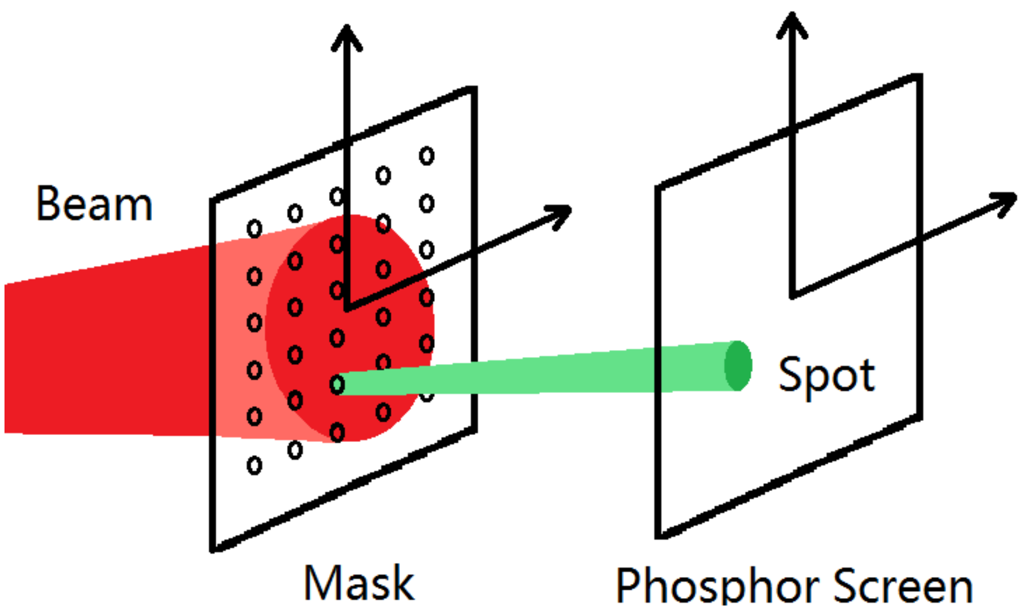
PAT Instruction Manual

Pepper-Pot Analysis Tool



Ding Wang

Department of Physics

wangd@nscl.msu.edu

Version 2.0

May 1, 2013

Contents

1 Introduction

In order to minimize beam losses, it is very important to have a good knowledge of the beam condition. Among all the beam parameters, one of the most important parameters is the emittance of the particle beam. We use the device Pepper-Pot Emittance Meter (PEM) to study the beam phase space.

A Pepper-Pot Emittance Meter is a particle beam imaging system which can reconstruct the 4D beam transverse phase space distribution of the charged particle beam. Compare with the other emittance measurement devices such as Allision Scanner and Slit-Slit Scanner, Pepper-Pot Scanner can give more detailed information which include both $X - X'$ and $Y - Y'$ transverse phase space. The correlation between the $X - X'$ and $Y - Y'$ plane can be measured with Pepper-Pot Scanner which can not be measured by Allision Scanner and Slit-Slit Scanner.

2 Functional Principle

2.1 Pepper-pot scan principle

A Pepper-Pot Emittance Meter is a particle beam imaging system, consist of Pepper-Pot mask, Multi-channel plate(MCP), Phosphor screen and CCD camera. Figure 2.1 shows the scheme of our Pepper-Pot Meter.

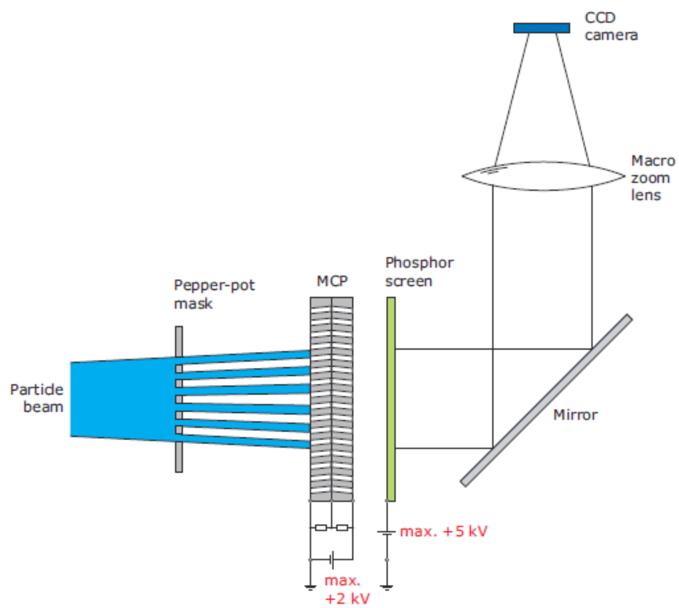


Figure 2.1: Pepper-pot scheme

The Pepper-Pot mask is a metal plate with a number of holes on it. The diameter of the holes is typically very small which also called pinholes. In our Pepper-Pot meter, the holes diameter is $0.1mm$ and the distance between the holes is $1mm$. To measure the beam emittance, the Pepper-pot Meter is mounted in a diagnostic box with a high vacuum environment. The beamlets transmitted by the Pepper-pot mask will reach the MCP and Phosphor screen after a fixed distance and the image is recorded by CCD camera which is typically a pattern of holes. The emittance of the beam can be calculated from the position and the intensity distribution of these spots. The existence of the solution is determined by the requirement that the patterns of the beamlets do not overlap in the imaging plane. The following figure 2.2 shows the imaging principle of Pepper-pot Meter.

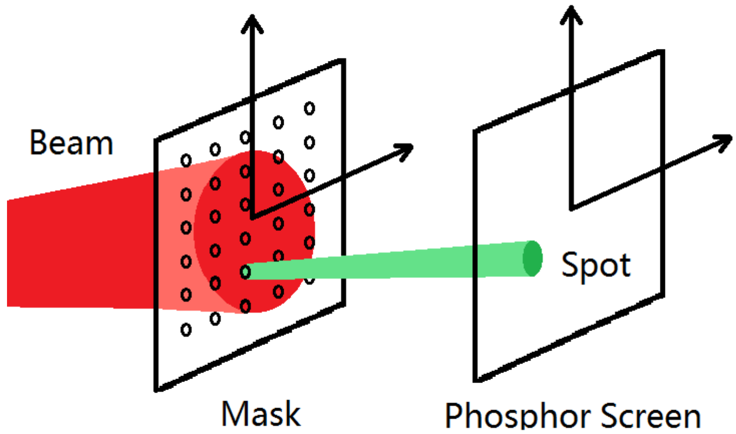


Figure 2.2: Pepper-pot principle

In the early design of the Pepper-Pot meter, meters kapton foils or photographic films were used as the beam position detectors. Scintillation screen with a CCD camera is widely used nowadays which can directly readout the intensities of beam distribution. The scintillation screen also arise lots of problems such as inhomogeneity, nonlinear response, easily damaged and aging problem.

2.1.1 Data analysis principle

The software for the Pepper-pot Emittance Meter can be separated into two parts: The acquisition program to capture the image from the CCD camera and the analysis program to reconstruct the beam transverse phase space from measurement data. The acquisition program we use is the official program for the CCD camera, which can capture the images from the camera with setting parameters. I designed the analysis program called Pepper-pot Analysis Tool(PAT), which can reconstruct the 4-D beam phase space from the Pepper-pot measurement. The analysis program is coding with Qt C++, which can be used under open source (LGPL v2.1) or commercial term. Qt is a cross-platform application and UI framework for developers using different languages such as C++ or QML.

Here I will present the algorithm to reconstruct the beam transverse phase space and discuss the accuracy of the measurement. Let d denote the distance between the holes of the Pepper-pot mask, D denote the distance from the mask to the MCP, r denote the real dimension of one CCD pixel. For the Pepper-Pot mask, let (i, j) denote the index of each hole. Therefore, $(0, 0)$ means the center of the mask. For the image, let (p, q) denote the position of the pixel and I_{pq} denote the intensity of the pixel. First, we need to setup one center pixel which is the center of the beamline, let it be (p_0, q_0). Please note that, with different (p_0, q_0), the shape of the reconstruct beam phase space will not changing. Therefore, we can roughly setup one (p_0, q_0) and recalculate the beam center after we find out the beam phase space.

Since the patterns from the holes are not overlapped, we can divided the pixel array to sub-arrays. For each hole (i, j), we can find the matching beam spot A_{ij} in the range of $(x_{imin}, x_{imax}, y_{imin}, y_{imax})$. Thus, for each pixel $(p, q) \in A_{ij}$, we can calculate the particle 4-D parameter (x, y, x', y') with these equations:

$$\begin{aligned}
x_{ip} &= i \cdot d \\
y_{jq} &= j \cdot d \\
x'_{ip} &= \frac{r(p - p_0) - i \cdot d}{D} \\
x'_{jq} &= \frac{r(q - q_0) - j \cdot d}{D}
\end{aligned} \tag{2.1}$$

To obtain the 4-D phase space, we need to calculate the beam intensity distribution $\rho(x, y, x', y')$. Since we already find out the particle 4-D parameter $(x_{ip}, y_{jq}, x'_{ip}, y'_{jq})$ for each pixel of the image which means $\rho(x_{ip}, y_{jq}, x'_{ip}, y'_{jq}) = I_{pq}$. Therefore, we can find out $\rho(x, y, x', y')$ via

$$\rho(x, y, x', y') = \int\int \rho(x_{ip}, y_{jq}, x'_{ip}, y'_{jq}) dpdq = \sum_p \sum_q \rho(x_{ip}, y_{jq}, x'_{ip}, y'_{jq}) \tag{2.2}$$

Therefore, the 4-D phase space distribution $\rho(x, y, x', y')$ can be construct from the image data. To compare with other measurements with 2-D transverse phase space, we can also calculate the horizontal and vertical transverse emittance distribution $\rho(x, x')$ and $\rho(y, y')$. Further more, we can calculate the correlation distribution such as $\rho(x, y')$ and $\rho(y, x')$.

$$\begin{aligned}
\rho(x, x') &= \int\int \rho(x, y, x', y')dydy' \\
\rho(y, y') &= \int\int \rho(x, y, x', y')dxdx' \\
\rho(x, y') &= \int\int \rho(x, y, x', y')dydx' \\
\rho(y, x') &= \int\int \rho(x, y, x', y')dxdy'
\end{aligned} \tag{2.3}$$

After we construct the horizontal and vertical transverse emittance distribution, we can easily obtain the horizontal and vertical emittance ϵ_x and ϵ_y. Further more, Assume the motions at horizontal and vertical planes are not coupled, we can calculate the twiss parameters for both horizontal and vertical phase space. The equations for calculating the transverse Twiss parameters are given by

$$\begin{aligned}
< xx > &= \int\int (x- < x >)^2 \rho(x, x')dxdx' \\
< x'x' > &= \int\int (x'- < x' >)^2 \rho(x, x')dxdx' \\
< xx' > &= \int\int (x- < x >)(x'- < x' >)\rho(x, x')dxdx' \\
\epsilon_x &= \sqrt{< xx >< x'x' > - < xx' >^2} \\
\alpha_x &= -\frac{< xx' >}{\epsilon_x} \\
\beta_x &= \frac{< xx >}{\epsilon_x} \\
\gamma_x &= \frac{< x'x' >}{\epsilon_x}
\end{aligned} \tag{2.4}$$

$$
\begin{aligned}
<yy> &= \int\int (y-<y>)^2 \rho(y,y')dydy' \\
<y'y'> &= \int\int (y'-<y'>)^2 \rho(y,y')dydy' \\
<yy'> &= \int\int (y-<y>)(y'-<y'>)\rho(y,y')dydy' \\
\epsilon_y &= \sqrt{<yy><y'y'> - <yy'>^2} \\
\alpha_y &= -\frac{<yy'>}{\epsilon_y} \\
\beta_y &= \frac{<yy>}{\epsilon_y} \\
\gamma_y &= \frac{<y'y'>}{\epsilon_y}
\end{aligned}
\tag{2.5}
$$

$\rho(x, x')$, $\rho(y, y')$ need to be normalized before using these equations.

2.2 Denoise principle

We can improve the image quality by improving the device enclosure and substrate the background image, however, there is still noise background after these process. Therefore, we need a denoise process to improve the accuracy of the measurement data. Different process might be chosen for different cases.

2.2.1 Signal mark

In this process, the program can automatically find the signal part from the measurement image. The whole process contains the following steps.

(1) Reading parameters from the configure file, including minimum intensity min_I and minimum area min_A.

(2) Detect the background intensity, let it be b.

(3) Copy the original data to a signal array.

(4) Do a median filtering signal processing on the signal array, which can greatly decrease the speckle noise and salt-and-pepper noise.

(5) Do a mean filtering signal processing on the signal array, which can make a image

with smooth intensity change.

(6) For the signal array, if the pixel intensity is larger than $b + min_I$ and spot size is larger then min_A, then mark it with 1; otherwise, mark it with 0.

The signal mark process perform two kinds of noise reduction on the signal array, which can identify most of the real signal. This process is only apply on the signal array, not the real data.

2.2.2 Weak noise background

If there is only weak noise on the background, the signal mark process is enough to find out all the real signal. In this case, we only need to remove the data which signal array marked with 0.

2.2.3 Strong noise background

If there is strong background noise on the background, the signal mark process is not able to find out all the real signal. In this case, we can do a Gaussian curve fitting process. Assume the beam spot intensities observe a two dimension gaussian distribution as a ideal case. Then we can do the curve fitting with the following equation

$$I(x, y) = A \cdot exp[-\frac{(x - x_0)^2}{2\sigma_x^2} - \frac{(y - y_0)^2}{2\sigma_y^2}] \tag{2.6}$$

In this case, there is strong background and might also have overexposure pixels. Therefore, we need to improve our fitting curve as shown in Figure 2.3. The fitting rate is defined as

$$Fittingrate = \frac{Overlap}{Combination} \tag{2.7}$$

Therefore, we can find out the fitting parameters A, x_0, y_0, σ_x, σ_y by finding out the maximum fitting rate.

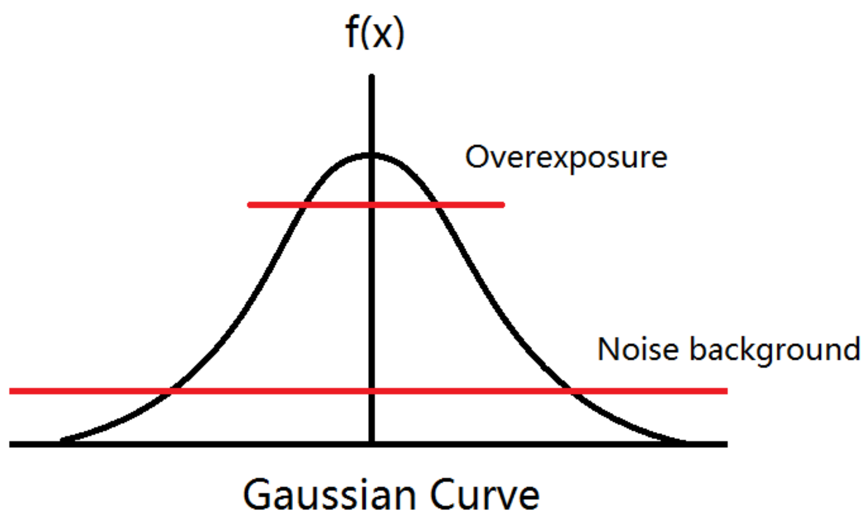
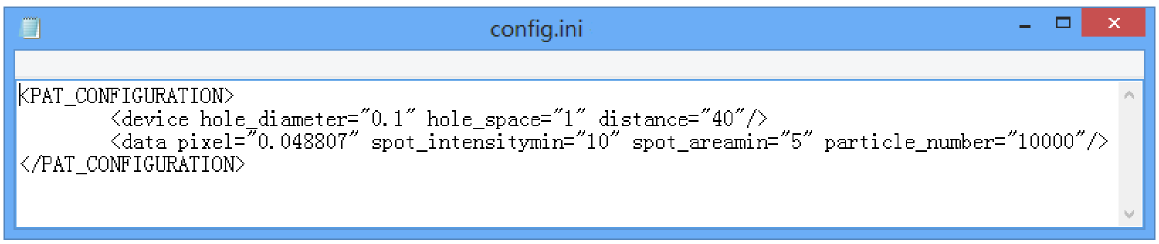


Figure 2.3: Gaussian fitting in PAT

3 PAT interface

In this section, we can show the PAT interface and reconstruct the 4D-beam phase space with the data analysis of one measurement data. Figure 3.1 shows a typical configure file for the Pepper-pot meter at Box 3. The descriptions for each parameter in the configure file can be found in Appendix A.




```
<PAT_CONFIGURATION>
        <device hole_diameter="0.1" hole_space="1" distance="40"/>
        <data pixel="0.048807" spot_intensitymin="10" spot_areamin="5" particle_number="10000"/>
</PAT_CONFIGURATION>
```

Figure 3.1: PAT: Configure file

3.1 Image load

Before we can analyze the image, we need to load the image and identity every spot from the captured image. The descriptions of functions are shown in Table 3.1.

Function	Description
Load image	Load the CCD image
Select range	Select the signal range from the image
Rotate image	Rotate the image if there is a tilt angle
Separate spots	Separate every spot in the image by grids
Auto separate	Automatically detect the position of every spot and do the separation

Table 3.1: PAT interface: Image load

3.1.1 Load image

The first step is load the CCD captured image. The PAT interface for load image is shown in Figure 3.2.

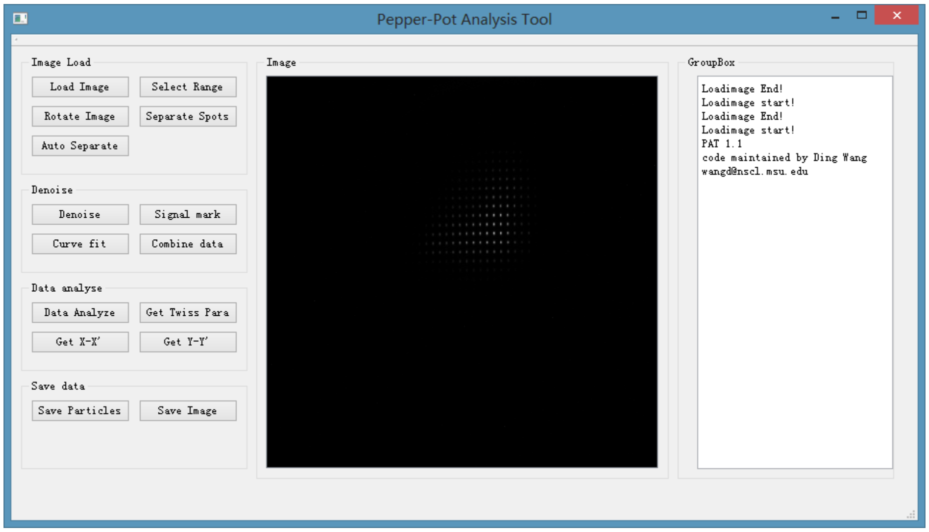


Figure 3.2: PAT interface: Load image

3.1.2 Select Range

The signal image is usually a small part of the captured image, then we can select the range of the signal part. The PAT interface for select range is shown in Figure 3.3.

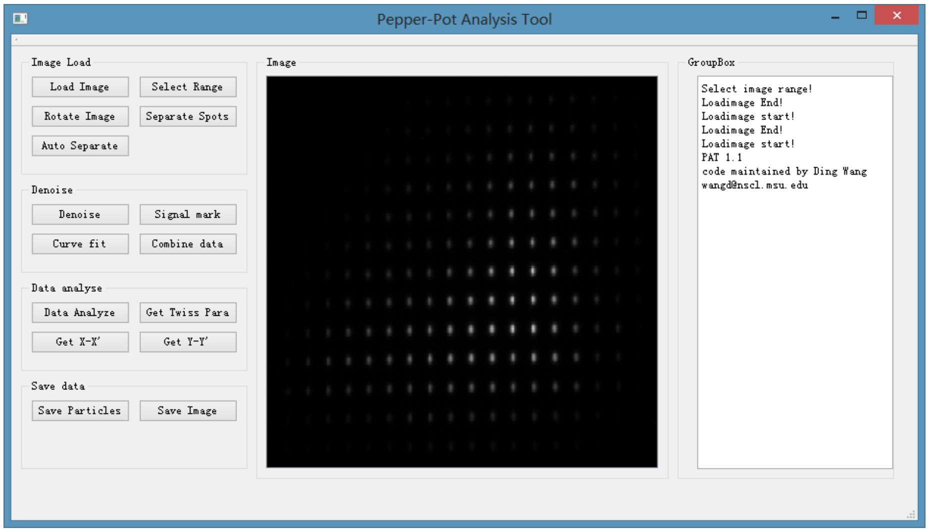


Figure 3.3: PAT interface: Select range

3.1.3 Separate spots

The signal image consist of a set of beam spots which must not be overlapped. In this case, we can separate the beam spots with grids. The PAT interface for Separate spots is shown in Figure 3.4.

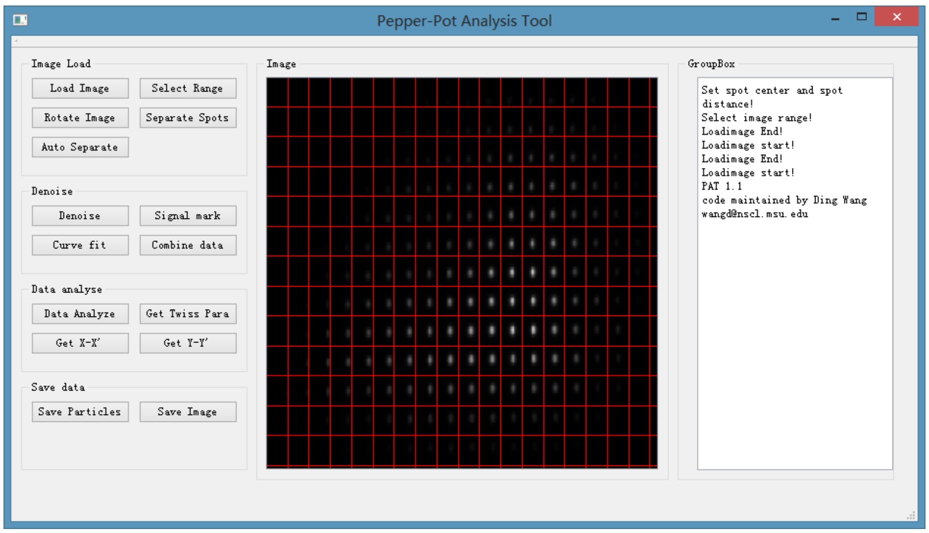


Figure 3.4: PAT interface: Separate spots

3.2 Denoise

If there is noise background on the image, we can perform the denoise process. The descriptions of functions are shown in Table 3.2.

Function	Description
Signal mark	Mark the signal part base on the intensity
Denoise	If there is weak background noise, simply denoise with the signal mark result
Curve fit	If there is strong background noise, then we need to do the Gaussian curve fitting
Combine data	Combine the real data and Gaussian curve fitting data

Table 3.2: PAT interface: Denoise

3.3 Data analysis

For data analysis, we can caculate the Twiss parameters and get the phase space plots. The descriptions of functions are shown in Table 3.3.

Function	Description
Data analysis	Analyze the image data and generate a set of particles
Get Twiss parameters	Calculate the Twiss parameters, including α, β, γ and emittance
Get X-X’	Get the beam phase space plot on X-X’ plane
Get Y-Y’	Get the beam phase space plot on Y-Y’ plane

Table 3.3: PAT interface: Data analysis

3.3.1 Get phase space

After separate all the spots, we can do the data analysis to reconstruct the beam phase space. We can also obtain the Twiss parameters and generate a set of particles for further beam simulation. The PAT interface for Get phase space is shown in Figure 3.5.

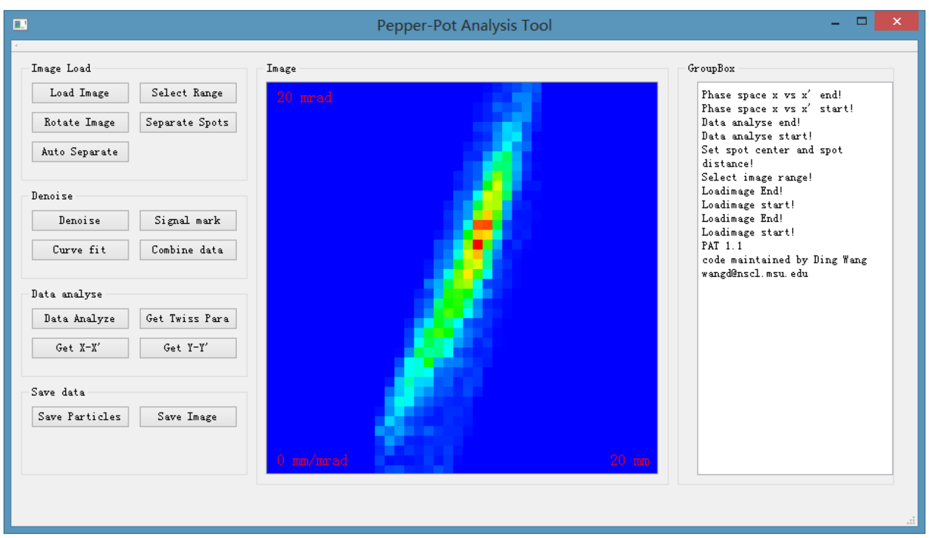


Figure 3.5: PAT interface: Get phase space

3.3.2 Get phase space after denoise

The captured images from CCD are usally come with a noise background. Figure 3.6 shows the reconstruct phase space after denoise process. Compare with Figure 3.5, we can see the denoise function can help removing the noise signal.

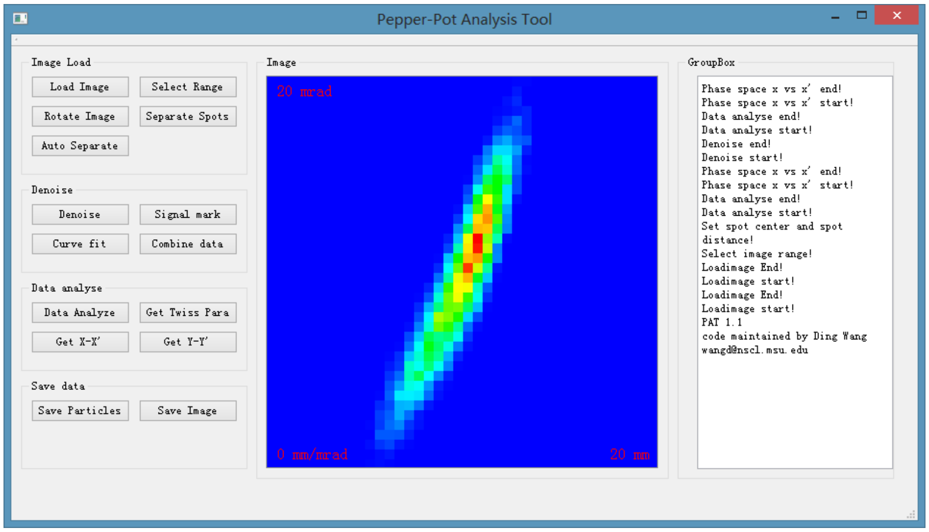


Figure 3.6: PAT interface: Get phase space after denoise

3.4 Save data

We can saved our phase space plot and generated particles to file. The generated particles can be used for future beam simulation or more data analysis like beam coupling. The descriptions of functions are shown in Table 3.4.

Function	Description
Save particles	Save a set of particles to file with Dynac format
Save image	Save the phase space plot to file

Table 3.4: PAT interface: Save data

4 Measurement example

4.1 Pepper-pot Meter at Box 3

Our Pepper-pot Emittance Meter is located at box 3 before the RFQ entrance. Figure 4.1 shows the position of the PEM.

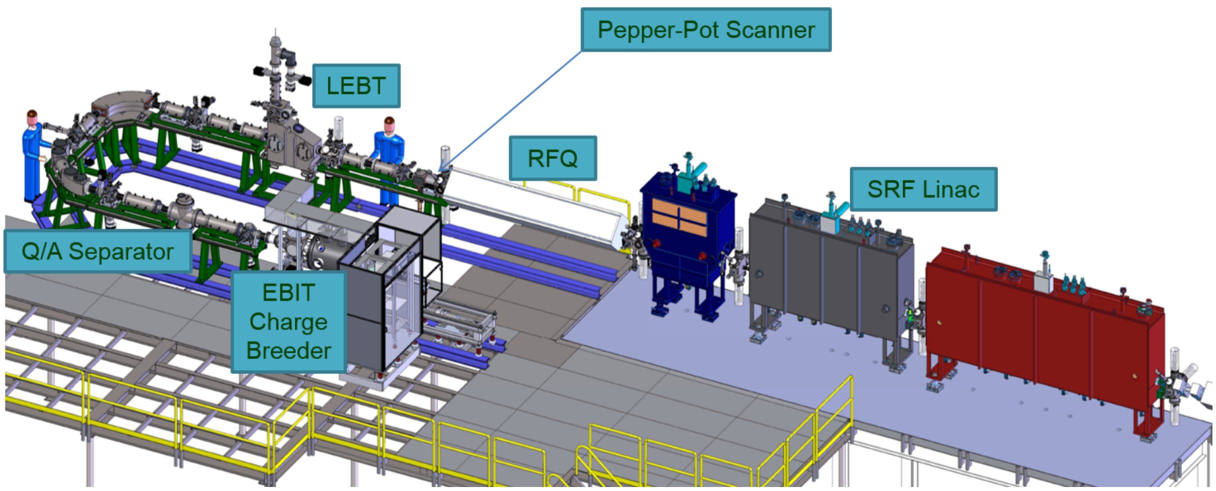


Figure 4.1: Pepper-pot position

4.2 Experiment setup

The Pepper-pot Emittance Meter is mounted in diagnostic box 3 which is right before the RFQ. We need to turn off the solenoid before the Pepper-pot to prevent getting overlapped pattern. To obtain good image quality, we had done several improvements:

1. Improve the device enclosure to avoid the light
2. Use maximum shutter number
   2.1 Effect: shutter>gain>AE(did not see any effect)
   2.2 Gain will increase the noise background, better set to 0
3. Record the background

Figure 4.2 shows the image changing after we apply these improvements. For our most recent measurements, we obtained very good images with only slightly noise background.

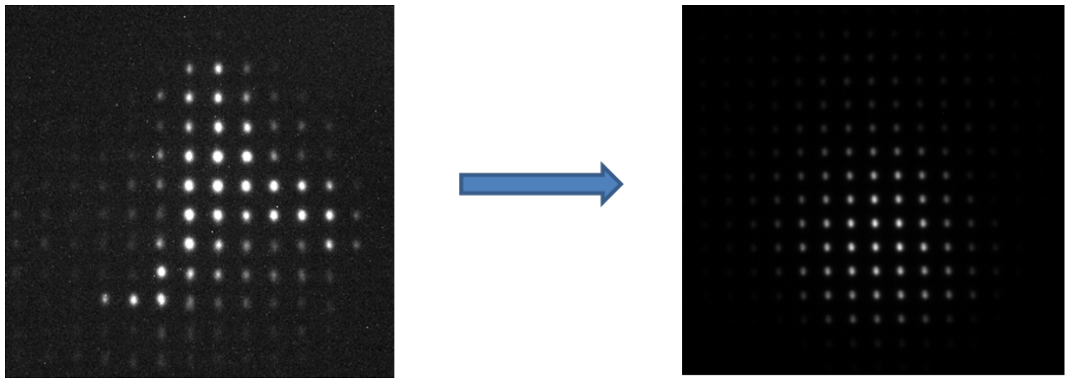


Figure 4.2: Pepper-pot measurement enclosure improvement

4.3 Measurement result

Several measurements had been done with Pepper-pot Meter and PAT program was used to analyze the measurements data and compare with the ideal case from Dyanc simulation. The beam properties of one recent measurement are: He^{+} beam, current 55 pA. The measurement result compare with Dynac simulation is showing in Figure 4.3.

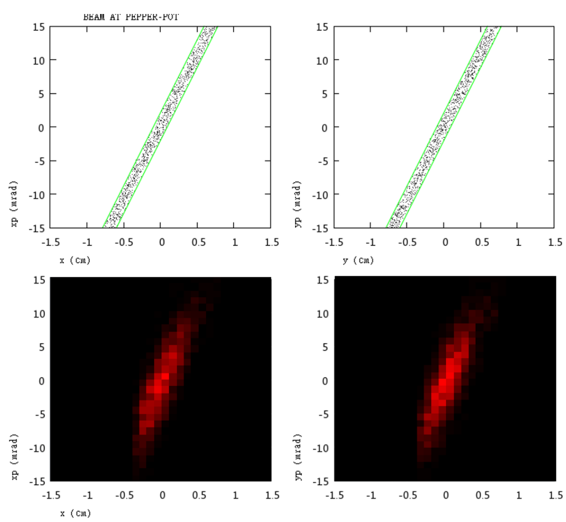


Figure 4.3: Pepper-pot measurement result compare with simulation

To better understanding our beam dynamics and compare with the ideal case, I did a series of measurements by changing the solenoid strength and record the Pepper-pot images as a viewer. From our measurement, we noticed a waist point which also agree with the

Dynac simulation. The series of beam images are showing in Figure 4.4.

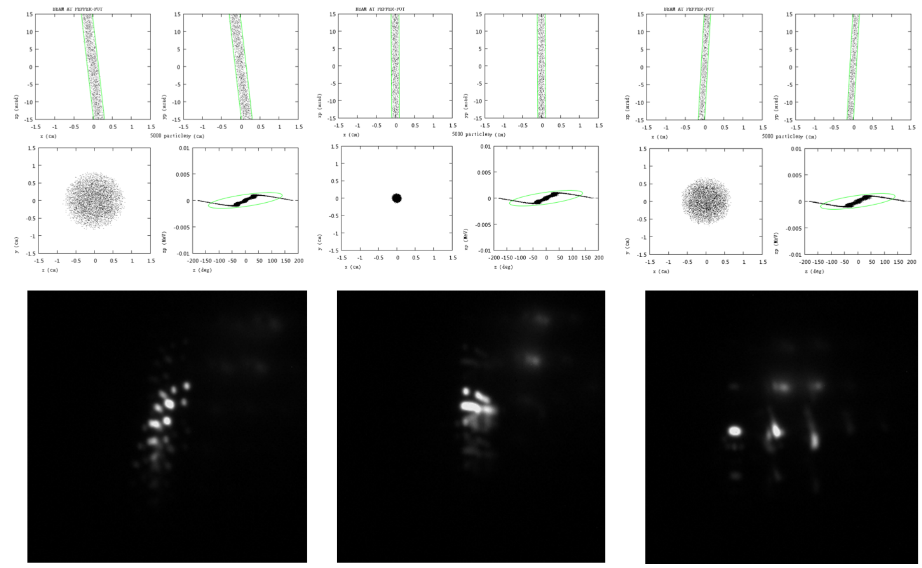


Figure 4.4: Pepper-pot simulation and measurement with I=225A, 250A, 275A

4.4 Monte carlo method

Monte Carlo methods are widely used in physical and mathematical problems, which are computational algorithms that can repeated random sampling to obtain numerical results we want. In this case, we want to obtain a set of particles agree with the Pepper-pot measurement result. Therefore, after we find out the 4-D phase space distribution $\rho(x, y, x', y')$, we can using Monte carlo method to generated the particles. We can consider transverse phase space is independent with longitudinal phase space and then using the ideal longitudinal phase space to obtain the 6-D beam particle set. This particle set can used for further beam simulation.

4.5 Beam coupling of transverse phase space

Pepper-pot is doing a 4-D phase space measurement and we can use PAT program to reconstruct the 4-D phase space. The data of the 4-D phase space can be saved within a set of particles generated from the 4-D phase space with Monte carlo method. Figure 4.5 show a typical phase space plot from the saved particles data. We can see these data mostly agree with our measurement result and contains the full information from the Pepper-pot measurement. The reconstruct particles data can be used to analyse the beam coupling of transverse phase space.

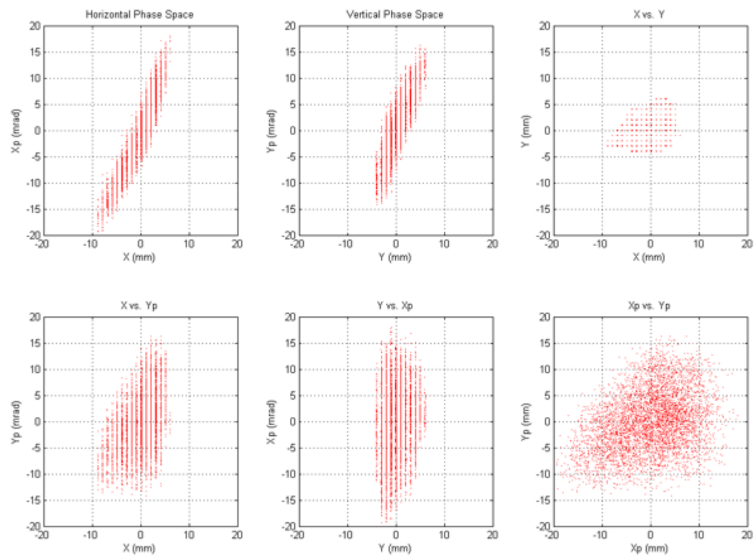


Figure 4.5: Phase space plot for Pepper-pot reconstruct particles

With the full set of particles data, we can find the correlations between x, x', y, y'. The result is showing in Table 4.1.

Correlation	X	X'	Y	Y'
X	1	2.1069	0.2454	0.8364
X'	2.1069	5.1025	0.3042	1.3427
Y	0.2454	0.3042	0.5385	1.1632
Y'	0.8364	1.3427	1.1632	3.4424

Table 4.1: Normalized correlations of reconstruct particles

To compare with the measurement data, we set up the Dynac simulation with the ideal case to obtain the beam particles at the position of the Pepper-pot. The phase space plot is shown in Figure 4.6. The calculated correlations is shown in Table 4.2. For the ideal simulation, there is no correlations between horizontal and vertical transverse phase space.

For the beam particles reconstruct from the Pepper-pot measurement, we can clearly see that the real beam has a strong coupling effect. We can further study the coupling effect with Dynac simulation.

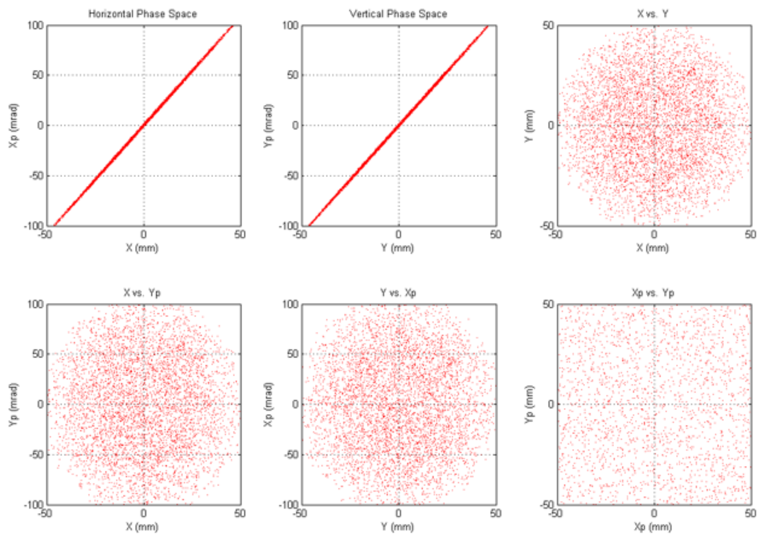


Figure 4.6: Phase space plot for Pepper-pot ideal particles

Correlation	X	X'	Y	Y'
X	1	2.1651	-2.00E-04	-5.00E-04
X'	2.1651	4.689	-5.00E-04	-0.001
Y	-2.00E-04	-5.00E-04	1.0011	2.1659
Y'	-5.00E-04	-0.001	2.1659	4.6878

Table 4.2: Normalized correlations of ideal particles

5 Program management

The PAT project files are saving at network drive/projects/rea3/software/PAT. The folder contains the compiled programs, source code, user manual, environment files and measurement data.

5.1 Environment setting

PAT is a QT based program. When using the PAT program in windows, several dll files are needed, including

QT environment files: $libstdc++-6.dll$, $QtCored4.dll$, $QtGuid4.dll$, $QtXmld4.dll$.

5.2 Code maintenance

PAT is a QT based program and the source code is written by C++. PAT source code is saving at network drive/projects/rea3/software/PAT/code. PAT source code is maintained by Ding Wang (wangd@nscl.msu.edu).

A PAT configure

```
<PAT_CONFIGURATION> //configuration root
  <device                //parameters for device
    hole_diameter        //diameter for the hole on the mask
    hole_space           //space between holes on the mask
    distance             //distance between mask and MCP
  />
  <data                  //parameters for data analysis
    pixel                //dimension for each pixel on the CCD camera
    spot_intensitymin    //minimum intensity for image denoise
    spot_areamin         //minimum spot area for image denoise
    particle_number      //generate particles number
  />
</PAT_CONFIGURATION>     //end of configuration
```