



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**

3D AUGMENTED REALITY

PROJECT REPORT - B.2

**Casagrande Nicola, De Toni Alberto
... - 1236657**

MASTER DEGREE IN ICT

02 · 2021

Local feature compression using autoencoders - SfM tests

Design a compression strategy for local SURF descriptors using autoencoders. Training data can be generated using the images of dataset Portello and Castle. Testing must be done on dataset FountainP-11 and Tiso (available at https://github.com/openMVG/SfM_quality_evaluation/tree/master/Benchmarking_Camera_Calibration_2008 and <http://www.dei.unipd.it/~sim1mil/materiale/3Drecon/>). Software must be implemented in MATLAB, Keras or Pytorch.

Testing on 3D reconstruction using SfM: The reconstructed descriptors (only for the test set) are used to perform a SfM reconstruction using COLMAP (using the two test dataset).

Programming languages: MATLAB/Python/C++.

1 Introduction

The experiment was firstly performed by prof. Arecchi Fortunato around 1965 and it exhibits the difference between a coherent and a thermal source. Despite the important results, the setup is quite straightforward: it consists in a laser source, an *Arecchi wheel* (a reflecting sandpaper planar wheel) and a photodetector capable of collecting light via a single-mode fiber. The laser source shines on the *Arecchi wheel* and reflects inside the photodetector. If the wheel is still then the photodetector “sees” a coherent light source, while if it is spinning then a thermal source is simulated, because of the multiple reflections from the rough surface of the sandpaper. In figure 1 the setup of the experiment is graphically represented.



Figure 1: Setup for the Arecchi experiment.

The objective of the experience is to take an interval of time T (a good one is around $15\mu s$) and count how many photons reached the detector (*occurrences*)

in about 2s of data collection. The histogram of the occurrences with respect to the number of photons in the time bin T should follow a statistic probability in the two cases:

- *still wheel*: a *Poisson* distribution $\mathcal{P}(\lambda) = \frac{\lambda^n \exp(-\lambda)}{n!}$ with $n = 0, 1, 2, \dots$;
- *spinning wheel*: a *Bose-Einstein* distribution $BE(\lambda) = \frac{1}{\lambda + 1} \left(\frac{\lambda}{\lambda + 1} \right)^n$ with $n = 0, 1, 2, \dots$

where λ represents the mean arrival of photons in T .

2 Data

The data of the experiment are collected in 10 files for both regimes (still and spinning wheel). Each file contains a dataset with two columns, being the first the time tags in *machine time* ($\approx 81ps$) and the other the channel, like this:

| | |
|--------|-----|
| 0, | 1 |
| 18009, | 1 |
| 18835, | 1 |
| ... | ... |

In this case there is only one channel collecting the data so the last column isn't useful. To convert the time tags from machine time to actual time it's sufficient to multiply the first column by $81ps$ obtaining the time tags in seconds. The data can be merged together by putting in column all the datasets and making sure that at each iteration the time begins where it left (for example, if the first dataset ends at 2s the second must begin at 2s).

3 Development

Having the time tags in seconds it is sufficient to take a time bin ($T = 15\mu s$), and count how many of the dataset "clicks" fall inside the range $[nT, (n+1)T]$, with $n = 0, 1, 2, \dots$. Plotting on the x-axis the number of clicks and on the y-axis the number of occurrences inside the time slot, an histogram is derived. The empirical distribution $P(n)$ can be found by dividing the number of clicks $h(n)$ by N , number of occurrences (a normalization factor): $P(n) = \frac{h(n)}{N}$. The result is shown on figure 2 and 3.

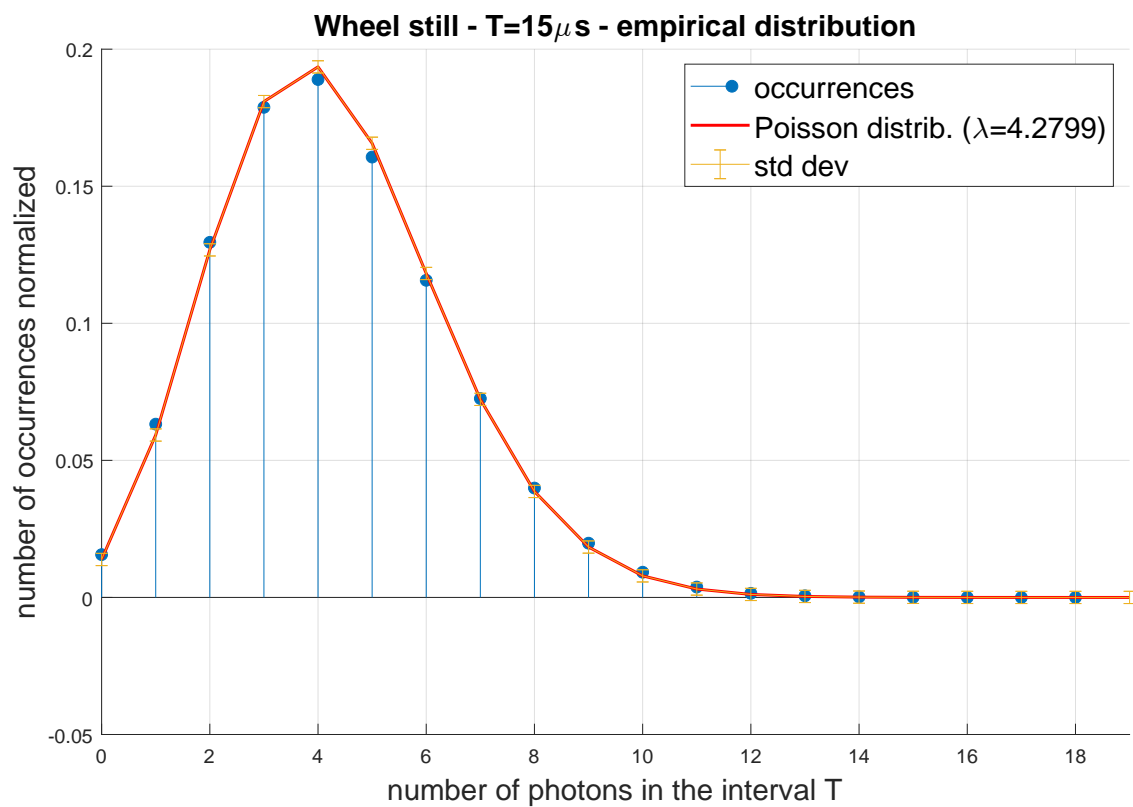


Figure 2: Empirical distribution with $T = 15\mu\text{s}$.

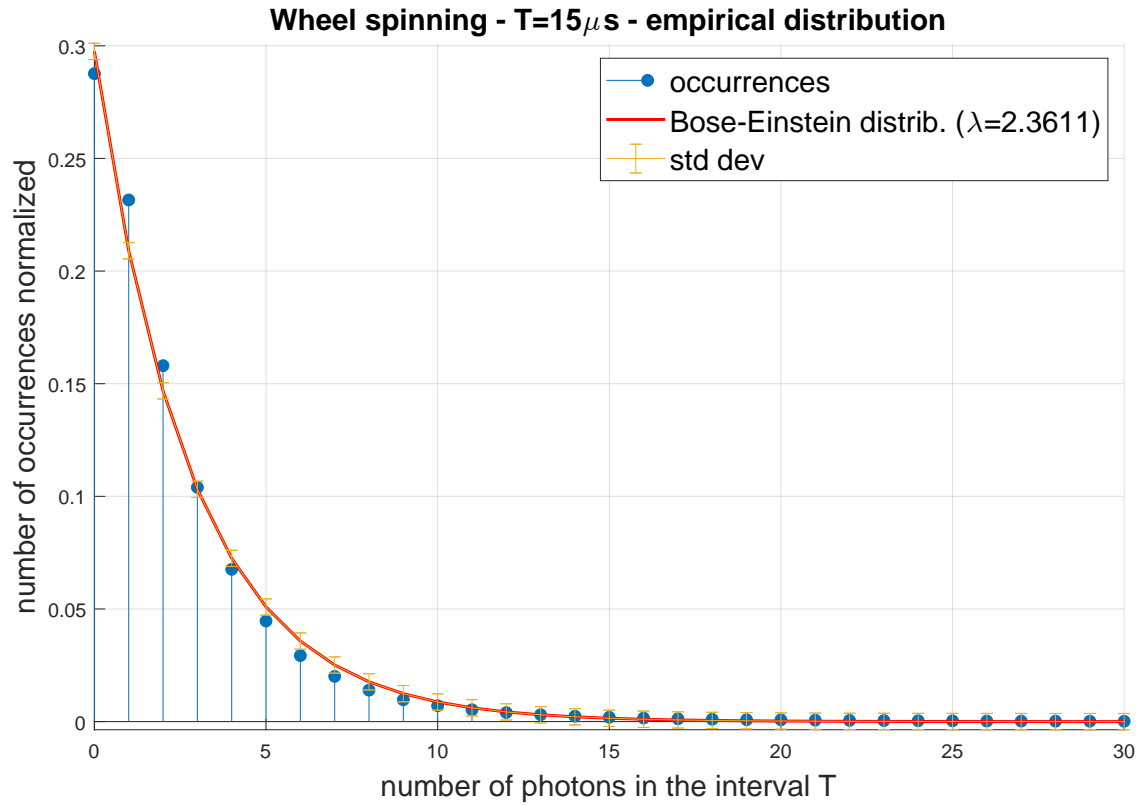


Figure 3: Empirical distribution with $T = 15\mu\text{s}$.

In both regimes a plot of the relative distribution is shown for comparison.

By changing the size of the time bin, for example to $T = 30\mu\text{s}$, it is possible to see how the values spread out and the theoretical distributions tend to drift away from the empirical data (by increasing the time bin it is increasing the mean and also the variance). The results are shown on figures 4 and 5.

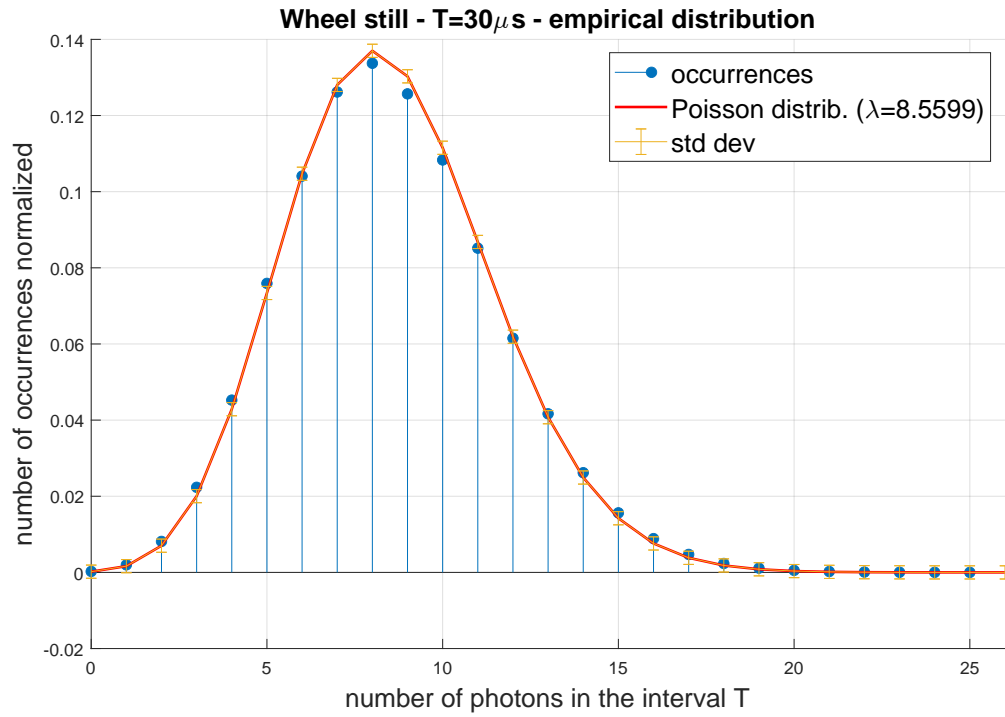


Figure 4: Empirical distribution with $T = 30\mu\text{s}$.

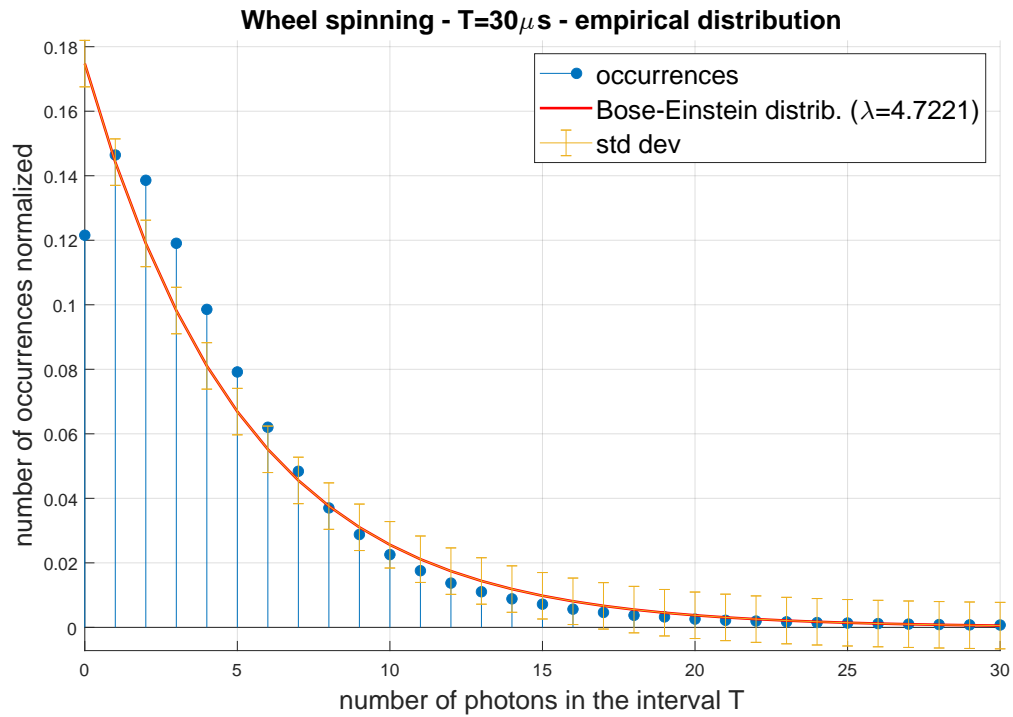


Figure 5: Empirical distribution with $T = 30\mu\text{s}$.

4 Momenta Analysis

A statistical comparison between the theoretical and empirical model can be performed by analyzing superior orders of the distributions momenta.

4.1 Poisson distribution

The first four momenta are described as:

$$\begin{aligned}m_{1,P} &= \sum_n \frac{n \cdot e^{-\lambda} \cdot \lambda^n}{n!} \Rightarrow \lambda \\m_{2,P} &= \sum_n \frac{(n-\lambda)^2 \cdot e^{-\lambda} \cdot \lambda^n}{n!} \Rightarrow \lambda \\m_{3,P} &= \sum_n \frac{(n-\lambda)^3 \cdot e^{-\lambda} \cdot \lambda^n}{n!} \Rightarrow \lambda \\m_{4,P} &= \sum_n \frac{(n-\lambda)^4 \cdot e^{-\lambda} \cdot \lambda^n}{n!} \Rightarrow \lambda + 3\lambda^2\end{aligned}$$

where λ is the mean of the Poisson distribution and $n = 0, 1, 2, \dots$. The comparison can be done by subtracting the terms on the left with the terms on the right. The results are:

$$\begin{aligned}|m_{1,P} - \lambda| &\simeq 1.0 \cdot 10^{-5} \\|m_{2,P} - \lambda| &\simeq 1.4 \cdot 10^{-4} \\|m_{3,P} - \lambda| &\simeq 2.6 \cdot 10^{-3} \\|m_{4,P} - (\lambda + 3\lambda^2)| &\simeq 4.5 \cdot 10^{-2}\end{aligned}$$

Values can be affected by the choice of n : the more n increases, the more the differences become close to zero (for the calculations was set to its maximum value).

4.2 Bose-Einstein distribution

The first four momenta are described as:

$$\begin{aligned}m_{1,BE} &= \sum_n \frac{n}{1+\lambda} \cdot \left(\frac{\lambda}{1+\lambda}\right)^n \Rightarrow \lambda \\m_{2,BE} &= \sum_n \frac{(n-\lambda)^2}{1+\lambda} \cdot \left(\frac{\lambda}{1+\lambda}\right)^n \Rightarrow \lambda + \lambda^2 \\m_{3,BE} &= \sum_n \frac{(n-\lambda)^3}{1+\lambda} \cdot \left(\frac{\lambda}{1+\lambda}\right)^n \Rightarrow \lambda + 3\lambda^2 + 2\lambda^3 \\m_{4,BE} &= \sum_n \frac{(n-\lambda)^4}{1+\lambda} \cdot \left(\frac{\lambda}{1+\lambda}\right)^n \Rightarrow \lambda + 10\lambda^2 + 18\lambda^3 + 9\lambda^4\end{aligned}$$

where λ is the mean of the Bose-Einstein distribution and $n = 0, 1, 2, \dots$. The comparison can be done by subtracting the terms on the left with the terms on the right. The results are:

$$\begin{aligned}|m_{1,BE} - \lambda| &\simeq 1.3 \cdot 10^{-9} \\|m_{2,BE} - (\lambda + \lambda^2)| &\simeq 9.0 \cdot 10^{-8} \\|m_{3,BE} - (\lambda + 3\lambda^2 + 2\lambda^3)| &\simeq 6.3 \cdot 10^{-6} \\|m_{4,BE} - (\lambda + 10\lambda^2 + 18\lambda^3 + 9\lambda^4)| &\simeq 4.5 \cdot 10^{-4}\end{aligned}$$

Values can be affected by the choice of n : the more n increases, the more the differences become close to zero (for the calculations was set to its maximum value).

5 Conclusions

The plots and the momenta analysis display a strong similarity between the data collected and the theoretical models. The Arecchi experiment exhibits the difference between a coherent and a thermal source: the statistical description of a coherent source follows a Poisson distribution of mean λ , while a thermal source follows a Bose-Einstein distribution of parameter λ . This can be verified empirically because by reducing the time bin the less photons are collected and the Bose-Einstein distribution follows this kind of behaviour (strong peak around 0), while the Poisson distribution tends to keep a bell-shape around the mean (like a coherent light source) and by changing T all it does is shifting and spreading the “bell”. Note that the differences between the empirical and theoretical momenta are really small: the twos can be considered approximately equal despite some error due to the small set of n .

6 MATLAB Code

```
1 %% STILL
2
3 clear all;
4 close all;
5 clc;
6
7
8 files = dir('Wheel_still_2020\*.txt');
9
10 datasets=[];
11 for i=1:length(files)
12
13     data=importdata(strcat(files(i).folder, '\', files(i).name));
14     if i>2 && length(data(:,1))<length(datasets(:,i-1))
15         datasets(length(data(:,1))+1:length(datasets(:,i-1)),:)=[];
16     end
17     datasets(:,i)=data(:,1);
18
19     clear data;
20 end
21 clear files i;
22
23 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
24 T=15; %us
25 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
26
27
28 tags=datasets.*81e-12*1e6; %tags in us (microseconds)
29
30 clicks=tags(:,1);
31 for i=2:10
32     clicks=[clicks; tags(2:length(tags(:,i)),i)+clicks(length(clicks))];
33 end
34 tags=clicks; clear clicks i;
35
36
37
38 m=discretize(tags,0:T:max(tags));
39 figure('Renderer','painters','Position',[500 300 900 600]);
40 hold on; grid on;
41
42 [GC,GR] = groupcounts(m);
43 GC(length(GC)+1:max(GR))=0; %zero padding
44
45 [vals_y,-]=histcounts(GC, 'BinMethod','Integers');
46
47 vals_x=0:length(vals_y)-1;
48
49 N=sum(vals_y);
50
51 vals_x=vals_x';
52 vals_y=vals_y';
53
54 stem(vals_x,vals_y/N, 'filled')
55
56 lam=length(tags)/tags(length(tags))*T;
57
58
59 n=0:length(vals_x); n=n';
60 poisson=exp(-lam).*(lam.^n)./factorial(n);
61
62 plot(n,poisson, 'r', 'LineWidth', 1.5)
63
```

```

64 xlim([0 length(vals_x)])
65 title(strcat('Wheel still - T=', string(T), ...
66     '\mus - empirical distribution'), 'FontSize', 15)
67 ylabel('number of occurrences normalized', 'FontSize', 15);
68 xlabel('number of photons in the interval T', 'FontSize', 15);
69
70 pbaspect([1.5 1 1])
71 clear i j initial final datasets;
72
73
74 %%%%%%%%%% MOMENTA %%%%%%%%%
75
76 mu=lam;
77 m1P=sum((n.*exp(-mu).*mu.^n)./(factorial(n)));
78 diff1P=m1P-mu;
79 m2P=sum(((n-mu).^2.*exp(-mu).*mu.^n)./(factorial(n)));
80 diff2P=m2P-mu;
81 m3P=sum(((n-mu).^3.*exp(-mu).*mu.^n)./(factorial(n)));
82 diff3P=m3P-mu;
83 m4P=sum(((n-mu).^4.*exp(-mu).*mu.^n)./(factorial(n)));
84 diff4P=m4P-(mu+3*mu^2);
85
86 errorbar(n,poisson,...
87     std(vals_y/N-poisson(1:length(vals_y)))*ones(size(poisson)))
88
89 legend('occurrences',strcat('Poisson distrib. (\lambda=', string(lam), ...
90     ')'), 'std dev', 'FontSize', 15);
91
92 %% SPINNING
93
94 clearvars -except T;
95 clc;
96
97
98 files = dir('Wheel_spinning_2020\*.txt');
99
100 datasets=[];
101 for i=1:length(files)
102
103     data=importdata(strcat(files(i).folder, '\', files(i).name));
104     if i>2 && length(data(:,1))<length(datasets(:,i-1))
105         datasets(length(data(:,1))+1:length(datasets(:,i-1)),:)=[];
106     end
107     datasets(:,i)=data(:,1);
108
109     clear data;
110 end
111 clear files i;
112
113 tags=datasets.*81e-12*1e6; %tags in us (microseconds)
114
115 clicks=tags(:,1);
116 for i=2:10
117     clicks=[clicks; tags(2:length(tags(:,i)),i)+clicks(length(clicks))];
118 end
119 tags=clicks; clear clicks i;
120
121 m=discretize(tags,0:T:max(tags));
122 figure('Renderer', 'painters', 'Position', [500 300 900 600]);
123 hold on; grid on;
124
125 [GC,GR] = groupcounts(m);
126 GC(length(GC)+1:max(GR))=0; %zero padding
127
128
129 [vals_y,-]=histcounts(GC, 'BinMethod', 'Integers');
130
131 N=sum(vals_y);

```

```

132
133     vals_x=0:length(vals_y)-1;
134
135     vals_x=vals_x';
136     vals_y=vals_y';
137
138     stem(vals_x,vals_y/N, 'filled')
139
140     lam=length(tags)/tags(length(tags))*T;
141
142     n=0:length(vals_x); n=n';
143     BE=(1/(lam+1)).*(lam/(lam+1)).^n;
144
145     plot(n,BE, 'r', 'LineWidth', 1.5)
146
147     title(strcat('Wheel spinning - T=', string(T), ...
148         '\mus - empirical distribution'), 'FontSize', 15)
149     ylabel('number of occurrences normalized', 'FontSize', 15);
150     xlabel('number of photons in the interval T', 'FontSize', 15);
151     xlim([0 30]); ylim([-inf inf]);
152
153     pbaspect([1.5 1 1])
154     clear i j initial final datasets h;
155
156     %%%%%%%%%% MOMENTA %%%%%%%%%%
157
158     mu=lam;
159     m1BE=sum(n.*1./(1+mu).*(mu./(1+mu)).^n);
160     diff1BE=m1BE-mu;
161     m2BE=sum((n-mu).^2.*1./(1+mu).*(mu./(1+mu)).^n);
162     diff2BE=m2BE-(mu+mu^2);
163     m3BE=sum((n-mu).^3.*1./(1+mu).*(mu./(1+mu)).^n);
164     diff3BE=m3BE-(mu+3*mu^2+2*mu^3);
165     m4BE=sum((n-mu).^4.*1./(1+mu).*(mu./(1+mu)).^n);
166     diff4BE=m4BE-(mu + 10*mu^2 + 18*mu^3 + 9*mu^4);
167
168     errorbar(n,BE,std(vals_y/N-BE(1:length(vals_y)))*ones(size(BE)))
169
170     legend('occurrences',strcat('Bose-Einstein distrib. (\lambda=', ...
171         string(lam), ')'), 'std dev', 'FontSize', 15)

```