Sign Language Recognition

version 0.2 | Byron Hood

Computer Systems Lab Project 2007-08

Overview

- **Average person typing speed**
  - *Composing: 19 words per minute*
  - *Transcribing: 33 words per minute*

- **Sign speaker**
  - *Full sign language: 200—220 wpm*
  - *Spelling out: est. 50-55 wpm*
  - *Faster by 1.5x to 3x*

Purpose and Scope

- **Native signers can input faster**
- **Benefits:**
  - *Hearing & speaking disabled*
  - *Sign interpreters*
- **Just letters & numbers for now**

Research

- **Related projects**
  - *Using mechanical gloves, colored gloves*
  - *Tracking body parts*
- **Image techniques**
  - *Edge detection (Robert's Cross)*
  - *Line detection (Hough transform)*
  - *Line interpretation (various)*

Program Architecture

- **Webcam interface**
- **Edge detection**
- **Line detection**
- **Line interpretation**
- **Hand attribute matching**

Testing Model

- **Human interaction necessary**
- **General testing model:**

```
bhood@bulusan: ~/syslab-tech $ \
   > ./main images/hand.pgm in.pgm

[DEBUG] Edge detect time: 486 ms
[DEBUG] Wrote image file to `in.pgm'

Errors: 0          Warnings: 0
```

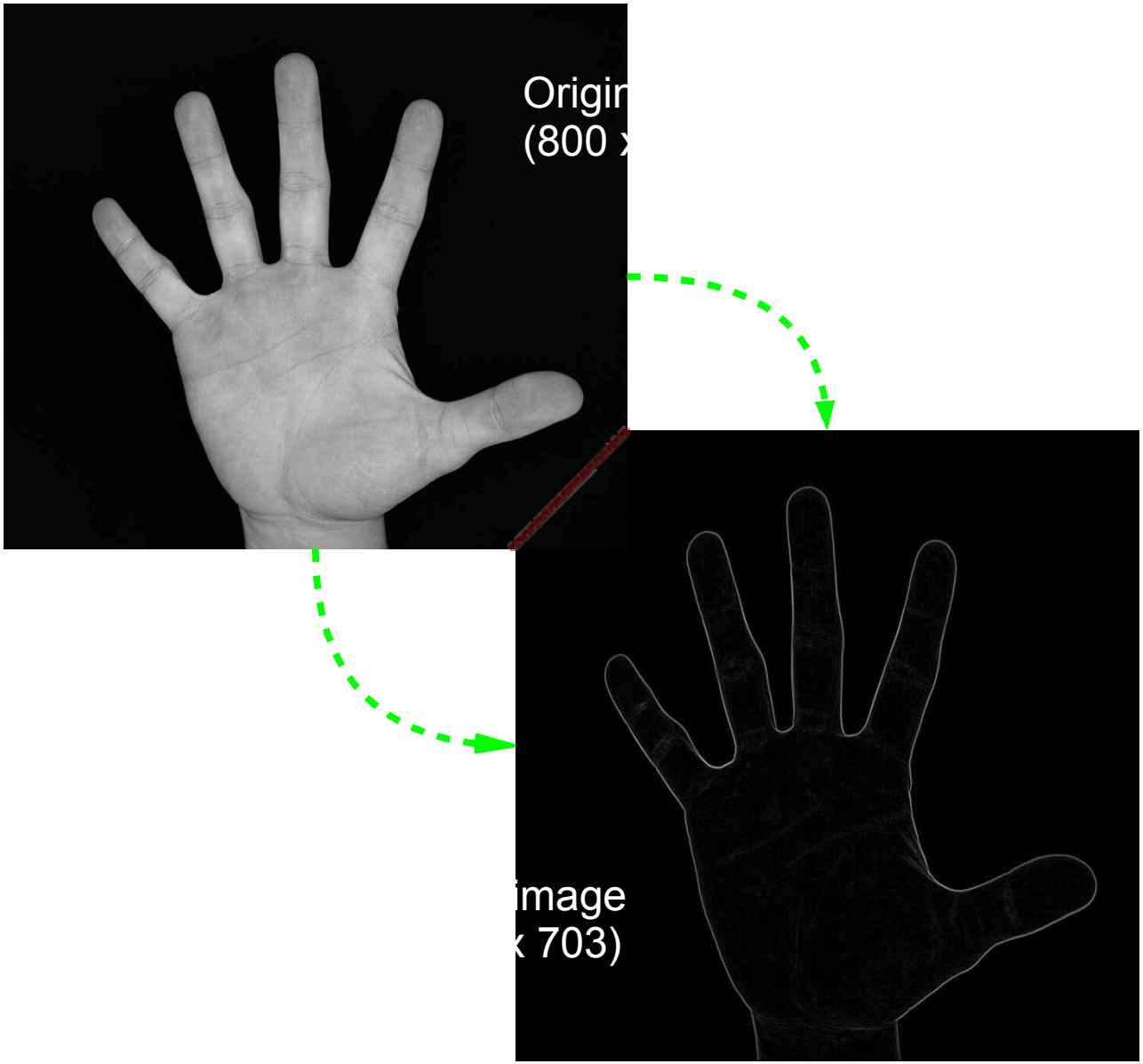
Code sample

- **Robert's Cross edge detection:**

```
int row,col;
// loop through the image
for(row=0; row<rows; row++) {
   for(col=0; col<cols; col++) {
      // avoid the problems from trying to calculate on an edge
      if(row==0 || row==rows-1 || col==0 || col==cols-1) {
         image2[row][col]=0; // so set value to 0
      }
      else {
         int left   = image1[row][col-1]; // some variables
         int right  = image1[row][col+1]; // to make the final
         int top    = image1[row-1][col]; // equation seem a
         int bottom = image1[row+1][col]; // little bit neater
         image2[row][col]=(int)(sqrt(pow(left - right , 2) +
                                     pow(top  - bottom, 2)));
      }
   }
}
```

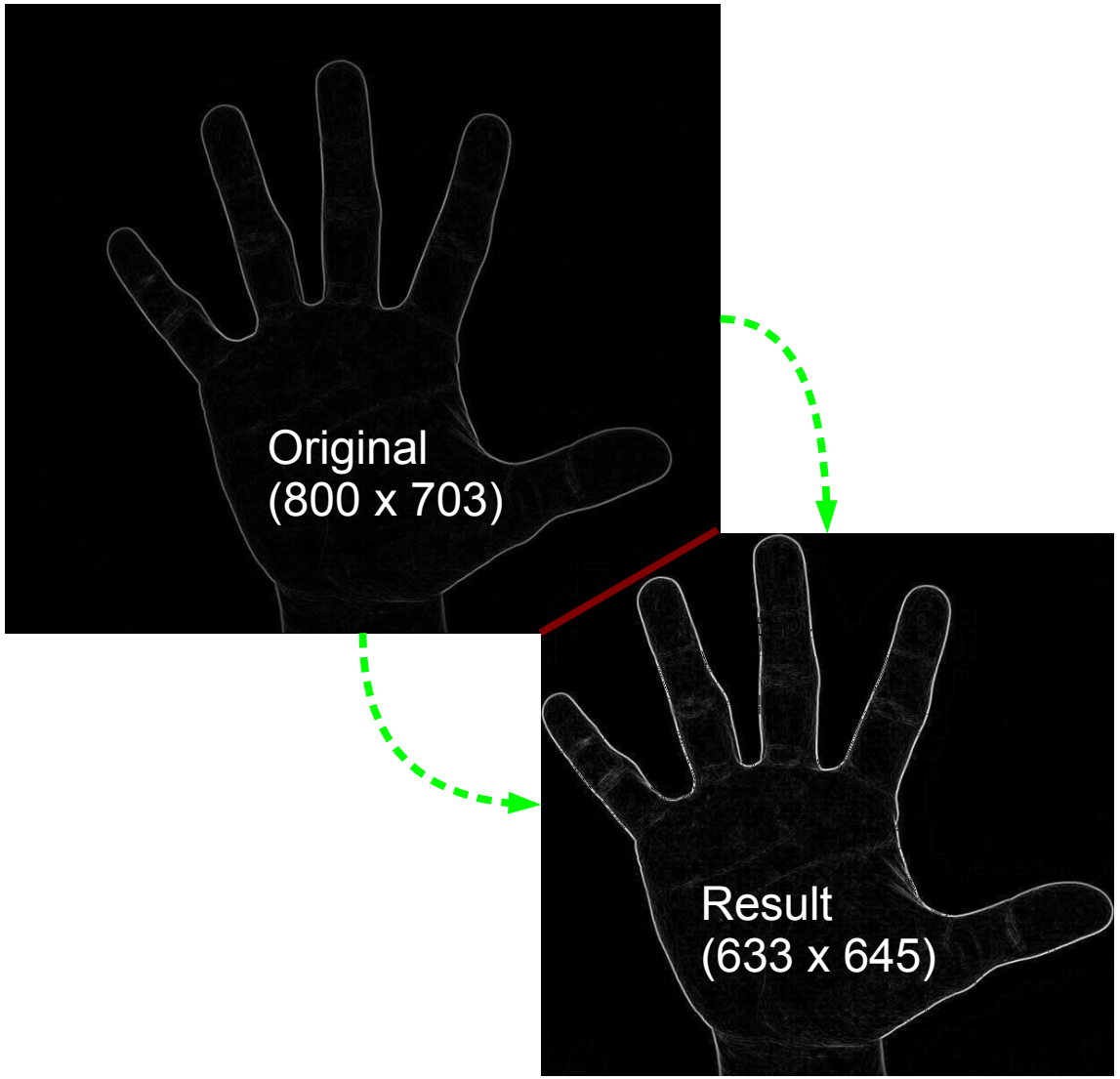
Edge Detection Results

- **Results:**
  - *Shows the important edges but little else*
  - *Robert's Cross method the optimal balance*



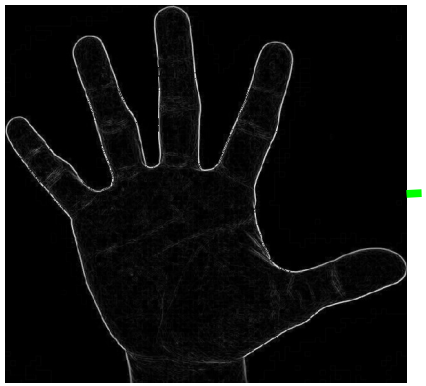
Cropping Results

- **Remove useless rows with no features**
- **Very large optimization**
  - *Area difference*
  - *Input/output*



Line detection

- **Mostly complete**
  - *Still some bugs to iron out*
  - *Detects nonexistent lines occasionally (see image!)*



The Mysterious Future

- **Line interpretation**
  - *Build AI to interpret lines*
  - *Possible methods: chaining, small-to-large object recognition (building)*
- **Finish camera-computer interaction**
  - *Device control must be precise*