Application Example: Motion Control

Motion Control
- is one of the most important tasks of mobile embedded systems
- is subject to real-time and reliability requirements
- heavily depends on the sensory input

- Single sensors have a partial, inaccurate view
- Distributed sensor fusion allows achieving a more complete and accurate perception of the environment
Example: RoboCup

- Getting wide acceptance as standard benchmark for team robotic
- Annual world and national championships in robot soccer
- Research done as part of a nationwide DFG-program „Cooperating teams of mobile robots in dynamic environments“
Example: RoboCup

Where is the ball?
Sensor Data Processing – Filters and Abstraction Levels

contours

contour filter

object filter

element filter

points

elements

Motion Control
Sensor Data Processing - Multi-Level Sensor Fusion

point fusion

contour filter

contour fusion

object filter

object fusion

element filter

element fusion

point fusion

contour filter

contour fusion

object filter

object fusion

element filter

element fusion
Environment-Dependent Execution Times of the Filter Modules

- Execution times depend on input size
- Execution times depend on input content
Environment-Dependent Execution Times of the Filter Modules (cont’d)

execution times of the arc filter

input for max. execution time
Environment-Dependent Execution Times of the Filter Modules (cont’d)

Execution times of the arc filter

Input for min. execution time
Contour Filter: Execution Time vs. Mean Distance

- Execution times > 57ms are only observed for distances < 32cm
Environment-Dependent Execution Times of the Filter Modules (cont’d)

execution times of the arc filter

→ Scheduling WCETs is not an acceptable solution to achieve a predictable timing behavior of the filter modules!
Functional Redundancy in the Arc Filter

- The arc filter evaluates potential ball positions
- Only the best estimate is relevant to higher layers (corresponds to ball)
Structural Redundancy in the Edge Filter

- Aborted instances deliver results for a fraction of the scene
- Incomplete results are valuable input for the sensor fusion
Structural Redundancy – Distributed Fusion

- Environment is observed by several, distributed sensors
- Fusion complements missing results from one sensor by results from others
Application-Level Adaptation

- Raising the fusion level allows reducing system load
  - Reduced execution times (CPU load)
  - Reduced data volume (network load)
- Raising the fusion level may decrease accurateness of results

<table>
<thead>
<tr>
<th></th>
<th>2 scanners</th>
<th></th>
<th>3 scanners</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>execution times [μs]</td>
<td>data volume [byte]</td>
<td>execution times [μs]</td>
<td>data volume [byte]</td>
</tr>
<tr>
<td>point fusion</td>
<td>10304</td>
<td></td>
<td>20427</td>
<td></td>
</tr>
<tr>
<td>arc filter</td>
<td>13157</td>
<td></td>
<td>63093</td>
<td></td>
</tr>
<tr>
<td>edge filter</td>
<td>1436</td>
<td></td>
<td>2216</td>
<td></td>
</tr>
<tr>
<td>object filter</td>
<td>2570</td>
<td></td>
<td>3708</td>
<td></td>
</tr>
<tr>
<td>element filter</td>
<td>77</td>
<td></td>
<td>98</td>
<td></td>
</tr>
<tr>
<td>sum</td>
<td>27544</td>
<td>11616</td>
<td>89542</td>
<td>17424</td>
</tr>
<tr>
<td>contour fusion</td>
<td>3398</td>
<td>3688</td>
<td>3398</td>
<td>3688</td>
</tr>
<tr>
<td>arc filter</td>
<td>577</td>
<td>578</td>
<td>577</td>
<td>578</td>
</tr>
<tr>
<td>edge filter</td>
<td>1955</td>
<td></td>
<td>3253</td>
<td></td>
</tr>
<tr>
<td>contour fusion</td>
<td>1636</td>
<td></td>
<td>2067</td>
<td></td>
</tr>
<tr>
<td>object filter</td>
<td>70</td>
<td></td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>element filter</td>
<td>7927</td>
<td>1384</td>
<td>11290</td>
<td>1912</td>
</tr>
</tbody>
</table>
Application-Level Adaptation: Accuracy Tradeoff
Application-Level Adaptation: Accuracy Tradeoff (cont’d)
Application-Level Adaptation: Accuracy Tradeoff (cont’d)
Application-Level Adaptation: Accuracy Tradeoff (cont’d)
Exploiting Inherent Redundancy

- To tolerate transient faults (short term peaks), use application-inherent redundancy
  - Functional redundancy within the module instances
    → Design modules as any-time algorithms
  - Structural redundancy within the executions of the sensor fusion
    → Combine results from distributed sensors
  - Time redundancy within a sequence of executions
    → Schedule a frequency above the minimum

- To tolerate permanent faults (persistent overload), use application-level adaptation (graceful degradation)
  - TAFT supports detection of persistent overload
  - Application-level adaptation allows lowering system load