Cross Section Study of Beam Gas Curtain

a project report for

CERN Summer Student Programme 2025

by

Ewout Voorrips

Supervisor: Daniele Butti

Beam Instrumentation Group (SY-BI)

Abstract

During this Summer Student project, postprocessing was done on images from the novel Beam Gas Curtain (BGC) instrument, a new instrument to be used in the High-Luminosity LHC for beam profile measurements. The aim of the study was to use the photon counting method to study the cross section of the beam. This method proved useful in determining beam dimensions, and the evolution of background and ROI photons was determined for multiple fills. A new analysis framework, including a new photon counting algorithm was made to improve future analysis.

Geneva, Switzerland

August 22, 2025

Contents

1 Introduction

This report is written as a part of the CERN Summer Student Programme 2025. As a Master's student in Engineering Physics at the Technical University of Denmark I had the opportunity to work in the Beam Instrumentation group in the Accelerator Systems Department.
As part of the upcoming upgrades for the HL-LHC, the BI group is developing a new Beam Profile Monitor. This instrument uses the fluorescence coming from the interaction between the beam and a thin curtain of gas to generate a 2D image of the beam profile.
The Beam Gas Curtain (BGC) has been in development for over 10 years, and has been installed for Run 3 of the LHC. It is intended to be fully operational after LS3, at which point it should provide more precise measurements of the beam profile.
Currently, the BGC is being tested extensively. My project revolved around studying the cross section using tests done during various fills of the LHC.

2 Theory

The Beam Gas Curtain (BGC) is based on the interaction of a highly energetic beam of the LHC with a supersonic curtain of gas. This flat curtain of gas is produced by an ensemble of skimmers, nozzles and chambers and has been developed in collaboration with the Cockcroft Institute and GSI for the past decade. Since my project did not involve this setup, I will not delve deeper into the design of the gas curtain source. Those interested are encouraged to read [**?**].

The gas curtain is angled at 45° relative to the beam. When passing the beam, the gas atoms interact with the high energy protons (or heavy ions) in the beam. The gas atoms are excited into a higher state and through their subsequent relaxation emit a photon, which passes through an optical system and onto an image intensifier. This amplifies the light signal, which is ultimately detected by a camera.
The interaction between the gas and the beam at these energies is not well understood. Most theories are based on interpolations of theories applicable at lower energies. Solid evidence for different behaviour could indicate different processes taking place at ultra-relativistic velocities
The postprocessing of the images taken by the BGC was the main concern of my project. Since the intensity of the light detected was low enough, it enabled us to measure single photons. My project revolved around detecting these photons and using these to do measurements on the cross section of the beam.

3 Methodology

The first week of the project was primarily about me familiarizing myself with the theory, as well as with the datatypes I would be working with. The outputs of the BGC I would be working with were .npy arrays, which corresponded to brightness levels of pixels as recorded by the camera. The photons coming in appear as bright spots on a dark background with random noise. Multiple images stacked together will then give a depiction of the outline of the beam in the transverse direction.

3.1 Photon Aggregation

The first step was to determine the best method for integrating multiple images to create an image of the beam profile. The single images contained O(100) single photons, meaning that you have to integrate multiple images before you start to see the outline of the beam. There are two main methods by which this is done. The first is image stacking, in which you simply add different images onto another. The other, and the one I ended up using, was photon histogramming. In this method, you create a histogram and count the individual photons and plot their positions.

After some experimenting with different parameters and methods, I decided that the photon counting algorithm was the way forward, where I would preprocess the image by applying a median filter twice and a minimum threshold once. This process can be seen in the first two images of figure 1.

3.2 Photon Counting Algorithm

After preprocessing an image, a photon counting algorithm is applied to mark the individual photons. Previously, this was done using an astropy package. Due to good experience with the LoG (Laplacian of Gaussian) algorithm in previous projects, I tried this method. Compared to the previous method it proved better at detecting edge photons, however, there still remained the problem of clustering, meaning that photons close to each other could not be distinguished.

3.3 Data Gathering

Over the course of two days, several tests were done with the BGC. In total, four fills were analysed (10847,10849,10850 and 10852) during these days. The last two were long fills (+10 hours) in which it was possible to study the long-term behaviour as the intensity of the beam decreased due to collisions and losses. More test fills were done during the first week of August. These included fills 10905 and 10907. Finally, I also examined fills that were recorded in 2024 by a PhD student from University of Liverpool, namely fills 9969, 9978 and 10004. For every fill the individual images are saved to a sperate folder.

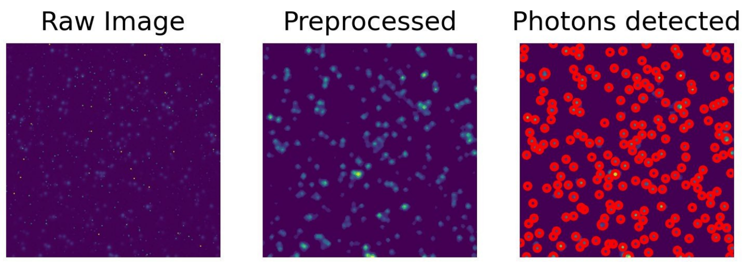


Figure 1: Image Processing Method. Two median filters and a threshold are applied to the raw image, which is shown in the middle. The photon counting algorithm is then applied to determine the number and coordinates of the .

3.4 Data Processing

After working in several Swan Notebooks for the first few weeks I created an easier-to-implement system, which also aimed to improve processing speed. I created two python files which could be run from a virtual machine, one which would handle the analysis of the images and return and save arrays of the coordinates of the detected photons, and another which would retrieve these arrays and do the analysis. After finding out that the calculation of one fill (almost 500,000 images!) would take approximately 72 hours, I decided to use multi-threading, to speed up the process by almost 10x.

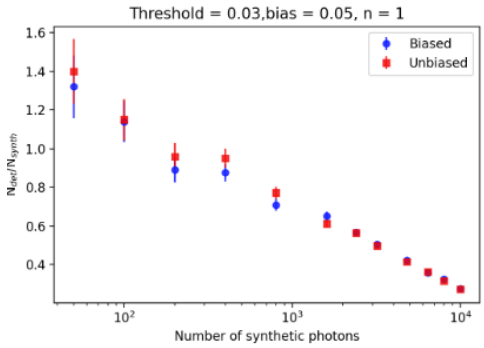


Figure 2: Stress test for a biased photon placement. Photons are more likely to be detected in the center (at the beam), so the photons are more likely to cluster.

4 Results

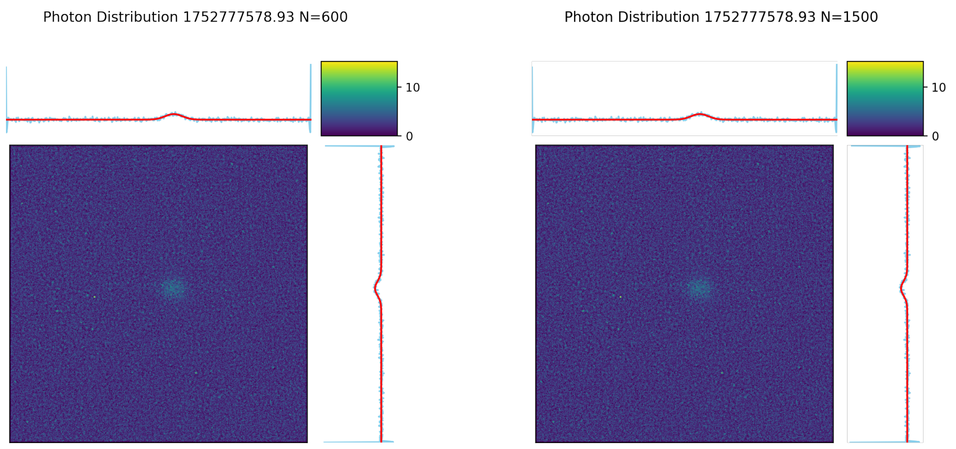
4.1 Algorithm stress-test

To attempt to quantify the photon clustering problem I performed a stress-test on the algorithm. By depositing 'synthetic' photons (i.e. dots) on a blank background and adding random noise, I recreated an image, but with a controllable amount of photons. This way, I could see how many of the synthetic photons the algorithm was able to detect.

I tested this for 12 different photon counts between 50 and 10,000 photons. I distributed the synthetic photons in two ways: the first was a uniform distribution and the second one had a bias for a photon to be close to the center. This replicates the increased likelihood for an actual photon to be in the region of interest, since that is where the beam is. This resulted in a slight, but noticable difference in detected photons, as can be seen in figure 2 . Overall, we can see an unphysical ratio higher than unity for low photon counts, indicating that the algorithm detects background noise as photons. The ratio then decreases gradually. For higher photon numbers, less than 50% of the synthetic photons are detected, most likely due to clustering. From this, it becomes clear that the algorithm performs 'best' in images with in the order of a couple hundred photons. Note that this only applies to this image size, smaller images need fewer photons.

4.2 Vertical and Horizontal Distribution

In order to determine characteristics of the beam, such as width or height, it is essential to determine a vertical and horizontal distribution of a photon histogram. This is done by integrating the photon counts along the x- and y-axes. This is done for an integration time of 1 minute in figure **??**. The red line is a Gaussian fitted to the distribution. This is used to estimate the width and height of the beam. Note the high peaks near the edges of the distributions. This is most likely due to edge photons being over-counted. These edge photons can be slightly out of the focus area, but still affect the pixels on the edge, thus contributing to the total photon count. For this reason, I do not take the edges into account when fitting the Gaussian. To assess how good the fit was, and therefore how good of an estimator



(a) T_{int} = 1 min

(b) T_{int} = 2,5 min

Figure 3: Two photon histograms and their corresponding distributions.

of the beam width/height this can be I took a closer closer look at the vertical distribution for N_{int}=600. This is shown in figure 4. The normalized value of χ^2 is ≈ 1.2, which indicates that, although not a perfect fit, a Gaussian is a good approximation of the distribution.

4.3 Long-fill Behaviour

In order to calculate the cross section from the counted photons, it is essential that a consistent intensity is obtained. For this purpose, a first step to take is to

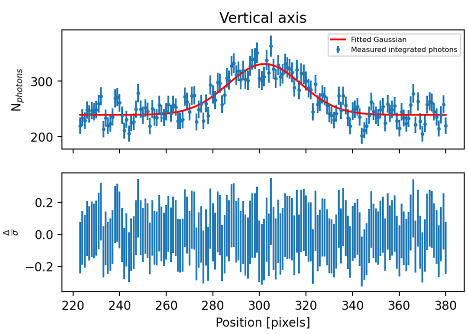


Figure 4: Fitted vertical distribution of photon histogram. Bottom plot shows the deviation of the datapoint to the fitted Gaussian.

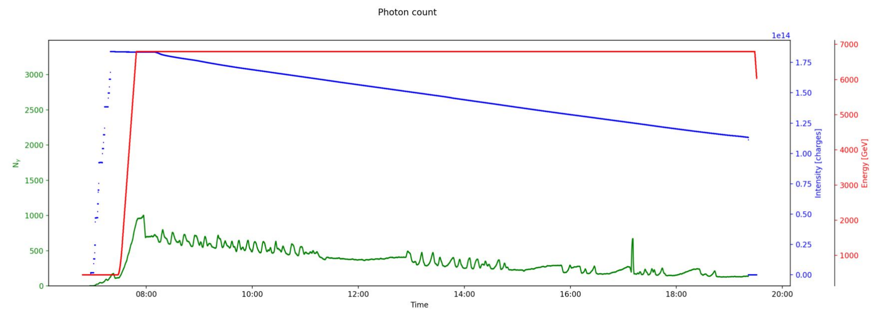


Figure 5: Photon count over the entire fill. Averaged over 1000 images to remove noise.

look at the photon count over the entire fill. Figure 5 shows the photon count during fill 10852. We observe a rise during injection/ramp and a steady fall as the intensity falls during stable beams. The periodic fluctuations are most likely caused by losses in the beam pipe close to the instrument.

4.4 Background vs ROI

The spatial distribution of the counted photons is of interest since it gives the ability to discern background photons versus signal photons in the ROI. In order to get an estimate of the background photon level during a fill, I defined 8 background 'boxes' and counted for every frame how many photons were detected in that box. The boxes as I designed them are shown in figure 6.

The region of interest can be determined dynamically or statically. For dynamically, every so often the ROI is determined by integrating multiple frames and afterwards photons are counted within this region. The frequency with which the ROI is refreshed depends on the fill phase. During the injection and ramp, this is done more frequently to determine the behavior of the beam size during these dynamical phases. During flat-top, the beam size is expected to not change much, so the ROI is determined less often. During static analysis, the ROI is calculated during injection, when the beam is larger. Subsequently, the number of photons in this ROI are counted for the rest of the fill.

I have tried both methods, but found that due to the small size of the beam during flattop and the fact that we are dealing with individual photons, the ROI photon count vs the background photon count for dynamic counting was disturbed by noise to a high degree. This makes the signal more unreliable. When using static counting, a larger area is taken into account, thus increasing the chance that all signal photons are counted. The excess of this area compared to the background is then assumed to be signal. This results in plots 7 and 8. A few interesting observations can be made based on this. The first is that during flattop, the signal photon rate slightly increases with decreases intensity and looks flat at the end. This seems nonphysical, since you would expect that the interaction will decrease with de-

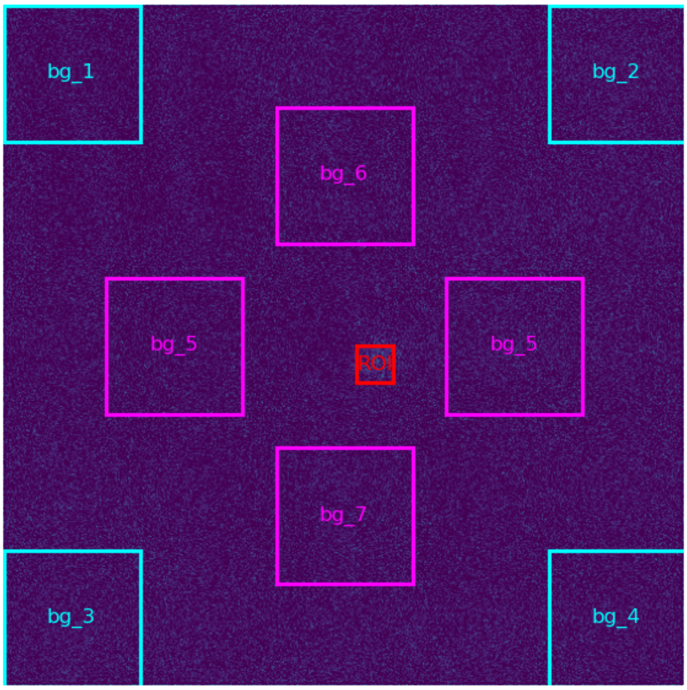


Figure 6: Diagram of relative 'counting' boxes to determine background photon levels. The background boxes each have a width and height equal to 20% of the total image dimensions.

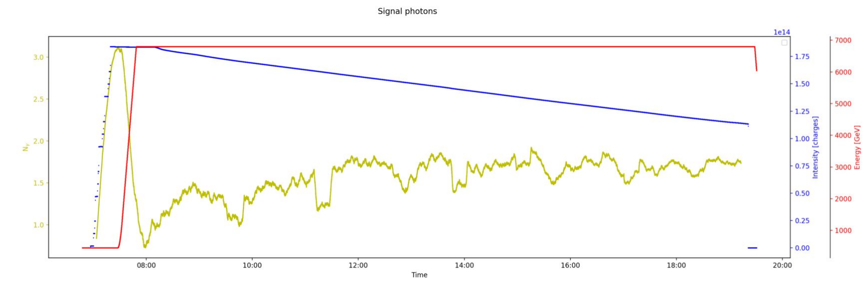


Figure 7: Signal photons

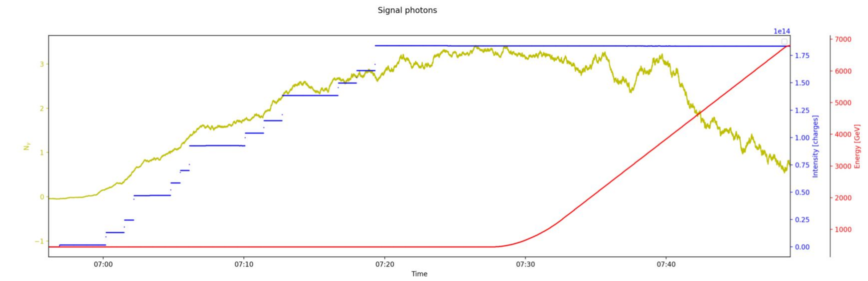


Figure 8: Signal photons during injection/ramp

creasing intensity. I have tried to exclude factors that could cause this shape, but I have not been successful. If it is physical, it means that the interaction between the gas and the beam is not well understood.

Another observation can be made during injection/ramp. A clear increase in the signal photons can be seen during injection, and when the energy ramps up, the signal photons decreases again.

These results are contrary to what we would expect, so more data is needed to confirm this.

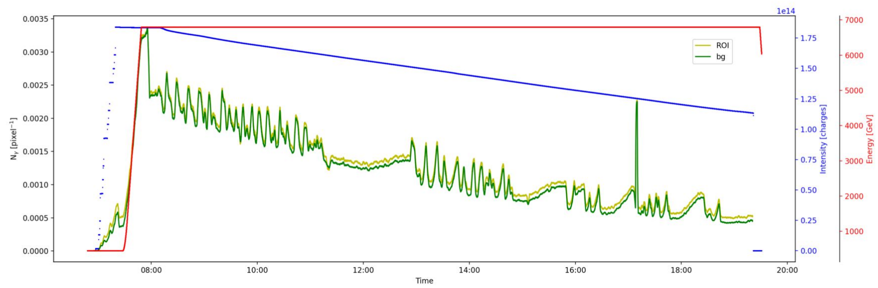


Figure 9: ROI and Background photon rates during fill 10852

5 Conclusion

In this study I attempted to assess the viability of the photon counting method in determining the cross section using the Beam Gas Curtain instrument. I used data from several test fills of the LHC to do this. I created a new analysis framework to analyse these fills. Using this framework, I calculated the evolution of beam dimensions over a fill. There are indications that the amount of signal photons during ramp-up decrease, before staying relatively stable during flattop. Nonetheless, further evidence is required before this can be stated with certainty. If this is the case, it could indicate the beam-gas interaction does not occur as previously thought. I recommend that for future studies, long, clean fills are used to determine behaviour across more fills. An interesting aspect to look at is altering threshold values, to see if signal-to-noise can be improved.

On a personal level, this project was challenging but fun. It was very interesting to work with such new technologies. My supervisor and colleagues were very nice and helpful.