Week 0 Review

Zachary Daniels

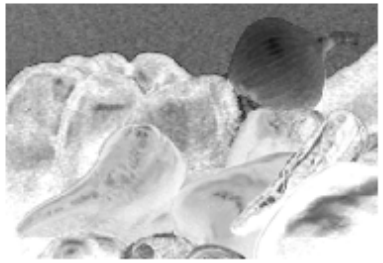
Separation of Images into Channels (HSV Example)



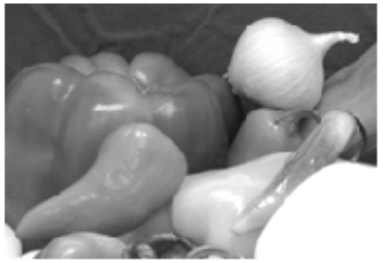
original



hue



saturation

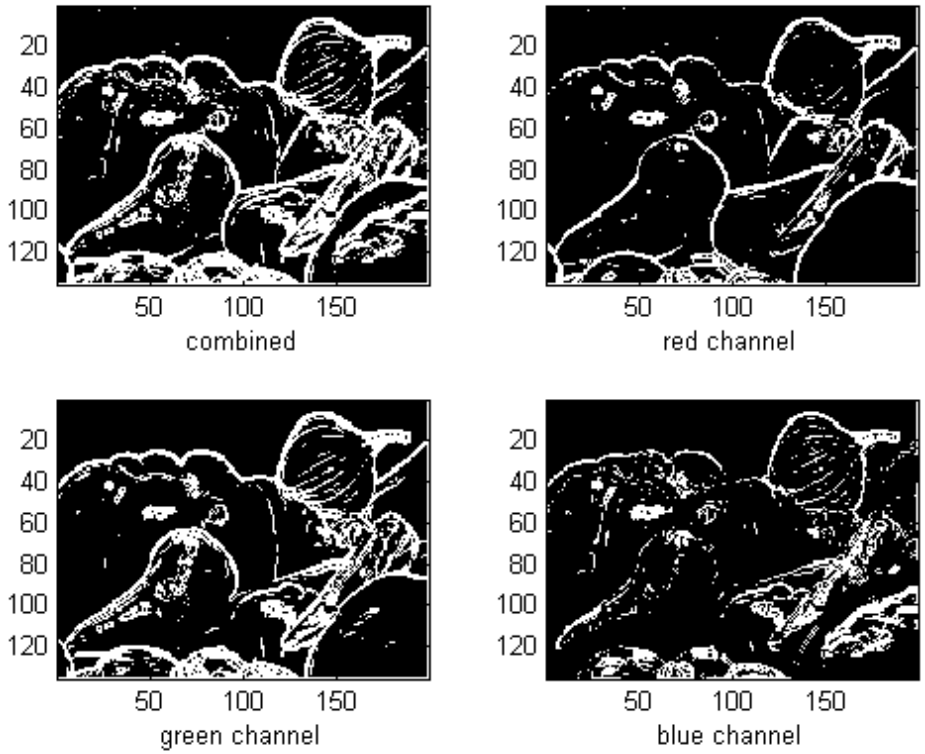


value

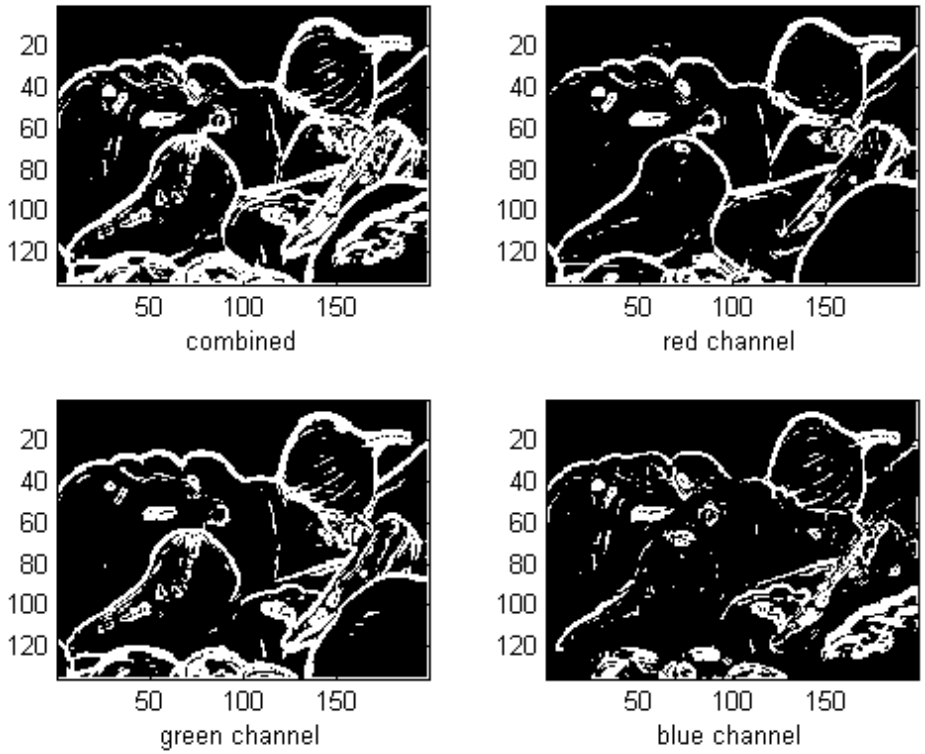
Basic Edge Detection

- Roberts' Edge Detector
  - Kernels: $\mathrm{K_x} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \mathrm{K_y} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
- Sobel Edge Detector
  - Kernels: $\mathrm{K_x} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \mathrm{K_y} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
- Intensity = $\sqrt{K_x^2 + K_y^2}$

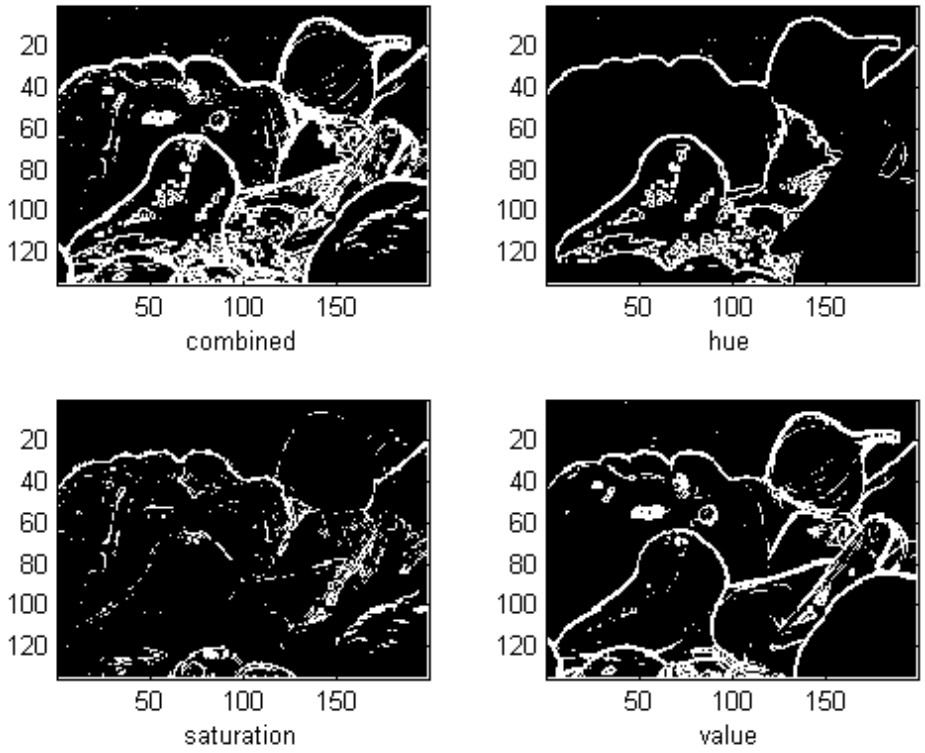
Results (Roberts' + RGB)



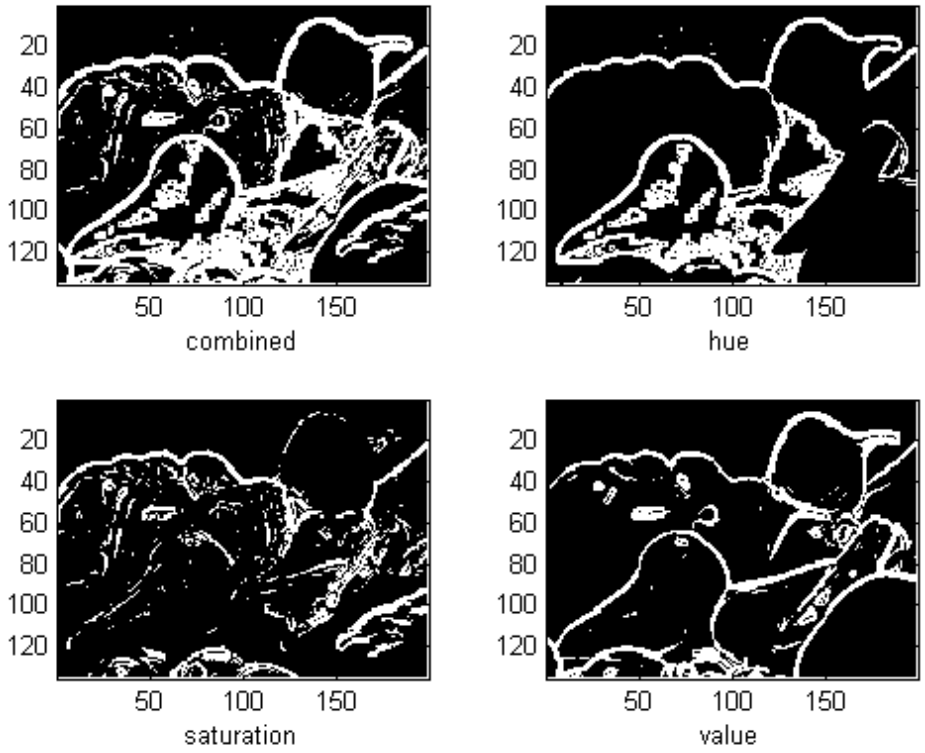
Results (Sobel + RGB)



Results (Roberts' + HSV)



Results (Sobel + HSV)



Discussion

- When an edge is white in the three-channel image, it means that it appears in one of the three component images (i.e. edges are OR-ed together).
- The results for HSV differ greatly from those of RGB. ‘
  - With RGB, edge detection performed similarly for each channel.
  - For HSV, the edges produced using each channel were significantly different, and the resulting edges were much noisier.
  - Likely because HSV channels each represent different concepts, some of which lend themselves well to edge detection like ‘value’ while others like ‘hue’ were noisy.
  - RGB channels each represent similar concepts: the amount of a single component color.

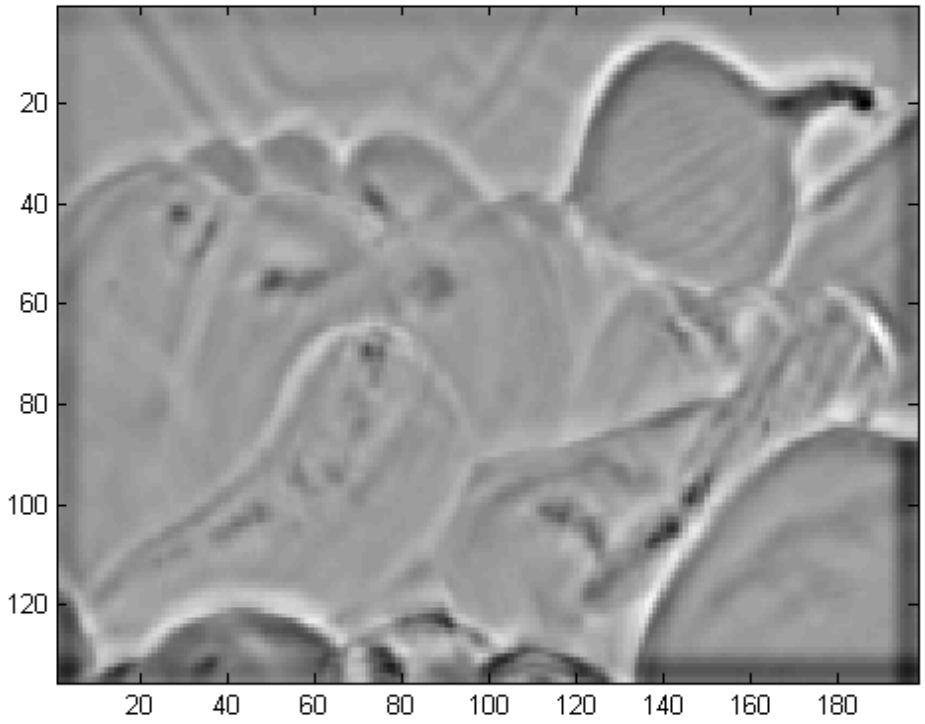
Separability

- Smoothing Gaussian:
  - If a 2-d convolution were to be performed, the asymptotic runtime would be n^2.
  - The asymptotic runtime of two 1-d convolutions is only 2n.
  - Convolution by two 1-d kernels is faster than by one 2-d kernel.
- For the Sobel kernel:
  1. To get the gradient in the x-direction:
     1. Convolve [1 2 1]' across an image.
     2. Convolve [1 0 -1] across the newly produced image.
  2. To get the gradient in the y-direction:
     1. Convolve [1 0 -1]' across the image
     2. Convolve [1 2 1] across the newly produced image.
  - Asymptotic runtime: 4n

Laplacian of Gaussian (Using Separability)

- 4 Convolutions:
  - $G * (I * G_{xx}) + G * (I * G_{yy})$
- To convert to an edge detector:
  - Find the zero-crossings of the LoG of an image

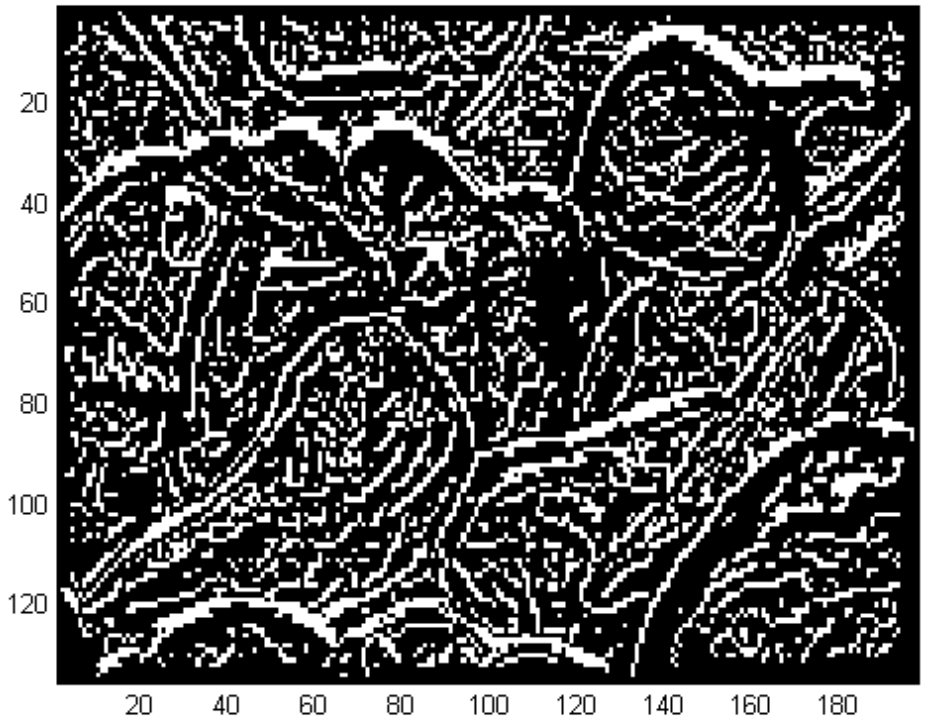
Results (LoG)



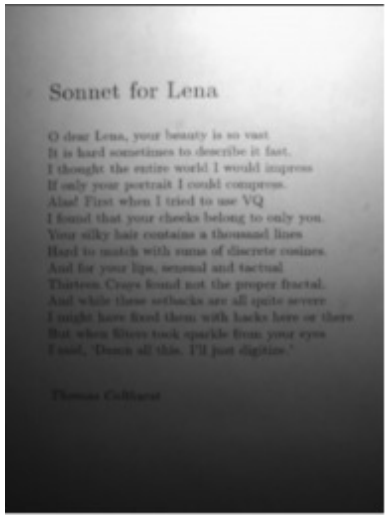
Canny Edge Detector

1. Smooth using a Gaussian Kernel
2. Detect edges using a first derivative kernel
3. Perform non-maximum suppression
4. Implement hysteresis thresholding

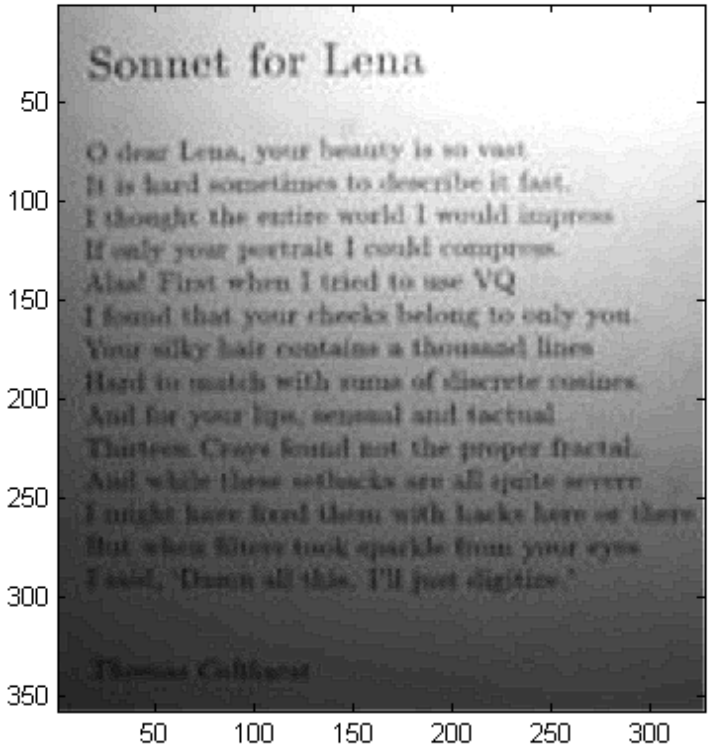
Results (Canny Edge Detector w/out hysteresis thresholding)



Cleaning Images (Before)



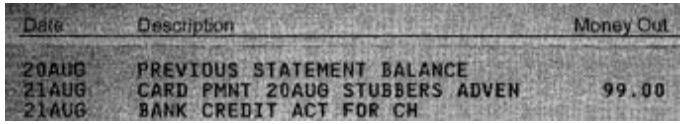
Cleaning Images (After)



How?

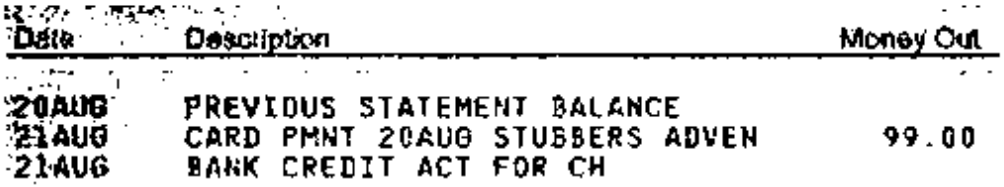
- What worked:
  - Resizing
  - Cropping
  - Sharpening
- What failed:
  - Histogram equalization
  - Thresholding
  - Edge detection
  - Gaussian Smoothing
  - Denoising
    - Both median and mean filters
  - Rotation and thresholding line-by-line
  - Segmentation and thresholding
  - Adjusting brightness using a radial-shaped kernel

Cleaning Images Cont. (Before)



Date	Description	Money Out
20AUG	PREVIOUS STATEMENT BALANCE	
21AUG	CARD PMNT 20AUG STUBBERS ADVEN	99.00
21AUG	BANK CREDIT ACT FOR CH	

Cleaning Images Cont. (After)

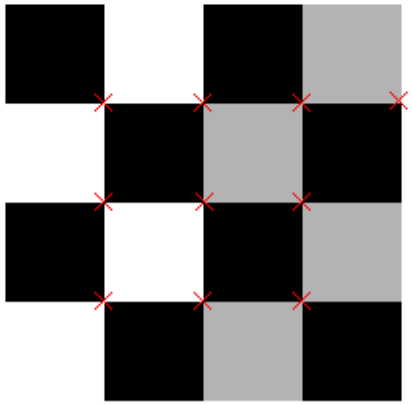


Date	Description	Money Out
20AUG	PREVIOUS STATEMENT BALANCE	
21AUG	CARD PMNT 20AUG STUBBERS ADVEN	99.00
21AUG	BANK CREDIT ACT FOR CH	

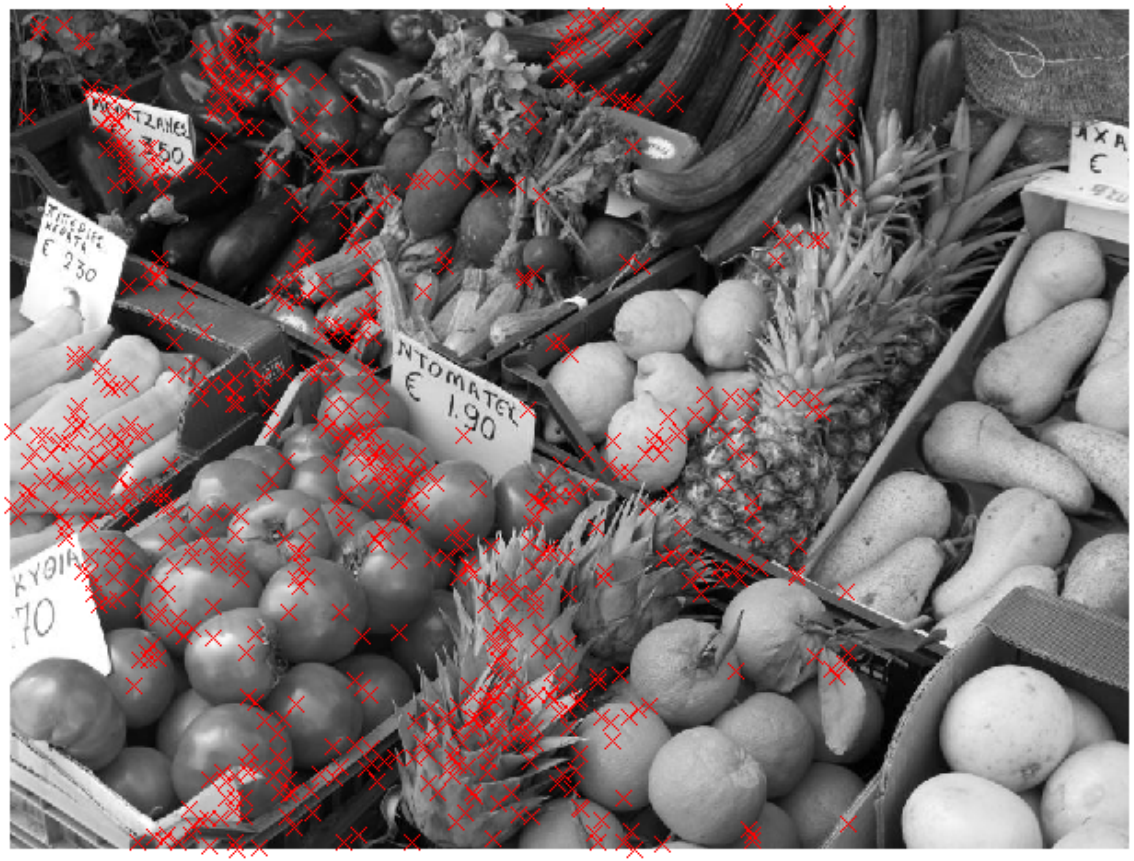
How?

- What worked:
  - Segmentation and thresholding
  - Median filter for denoising
  - Resizing
  - Sharpening

Harris Corner Detector



Harris Corner Detector



Optical Flow (Lucas-Kanade, No Pyramids)

$$-f_t = f_x u + f_y v$$



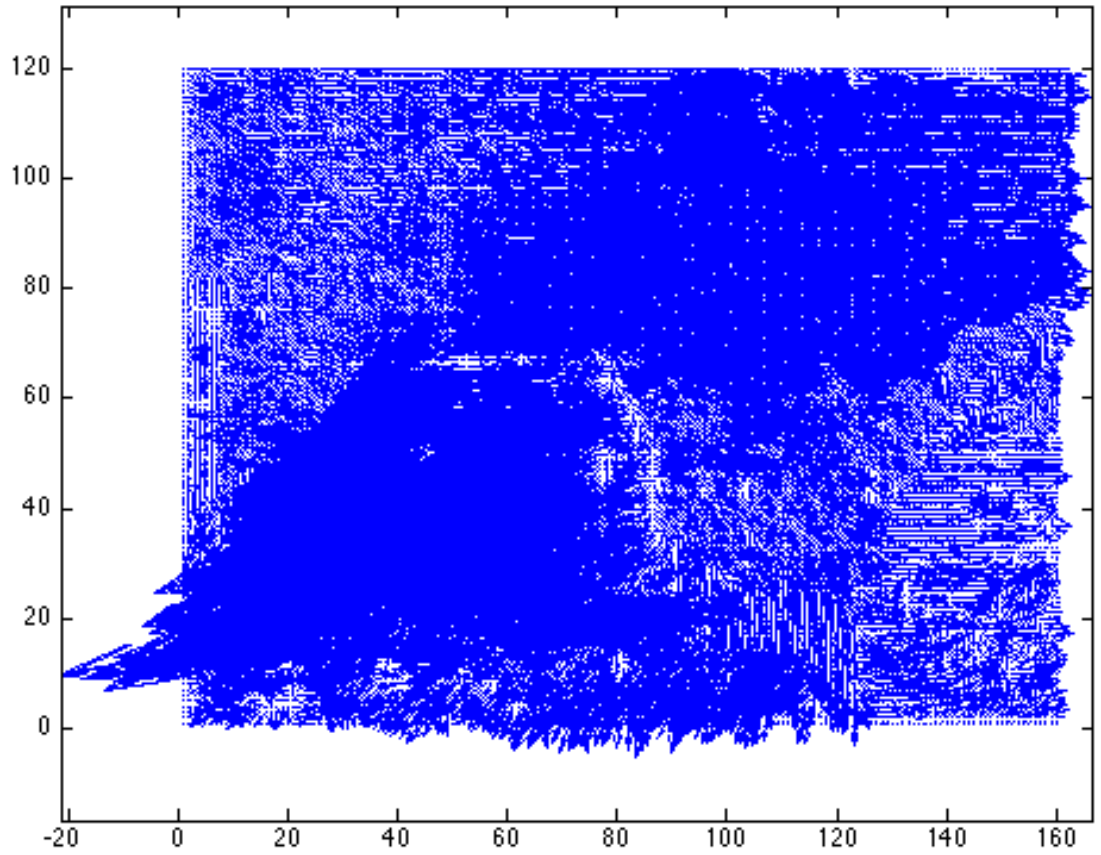
Optical Flow (Lucas-Kanade, No Pyramids)



Optical Flow (Lucas-Kanade w/ Pyramids)



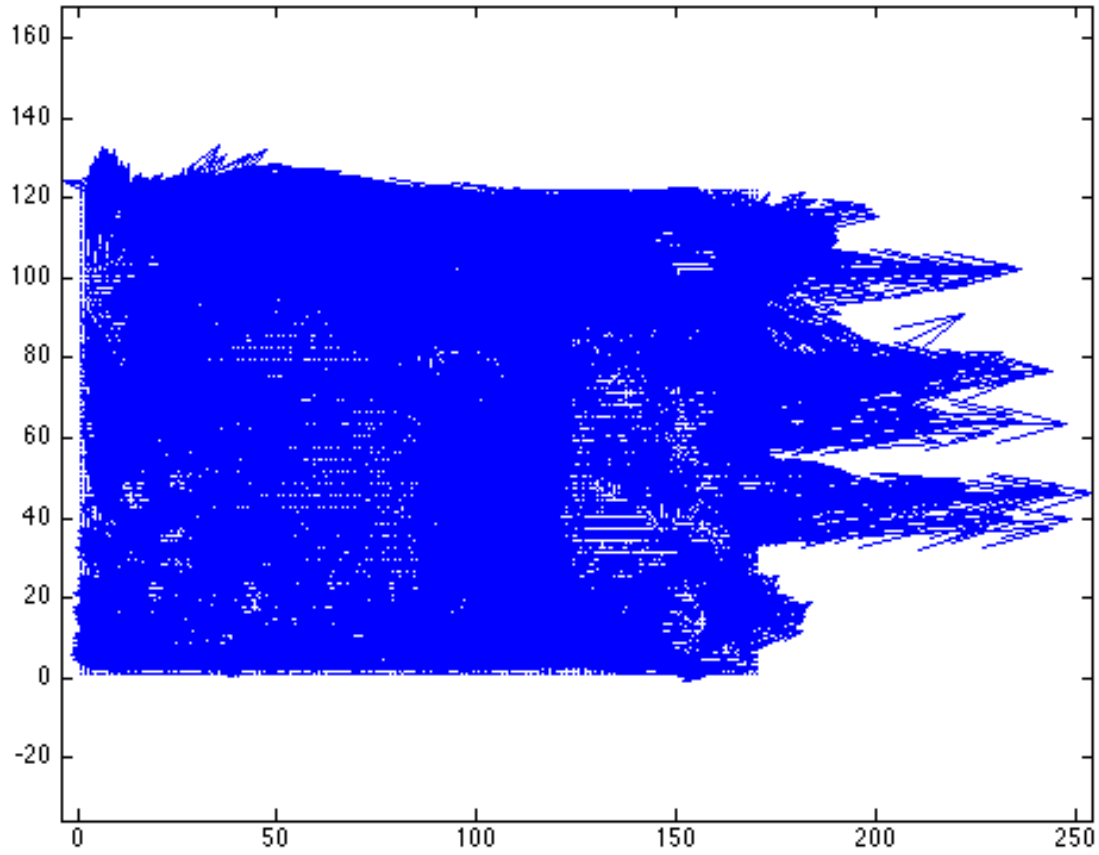
Optical Flow (Lucas-Kanade w/ Pyramids)



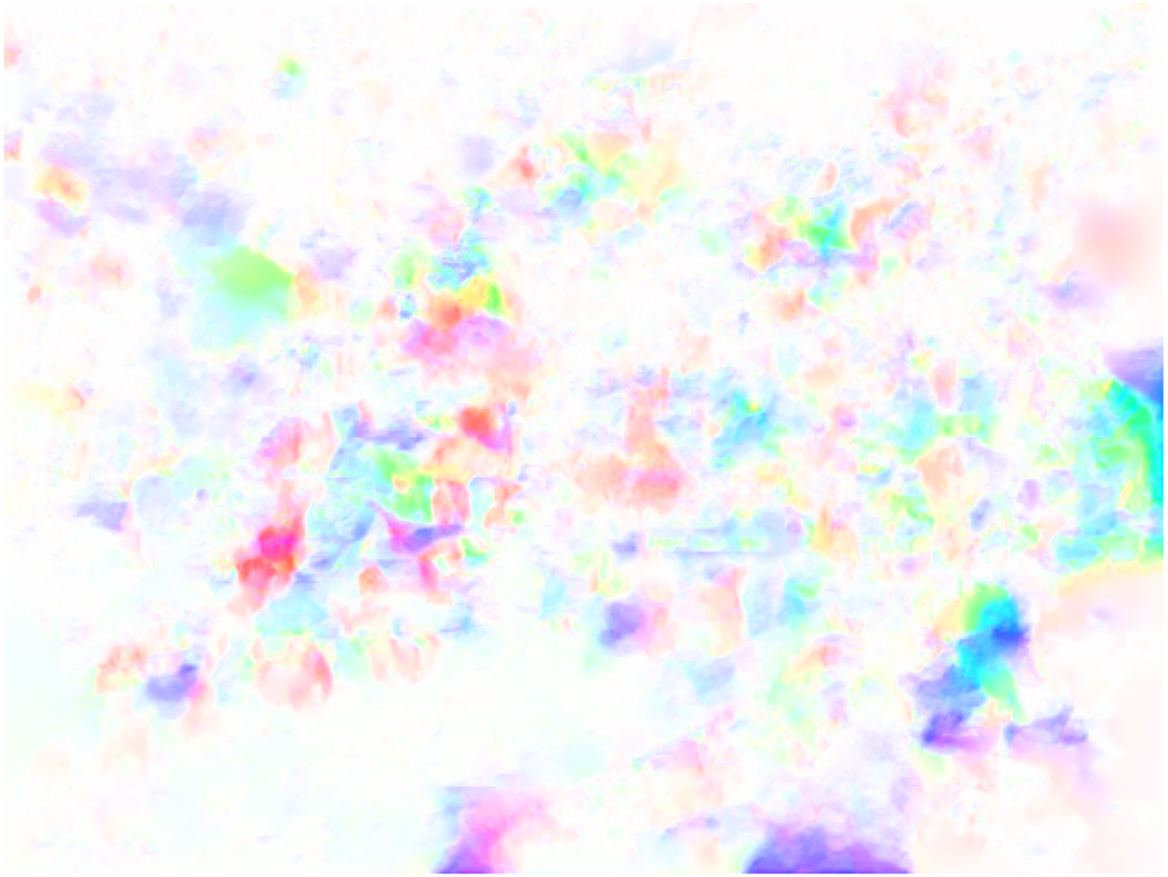
Optical Flow (Lucas-Kanade w/ Pyramids)



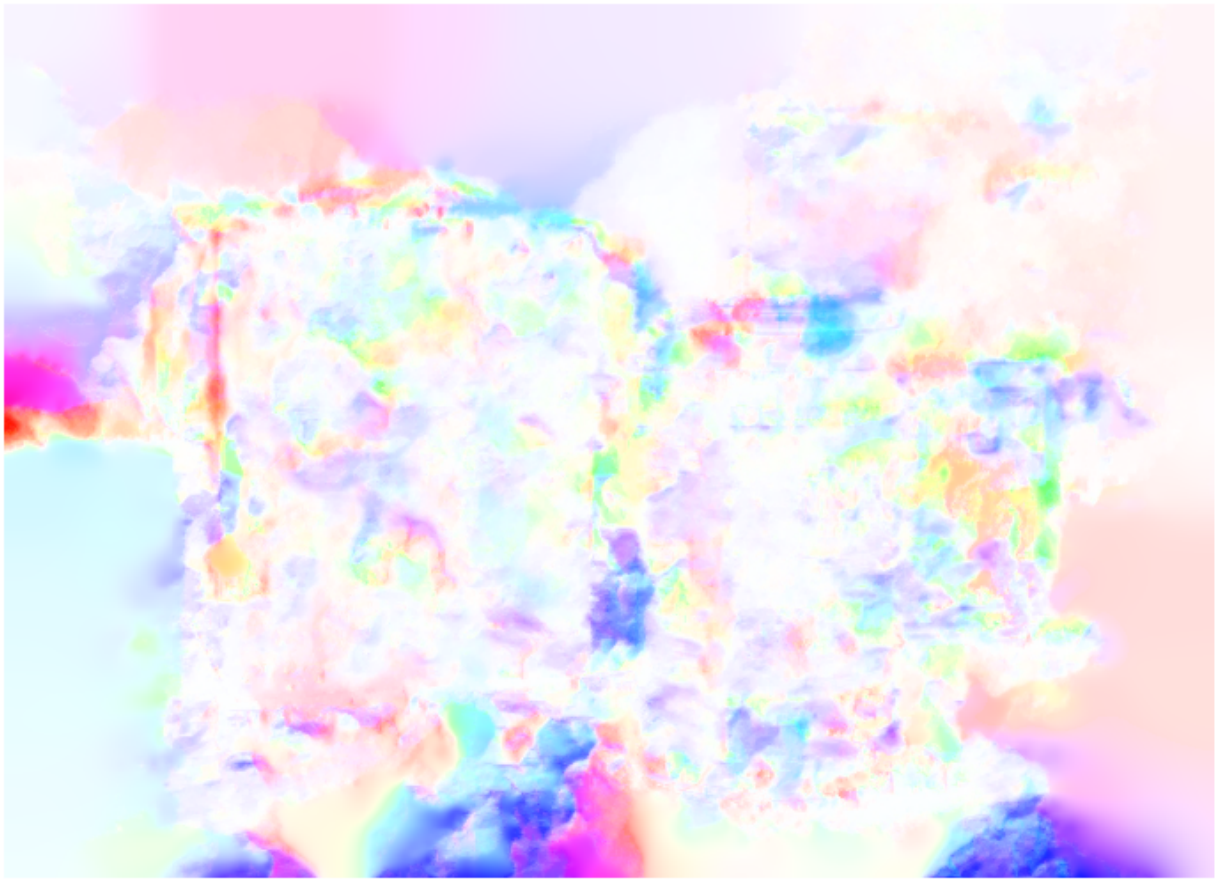
Optical Flow (Lucas-Kanade w/ Pyramids)



Optical Flow (Liu)



Optical Flow (Liu)



SVM w/ SIFT Problem

- Given images representing 15 scenes, extract SIFT features from each image (sampled at even intervals), and attempt to classify new images using SVMs.
- 4485 images,
  - 80% for training (3583 images)
  - 20% for testing (902 images)

SVM w/ SIFT Results

- Using a linear kernel: ~33.8%
- Using a histogram intersection kernel: ~39.0%
  - $K(X, Y) = \sum_i \min(x_i, y_i)$

SVM w/ Bag of Words

- First Run
  - Normal sift
  - 50 descriptors per image randomly sampled
  - k-means clustering to form bag of words
    - k = 30
- Second Run
  - Dense sift
  - 70 descriptors per image randomly sampled
  - k-means clustering to form bag of words
    - k=100
- Results
  - 5-Fold Cross Validation
  - Only 1-2% accuracy for both runs
- Problems
  - With large numbers of descriptors and/or a large value of k, too much memory and time is used
  - Hand-coded dense sift takes too long to run

Other Topics Covered

- Scale-Space
- AdaBoost
- Viola-Jones Face Detection
- Gaussian and Laplacian Expansion/Reduction

Project Selections

- Subspace Clustering Using Graph-Regularized Sparse Coding
  - Mentor: Nasim Souly
  - Overview: Sparse coding is a technique for dimensionality reduction which finds a sparse representation of feature vectors (i.e. combinations of basis vectors that when combined form equivalent or approximations of the initial feature vectors) by first discovering a dictionary of basis vectors. Spectral graph theory can provide structural information about an image. Combining the two might be useful in creating a technique for motion segmentation and video clustering.
  - Background:
    - Interest in unsupervised feature learning
    - Read/skimmed suggested papers + few others

Project Selections Continued

- Multimodal Deep Learning
  - Haroon Idress
  - Overview: Using a special type of neural network called a stacked denoising autoencoder, determine tags from images and images from tags by learning a shared representation.
  - Background:
    - Interest in unsupervised feature learning
    - Past experience building (non-stacked, non-denoising) autoencoders

Project Selections Continued

- Subspace Clustering
  - Gonzalo Vaca-Castano
  - Overview: Survey methods for clustering in high-dimensional spaces
  - Background:
    - Limited