Hough Transform of Particle Detector Hits

Steven Clark

April 30, 2018

1 Data Input

The goal of this project is to plot and transform data from Gaseous Electron Multiplier (GEM) particle detectors for the MUon proton Scattering Experiment (MUSE). The data is from a Fall 2017 dress rehearsal run. Originally the data was written as Midas files, then converted to ROOT files. I took the data from the ROOT file and wrote it to text files for the x and y coordinates. The next task is to read in the text file and save the data as arrays in Python. The code used for this is shown below:

```
xcoords = np.loadtxt("GEM_xcoords.txt", delimiter="\n")
ycoords = np.loadtxt("GEM_ycoords.txt", delimiter="\n")

ncoords = len(xcoords)

xscoords = xcoords[:1000]
yscoords = ycoords[:1000]
```

I use the numpy loadtxt function which automatically reads in text files and saves the data to arrays. The variable *ncoords* determines how long each array is: 500628 points. Because this is very long, I then make arrays *xscoords* and *yscoords* to hold only the first 1000 points. These are the arrays I will use for data processing to allow for faster computation.

2 Detector Hits

The GEM detectors are planar detectors that provide data in *xy* coordinates. The first step in this code is to plot the coordinates from these detectors. This is done using a 2D histogram. The code is shown below:

```
fig = plt.figure(figsize=(9,5))
 #Plot histogram of hits
 plt.subplot(121)
 plt.hist2d(xscoords, yscoords, bins=40)
 plt.xlim(0,200)
 plt.ylim(0,200)
 plt.title("GEM Hits")
 plt.xlabel("X (mm)")
 plt.ylabel("Y (mm)")
```

The 2D histogram is shown below:

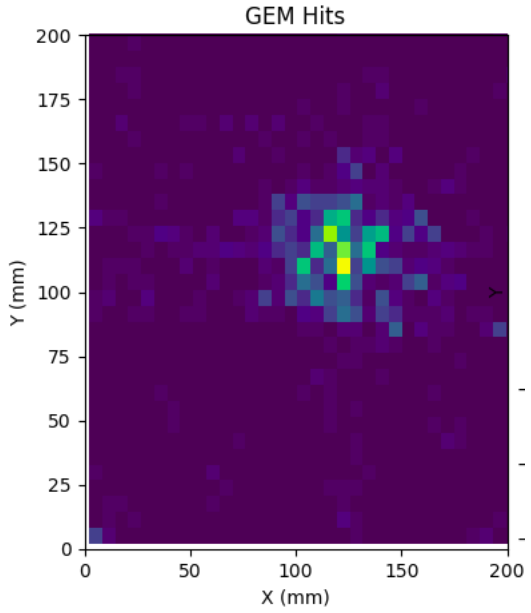

Figure 1: Histogram of hit coordinates from GEM detectors

The coordinates appear to be Gaussian and are centered around (125,125) which is the zero point for the detectors. This is further verified through a plot of all the coordinates as done in the original ROOT code. This is what we expect as the beam of incoming particles should be fired at about a Gaussian distribution.

3 Fourier Transform

Next, the 2D Fourier transform of the data is calculated and plotted as a 2D histogram. The code is shown below:

```
def Fourier(arr1, arr2):

    arr = [arr1, arr2]

    farr = np.real(fft.fft2(arr))

    return farr

#Plot Fourier transform of hits
    plt.subplot(122)
    ft = Fourier(xscoords, yscoords)
    plt.hist2d(ft[0],ft[1], bins=150)
    plt.title("Fourier Transform of GEM Hits")
    plt.xlim(-6000,6000)
    plt.xlabel("X")
    plt.ylabel("Y")
```

The Fourier transform is found using the numpy fft2 function. This is one of the fastest and most reliable ways to do a Fourier transforms and thus was chosen over writing the transform from scratch. The transform is plotted using a 2D histogram from Matplotlib. The histogram is shown below:

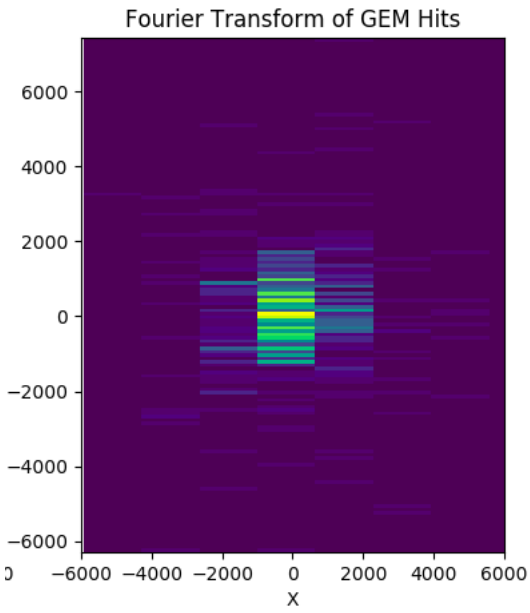

Figure 2: Histogram of Fourier transform of GEM hits

The Fourier transform is also a Gaussian as expected.

4 Hough Transform

The processing of the data comes in the form of a Hough transform. The goal of the transform is to convert coordinates from xy space to $\rho\theta$ space. A point in xy space corresponds to a line in $\rho\theta$ space, so each hit on a GEM detector will produce a line in $\rho\theta$ space. If one particle passes through all 4 detectors, which line 5 cm apart from each other, then all 4 of the lines will intersect at one point. This allows MUSE to identify particles. This intersection, which is a point in $\rho\theta$ space, corresponds to a line in xy space. This line represents the actual track of the particle.

The Hough transform used here is the Hesse Normal Form. This form takes each point and calculates ρ, the distance from the origin, and θ, the angle above the x axis [1]. This is shown below in Figure 3

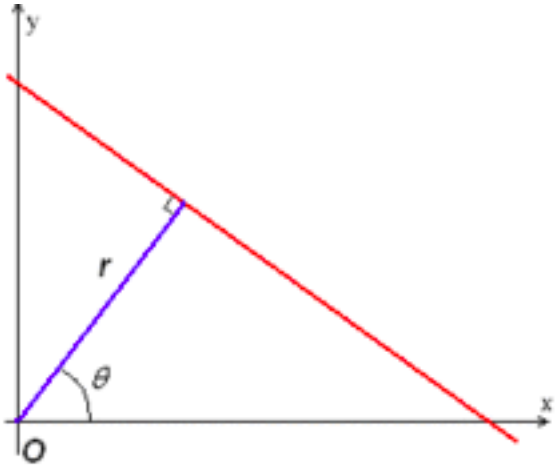

Figure 3: Hesse Normal Form of Hough Transform

Given a point in the xy plane, it is technically the set of all straight lines passing through that point that form a curve in $\rho\theta$ space and this is unique to that point [1]. This is what is plotted. The code is shown below:

```
def hough(xcoords,ycoords):
  thetas = np.deg2rad(np.arange(-90.0, 90.0, 1))
  width = 200
  height = 200

  diag_len = int(np.sqrt(width**2 + height**2) + 1)

  rhos = int(round(np.sqrt(width**2 + height**2)))

  cos_t = np.cos(thetas)
  sin_t = np.sin(thetas)
  num_thetas = len(thetas)

  accumulator = np.zeros((2*diag_len, num_thetas), dtype=np.uint8)

  for ii in range(len(xcoords)):
    rx = random.normal(0,0.5)
    ry = random.normal(0,0.5)
    x = xcoords[ii] + rx
    y = ycoords[ii] + ry

    for tt in range(num_thetas):

      rho = diag_len  + int(round(x*cos_t[tt] + y*sin_t[tt]))
      accumulator[np.abs(min(rho, 2*diag_len-1)), np.abs(min(tt, num_thetas-1))] += 1

  return accumulator, thetas, rhos
```

The Hough transform is implemented by allowing points to vote in an accumulator. The code loops through the hit coordinates, and if there is a nonzero point at the current index, the accumulator is incremented by one at that index. The accumulator is then what you plot and it will appear as a sinusoid in $\rho\theta$ space [1]. The Hough transform for the GEM hits is shown in Figure 4.

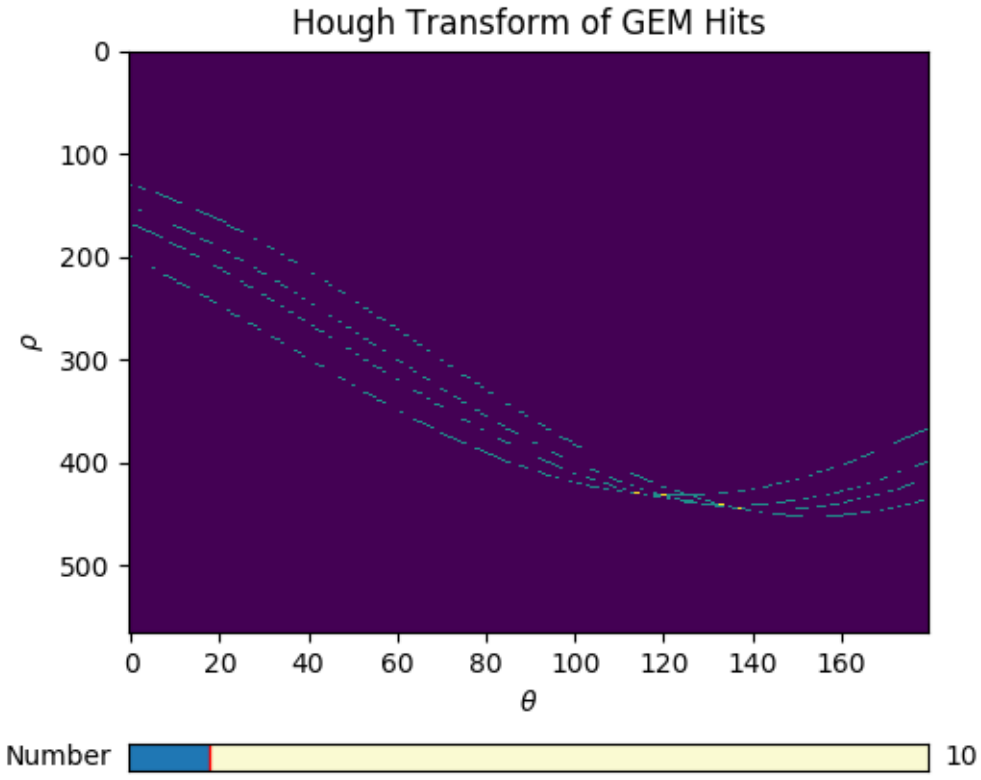

Figure 4: Hough transform of 4 GEM hits with a slider.

Figure 4 is a plot of the Hough transform of four GEM hits. Because there are four GEM detectors in a row, Figure 4 shows that this event corresponds to one particle passing through three detectors and one particle from a different event because the three of the sinusoids intersect at one point and one does not. This intersection can be converted back to xy space in which it will be a line representing the trajectory of the particle.

5 Slider

The goal of the Hough transform is to identify when a single particle passes through all detectors, which corresponds to all four sinusoids intersecting at one point. To help see when this happens, a slider was added to change which GEM hits are being transformed an plotted. The code for the slider is shown below:

```
def slide(accumulator):
  fig, ax = plt.subplots()
  plt.subplots_adjust(left=0.25, bottom=0.25)


  l = plt.imshow(accumulator, interpolation='nearest', aspect='auto')

  plt.title("Hough Transform of GEM Hits")
  plt.xlabel(r"$\theta$")
```

```
plt.ylabel(r"$\rho$")

plt.autoscale(False)

axcolor = 'lightgoldenrodyellow'
axnum = plt.axes([0.25, 0.1, 0.65, 0.03], facecolor=axcolor)

n0 = 10
snum = Slider(axnum, "Number", 0, 100, valinit=n0, valfmt='%0.0f')

def update(val):

  num = np.around(snum.val)
  p = int(num)
  l.set_data(hough(xscoords[(p):(p+4)], yscoords[(p):(p+4)])[0])

  fig.canvas.draw_idle()

snum.on_changed(update)

plt.show()
```

The slider is set up to slide through integer values between 0 and 100. The plot that is shown corresponds to the hit number that is shown by the slider and the three hits immediately after. This allows the user to slide search through all the hits and identify when one particle passed through all four detectors.

References

[1] KUtkarsh Sinha, *The Hough Transform*. http://aishack.in/tutorials/hough-transform-normal/. April 28, 2018