KnightVisor

Matthew Senn
Anish Patel

Project

Slide 2

To create an app that allows you to use Android to see objects better in the dark in real time.

Inspiration

Hurdles

- use of a new platform unfamiliar to us
- make program real-time
- although real-time, a usable program

Real-Time Processing Slide 4

- decided to switch from Java to C using the Android's Native Development Kit (NDK)
  - again, more learning of the platform and API
- algorithm optimization

Algorithm Optimization Slide 5

Matlab functionality and programming techniques are very inefficient.

We:

- made algorithms become much more expansive and precise rather than a few lines of inefficient code
- hard coded inner, repeated processes
  - e.g., filtering using a convolution mask with zeros created unneeded computation eliminated through hard coding

The Heart of the Problem Slide 6

Traditional image convolution:

$$\sum_{r=1}^{3}\sum_{c=1}^{3} f(i+c, j+r) * k(c, r)$$

But this generic algorithm can be improved when we make assumptions about the kernel

The Heart of the Problem

Sobel Kernel is defined to be

-1	2	-1
0	0	0
1	2	2

(and its transpose)

How can we improve performance with this definition?

The Heart of the Problem Slide 8

-1	2	-1
0	0	0
1	2	2

f(r-1,c-1)*-1	f(r-1,c+0)*2	f(r-1,c+1)*1
f(r+0,c-1)*0	f(r+0,c+0)*0	f(r+0,c+1)*0
f(r+1,c-1)*1	f(r+1,c+0)*2	f(r+1,c+1)*2

One Sobel kernel:

```
-(f(r-1,c-1)+2f(r-1,c)+f(r-1,c+1))
+(f(r+1,c-1)+2f(r+1,c)+f(r+1,c+1))
```

Noise problem

- Sobel filtering creates a noisy edge map
- Canny is too slow
- we use thresholding on the edge map to reduce the visibility of noise

Threshold

- Note: Sobel edge detection is intensity valued rather than binary (0-255 instead of 0 and 1)
- Intensity values above the threshold are kept
- Intensity values below the threshold are transparent
- The intensity value then represents the amount of color used to outline

12	30	240
135	116	242
138	151	184

T = 128

0	0	240
135	0	242
138	151	184

Demo

Shortened URL: http://du.cx/2h6

Final Algorithm

1. Get current frame
2. Convert Android Java data to C data
3. Convert from YCbCr to gray scale
4. Convert data for C to work with
5. Apply logarithmic intensity transform
6. Apply Sobel edge detection
7. Threshold* edge map, two options
    a. manual threshold by user
    b. automatic threshold by Otsu's method
8. Color edge map
9. Convert C data to Android Java data
10. Draw edge bitmap onto the current frame

Questions?