ECE 533 Final Project
Particle Flow

Jon Van Veen, Cristian Sanchez, Seunghyun Jeon, Brandon Krzyzanowski

Project Goals

- Analyze particle flow across images using the movement of sheep through a small opening with Particle Image Velocimetry techniques
- Determine appropriate density, resolution, and average velocity for the sheep to compute the average flow of sheep across all four given images

Introduction

- Four steps: preprocessing, heat map, determining resolution, computing average velocities
- Discussion, observations, conclusions for each step and the project

Step 1: Preprocessing of Images

- Our preprocessing consists of standard steps, including normalization and converting image values to doubles
- We threshold all four images in order to pick out distinct sheep for future particle flow analysis

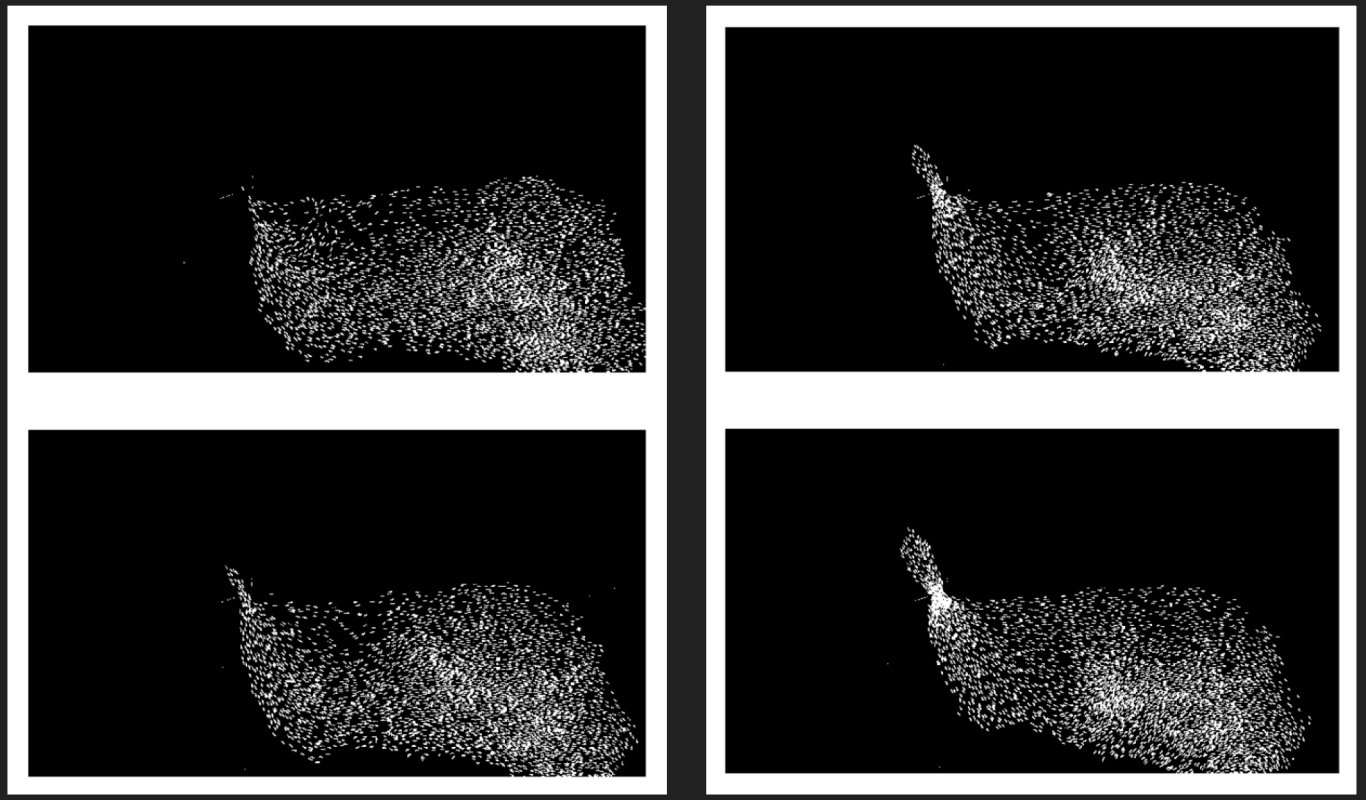
```
sheep1 = imread("seq1.png");
sheep2 = imread("seq2.png");
sheep3 = imread("seq3.png");
sheep4 = imread("seq4.png");
imshow(sheep1)
imshow(sheep2)
imshow(sheep3)
imshow(sheep4)

sheep1Gray = double(rgb2gray(sheep1));
sheep2Gray = double(rgb2gray(sheep2));
sheep3Gray = double(rgb2gray(sheep3));
sheep4Gray = double(rgb2gray(sheep4));

sheep1Norm = sheep1Gray/max(sheep1Gray(:));
sheep2Norm = sheep2Gray/max(sheep2Gray(:));
sheep3Norm = sheep3Gray/max(sheep3Gray(:));
sheep4Norm = sheep4Gray/max(sheep4Gray(:));

sheep1Threshold = mythreshold(sheep1Norm, 0.6);
imshow(sheep1Threshold)
sheep2Threshold = mythreshold(sheep2Norm, 0.6);
imshow(sheep2Threshold)
sheep3Threshold = mythreshold(sheep3Norm, 0.6);
imshow(sheep3Threshold)
sheep4Threshold = mythreshold(sheep4Norm, 0.6);
imshow(sheep4Threshold)
```

Thresholding

- We threshold all four images in order to pick out distinct sheep for future particle flow analysis

```
function newImage = mythreshold(image, num)
    [m,n] = size(image);
    newImage = image;
    for i = 1:m
        for j = 1 : n
            if newImage(i,j) > num
                newImage(i,j) = 1;
            else
                newImage(i,j) = 0;
            end
            if (i < 286 || j < 483)
                newImage(i, j) = 0;
            end
        end
    end

end
```

Step 1: Preprocessing of Images

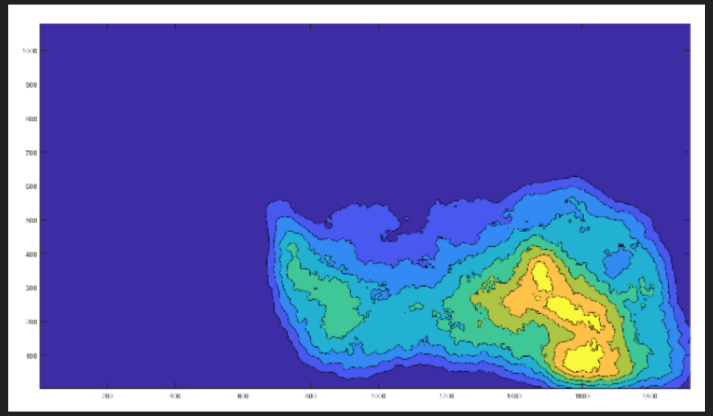


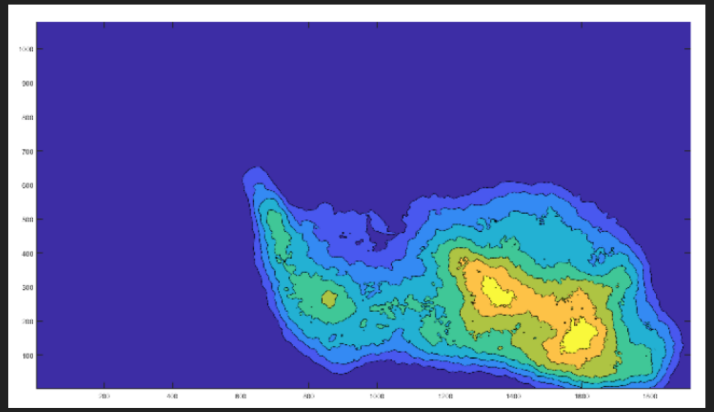
Step 2: Sheep “Heat Map”

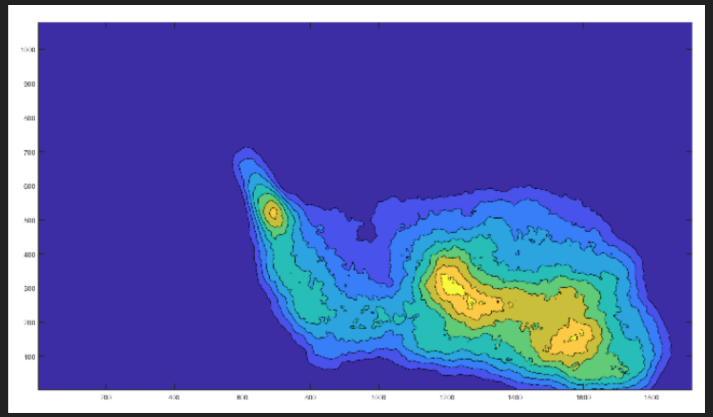
- We created a “heat map” for each frame in order to find the density of the sheep herd in each image
- Used a disk of radius 50 as a structural element
- Convoluted each thresholded image with the structural element
- Had to flip due to the way convolution works in matlab
- Created a contour map to show the density of the flock

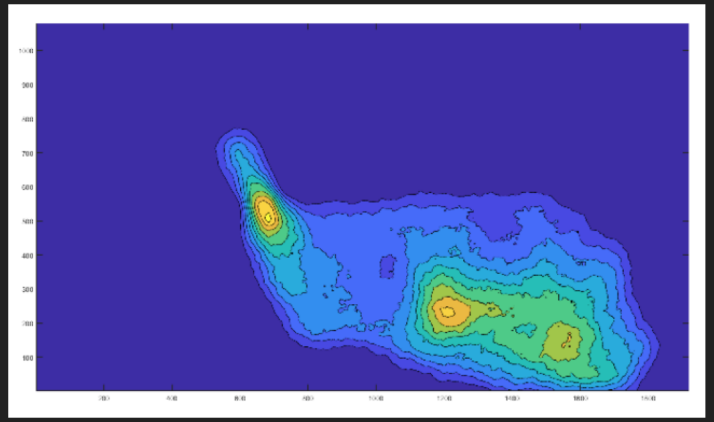
```
SE = strel('disk',50);
SE = SE.Neighborhood;

disp('Heatmap for Image 1')
heat1 = contourf(conv2(flip(sheep1Threshold,1),SE,'same'));
imshow(heat1)
```

Step 2: Sheep “Heat Map” -- Results



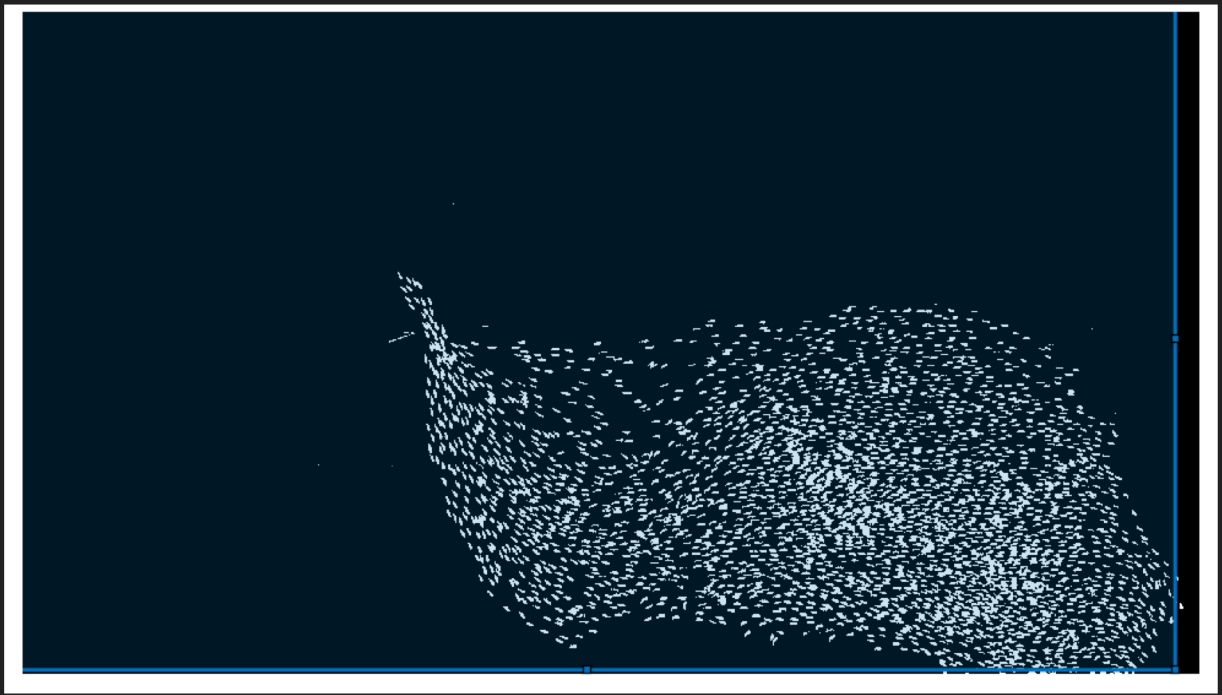






Step 3: Determining Resolution

Step 3: Determining Resolution

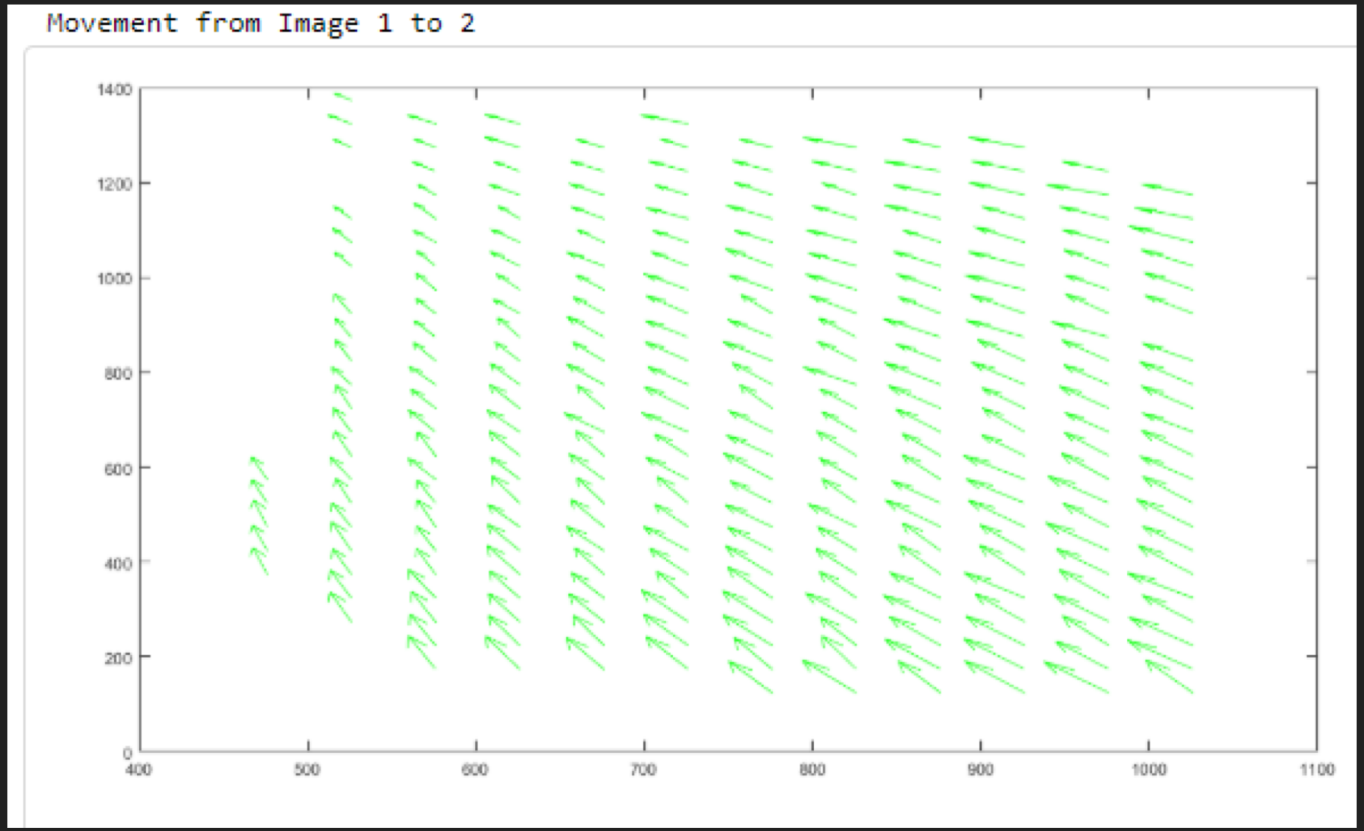


- First attempt at finding movement between frames
- Searches over entire image
- Shows overall movement up and to the left

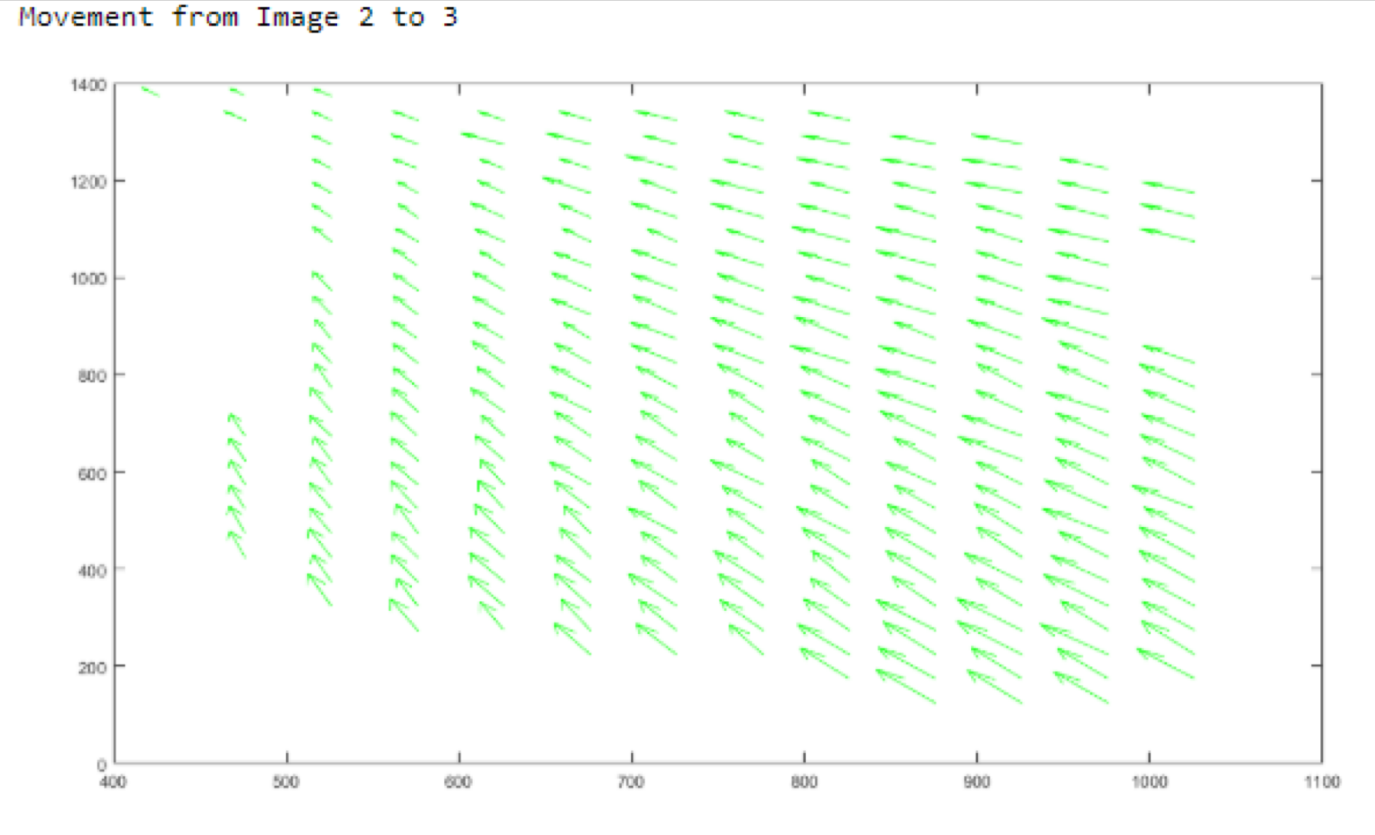
Step 3: Determining Resolution

```
function [x0, y0 , x1, y1] = findMovement(img1, img2, step)
    x0 = []; y0 = []; x1 = []; y1 = [];

    temp2 = zeros(max(size(img2))+200);
    [m,n] = size(img1);
    temp2(201:200+m, 201:200+n) = img2;
    currentarea = 1;

    for i = 1: step : m - step +1
        for j = 1 : step : n -step +1
            temp1 = img1(i:i+step , j:j+step);
            frame = temp2(i:i+5*step, j:j+5*step);

            if sum(temp1(:)) > 15
                x0(currentarea) = i + floor(step/2);
                y0(currentarea) = j + floor(step/2);
                correlate = normxcorr2(temp1, frame);

                [ymax , xmax] =  find(correlate==max(correlate(:)));

                yoffset = xmax(1) - size(temp1,1);
                xoffset = ymax(1) - size(temp1,2);

                x1(currentarea) = xoffset- x0(currentarea);
                y1(currentarea) = yoffset - y0(currentarea);
                currentarea = currentarea +1;
            end
        end
    end
end
```

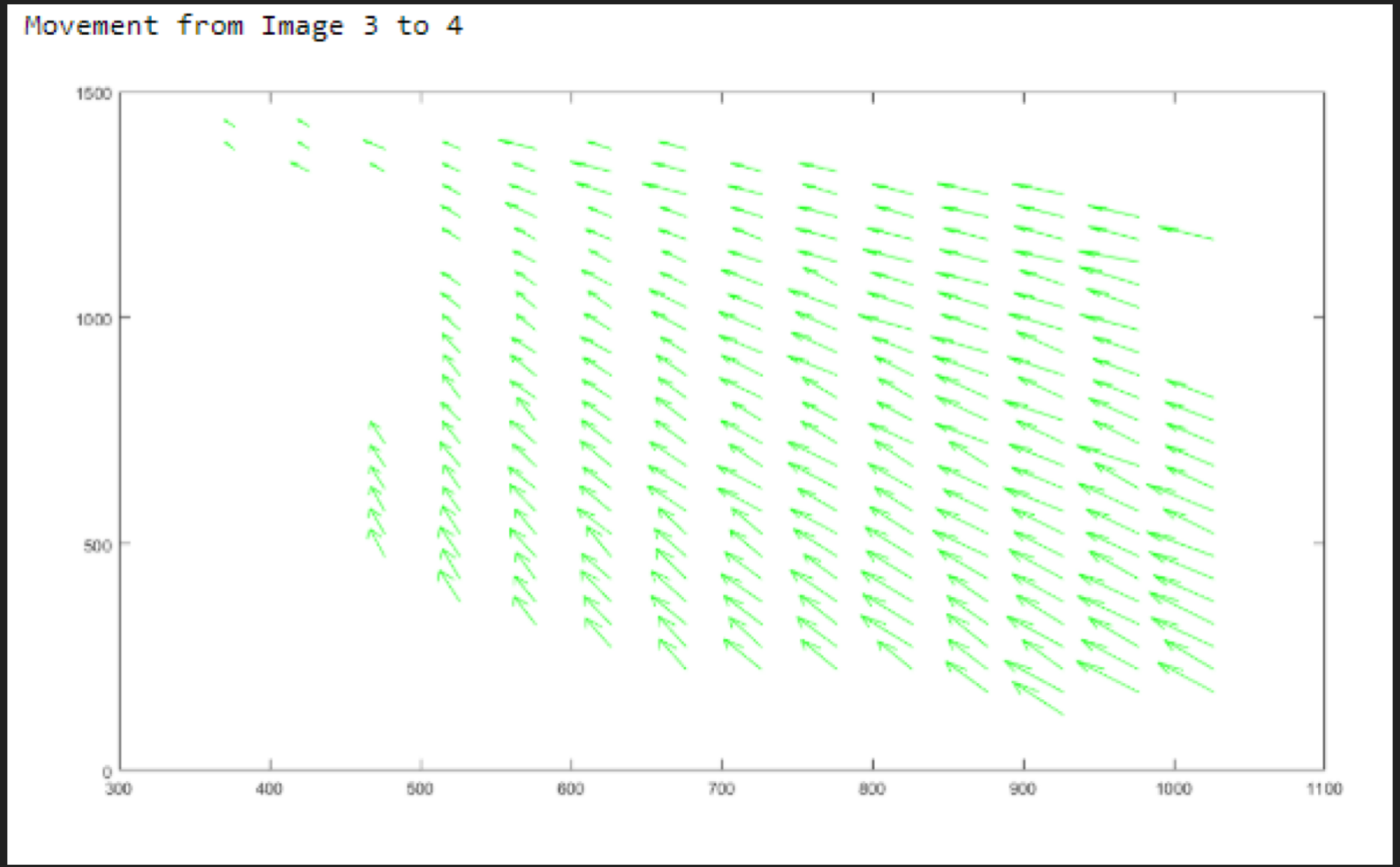
Step 3: Determining Resolution -- Results



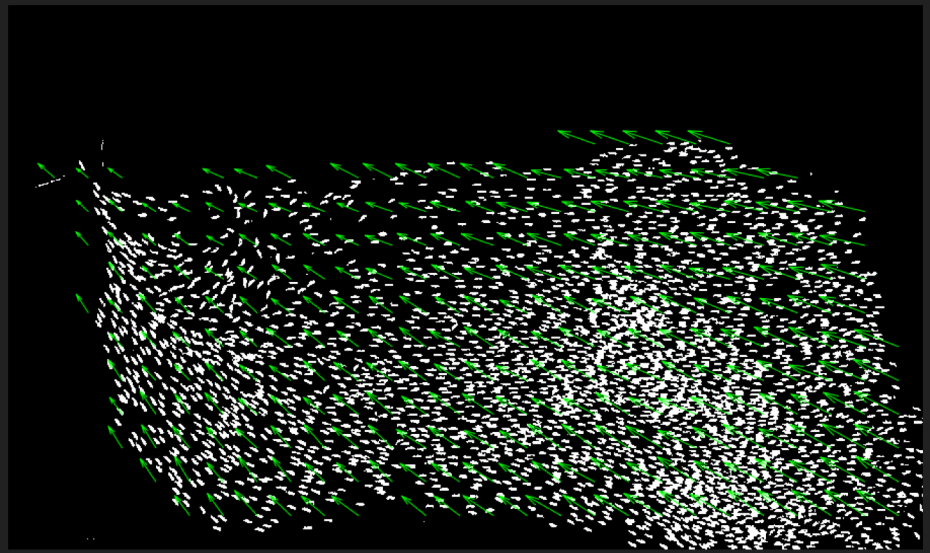
Step 3: Determining Resolution -- Results

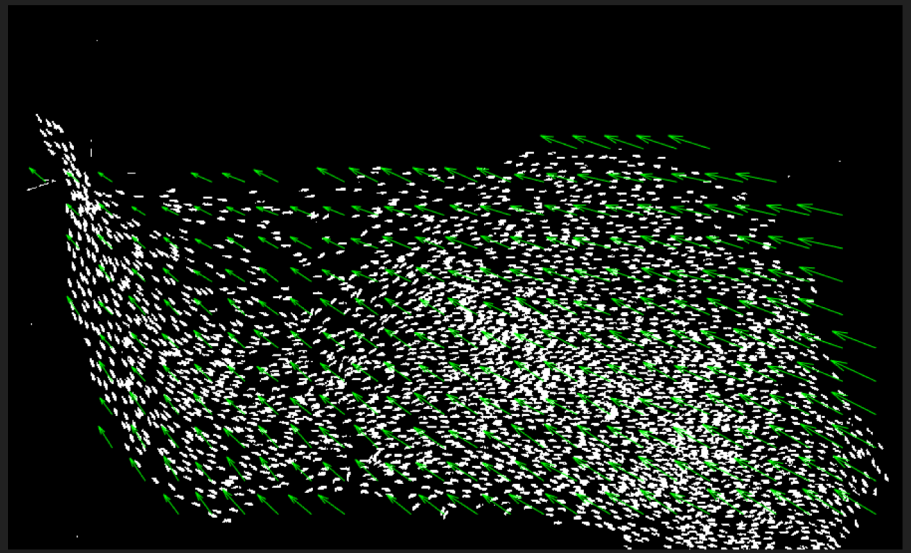


Step 3: Determining Resolution -- Results

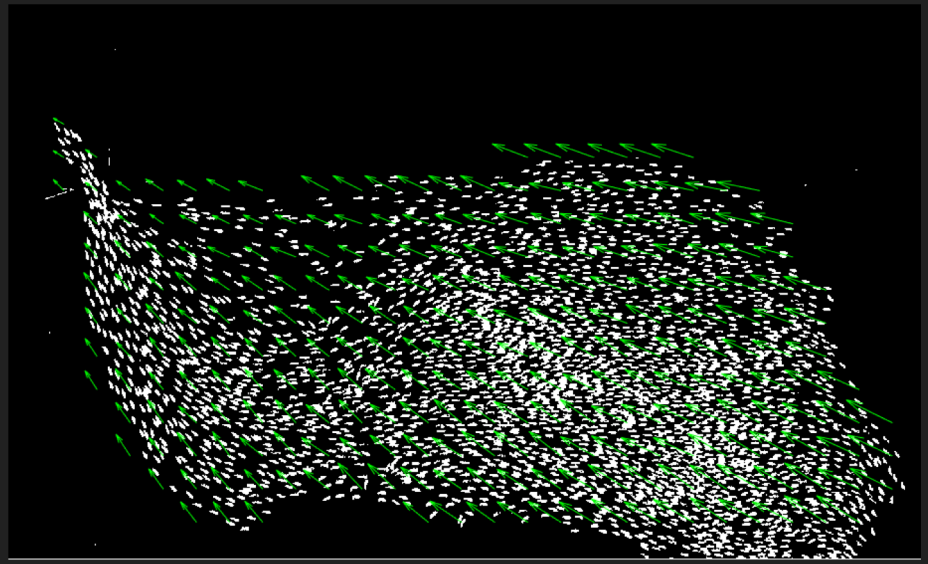


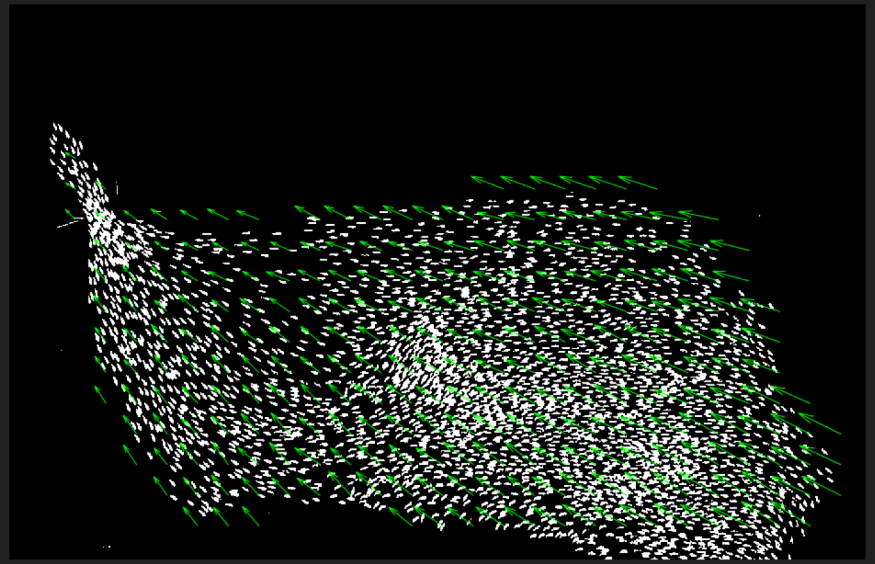
Step 3: Determining Resolution -- Results



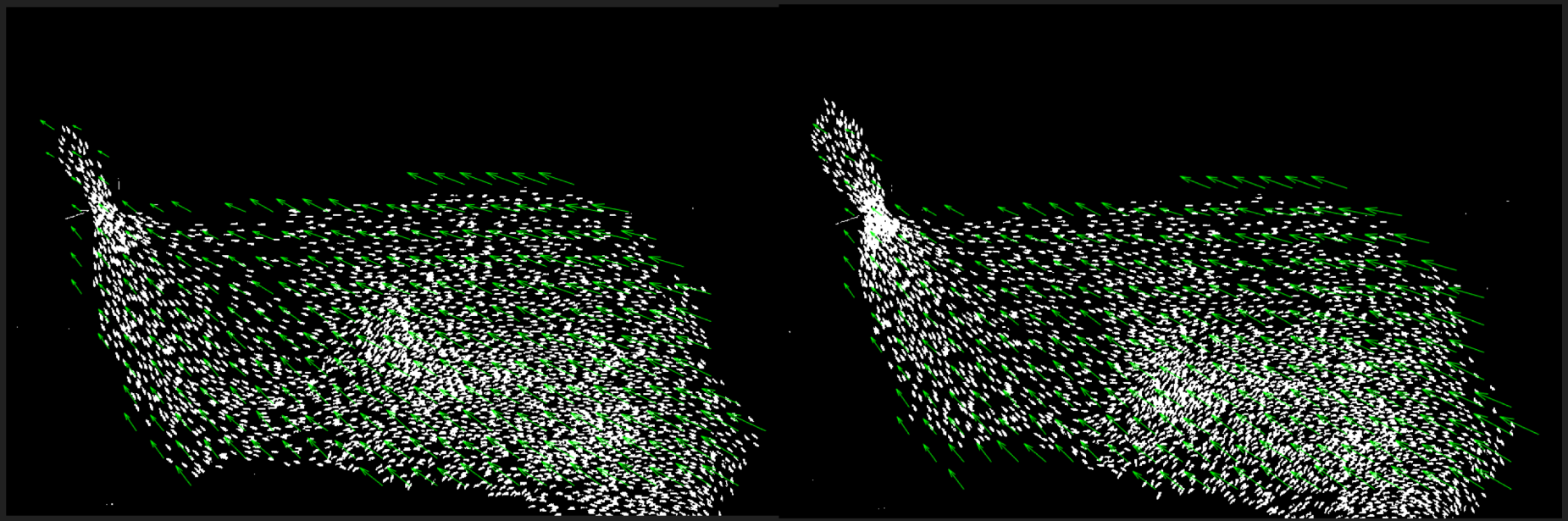


Step 3: Determining Resolution -- Results





Step 3: Determining Resolution -- Results



Step 3: Determining Resolution -- Results

Average Velocity from Frame 1 to Frame 2

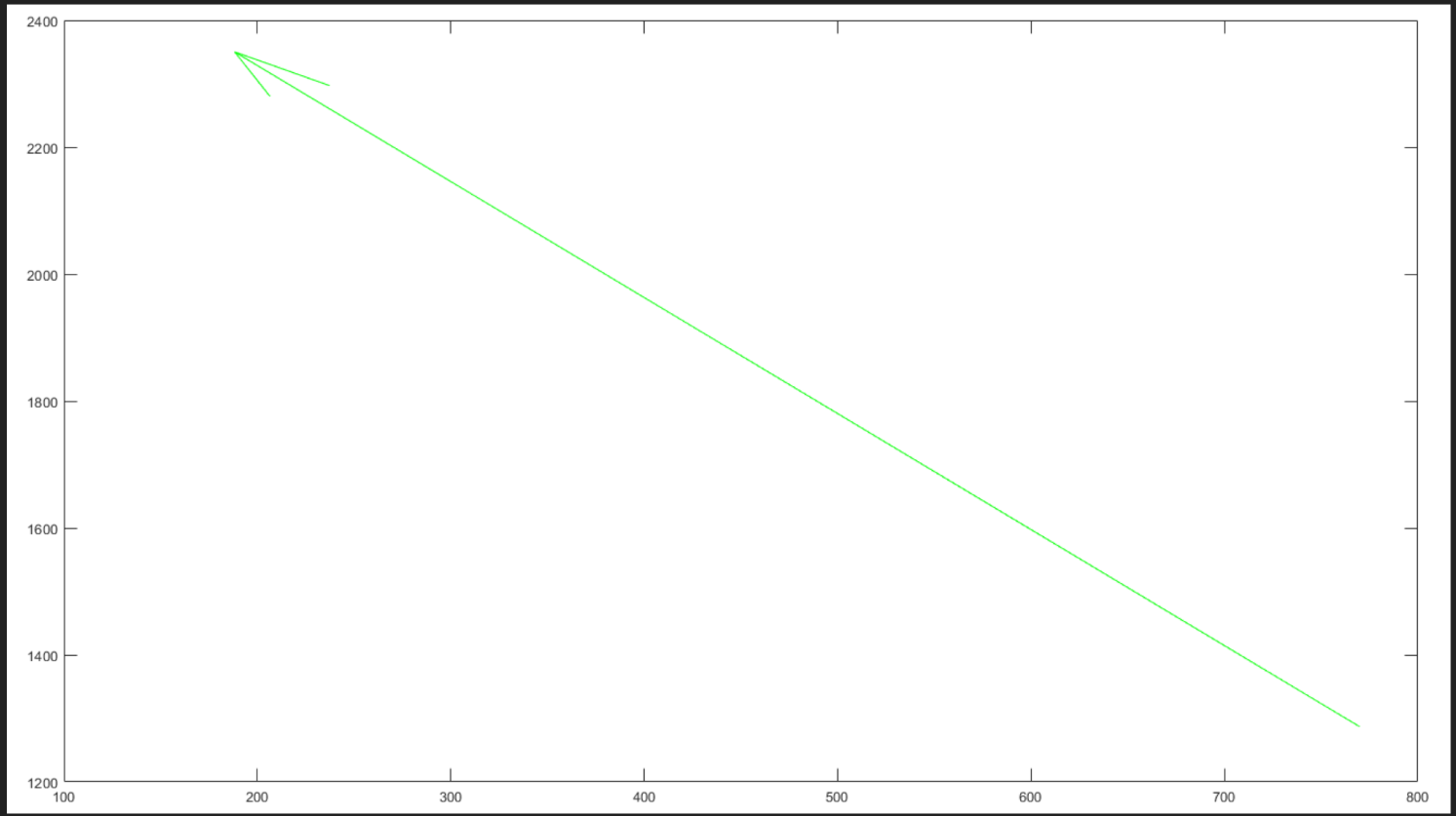
```
disp('Average Velocity from Image 1 to 2 w/ Corresponding Images:')
disp1 = ['Velocity of ', num2str(velocity1) ' Pixels/Frame'];
disp(disp1)

imshow(sheep1Threshold)
hold on
quiver(mean(y0), mean(x0), mean(v0), mean(u0),'color',[0 1 0])
hold off

imshow(sheep2Threshold)
hold on
quiver(mean(y0), mean(x0), mean(v0), mean(u0),'color',[0 1 0])
hold off
```

Step 4: Computing Average Velocity

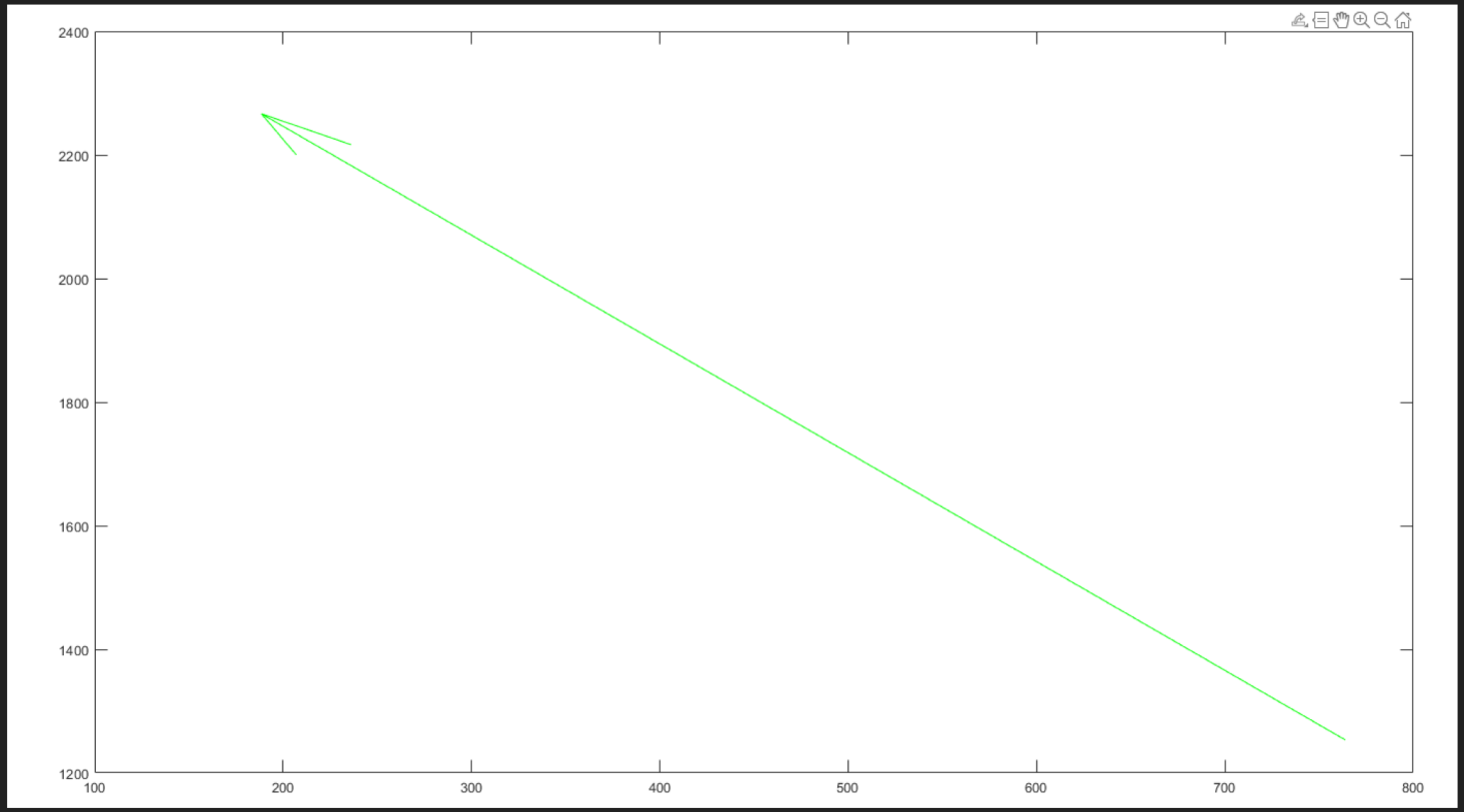
- findPIV function

```
function [velocity] = findPIV(x, y, u, v)
    quiver(mean(x), mean(2000-y), mean(u), mean(v),'color',[0 1 0])
    avgX = (u - x);
    avgY = (v - y);
    velocity = mean(sqrt(avgX.^2 + avgY.^2)/2);
end
```

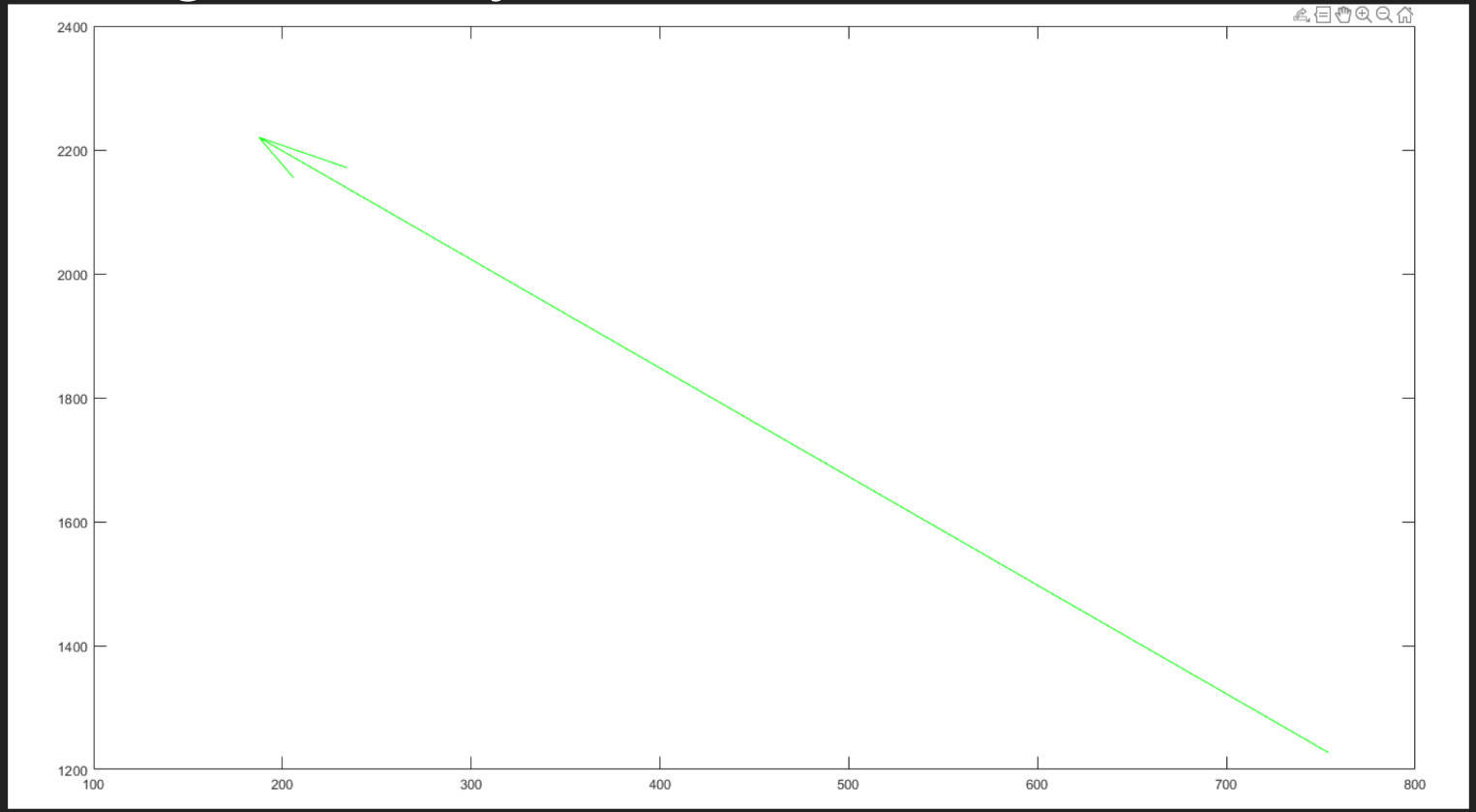
Average Velocity from Frame 1 to 2



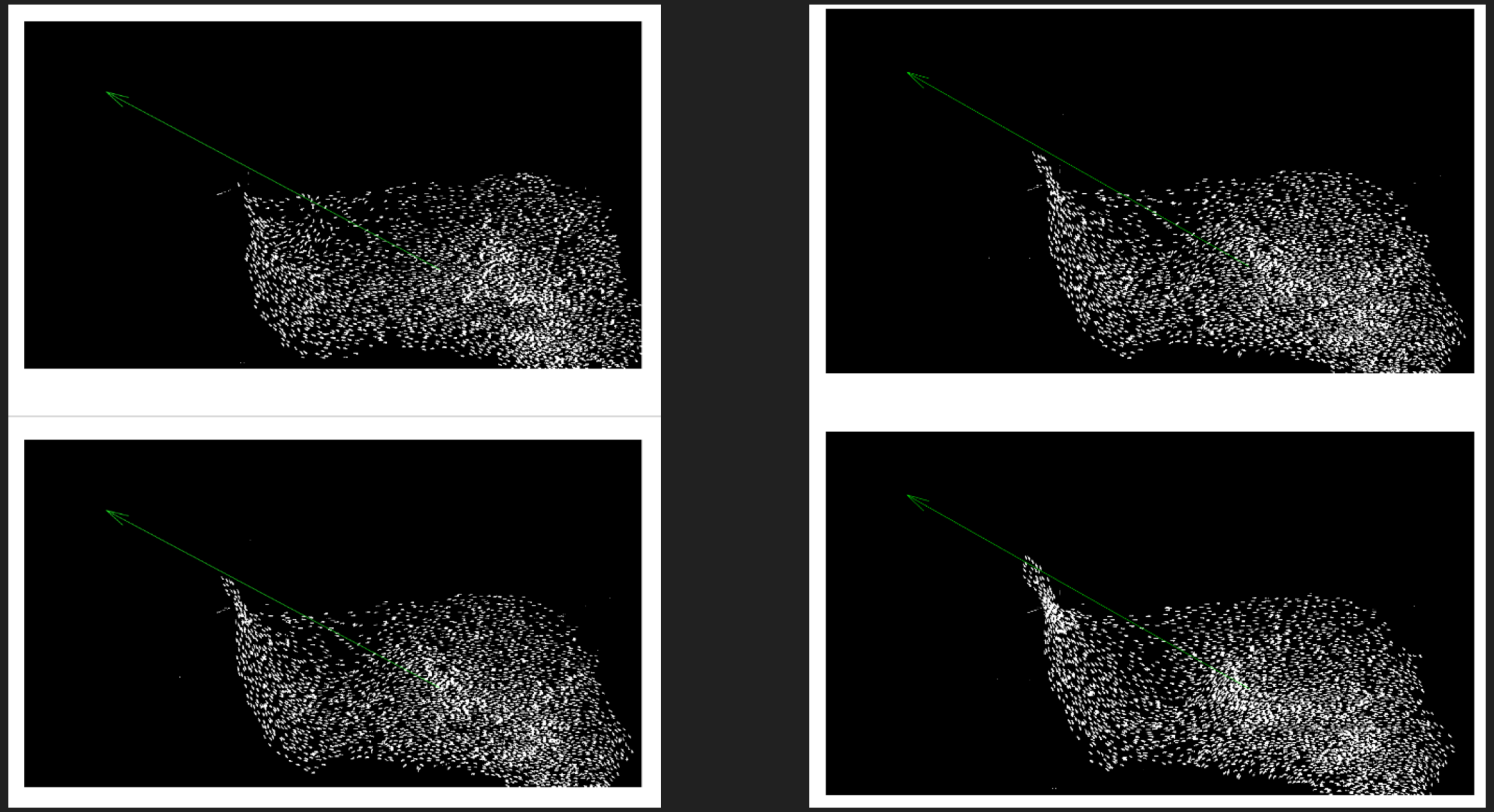
Average Velocity from Frame 2 to 3



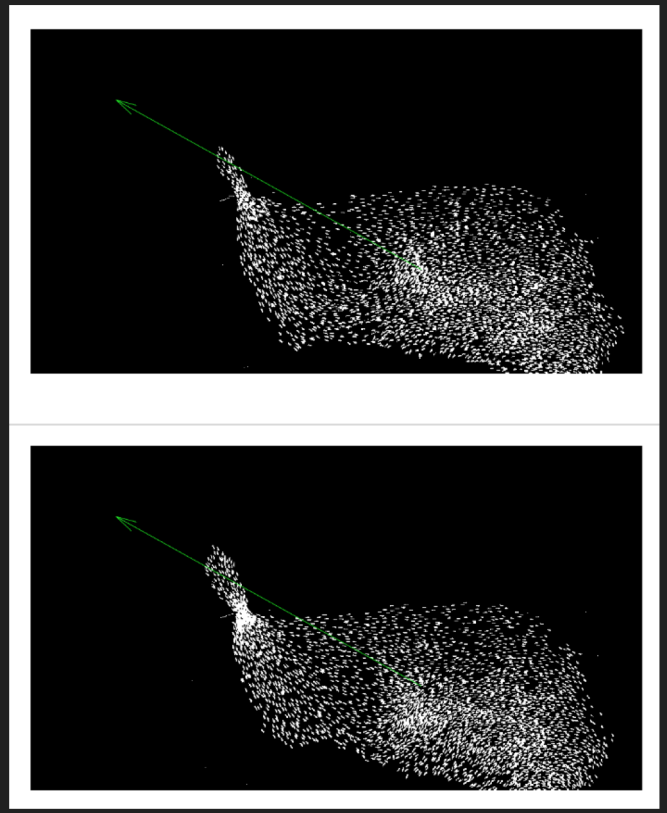
Average Velocity from Frame 3 to 4



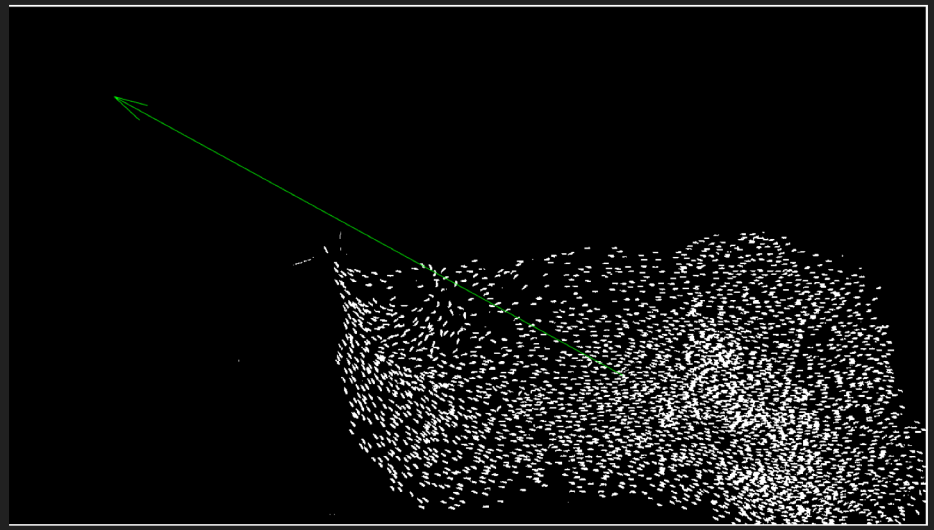
Step 4: Computing Average Velocities -- Results



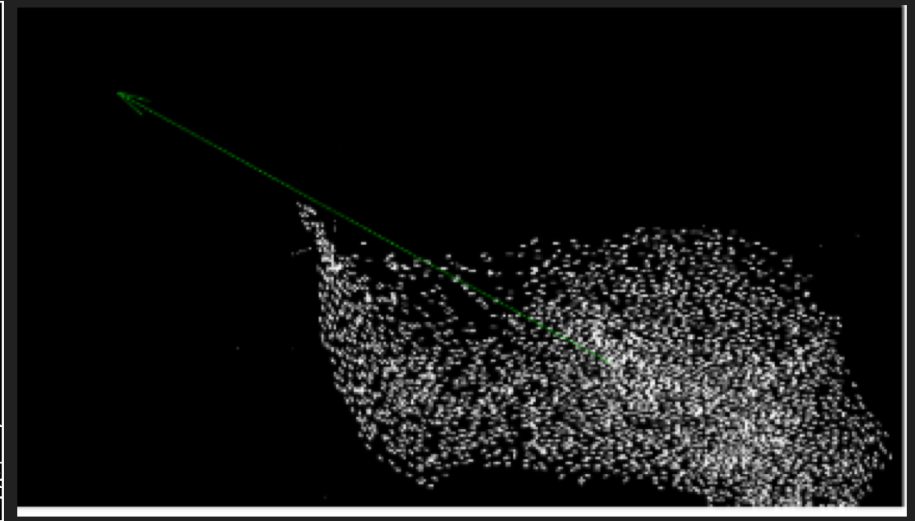
Step 4: Computing Average Velocities -- Results



Average Velocity Overlayed

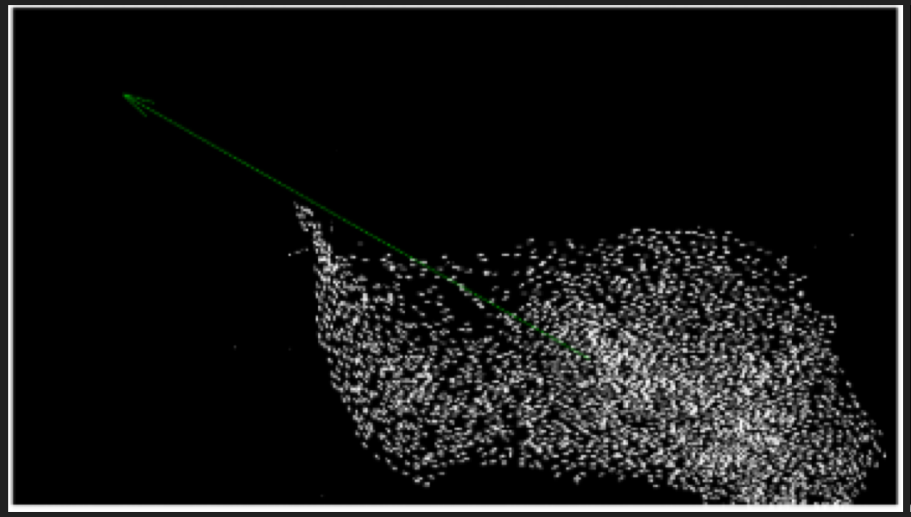


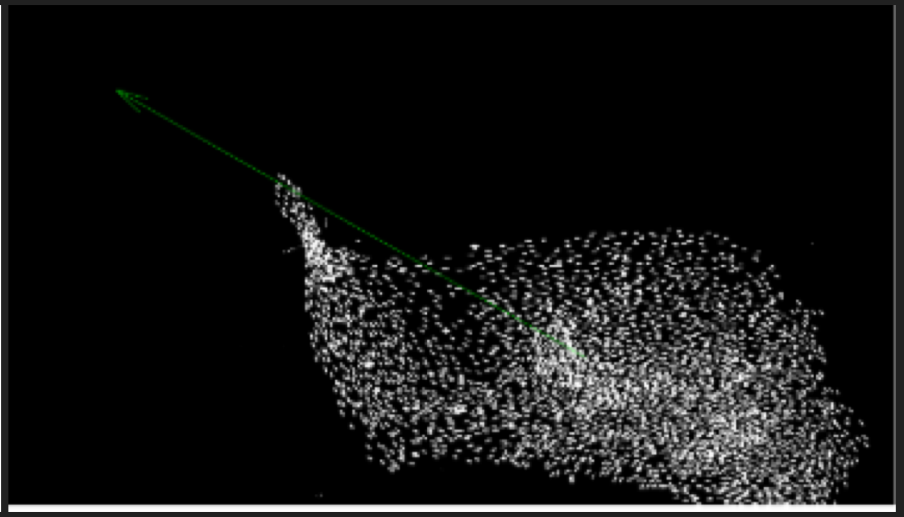
Velocity of 821.0131 Pixels/Frame
Or 235.3571 Feet/Frame



Velocity of Individual Sheep: 2.7367 Pixels/Frame
Or 0.78452Feet/Frame

Average Velocity Overlayed

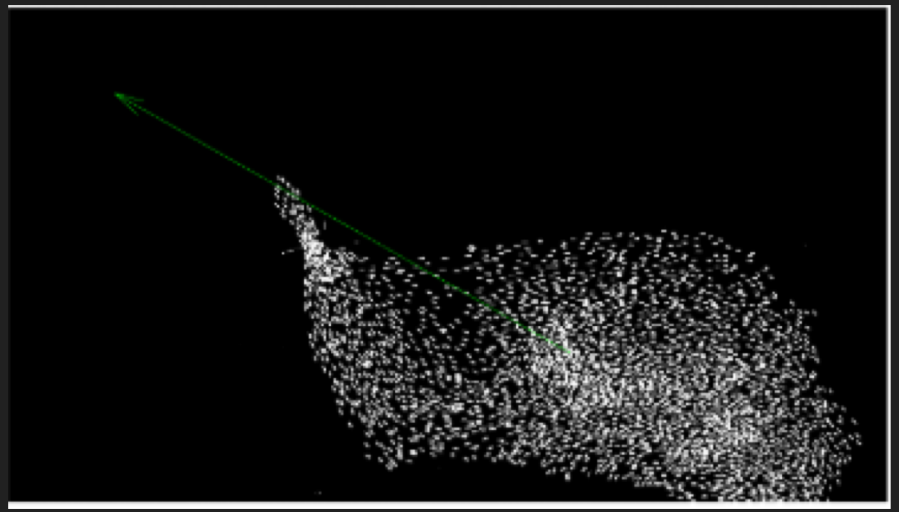


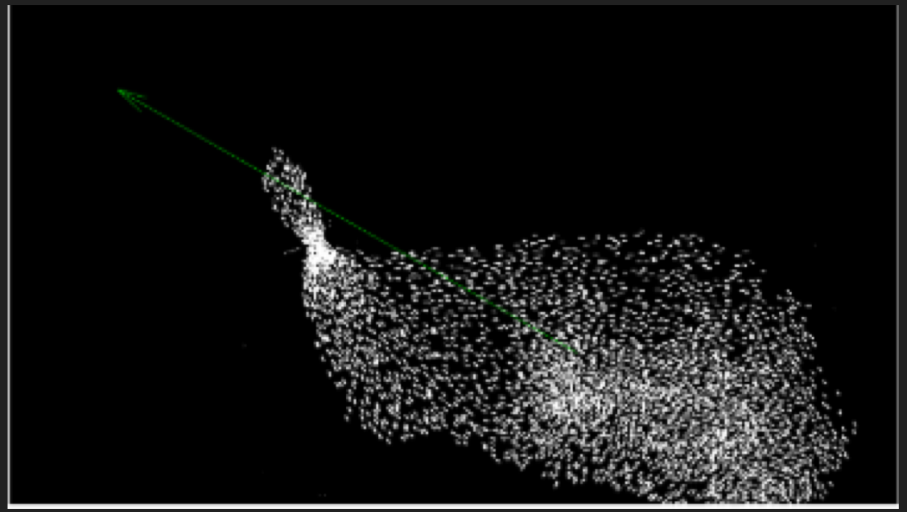


Average Velocity of 804.7128 Pixels/Frame
Or 235.3571 Feet/Frame

Velocity of Individual Sheep: 2.6824 Pixels/Frame
Or 0.76895Feet/Frame

Average Velocity Overlayed





Velocity of 792.8696 Pixels/Frame
Or 230.6843 Feet/Frame

Velocity of Individual Sheep: 2.6429 Pixels/Frame
Or 0.75763Feet/Frame

Final Observations and Conclusion

- There are several different ways to solve this problem, but we believe we achieved good results with the methods we used
- This was a difficult but engaging project, and gave a taste of how PIV is applied in engineering