CS 525M – Mobile and Ubiquitous Computing Seminar

Fingertip Cursor Control

by

Jeffrey R. Bacon

Background Info

- Ubiquitous Computing
  - Home Theater PC
  - Electronic Whiteboards
- User Interface
  - “A good tool is an invisible tool. By invisible, I mean that the tool does not intrude on your consciousness; you focus on the task, not the tool.” – Mark Weiser
- No mouse!
  - Environment is unsuitable
  - Can’t have dozens of mice in a room

Project

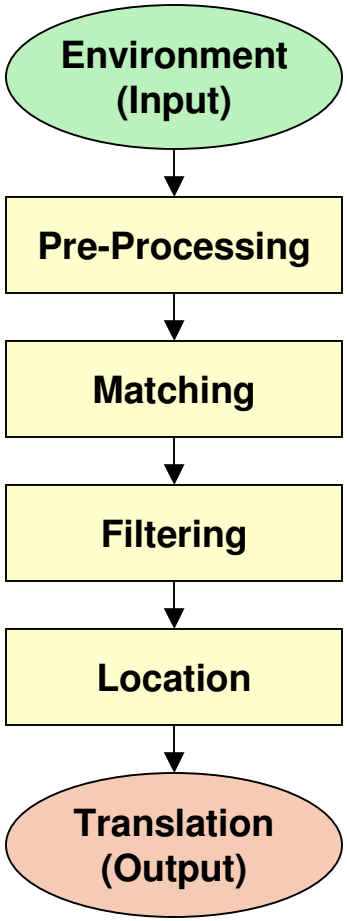
- Goal:
  - Cursor control using fingertips
  - Simple Point-and-Pinch gesturing
- Constraints:
  - Commodity hardware (“commodity” meaning “cheap”)
  - Color recognition
  - Budget: $0

Outline

- Architecture
- Program
  - Environment
  - Pre-Processing
  - Matching
  - Filtering
  - Location
  - Translation
- Results
  - Program Features
  - Algorithm Descriptions
  - Conclusion

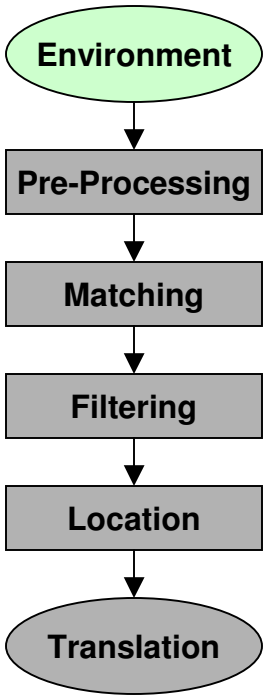
Pipe & Filter Architecture

- Multiple phases to recognition process



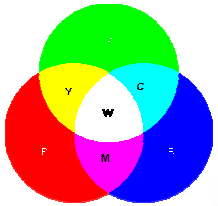
Environment

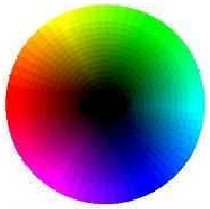
- Description:
  - Webcam
  - Colored Finger Caps
- Discoveries:
  - What we see is NOT what camera sees
  - Immense noise & color problem
  - Must be large or close, or else blocks are indistinguishable from noise
- Future Work:
  - Better camera settings & environment
  - More unique identifiers (LEDs, luminescent paint, etc)

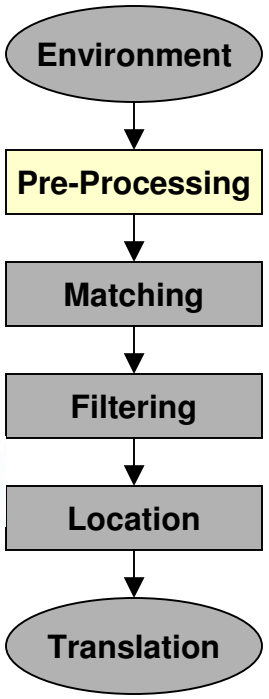


Pre-Processing

- Description:
  - What is done before any kind of recognition is done
- Successful:
  - RGB Color to HSB Color
- Unsuccessful:
  - RGB Blurring (expensive & ineffective)
  - HSB Blurring (expensive & ineffective)
- Future Work:
  - Noise Reduction
  - Other Color Encoding Schemes

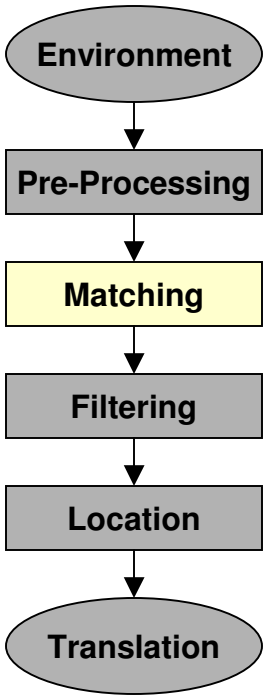






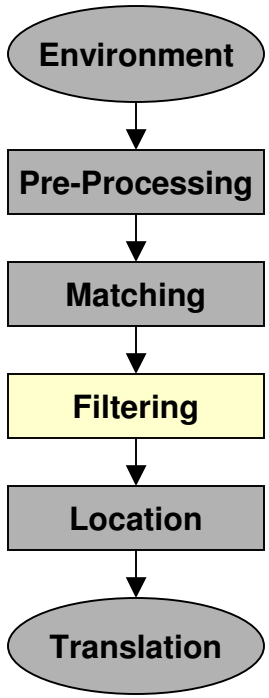
Matching

- Description:
  - Identifying pixels that are the appropriate color or similar
- Successful:
  - Accuracy Scores
  - Multiplicative accuracy (H×S×B)
- Unsuccessful:
  - Range matching
  - Additive accuracy
  - Blurred accuracy
- Future Work:
  - Better recognition methods
  - Combinations of methods



Filtering

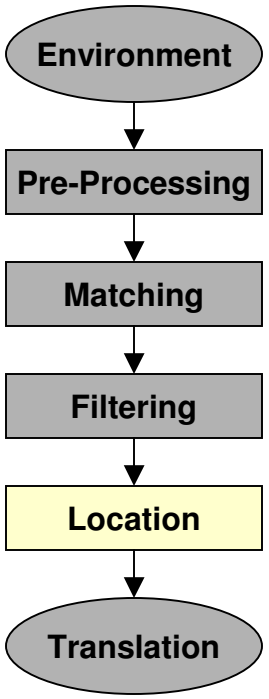
- Description
  - Altering of matched pixels after matching stage
- Successful:
  - Stray pixel reduction
  - Gap filling
- Unsuccessful:
  - Wide area calculations
- Future Work:
  - Faster methods
  - Larger radius survey
  - Based on accuracy



	0.764	1.000	0.764	
0.764	1.586	2.000	1.586	0.764
1.000	2.000	3.000	2.000	1.000
0.764	1.586	2.000	1.586	0.764
	0.764	1.000	0.764	

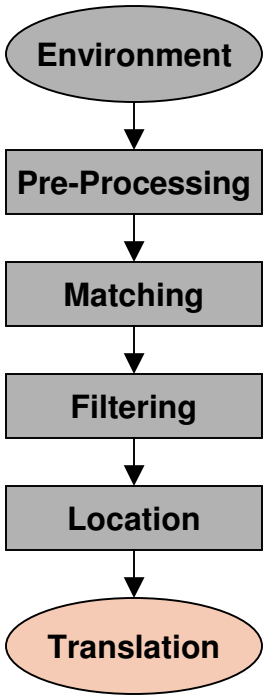
Location

- Description
  - Identifying location of cursor based on matched pixels
- Successful:
  - Cluster detection
  - Average of all pixels in cluster
- Unsuccessful
  - Banding peaks



Translation

- Description:
  - Translation of data to cursor actions
- Successful:
  - Jitter reduction by weighted averaging
  - Weighted distance averaging
  - Point-and-pinch gesturing
  - Boundary frame
- Unsuccessful:
  - Jitter reduction by plain averaging
- Future work:
  - Better jitter-reduction
  - Other gestures



Program Features

- Image Filters
  - HSB Matches
  - HSB Color Codes
  - Identified Pixels
  - Filter results (strays & gaps)
  - Clustering Information
  - Channels
- Adjustments
  - Camera Adjustments
  - Tolerance Adjustments
  - Recognition Adjustments
- Display
  - Numerical Information
  - Extra visual data

Algorithm Descriptions

- Algorithms:
  - Pre-Processing: RGB Color to HSB Color
  - Matching: Accuracy matching
  - Matching: Tolerance matching (failed)
  - Filtering: Stray reduction / gap filling
  - Location: Cluster identification
  - Location: Banding peaks (failed)
  - Translation: Jitter reduction
  - Translation: Cluster distances
  - Translation: Screen coordinates

Conclusion

- Color recognition cursor control…
  - Works!
  - … but isn't ready for prime time
  - … but is fun to play with anyway
- Key issues:
  - Lighting (affects everything)
  - Camera quality (colors, noise)
  - Processing power (may be prohibitive)
  - Configuration (usability issues)
  - Low resolution (low accuracy)
  - Low frame rate (slow response)

Questions?