple input systems, and
• although primitive, usually produce an effect more desirable than if no action at all had been taken.

The deliberative component controlling the input and output subsystems is called the Reasoner. A major aspect of this reasoning capability is the need to maintain some sort of world model based on sensory input, at least for anything more than basic reactive behavior.

The vast majority of intelligent machine research has assumed that the input/output devices, just as in a conventional computer, are largely independent in their low-level operation (at or below the level of the device driver). For input devices, the combination of these independent streams of data has often been referred to as sensor integration, and we include a process, called the Integrator, to perform this task. On the output side of the structure, the most appropriate term is coordination, although the specific definition varies considerably in the literature. Corresponding to the Integrator, we include a process called the Coordinator.

These additional parallel processes are illustrated in Figure 2. The independent sensor subsystems are called logical sensors, in much the same sense as those of Henderson [15] or Crowley [13]. At this point, a logical sensor is best thought of as a combination of a physical sensor, capable of estimating some property of the environment or the machine itself, and a generalized device driver. The extension of this concept to the logical effector is straightforward. Taken together, logical sensors and logical effectors are called logical devices. A single logical device can be composed of multiple physical devices, with appropriate drivers. This would be desirable in cases where the physical devices were virtually identical (except perhaps in physical location, scaling, or some other trivial factor), allowing the main driver to give the appearance of a single effective logical device.

These processes (Reasoner, Integrator, Coordinator, and representative logical devices) have been described using the notation of Communicating Sequential Processes (CSP) [16, 17]. By combining them into a system, it is possible to construct a proof showing that the system is free of deadlock [12]. By following basic design principles at the logical device layer and the Integrator/Coordinator layer, we provide a means of fault isolation to individual logical devices.

An implementation of this architecture is being developed based on the Inmos T800, a member of the transputer family of microprocessors developed for parallel processing. Each transputer provides four high-speed serial links for the

![Figure 2. ANIMA structure.](image-url)
required processor interconnection. While several languages are supported, Occam is the most effective for parallel processing, providing constructs that implement the important CSP operators.

The ANIMA architecture requires no parallel data busses or backplanes of any kind. Instead, it consists of modular components connected only by high-speed serial links. This allows the processors to be distributed to any convenient locations within the intelligent machine.

Relatively early in the architectural development, the entire structure was simulated on a single transputer and on multiple transputers in order to verify its operation. A fundamental premise of the simulation was that most of the processes would be directly portable to a real machine. Specifically, by making reasonable models of the environment, sensors, and effectors, one can use essentially the same Integrator, Coordinator, and Reasoner processes as would be used on a real machine [12]. The simulated machine wandered through a simple world with walls and obstacles, using simulated sonar and touch sensors. Sensors and effectors were deliberately modeled as being imperfect, and the machine (as part of its Reasoner) had to maintain its own model of the simulated world. The initial Reasoner was essentially a schema-based implementation, but hierarchical and subsumption versions have also been tested.

5 Case study – “Buzz”

AuRA and ANIMA were first brought together in the development of a machine called “Buzz,” to compete in the first robot exhibition and competition sponsored by the American Association for Artificial Intelligence (AAAI). The competition stressed the ability of mobile robots to explore an arena, avoid static or moving obstacles, locate goals, and visit goals in specified order. Many of the basic reactive behaviors that were needed had already been developed (using AuRA) within the Mobile Robotics Laboratory of the College of Computing at Georgia Tech. The previous work had been done on an older robot, and a new one was made available for the competition by Denning Mobile Robotics (Wilmington, Massachusetts). Much of the required programming effort would be to modify communication routines for the new robot, to develop new perceptual schemas for previously-uninteresting phenomena, and to combine the available schemas in ways appropriate for the competition tasks.

Realizing that the use of radio communication was both a reliability issue and a “showmanship” limitation, the team also added another development task: to use onboard computation for at least one phase of the competition. The ANIMA structure, although never before used on an actual robot, had been prototyped and used in simulations of both hierarchical and reactive robot systems. Most of the required work was to package it for the Denning robot, add an appropriate interface to the robot, and port all of the schemas as they were developed. With only about four months to complete all of these tasks, including the basic schema and communication development, it was clear that this would be a test of the flexibility and portability of both the hardware and software architectures.

The competition included three phases. In the first phase, each robot was to navigate the arena cluttered with obstacles without hitting anything, including the human judges. In the second phase, ten poles (labeled according to the needs of each robot) had to be recognized and visited, if possible, within a designated period. In the final phase, three of the previously-visited poles were designated to be visited in order. Additional information about the competition may be found in [8].

5.1 Robot description

The Denning MRV-3 is a three-wheeled cylindrical robot intended for general-purpose use, mainly in research. All three wheels turn simultaneously, providing (approximately) the ability to turn in place. The body itself does not rotate, except for gradual precession resulting from non-uniform slippage of the wheels against the floor. Twenty-four sonar sensors are equally-spaced around the body, as are six contact-switch bumpers. A single CCD camera was added to the standard configuration for use in the second and third phases of the competition. This camera was mounted on the top plate, which rotates to point in the direction of travel (along with the wheels). An infrared beacon detector was also available, but was not used during the competition.

The transputer architecture used for this implementation of ANIMA utilized five processors and an RS-232 interface spread over six TRAMs (integrated transputer daughterboards). The TRAMs were mounted on an PC-bus host board within a specially-packaged IBM-PC compatible system, complete with an electroluminescent display, a floppy disk drive, and a ruggedized hard drive. We have designed some other implementations which provide more flexibility with regard to usage of the processor links, but this system was more than adequate for the required tasks.

Although the performance of ANIMA benefits from separate high-speed channels to each physical sensor and effector, the Denning MRV-3 platform (like most commercial mobile robots) provides a single standard interface, in this case an RS-232 port. All communication with the sonar, infrared detectors, bumper switches, and motor controllers had to be multiplexed through this port.

The ANIMA hardware, of course, was restricted to using the onboard power sources. Since the robot may only function reliably for several hours even without the added
burden of multiple transputers and a PC host, it was important that ANIMA not consume any more power than necessary. Even with the disk drives and electroluminescent display active, the ANIMA system and host required only about 100 watts and did not significantly affect the battery life of the system.

5.2 Parallel structure

The utilization of the five processors is shown in Figure 3. AuRA's motor schemas and much of its perceptual schemas were included in the Reasoner process (which can easily be split among additional processors as necessary). Some aspects of the perceptual schemas (sensor data processing, mostly) were included within the appropriate logical sensors.

Because of the relatively low processing demands placed on the Integrator and Coordinator, these were combined onto a single processor, and messages to all logical devices were multiplexed on a single channel. These logical devices also ran as parallel tasks on a single processor, since no especially sophisticated processing was done at this level. Provision was made for inclusion of a separate processor (actually, a group of processors) to perform vision, using the remaining link from the logical devices processor. Although speech output was not used in Buzz, we have the appropriate logical effector to add it at any time, as indicated in the figure.

The Environment process actually serves a dual role. In normal operation, it passes messages along to the RS-232 handler process. In simulation mode, it intercepts commands to the robot and emulates the behavior of the robot in a grid-based environment, passing back sonar and bumper data when requested. An additional processor (not shown) is used in simulation mode just to provide a graphical display of the simulation status. The impact of this is that simulation capability is built into the real code—no porting is required to keep the simulation current relative to the actual robot software. Of course, the usefulness of any simulation depends on its fidelity. This organization allows the simulator, as a separate parallel process, to be enhanced at any time. We found that the simulation provided good qualitative results with regard to new robot behaviors which were subsequently tested on the actual robot.

5.3 Performance

Throughout the porting process, we were pleased with ANIMA's ability to perform sensor processing in the background. It was also possible to continuously keep track of the time between robot responses, providing the basis for a dead-man switch if the robot ceased communicating for any reason. The ANIMA-controlled system was able to negotiate obstacle-strewn areas about 50% faster than the Sun-controlled system, mainly because of the decreased latency of sonar data and the managed use of the RS-232 channel. We did not have sufficient development time to fully utilize the motor capabilities of the MRV-3, but we felt that additional performance improvements were easily possible. Much more benefit can be derived from the parallel structure as complex sensors like vision, more-sophisticated motor control algorithms, and additional schemas are added.

Phase 1 of the competition was intended primarily as a means of weeding out any robots which could not safely navigate within the arena in the presence of human beings. The judges deliberately stepped in front of the robots and corralled them into tight spaces, and Buzz performed satisfactorily. At about this time, some of the other robot teams were experiencing communication problems, since most were using the same frequencies for digital commands and/or video data. As one of the relatively few entries with all processing performed onboard (the eventual winner, the University of Michigan's CARMEL, was another), Buzz was immune from these problems in Phase 1. For the same reason, Buzz was also able to perform in combined demonstrations for the news media and the public, along with CARMEL and SRI's "Flakey."

Figure 3. Partitioning of tasks on Buzz.
5.4 Conclusions

Based on the relative ease in which the AuRA software was ported to the ANIMA architecture, it was clear that both components were sufficiently flexible and portable. Some of the specific features which aided this process were:
- the integrated development environment
- the inherent support for parallelism
- the use of generic proven CSP models
- the inclusion of simulation as a removable process

In phase 2 of the AAAI competition, the onboard Sun computer and radio link performed well, and Buzz ended up in second place, but radio problems in phase 3 limited us to a fifth-place finish overall. This seemed to indicate that a full port of the vision schemas would have improved our overall standing, since the machine would have been immune to radio interference. Since visual data could have been processed at much higher frame rates, it would have been possible to perform nearly continuous tracking, also improving Buzz's performance.

6 Future work

We have begun to adapt this system to a practical application for the Savannah River Site of the Department of Energy. Using another Denning robot, and adding onboard vision, we are building a prototype survey vehicle to monitor the condition of stored radioactive waste. As part of this effort, we intend to investigate the performance of ANIMA with a robot which has dedicated channels to the sonar and motor systems. This would eliminate the need to multiplex the data on a single RS-232 line, and overall system performance would improve considerably. ANIMA will also be used in all 3 phases of the 1993 AAAI competition.

As additional motor schemas are included, they will be placed in a parallel configuration, utilizing additional processors as necessary for the Reasoner. Eventually, we would like to make performance comparisons with other software architectures on the same hardware platform. Such comparisons would provide insight into the type of applications best suited for differing software architectures.

References

Navigation and Collision Avoidance System for a Mobile Robot
Subtask 1.3 - Vision (Landmark Tracking) Positioning System

Submitted to:
Westinghouse Savannah River Company

through
Education, Research and Development Association (ERDA) of Georgia Universities

in response to
RFP 92006EQ

Prepared by:
Frederick Anders
Abstract

To facilitate the inspection of drums of radioactive material in a storage warehouse, the Westinghouse Savannah River Company is considering the use of an autonomous mobile robot outfitted with video cameras. For accurate navigation of such a robot through the 3'-wide, 180'-long warehouse aisles, a machine-vision-based position sensing system utilizing retroreflective landmarks as reference aids is being investigated. The purpose of this research is to characterize the design issues involved -- including the relationships between optics, accuracy, illumination, and robustness -- in the course of constructing a prototype system.

At this point, preliminary data indicates that the resulting system meets the requirements for successful navigation of the inspection vehicle, although many more tests should be performed to work out any bugs before installation in such a safety-critical area. Lateral accuracies of ±0.5" can be expected for aisles up to 180' long at an update rate of about 40 Hz. The summary in Table 5 provides a qualitative sample of the design parameter relationships that must be considered when implementing such a camera-based system.

Given the expected ambient lighting conditions (indoor, fluorescent lighting), tests show that using a 2 mm aperture and a 300 watt incandescent bulb for illumination will yield very high contrast images that effectively subdue any background visual noise. Consequently, a simple fiducial design such as that depicted in Figure 13 should be sufficiently unique unless the illumination source causes excessive specular reflections off nearby drums within the field of view.

Finally, the hardware and software architectures of the system itself allow for great flexibility. Software options included in the camera firmware can be applied as needed for illumination control, camera gimballing, or selection of the fiducial geometry itself, while the host PC has various features from online camera debugging to specification of all landmark locations within the warehouse. The control and communications flow from pixel data to final output position was described and demonstrated to be an effective hierarchical structure for what could be considered a modular tracking system.

Research continues in two areas: First, further experimentation with strobe illumination should confirm whether or not it can serve as a practical, low-power replacement for the incandescent lighting used onboard the vehicle in the tests thus far. A bank of IR LED's would seem to be a particularly good approach. Secondly, a landmark design must be generated which works at both very short and very long distances. Such a fiducial could be either scalable or a hybrid of two landmarks used at different distances. Either way, both lateral offset and ranging information must be reliably obtained at all times.
Table Of Contents

Task Description .............................................................................................................. 1
Stinger 70C IVS .................................................................................................................. 2
General Approach ............................................................................................................ 3
Lens Selection ................................................................................................................... 5
Illumination ........................................................................................................................ 6
Landmark Design ............................................................................................................. 9
Hardware Considerations ................................................................................................. 13
Triangulation .................................................................................................................... 14
System Performance ........................................................................................................ 15
  CCD Resolution ............................................................................................................. 16
  Lateral Measurement .................................................................................................... 17
  Longitudinal Measurement .......................................................................................... 21
  Update rate ................................................................................................................... 22
Conclusions ....................................................................................................................... 24
References .......................................................................................................................... 27
Task Description

To facilitate the inspection of drums of radioactive material in a storage warehouse, the use of an autonomous mobile robot outfitted with video cameras is being considered. As the vehicle automatically navigates through aisles of stacked drums, cameras and other onboard sensors will be able to search for cracks, spillage, or other signs of contamination that merit attention.

The robot itself is a Transitions Research Corporation vehicle with two active drive wheels in the center and four additional caster wheels for stability. Its square perimeter measures approximately 40" long by 28" wide. The workspace of the robot will be the aisles of the warehouse, which may vary slightly from one warehouse to another, but which generally conform to the specifications below:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>Up to 180 feet</td>
</tr>
<tr>
<td>Width</td>
<td>Nominally 36&quot; from pallet to pallet, although drums may overhang pallets somewhat</td>
</tr>
<tr>
<td>Height</td>
<td>Three 35&quot; drums on 4&quot; pallets (≈10 feet)</td>
</tr>
<tr>
<td>Number of aisles</td>
<td>Typically 10 in each warehouse</td>
</tr>
<tr>
<td>Width at ends of aisles</td>
<td>36&quot; minimum. One end is usually large enough to accommodate a forklift (60&quot;).</td>
</tr>
<tr>
<td>Straight?</td>
<td>Yes, except for overhanging drums. It is possible to see from end to end through the aisles.</td>
</tr>
<tr>
<td>Level?</td>
<td>The concrete floor is sloped for drainage, and contains cracks at the slab joints.</td>
</tr>
<tr>
<td>Overhead clearance</td>
<td>Drums do not extend to the ceiling, but fire suppression equipment and support pillars do occupy the overhead area.</td>
</tr>
<tr>
<td>Ambient lighting</td>
<td>Varies. Typically fluorescent tubes. No windows to outdoors.</td>
</tr>
<tr>
<td>Background visual noise</td>
<td>Drums, support pillars, exit signs. Walls can be expected to have writing on them, such as fire codes, aisle numbers, etc.</td>
</tr>
</tbody>
</table>

Several efforts are being conducted in parallel to develop short range obstacle avoidance techniques based on ultrasound and computer vision. Another key aspect of the
the vehicle with its current \([X, Y]\) position in a global reference frame. Such information is necessary to close the position feedback loop and allow for making coarse navigational decisions as the vehicle moves down the aisle.

Many different approaches exist for such position measurement, including systems based on ultrasound, radio telemetry, laser scanning or rangefinding, and machine vision. The latter technique will be employed in this work, although the number of possible strategies still remains relatively broad: Functioning machine-vision-based positioning systems currently exist based on line following, counting visual landmarks as they are passed, detecting motion in the surrounding scene, etc. In particular, this work will focus on the design and construction of a system that triangulates position using the sight angles observed by onboard cameras to multiple retroreflective landmarks in the vehicle's surroundings. The design issues encountered in the system's development and their interrelationships will be investigated, and recommendations for use in the Savannah River Plant application will be given.

**Stinger 70C IVS**

The unique camera employed for this work is the Stinger 70C Integrated Vision System. This standalone, single-board CCD camera platform was originally designed for fixed position industrial automation and inspection tasks, and contains its own onboard microprocessor and serial communications interface. Several researchers have investigated its use in various applications such as industrial part presentation and robotic end-point feedback control. In particular, the references describe use of the Stinger 70C for offboard tracking and position measurement of an autonomous aerial vehicle over a workspace of very similar dimensions as those in the Savannah River Plant.

Figure 1 is a block diagram of the unit's hardware architecture. At the top of the camera, a 165 row by 200 column CCD ("charged coupled device") array integrates light from the scene. The resulting analog pictures are then shifted out of the CCD row by row through an A/D converter and stored in RAM memory, where the digital image data is then made accessible to the CPU: a Motorola 68000 microprocessor running at approximately 10 MHz. The CPU operates on the image data using machine-language instructions from a vision algorithm that has previously been written and stored in 16K of either RAM or EEPROM. Any results or external command inputs can then be passed through either an RS-232 serial port or a 5-bit parallel I/O port.

The strength of this camera lies in its ability to work as a standalone unit. Traditional machine vision platforms typically require a separate interface (such as a frame-grabber) and processor to support each camera, or they at least must divide the available processing power among the number of cameras that are attached. For its price (\(\approx 3,000\)), portability (each unit weighs 32 oz.), power consumption (800 mA at 5V), and C programming interface, it presents an attractive way to efficiently distribute processing power where it is needed.
The novel nature of the Stinger 70C camera also contributes its share of drawbacks, however. Since it was designed for semi-fixed industrial automation uses, it lacks some of the features which give most MV systems their flexibility. The limited resolution (165 x 192 pixels vs. 480 x 512 for most TV standard cameras), fixed optics, and lack of a C-mount for attaching standard lenses and irisers all led to many design tradeoffs, which will be discussed in the following sections.

**General Approach**

In implementing the video positioning system, the following design constraints were either stated or implied:

Range: The system had to be able to resolve vehicle positions over a rectangular workspace area of dimensions $3 \times 180$ feet.
Accuracy: With less than 8" total clearance between the sides of the vehicle and the drums, lateral positioning accuracy to ±1/4" or better along the entire length of the aisle was mandatory. Less critical was the longitudinal position, for which ±6" was targeted.

Robustness: With radioactive material involved, reliable, error-free position data was a must to ensure that the vehicle’s onboard control system wouldn’t respond violently to incorrect or sporadic position data and result in a crash.

Update Rate: To insure adequate data availability to control the vehicle at its maximum speed, as well as to allow for a timely response should it veer off course, the navigation system was required to have an update rate of at least 10 Hz.

Power consumption: Since the vehicle was battery powered, onboard power requirements were to be kept to a minimum to prolong the vehicle’s duty cycle.

Passive components: To promote serviceability and minimize human intervention in a radioactively "hot" environment, all active components were to be located onboard the vehicle.

Vehicle space: Finally, attention had to be paid to the size and weight of any onboard components, since real estate on the vehicle was at a premium.

The robot platform being considered has 3 degrees of freedom: X, Y, and Θ. In order to solve for the vehicle’s position, three unique measurements -- whether distances or angles -- were thus required. Initially, two of these could be based on landmark targets at opposing ends of each aisle. Two cameras carefully oriented 180° from one another could then resolve lateral deviations off the axis connecting the two targets down to the pixel resolution of the cameras. This angle alone is not sufficient to predict the position, however, as is illustrated in Figure 2. Since the vehicle is traveling directly between two targets, a triangulation singularity results along this line where very small angular errors in measurement lead to very large deviations in longitudinal (X) position. While longitudinal position accuracy is not of the utmost importance in this application, a third piece of information is required to nail down this last degree of freedom to reasonable accuracy.

The third measurement may come from a number of sources. The most accurate method (in terms of overall cartesian error) involves using a third camera viewing another landmark in a direction roughly perpendicular to the other two. This is so because given errors in the camera sight angles are magnified along the direction of the sight angles as they approach becoming parallel. Unfortunately, the height of the stacked drums here impeded seeing over the aisles to another wall, and the work involved in placing multiple targets on the ceiling or floor was also prohibitive.

Instead, it was decided to garner the last piece of data either from the size of one of the two facing targets or from the angle to an additional target nearby. Such information from each camera would provide redundant depth information for gauging the vehicle’s distance from the ends of the aisles. The final choice, however, could not be made until other tradeoffs had been resolved.
For a tracking application of such a dynamic depth of field (between 2 and 180 feet), selection of the proper fixed focus lens was a decision that clearly would affect every other. On one hand, the lens had to have a wide enough view to accommodate the size of a landmark target at very close range. On the other hand, the lens had to be telescopic enough for the CEO to be able to resolve the details of a landmark at the farthest expected distance. To aid in the selection, a spreadsheet was generated (Table 1) to compute both the field of view and pixel resolution of each available lens as a function of distance.

Given past experience and the anticipated level of background visual noise from the surrounding drums and pallets (see the section on 'Fiducial Design'), it was conservatively estimated that a typical landmark would have to span at least 8 columns in the image to contain enough information to guarantee uniqueness from background noise. While the width of the aisles themselves constrain the target size to a maximum of 36 inches, the size is further constrained by the field of view itself at close range. What results is a "catch 22" between the target size and the focal length: For a given target size, no lens exists with both the field of view to encompass it at close range and the resolution per pixel to detect it from afar.

Nevertheless, assuming an actual maximum target width of two feet to allow line-of-sight clearance for any drums that might be hanging over into the aisle, the lens would require a minimum resolution of $24/8 = 3$ inches per pixel. The 12 mm lens was thus chosen for its 2.38"/pixel resolution at 180 feet; however, it would be incapable of viewing the same two-feet-wide target at closer than 9 feet, even if it were perfectly centered in the image. Thus,
at the ends of the aisles, either a special case must be developed or positioning must be assumed by one of the other onboard means.

Illumination

The lack of camera support for anything but a fixed-diameter aperture made illumination of the landmark target another area of considerable difficulty. The Stinger 70C itself has two built-in means of altering the brightness of the picture. The first is by directly controlling an external strobe, LED, or incandescent light source. An external "ILLUMINATION" output pin doubles both to enable the amount of time a strobe is allowed to charge as well as to trigger the strobe or other illumination source when taking a picture.

<table>
<thead>
<tr>
<th>Constants: 2.64 mm = CCD length</th>
<th>200.00 pixels = CCD horizontal resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Lens Focal Lengths (mm):</td>
<td>9.00 12.00 18.00 24.00</td>
</tr>
<tr>
<td>FOV/range ratio:</td>
<td>0.29 0.22 0.15 0.11</td>
</tr>
<tr>
<td>Range (ft)</td>
<td>Field of view (ft)</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5.00</td>
<td>1.47</td>
</tr>
<tr>
<td>10.00</td>
<td>2.93</td>
</tr>
<tr>
<td>15.00</td>
<td>4.40</td>
</tr>
<tr>
<td>20.00</td>
<td>5.87</td>
</tr>
<tr>
<td>25.00</td>
<td>7.33</td>
</tr>
<tr>
<td>30.00</td>
<td>8.80</td>
</tr>
<tr>
<td>35.00</td>
<td>10.27</td>
</tr>
<tr>
<td>40.00</td>
<td>11.73</td>
</tr>
<tr>
<td>45.00</td>
<td>13.20</td>
</tr>
<tr>
<td>50.00</td>
<td>14.67</td>
</tr>
<tr>
<td>55.00</td>
<td>16.13</td>
</tr>
<tr>
<td>60.00</td>
<td>17.60</td>
</tr>
<tr>
<td>65.00</td>
<td>19.07</td>
</tr>
<tr>
<td>70.00</td>
<td>20.53</td>
</tr>
<tr>
<td>75.00</td>
<td>22.00</td>
</tr>
<tr>
<td>80.00</td>
<td>23.47</td>
</tr>
<tr>
<td>85.00</td>
<td>24.93</td>
</tr>
<tr>
<td>90.00</td>
<td>26.40</td>
</tr>
<tr>
<td>95.00</td>
<td>27.87</td>
</tr>
<tr>
<td>100.00</td>
<td>29.33</td>
</tr>
<tr>
<td>105.00</td>
<td>30.80</td>
</tr>
<tr>
<td>110.00</td>
<td>32.27</td>
</tr>
<tr>
<td>115.00</td>
<td>33.73</td>
</tr>
<tr>
<td>120.00</td>
<td>35.20</td>
</tr>
<tr>
<td>125.00</td>
<td>36.67</td>
</tr>
<tr>
<td>130.00</td>
<td>38.13</td>
</tr>
<tr>
<td>135.00</td>
<td>39.60</td>
</tr>
<tr>
<td>140.00</td>
<td>41.07</td>
</tr>
<tr>
<td>145.00</td>
<td>42.53</td>
</tr>
<tr>
<td>150.00</td>
<td>44.00</td>
</tr>
<tr>
<td>155.00</td>
<td>45.47</td>
</tr>
<tr>
<td>160.00</td>
<td>46.93</td>
</tr>
<tr>
<td>165.00</td>
<td>48.40</td>
</tr>
<tr>
<td>170.00</td>
<td>49.87</td>
</tr>
<tr>
<td>175.00</td>
<td>51.33</td>
</tr>
<tr>
<td>180.00</td>
<td>52.80</td>
</tr>
</tbody>
</table>
The second means is by software manipulation of the length of time that the CCD pixel elements are exposed to the scene.

The camera was designed primarily with indoor, strobe-lit applications in mind. Strobe lighting is highly desirable on an assembly line, where the reduced exposure times speed image acquisition and tend to "freeze" the image, reducing inaccuracies due to blurring. By having some active control over the light source itself, one can also better tolerate changes in any ambient lighting, especially if the strobe power is significantly greater than the ambient sources. The disadvantages of strobe lights primarily relate to their inherent trade-off between flash energy and flash rate. Strobe tubes with a high capacity for both tend to be quite specialized and require their own power supplies.

When a continuous light source is employed instead of a strobe, then the exposure time of the image must be considered. The camera hardware itself dictates a minimum time of 6 ms to shift the rows of analog image data from the CCD through an A/D converter to VideoRam. Since the camera is shutterless, the CCD pixels remain exposed to ambient light during this time, and vertical streaks through the image result if the lighting is too intense during this shifting. For this reason, the continuous light level must be controlled to produce the desired image gray-scale values with a total exposure time (shift out plus normal integration) of about 20 ms. With this exposure, approximately two-thirds of the resulting image is due to the actual scene, out-weighing the effects of the one-third that may have streaked. Smaller total exposure times increase the relative percentage of the image that is made up of the shift-out streaks. Longer exposures slow down whatever process the camera is running, since the camera can only take an image or process it -- not both at the same time.

Indoor landmark tracking has the important advantage that it is usually possible to constrain the environment. Background visual noise that might interfere with the tracking algorithm is typically known or can be avoided. Ambient lighting is usually constant, and this allows for using fixed camera optics. Finally, since the light levels are typically quite low indoors, the landmark itself can be illuminated in such a way as to further contrast it from background noise.

The most attractive way to achieve this latter goal is through the use of retroreflective landmarks. Retroreflectior\textsuperscript{a} -- most commonly seen on roadway signs -- has the property of reflecting light waves exactly back along the incident path, regardless of the angle of incidence. Thus, placing a light source just behind a camera aimed at a landmark covered with retroreflective material results in the retroreflector appearing very bright in the picture - - almost as bright as if the camera were looking at the illumination source itself. In fact, it was found that using the proper combination of illumination power and aperture size resulted in images that often contained the landmark and nothing else. (Fig. 3). For indoor operation, a fixed 2 mm diameter aperture was used in conjunction with a 300 Watt incandescent bulb for effective landmark illumination at distances up to 140 feet at the ideal 20 ms exposure time. For any given application, the design problem is one of picking a small enough aperture to subdue the effects of the ambient lighting and a bright enough illumination source to
illuminate the retroreflector. Experimentation showed that the ambient illumination power of most standard fluorescent, incandescent, or sodium vapor lighting is negligible compared to the retroreflected intensity of a lightbulb. Thus, indoor images similar to Figure 3 can be expected in most cases.

Figure 3 - Sample Indoor Retroreflective Target Image

For its ease of use and availability, all the tests for this work were performed using a 300 watt incandescent bulb. For greater range and lower power consumption, however, several alternatives merit investigation. By releasing its energy only when signalled by the camera, a relatively low-power xenon strobe light could have been used with similar optics. This approach was not taken due to the effects the sporadic pulses of light might have had on the other onboard vision systems (such as those utilized in Subtask 1.2), as well as the additional driver circuitry required.

As a final alternative, infrared light emitting diodes themselves would make excellent single-point landmarks. As shown in Fig. 4, the spectral response of the CCD chip peaks in the near infrared range. Experiments showed that a single 5 V, 40 mA IR LED was able to saturate the pixel elements at the same distances and using the same optics as the retroreflective arrangement described above. Although mis-aiming the LED by only 10° resulted in a near total loss of the radiated energy, this would not present a problem in a 180° long, 3' wide corridor. Furthermore, since indoor lighting typically contains little IR, a IR-transmitting filter could be used to filter out background noise while a simpler, faster "brightest pixel" search could be used in tracking such a landmark. Unfortunately for this application, IR beacons violate the requirement of having all active elements onboard the vehicle; however, banks of IR LEDs should provide enough illumination at the camera to illuminate a retroreflective landmark. Even if 100 LEDs were required, their resulting 20 W
[continuous] current draw would be a drastic reduction from the 300 W wasted by the incandescent bulb.

![CCD Spectral Responsivity](image)

Figure 4 - Spectral Responsivity of the TC-211 CCD

**Landmark Design**

Since the key to a video tracking system is "seeing" the target, design of the landmark and its corresponding search algorithm should reflect everything that is desired about the system as a whole: namely speed, accuracy, and robustness. To achieve speed, the fiducial must be simple to recognize and process. For accuracy, it must contain elements which enable measurement of exact locations important to position computations. Finally, to insure robustness, the fiducial must be unique enough to guarantee that it will not be found among the background noise in the image.

For simplicity's sake, it was assumed that the landmarks would be constructed of laminated paper or plastic and affixed directly to the walls of the warehouses. The landmarks would most likely be simple geometric shapes or patterns, but had to be reliably dissimilar from any naturally-occurring structured or random visual noise in the warehouse environment - such as the repeating edges of barrels or lettering on the walls. In addition, a fiducial designed for tracking had to be scalable to be recognized whether nearby or at great distance, whether it spanned the entire image or the minimum number of pixels to detect it. These two factors together made meeting the "uniqueness" constraint so difficult. In the average environment, the background visual "noise" contains a little bit of everything a search algorithm could be based on, from even shadings to repeating patterns to total randomness.
Design of the fiducial geometry is an inexact science that involves a number of trade-offs. In a mobile system such as this, the basic tenet is never to lose sight of the landmark. The tradeoff is one of speed versus complexity. On one hand, a system will work fine with the most basic of targets as long as it can operate so rapidly that the target is never lost. Once lost, however, correct reacquisition can be difficult if the target is not unique to the image. At the other extreme, a highly unique target might process more slowly given the same hardware, but would be much more reliable to reacquire in the event of a temporary loss.

The approach taken here is the former, following the "K.I.S.S." principle. The reason for this lies primarily in the hardware being utilized: The Stinger 70C's pixel resolution and lack of onboard floating point support simply prohibit high-level, computationally intensive techniques such as pattern transformations or most correlations. The fact that the size of the targets varies with distance would require size scaling of a correlation mask in addition to the normal translation over the rows and columns of the image. A gray-scale correlation would have to be adaptive from frame to frame to account for changing lighting conditions as the vehicle passed under overhead lights or traveled closer to a reflective landmark. What is left are only the most basic of image processing tools, but, since the freedom to design the landmarks is reserved, they are all that is necessary. For this application, simple edge detection (using a gradient threshold) and brightest/darkest pixel searches are all that will be employed. For a general introduction to all of the abovementioned techniques and their usage, see the references.

The primary edge detection algorithm used is a simple gradient threshold approach: As pixels are sampled in a straight line, the differences in gray scale level from one pixel to the next are computed. If the difference between two pixels exceeds a certain threshold value and that difference is greater than the differences immediately bounding it, then that local gradient maximum/minimum is assumed to be an edge in the image. This concept is illustrated in Figure 5. To maximize the gradients at each edge (and hence the likelihood of edge detection), the landmarks are all high-contrast black and white patterns. Some of those that were considered and tested for uniqueness are depicted in Figure 6.

In searching for a landmark, the cameras always begin by scanning across a horizontal row for a given number of alternating light-dark and dark-light transitions. The initial horizontal row scan is a key aspect of the fiducial search algorithms: Since the vehicle moves at roughly a constant height, the image row at which the fiducial appears from frame to frame is expected to remain relatively constant as the vehicle traverses the workspace. By initially scanning outward from the same row as the last successful acquisition, the odds are favorable that the fiducial in the next frame will also be acquired on this same row or on a row very close to it, speeding the search process.

Once a user-programmable number of light-dark / dark-light transitions are detected within a row, each fiducial search algorithm then calls any of a number of generalized routines which can (1) check the horizontal spacing between each transition against what was expected, (2) perform additional gradient tests in other directions, (3) compare the assumed landmark
Figure 5 - Gradient Threshold Edge Detection

Slope here exceeds a threshold and is a local minimum; therefore this is an edge.

Figure 6 - Typical High-Contrast Fiducials

location against the trend of previous values, or (4) compute the center of the fiducial for
passing back as a result. To achieve the angular resolution required to yield the desired cartesian accuracy, the straightforward method for performing this last task is to employ "subpixelization" to a known edge. Subpixelization is the process of fitting a curve to the discrete pixel data to interpolate a more accurate result, and is illustrated in Figure 7. A simple, integer math formula fits the magnitudes of three successive gradients to a parabola, yielding the position of the parabola's peak (the interpolated maximum gradient, or the location of the high-contrast edge) as an offset from the middle pixel. Using this method usually produced landmark locations accurate to within one tenth of a pixel.

\[
dp = \frac{g(p+1)-g(p-1)}{2*\left[2g(p)-g(p+1)-g(p-1)\right]}
\]

Figure 7 - Subpixelization

With the knowledge now acquired about the optics and search strategy, the design of the actual fiducial geometry itself could be finalized. As illustrated in Figure 6, many different target patterns were investigated. As target complexity and the number of edge tests increased, both the incidence of "false" targets and the processing loop speed decreased. For indoor, retroreflective tracking, it was found that the "ring" fiducial was sufficiently unique for most uses. If sample images of the background environment are available, the best way to choose the fastest target that won't accidentally be picked up in background noise is simply to load several fiducial search algorithms and allow a camera to scan the background for matches. Given the visual noise that is potentially present in any given surroundings, it is impossible to design a single "best" fiducial, and one can only design for what one expects.

To wrap up the issue of how to obtain the third reference required for triangulation, recall that increased target separation yields more accurate triangulation results. However, an excessively removed, identical, ring-shaped third target would lie outside the fields of view of the two original cameras at close range. As a compromise solution, the segmented arrangement shown in Figure 8 allows the two cameras to gather the maximum depth information available to them at a given distance. After initially locating the ring (satisfying the uniqueness constraint), each camera scans vertically upward to locate the farthest depth gauging bar in its field of view. Subpixelization is performed on both the center of the circle and the distance to the gauging bar, and this information plus the identification of which gauging bar is being used is passed to another machine for triangulation. Actually, since the cameras are both gauging their distance from the opposing walls, there is redundant information, and two separate triangulations can be averaged together for more noise-immune
results. Expected and measured accuracies of this scheme are presented in the "System Performance" section.

**Hardware Considerations**

The two cameras are mounted on the vehicle as shown in Figure 9. With the cameras bracketed together one above the other, the same light source can be used for coaxial illumination of the retroreflective landmarks for both. The bottom camera is permanently secured in the holding bracket, while the top camera can be adjusted within the bracket around its yaw axis by means of the ¼-20 screws through its case. This allows for simple manual calibration of the lines of sight of both cameras along parallel lines -- facing exactly 180° apart. To facilitate accurate alignment, the bracketed pair is physically placed directly along the axis connecting the landmark targets with the aid of a laser pointer. With the cameras running, their outputs are then observed, and the adjustment screws are turned until the camera's angle-to-target measurements yield a difference of 180.0°. Note that the ≈1 foot offset between the optic elements shown in the figure can translate into a finite positional error, but this error is negligible at the distances and angles involved.
A special case that must be considered is when the robot base is not aligned with the aisle -- for example, when turning to avoid a protruding drum or to navigate to the next aisle. If the cameras were rigidly attached to the base, they would no longer point at the landmarks, and the vehicle would become lost. To allow for such changes in the robot's orientation ($\Theta$), the cameras are mounted on a 180° turntable which is activated by a servomotor controlled by the NavCom PC (described below). Since this supervisory machine is in constant communication with the cameras, it can adjust the turntable to keep the landmarks centered in the cameras' fields of view. Additionally, it can stop the vehicle and alert an operator if the vehicle somehow moves into a position that makes it impossible to simultaneously view both landmarks. Note that neither the resolution nor the accuracy of the motor drive are of particular consequence. The parallel alignment of the cameras and the fact that they both are able to see their respective targets are all that is required. For an example of a low-cost (<$100) servomotor-driven gimbal used previously in conjunction with the Stinger 70C, see the references$^5$.

**Triangulation**

As the cameras are busy searching for targets, they are constantly sending their results over an RS-232 serial line to a central navigation computer (referred to here as the "NavCom"). This machine combines the sight angle results from the cameras to compute a vehicle position. It is also used for the initial manual calibration of each camera and the debugging of camera operations on the fly.

In its present configuration, the NavCom is a 486/50 MHz PC-compatible machine with sufficient floating point math and graphics speed to compute and display camera views.
and vehicle positions in real time. Ultimately, the transputer control system being developed as S.O.S. Item 1.4 of this contract would assume these duties. The NavCom software was written in the C programming language for portability to another such platform.

Similar to the camera software architecture, the NavCom software defaults to looping continuously unless external commands are given. Under normal operation, the process begins by repeatedly polling the camera serial ports until a new result is received from one or both cameras. If any result is different from the last one a camera reported, the NavCom recomputes a new vehicle position and velocity using buffers containing the most recent data from each camera. These results are updated on the screen and optionally passed on to another microprocessor responsible for navigating the vehicle. The program then returns to the beginning of the loop, checking for more new data.

Optionally, as each new position update is computed (frequently at almost 50 times a second), it is stored with a timestamp in a circular buffer which holds the most recent results. These data points are then differentiated with respect to time and averaged over the buffer to obtain a smoothed estimate of the vehicle's velocity. This value can then be compared with the output of other sensors (such as shaft encoders on the vehicle's wheels) as a check on the validity of the data. The size of the buffer controls the smoothness and update delay of this velocity information, i.e. a bigger buffer size differentiates and averages output results over a longer period of time, reducing the effects of any noise in the data but producing a longer delay.

In addition to the default tracking loop, the NavCom software contains several important user interface routines to facilitate setup and monitoring of the system. Although the cameras themselves operate as independent units, the NavCom machine serves as a terminal interface for altering camera parameters, downloading images for inspection, and observing more detailed results of the fiducial search algorithm in near-real-time. A calibration menu allows the user to correct for any errors in alignment of the cameras. Finally, sound and display options facilitate debugging of the system.

**System Performance**

**CCD Resolution**

To confirm the ability of a camera itself to resolve landmark positions to pixel or even subpixel accuracies, a retroreflective bar code target was moved along a tape measure oriented normal to its line of sight at a perpendicular distance of 75 feet (Figure 10). Note that, according to Table 1, at 75 feet the 12mm lens should have yielded a CCD resolution of 0.99 inches per pixel. The experimental data (Table 2) approximately confirms that for every inch the target was translated along the tape, the landmark moved one column pixel within the image.
Figure 10 - Experimental Determination of CCD Resolution

Table 2 - Normal Pixel Resolution Correlation
(12 mm focal length lens at 75 feet, nominal 1:1 inch:pixel correlation)

<table>
<thead>
<tr>
<th>Landmark Y-position on tape measure (inches)</th>
<th>Camera-observed landmark location (pixel #)</th>
<th>Incremental change in observed location (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2.0</td>
<td>102</td>
<td>---</td>
</tr>
<tr>
<td>-1.5</td>
<td>102</td>
<td>0.0</td>
</tr>
<tr>
<td>-1.0</td>
<td>101</td>
<td>-1.0</td>
</tr>
<tr>
<td>-0.5</td>
<td>101</td>
<td>0.0</td>
</tr>
<tr>
<td>-0.0</td>
<td>100</td>
<td>0.0</td>
</tr>
<tr>
<td>0.5</td>
<td>100</td>
<td>-1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>100</td>
<td>0.0</td>
</tr>
<tr>
<td>1.5</td>
<td>99</td>
<td>-1.0</td>
</tr>
<tr>
<td>2.0</td>
<td>99</td>
<td>0.0</td>
</tr>
<tr>
<td>2.5</td>
<td>98</td>
<td>-1.0</td>
</tr>
<tr>
<td>3.0</td>
<td>98</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Observation of the fact that the column pixel output from the camera changed very abruptly as the landmark was moved slowly across the tape led to speculation that the CCD was in fact capable of resolving angles at a grain finer than the pixel spacing itself. As a more rigorous test, the software was enhanced to locate the target edges to subpixel accuracies using the method described in the 'Fiducial Design' section. While such an algorithm nominally computed the location of these edges to within 1/256th of a pixel, hardware-
dependent fluctuations led to too much noise to accurately determine the hundredths place. As the sample data in Table 3 shows, however, the camera was able to reliably resolve the expected one pixel to one inch correspondence nearly down to tenths of a pixel. One notable effect that must be considered during normal operation is that the particular target edge that a camera is tracking may "bloom" outward with excessive incident light, distorting the results as the robot closes the distance to the landmark. This is currently compensated for with an autoexposure routine on the camera, which samples the pixel values over the image and adjusts the exposure time accordingly to maintain an evenly lit image.

Table 3 - Pixel Resolution Correlation with Subpixelization
(12 mm focal length lens at 75 feet, nominal 1:1 inch:pixel correlation)

<table>
<thead>
<tr>
<th>Landmark Y-position on tape measure (inches)</th>
<th>Camera-observed landmark location (pixel #)</th>
<th>Incremental change in observed location (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>100.3</td>
<td>---</td>
</tr>
<tr>
<td>0.12</td>
<td>100.2</td>
<td>-0.1</td>
</tr>
<tr>
<td>0.25</td>
<td>100.0</td>
<td>-0.2</td>
</tr>
<tr>
<td>0.37</td>
<td>99.8</td>
<td>-0.2</td>
</tr>
<tr>
<td>0.50</td>
<td>99.6</td>
<td>-0.2</td>
</tr>
<tr>
<td>0.62</td>
<td>99.5</td>
<td>-0.1</td>
</tr>
<tr>
<td>0.75</td>
<td>99.4</td>
<td>-0.1</td>
</tr>
<tr>
<td>0.87</td>
<td>99.3</td>
<td>-0.1</td>
</tr>
<tr>
<td>1.00</td>
<td>99.2</td>
<td>-0.1</td>
</tr>
<tr>
<td>1.50</td>
<td>98.7</td>
<td>-0.5</td>
</tr>
<tr>
<td>2.00</td>
<td>98.3</td>
<td>-0.4</td>
</tr>
<tr>
<td>2.50</td>
<td>97.8</td>
<td>-0.5</td>
</tr>
<tr>
<td>3.00</td>
<td>97.4</td>
<td>-0.4</td>
</tr>
<tr>
<td>3.50</td>
<td>96.6</td>
<td>-0.8</td>
</tr>
<tr>
<td>4.00</td>
<td>96.3</td>
<td>-0.3</td>
</tr>
<tr>
<td>4.50</td>
<td>95.7</td>
<td>-0.6</td>
</tr>
<tr>
<td>5.00</td>
<td>95.4</td>
<td>-0.3</td>
</tr>
</tbody>
</table>

Lateral Measurement

To avoid hitting the storage drums, lateral (Y) position relative to the aisle centerline is by far the most important measurement. Such a measurement is primarily dependent on the lateral offset angle, which is defined as the yaw angle lying in the plane parallel to the ground and measured by the cameras between the lines of sight to two landmarks (angles $\frac{\pi}{4}V_1L_2$ in Figure 11). The curves in the figure show possible locations for a vehicle given different lateral offset angles. For example, angles of $>180^\circ$, $180^\circ$, and $<180^\circ$ in the figure would place
the vehicle above, on, and below the X-axis centerline, respectively. For lateral positioning, notice that given values of the offset angle result in the greatest lateral deviation when the vehicle is centered between the landmarks, at $X = 90$ ft.

Using a 12 mm lens, each column pixel in the camera encompasses a field of view of $(12.55°/200$ pixels) = .0628°/pixel. Based on the subpixelization results of the preceding section, it is assumed that a single camera can reliably determine landmark locations within the image accurate to one quarter of a pixel, for a theoretical best resolution of .0157° for a single camera. Since the lateral offset angle is obtained as the difference of the angular measurements of two cameras, this tolerance must be doubled to .0314°. For a vehicle positioned halfway down the 180° aisle, such a tolerance trigonometrically should therefore yield a maximum lateral deviation of 0.59". While this is better than the design specification of 1.0", errors inherent in calibration and landmark installation would probably result in typical accuracies on the order of 1.0°. As an aside, angular measurement errors will trigonometrically have a greater effect on the accuracy of the result as the nominal offset angle deviates farther from 180°. In a long corridor with an aspect ratio of 360:1, however, this effect can be disregarded.

To confirm this prediction, a typical vehicle workspace was simulated in an outdoor setting (Figure 12). Two simplified, retroreflective landmarks (Figure 13) were positioned 180.3’ apart, and a tape measure was run exactly down the axis between them (a laser pointer was employed to confirm linearity to within ±1”). To avoid having to deal with background noise, tests were conducted at night using a 300 Watt incandescent bulb for illumination of both landmarks. Similar to what might be expected in a warehouse, the center expanse was
composed of jointed concrete sidewalk with a slight uphill grade (perhaps $\frac{1}{2}\degree$) from one landmark to the other ($X=0$ ft to $X=180.3$ ft). As described in the 'Hardware Considerations' section, two cameras were bracketed together and placed at the point [90, 0]. The bracket
adjustment screws were then turned until the two cameras indicated a 180.0° difference in the sight angles to their respective landmarks.

To test lateral resolution and accuracy, a second tape measure was oriented perpendicularly to the first across the point [90, 0]. The simulated vehicle was then moved along this lateral tape while the camera/PC-measured deviations were recorded against the actual tape reading. Data was taken for approximately four feet of lateral movement. As the resulting X-Y plot in Figure 14 shows, the system measurement was reliably within ±1" of

![Figure 14 - Lateral Accuracy Test Results](image)

the actual position. Furthermore, since the intended vehicle is physically constrained to only ±4 inches of lateral movement, measurements were repeated for a short distance at up to ¼" increments. Again, the data indicates that the cameras are capable of resolving the lateral offset angle to sufficient accuracy.

Longitudinal Measurement

The distance in pixels between visible depth gauging markers yields the range information needed to constrain the vehicle’s longitudinal (X) position. Expected angular resolution is the same as in the preceding section, although the accuracy of the resulting depth information will depend on the nominal height measurements. That is, errors in the height angle measurements will be magnified by shallow height angles themselves (according to the cosecant of the angle), and the ability to utilize depth marks that encompass the full camera field of view should result in more precise rangefinding.
Figure 15 shows the results of a longitudinal accuracy test using the same equipment arrangement as in the previous test. As the vehicle was translated in 5-foot increments down the "aisle" centerline (as measured on a tape measure), this actual position was compared to that returned by the camera system. The graph is a plot of the deviation between the two measurements as a function of the actual position. Since the ranging information from both cameras is redundant, deviations based on each single camera alone are included for comparison.

The data indicates that the ranging information obtained from the height of the landmarks is not nearly as accurate as the angular data used for lateral positioning. In fact, deviations up to nearly 8 feet (out of 180) were observed in one of the cameras. Such a gross deviation might be because the experimental test used only a single, fixed pair of depth-gauging markers, which did not take advantage of the camera’s full field of view at long distances. As the size of the entire landmark shrank within the CCD’s view, the "signal-to-noise" ratio might have as well. This is evidenced in the graph where the largest deviations originate from camera #1 at X=35', at which point camera #1 would have been viewing its landmark from 145' away. Other possibilities include bugs in either the camera or PC software, although none have been isolated to date.

Nevertheless, it should be pointed out that, despite the errors in longitudinal measurement, the more-important lateral reading never deviated from the nominal centerline position (Y=0) by more than 1" as the vehicle was moved down the length of the aisle.
Other Sources of Error

Many small factors contribute to errors in the measurements. Those related to the cameras themselves include deviation of the CCD's yaw-axis from perpendicular to the workspace plane, defocussing of the lens, position and orientation errors during calibration, and the resolution of the pixels themselves. Similarly, errors in the initial positioning and orientation of the fiducials — particularly the spacing of the depth gauging markers — also lead to resulting inaccuracies. The relative expected magnitudes of such error sources are covered in more detail in the references \(^5\). Experimentation showed, however, that the targeted accuracy could easily be attained using the calibration procedures described in the experiments.

Vehicle movements could be expected to induce some variability in the measurements. Small bumps and vibrations at the vehicle could potentially translate into significant blurring of a landmark at 100'+ distances. This was found not to be a the case, however. With the vehicle moving at a typical speed of \(\approx 6\) inches/sec, examination of the raw data stream revealed very few fluctuations of the lateral position data. This could be the result of measuring a differential angle between two cameras, where bumps affecting both cameras are effectively cancelled out. Furthermore, what noise did exist could be filtered or averaged out due to the fast update rate.

**Update rate**

The update rate and time lag of the system are dictated by a serial chain of processes that individually run quite fast, as shown in Table 4, below. During ideal operation, the cameras are independently and continuously tracking the target at 37 Hz, while the NavCom PC is combining new results and generating new output at 41.7 Hz. Adding these two together in series yields the time delay induced while a change at one camera propagates through to a new positional result: 51 ms or 19.6 Hz.

There are many factors which will slow down the camera loop, although the NavCom loop will keep running at the same rate whether there is new data from the cameras or not. The main occurrence that slows the cameras to a relative crawl is losing sight of the landmark. Losing the landmark potentially means having to process the entire frame until it is reacquired, which can take three quarters of a second due to the number of rows that must be scanned. As long as the landmark is being tracked, however, the cameras can run nearly as fast as they can take a picture (\(\approx 60\) Hz with strobe illumination).
Table 4 - Component Timing

<table>
<thead>
<tr>
<th>Component</th>
<th>Time</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stinger 70C's:</td>
<td>13 ms</td>
<td>CCD exposure time (typical, using incandescent light)</td>
</tr>
<tr>
<td></td>
<td>6 ms</td>
<td>CCD image shift-out time</td>
</tr>
<tr>
<td></td>
<td>6 ms</td>
<td>Image processing and computations</td>
</tr>
<tr>
<td></td>
<td>2 ms</td>
<td>Output results (7 bytes @ 38.4 Kbd)</td>
</tr>
<tr>
<td>NavCom:</td>
<td>21 ms</td>
<td>(average) Retrieve any new results from buffers and recalculate position and velocity</td>
</tr>
<tr>
<td></td>
<td>3 ms</td>
<td>Output results (10 bytes @ 38.4 Kbd)</td>
</tr>
<tr>
<td>Total:</td>
<td>51 ms</td>
<td>(19.7 Hz)</td>
</tr>
</tbody>
</table>

Conclusions

In the course of developing a landmark-based video positioning system for a wheeled mobile robot, the relationships and interdependencies between many design parameters were characterized. Table 5 provides a qualitative sample of those that must be considered when implementing such a camera-based system.

Preliminary data indicates that the system as constructed would meet the requirements for successful navigation of the inspection vehicle at the Savannah River Plant, although many more tests should be performed to work out any bugs before installation in such a safety-critical area. Lateral accuracies of ±0.5" can be expected for aisles up to 180' long at an update rate of about 40 Hz.

Given the expected ambient lighting conditions (indoor, fluorescent lighting), tests show that an aperture size and illumination power similar to those used in the experiments (2 mm and 300 W, respectively) will yield very high contrast images that effectively subdue any background visual noise. Consequently, a simple fiducial design such as that depicted in Figure 13 should be sufficient unless the illumination source causes excessive specular reflections off nearby drums within the field of view.

Finally, the hardware and software architecture of the system allows for great system flexibility. Software options included in the camera firmware can be applied as needed for
illumination control, camera gimballing, or selection of the fiducial geometry itself, while the NavCom has various options for camera debugging and control as well as specification of all landmark locations within the warehouse. The control and communications flow from pixel data to final output position was described and demonstrated to be an effective hierarchical structure for what could be considered a modular tracking system.

Research continues in two areas: First, further experimentation with strobe illumination should confirm whether or not it can serve as a practical, low-power replacement for the incandescent lighting used onboard the vehicle in the tests thus far. A bank of IR LED’s would seem to be a particularly good approach.

Secondly, a landmark design must be generated which works at both very short and very long distances. Such a fiducial could be either scalable or a hybrid of two landmarks used at different distances. Either way, both lateral offset and ranging information must be reliably obtained at all times.
Table 5 - Summary of Camera System Dependencies and Interrelationships

<table>
<thead>
<tr>
<th>Distance to Landmark</th>
<th>Focal Length / Field of View</th>
<th>Aperture Size</th>
<th>Illumination Source</th>
<th>Fiducial Geometry</th>
<th>Update Rate</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focal Length / Field of View</td>
<td>Lens must be wide enough to encompass target, yet telephoto enough to resolve details at maximum distance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aperture Size</td>
<td>Smaller aperture gives longer depth of field</td>
<td>A more telescopic lens requires a wider aperture to achieve the same image brightness</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Illumination Source</td>
<td>Further distance requires more illumination power to view retroreflector.</td>
<td>(&lt;- same as)</td>
<td>Less light admitted through a smaller aperture requires a more powerful illumination source</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fiducial Geometry / Size</td>
<td>Similar to Distance/FOV relation. Additionally, in this appl., using a &quot;variable height&quot; fiducial allows the camera to capitalize on the best info. available for ranging</td>
<td>Fiducial size is typically limited by the available space, which in turn dictates the FOV necessary to achieve the desired resolution at max. range</td>
<td>Smaller apertures can effectively subdue background noise normally illuminated by ambient lighting, reducing the need for complex and unique landmarks.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Update Rate</td>
<td>With a gimbaled camera system, a narrower FOV lens requires a faster update rate for the sensor system just to keep the landmark within the image. Conversely, a wider FOV demands less frequent movement.</td>
<td>Smaller aperture -&gt; longer exposure time -&gt; slower loop speed</td>
<td>More powerful illumination -&gt; shorter exposure time -&gt; faster loop speed</td>
<td>A simpler landmark search algorithm will process more quickly</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>As the fiducial becomes smaller within the image, the &quot;signal-to-noise&quot; ratio decreases. That is, the measurement of actual edges is overcome by blurring due to vibrations, hardware dependent variations, etc.</td>
<td>Increased depth of field results in a better-focused image at a wider range of distances, yielding more consistent results</td>
<td>A consistent reflected energy will reduce variations in edge detection due to target blooming/dimming. A xenon strobe or other instantaneous source will also freeze the image to reduce blooming.</td>
<td>The fiducial must contain elements which enable very accurate registration of key locations needed for triangulation</td>
<td>In combining data from multiple, asynchronous cameras, faster update rates reduce the time delay between independent samples, improving both the position est. as well as the velocity computation</td>
<td></td>
</tr>
<tr>
<td>Robustness</td>
<td>Same as Accuracy vs. Distance. A barely perceptible landmark will more likely be lost due to abrupt vehicle movements or lighting variations</td>
<td>(Indirect - see Accuracy vs. Distance &amp; Distance vs. FOV)</td>
<td>Too little illumination makes it impossible to see the landmark. Too much results in a loss of the target due to &quot;blooming.&quot;</td>
<td>Simple fiducials may not be unique to the image (i.e. background visual noises)</td>
<td>Faster update rates allow for more timely response should the system lose track of the vehicle's position.</td>
<td>With only 8&quot; total clearance between the vehicle and storage drums, accurate measurements are critical to successful operation</td>
</tr>
</tbody>
</table>
References

[1] Telephone conversation with Mr. David Wagner, Westinghouse Savannah River Company

[2] Dickerson Vision Technologies, Advanced Technology Development Center, Georgia Institute of Technology, Atlanta, Georgia.


Computer Controllable Xenon Strobe for Machine Vision Applications

Michael J. Schreiber, Graduate Research Assistant
Stephen L. Dickerson, Professor
George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

I Objective:
This paper introduces a computer controllable strobe illumination device for machine vision applications. The overall strobe operation is explained as well as the purpose of individual components. Strobe testing which was performed to determine strobe specifications is presented.

II. Strobe Operation
The strobe tube circuit is designed to take as inputs: 115 VAC, 12 VDC, and a pulse width modulated 5 Volt signal to produce strobe flashes. The strobe's light energy outputs is controlled by a PWM signal. This is a prototype of DC powered units which will be used for remote camera operations.

Figure 1 shows the strobe circuit used for testing purposes. There are primarily two parts to the strobe circuit; a strobe power circuit and a flash triggering circuit. The strobe power circuit transforms a 115 VAC to 230 VAC signal which supplies energy to the two strobe capacitors, 1uF and 40 uF after being full wave rectified. The maximum voltage across the capacitors is ~335 volts. The smaller capacitor is used to rapidly get the voltage across the strobe tube to ~300 volts for triggering. Whereas the 40 uF capacitor is used to vary the strobe light energy. These capacitors are combined with power resistors to provide the time constants of the system. A 100 ohm and a 340 ohm resistance rated at 10 Watts are used. Experimentally, this provides a 112 us time constant for the small capacitor which is accurate to ~290 V. The larger capacitor has an experiment time constant of 18.9 ms. Thus 98.1% of the maximum energy can be achieved in 77 ms (4 time constants). A small portion of the energy goes to the 0.047 uF trigger capacitor which stores energy needed to start the illumination process. A large 380 uF capacitor rated at 350 Volts is used to act as a voltage source so the current draw on the transformer is buffered. Note, transistors and diodes are used that are rated for currents lower than seen during operation. This is acceptable since the component only see this current for a the very short time.

The variable strobe light intensity is accomplished by varying the charging time of the strobe capacitors before the strobe is signaled to fire. This charge time is represented by the low time of the trigger signal. When the trigger signal is low, the PNP transistor base is ground which allows a positive voltage at the power mosfet base. This connects the capacitors to ground which completes the circuit and allows the capacitors to charge. When the trigger signal goes high, the base of the mosfet is grounded and charging stops. At the same time, the strobe tube is triggered to flash.

Trigger occurs because the base of the NPN transistor is set high signal which allows 12V on the 6 A Mosfet allowing the 0.47 uF capacitor to induce a current through the trigger transformer. The capacitors energy is transformer to ~6000 V which starts the strobe to begin emitting light.

Once the strobe has started to emit light this has the effect of shorting the circuit. Therefore, all the stored capacitor energy is discharged through the strobe tube. Discharging continues until the voltage across the capacitors is too low to maintain the flow of ions, ~5V. The fast recovery diode allows the 40 uF to discharge without energy dissipation through the 340 W resistor.
Figure 1
## Strobe Parts List

### Resistors
- **R1**: 100 ohm, 10W
- **R2**: 335 ohm, 10W
- **R3**: 46 Kohm
- **R4**: 5.6 Mohm
- **R5**: 270 ohm
- **R6**: 500 ohm
- **R7**: 1 Kohm
- **R8**: 4.7 Kohm
- **R9**: 1 Kohm

### Capacitors
- **C1**: 380 uF, 350 V
- **C2**: 1 uF, 630 V
- **C3**: 40 uF, 450 V
- **C4**: 0.47 uF, 600 V
- **C5**: 500 uF, 16 V

### Transistors
- **Q1**: Power Mosfet 6 A, 450 V, SK9505 (Ack)
- **Q2**: NPN transistor, 2N2222A
- **Q3**: Power Mosfet 6 A, 450 V, SK9505 (Ack)
- **Q4**: PNP transistor, 2N4209

### Integrated Circuits
- **IC1**: Hex Buffer, 7407
- **IC2**: 5 Volt Regulator, 7805

### Diodes
- **D1**: Full Wave Rectifier, CEN CBR1-100
- **D2**: Exopy Rectifier, 3A, 400 PIV (Radio Shack)

### Transformers
- **T1**: Pulse Trigger Transformer
- **T2**: (Not Shown) 120VAC to 230 VAC, 0.050 A
II.1 Specifications:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Reliable Strobe Energy</td>
<td>2.26 Joules at 10 Hz (90 ms charge time)</td>
</tr>
<tr>
<td>Maximum Power Consumption at 10 Hz</td>
<td>23.3 Watts</td>
</tr>
<tr>
<td>Light discharge time at 90 ms</td>
<td>500 us</td>
</tr>
<tr>
<td>Minimum Reliable Strobe Energy</td>
<td>0.052 Joules at 10 Hz (1.1 ms charge time)</td>
</tr>
<tr>
<td>Light discharge time at 1.1ms</td>
<td>74.4 us</td>
</tr>
<tr>
<td>Maximum Reliable Frequency</td>
<td>48 Hz at 1.6 ms charge time</td>
</tr>
<tr>
<td>Note: It can be run at 12 Hz with a 22 Joule strobe output (73 ms charge time)</td>
<td></td>
</tr>
</tbody>
</table>

III Power Consumption Testing:

The desired test results were the minimum and maximum reliable light energy emitted, the associated charge times, and the strobe power consumption. Also of interest was the experimental time constants of the two strobe charge capacitors, and the resulting internal capacitor resistance. Most experiments were run at a 10 Hz strobe rate. These values are compared to a simplified circuit model.

III.1 Setup and Procedure:

III.1.1 Minimum and Maximum Charge Times: The IVU's external illumination pin (pin 12) was used to vary the charge time of the strobe capacitors. This pin's signal was combined with the camera's 5 volt source, a 4.7 kilohm resistor, an optoisolator, a hex Schmidt trigger and software timing to produce a pulse width modulated signal (Figure 2). The optoisolator provided protection from the high voltage of the strobe circuit. The Schmidt trigger was used to produce a clean rising edge which was not generated otherwise. The acp.c C "case" statement used to generate the PWM signals is provided in Appendix A.

Using software to control the timing, different pulse modulated signals were generated by the camera. To determine reliability, the picture taken by the CCD was analyzed for a gradient which is successfully produced if the strobe is fired. For each case, the strobe was fired 256 times in succession. The light energy was determined by using the maximum and minimum charging times with the experimental capacitor time constants.

III.1.2 Experimental Charging Capacitor Time Constants and Internal Capacitor Resistance: The charging capacitor time constants were determined by measuring the voltage across the capacitors using an oscilloscope. (Note: the Mosfet transistor which provides charging time control was removed to eliminate the floating ground problem). No light illumination control was needed for this portion of the test.

The time constant was determined by measuring the time for the capacitor to reach 63% of the maximum voltage value. The internal resistance was computed by assuming two first order systems and determining the value of internal resistance which would yield the time constants obtained experimentally. Figure 3 shows the simplified circuit used for computational analysis.

III.1.3 Strobe Discharge Time: The strobe discharge time was measured using a phototransistor and an oscilloscope. Using the circuit shown in Figure 4, the illumination discharge time for the maximum and minimum charge times were experimentally determined.

III.2 Results:
The minimum charge time for a 10 Hz strobe rate was determined to be 1.1 ms with perfect reliability for 256 continuous flashes. The maximum charge time at the same flash rate was 90 ms. The experimental time constants were 112 us and 18.9 ms for the 1 uF and 40 uF capacitors, respectively. Both voltage curves showed a good compliance to a first order model in the low time range. However, the 1 uF capacitor's voltage curve makes a transition at 290 volts to a curve more closely representing a higher time constant system (Figure 5). This transition is due to the selection of resistors used.
IVU SETUP TO PRODUCE STROBE PWM SIGNAL

Figure 2:
Figure 3: Simplified Circuit

Figure 4: Phototransistor setup to measure strobe light duration
Figure 5: Charging curve of 1 uF Capacitor
The charge time and experimental time constants were used to compute the minimum and maximum theoretical power consumption of the strobe circuit. This value can be used to determine the strobe energy output for a given efficiency value. The minimum and maximum single flash energy consumption were computed to be 0.052 Joules and 2.26 Joules, respectively. The gives a maximum power consumption at 10 Hz of 23.3 Watts (Appendix B).

Using the experimental capacitor time constants and the simplified first order models, the internal resistance of the capacitors are estimated (Appendix C). An effective resistance in series with each of the capacitors was computed. The 1 uF capacitor’s internal resistance was calculated to be 12 ohms. The 40 uF capacitor had an internal resistance of 33 ohms.

The strobe discharge time curves measured by the phototransistor are shown in Figure 6. These curves show the phototransistor becomes unsaturated after 74.4 us when the strobe charge time is 1.1 ms. At charge times between 77 and 90 ms, the unsaturation point is 500 us. The first order rising portion of the curve is believed to be due to hardware. In order to determine the strobe light discharge time either extrapolation is necessary to determine when the first order curve begins its ascent. If the floating ground was not a problem the best way to determine the approximate light discharge time would be by using the oscilloscope to measure the capacitor’s voltage.

III.3 Conclusion:
From these experiments several characteristics of the strobe circuit were determined and can be used in other calculations. The strobe energy consumption was observed to vary reliably from 0.052 Joules to 2.26 Joules. A difference of 50 times the lower energy output. It is noted that 4 time constants of the system produces 98.1% of the maximum energy output or 2.21 Joules. This corresponds to 77 ms which allows a strobe flash rate of ~12 Hz for 98% of the maximum energy output. However, the maximum power consumption rises to 27 W. It is noted that the actually light energy is smaller than the computed energy consumption due to inefficiencies. As expected, the strobe rate increases if the charge time is decreased. For a 1.6 ms charge time, a 48 Hz strobe frequency was reliably realized. However, computer computation time would begin to be factor at these higher frequencies. This experiment used only delay commands with minimum computations. It is also noted that the minimum charge time of 1.1 ms is 10 times larger than the 112 us time constant of the smaller capacitor. It is believed that the extra time is actually needed to rise the voltage of the trigger capacitor to the necessary value instead of the strobe discharge capacitors.

The strobe discharge times produced by the phototransistor give a good maximum basis for the needed exposure time when using the strobe at different energy levels. However, it is believed the times are the result of saturation and charge dissipation of the phototransistor instead of the strobe light itself. Since exact synchronization is not likely, some extra time should always be provided.

The experimental time constants and resistor values will be used in future experiments which will try to use the CCD sensor to determine the actual light output and the efficiency of the strobe. These values will also be used in future tests for correlating image characteristics with strobe light output.

IV Strobe Light Energy Testing:
The test was conducted to determine the actual range of light energy values achievable using the computer controlled strobe circuit designed and build for the DOE project. This is important because resistors and other circuit component dissipate energy by generating heat. The actual power consumption of the circuit was determined in a previous test to be controllable from 0.5 to 23 Watts for a 10 Hz signal. However, the actual light flux generated by the strobe is more suitable for other light energy calculation needed later. Also, the strobe efficiency indicates a benchmark for improvements in efficiency for more energy conscious projects.

IV.1 Equipment:
Figure 6: Light discharge Time Curves

Phototransistor Response Curve to 1.1 ms Charge Time

Phototransistor Response Curve for 77 ms Charge Time
The measurements were taken with the CCD sensor used in the integrated vision unit. The important characteristics of the CCD sensor are given below (not all are used for test calculations).

**CCD Sensor** TC211 Texas Instruments

- **Sensitivity** 280 mV/lux
- **Size** 2640 um x 2640 um
- **Pixel size** 192 (H) pixels by 165 (V) pixels
- **Saturation** typically 450 mV with antiblooming (min 350 mV)
- **Saturation** typically 600 mV without antiblooming (min 400 mV)
- **Dark Signal** typically 10 mV (max 15 mV)
- **Charge converted to signal** @ 4 eV per electron

CCD Spectral Responsivity Curve (Figure 1).

The camera optics setup was a 1 mm lens diameter (opening) and a 12 mm focal length.

### IV.2 Setup and Procedure:

The strobe light energy was determined by measuring the light energy captured by the CCD. To minimize the incident angle, the strobe light was attached to the vision system just above the CCD head. A signal sent by pin 12 of the vision system controlled the charge time and strobe trigger. A 100 us CCD exposure time was used in the experiment. A white diffuse surface was held between 0.295 and 0.93 meters from the camera and strobe system.

Many assumptions were made to simplify calculations. The strobe tube was assumed to be a light source whose intensity varied with the angle cosine from the normal direction instead of the normal point source assumption. All light was assumed 555 nm in wavelength. This assumption allows the use of the mV/lux conversion given by the CCD manufacture and also allows a lum/W conversion of 683. The white paper reflectivity was assumed 1. The incident angles were assumed small such that the cosine terms become 1. Zero and 255 pixel values were assumed to correspond to the dark signal of 10 mV and the typical saturation voltage of 600 mV. This yields a conversion from pixel value to voltage of $\Delta mV = 2.314 \times A(pixel\_value)$. The $A(pixel\_value)$ is the change in pixel value from no strobe light present (caused by ambient light) to when the strobe light is used for the same view.

Given the above assumptions, the conversion from pixel values to light flux was determined (Appendix D). The energy, in terms of W/m², is $E = \pi \times L \times \cos(a_2)/(4 \times (f\_stop)^2)$, where $L$ is the brightness, $a_2$ is the incident angle and $f\_stop$ is the focal length over the lens diameter. Brightness, $L$, is $L = R \times E \times \cos(a_1)/p$, where $R$ is the reflectivity of the illuminated surface, $a_1$ is the incident angle, and $E$ is the illumination of a point source on a surface given by $\Phi/(4 \times \pi \times r^2)$. Due to the light source assumption this value becomes $E/1.57$. The $E$ is the light flux which we are trying to find, and $r$ is the distance from the light source.

The energy on the detector is also given by the pixel value and the CCD parameters. As previously mentioned, the pix value can be converted to mV. The sensitivity of the detector at 555 nm of 280 mV/lux converts volts to lux (lum/m²). One lumen at 555 nm is equal to 1/683 Watts. Thus, conversion from pixel value to W/m² becomes $(2.314 \times (pix\_value - ambient\_value))/(280 \times 683)$.

Using these two equations for W/m² on the detector, the light source flux (strobe light) is $\Phi = [2.314 \times (pix\_value - ambient\_value) \times \pi \times 16 \times r^2 \times (f\_stop)^2] / [1.57 \times 280 \times 683 \times R \times \cos(a_1) \times \cos(a_2)]$. With the assumed reflectivity of 1 and the incident angles of 0, along with the fixed camera optics of 12 f-stop, the equation reduces to a function of the papers distance from the strobe and the CCD average pixel value, $\Phi = 0.0558 \times (pix\_value - ambient\_value) \times r^2$. Since the strobe time and integration times are build into the equation, $\Phi$ really represents the energy in Joules from the strobe on the detector. In order to get reliable results, most of the pixels corresponding to the diffuse surface were averaged and 15 images were captured for each charge time. The distance from the light source was adjusted to capture all strobe energy values.
IV.3 Results:
Images were taken at 3 different locations; 930 mm, 600 mm and 295 mm. The calculated light energy graph shows the maximum light energy to be 1.63 Joules per flash at 930 mm with the current strobe setup and a 77 ms charge time (Figure 7). The minimum light energy was 0.0184 Joules for a 1.3 ms charge time at 295 mm distance. The graph also illustrates the model used to calculate the strobe light energy does not conform perfectly with the experiment since the energy values at different distance at the same charge time were not equal. The discrepancy is greatest at the longer charge times. For a 77 ms charge time there is a discrepancy of 50% between the light energy value calculated at 295 mm and 930 mm. This may be due to the angular effects that are accentuated at closer distances. Since most of the effects are smaller at larger distances, the 930 mm number should be most accurate.

The strobe efficiency curve shows the efficiency is approximately 70% for charge time greater than 12 ms(Figure 8). Efficiency is the measure of light energy compared with the amount of energy stored in the capacitors. The shape of the curves are similar for all distances. The efficiency falls rapidly below 70% for charge times below 12 ms to a minimum of 20% at 5 ms. Note, this effect actually increases the dynamic range of the strobe.
Figure 8: Strobe Efficiency Curve
The transient response of the strobe is seen in Figure 9. The strobe energy level output was observed to begin high and settle out to its lower steady state value after about 6 strobe flashes. A normalized curve is shown in Figure 10. Here it is seen for a 30 ms charge time the steady state value is 80% of the initial value. This effect is due to the full charged larger capacitor losing its initial power and not being able to supply the needed current the other capacitors want. The light output oscillated +/- 5% of its mean value at a 6 ms charge time. Note, the range in pixel values across the white diffuse image was ~8 pixel values.

Figure 9: Strobe Transient Response at 10 Hz and 295 mm
Testing also was performed to determine the amount of infrared light present in the strobe flash. Figure 11 shows the results for different charge times. Note this graph is not corrected for the nonlinear effects of the CCD head and the CCD head is most sensitive in the near infrared range. The amount of change in pixel value due to infrared light output by the strobe was approximately 47% for charge times greater than 15 ms seconds. This value dropped for lower charge times.
IV.4 Conclusion:
From the results it is seen that a point light source is not a good means to describe the strobe flash tube light intensity output. The cylindrical model used does appear to give more realistic results. The efficiency of the strobe is approximately 70% with 50% of the pixel value being generated by infrared light for charge time greater than ~15 ms. The transient response curve shows that the energy output decreases to a steady state value approximately 80% of its maximum value for a 30 ms charge time. Since the efficiency and light energy calculations were based on the steady state value, higher outputs and efficiencies could be obtained if the current could be supplied as needed. However, calculations should be based on the steady state values which will be present when the strobe is continuously strobed. It should be noted that depending on the power source this effect will be increased or decreased.

Future test plans include testing the strobe life which entails measuring the number of flashes until the light output is some percent of the initial value. Also, a complete light distribution test will be conducted to further the accuracy of the strobe characteristics.
Copy of 'Case Statement found in ACP.C file to generate PWM signal

```plaintext
case 'Q': /* generate a square wave for the strobe */
  /* vital information to make signal */
  /* note final signal is approximate */
  /* due to timing of commands and round off error */
  time_mult=1;
  num_flashes=ReadHexWord();
  /* note input as frequency */
  period=(1000000/ReadHexByte())-10000;
  /* in microseconds (-10000 for command times */
  /* note input as percent time on */
  micro_time=ReadHexByte();
  time_expon=ReadHexNibble();
  for (i=0;i<time_expon;i++)
    time_mult=time_mult*10;
  }
  time_low=(micro_time)*time_mult;
  time_high=period-time_low;
  strobe_count=0;
  Putchar(13);
  Putchar(10);
  for (i=0;i<num_flashes;i++)
    Delay(time_high);
    Picture(time_low, AntiBlooming, 100, LEDEnable);
    for (j=7;j<192;j++)
      if (((int)VideoRam[30][j+1]-int)VideoRam[30][j])>test_grad{
        strobe_count++;
        j=192;
        WriteHexByte(strobe_count);
        Putchar(13);
      }
  }
  Putchar(13);
  Putchar(10);
  break;
```

Appendix A
Appendix B

Energy & Power Calculations

Known:

\[ \begin{align*}
V_1 &= 100 \text{V} \\
V_2 &= 340 \text{V} \\
C_1 &= 1 \text{mF} \\
C_2 &= 30 \text{nF} \\
\text{Strobe Frequency} &= 10 \text{Hz}
\end{align*} \]

Minimum Energy & Power - 10ms Charge Time

\[
E_{\text{min}} = \frac{1}{2} C_1 V_1^2 + \frac{1}{2} C_2 V_2^2
\]
\[
= \frac{1}{2} C_1 (V_{\text{max}} (1 - e^{-\frac{100}{100}}))^2 + \frac{1}{2} C_2 (V_{\text{max}} (1 - e^{-\frac{100}{100}}))^2
\]
\[
= \frac{1}{2} \times 1 \times 10^{-6} (335 (1 - e^{-\frac{100}{100}}))^2 + \frac{1}{2} \times 40 \times 10^{-6} (335 (1 - e^{-\frac{100}{100}}))^2
\]
\[
\text{Since } C_1, V(t) \text{ curve is } \int \text{ let } V_1 = 300
\]
\[
= \frac{1}{2} \times 1 \times 10^{-6} (300)^2 + \frac{1}{2} \times 40 \times 10^{-6} (335 (0.056))^2
\]
\[
= 0.045 + 0.00717
\]
\[
E_{\text{min}} \approx 0.052 \text{Joules}
\]

Maximum Energy & Power - 90ms charge time

\[
E_{\text{max}} = \frac{1}{2} (41 \times 10^{-6} (1 - e^{-90 \times 10^{-6}}) 335)^2
\]
\[
E_{\text{max}} = 2.26 \text{ Joules}
\]

\[
\text{Power}_{\text{max}} @ 10 \text{Hz} = (2.26) \times 10^2 = 23 \text{W} = P_{\text{max}} @ 10 \text{Hz}
\]

\[
\text{Maximum Power for } 44 = 77 \text{ms} = 98\% \text{ Energy of system max}
\]
\[
\text{at } 12 \text{Hz charge was 78ms}
\]
\[
E = 2.21 J \quad W = 26.6 \text{W}
\]
Appendix C

Internal Capacitor Resistance Calculations

Assume two first order system with an unknown resistance due to each capacitor. Due to the large difference in time constants this is possible.

\[ R_c_1 = \frac{100 \Omega}{1 \text{mF}} \]

\[ 335 \text{V} \]

\[ \frac{1}{R_c_1} \]

\[ R_c_2 = \frac{440 \Omega}{40 \text{mF}} \]

\[ 335 \text{V} \]

\[ \frac{1}{R_c_2} \]

Known: Experimental Time Constants

\[ 1 \text{mF} \Rightarrow T_{c_1} = 112 \mu s \]

\[ 40 \text{mF} \Rightarrow T_{c_2} = 18.9 \text{ ms} \]

\[ 1 \text{mF}\text{ Capacitor} \]

\[ \text{theoretical} \ T_{c_1} = R_c C_1 = 100 \Omega \times 1 \text{mF} \]

\[ \text{experimental} \ T_{c_1} = 112 \mu s \]

\[ 112 \times 10^6 s = (R_c + R_c) C_1 = (100 + R_c) \times 10^{-6} F \]

\[ R_c_1 \approx 12 \Omega \]

\[ 40 \text{mF} \text{ Capacitor} \]

\[ \text{theoretical} \ T_{c_2} = R_c C_2 = 440 \times 10^{-6} \text{F} \]

\[ \text{experimental} \ T_{c_2} = 18.9 \text{ ms} \]

\[ 18.9 \times 10^3 s = (440 + R_c) C_2 = (440 + R_c) \times 10^{-6} \text{F} \]

\[ R_c_2 \approx 32.5 \Omega \]
Strobe Light Energy Calculations

Task: Approximate the strobe light energy from the grayscale value of a diffuse white surface image.

Assumptions:
1. Angles $\alpha_1, \alpha_2, \theta$ are small
2. The reflectivity of the diffuse white surface is 1
3. Optics are known to be 12-f-stop.
4. $r_1, r_2$ are equal
5. All light is at 555nm wavelength
6. CCD specs are accurate
   - Sensitivity: $280 \text{ mV}/\text{x}$ for 555nm
   - Saturation: 600 mV w/out Anti-blooming
   - Dark Signal: 10 mV typically

Solution:

Energy ($\text{W/m}^2$) on the detector can be represented by two equations:

(i) From CCD Detector parameters

$$E = f(\text{pixel value}, \lambda) \xrightarrow{\lambda=555\text{nm}} f(\text{pixel value})$$

Note: @ 555nm, the CIE relative luminosity function is $683 \text{ lum/w}$

thus, assuming a linear electrical response from the detector

and knowing

- 255 pixel value = 600mV
- 0 pixel value = 10mV
So the conversion from pixel value to mV is

\[
\begin{align*}
\text{mV} & \quad \text{pixel} \\
10 & \quad 285
\end{align*}
\]

Thus the Energy on the detector becomes

\[
E = \frac{2.314 \text{(pixel value)} \text{mV}}{280 \text{lux} \left( \frac{1 \text{ lum}}{1 \text{ lux}} \right) \left( \frac{683 \text{ lum}}{1 \text{ W}} \right)} = \frac{2.314 \text{(pixel)}}{(280)(683)} \text{ W/m}^2
\]

\[
E = f(\text{pixel value})
\]

(2) For an external point light source with \( \lambda = 555 \text{nm} \) and flux \( \Phi \) (Dickerson Handout)

(a) The brightness of a surface due to a point light source is

\[
L = \frac{(R \text{E}_\text{x})}{\pi r^2}
\]

where \( R \) is the reflectivity

\( \text{E}_\text{x} \), is the irradiance \( \text{W/m}^2 \)

and \( E = \text{cos} \alpha \)

\( \Phi = \text{flux (watts)} \)

\( I = \text{intensity (watts/steradian)} = \frac{\Phi}{4\pi} \)

Thus,

\[
L = \frac{R \Phi \text{cos} \alpha}{4\pi r^2}
\]

(b) The energy projected from a surface to the detector

The projected Area on Detector: \( dA_L = dA \cos \Theta \left( \frac{d_0^2}{d_0^2} \right) \text{cos} \alpha \)

The irradiance at the lens: \( E_2 = \frac{I \text{cos} \alpha}{(\frac{d_0}{\text{cos} \alpha})} \text{actual distance} \)

Intensity of the surface: \( I = dA \cos \Theta \cdot L \)

Area of Lens: \( A_L = \frac{\pi}{4} D^2 \)
Due to Conservation of Energy

\( E = \frac{w}{m^2} \) is energy at the detector

\[ A_x E_x = E \, dA_x \]

then,

\[ \frac{1}{4} D^2 \frac{dA \cos \theta \, L \cos^2 \theta_x}{d_0^2} = E \, dA \cos \theta \left( \frac{d_0^2}{d_0} \right)^{\frac{1}{2}} \cos \theta_x \]

Solving for \( E \),

\[ E = \frac{\pi L \cos \theta_x}{4 \, (\text{f-stop})^2} \]

where \( \text{f-stop} = \frac{\text{focal length}}{\text{lens diameter}} = \frac{d_0}{D} \)

Substituting in for \( L \)

\[ E = \frac{R \, \cos \theta_x \, \cos \theta_x \, \left( \frac{w}{m^2} \right)}{16 \pi \, r^2 \, (\text{f-stop})^2} = f(R, \theta_x, \theta_x, \theta, r, \text{f-stop}) \]

Combining the two equations for Energy \( \left( \frac{w}{m^2} \right) \) at the detector

From point source:

\[ \frac{R \cdot \cos \theta_x \, \cos \theta_x}{16 \pi \, r^2 \, (\text{f-stop})^2} = \frac{2.314 \, (\text{Apixel})}{280 \, (653)} \]

Since the CCD energy equation depends on the integration time, and the change in pixel value do strictly to the strobe can be determined. The time element factors out of the equation, Therefore, \( D \) represents the Joules of Energy from the strobe light.

\[ D = \frac{2.314 \, (\text{Apixel}) \, 16 \pi \, r^2 \, (\text{f-stop})^2}{280 \, (653) \, R \, \cos \theta_x \, \cos \theta_x} \]

However, it is known that the strobe tube is not accurately modeled by a point source with uniform illumination, instead it is believed a better model is cylinder with no light out the sides.

\[ \text{intensity} \propto \frac{1}{\text{dist}} \]

\[ \text{intensity} \propto \frac{1}{\text{dist}^2} \text{varies w/} \sin \theta \]
If the light flux from the sources are the same, then the integration of light intensity should be equal. Since both are symmetric about the axis $\psi$, we need to only integrate over part of the distribution

$$\int_0^{\pi} d\theta = \int_0^{\pi/2} R \sin \theta d\theta$$

$$\pi = 2R$$

$$\Rightarrow R = \frac{\pi}{2} = 1.57$$

This states that the intensity of light perpendicular to the strobe surface is 1.57 that of a point light source. Since, the camera's field of view is $\frac{\pi}{6}$.

$$\cos \frac{\pi}{6} = 0.996.$$ Therefore, we can approximate the light intensity as 1.57 that of a point source.

Thus, $I_c = 1.57 I_p$ which gives

$$\Phi = \frac{2.314 (\Delta \text{pixel}) 16 \pi r^2 (\text{f-stop})^2}{(1.57) 280 \times \cos \psi \cos \theta}$$

$$\Phi = f(\Delta \text{pixel}, r, \text{f-stop}, \psi, \alpha, \phi_2, \theta)$$

with the assumptions and knowns

$$\Phi = \frac{2.314 (\Delta \text{pixel}) 16 \pi r^2 (\text{f-stop})^2}{(1.57) 280 \times \cos \theta}$$

$$\Phi = 0.0558 (\Delta \text{pixel}) r^2$$
Analysis of Stepper Motor
Repeatability

A Report
by
Stephen E. Fuks
Graduate Research Assistant
Georgia Institute of Technology
George W. Woodruff School of Mechanical Engineering
**Objective:**

Experimentally determine the repeatability of measurements made with a machine vision system in which the CCD detector is mounted onto the spindle of a stepper motor for computer-controlled line-of-sight.

**Description of Experimental Setup:**

A target that consists of a rectangular diffuse white background with a solid black rectangle near the center was constructed. An off-the-shelf stepper motor with a step size of 1.8 degrees was mounted to the top of a DVT Stinger 70C Integrated Vision Unit (IVU); and the CCD housing and wiring was modified accordingly to allow the detector to be mounted on top of the stepper motor spindle and rotate freely as shown in figure 1. The IVU was fixed to a sturdy table top with a vise clamp to simulate being mounted atop an AGV.

This stepper motor and driver module require an external power supply that provides anywhere from 15V to 40V of direct current (DC). The Stinger IVU provides the TTL compatible signals to specify the direction of rotation (clockwise or counterclockwise) and to step the motor. An opto-isolating circuit is also used the IVU I/O to reduce the possibility of accidentally sending a signal to the camera that could damage its Motorola 68901 chip.

Illumination is provided by general purpose fluorescent lights mounted into the ceiling of the machine vision lab in the Manufacturing Research Center at Georgia Tech.

Borland C++ software to locate the position of a vertical and horizontal edge within an image to subpixel precision was developed, tested and downloaded into the camera. The software allows the user to pre-select a particular row to search for a vertical edge and a particular column to search for a horizontal edge. Other parameters to be adjusted by the user include the exposure time and the delay time between sending a pulse to the stepper motor and acquiring an image for processing.

**Description of Experiment:**

The target described in the previous section was positioned at a distance measured with a tape measure of approximately 83 inches from the 12mm focal length lens in the CCD optical unit. For this experimental setup, a qualitative assessment of the image indicated that a total exposure time (shift out plus normal integration time) of about 20ms provided good contrast between the black and white portions of the target in the acquired image. Testing of the system with no power applied to the stepper motor driver, referred to as the baseline configuration, was performed.
to validate this assumption.

An image was acquired and the subpixel locations along row 35 for the left vertical edge and along column 60 for the top horizontal edge of the black box on the white background were computed to subpixel precision. This process was repeated 100 times and the results were saved to a file.

The stepper motor was then powered and pulsed 10 steps clockwise and 10 steps counterclockwise between each image acquisition. Edge position data was collected using the same algorithms applied to the baseline test. A number of different tests were performed with different values for the delay period between stepper motor motion and image acquisition.

Results:

Statistical analysis of the unpowered (baseline) configuration data yielded standard deviations of .0104 pixels for the position of the vertical edge and .0209 pixels for the position of the horizontal edge.

Experiments with different delay cycles between the end of the last pulse sent to the motor and the image acquisition indicate that the experimental optical unit undergoes a torsional vibration as a result of motor motion. By increasing and increasing the delay time, the vibration amplitude dies out due to structural damping and the repeatability of the measurements increases as demonstrated by the data sets for two different delay values given in the appendix. These results highlight the need to minimize the appearance of vibration in the design and construct of the optical unit that is mounted onto the stepper motor spindle.

Using the largest delay results as a measure of the stepper motor repeatability indicates that estimating the position of a fiducial by using one edge crossing would have a standard deviation of .0694 pixels for a vertical edge and .0647 pixels for a horizontal edge. Using a 12 mm lens, each column pixel in the camera encompasses a field of view of 12.550/200 pixels = .0628 pixels.

Using 3σ repeatability data indicates that a landmark which is located by estimating the position of a single edge crossing with one camera has a resolution of 3*.0694*.0628 = .01310. For a vehicle located halfway down a 180 foot aisle, such a tolerance trigonometrically yields a maximum lateral deviation of .49 inches.

Future work:
Minimization of vibration induced by stepper motor motion must be investigated to minimize its effect on edge position estimations. Sensitivity of measurements to variations in illumination uniformity, intensity levels and vehicle motion must also be minimized.

Fiducial design to satisfy the positioning accuracy requirements can then be based on the repeatability measurements of the resulting system based on the following:

If the lateral position of a fiducial is based on two independent measurements with Gaussian distributions, \( u \) and \( v \). Then the position estimate, \( x \), can be given by \( x = au + bv \); and the associated variance in \( x \) is given by \( q_x^2 = a^2 q_u^2 + b^2 q_v^2 \). In our case \( q_u^2 = q_v^2 \), so \( q_x^2 = (a^2+b^2)q_u^2 \). If \( a=b \), this simplifies further to \( q_x^2 = (2a^2)q_u^2 \). If a simple averaging between the edge positions is employed, then \( a=b=\frac{1}{2} \), then the standard deviation in \( x \) is now \( \frac{1}{\sqrt{2}} \) times the standard deviation of the individual measured edge position.
Figure 1
10 steps with delay = 300000

standard dev = 0.08174

standard dev = 0.1762
10 steps with delay=500000

**standard dev = 0.06475**

**standard dev = 0.06942**

<table>
<thead>
<tr>
<th>col_pixel pos. vertical edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>66</td>
</tr>
<tr>
<td>65.8</td>
</tr>
<tr>
<td>65.6</td>
</tr>
<tr>
<td>65.4</td>
</tr>
<tr>
<td>65.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>row_pixel pos. horizontal edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
</tr>
<tr>
<td>21.8</td>
</tr>
<tr>
<td>21.6</td>
</tr>
<tr>
<td>21.4</td>
</tr>
<tr>
<td>21.2</td>
</tr>
</tbody>
</table>

# of measurements
MULTI-BODY SIMULATOR
USER’S GUIDE

Paul R. Schmitt
John E. Hogan
Wayne J. Book

Georgia Institute of Technology
The George W. Woodruff School of Mechanical Engineering
Atlanta, Georgia

Sponsored by
The Westinghouse Savannah River Plant

March 31, 1994
**Table of Contents**

- Introduction 1
- Joint Paradigm 1
- Sensors 1
- Map 2
- Dynamics 3
- Using MBSim: Script Files 3
  - Describing a Mechanism 3
  - Graphics Description 8
  - Sensor Description 10
  - Map Description 14
  - Configuration Description 15
  - Velocity Description 16
  - Position Description 17
  - Environment Description 18
  - Graphics View 20
- Force–Torque Plots 20
- Sample Program 21
- Continuing Developments 24
- References 25
INTRODUCTION – What is a Multi-Body Simulator?

A Multi-Body Simulator (MBSim) is an object-oriented system that models, simulates, and animates the kinematics and dynamics of robotic arms and vehicles. This system creates a three dimensional graphical environment which can be used as a powerful tool in robotic design and control. In addition, MBSim incorporates rangefinder as well as ultrasonic sensor model to yield environmental feedback.

In fact, a motivation for developing MBSim stemmed from the realization that most commercially available packages are not as flexible as is often needed. For example, while it is rare for a typical system to handle industrial robots AND vehicles, it is quite uncommon for them to be able to handle sensors as well as a mechanism’s kinematics and dynamics.

To give you an idea of where MBSim is heading, our ultimate goal for MBSim is studying safe mechanism motion through sensor-based motion control. We will be testing new motion planning and control algorithms on numerous mechanisms in their typical environments. This is especially important for vehicles with car-like steering in highly constrained environments (e.g., parallel parking a Cadillac).

This document gives an overview of MBSim’s capabilities and describes in detail how one utilizes these capabilities. First an explanation of our joint paradigm, or mechanism construction, is given. From there we give an overview of sensors, sensor data storage array (i.e., map), and dynamics in MBSim. Following this, there is a detailed description of what script files are and how a user formulates the script files for an application. Then an explanation of how the user can determine the kinds of forces and torques needed for a desired motion (displaying force–torque plots) is given. An example program is given and explained followed by a description of additional developments currently being worked upon.

JOINT PARADIGM

In MBSim, each mechanism is constructed as a series of links. Each link incorporates the joint kinematics that connect it to the preceding link. In other words, each link "knows" how to move itself with respect to the previous link. This general joint paradigm allows holonomic and non-holonomic constraints to be treated similarly. This paradigm also facilitates modeling and simulation of the kinematics and dynamics of multibody mechanisms with a wide variety of link types. The links contain graphic objects (See GRAPHICS DESCRIPTION below) which describe the physical shape of the link and sensors which provide feedback of the environment.

SENSORS

MBSim incorporates an idealized rangefinder and ultrasonic model. When activated, the Rangefinder sensor determines whether the objects’ enclosing spheres lie on the sensor’s line of sight. Ultrasonic determines whether the sphere intersects its cone projected from the sensor. To gain a more accurate range that takes the sensed objects geometric characteristics into account, ray tracing techniques are being implemented. Therefore the enclosing sphere’s method will be
used as a rough scan that returns objects for further calculation. This two-stage process reduces computational overhead.

Through a sensor description file, the user places any number of sensors in any position or orientation on any link. See SENSOR DESCRIPTION for more details.

**MAP**

Map is an intelligent range data storage array [1]. It is designed to "move" along with the mechanism, automatically storing sensor data (Rangefinder and Ultrasonic) from pinged sensors. Range data is added to previous data. In this way collective range data is held to yield a good estimation of obstacle locations in the environment. The user tailors the size of a map and the number of sensors utilized as needed through a map description file. See MAP DESCRIPTION for more details.

Below, in Figure 1, is a sample output from all 24 of SWAMI's sensors as it moves down a narrow aisle of pallets. SWAMI's middle is located at (19,19) and it is facing to the right. Note the large values (i.e., 6, 8, 10,11) in the cells to SWAMI's immediate left and right. These values indicate duplicating sensor range data, and a high certainty level of obstacle presence in those cells.

![Sample Map output](image)

**Figure 1 – Sample Map output**
Under development is a Path Planner that generates a feasible path avoiding obstacles based upon information located in Map.

DYNAMICS

We are currently controlling velocity driven mechanisms. The inverse dynamics of these mechanisms yield the forces and torques that each link's actuator would need to generate motion based on each link's positions, velocities, and accelerations. This is useful to determine whether the desired velocity commands are feasible for a particular actuator. The inverse dynamics of the system is calculated using an iterative Newton–Euler dynamic formulation.

For a mechanism driven by force/torque commands or a low level controller, the forward dynamics of the system is required. Although this is currently not implemented on MBSim, it is readily implemented via the iterative Newton–Euler dynamic formulation.

Each link has a consistent velocity and acceleration interface which enables only one function at the mechanism level to calculate the dynamics for each link in any mechanism in any configuration. By embedding the specific characteristics of a link inside the link class and using consistent interfaces to each link, general functions can be used to yield desired quantities regardless of the mechanism being studied.

USING MBSim: SCRIPT FILES

The MBSim user describes a new mechanism through script files. These are simple text files that are read by MBSim during program execution. In this way a vehicle can be built up or modified without having to recompile the source code. Thus the main key to using MBSim is mastering these script files. Currently script files are used to describe the mechanism's joints, mass properties, physical appearance, sensor type and placement, initial configuration, joint velocities, and environment.

One creates and uses a new simulation in the following manner. First one must set up script files that describe the mechanism, sensors, initial configuration, velocity, environment, and the graphics view. Then the main() should be written. This code reads the script files, drives the mechanism, and activates the sensors (See SAMPLE PROGRAM).

All dimensions are in millimeters, angles in radians, mass in grams, and time in seconds. Another consistent set of dimensions can be substituted. Commas should not be used between values.

DESCRIBING A MECHANISM

A mechanism is defined with a mechanism description file. In MBSim, each mechanism is constructed as a series of links. And the user defines the parameters, derived from Denavit–Hartenberg convention [2][3], for each link and its associated joint. Note that a link is currently assumed to have only one joint. These parameters are followed by a graphics description filename and an optional sensor description filename and a list of sensor identification numbers. These
numbers will be used to activate the sensors they refer to.

Mechanism description file elements are now described.

Mechanism Description Files (.md) (Describes and builds Mechanism)

The first line contains the name of the mechanism. Then the list of joints and their parameters. Available joints are:

**Revolute Joint (R):** alpha, a, d, theta (D–H parameters Fig. 2)
Standard rotary joint found on most industrial robots.

**Prismatic Joint (P):**
Telescopic joint.

**Unactuated Joint (U):** alpha, a, d, theta (D–H parameters)
Often used to represent a tool in a manipulator's gripper.

**Knife Joint (K):** x,y,z, theta
Can be thought of (with Dual Wheel and Tricycle) as a joint between the mechanism and the ground. moves similar to a pizza cutter: forward, backward, or rotational motion is allowed, but not sideways. (See Figure 4)

**Tricycle Joint (T):** x,y,z, theta (link), phi (wheel), length
Used for modeling forklifts, automobiles, and MoRT (See Figures 5 and 8). For the turning wheel, an additional graphics description file "Wheel.gd" is needed. In Figure 8, "Wheel.gd" contains the description of two thin horizontal cylinders (See GRAPHICS DESCRIPTION).

**Dual Wheel Joint (D):** x,y,z, theta, wheel_radius, axle separation
modeled after SWAMI: a mechanism driven by two parallel motorized wheels (See Figures 6 and 7).

After the joint specification and its parameters add the following:

graphics filename, sensor filename (if any), sensor numbers (if any),
mass, center_of_mass vector (x, y, and z values), inertia matrix (3x3)

If there are no sensors put "none" in place of sensorfilename and sensor numbers.
The center of mass vector is from the joint's origin to its center of mass.

**Example 1: Mechanism description – SWAMI: 24 sensors, estimated mass properties (Fig. 7)**

SWAMI is modeled as a Dual Wheel joint with an initial location of (−850, −1850, 0) in the environment at an angle of 1.595 radians from the environment x–axis. Wheel radius and axle separation are 152 mm and 406 mm, respectively. SWAMI’s
Figure 2 – Revolute joint parameters

Figure 3 – Prismatic joint parameters

Figure 4 – Knife joint parameters

Figure 5 – Tricycle joint parameters

Figure 6 – Dual Wheel joint parameters
graphics description is found in the file "SWAMI.gd." SWAMI's sensor description filename is followed by a list of sensor identification numbers: one for every sensor described in SWAMI.sd. SWAMI's mass is estimated to be 100000 g. The center of gravity is located (from the midpoint between the wheel centers) 15 mm toward the front, 20 mm towards the right wheel, and 25 mm up. The inertia matrix [3] is given by a 3 x 3 matrix. Note: the mass properties shown below are only place holders and are not realistic.

Example 2: Mechanism description – MoRT (Fig. 8):
MoRT is modeled as a Tricycle joint (base) followed by a series of Revolute joints (Shilling arm). The midpoint of the rear axle (origin of Tricycle joint) starts at the origin. The body starts at zero radians with respect to the x–axi). The turning wheel ("Wheel.gd") starts at zero degrees with respect to the body. The distance between the front and rear axle is 1108 mm. MoRT's graphics description is contained in "Mort.gd." The base has a mass of 100,000 g and a center of gravity at the midpoint of the rear axle.

The Shilling arm has a Revolute joint base located 128 mm back and 914 up from the Tricycle origin. It rotates in the horizontal plane. It is initialized pointing toward the mechanism rear (3.1415 rad from a vector pointing forward on the vehicle).

Note: the mass properties shown below are only place holders and are not realistic.
0 0 10
R 0 927 0 -.4 ShillingElbow.gd
10 0 0 0
10 0 0
0 10 0
0 0 10
R 0 508 0 0 ShillingWrist1.gd
10 0 0 0
10 0 0
0 10 0
0 0 10
R -1.5708 0 0 0 ShillingWrist2.gd
10 0 0 0
10 0 0
0 10 0
0 0 10
U 0 330 0 0 ShillingHand.gd
10 0 0 0
10 0 0
0 10 0
0 0 10

Figure 7 – SWAM1

Figure 8 – MoRT
GRAPHICS DESCRIPTION

All mechanism dimensions (e.g., the Shilling arm graphics descriptions above) and environmental objects must be approximated by a suitable number (determined by the user) of graphics primatives or elements in a graphics description file. Currently available elements for this file are Plane, Rectangular Prism, Frustum, Cylinder, and Cone.

Graphics Elements and parameters

<table>
<thead>
<tr>
<th>Plane(PI):</th>
<th>x, y, color number, transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>(origin at center)</td>
<td></td>
</tr>
<tr>
<td>Rectangular Prism(RP)</td>
<td>x, y, z, color number, transform</td>
</tr>
<tr>
<td>(origin at center)</td>
<td></td>
</tr>
<tr>
<td>Cylinder(C):</td>
<td>radius, height, number of side faces, color number, transform</td>
</tr>
<tr>
<td>(origin at center, height along z-axis)</td>
<td></td>
</tr>
<tr>
<td>Frustum(Fr):</td>
<td>length, basewidth, base height, endwidth, endheight, color number, transform</td>
</tr>
<tr>
<td>(origin at center, length along x-axis)</td>
<td></td>
</tr>
<tr>
<td>Cone(C):</td>
<td>one radius, height, number of faces, color number, transform</td>
</tr>
<tr>
<td>(origin at center, height along z-axis)</td>
<td></td>
</tr>
</tbody>
</table>

Color Table*

<table>
<thead>
<tr>
<th>Color</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>0</td>
</tr>
<tr>
<td>white</td>
<td>1</td>
</tr>
<tr>
<td>dark_grey</td>
<td>2</td>
</tr>
<tr>
<td>grey</td>
<td>3</td>
</tr>
<tr>
<td>light_grey</td>
<td>4</td>
</tr>
<tr>
<td>brown</td>
<td>5</td>
</tr>
<tr>
<td>light_blue</td>
<td>6</td>
</tr>
<tr>
<td>light_green</td>
<td>7</td>
</tr>
<tr>
<td>yellow</td>
<td>8</td>
</tr>
<tr>
<td>light_yellow</td>
<td>9</td>
</tr>
<tr>
<td>red</td>
<td>10</td>
</tr>
</tbody>
</table>

*Can be expanded, if needed

Graphics transform consists of a rotation matrix and a position. The rotation is about the origin of the Graphics_Element. Taking three consecutive rotations about a fixed set of axes yields the general rotation matrix form [3]:

\[
\begin{bmatrix}
\alpha \cos \beta & \alpha \sin \beta \sin \gamma - \alpha \cos \gamma & \alpha \sin \beta \cos \gamma + \alpha \sin \gamma \\
\lambda \sin \beta & \lambda \sin \beta \cos \gamma + \lambda \cos \beta \sin \gamma & \lambda \sin \beta \sin \gamma - \lambda \cos \beta \cos \gamma \\
-\gamma & \gamma \cos \beta & \gamma \sin \beta \\
\end{bmatrix}
\]
where $\alpha$ represents the $\cos(\alpha)$, $\alpha$ represents $\sin(\alpha)$, etc. The rotations must be about the X axis first ($\alpha$), then Y ($\beta$), then Z ($\gamma$). This matrix simplifies greatly for the degenerate cases.

The position is a vector from the joint origin (depends on joint type) to the element center.

Example: Transform – no rotation (identity matrix) and [1 2 3] position vector

\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
\]  

$\leftarrow$ Rotation matrix (identity)

\[
\begin{pmatrix}
1 & 2 & 3 \\
\end{pmatrix}
\]  

$\leftarrow$ Position vector (x y z) in mm

Example: Graphics Description file – SWAMI modeled with 4 Rectangular Prisms

(See SWAMI Fig. 7)

RP  203 711 1372 6  

\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
\]  

$\leftarrow$ Rectangular Prism of 203 by 711 by 1372 mm  

color 6 (light_blue)

483 0 686  

$\leftarrow$ Position vector from joint origin to RP center

RP  762 711 356 6

\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
\]

$\leftarrow$ Identity rotation matrix

0 0 178  

$\leftarrow$ Position vector from joint origin to RP center

RP  813 711 1346 6

\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
\]

-25 0 1029  

$\leftarrow$ Position vector from joint origin to RP center

RP  254 711 102 6

\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
\]

712 0 51  

$\leftarrow$ Position vector from joint origin to RP center
SENSOR DESCRIPTION

Sensor description files are used to attach sensors to links. A sensor graphic element is automatically placed on the mechanism at the specified position and orientation.

Currently, the sensor range is "hard coded" at 10m.

There needs to be a one-to-one correspondence between the sensors in this file and the sensor identification number list in the mechanism description file. This format was chosen so that a single sensor description file could be used for different links. Note that sensor placement does not depend upon the geometry of link. It is therefore possible to place sensors inside a mechanism or "off in space."

Sensor Description Files (.sd)

- Rangefinder:
  - Rotation matrix (3x3)
  - Position vector

- Ultrasonic:
  - Ultrasonic, ultrasonic cone angle in radians
  - Rotation matrix (3x3)
  - Position vector

Example 1: Sensor Description file – SWAMI 24 sensor array numbered 1 to 24 consecutively, cone angle of .1 radians (See Fig. 9). All sensors are included for reference.

 Ultrasonic .1
    1 0 0
    0 1 0
    0 0 1
    584 279 254
    <- Ultrasonic sensor with cone angle of .1 radians
    "Ultrasonic" must have a capital U.
    <- Identity rotation matrix
    <- Position vector from link center to sensor center

 Ultrasonic .1
    1 0 0
    0 1 0
    0 0 1
    584 -279 254

 Ultrasonic .1
    .965926 .258819 0
    -.258819 .965926 0
    <- Rotation matrix, -15 degrees about Z-axis
    (0 degrees about X and Y)
0 0 1
584 279 406

Ultrasonic 1

0.965926 -0.258819 0
0.258819 0.965926 0
0 0 1
584 -279 406

Ultrasonic 1

0.965926 -0.258819 0
0.258819 0.965926 0
0 0 1
584 279 559

Ultrasonic 1

1 0 0
0 1 0
0 0 1
584 279 864

Ultrasonic 1

1 0 0
0 1 0
0 0 1
584 -279 864

Ultrasonic 1

0.130527 0.991445 0
-0.991445 0.130527 0
0 0 1

483 -355 299

Ultrasonic 1

<-Rotation matrix, -82.5 degrees about Z axis
Figure 9– SWAMI sensor identification numbers.
.5 -.866025 0
.866025 .5 0
0 0 1
483 355 692

Ultrasonic .1

.965926 .258819 0
-.258819 .965926 0
0 0 1
483 –355 806

Ultrasonic .1

.965926 -.258819 0
.258819 .965926 0
0 0 1
483 355 806

Ultrasonic .1

0 1 0
-1 0 0
0 0 1
711 –279 152

Ultrasonic .1

.707107 .707106 0
-.707106 .707107 0
0 0 1
711 –279 152

Ultrasonic .1

1 0 0
0 1 0
0 0 1
711 –279 152

Ultrasonic .1

1 0 0
0 1 0
0 0 1
711 279 152

Ultrasonic .1
.707107 -.707106 0
.707106 .707107 0
0 0 1
711 279 152

Ultrasonic .1
0 -1 0
1 0 0
0 0 1
711 279 152

Example 2: Sensor Description file – Various sensor examples

Rangefinder

<-Rangefinder sensor, capital R, the rest must be lower case

.707107 -.707106 0
.707106 .707107 0
0 0 1
10 20 30

Ultrasonic .212

<-Ultrasonic sensor with cone angle of .212 radians

0.852869 -0.150384 0.5
0.331588 0.895721 -0.296198
-0.403317 0.418412 0.813798

987 654 321

MAP DESCRIPTION

The user has the ability to tailor sensor range data output as needed. Map dimensions, scale, initial value, and sensors can be adjusted. Map is currently displayed in a text window. Therefore, the map dimensions should reflect your output window size.

A small map with long sensor readings is impractical. Therefore adjust the scale to fit in most of the readings. The scale value is the side dimensions of a square cell.

The user specifies which sensors will update this map by listing their identification numbers (see MECHANISM DESCRIPTION). This is useful if it is desired to have maps for different heights (e.g., high, medium, low).
Some useful map commands are UpdateMap and ClearMap. UpdateMap() activates the sensors as given in the map description file and adds the data to the array. ClearMap(num) wipes the map clean and inserts the value "num" into each cell. This is currently necessary because Map has no memory: it has no idea which way the vehicle is moving.

================================================================================================

Map Description Files (.mp)

map width, map height
range scale, initial fill value
mechanism name, sensor ID number list

Example: Map Description File – SWAMI.mp

68 50 <-Map width of 68 numbers by 50 high
100 1 <-Cell is .1 m by .1m. Initial cell value is 1 for all cells

SWAMI 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
All 24 sensors on SWAMI will be read

================================================================================================

CONFIGURATION DESCRIPTION

This is an optional script file which defines the initial configuration (time = 0) of the mechanism. This is useful when starting a mechanism from a configuration different from that which was developed in the mechanism description file. In this way, the mechanism description file does not need to be redone with each new configuration.

================================================================================================

Configuration Files (.cfg) (joint values) (Use for initial conditions)

Revolute Joint (see Fig. 2): θ, position, θ, velocity, θ, acceleration, joint torque

Prismatic Joint (see Fig. 3): d, position, d, velocity, d, acceleration, joint force

Unactuated Joint: (none)

Knife Joint (see Fig. 4): x, y, θ, position, body velocity, θ, velocity, body acceleration, θ, acceleration, force, torque

Tricycle Joint (see Fig. 5): x, y, θ, position, φ, body velocity, φ, velocity, body acceleration, φ, acceleration, force, torque

*Dual Wheel Joint (see Fig. 6): x, y, θ, position, φ, velocity (right wheel), φ, velocity (left wheel), φ, acceleration (right wheel), φ, acceleration (left wheel), torque (right wheel), torque (left wheel)
* \( \phi \) for Dual Wheel is the angle of rotation of a motorized wheel.

Example: Configuration file – Tricycle with one revolute joint
Tricycle is initialized at no velocity or acceleration with the middle of its rear axis on the origin. Its main body axis lies along the x-axis. The steering wheel is straight.

```
0 0 0 0 0 0 0 0 <- Tricycle joint parameters
0 0 0 0 <- Revolute joint parameters
```

VELOCITY DESCRIPTION

Motion of mechanism joints is currently handled by a velocity file. Work is underway on a controller that provides joint velocities. Velocities are linearly interpolated over the time interval so acceleration is constant.

---

**Velocity Files (.vel)**

Note: Each line contains all Mechanism velocity values with the first value on each line being time.

- **Revolute Joint (see Fig. 2):** \( \theta_{velocity} \)
- **Prismatic Joint (see Fig. 3):** \( d_{velocity} \)
- **Unactuated Joint:** (none)
- **Knife Joint (see Fig. 4):** body velocity, \( \theta_{velocity} \)
- **Tricycle Joint (see Fig. 5):** body velocity, \( \phi_{velocity} \)
- **Dual Wheel Joint (see Fig. 6):** \( \phi_{velocity} \) (right wheel), \( \phi_{velocity} \) (left wheel)

---

Example: Velocity file – MoRT with 5 DOF Shilling Arm

```
0.5 50 0 0 0 0 0 0
1.0 100 0 -.1 -.5 -.5 0 0
1.5 200 0 -.3 -.5 -.5 0 0
2.0 400 -.2 -.3 .1 -.5 0 0
2.5 400 -.3 -.3 .2 -.5 0 0
3.0 400 -.3 -.3 .2 -.5 0 0
3.5 400 -.3 -.3 .2 -.5 -.3 0
4.0 400 -.2 0 0 0 -.4 0
4.5 400 -.2 0 0 0 -.4 0
5.0 400 -.1 0 0 0 0 .2
5.5 400 -.1 0 0 0 0 .3
6.0 400 -.2 0 0 0 0 .3
```
Animation of mechanisms can be performed by reading a script file which contains joint positions. Position script files are useful when animation is desired from data generated by an outside program. Animation of the outside data can give the user a better understanding of the results. The required joint values for each Link are shown below.

---

**POSITION DESCRIPTION**

Note: Each line contains all Mechanism position values with the first value on each line indicating the time.

<table>
<thead>
<tr>
<th>Revolute Link (see Fig. 2):</th>
<th>θ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prismatic Link (see Fig. 3):</td>
<td>d</td>
</tr>
<tr>
<td>Unactuated Link:</td>
<td>(none)</td>
</tr>
<tr>
<td>Knife Link (see Fig. 4):</td>
<td>x, y, θ</td>
</tr>
<tr>
<td>Tricycle Link (see Fig. 5):</td>
<td>x, y, θ, φ</td>
</tr>
<tr>
<td>Dual Wheel Link (see Fig. 6):</td>
<td>x, y, θ</td>
</tr>
</tbody>
</table>

**Example 1: Position file – Knife**

<table>
<thead>
<tr>
<th>t</th>
<th>x</th>
<th>y</th>
<th>θ</th>
<th>d</th>
<th>Time at end of velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>-1341.16</td>
<td>-1555.31</td>
<td>1.5</td>
<td></td>
<td>1.5</td>
</tr>
<tr>
<td>1.0</td>
<td>-1323.47</td>
<td>-1305.94</td>
<td>1.5</td>
<td></td>
<td>1.5</td>
</tr>
<tr>
<td>1.5</td>
<td>-1305.79</td>
<td>-1056.57</td>
<td>1.5</td>
<td></td>
<td>1.5</td>
</tr>
</tbody>
</table>
Example 2: *Position file – MoRT (tricycle) with 5 DOF Shilling Arm (revolute joints)*

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>-1272.63</td>
<td>-809.267</td>
<td>1.375</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td>-1194.41</td>
<td>-573.872</td>
<td>1.125</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>-1060.39</td>
<td>-365.146</td>
<td>0.875</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{\_\_ Tricycle \( \theta_{\text{position}} \) (body rotation)}
\]

\[
\text{\_\_ Tricycle \( y \) (y axis position)}
\]

\[
\text{\_\_ Tricycle \( x \) (x axis position)}
\]

\[
\text{\_ Time at this position}
\]

---

**ENVIRONMENT DESCRIPTION**

"Environment" refers to the fixed surroundings. Building these description files is a process similar to that of mechanism description files where the environment is composed of inanimate objects. Each object has a graphics description file associated with it (See GRAPHICS DESCRIPTION above).
Environment Files (.env)

Object (O): x, y, z, graphics filename

Example: Environment file – Pallet.gd is drawn at the origin

O 0 0 0 Pallet.gd

Example: Environment graphics file: Pallet.gd – Pallet(Brown RP) with 4 Barrels (C of 11 sides). Three are shown in Figure 10.

RP 1000 1000 150 2 <-Rectangular Prism of 1m x 1m x .15m and color 2 (brown, see GRAPHICS DESCRIPTION)

  1 0 0
  0 1 0
  0 0 1
  0 0 75 <-Identity rotation matrix

  <-offset center .075m (half of Pallet’s thickness) up from the floor

C 250 850 11 8

  1 0 0
  0 1 0
  0 0 1
  250 250 575

C 250 850 11 9

  1 0 0
  0 1 0
  0 0 1
  250 -250 575

C 250 850 11 6

  1 0 0
  0 1 0
  0 0 1
  -250 250 575

C 265 850 11 3

  1 0 0
  0 1 0
  0 0 1
  -250 -250 575

Figure 10 – Three adjacent Pallet.gd.
GRAPHICS VIEW

A view file defines the graphics window size, the "eye" position, and the point the eye is looking toward (point of attention). The graphics window size requires an x and y input: values between 500 and 1200 are best. "Eye" position is a three dimensional point in millimeters, as is the point of attention.

--- VIEW FILES (.view) ---

Window length, window height

\( x_{\text{eye}}, y_{\text{eye}}, z_{\text{eye}} \)
\( x_{\text{poa}}, y_{\text{poa}}, z_{\text{poa}} \)

**Example: View File**

```
700 700 \quad \text{<-Square graphics window of 700 x 700}
-3500 1500 4000 \quad \text{<-eye position at -3.5 m, 1.5 m, 4 m}
0 0 0 \quad \text{<-point of attention, eye is looking toward the origin.}
```

--- FORCE-TORQUE PLOTS ---

Do your joint velocities saturate the mechanism actuators? It is possible to plot torque/force plot from the joints in graphical form. Several steps must be followed: create plot object, read in plot limits and send data to plot. An example is studied step by step. This example is then implemented in a sample program.

**Example: Force/Torque Plot Tricycle.**

In `main()`
Set up Plot object for each force/torque to be plotted. Give the plot a name in this declaration (e.g., "Torque").

```
Plot torqueplot("Torque");
```

Read in y-value force/torque plot limits.
```
torqueplot.SetLimits("torquet.plot");
```

Sample torqueright.plot:
```
-500 1500 \quad \text{<- lower limit then upper limit}
```
Declare a list of joint values. This list will hold the y plot value.

Iterator<Joint_Value> forces_iter; // for torque plots

Calculate current inverse dynamics of Tricycle and put in forces_iter

forces_iter = tricycle.InverseDynamics(zeros, zeros);

Draw force and torque plots

forces_iter.Head(); // Go to head of forces_iter
torque = forces_iter()->JointValue();
forces_iter++;

time = tricycle.Time(); // Time stamp tricycle torque data

Example: source code

// Tricycle – Ultrasonic sensors
// with force torque plots

#include <Graphics_View.hh>
#include <Environment.hh>
#include <Mechanism.hh>
#include <unistd.h>
#include <Sensor.hh>
#include <Map.hh>

main() {
   //Declare Graphics View, give it an arbitrary name, e.g., view_one
   Graphics_View view_one;
   view_one.ReadView("Tricycle.view");

   //Declare Environment, give it an arbitrary name, e.g., environment
   Environment environment;
   environment.ReadDescription(view_one, "Tricycle.env");

   //Start reading in the mechanism, named tricycle
   Mechanism tricycle;
   tricycle.ReadDescription(view_one, "Tricycle.md");

   // Declare a list of joint values. This list will hold the y plot value.
   Iterator<Joint_Value> forces_iter; // for torque plots

   // Calculate current inverse dynamics of Tricycle and put in forces_iter
   forces_iter = tricycle.InverseDynamics(zeros, zeros);

   // Draw force and torque plots
   forces_iter.Head(); // Go to head of forces_iter
torque = forces_iter()->JointValue();
forces_iter++;

time = tricycle.Time(); // Time stamp tricycle torque data
cout << "Time: " << time << "\n";
torquestplot.Draw(time, torque);
}
Figure 11 – Tricycle with Force (for base) and Torque (for front wheel) plots.

```
//Read configuration (optional initial conditions or use .md positions)
//  tricycle.ReadConfiguration(view_one, "Tricycle.cfg");

Plot forceplot("Force");  // See example above
  forceplot.SetLimits("force.plot");

Plot torqueplot("Torque");  // Torque required to turn front wheel
  torqueplot.SetLimits("torque.plot");

// Create a Map
  Map DataMap;
  DataMap.ReadMap("SWAMI.mp");

//Begin motion, read velocity file
```
ifstream inputvelfile("Tricycle.vel",ios::in);

int sensor_on = 1;  // turn sensors on
float time;
double force, torque;
Vector3 zeros;
Iterator<Joint_Value> forces_iter;  // intelligent pointer for force/torque plots

while( !inputvelfile.eof() ) {

    tricycle.ReadVelocity(inputvelfile);  // Read next line of velocity file
    tricycle.IntegrateVelocity();  // Get joint position data from given velocities

    Sensor_Data *Range;
    if ( sensor_on ) {  // Activate sensors
        Range = tricycle.ActivateSensorNum(6);
        cout << "Range from sensor number 6: " << Range->Range() << "\n";
        // used to retrieve individual sensor range readings
    }

    DataMap.UpdateMap();  // activate all sensors listed in SWAMI.mp and put
    DataMap.ClearMap(1);  // clear map and fill each cell with value of 1

    view_one.Regenerate_Screen();  // Update and redraw screen

    // Calculate current inverse dynamics of tricycle and put in forces_iter
    forces_iter = tricycle.InverseDynamics(zeros, zeros);

    // Draw force and torque plots
    forces_iter.Head();  // Go to head of torque_list
    force = forces_iter()->JointValue();
    forces_iter++;
    torque = forces_iter()->JointValue();

    time = tricycle.Time();  // Time stamp torque data
    cout << "Time: " << time << "\n";
    forceplot.Draw(time, force);  // Sends data to plot and draws
    torqutplot.Draw(time, torque);

}

inputvelfile.close();  // Stop reading velocity file and close it.

cout << "End of Program\n";
}
CONTINUING DEVELOPMENTS

The two main areas of continuing development are involved with Map and path planning. Currently, Map does not retain the Sensor readings from the previous time step. Since the Map moves with the vehicle, in order for the retained data to be meaningful, the previous Map data must be moved as the vehicle moves. The movement of the Map data has not been implemented in the program code above and therefore the Map is instantaneous at each time step. The Map is cleared each time step of the motion loop before the next round of Sensor readings. If it did not clear the Map, the data in the Map would not be very useful. The Map data can be translated and rotated according to the vehicle’s motion or only translated. By only translating, fewer computations are performed, the Map stays consistently oriented, and discretization round off errors will not occur to the moved data. From the discretized Map data, useful information needs to be extracted for use in path planning and obstacle avoidance. Some methods being considered for extracting Map data are potential field, edge detection, and histogrammatic methods. The connection between these methods which further interpret the Map data and the path planning and obstacle avoidance schemes needs further work.

From the other side, path planning along a nominal path for nonholonomic vehicles is underway. A common approach with nonholonomic vehicles is to compose the path of straight lines and circles. This works fine if the vehicle comes to a stop at the interface of each path segment. For example, a tricycle going around a circle will have a constant non-zero steering angle. On a straight line, the tricycle will have a steering angle of zero. At the circle to line interface the steering wheel must instantaneously change from one angle to zero. A real vehicle cannot achieve this since it would require an infinite amount of torque by the steering actuator. In the absence of a closed loop controller since we are initially assuming no disturbances and that we can follow the path, the open loop path planning must transition between the circle and line segments. Since the nonholonomic vehicles have a minimum turning radius greater than zero when the vehicle speed is nonzero, it is desired to use a curve that can be specified for a known curvature. The curvature is the inverse of the radius of the curve at that instant. The transition curve chosen was quarter cosine wave since the curvature at the peaks can be readily matched to the circle and the inflection point curvature at a quarter of the wave’s period matches the straight line. There are some difficulties with the cosine segment if the arc around the circle is too small.

A potential method for determining the joint variables of nonholonomic vehicles uses the curvature and rates of curvature of the path segments and the vehicles kinematic equations. Curvature relations of the Knife, Tricycle, and Dual Wheel links have been initially developed but not fully tested. The Knife link has been initially simulated and tested with mostly success. All of these issues are currently under development and look to be promising.
REFERENCES


NAVIGATION AND COLLISION AVOIDANCE SYSTEM FOR A MOBILE ROBOT

FINAL REPORT

April 1, 1994

Submitted to
Westinghouse Savannah River Company

through
Education, Research and Development Association (ERDA)
of Georgia Universities

Wayne J. Book, Principal Investigator
Co-Investigators:
Ronald Arkin
Thomas Collins
Stephen Dickerson
Andrew Henshaw
NAVIGATION AND COLLISION AVOIDANCE SYSTEM FOR A MOBILE ROBOT

FINAL REPORT

April 1, 1994

Submitted to
Westinghouse Savannah River Company

trough
Education, Research and Development Association (ERDA)
of Georgia Universities

Westinghouse Savannah River Subcontract No. AA46420T
(GIT Account E25-X48)

Performance period: 9/15/92 through 3/30/94

Wayne J. Book, Principal Investigator
Co-Investigators:
    Ronald Arkin
    Thomas Collins
    Stephen Dickerson
    Andrew Henshaw
Abstract

Research and development to improve the ability of mobile inspection robots to move down aisles between stacked drums is described. In one subtask a ring of ultrasonic transducers mounted on a laboratory robot detects and characterizes the location of the drums. A second subtask develops a general purpose simulator to assist in the study of sensor based collision avoidance algorithms and applies it to several relevant vehicles and to ultrasonic sensors on the vehicles. The third subtask evaluates an integrated vision unit for determining the vehicle's position relative to landmarks placed at the end of the aisles. The fourth subtask implements line finding algorithms for camera images on a transputer to establish the aisle between the drums. Images from the laboratory and the DOE site proposed for test deployment were used.
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Abstract</td>
<td>i</td>
</tr>
<tr>
<td>2. Executive Summary</td>
<td>1</td>
</tr>
<tr>
<td>Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning</td>
<td>2</td>
</tr>
<tr>
<td>Subtask 2: Vehicle Positioning and Obstacle Avoidance</td>
<td>7</td>
</tr>
<tr>
<td>Subtask 3: Vision (Landmark Tracking) Positioning System</td>
<td>11</td>
</tr>
<tr>
<td>Subtask 4: Transputer Control System</td>
<td>15</td>
</tr>
<tr>
<td>3. Appendices</td>
<td>19</td>
</tr>
</tbody>
</table>

**Subtask 1:**

**Subtask 2**


**Subtask 3**


Subtask 4

Executive Summary

NAVIGATION AND COLLISION AVOIDANCE SYSTEM FOR A MOBILE ROBOT

The work performed under this contract is intended to enhance the safety to workers involved in monitoring stacked drums of radioactive materials. The robotic operations that lower exposure must be reliable so that the reduction is not lost by the need for repair and correction of errors. Sensor based planning and control is a key to achieving the needed reliability.

Four subtasks were performed to enhance robot operation with improved sensing, planning and control capabilities:

- Ultrasonic obstacle avoidance and corridor positioning,
- Vehicle positioning and obstacle avoidance,
- Vision (landmark tracking) positioning system and
- Transputer control system.

The executive summary is organized by those subtasks. Following the Executive Summary, several appendices describe the work in greater technical detail. In addition to this final report, the technical monitor, Clyde Ward at the Westinghouse Savannah River Technical Center, has been provided with demonstrations at Georgia Tech, video tapes showing the nature of the results, listings of code used but not included as a deliverable, and the electronic form of code has been made accessible to him.

As the project progressed, adjustments were made for developments in the proposed test deployment of a mobile inspection robot. Multiple visits to the Savannah River Site enabled us to see the developments there. On one of these visits we participated in an integrated demonstration open to qualified visitors. A visit to the deployment site at Fernald in Ohio enabled two of our researchers to observe the conditions in person. Video tapes shot at the site were used in obtaining realistic results for this application. Negotiations are under way to bring the results even closer to application with an extended statement of scope for the work.

The investigators would like to express their appreciation to Clyde Ward and the other personnel at WSRTC that made this project possible and more effective. In addition to the knowledge and experience gained by these investigators, the education of at least eight Georgia Tech students was enhanced by the project.
Subtask 1: Ultrasonic Obstacle Avoidance and Corridor Positioning

The target deployment facilities at Fernald Ohio consist of large accumulations of drums containing nuclear waste. The drums are arranged in parallel rows and separated by a narrow aisle space between them. Due to the nature of ultrasonic sensors, the characteristics of the drums (i.e., smooth rigid surfaces), and the tight space, there was a suspicion that the potentially large number of echoes and resulting noise could render ultrasonic sensors unsuitable for navigational purposes.

Statement of Objectives

The original objectives of the project were to analyze the feasibility of the use of ultrasound for position location and obstacle avoidance in a waste drum repository environment and to study specific strategies for collision avoidance, obstacle circumnavigation, center of aisle positioning and drum counting. Based on discussions with SRTC and preliminary results, we were redirected to explore the use of ultrasound for drum recognition, especially in regard to finding drums in order to aim vision systems.

Our study was conducted using a Denning MRV-II robot equipped with a Denning ultrasonic ring. This ring contains 24 Polaroid type sensors arranged in a circular fashion and equally spaced every 15 degrees. We used empty 55 gallon drums for the experiments that were arranged in a large room to simulate the Fernald site.

There were five phases in our work: preliminary study, an analysis of alternatives, the design, the implementation, and validation of the results. During the preliminary study, we performed several experiments to analyze the characteristics of the data produced by ultrasonic sensors working under specialized environments; in particular, we analyzed the influence of several parameters such as firing rate and patterns. During the second phase (analysis of alternatives), we tested different methods to enhance the quality of the data and improve its use for robot navigation. A model-based recognition method was selected as the most promising one (other alternatives were based on certain low-level features and proved not to be as robust). During the design, implementation, and test phases, the algorithms were refined to the point to which they can be used to effectively localize drums in the tested environment. Incremental implementation and partial tests were used to progressively improve the methods. Simulation and real world experiments were used to validate the method. A paper was generated reporting the algorithms and will be available as a College of Computing Technical Report.

Approach

This section describes the approach used for model-based drum recognition using ultrasonic data. Subsection 2.1 discusses the conceptual design and Subsection 2.2 describes the details of an implementation of the method on the Denning platform.
Conceptual Design

The objective of the system is to be able to recognize drums using ultrasonic data. This information can be used to aim vision systems and perform collision avoidance, obstacle circumnavigation, center of aisle positioning, and drum counting. Due to the characteristics of the environment and ultrasonic sensors, information derived from a single sample from an ultrasonic sensor is unreliable. Information based on several samples distributed in time and in space improve significantly the reliability of this information. On the other hand, deriving information from large quantities of information is expensive due to the increased size of the solution space. It is desirable to design a system that is able to improve the quality of the information as more data is gathered while minimizing the cost of deriving new information from the additional information (i.e., to avoid reprocessing old data).

A model-based recognition method was designed to recognize environmental objects using ultrasonic range data. The method emphasizes shape recognition as well as obstacle location. It is able to improve the localization of the objects as more data is added while constraining the solution space. A relaxation technique is used that significantly improves the reliability of two dimensional model matching. Thus the recognition method is able to recognize and localize drums using ultrasonic data. This information can potentially be used to aim vision systems, assist in collision avoidance, obstacle circumnavigation, center of aisle positioning, and drum counting.

Implementation

The model based recognition system was implemented on a Denning MRV-II robot. The robot was equipped with a Denning ultrasonic ring which contains 24 ultrasonic sensors arranged in a circular fashion and equally spaced every 15 degrees. All the phases of the algorithm were implemented taking into account the structure and arrangements of the sensors on this particular platform.

Test Methodology

During the preliminary phase of the project, the objective was to evaluate the quality of the data using different configuration parameters. Two programs were developed to accomplish this objective: Sampler and Tool. Sampler was used to get data from the robot and Tool was used to visualize sensor data. Experiments were designed using different firing rates and patterns. The experiments were performed in a large room where the robot could navigate between two columns with 20 drums each. The data gathered during these experiments was also used to analyze different alternatives for drum counting and to test the proposed model-based recognition algorithm.

An additional program (Driver++) was developed to further test the model-based recognition algorithm. This program implements the perception and action aspects of the AuRA architecture and it was used to simultaneously collect ultrasonic data and control the robot as it navigates
through a terrain. The program can also simulate a robot's actions and generate sensor readings on simulated environments. This capability was extensively used to evaluate the results of the algorithms developed.

Results

Table 1 shows the actual world coordinates for the drums and the estimated world coordinates obtained with the algorithm for a simulated environment. Additional results and the details of the algorithm are available in the Technical Report to be issued later or in [1].

<table>
<thead>
<tr>
<th>Drum</th>
<th>World Coordinates (ft.,ft.)</th>
<th>Estimated World Coordinates (ft.,ft.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drum 1</td>
<td>(1.00, 4.00)</td>
<td>(1.21, 3.81)</td>
</tr>
<tr>
<td>Drum 2</td>
<td>(4.00, 4.00)</td>
<td>(4.02, 3.83)</td>
</tr>
<tr>
<td>Drum 3</td>
<td>(7.00, 4.00)</td>
<td>(6.80, 3.86)</td>
</tr>
<tr>
<td>Drum 4</td>
<td>(1.00, -4.00)</td>
<td>(1.32, -3.86)</td>
</tr>
<tr>
<td>Drum 5</td>
<td>(4.00, -4.00)</td>
<td>(4.11, -3.82)</td>
</tr>
<tr>
<td>Drum 6</td>
<td>(7.00, -4.00)</td>
<td>(6.84, -3.83)</td>
</tr>
</tbody>
</table>

Table 1: Simulation Results

Summary and Conclusions

An algorithm that is able to recognize the shape and location of drums in the environment using only range ultrasonic data was designed, implemented and tested. The algorithm exploits the physical model of the ultrasonic beam and the shape of the drums to accomplish its task. Outline segments are extracted from the raw range readings to form a map of the environment that consist of object surface outlines. Then, the outline segments are clustered to find possible candidates for object locations. Finally, a relaxation algorithm progressively refines the match between the shape of the object and the outline segments by incrementally moving and rotating the shape of the object until the best match is found.

Reliable estimates of drum positions are useful to perform collision avoidance, obstacle circumnavigation, center of aisle positioning and drum counting. Also, this information is useful to aim vision systems. This algorithm has been demonstrated to provide robust results in realistic settings for drum recognition tasks.

Finally, the algorithm can be ported to the SRTC platform. To revise the algorithm it is necessary to consider the position and configuration of the ultrasonic sensors used in this platform, and to redesign the outline segment extraction formulas. The location and orientation
of the ultrasonic sensors are crucial to good performance. The algorithm works best when the sensors are oriented in a direction perpendicular to the line of motion of the robot. Also, a good estimation of the beam width of the ultrasonic sensors improves the accuracy of the physical model of the ultrasonic beam.

REFERENCES


Appendix - User Manuals

• **Sampler**

Sampler is a character based program that gathers data from the robot or an input file and outputs to standard output the position of the robot and the ultrasonic sensor data.

Syntax: info [-i input file] [-o output file] [-p period] where input file and output file are the names of files and period is the desired time in seconds between output samples. If an input file is not specified then input is gathered from /dev/ttya which is assumed to be connected to the robot.

• **Tool**

Tool is a graphic based program that gathers data from standard input (Sampler outputs) and presents the data in a graphical representation. For each sample, Sample uses two linear transformations on the input points: the first transformation converts data from local coordinates of the robot (both translation and rotation) to world coordinates. The second transformation converts points from world coordinates to screen coordinates. Points are represented using homogeneous coordinate systems and linear transformations are represented as 3x3 matrices.

Syntax: tool - runs TOOL using the information stored in `data' file.
tool robot - runs TOOL using the information from the robot.

An environment information is supplied in the file: default.env
- **Driver++**

The Driver++ environment is a C++ program that implements the pilot level of the AuRA architecture. The environment can control the real robot in real environments or a simulated robot in simulated environments.

**Syntax:**

`driver definition.def [number-of-samples]`

The definition file contains robot configuration details. When the program is executed in simulation mode, this file contains a reference to an environment definition file (.env).
Subtask 2: Vehicle Positioning and Obstacle Avoidance

Autonomous inspection of nuclear waste stored in rows of drums will require navigation and obstacle avoidance by the robot. Sensors will provide information on the nominal path and on the obstacles. To study the sensor location and motion planning algorithms a versatile simulation of the vehicle that accurately represents its shape, kinematics and dynamics is created. Simulated sensors detect obstacles and store the information in a map.

Statement of Objectives

The effect of sensor placement and motion planning algorithms was proposed for study. Initially the details of the vehicle were not known and initial modeling was based on the Denning robot used in the Georgia Tech experiments. When the TRC vehicle was chosen the new vehicle dynamics and kinematics were added. Very simple sensor models were initially proposed but discussion with the Technical Monitor indicated more realism was needed to represent the ultrasonic sensor was needed.

Approach

This subtask studies obstacle avoidance in the course of path following. Prior work in this area usually suffer from three deficiencies: vehicles are assumed to move in any direction, the shape of the vehicle is not accounted for explicitly (the shape is assumed to be cylindrical), and the dynamics and kinematics of the vehicle are not realistically accounted for. When a vehicle is regularly moving down very narrow aisles, these omissions could be critical.

To understand the directions taken in this subtask it is helpful to know the history of the project. The project for sensor based collision avoidance was initiated when the vehicle to be used for inspection of stored drums was undecided. The sensor suite of the vehicle was similarly unknown, and the ability of the vehicle's navigation and obstacle avoidance schemes to perform in the waste storage site was unknown. In fact, the waste storage site to be used for the test implementation itself was uncertain. To study the obstacle avoidance as applied to the target problem a versatile simulator was needed, and one was created. Although many of these uncertainties have been resolved for this one application, many similar applications with different vehicles, algorithms, or obstacles may need a similar evaluation. The existence of a physical example as we conclude this period of work presents an extraordinary opportunity to evaluate the validity of this work.

Vehicles with wheels or tracks have a variety of steering mechanisms. Each of these steering mechanisms requires additional modeling effort. In addition, articulated arms, sensor mounts, or lifting devices will create additional degrees of freedom with more conventional joints allowing rotation or translation. These have all been treated as generalized forms of joints using an innovation of this and related Georgia Tech research.[1] This uniform treatment allows the creation of an infinite variety of devices consisting of joints of various types and relative orientation and position. MBSim not only insures that these relative motions are feasible, but
models the dynamics via a Newton-Euler formulation to calculate the joint torques necessary to
move the joints with the specified accelerations and resulting velocities, and positions. In some
instances the motion might be slow enough to ignore the required actuator forces. Surprises lurk
in this assumption, however. Consider the popular path planned for vehicles consisting of circles
and straight lines, for example. To follow this path would require infinite acceleration of the
wheels of a TRC Helpmate at the point of transition from the circle to the tangent straight line,
unless both wheels are stopped at that point. Either more complicated paths must be chosen or a
deviation from the path must be accepted. How much deviation will result? The dynamics must
be known to answer this question by analysis.

Implementation

The simulator used in this work has been named MBSim, for Multi-Body Simulator.[2] It has
been written in C++ in an object oriented fashion.[2] This has proven highly valuable in
enabling the addition of new features. The selection of the TRC Help Mate vehicle as the
platform for evaluation in hardware, for example, brought about the need to create a new vehicle
shape and new vehicle kinematics and dynamics. A new joint type was created in 1.5 hours of
work. This was not simply a new collection of existing modules, but the creation of a new
kinematic behavior with non-holonomic constraints. The vehicle shape was created in 30
minutes. This required only the use of existing shapes united together to create the Help Mate
shape. Initially, the simulator was implemented on a Sun IPC workstation, which is a low end
machine and not optimized for graphics. When a faster machine (SGI Indigo 2 XZ) became
available, we were encouraged to convert the software to this machine which was more
compatible with that in use at WSRTC. This conversion was accomplished in less than one
month. Although there has been some price paid in terms of the number of capabilities added,
the improved productivity of the faster machine will be well worth this investment.

Six joint types have been implemented. Three of these represent popular types of robotic
vehicles. The "knife" allows the wheel(s) to be pointed in any direction like the Denning robot.
The "dual wheel" has two independently driven wheels along the same axis, like the TRC
vehicles. It is also a reasonable approximation of skid steering. The "tricycle" represents either
the tricycle or Ackermann type steering used in automobiles. Three joint types represent
mechanisms or manipulators that might be on a vehicle, they include the rotary joint, the
prismatic joint and the unactuated joint. Instances of these six joints are assembled into a
mechanism. Multiple mechanisms may exist simultaneously in a simulation. The creation of a
new simulation does not require recompilation, but only the creation of a script file with the
joints, environment, and motions specified.

A concession often made in vehicle modeling is to assume the world is two dimensional.
MBSim does not make this concession. The motion of vehicles is assumed to be on a plane, but
the obstacles and the placement of sensors is three dimensional. This is essential with arms
placed on a vehicle. It should also be helpful when a vehicle is moving in tight quarters. The
WSRTC vehicle SWAMI, for example, has a sensor that sticks out in front at the same level that
the pallets in the warehouse stick out. It is important not to hit the pallet, but it is also important
to maneuver in very tight spaces. MBSim does not have to make overly conservative
assumptions to avoid collisions. The primitive shapes in MBSim are simple: cylinders,
rectangular prisms, planes, cones, and frustrums. They can be arbitrarily positioned and oriented. They may be fixed or moving with a mechanism.

In order to simulate vehicle action in avoidance of obstacles, some knowledge of the vehicle location must be assumed. The original proposal called for this knowledge to be absolute, that is the action of the sensor was not to be part of the simulation. This was extended first to include a simple range finding sensor detecting the first obstacle along a line. With encouragement from the technical monitor this was extended to be a cone representing an ultrasonic sensor. The value returned is the nearest intersection of the cone with an obstacle. As currently implemented the exact intersection is not found, but only the intersection with a bounding sphere. This algorithm will be extended, but the bounding sphere will be checked first to see which objects should be checked by more precise means. The results of a sensor are placed in a map that indicates which two dimensional cells are suspected of having objects in them. New information can be used to replace the earlier information, or the old and new information can be combined in a quasi-statistical way to refine the knowledge about object location.

To develop the program capabilities the vehicles were initially moved using a series of time intervals with constant velocity. To move into obstacle avoidance the generation of paths from parametric curves was developed. The combination of circles and straight lines was used as is often cited in the literature. [5] The pitfalls of these curves are mentioned above under the discussion of dynamics. Various blending approaches are now being explored to keep actuator torques feasible while placing the vehicle back onto the straight line or circular path. Avoidance algorithms will then link the parameters of the curve with the obstacles detected and recorded on the map.

**Evaluation of MBSim**

MBSim has been evaluated by constructing models of three robots for which we have laboratory examples: SWAMI, a Denning Robot with five degree-of-freedom arm, MoRT with a mounted Schilling arm. In addition a hypothetical tricycle and fork-lift truck were constructed. The pallets and drums typical of the target application were constructed for an environment. The sensors on SWAMI were approximated with the best available information at the time. The modeling of new vehicles from the existing joint types is very easy and satisfactory. The graphics to describe the vehicles and environment is not difficult but can be tedious due to the lack of a user interface.

A real need for an improved user interface for entering motions is needed. Currently a file with velocities is entered. We want to enter points on paths and have the system calculate the joint velocities to follow these paths. This will not be unique but should be feasible for the type of joint used. For the nonholonomic vehicle joints the conversion to joint velocities is difficult and currently underway.

In the present form the motions are simulated and animated in faster than real time expected for the actual inspection of SWAMI. The sensors of SWAMI successfully transfer their data to a local map. To this point collision avoidance algorithms have not been tested, although this is
under way. The priority of this task was reduced by the initial satisfaction of WSRTC personnel with the prepackaged algorithms from the TRC.

**Conclusion**

The creation of a simulation capability has been very successful. The capabilities include three dimensional animation, sensor simulation, and dynamics that were not originally proposed. The software was adapted to the equipment procured by WSRTC and transferred to a computer platform compatible with that used by WSRTC. Collision avoidance algorithms can now be evaluated, but have not been evaluated to date.

Future work should use the simulator to evaluate recently discovered needs of the SWAMI vehicle for backing up that are not met by the initial SWAMI vehicle and its sensors. The use of the software for detecting collision of MoRT with itself and its surroundings also appears to be a need that MBSim can meet more cost effectively than existing commercial packages.

**REFERENCES**


Subtask 3: Vision (Landmark Tracking) Positioning System

Statement of Objectives

The objective of subtask 1.3 of the DOE proposal is to investigate the feasibility of using machine vision as the basis for a stand-alone position measurement system. The system is required to measure the position of a mobile robot in a 100 foot aisle to within one inch laterally and to within one foot along the aisle. The necessary optics, lighting, and fiducial material, as well as the processing and power requirements were to be studied. The final result of the study was to generate a design strategy for a mobile vehicle position measurement system.

Approach

To determine the feasibility, theoretical studies were conducted and laboratory experiments were implemented as proof of concept. The measurement system investigated was two onboard integrated vision systems with fixed optics viewing artificially illuminated retroreflective landmarks located at the ends of the aisle. Computer algorithms would then process the captured images to determine the vehicle's position. The purpose of the study is to determine the required optics, lighting, and the proper computer algorithm to make this system meet the design constraints.

The system's main component is an integrated vision unit (IVU) manufactured by Dickerson Vision Technology, (DVT). The IVU uses a Motorola 68000 processor with memory, I/O (input/output) and a CCD (charge coupled device) video array. The IVU's microprocessor can interact with external hardware and perform image processing tasks. The unit's size is approximately 30 cm x 15 cm x 5 cm.

The landmarks at the ends of the aisle were constructed of retroreflective material. Retroreflective material reflects light directly back along the incident angle to the light source. This effect makes retroreflective material appear approximately 1000 times brighter than diffuse white surfaces illuminated with the same light. This produces a high contrast image using a minimal amount of light.

A xenon strobe tube was chosen for fiducial illumination due to its function and form. The small xenon strobe tube can be placed close to the CCD lens minimizing the angle between incident and reflected light. The xenon strobe emits light for less than 1/1000 second which can be electronically synchronized with the vision system image capture. Therefore light energy is not wasted during the large portion of time when the IVU is not capturing an image an important feature for a battery powered vehicle. Strobe tube illumination's flash rate is not important for the proposed application.

The software extracts information about fiducial geometry and fiducial location within the image. This information determines the distance from the aisle ends and the lateral position. Due to the IVU's low 165x192 pixel resolution, an edge subpixelization algorithm is used to increase
Subpixelization determines the edge location of a grayscale image to a subpixel resolution by fitting a parabola to three points representing the edge of the image. Subpixelization typically increases resolution to one tenth of a pixel. Given the fiducial locations within the image, a triangulation algorithm determines the lateral offset. The geometric size data and past information are used to determine the vehicle's position in the direction of the aisle centerline.

**Implementation**

The design concepts were tested experimentally to verify the theoretical results. To test illumination, external strobe hardware was developed[3]. An electronic circuit was fabricated to provide computer (IVU) controlled light intensity and strobe frequency. This controls strobe power and minimizes energy consumption. The unit was tested for time to failure by letting the strobe run continuously at 10 Hz and full power. In addition, the light output was measured by the change in the pixel values when a white surface at a given distance was illuminated. The power consumption and reliability were tested as well. Strobe characteristics such as the time constants, the strobe pulse duration, and the efficiency of the strobe were examined.

To provide tracking capabilities and increase flexibility, a CCD head positioning system was built. The IVU controlled the stepper motor drive. This unit provides the flexibility needed for more difficult cases such as turning corners and reduces the number of CCD heads needed to two[4]. Stepper motor tests were conducted to determine the rate at which the stepper could turn without losing a step and the repeatability of determining a landmarks position when using the stepper motor[4].

Software was developed and tested to track fiducials similar to the one proposed for this project. The IVU software controls the stepper motor and the strobe illumination as well and calculates subpixelized fiducial location. Triangulation was also incorporated for the inverted case of stationary cameras and a moving fiducial. A separate experiment was performed to determine the lateral position measurement accuracy[1].

**Evaluation/Results**

A prototype strobe/stepper/IVU combination successfully tracked a fiducial indoors at 100 feet under normal ambient light conditions. A separate experiment tested the sensitivity of the IVU at 75 feet. Results showed a lateral derivative of ±1.5 feet[1].

Experimental results and vendor data indicate the following powers consumption associated with the proposed system. Each strobe's maximum power consumption is 23 watts when emitting maximum light energy at 10 Hz[3]. The power consumption depends on strobe intensity and frequency. It is estimated under normal conditions where less than full power is required, the two strobes will consume under 30 Watts at 10 Hz. The stepper motor consumes approximately 14 Watts and each IVU consumes ~4 Watts. The total estimated power consumption for the system is approximately 52 Watts.
The speed of the IVU integrated system is limited to 12 Hz by the strobe at maximum light power. This frequency can be increased by lowering the light intensity, however only at a few instants such as turning a corner is it envisioned that increasing the frequency will be necessary. For normal aisle operation, rates of less than 10 Hz are recommended and 1 Hz may be sufficient.

Strobe test results show the strobe has a charging time constant of 18.9 ms. This time constant dictates the frequency of the strobe for different energy levels. The duration of the strobe light emission is 500 us. The strobe life is measured by the number of flashes generated before the strobe light intensity falls below a predefined percentage of the initial light intensity value. The drop in light intensity is due to the formation of deposits inside the glass enclosure. The xenon tube life is dictated by factors such as the amount of energy per flash, strobe frequency, and environment. The strobe ran a week at full power and 10 Hz without an appreciable decrease in intensity. The strobe tube reliability was ~99.99% when running at 10 Hz and full power[3].

The test results for the stepper motor with video head attached show it can return to a position with a standard deviation of 0.0694 pixels. This is equivalent to a standard deviation of 0.093 inches at 100 ft with a 12 mm lens. However, the stepper motor error is eliminated with the two head system observing two fiducials at opposite ends of an aisle.

**Conclusions**

These results show that the proposed integrated vision scheme for vehicle position measurement is highly feasible. The practicality of this application was demonstrated in laboratory experiments.

Results indicate an integrated vision system with a controllable external xenon strobe and stepper motor can detect and track retroreflective targets at distances greater than 100 feet. Tests show the vehicle's range can be determined to within +/- 1.5 feet along the aisle with greater resolution possible by using filtering techniques. The results also show that the subpixelization software combined with a triangulation algorithm can detect lateral position to within one inch. Therefore, it has been determined that two vision systems mounted back-to-back can be used to measure position along a 100 foot aisle to within 1 inch laterally and 1.5 feet longitudinally when retroreflective targets are placed at each end of the aisle. The power consumption of the system is less than 52 Watts. In fact, at 1 Hz and with redesigned stepper motor control it is anticipated that the entire unit will consume less than 25 Watts.

The use of machine vision to track a fiducial is being used for other applications at Georgia Tech. One such application is the Unmanned Flying Vehicle competition were a fiducial mounted on a helicopter is tracked with an IVU to determine the vehicle position and velocity[2]. In addition, work is currently being continued to develop the specific hardware for the proposed measurement system.
REFERENCES


Subtask 4: Transputer Control System

Routine monitoring of stored radioactive materials can be performed safely by mobile robots. In order to accomplish this in an environment consisting of aisles between drums, it is necessary to have a reliable means of aisle-following coupled to a high-performance on-board controller. This work describes the adaptation of successful road-following methods based on visual recognition of road boundaries to the waste storage problem. Since the effort is targeted for near-term usage in normal operating conditions, special emphasis has been given to the implementation of the visual processing on practical (i.e., small, low-cost, and low-power) parallel-processing platforms. The system described here has been implemented on a robot in our laboratory.

Statement of Objectives

In “Navigation and Collision Avoidance System for a Mobile Robot” (a response to the Westinghouse Savannah River Company Request for Proposal 92006EQ), the following statement of objectives is given:

Existing algorithms already tested for indoor corridor following, road following and outdoor path following will be developed and tested for use in drum corridor navigation. In particular, a version of the fast line finder stay-on-path perceptual schema will be utilized to enable visual servoing of a holonomic mobile robot (e.g., Denning, Cybermotion) in a drum corridor.

Specific focus-of-attention and expectation mechanisms will be used to make the computation tractable and perform in real-time. A fast region segmentation algorithm will also be fielded and tested in the context of the same task. Both algorithms use off-the-shelf video hardware. Tests for robustness and accuracy of position determination will be conducted.

An architecture for performing real-time computer vision navigation will be developed using the Inmos Transputer. The Transputer is a 32-bit processor designed for parallel processing systems. Using available Transputer-based frame grabbers and other Transputer processing modules, a powerful distributed system will be built and tested using the algorithms listed above. Additionally, the architecture will be designed to readily incorporate other tasks associated with robot navigation including ultrasonic obstacle avoidance, active and passive position estimation, and motor control command generation.

Deliverables will include monthly status reports, final technical report, College of Computing technical report, demonstration at Georgia Tech of the technology, laboratory grade videotape of results, and formal presentation for Georgia Tech and interested SRC personnel at Georgia Tech.
These objectives were met in the research conducted for this task, as shown in this summary. However, discussion with WSRC representatives during the course of this research indicated a shifting of priorities and focus. Significantly, a de-emphasis of the use of a specific processor architecture (in this case Transputers) was made in favor of the development and testing of the specified vision algorithms. Additionally, research into parallel processing control systems was limited due to integration difficulties with the WSRC-approved robotic platform.

**Approach**

To provide navigation in the semi-structured environment found in a hazardous waste storage facility, a sensor system should use the inherently available information of the storage structure (building). Since these structures tend to be organized for the convenience of visual observation, it is assured that visual navigation cues will be present. For this reason, vision currently provides the greatest information content among reasonably-priced sensor systems. In addition, vision can be applied to the task of drum inspection, making dual use of a single sensor.

A drawback to the use of vision in a sensor system is the large volume of data that must be processed to extract the necessary navigational cues. Associated with this drawback is the high computational processing needed to provide reasonable throughput rates for real-time control. Sophisticated image processing algorithms can alleviate the first problem, at the expense of adding to the second, simultaneously reducing the quantity and increasing the quality of the sensor data.

Our approach to this problem is to:

- utilize the simplest reliable navigational cues available from the environment,
- perform the necessary image processing to extract those cues and reduce the data volume to a manageable level,
- provide processing power sufficient to implement these algorithms within the time constraints provided by the robotic control system, and
- implement the system as a separate module so that it can operate without impacting the performance of established components

**Implementation**

**Hardware**

To the end of making this system completely portable, we have designed a modular parallel-processing architecture that is largely host-independent. The central processing element of this architecture is the Inmos Transputer, a high-performance microprocessor that has been designed for parallel processing. For this application, we use Transputer-based frame grabbers, processing elements, and interfaces for communication and storage. All of these components are available
as modular, off-the-shelf products that communicate through high-speed communication links. Host interfaces for Transputer systems are available for most popular platforms. For many of these systems, the interface is integrated into a carrier board that provides for the physical mounting of Transputer-based modules (or TRAMs). To demonstrate the capabilities of a Transputer-based control system on a commercial (Denning) robot, a custom carrier was developed to match the robot's native STD-bus form factor.

Another assembly used for testing of the system was a single Eurocard carrier (similar to the GISC VME form factor) that contained all of the modules necessary for control of a three-axis motorized camera platform. Output of the vision system was used to direct the camera motor control system, which emulated the essential aspects of robot steering.

**Software**

Our research concentrated on aisle boundaries as the essential visual cues for safe navigation. Two algorithms were studied for effectiveness of aisle-boundary extraction: the Fast Line Finder (FLF) and the Fast region segmenter (FRS).

The Fast Line Finder algorithm extracts left and right aisles by sampling a single video frame, performing edge detection, building connected regions of pixels with similar line orientations, and filtering out low-interest orientations. The Fast Region Segmenter algorithm is similar to the FLF, however it attempts to partition the image into contiguous regions and then provide parameters for those segments. In either case, the output from these algorithms is used to derive an estimate of the aisle center. Using this estimate and knowledge of the current heading, the controller servos the robot to follow the expected center line.

An example motor controller is provided in a set of library routines written to drive the three-axis camera platform. The controller using these libraries communicates with the vision subsystem through high-speed serial data links which are standard across the Transputer family.

**Testing**

The environment for the mobile robot application can be generally described as rows of closely spaced pallets, each containing stacks of drums. The rows are separated by aisles that are large enough for the robot to pass through (about 1 m). It is required that the robot be able to travel down these aisles without colliding with any drums or pallets. The drums vary in size, and they may extend past the edge of the pallets. The resulting visual boundary (at the floor) is a series of line segments and curves, sometimes interrupted by gaps. In order to provide steering cues, the visual algorithm must be able to identify the approximate left and right boundaries, in spite of their deviation from straight lines, and project an imaginary center line.

Two methods were established for testing the vision system and related algorithms. The first setup used videotape images from Fernald that simulated motion of a robot through the waste storage facility. These images were digitized and stored for processing by both a Sun workstation and the Transputer-based vision system. The second test setup consisted of
simulated environment constructed from 50-gallon drums and pallets assembled into aisles. These aisles were navigated by a camera-equipped Denning robot driven by a remote Sun workstation running an image processing algorithm.

Conclusions

The Transputer-based vision system achieved about 75 percent of the processing speed of the Sun workstation, which was still more than adequate for this application. In order to perform thorough radiation detection, the robot's speed would be relatively low, which constrains image processing speed more than the on-board hardware. Since Transputers were used, the architecture can easily be expanded to achieve higher frame rates if required. An example of an obvious parallelization would be to use a different processor for the left and right boundary extractions.

Testing of the FLF algorithm with images of stored hazardous waste showed that this implementation is robust and well suited for autonomous navigation. Use of aisle-boundary extraction techniques appears to provide suitable navigational cues and is tolerant of variations in the storage environment.

This research has shown that a low-cost, high-performance vision system can be built using a modular parallel-processing system. Because of the modularity and communication techniques, portions of this system could be rapidly integrated into a completely different hazardous waste inspection system. Our research has not targeted the current SWAMI architecture explicitly, as this vision system is completely independent of the host architecture. For future work, this system could be integrated with SWAMI II simply by packaging it onto commercially-available VME carrier boards. Additionally, this architecture supports both automated and manual visual inspection since the vision system (and its associated computing power) need not be dedicated solely to the visual navigation task. Standard image compression techniques could be easily adapted to this system with the parallel-processing ability providing the necessary computing power for real-time image archiving.

REFERENCES


NAVIGATION AND COLLISION AVOIDANCE SYSTEM
FOR A MOBILE ROBOT

APPENDICES
FINAL REPORT

NAVIGATION AND COLLISION AVOIDANCE SYSTEM FOR A MOBILE ROBOT: TRANSPUTER CONTROL SYSTEM FOR VISUAL NAVIGATION

MARCH 1994

THOMAS R. COLLINS
ANDREW M. HENSHAW

CONCEPTS ANALYSIS DIVISION
ELECTRONIC SYSTEMS LABORATORY
GEORGIA TECH RESEARCH INSTITUTE
GEORGIA INSTITUTE OF TECHNOLOGY
ATLANTA, GEORGIA 30332-0800

PREPARED FOR:

EDUCATION RESEARCH AND DEVELOPMENT ASSOCIATION OF GEORGIA UNIVERSITIES (SUBCONTRACT 92006)
900 ATLANTIC DRIVE
ATLANTA, GEORGIA 30332-0425

CONTRACT A-9264-000

GEORGIA TECH RESEARCH INSTITUTE
GEORGIA INSTITUTE OF TECHNOLOGY
A UNIT OF THE UNIVERSITY SYSTEM OF GEORGIA
ATLANTA, GEORGIA 30332-0800
1. Introduction

Routine monitoring of stored radioactive materials can be performed safely by mobile robots. In order to accomplish this in an environment consisting of aisles of drums, it is necessary to have a reliable means of aisle-following coupled to a high-performance onboard controller. This work describes the adaptation of successful road-following methods based on visual recognition of road boundaries to the waste storage problem. Since the effort is targeted for near-term usage in normal operating conditions, special emphasis has been given to the implementation of the visual processing on practical (i.e., small, low-cost, and low-power) parallel-processing platforms. The system described here has been implemented on a robot in our laboratory.

To provide navigation in the semi-structured environment found in a hazardous waste storage facility, a sensor system should use the inherently available information of the storage structure. Since these structures tend to be organized for the convenience of visual observation, it is assured that visual navigation cues will be present. For this reason, vision currently provides the greatest information content among reasonably-priced sensor systems. In addition, vision can be applied to the task of drum inspection, making dual use of a single sensor.

A drawback to the use of vision in a sensor system is the large volume of data that must be processed to extract the necessary navigational cues. Associated with this drawback is the high computational processing needed to provide reasonable throughput rates for real-time control. Sophisticated image processing algorithms can alleviate the first problem, at the expense of adding to the second, simultaneously reducing the quantity and increasing the quality of the sensor data.
Our approach to this problem is to:

- utilize the simplest reliable navigational cues available from the environment,
- perform the necessary image processing to extract those cues and reduce the data volume to a manageable level,
- provide processing power sufficient to implement these algorithms within the time constraints provided by the robotic control system, and
- implement the system as a separate module so that it can operate without impacting the performance of established components.

To the end of making this system completely portable, we have designed a modular parallel-processing architecture that is largely host-independent. The central processing element of this architecture is the Inmos Transputer, a high-performance microprocessor that has been designed for parallel processing. For this application, we use Transputer-based frame grabbers, processing elements, and interfaces for communication and storage. All of these components are available as modular, off-the-shelf products that communicate through high-speed communication links. Host interfaces for Transputer systems are available for most popular platforms. For many of these systems, the interface is integrated into a carrier board that provides for the physical mounting of Transputer-based modules (or TRAMs). To demonstrate the capabilities of a Transputer-based control system on a commercial (Denning) robot, a custom carrier was developed to match the robot's native STD-bus form factor.

Another assembly used for testing of the system was a single Eurocard carrier (similar to the GISC VME form factor) that contained all of the modules necessary for control of a three-axis motorized camera platform. Output of the vision system was used to direct the camera motor control system, which emulated the essential aspects of robot steering.
Our research concentrated on aisle boundaries as the essential visual cues for safe navigation. Two algorithms were studied for effectiveness of aisle-boundary extraction: the Fast line finder (FLF) and the Fast region segmenter (FRS).

The Fast line finder algorithm extracts left and right aisles by sampling a single video frame, performing edge detection, building connected regions of pixels with similar line orientations, and filtering out low-interest orientations. The Fast region segmenter algorithm is similar to the FLF, however it attempts to partition the image into contiguous regions and then provide parameters for those segments. In either case, the output from these algorithms is used to derive an estimate of the aisle center. Using this estimate and knowledge of the current heading, the controller servos the robot to follow the expected center line.

An example motor controller is provided in a set of library routines written to drive the three-axis camera platform. The controller communicates with the vision subsystem through high-speed serial data links that are standard across the Transputer family.
2. Transputer-Based Control System

Autonomous machines with sensory, manipulative, and locomotive capabilities are a significant class of intelligent systems holding great promise for performing hazardous or mundane tasks. Although much work has been performed with isolated aspects of intelligent machines, including vision, sonar, manipulator control, and knowledge-based reasoning, the algorithms are often not considered within the context of a complete machine. In many prior efforts, both software architectures and hardware architectures have been developed to meet the requirements of specific projects, with little regard to reusability in other applications. Often, experimental systems are not robust, failing due to relatively minor environmental variations or task redefinitions. This section describes the integration of two separate efforts to address these problems. One is a reactive software architecture that has been demonstrated to perform a variety of tasks well, and the other is a targeted, yet flexible, computer architecture that provides modularity and expandability.

Most of the earliest work with intelligent machines relied on direct programming in declarative languages [27, 28, 29]. This resulted in software architectures that could only accommodate certain situations, and even then only in a “do this, then do this” fashion. There has been a gradual trend toward reactive software architectures, which combine relatively simple behaviors to implement complex tasks. Since these simple components are designed to respond to general stimuli in “common-sense” fashion, greater robustness is achieved, even when a higher level of deliberative behavior is added.

From a hardware architectural standpoint, many early autonomous machines relied on offboard control, since the necessary computers were so large. Then, as microprocessors became widely available to provide onboard intelligence in relatively small packages, many more projects began. Even today, however, many mobile robots depend on
detached workstations, because of the increasing computational demands and the need for sophisticated user interfaces. It has become increasingly evident that autonomous machines must incorporate multiprocessor architectures, but it is not quite clear what form the architectures should take.

The majority of these specialized multiprocessor architectures have taken a hierarchical form, often a simple tree [17, 27, 29]. In the context of an intelligent machine, the processing tasks closest to the environment (i.e., the "low-level" tasks) tend to map most obviously into a tree. Higher levels of thought prefer to describe both perceptions and actions in concise symbolic terms, while environmental interactions tend to involve large amounts of data, often to or from loosely-coupled subsystems. It is thus advantageous to have multiple perceptual processes and multiple control processes running simultaneously, each having extensive interaction with the environment, but little interaction with each other. Communication with higher levels is less frequent, and the messages tend to contain condensed information in a symbolic format. Intermediate levels of processing act to integrate data (on the sensory side) or to coordinate simple tasks (on the control side). In this simple paradigm, only one highest-level, or reasoning, task is required. Unfortunately, this process usually forms a computational bottleneck, since it is an integral part of every sensor/control path.

More recently, hierarchical designs have emphasized connection across the hierarchy, as in the NASA/NBS standard reference model for telerobot architectures (NASREM) developed by Albus [1] and the Multiresolutional Control Architecture of Meystel [26]. These architectures allow nested control loops via distinct task levels within the hierarchical structure, providing higher bandwidth for low-level tasks that require shorter response times. Each higher level in the hierarchy thus provides for increasing levels of abstraction in perception, reasoning, and control, and each maintains a model of the world appropriate for its purposes. No clear consensus exists, however, for the number and type
of levels to use in such architectures, and it is paradoxical that the attempt to provide abstractions actually requires such concrete definitions.

The hierarchical viewpoint is increasingly being challenged by reactive approaches. The concept of schemas, as typified by Lyons [25] and Arkin [2, 3], implements sensor-effector connections in a more flexible fashion. A schema is a pattern of behavior exhibiting a stimulus-response characteristic. Typically, each schema monitors only a portion of the available input data and produces an output that may have to be combined, superimposed, or otherwise reconciled with other outputs. Schemas may communicate with other schemas or with sensors and effectors. Normally, an intelligent machine would create instances of predefined schemas as necessary to produce more complex behaviors. Such apparent complexity arises both from the ability of schemas to use other schemas and from the parallel actions of independent schemas. Multiprocessor architectural implementations of schemas will generally be more dependent on dynamic resource allocation than hierarchical approaches, which can anticipate processing requirements within each level.

Another significant reactive approach is the subsumption architecture developed by Brooks [10, 11, 12]. Multiple levels of competence are defined, connecting input and output in a layered system. Higher levels of competence inhibit or subsume all lower levels, and the hardware usually provides direct support for this subsumption characteristic by implementing each level within its own processing subsystem. This structure allows a machine to be developed in stages, building each level on top of a machine that already functions at some given degree of competence. Another advantage is that the lower levels still exhibit useful behaviors that are activated in the absence of any inhibition from above. For example, a low-level obstacle avoidance behavior is still useful even when a path planner exists to provide intermittent goals.
In spite of the emergence of these promising approaches, there is still a considerable amount of disarray in the overall architectural scene. The key limitations of much of the previous work, including both hardware and software considerations, have been:

- reliance on offboard computational facilities and radio communication,
- ad hoc, inflexible hardware architectures,
- lack of inherent support for parallelism,
- awkward development environments, and
- lack of portability.

These limitations hamper the development of reusable, modular hardware and software components, and they have thus slowed the development of a significant commercial market in sophisticated autonomous machines.

2.1. **AuRA – the Autonomous Robot Architecture**

AuRA is a hybrid architecture encompassing aspects of both deliberative and reactive control. It consists of 5 major subsystems:

- Perception – charged with collecting and filtering all preliminary sensory data.
- Cartographic – concerned with maintaining long-term memory (*a priori* models of the environment), short-term memory (dynamically acquired world knowledge), and models of spatial uncertainty.
- Planning – consists of a hierarchical planner (the deliberative component) and the motor schema manager (the reactive component).
- Motor – the interface software to the specific robot to be controlled.
- Homeostatic – a component concerned with dynamic replanning in light of available internal resources [7].

The overall architecture has been described in detail elsewhere. The reader is referred to [4, 6] for more information.
The hardware migration to ANIMA thus far has been concerned with the reactive and perceptual components of the system that run within the confines of the motor schema manager. Figure 1 presents the logical relationships between the varying schemas which constitute this portion of AuRA.

Action-oriented perception forms the underlying philosophy for channeling sensory information to the motor schemas (behaviors) [5]. Only the information that is essential to a particular motor behavior is transmitted to it, essentially on a need-to-know basis. The message-passing paradigm found in ANIMA is well suited for this type of information flow.

Each of the active motor schemas generates a velocity vector in a manner analogous to the potential fields method [23, 24] with the individual results being summed, normalized and transmitted to the robot for execution. The speed of operation is highly dependent on the rate of processing of incoming sensory data. The parallelism found in the Transputer implementation described below is a natural match for this aspect of the AuRA architecture.
2.2. ANIMA hardware architecture

We have developed a flexible, real-time platform for the development of AuRA and other software architectures. The skeleton of our hardware architecture, ANIMA (Architecture for Natural Intelligence in Machine Applications), has been developed from basic principles. It incorporates a triad of basic systems, just as a conventional computing system includes input, output, and processing subsystems. This fundamental triad of subsystems carries over into the architecture of an intelligent machine, but a more general interconnection pattern is required. The addition of a communication channel between the input subsystem and the output subsystem allows the machine to exhibit reflexive behaviors. Such behaviors are analogous to reflexes in biological systems, where the communication channel is implemented by structures within the spinal cord and lower brain. While it would be possible to develop an autonomous machine without such a channel, it would not take advantage of the localized intelligence within the input and output subsystems. The resulting increase in computational load on the processing subsystem would result in slower response time.

Clearly, reflexive behaviors are virtually “hard-wired” into the system, and their implementation is best reserved for behaviors that:

- must be performed reliably and quickly, usually to avoid danger to the machine or to humans,
- require little or no integration of information from multiple input systems, and
- although primitive, usually produce an effect more desirable than if no action at all had been taken.

The deliberative component controlling the input and output subsystems is called the Reasoner. A major aspect of this reasoning capability is the need to maintain some sort
of world model based on sensory input, at least for anything more than basic reactive behavior.

2.2.1. Parallel sensors and effectors

The vast majority of intelligent machine research has assumed that the input/output devices, just as in a conventional computer, are largely independent in their low-level operation (at or below the level of the device driver). For input devices, the combination of these independent streams of data has often been referred to as sensor integration, and we include a process, called the Integrator, to perform this task. On the output side of the structure, the most appropriate term is coordination, although the specific definition varies considerably in the literature. Corresponding to the Integrator, we include a process called the Coordinator. Often, researchers wish to place task planning in this position. The most common form of task planning is trajectory planning (or path planning) in which one or more effectors are given commands that, taken together, result in a desired trajectory within the environment. Although this may be viewed as a sort of coordination, it is really a process much more closely aligned with the Reasoner in our nomenclature. A planned trajectory, given in a high-level description, could serve as an input to the Coordinator process, which would then take control of the effectors in order to implement the trajectory.

These additional parallel processes are illustrated in Figure 1. The independent sensor subsystems are called logical sensors, in much the same sense as those of Henderson [18] or Crowley [15]. At this point, a logical sensor is best thought of as a combination of a physical sensor, capable of estimating some property of the environment or the machine itself, and a generalized device driver. The extension of this concept to the logical effector is straightforward. Taken together, logical sensors and logical effectors are called logical devices. A single logical device can be composed of multiple physical
devices, with appropriate drivers. This would be desirable in cases where the physical devices were virtually identical (except perhaps in physical location, scaling, or some other trivial factor), allowing the main driver to give the appearance of a single effective logical device.

![ANIMA structure](image)

These processes (*Reasoner, Integrator, Coordinator, and representative logical devices*) have been described using the notation of Communicating Sequential Processes (CSP) [19, 20]. By combining them into a system, it is possible to construct a proof showing that the system is free of deadlock [14]. By following basic design principles at the logical device layer and the *Integrator/Coordinator* layer, we provide a means of fault isolation to individual logical devices.
2.2.2. Hardware implementation

Implementations of this architecture have been developed based on the Inmos T800, a member of the Transputer family of microprocessors developed for parallel processing. Each Transputer provides four high-speed serial links for the required processor interconnection.

The ANIMA architecture requires no parallel data busses or backplanes of any kind. Instead, it consists of modular components connected only by high-speed serial links. This allows the processors to be distributed to any convenient locations within the intelligent machine.

Relatively early in the architectural development, the entire structure was simulated on a single Transputer and on multiple Transputers in order to verify its operation. A fundamental premise of the simulation was that most of the processes would be directly portable to a real machine. Specifically, by making reasonable models of the environment, sensors, and effectors, one can use essentially the same Integrator, Coordinator, and Reasoner processes as would be used on a real machine [14]. The simulated machine wandered through a simple world with walls and obstacles, using simulated sonar and touch sensors. Sensors and effectors were deliberately modeled as being imperfect, and the machine (as part of its Reasoner) had to maintain its own model of the simulated world.

2.3. Case studies – “Buzz” and “Stimpy”

AuRA and ANIMA were first brought together in the development of a machine called “Buzz,” to compete in the first robot exhibition and competition sponsored by the American Association for Artificial Intelligence (AAAI). The competition stressed the ability of mobile robots to explore an arena, avoid static or moving obstacles, locate...
goals, and visit goals in specified order. Information about Buzz and the competition may be found in [15] and [8]. Buzz required an onboard PC to host its network of Transputers. To address more directly the needs of WSRC, a compact, low-power, onboard implementation was developed and used in a robot called "Stimpy." Like Buzz, this version was designed for one of the Denning commercial robots (an MRV-2), and it was used in an AAAI competition (held in July 1993 in Washington, D. C.). As part of this effort, the performance of ANIMA was investigated with a robot that had dedicated channels to the sonar and motor systems. This eliminated the need to multiplex the data on a single RS-232 line, and overall system performance improved considerably.

The Denning MRV-2 is a three-wheeled cylindrical robot intended for general-purpose use, mainly in research. All three wheels turn simultaneously, providing (approximately) the ability to turn in place. The body itself does not rotate, except for gradual precession resulting from non-uniform slippage of the wheels against the floor. Twenty-four sonar sensors are equally spaced around the body.

Stimpy was required to perform three tasks designed to resemble activities in an office environment. Only the first two tasks were performed with an ANIMA controller. Lack of development time prevented the implementation of the third task on ANIMA, and a remote Sun controller was used instead. The first task ("Escape from the Office") was an exercise in escaping from a room cluttered with office furniture. This required avoiding obstacles while searching for an open door, exiting as quickly as possible, then crossing an obstacle-strewn open area to a finish line. The second task ("Office delivery") required that the robot first interpret a geometric description (map) of a typical office environment, then find a marked object (a teapot) at an unknown location within this environment.
Additional enhancements for this version of ANIMA included

- a host-less implementation,
- EEPROM storage of all code, and
- “pass-through” communication with remote systems.

Unlike the previous competition robot, no ISA personal computer system was carried onboard. All software development was performed on a separate ISA PC, and the compiled code was downloaded through a quick-disconnect port on the robot. This reduced system weight, fragility, and power consumption. A board containing EEPROM was added, enabling the downloaded code to (optionally) be stored onboard the system. By changing some jumpers the system would then execute the stored code after any reset. The pass-through communication utilized one of four serial ports to provide a link with a remote system. Since all control was performed onboard, this feature was used only for relaying telemetry and for providing startup menus. Two of the other serial ports, as noted earlier, were used for ultrasonic perception and motor control, respectively. The third serial port was connected to the laser barcode scanner, which became another logical sensor within the system.

2.4. STD Host Adapter

Integration of the Transputer Control System with the Denning MRV-2 was achieved by building two STD host adapters. These adapters meet the specifications for Transputer Module (TRAM) motherboards provided by Inmos [21] and allow for the mounting of up to four TRAMs on each adapter. The TRAM format has been defined by Inmos to cover various-sized modules that interconnect through Transputer-compatible links. The basic size of a TRAM is 3.66” by 1.05” (referred to as a Size 1 TRAM) with 16 pins for electrical services and mechanical retention. The largest TRAM, a Size 8, is 3.66” by 8.75” with 128 pins for retention, of which only 16 are used by the TRAM for electrical
services. On TRAMs larger than Size 1, the extra unused pins are passed through the module so that other TRAMs may be piggybacked to avoid the loss of TRAM sites. In this fashion, the STD host adapter could support one Size 4 TRAM plus three Size 1 TRAMs. However, in no cases can more than four TRAMs be mounted on a single STD host adapter.

The host adapters were constructed on STD wire-wrap prototyping boards with the STD bus connector providing only power and ground connections (see schematics at the end of this section). Each TRAM slot has two links brought to a 96-pin connector and the other two links interconnected among the other TRAMS in the following manner:

Slot 0 Link 2 connected to Slot 1 Link 1
Slot 1 Link 2 connected to Slot 2 Link 1
Slot 2 Link 2 connected to Slot 3 Link 1
Slot 3 Link 0 connected to Slot 0 Link 3

The system services signals (reset, analyse, and error) may be daisy-chained between TRAM motherboards by attaching the Down System Services from one board to the Up System Services of the next. A processor in Slot 0 of a motherboard may act as the root processor by selecting its Subsystem Service port via jumpers J1 and J2. Link speed settings are specified for each TRAM on the motherboard by jumper J3.
<table>
<thead>
<tr>
<th>Jumpers J1 &amp; J2</th>
<th>System Services Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>pins 1 &amp; 2 connected</td>
<td>Up System Services</td>
</tr>
<tr>
<td>pins 2 &amp; 3 connected</td>
<td>Slot 0 Subsystem Services</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Jumper J3</th>
<th>Link speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>off</td>
<td>20 Mbit/second</td>
</tr>
<tr>
<td>on</td>
<td>10 Mbit/second</td>
</tr>
</tbody>
</table>

A two-pin header is provided for a normally-open momentary contact reset switch. This reset is active regardless of the settings for jumpers J1 and J2.
Title

96-pin edge connector pinout

A  1  GND  cut short  GND
  2  VCC  cut short  VCC
  3  cut short  cut short  cut short
  4  VCC  cut short  VCC
  5  GND  cut short  GND
  6  sleeved  sleeved  sleeved
  7  GND  GND  GND
  8  cut short  cut short  cut short
  9  Slot0Link1 Out  Slot1 Link0Out  Slot3Link20ut
 10  Slot0Link1 Out  Slot1 Link0Out  Slot3Link20ut
 11  Slot0Link1 Out  Slot1 Link0Out  Slot3Link20ut

STD Host Adapter - Connectors

Size

A-9464-000-001

Using vision to guide a mobile robot has many difficulties. Among these is the problem of reducing a large amount of pixel-based information into a smaller representation that lends itself to computer-based reasoning. We have investigated two approaches developed by the University of Massachusetts while performing research for the DARPA Autonomous Land Vehicle Project [21]. Both of these algorithms, the Fast Line Finder (FLF) and the Fast Region Segmenter (FRS), rely upon selective processing to achieve reasonable performance levels in the real-time environment of a mobile robot. Selective processing is derived from a priori knowledge of the robot's environment and goals and can be used to limit the processing of visual information to regions of high interest. In the FLF research performed here, areas of interest are confined to regions of the visual field in which pathways are likely to be found. Additional selection is performed by processing only objects whose features, such as line orientation, match the constraints necessary for visual navigation.

3.1. The Fast Line Finder

While the following discussion attempts to provide an overview of the Fast Line Finder, a more complete treatment can be found in the original paper [21] on which this summary is based.

The Fast Line Finder algorithm, based upon Burns's algorithm [13], works by extracting a set of image intensity gradients for each pixel in an image and groups pixels with related orientations and positions together. From these groups, lines can be developed. In this research, these lines are used to define speculative road edges from which a path can be derived.
The following steps are taken by the Fast Line Finder algorithm to generate a set of lines that are of use in vehicle navigation:

1. Computation of the image intensity gradient (magnitude and direction) for each pixel
2. Quantization of pixel direction into a set of orientation ranges.
3. Grouping of adjacent pixels of like orientation into regions
4. Fitting of lines to match regions

The Fast Line Finder optimizes the processing of these steps by ignoring irrelevant pixels during the early stages and irrelevant regions during the later stages.

3.1.1. Gradient computation

The Fast Line Finder computes both gradient magnitude and direction for each pixel in the image. As mentioned above, the gradient direction of the pixels is used in the determination of pixels that may contribute to a line of a particular orientation. Small gradient magnitudes indicate pixels that lie in the interiors of regions and do not contribute to the generation of productive line segments. These pixels can be eliminated from further processing, significantly reducing the computations required by the next steps in the algorithm.

Gradients are computed using the Prewitt operator convolution mask to determine the derivatives of the image intensity in the $x$ and $y$ directions. For a $3 \times 3$ operator, the mask would be $[-1 -1 -1] \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ for computation of the row derivative ($I_y$), and $[-1 0 1] \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ for computation of the column derivative ($I_y$). The gradient magnitude ($m$) would be exactly
computed by $m = \sqrt{I_x^2 + I_y^2}$. However, the FLF implementation uses a conservative approximation $m = |I_x| + |I_y|$ for this value, which is much faster to calculate. Analysis performed by Kahn, *et al.* [21] indicate that this approximation produces negligible errors for this application. The computation of angular direction for each pixel is derived from a conversion to radians by $\alpha = \arctan(I_y / I_x)$. To avoid the computation of the division and trigonometric function, a lookup table provides a coarsely quantized angular direction.

3.1.2. Bucket classification

Quantizing the angular direction is the first step toward classifying pixels into similarly oriented *buckets*. The quantization is controlled by user-defined parameters that can greatly affect the quality of the lines produced and the speed at which they are generated. Buckets are formed by limiting the gradient angles to a small set of regions; for example, eight buckets would be formed by quantizing to $\pi/4$ radians. Each pixel is assigned a label designating the bucket into which it has fallen. At this stage, pixels that fall into buckets that lie outside the range of possible interest can be marked as irrelevant, further reducing the amount of computation required by the next stage (pixel grouping).

3.1.3. Pixel grouping

Pixel grouping is accomplished through the application of widely used component connection algorithms. The FLF implementation associates nearest neighbor (above, below, left, and right) pixels that have identical bucket labels. As the pixel map is processed, lists of pixel locations for each distinct region are created. Statistics on region extents are tabulated and filtering can be done on regions with too few pixels.
3.1.4. Line fitting

Fitting of lines to the regions formed by the pixel grouping step is performed by viewing the regions as data sets to which lines must be fitted. A discussion of the method used by the FLF implementation must be deferred to Kahn et al. [21].

![Figure 3-1: Example of an accurate FLF centerline determination.](image)

3.1.5. Implementation

For robot navigation, the lines provided by the Fast Line Finder must be accompanied by a control system that can interpret the results in a useful manner [9]. In this system, the output of the FLF is used to drive a process that attempts to determine path edges and path centerline. The path edges are derived by applying the robot's knowledge about its current position and approximate position of the path from the last image. Using this information, the control system can make a prediction as to the location and orientation of
the path edges in the current frame. This prediction is used to tune the FLF to produce
sets of lines that can be used as fragments for candidate road edges. Fragments are
considered by first filtering out those line segments whose position or orientation does
not match the expectations for either of the path edges. Two sets of fragments, one for
each path edge are produced, from which line equations are generated. From these left
and right edges, a center line is computed which provides a basis for the generation of a
new directional heading for the robot.

Figure 3-1 shows the operation of the FLF under good conditions. Because the FLF has
produced a majority of line fragments that closely match the aisle boundaries, the
centerline is accurately computed. Due to orientation filtering, relatively few lines are
generated from the drum images, even in the regions occupied by the rectangular labels.
As will be discussed in more detail later, the most useful images are acquired when the
camera is pointed steeply downward, as seen here. Ideally, none of the field of view is
used for extraneous information above the horizon. Also note that these images are
printed as halftones, rather than continuous shades of gray.
As shown in Figure 3-2, interference from extraneous fragments can cause the FLF performance to deteriorate. In this image, unwanted fragments pull the left aisle boundary toward the edge of the frame. However, even in this case, the errors are not navigationally significant. Furthermore, the estimated boundaries are close enough to adequately predict their position in the next sequential image.
3.2. The Fast Region Segmenter

Like the Fast Line Finder, the Fast Region Segmenter (FRS) was also developed at the University of Massachusetts. Rather than attempt to extract lines from the image, it extracts connected regions of similar color or shading (if monochrome). The general approach is as follows:

1. determine a range of interest in the image to be processed,
2. categorize pixels into intensity classes,
3. group adjacent equivalent-class pixels into labelled regions,
4. select the candidate region that represents the expected aisle floor,
5. fit boundary lines to the region, and
6. determine an expected center line for the aisle.

The first step is to determine the extent of the image to be processed. As will be discussed below, this study generally bypasses this step and processes the entire image. The next step is to categorize each pixel into one of a set of predetermined intensity classes. For the monochrome images considered in this study, the full eight-bit range was divided nearly evenly into various numbers of classes. By using fewer classes, more image detail is lost, combining pixels into larger regions. The FRS supports uneven class sizes, such as making one range encompass only 10 intensity values while another encompasses 50, but this was not necessary for test images considered here. Satisfactory results were achieved with as few as three to five evenly-sized classes. When many more classes are used, the regions of interest tend to be broken up by variations in shading. Figure 3-3 shows a typical image and Figure 3-4 shows the resulting classified image, where five even classes were used. The regions have been shaded at evenly-spaced intervals to make them visually distinct. As before, note that all shades of gray (both in
the original image and in the classified image) are represented by halftones for printing purposes, and the image quality is not representative of what would actually be seen on a monochrome display. In particular, this sometimes makes it difficult to distinguish between an isolated black pixel and the black portion of a halftone representation of a gray shade. The text description will attempt to clarify this in the few cases where it is a significant factor.

Figure 3-3: Representative unprocessed image.
Once the image pixels have been classified, a connected-components algorithm is used, much as was done with the Fast Line Finder. In the case of the FLF, this was done to combine pixels that supported lines of similar orientation. For the FRS, neighboring pixels within the same class are labelled as belonging to the same region. This requires the formation of fragments that are successively merged to produce the largest-possible contiguous regions. The resulting region-labeled image, or RLI, will generally contain more regions than there were classes in the classified image, since islands of the same class are labeled as different regions. In the example shown in Figure 3-5, this is clearly the case. Note also that some pixel groups are not labelled as regions at all, since they are not large enough to warrant attention. The minimum region size is a variable parameter in the FRS, and it has been set to 30 for this entire study, since this seemed small enough to include all regions of possible interest, but not so small that too many regions were
created. In Figure 3-5, the solid black pixels are in unlabeled areas. Some of these are isolated pixels, as in the aisle itself, and others are small clusters of different classes, none of which are contiguous and large enough to warrant designation as a region. This is evident in the drum labels on the right-hand side of the aisle.

![Figure 3-5: Region-labeled image (RLI) of previous figure.](image)

The next step is to select a candidate region for the aisle. Several approaches are possible. Just as with the FLF, the images taken with the camera angled steeply downward provided more information, since the camera field-of-view was not wasted on imagery above the horizon. With this vantage point, the aisle surface should begin at a point low in the image frame, as long as the camera is not pointed directly into an aisle wall (in which case little information could be extracted from the visual information, anyway). Depending on how far off-center the robot is pointed, the aisle surface will extend either up or to one side. This leads to one approach that seems to provide good
results. If the camera is positioned so that a given pixel, say the center pixel, is within the aisle surface for most situations, the principal method of region selection is to choose the region occupied by this center pixel. Another possible approach is to choose the largest region, which works well much of the time, as long as the camera is still pointed downward. For higher camera angles, it becomes necessary to exclude the image above the expected horizon.

Experimentation with the actual robot in the actual surroundings should provide additional insight into the best means of region selection. This study indicates that a reasonable starting point would be to start with the center-pixel’s region, unless some detectable condition indicates otherwise. One such indication would be the size of the center-pixel region — if it does not exceed some fixed percentage of the image frame, an alternative region should be selected. The fall-back selection in this case could be the largest region found below the expected horizon. In those cases where the center pixel is not part of a labelled region (i.e., it belongs to an insignificant cluster), this study used the simple strategy of selecting the largest region.

Once a region is selected, the FRS continues by noting the leftmost and rightmost extents of the region in each row of interest. This results in two arrays of pixels that represent the left and right boundaries of the region. Each of these arrays is fed into a least-squares algorithm, and a straight line is fitted to each side of the region. It is possible to limit the rows of interest to a subset of the rows actually encompassed by the region. Experimentally, this was shown to be useful, since the regions tend to show rounded tops and bottoms. These rounded areas inject outlying points into the least-squares fit, pulling the linear boundaries inwards (toward the center of the image). Clearly, the robot would need a consistent rule in order to avoid this phenomenon, and it appears reasonable to always exclude the upper and lower 10% of each side boundary.
Midway through the research, WSRC indicated that the Transputer-based control was not going to fit into their immediate plans. Since the FRS analysis was performed in the latter part of the research period, no effort was made to port the algorithm to the Transputer system, as was done with the FLF. Instead, all available time was expended on evaluating the FRS on a Sun workstation. Like the FLF, however, the FRS was converted from VAX-specific C to ANSI C, so the algorithm can be easily ported to the Transputer or some other processor at a later time.

The remainder of this section illustrates the performance of the FRS on actual images, most of which were taken from the videotape produced at the Fernald facility. First, some of the potential failings are examined, providing some justification for the specific parameters that are eventually recommended. Then a collection of frames is processed with fixed parameters to demonstrate the expected performance of the FRS as it now stands in comparison with the FLF.

3.2.1. Classification parameters

Figure 3-6 shows the result of classifying an image into an excessive number of classes (ten). Although ten classes does not seem like very many classes, it is more than enough to cause normal variation in shadow, as well as floor markings, to alter the perception of the aisle floor. Figure 3-7, on the other hand, uses only five evenly-spaced classes. This image is somewhat typical, in that the aisle is still divided among multiple classes, but the splitting is more predictable, usually occurring near the edge (in deep shadows) or toward the horizon. The variation toward the horizon can be due to reflections, variation in ambient light, or the decreasing illumination with respect to distance of the robot’s onboard light source. Figure 3-8 shows the effect of reducing the number of classes to only three. Surprisingly, there is still a large amount of information contained in the image. In particular, the aisle floor is clearly visible.
Figure 3-6: Ten-level classification.
Classification mappings of greater complexity can be used. For example, it would be possible to determine a range of shades that encompass the typical aisle floor and to design a class specifically for the expected range. Remaining intensity values could then be split arbitrarily among two or more classes. Unfortunately, this would not be a robust scheme, since it would be more susceptible to lighting variations. If color vision were used, however, this would become much more practical. Since the waste facility environment is structured, it would be relatively easy to give the floor a distinctive color. Then, a tailored classification mapping becomes feasible, since the floor class could encompass the specific hue and saturation, with a large amount of latitude in intensity (to accommodate lighting variations).
3.2.2. Area-of-interest limiting

The FRS can be limited to only processing a certain rectangular subregion of the image. This is most useful for images such as that of Figure 3-9, where the camera tilt angle causes the horizon to appear well below the top of the image. The information above the horizon is extraneous, and it may result in a large contiguous region that could confuse the algorithm during region selection. Assuming the camera tilt is known, the approximate position of the horizon will also be known, and the extraneous visual data can be ignored. This also results in a savings of processor time. In order to best utilize the camera, however, this research concentrates on images taken with most or all of the frame below the horizon, so upper/lower limiting will not be utilized.
Limiting the left and right extents of the image is usually somewhat suspect. One can proceed much as was done with the FLF, where previous knowledge of the aisle position can be used to provide an expectation of the current aisle position. Then, parts of the image to the left or right could conceivably be excluded. The FLF, however, seems to be more suited to this sort of expectation-driven processing as it steers toward the vanishing point of perspective. The FRS may serve better as a means of "looking at one's feet" for the path, then looking away for the apparent extension of the path. In this mode, the FRS can serve as a complementary algorithm to the FLF by providing bootstrap input at startup, supporting evidence during normal operation, and fall-back steering when the FLF is completely unable to find aisle edges.
3.2.3. Region selection

Some discussion has already been presented about the two principal means of region selection:

- largest-region selection, and
- position-dependent selection.

The automatic selection of the largest region does not consistently result in desirable aisle-following behavior. As shown in Figure 3-10, the largest region can easily be a drum or a group of neighboring drums, or possibly a shaded region of the path that is off-center. This is the first image in which a selected region is marked. The region is outlined in white. Note that this particular region is a highly convoluted area with many islands of non-included pixels. The marked region is NOT the lighter central portion of the aisle, but rather the darker outer regions of the aisle, combined with the similarly-shaded light drum on the left.

This is also the first image to show the fitted left and right boundaries, which appear as straight white lines. In this case the right boundary is reasonably accurate, while the left boundary is pulled toward the drum, as would be expected.
Position-dependent selection appears to be more feasible. For the preferred camera tilt, the center pixel of the image (pixel $128,128$ in a 256-by-256 image) is usually in the path, even if the robot is pointed somewhat off-center. For other camera tilts, some other appropriate pixel may be identified. Since the FRS retains lookup information about the region that each pixel is a member of, it is possible to select the region on this basis. If the candidate region appears unpromising because it is too small or has boundaries that are not believable as potential road edges, several fall-back options would be available for the robot at run time. First, it could look for the nearest candidate regions to the left or the right. If neither of these appeared promising, the largest region could be selected by default. Finally, an underlying assumption is that alternative sensor data is available. In certain situations, less confidence should be placed in the FRS determination, and more
confidence may be placed in dead-reckoning, other vision algorithms (including the FLF), ultrasound, and other sensors.

The analysis up to this point has focused on several of the difficulties in the image processing. In reality, though, good results are usually achieved with the FRS. While it is helpful to define the intensity classes for optimal inclusion of aisle pixels, it is certainly not necessary. A good example is shown in Figure 3-11. Here, the obvious region is selected simply on the basis of size, which results in the large, convoluted region shown. The boundaries of this region clearly provide an accurate representation of the aisle. As in all of the previous images, the algorithm is insensitive to the irrelevant details, such as the drum lettering and the superimposed camera clock.

Figure 3-11: Successful location of aisle in spite of region fragmentation.
3.2.4. Boundary limiting

Independent of the area-of-interest limiting already discussed, the FRS allows the exclusion of specific rows during the left and right boundary-fitting process. Figure 3-12 shows an example of an image where the boundaries are pulled inward severely by points near the top or bottom of the selected region. This is easily minimized by automatically excluding about 10% of the pixels at the top and bottom of each side. The result of this exclusion for the same image is shown in Figure 3-13. The boundaries are still too far toward the center of the aisle, which is to be expected considering the region that was selected, but the resulting center line will be a fairly accurate navigation heading.

Figure 3-12: Inward pull associated with narrowing region.
Figure 3-13: Partial correction by automatic exclusion of upper and lower rows.
4. Testing

The environment for the mobile robot application can be generally described as rows of closely spaced pallets, each containing stacks of drums. The rows are separated by aisles that are large enough for the robot to pass through (about 1 m). It is required that the robot be able to travel down these aisles without colliding with any drums or pallets. The drums vary in size, and they may extend past the edge of the pallets. The resulting visual boundary (at the floor) is a series of line segments and curves, sometimes interrupted by gaps. In order to provide steering cues, the visual algorithm must be able to identify the approximate left and right boundaries, in spite of their deviation from straight lines, and project an imaginary center line.

Two methods were established for testing of the vision system and related algorithms. The first setup used videotape images from Fernald that simulated motion of a robot through the waste storage facility. These images were digitized and stored for processing by both a Sun workstation and the Transputer-based vision system. The second test setup consisted of simulated environment constructed from 50-gallon drums and pallets assembled into aisles. These aisles were navigated by a camera-equipped Denning robot driven by a remote Sun workstation running an image processing algorithm. Images from Fernald are used in the remainder of this section, and videotape of robot performance in the simulated environment has already been provided to WSRC.

For demonstration and testing purposes, a Transputer-based three-axis motor controller was developed to drive a pan-tilt-vergence camera platform. The implementation of this controller demonstrated the division of labor achievable with parallel-processing systems. The motor controller runs on a separate Transputer and interfaces with the 3-axis platform via an RS-232 serial channel. Communication with the rest of the Transputer network is through the high speed Transputer links. This real-time interface was added
without impacting the performance of the processor-intensive vision computations. Demonstration of the entire control system was achieved by servoing the camera to look in the direction of the computed aisle centerline. In these tests, the vision system did not use images acquired from the camera, but was instead driven by previously acquired images from the Fernald facility.

4.1. Sample Images

The following series of images illustrate the performance of both the FLF and the FRS on the same set of input data. By directly comparing them, one can see how the algorithms perform relative to each other. These images were chosen instead of other Fernald images primarily because the lighting consistency is better in the fabric structure facility. Good results would also be expected if the robot carries a sufficiently bright light, but careful consideration would have to be given to positioning the light to avoid distracting shadows. In fact, it is quite possible that better performance would be achieved with directed onboard lighting and minimal ambient lighting, but at this early stage of the analysis, even illumination eliminates several extraneous experimental variables.

For the FRS examples that are presented, no a priori knowledge of the aisle position is provided to the algorithm. For the FLF examples, almost 30% of the images (Figures 4-6, 4-7, 4-18, 4-20, 4-21, 4-23) resulted from only knowing that the aisle was approximately in the center of the image. For the remaining FLF examples, the algorithm was provided with expected aisle edge locations that were within about 20% of the true locations.
Figure 4-1.
Figure 4-3 is a clear failure of the FRS to find an appropriate representation of the aisle, similar to the example presented earlier in which a light-colored drum blended into the aisle. In this earlier example, it was noted that a different region selection would have solved the problem. Here, however, it is less clear exactly how a simple heuristic could be used to select the region in the upper left. The most obvious suggestion is that regions broken horizontally by large gaps (like the downward wedge-shaped gap above) should be treated with suspicion. If a secondary selection is not apparent, the FRS output should be disregarded in favor of the FLF or other sensory inputs.

It is also tempting to consider classification into more than five shade classes, in the hope that a distinction will emerge between the drum and the path. Figure 4-4 illustrates the result for ten classes. This does in fact separate the drum from the path, but the path is so broken up that it is not possible to clearly distinguish the left boundary.

Figure 4-4.
Figure 4-5.
Figure 4-6.
While the path edges shown in the FRS example of Figure 4-7 will in fact produce a center line that approximates the aisle direction, it can hardly be considered to be the desirable result. The alternate choice of the largest region instead of the region occupied by the center pixel may be made here, producing the result in Figure 4-8. Such an alternate selection could be made automatically in any case where the perceived aisle edges diverge, as they do in the previous example.
Figure 4-10.
Figure 4-10 once again shows the problem occurring when the FRS encounters a light drum extending into the aisle. As before, a ten-level classification is attempted, as shown in Figure 4-11, and the results are similarly disappointing. These two examples indicate the reason that only five classes were used for most of this analysis: the additional levels of distinction do not enhance the representation of the world to the FRS.
As would be expected, the FRS is unable to extract any meaningful aisle representation for the open area where a pallet is missing in Figure 4-17. The FLF, on the other hand, makes use of the line fragments generated by the floor marking and is able to construct the aisle boundaries accurately.
Figure 4-18.
Although not apparent from the results shown, Figure 4-20 is the same example that was presented in the earlier section on region selection for the FRS. When the largest region is selected for this image, the blending of the light-shaded drum again causes problems. This was one of the reasons that the center-pixel criterion was used as the primary means of region selection for all of these images. Here, it results in a region that does not represent the aisle very well, but at least no blending occurs, and the centerline of the apparent aisle is not too far from the actual centerline. As in other similar cases, however, the FLF performs more reliably.

Figure 4-21 illustrates the same phenomenon. Here, the divergence of the lines is once again a clue that some other region selection method should be attempted, although in this case it would result in another blending problem, similar to that seen next in Figure 4-22.
Figure 4-23.
In the last example (Figures 4-23 and 4-24), both the FLF and FRS performed fairly well, but the left aisle boundary formed by the FRS appears to somewhat inaccurate. This is due to a problem with the FRS that can be corrected. Whenever the outermost (to either side) pixels of a region are actually on the image edge, the corresponding row should be discounted for the purpose of determining the aisle boundary on that side. If this were done, only the boundary pixels near the top of the image would be used to determine the left line, which is appropriate since the line goes off the edge of the image. To some extent, this would have helped several of the other example images, but not so noticeably as in this case.
5. Conclusions

Both the Sun and Transputer implementations of the FLF and FRS ran in under six seconds per frame. Since only a single Transputer was used for the image processing, significant speedup could be achieved by using additional processors, without adding significantly to the size or power consumption (the frame grabber is already the largest single component). In order to perform thorough radiation detection, the robot's speed would be relatively low, which constrains image processing speed more than the on-board hardware. Since Transputers were used, the architecture can easily be expanded to achieve higher frame rates if required. An example of an obvious parallelization would be to use a different processor for the left and right boundary extractions.

Testing of the FLF algorithm with images of stored hazardous waste showed that this implementation is robust and well suited for autonomous navigation. Use of aisle-boundary extraction techniques appears to provide suitable navigational cues and is tolerant of variations in the storage environment. Nearly every failure of the FRS could be detected by divergence or near-parallel-orientation of the detected aisle boundaries. In many cases, this could also be supplemented by detection of significant gaps (moving horizontally) in the selected aisle region. In some of these failure instances, selection of an alternate region allows the FRS to perform satisfactorily, while in others it seems that alternate strategies are required.

The FLF, on the other hand, always found a reasonable approximation to the path, as long as it has a reasonable expectation of where to look. Normally, this expectation arises from previous knowledge combined with dead-reckoning to account for position changes, but it is also possible to use the FRS as a means of providing bootstrap information to the FLF. The FLF appears to be a more reliable primary strategy, supplemented by the FRS for bootstrapping or confirmation.
This research has shown that a low-cost, high-performance vision system can be built using a modular parallel-processing system. Because of the modularity and communication techniques, portions of this system could be rapidly integrated into a completely different hazardous waste inspection system. Our research has not targeted the current SWAMI architecture explicitly, as this vision system is completely independent of the host architecture. For future work, this system could be integrated with SWAMI II simply by packaging it onto commercially available VME carrier boards. Additionally, this architecture supports both automated and manual visual inspection since the vision system (and its associated computing power) need not be dedicated solely to the visual navigation task. Standard image compression techniques could be easily adapted to this system with the parallel-processing ability providing the necessary computing power for real-time image archiving.
6. References


