Prototype of a Microsurgical Tool Tracker

Team 5

Students: Sue Kulason, Yejin Kim

Mentors: Marcin Balicki, Balazs Vagvolgyi, Russell Taylor

600.466 Advanced Computer-Integrated Surgery
Project Summary

- Problem: A need for tool tracker in eye surgery
  - Monitor surgical protocol
  - Assess surgical performance
  - Improve surgical safety

- Project Goal: Micro-Surgical Tool Tracker
  - Build a prototype of a goggle
  - Provide positional feedback

Figure 1. Idea proposed by Marcin Balicki
Project Summary
Paper Selection

Title: Fast Inexpensive Color Segmentation for Interactive Robots
Authors: James Bruce, Tucker Balch, Manuela Veloso
Published: IEEE International Conference Intelligent Robots and Systems, 2000

<table>
<thead>
<tr>
<th>Relevance to Project</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Color segmentation</td>
<td>✔</td>
</tr>
<tr>
<td>Fast (real-time)</td>
<td>✔</td>
</tr>
<tr>
<td>Computationally inexpensive</td>
<td>✔</td>
</tr>
<tr>
<td>No special equipment</td>
<td>✔</td>
</tr>
<tr>
<td>Easy to implement</td>
<td>△</td>
</tr>
<tr>
<td>Robust</td>
<td>△</td>
</tr>
</tbody>
</table>
Key Results

Problem:
- Real-time segmentation relies on specialized equipment

Solution:
- Utilize algorithmic efficiency
- Track 32 colors at 30 Hz

Importance:
- Make real-time segmentation affordable and accessible
- Doesn’t sacrifice accuracy for efficiency
- Various computer vision applications
Step 1: Color Threshold

- Extract YUV matrices
- **Why?** Robust against luminance

---

Figure from Bruce et al.
Technical Approach

**Step 1: Color Threshold**

- Classify pixel color with logical AND gates
- **Why?** 2 operations rather than 192

\[
\text{Pixel\_in\_class} = \text{YClass}[Y] \ \text{AND} \ \text{UClass}[U] \ \text{AND} \ \text{VClass}[V]
\]

- \(\text{YClass}[Y] = \{0,1,1,1,1,1,1,1,1\}\)
- \(\text{UClass}[U] = \{1,1,1,0,0,0,0,0,0,0\}\)
- \(\text{VClass}[V] = \{0,0,0,1,1,0,0,0,0\}\)

- \(\text{YClass}[Y] = \{00,11,11,11,11,11,11,11,11,11\}\)
- \(\text{UClass}[U] = \{01,01,01,00,00,00,00,10,10,10\}\)
- \(\text{VClass}[V] = \{00,00,00,01,01,01,00,10,10,10\}\)

Figure from Bruce et al.
Technical Approach

*Step 2: Connect Components*

- 2 runs of a tree-based union
- **Why?** Linear time bound

1. Scan row for adjacent pixels of the same color
2. Create disjoint forest of ‘runs’ with identifier for parent node
3. Scan runs for four-connectedness
4. Point each run’s parent node to global parent
5. Run second pass to compress path
Technical Approach

1: Runs start as a fully disjoint forest

2: Scanning adjacent lines, neighbors are merged

3: New parent assignments are to the furthest parent

4: If overlap is detected, latter parent is updated

Figure from Bruce et al.
Step 3: Density-based Merging

- Merge similar objects based on a grouping force threshold
- Why? Account for bottom up region generation error and occlusion

Steps:
- Find pairs of components
- Merge
- Calculate density
- Check against threshold

\[
\frac{\text{blue}}{\text{blue} + \text{green}} = 0.88
\]
## Analysis

### Significance
- Color segmentation
- Fast (real-time)
- Computationally inexpensive
- Cheap
- No special equipment
- Robust to luminance

### Changes for our Project
- Capture in RGB
- Transform to YUV
- No density-merging
Analysis

Tested Applications:
- Probotics Cye Platform
- RoboCup-99 Robot

Improvements/Future Directions

Applications beyond robot-soccer

Analysis of accuracy of blob detection

Description of hardware set-up

Figure from Bruce et al.
Bibliography

Paper Reviewed:

Clinical Background:

Questions?