Real-Time Animated Video

6.111 Final Project
David Kang
Tarun Malik
Ariana Eisenstein
Juno, directed by Jason Reitman (2007; Los Angeles, CA: 20th Century Fox, 2008), DVD.
Cartoonifying Image(via hand) is Inefficient

“The hand-drawn opening credits for Jason Reitman's second feature film Juno: **The printing, hand-tracing, xeroxing, cutting and coloring of over 900 images** of the main character Juno MacGuff walking through her neighborhood ‘took nearly supernatural patience’”

- designer Gareth Smith of Shadowplay Studio on the opening scenes of Juno.

Citation:

*Interview with Gareth Smith on the opening scene of Juno, http://www.watchthetitles.com/articles/0069-Juno*
Cartoonifying Image (via software) is Inefficient

- Multiple filters on one image
- Takes time
- Not instantaneous
- Does not achieve the desired effect
Cartoonifying Image (via FPGA) is Efficient!

- Can do multiple processes at once
- Can implement fast calculation
- Parallel and High computational density
System Design

High Level Block Diagram

- Camera Output
- ZBT Interface
- Edge Detection
- Color Reduction
- Image Fuser
- VGA Output
ZBT Interface

- Store the YCrCb values from the NTSC camera
- Is the source of pixel frame information
- Synchronizes image processing with Input

Citation:
http://www.dailymail.co.uk/femail/article-2132896/Florence-Colgate-Girl-Britains-beautiful-face.html
High Level Block Diagram with Camera input stored in ZBT
Edge Detection

- Performs **Feature Extraction**
- Points at which image brightness changes sharply are marked as edges.
- Edge Detection Methods:
  - **Sobel**(Gradient)
  - Canny Edge Detection
  - Gaussian
Edge Detection

Input: Y value from ZBT

Output: Selector bit and Edge Detected Image
Sobel Filter

- Mask Kernels on the image.
- Compute partial derivatives in X & Y direction.
- Compute gradient magnitudes Gx & Gy.

[Kernel Operators]

-1  0  1  
-2  0  2  
-1  0  1  

1  2  1  
0  0  0  
-1 -2 -1
High Level Block Diagram with Edge Detected image
Color Reduction

- Input YCr and YCb
- Convert to HSV
- Threshold the HSV values
- Convert to RGB
- Output 24 bit RGB
High Level Block Diagram after Color Reduction Implementation
Image Fuser

- Mux 24 bit RGB from Edge Detection and 24 bit RGB from Color Reduction
- Use selector bit from Edge Detection
High Level Block Diagram with Cartoonified Image
Software Prototyping

All of the images above, save the original image, were generated by a MATLAB implementation of the project.
# Timeline

| Week 1  | (11/5-11/11) | ● Generate Block Diagram  
● Synthesize Algorithms in Matlab(Completed) |
|---------|--------------|------------------------------------------------------------------|
| Week 2  | (11/12 - 11/18) | ● Implement ZBT to VGA verilog  
● (Displaying images to screen)  
● Write verilog for Edge Detection  
● Write test benches for Edge Detection  
● Write verilog for Color Reduction  
● Write test benches for Color Reduction |
| Week 3  | (11/19 - 11/25) | ● Implement Edge Detection verilog  
● Implement Color Reduction verilog  
(Displaying images to screen) |
| Week 4  | (11/26 - 12/2) | ● Integration of entire system |
| Week 5  | (12/3 - 12/9) | ● Debugging & Checkoff |
Questions?