Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Ion trap image analysis

Semester Project

Albert Mitjans-Coma

June 10, 2021

Advisors: Dr. Ilia Sergachev, Dr. Matthew Grau

Trapped Ion Quantum Information Group, ETH Zürich

Abstract

In this report we use Python to analyse images taken from a camera installed on an ion trap to gather relevant information for the completely unattended operation of the trap. Given noisy black and white images of trapped ions we are able to extract features such as number of bright, dim and dark ions, intensity, crystal linearity, maximum spot size and roundness. These features will be used as feedback to control and stabilize the state of the trap. Furthermore, we present a web application where we display in real time the detected features together with the input image. Finally, we use InfluxDB and Grafana to store and visualize a time series database of the detected features, which helps us identify events in the past such as the increase of the reorder rate due to an ion falling into a dark state or a laser frequency fluctuation.

Contents

Chapter 1

Introduction

In recent years, ion traps have evolved from a basic research topic into a versatile tool for a wide range of research subjects and quantum technologies. Ions may now be studied in unprecedented detail thanks to the capacity to store, cool, and control them. This capability has resulted in significant scientific progress in fundamental research, such as the monitoring of cold collisions between trapped ions and cold atomic or molecular particles, the study of the interaction of light with trapped ions, or mass measurements with ultra-high precision. Beyond purely fundamental research, ion traps have become essential for a wide range of applications and technologies.

Many vital building elements for a quantum computer have been established in recent years, and trapped ions are now one of the most promising implementations. Furthermore, some of today's best atomic clocks are based on atomic ions. Many elements of our daily lives could be revolutionized by these applications.

One of the key elements of an ion trap is laser cooling. To have atomic particles under precise control we first need to slow them down, and this can be accomplished using lasers. Laser cooling comprises a series of techniques in which particles are cooled down to almost absolute zero. These techniques rely on the fact that when a particle (usually an atom) absorbs and re-emits a photon its momentum changes. The most common technique is Doppler cooling, which we describe thoroughly in Appendix A.

In addition to research applications, ion traps can also be used in the area of science communication, increasing the sense of wonder about these new scientific discoveries. More specifically, in partnership with the Swiss Science Center Technorama, the Trapped Ion Quantum Information (TIQI) group is working on a trapped ion setup which will allow members of the general public to see atoms with their eyes. The trap uses barium ions, with a detection wavelength nearly at the peak of the spectral sensitivity of the human eye (493 nm), confined in a linear Paul trap with blade-shaped electrodes that has been optimized for optical access and photon collection efficiency.

This setup can provide experiences ranging from allowing people to see single atoms directly with their eyes, manipulating the position and spacing of trapped ions by controlling electric fields, to observing the truly quantum random decay from a dark metastable excited state to the fluorescing ground state.

The goal of this project is to help fully automatize the ion trap to make it work completely unguarded. In this work we make use of real-time trapped ion images to gather continuous information of the state of the trap, which will be used as a feedback mechanism for stabilization and automatization of the trap. Making use of computer vision techniques we are able to extract information of the ions such as coordinates, state of the trap (crystallized ions, empty, ion cloud), number of bright, dim and dark ions, crystal linearity, maximum intensity, reorder rate, maximum spot size and roundness.

We organize this project as follows. Chapter 2 presents the theoretical framework needed to understand some of the concepts that are approached in this project. The motivation for this project is presented in Chapter 3. Chapter 4 describes the approach we take to obtain the aforementioned image features. Chapter 5 details the methods used to display these image features. Our code's performance and some interesting results obtained are discussed in Chapter 6. We briefly present the structure of our project's files in Chapter 7 and finally narrate our conclusions in Chapter 8. Additionally, we perform a mathematical description of Doppler cooling in Appendix A.

Chapter 2

Theoretical framework

2.1 The barium ion

Compared to other ions, barium ions have a lot of advantages. First and foremost, they have a single valence electron, which simplifies their atomic structure up to allow for laser cooling. Atomic species containing many valence electrons can have extremely complicated energy level structures, many of which it can spontaneously decay into. Because every long-lived ion state must be actively repumped, cooling such species can require a large number of lasers.

More crucially, we can work with a set of Ba^+ transitions that lie in the visible spectrum, which makes laser alignment much easier, and because none of the transitions are in the UV, all of these frequencies can be transmitted with minimal loss using optical fibers. Furthermore, barium ions have a detection wavelength that is approximately at the apex of the human eye's spectral sensitivity (493 nm), making them easier to see.

Ba^+ has an energy level structure that comprises two low dying, long-lived D states (see Fig 2.1). The Doppler cooling (see Appendix A) consists of a blue 493 nm transition from the ground state to the $P_{1/2}$ state, with a branching ratio of 0.244 to $D_{3/2}$. Because this state has a long lifetime, a repump laser at 650 nm is required for continuous cooling.

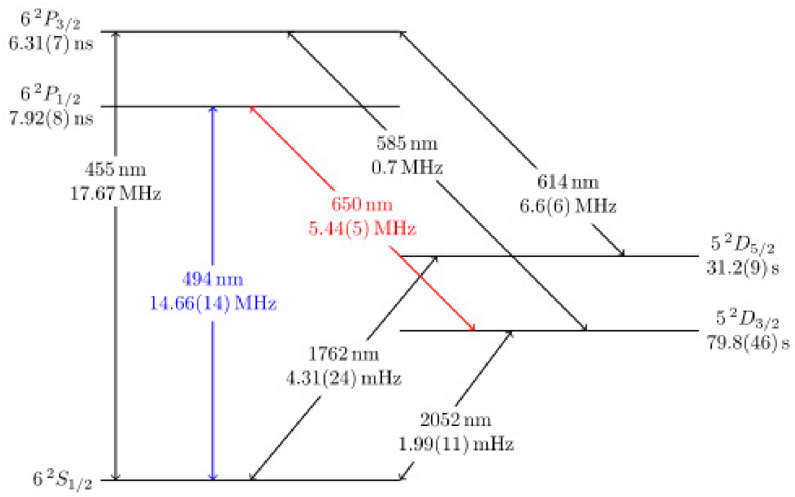

Figure 2.1: Energy level diagram for Ba^+ with all its transitions [1].

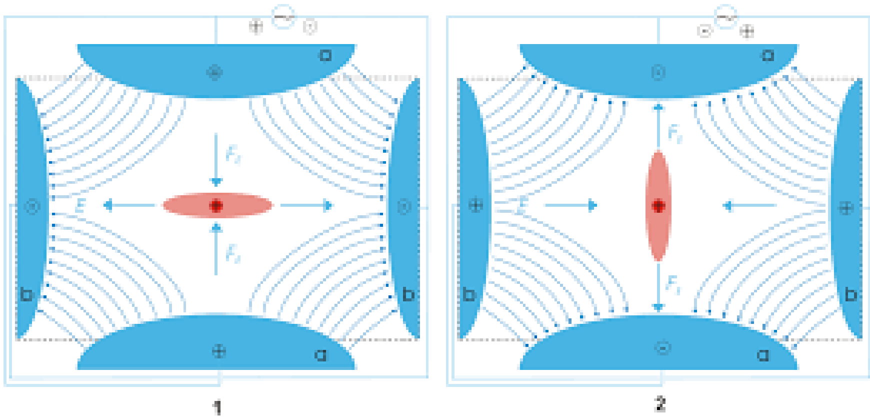

Figure 2.2: Scheme of a Paul trap with a particle of positive charge surrounded by a cloud of similarly charged particles [2].

2.2 Ion trapping

2.2.1 Paul traps

Unfortunately it is impossible to suspend a charged particle in free space using electrostatic forces alone (restriction known as Earnshaw's theorem). If that were the case, the electrostatic field would look like a harmonic well in all 3 directions. However, since any electrostatic potential field must obey Laplace's equation

$$\nabla^2\psi = \nabla^2\left(\frac{\alpha x^2}{2} + \frac{\beta y^2}{2} + \frac{\gamma z^2}{2}\right) = \alpha + \beta + \gamma = 0 \tag{2.1}$$

there exists no solution where all three spring constants (α, β, γ) are positive. The best we can manage is a saddlepoint.

In the 1950s the physicist Wolfgang Paul came up with a method to trap charged particles by using dynamic electric fields, for which he shared the Nobel Prize in Physics later in 1989. The idea was that, by creating a quadrupole electric field and rapidly rotating it (Fig 2.2), one can create an electric potential that behaves, to lowest approximation, as if it were an electrostatic harmonic potential. A trap that suspends a charged particle by application of this principle is called a Paul trap. This trap can also be named radio frequency (RF) trap, because the switching rate of confining and anti-confining directions of the trap is often at radio frequency.

2.2.2 Ion string configuration

When two or more ions are confined in a Paul trap, they will repel each other due to the Coulomb force. Consequently, such confined charged plasmas will have very low densities. When their thermal energy becomes sufficiently small compared with their Coulomb interaction energy, the plasma will condense into a crystalline state [3]. Given the highly anisotropic trapping potential, this crystalline state is, for a small enough number of ions, a simple chain of ions aligned in the axial direction. As the degree of anisotropy is decreased, or number of ions is increased, structural phase transitions can be triggered, which start at the center of the chain and extend towards the edges.

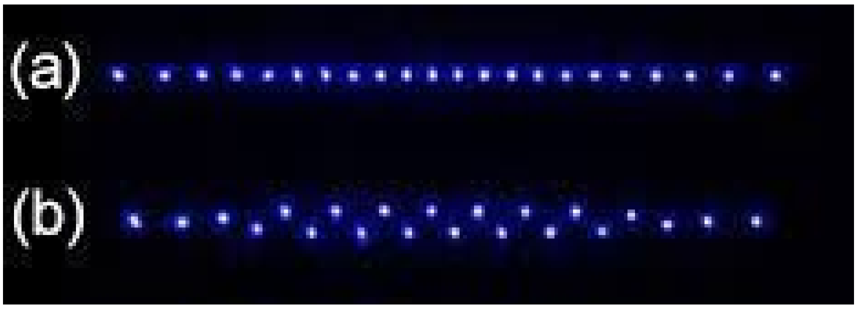

Figure 2.3: (a) An image of a string of 22 ytterbium ions along the axis of a linear RF trap. (b) The same ions but in a zigzag configuration. (Figure courtesy of Thompson, R. [4]).

In particular, a transition from the linear chain to a zigzag configuration has been well characterized (see Fig. 2.3).

The thermodynamic equilibrium state of the trapped ions is determined by the Coulomb coupling parameter [5]:

$$\Gamma = \frac{1}{4\pi\varepsilon_0} \frac{Q^2}{a_{WS} k_B T} \tag{2.2}$$

where a_{WS} is the Wigner-Seitz radius defined by $\frac{4\pi}{3} n_0 a_{WS}^3 = 1$, n_0 is the particle number density, and k_B is Boltzmann's constant. This correlation parameter can be seen as the ratio of the electrostatic energy of neighbouring charges to the thermal energy. Numerical solutions have shown that crystallization occurs for highly correlated systems, appearing at $\Gamma \approx 174$ [6].

In equation 2.2 we see how the correlation parameter is inversely proportional to the ion's thermal energy. Consequently, an increase of temperature might induce the ions to abandon the crystalline state. A temperature rise might be caused by a bad calibration of laser frequencies, or an excessive amount of ions in the trap. On the one hand, having wrong laser frequencies will affect the performance of Doppler cooling. On the other hand, a large number of trapped ions will most probably have a zigzag configuration (or more complex arrangements). Ions outside the trap axis will have excess micromotion and will increase kinetic energy. Moreover, laser intensity gradient makes cooling of farther ions less efficient, thus the longer the ion string the higher temperature is. Consequently, these two aforementioned events will cause the apparition of an ion cloud (see Fig. 2.4).

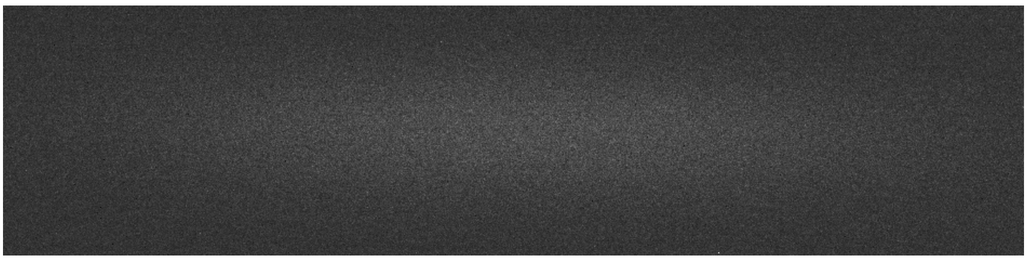

Figure 2.4: Image with an ion cloud.

2.2.3 Equilibrium positions of ions in a linear trap

The following derivation is based on [7].

Let us consider a chain of N ions in a trap. The position of the m^{th} ion, where the ions are numbered from left to right, will be denoted $x_m(t)$. The motion of each ion will be influenced by an overall harmonic potential due to the trap electrodes and by the Coulomb force exerted by all of the other ions. We will assume that the ions are weakly bound in the x direction, but strongly bound in the y and z directions. Consequently, motion along the y and z axes can be neglected. Hence the potential energy of the ion chain is given by the following expression:

$$V = \sum_{m=1}^{N} \frac{1}{2} M\nu^2 x_m(t)^2 + \sum_{\substack{n,m=1 \\ m \neq n}}^{N} \frac{Z^2 e^2}{8\pi\varepsilon_0} \frac{1}{|x_n(t) - x_m(t)|} \tag{2.3}$$

where M is the mass of each ion, Z is the degree of ionization of the ions, e is the electron charge, ε_0 is the permitivity of free space, and ν is the trap frequency, which characterizes the strength of the trapping potential in the axial direction.

Given the cold temperature of the trapped ions (usually a few mK), the position of the m^{th} ion can be approximated by the formula

$$x_m(t) \approx x_m^{(0)} + q_m(t) \tag{2.4}$$

where $x_m^{(0)}$ is the equilibrium position of the ion, and $q_m(t)$ is a small displacement. The equilibrium positions will be determined by the following equation:

$$\left[\frac{\partial V}{\partial x_m} \right]_{x_m = x_m^{(0)}} = 0 \tag{2.5}$$

If we define the length scale l by the formula

$$l^3 = \frac{Z^2 e^2}{4\pi\varepsilon_0 M\nu^2} \tag{2.6}$$

and the dimensionless equilibrium position $u_m = x_m^{(0)}/l$, then Eq. 2.5 may be rewritten as the following set of N coupled algebraic equations for the values of u_m:

$$u_m - \sum_{n=1}^{m-1} \frac{1}{(u_m - u_n)^2} + \sum_{n=m+1}^{N} \frac{1}{(u_m - u_n)^2} = 0 \tag{2.7}$$

These equations can be solved numerically. The numerical values of the solutions to these equations for 2 to 10 ions are given in Table 2.1.

N	Scaled equilibrium positions
2	-0.62996 0.62996
3	-1.0772 0 1.0772
4	-1.4368 -0.45438 0.45438 1.4368
5	-1.7429 -0.8221 0 0.8221 1.7429
6	-2.0123 -1.1361 -0.36992 0.36992 1.1361 2.0123
7	-2.2545 -1.4129 -0.68694 0 0.68694 1.4129 2.2545
8	-2.4758 -1.6621 -0.96701 -0.31802 0.31802 0.96701 1.6621 2.4758
9	-2.6803 -1.8897 -1.2195 -0.59958 0 0.59958 1.2195 1.8897 2.6803
10	-2.8708 -2.10003 -1.4504 -0.85378 -0.2821 0.2821 0.85378 1.4504 2.10003 2.8708

Table 2.1: Scaled equilibrium positions of the trapped ions for different total number of ions.

Chapter 3

Motivation

As mentioned in the Introduction, the goal of this project is to make use of the trapped ion images to fully automate the ion trap. Our aim is to create a tool that will ease some tasks related to the control of ion traps, such as loading of ions, laser frequency tuning and micromotion compensation. Moreover, we intend to continuously monitor the state of the trap to detect any anomaly or unwanted circumstance, such as an ion cloud.

In addition, we aspire to build an image analysis tool that does not depend on the trap or ion type. Consequently, this software could be easily adapted to any trapped ion setup with a camera.

All the measured features mentioned in Chapter 4 (some of which are imperceptible to the human eye) can also be used in quantum information experiments for calibration and debugging.

Chapter 4

Backend - Image Analysis

The camera outputs 16-bit black and white images with shape (960, 1280). These images are saved in shared memory, by another program running camera image acquisition, and later fetched by our image processing code. Given that the camera sensor is natively 12-bit, we throw away the lower 4 bits of the image pixels (they are all empty). The resulting 12-bit image (Fig. 4.1) is used for gathering all the following information.

4.1 Ion coordinates

4.1.1 Bright ions

Even though detecting visible trapped ions might seem an easy task, we must bear in mind all possible complications, such as noise, dead pixels and background light.

To avoid such problems, we first define a region of interest, which consists of a smaller region where we will always find the trapped ions. Knowing we have a linear trap oriented along the horizontal axis of the image, we can set the region of interest around

Figure 4.1: Image used for analysis. The region of interest is plotted in red.

this axis, as shown in Fig. 4.1. The image is thus cropped into this smaller region, removing unwanted pixels that may affect the detection, and also speeding up any image processing implemented later (we have less pixels to process).

In order to remove the noise and preserve the spatial information of the trapped ions, we implement a spatial band-pass filter that attenuates frequencies in the original gray-scale image that are far from the band center. In imaging science this is accomplished using a difference of gaussians (DoG) feature enhancement algorithm. DoG involves the subtraction of one Gaussian blurred version of an original image from another, less blurred version of the original. In the simple case of gray-scale images, the blurred images are obtained by convolving the original gray-scale images with Gaussian kernels having differing width (standard deviations) [8].

Moreover, this algorithm can be used to approximate the second derivative of the Gaussian (Laplacian of the Gaussian, LoG), which is a very common blob detector. BLOB stands for Binary Large OBject and refers to a group of connected pixels in a binary image.

Given an input image $f(x,y)$ and a Gaussian kernel $g(x,y,t) = \frac{1}{2\pi t}e^{-\frac{x^2+y^2}{2t}}$ we can convolve them obtaining

$$L(x,y;t) = g(x,y,t) * f(x,y) \tag{4.1}$$

The Laplacian ($\nabla^2 L = L_{xx} + L_{yy}$) is then calculated by convolving the image with a Laplacian kernel.

The Gaussian step before the Laplacian is used to smooth the image and attenuate noise. The Laplacian highlights regions of rapid intensity change. As the scale t of LoG increases, blob-like structures converge to local extrema at some scale. This intuition motivates using LoG filter in blob detection [9].

As mentioned before, the LoG can be approximated using the DoG algorithm. However, the DoG can be more efficienly computed than the LoG operation. Using the DoG, we can detect blobs with a radius s by using Gaussian filters with standard deviations s and $k \cdot s$. The ratio of $k = 1.6$ is recommended because of design considerations balancing bandwidth and sensitivity [10].

By using a DoG with standard deviations equal to 9 and $1.6 \cdot 9$, we can reduce the noise in the image, highlighting the visible (bright) ions (see Fig. 4.2).

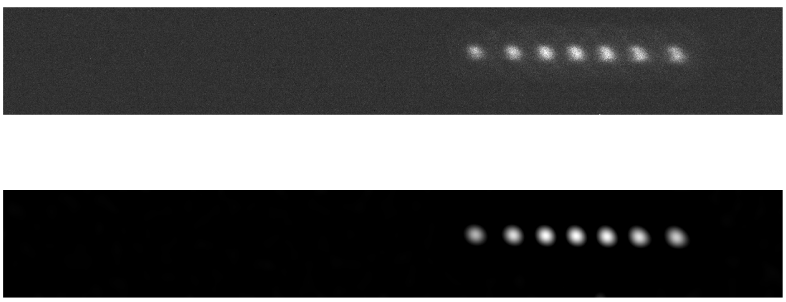

Figure 4.2: Up: Original cropped image. Down: Filtered image using the DoG algorithm.

The coordinates of the bright ions are then easily found by finding the local maxima in a region of $2 \cdot min\_distance + 1$ (peaks are separated by at least min_distance). We use a value of $min\_distance = 15$. When finding the local maxima we need to define a low threshold to discard all false peaks created by noise. This threshold influences directly on this algorithm's performance, as mentioned in Chapter 6.1.

4.1.2 Dark ions

In Chapter 2.1 we described how the Barium ion has an energy level structure with two long lived D states ($D_{3/2}$, $D_{5/2}$). Given that the $D_{5/2}$ state is inaccessible by the cooling cycle, we can consider it a dark state (the ion won't be visible). A trapped ion can get excited into this dark state by absorbing a 455nm photon, which are present in the room light, and thus exiting the cooling cycle and stop emitting visible light (see Fig. 4.3).

To detect dark ions we first make use of the derivation done in Chapter 2.2.3 of the equilibrium positions of ions in a linear trap. Numerically solving the system of equations we obtain the scaled positions of 2-20 ions. To simplify the task, we move the origin of coordinates from the center of the ion string to the leftmost ion and we save the final values in a csv (Comma-Separated Values) file named *mask.csv*. In addition, we compute the distances between neighbour ions and save the result in another csv file named *mask_dist.csv* (see Chapter 7).

Given that these are scaled values, we need a scaling factor when comparing them to the measured distances. This scaling factor can be easily computed when there are only bright ions in the trap. Therefore we created a small script named *calibrate.py* (see Chapter 7) that the user should run when she/he is certain that there are no dark ions present. This script detects the position of visible ions, compares their relative distances with respect to the theoretical values, computes the scaling factor and saves it in shared memory. Fortunately this script only needs to be executed once, given that the scaling factor should not change as long as axial trap potentials are not altered.

For every input image we follow the next steps. Firstly we detect the positions of bright ions (as mentioned in Chapter 4.1.1). Afterwards we calculate the relative distances of the detected ions with respect to the leftmost and rightmost ion together with the neighbouring distances. Subsequently we compare these three sets of values to the aforementioned theoretical values (after rescaling them) to find the number of ions that best fits the detected positions. In other words, we calculate a fitting loss (mean squared error) for each combination of theoretical positions of ions (from 2 to 20 ions) and for each of the three different methods that we used to measure the ions' position, and we keep the theoretical positions that have the smallest loss. Finally we add the missing positions (that correspond to dark states) to the measured positions of the bright ions.

We use these three different methods for the following reasons. On the one hand, using only the relative distance with respect to the leftmost ion would result in a deficient detection every time there is a dark ion in the leftmost position. This limitation can be

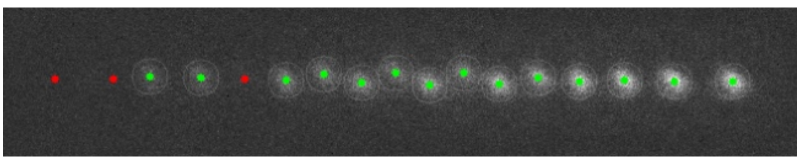

Figure 4.3: Detection of 14 bright ions (green) and 3 dark ions (red).

solved using the relative distance with respect to the rightmost ion. However, we still need the previous method to have a good detection when there is a dark ion in the rightmost position. On the other hand, the neighbouring distances are helpful when there are dark ions in both the leftmost and rightmost positions. However, this last method is inefficient when there are two consecutive dark ions. Consequently, we need the three methods to have a robust dark ion detection.

4.1.3 Most voted value

We noticed that in very specific cases our dark ions' detection algorithm incorrectly selects the total number of ions present in the trap, probably due to a bad calibration of the scaling factor. To avoid these seldom misdetections and increase the robustness of our algorithm, we save the last 5 ion coordinates obtained and output the most voted value. To decide whether two sets of coordinates are equivalent or not (two measurements of the same coordinates might differ a few pixels) we measure the fitting loss between the new obtained value and the previously saved values and compare it to a predefined threshold.

4.2 Trap state

A script controlling ion loading into the trap should be able to identify three different states of the ion trap: the trap contains crystallized ions, the trap is empty, the trap contains an ion cloud.

To decide whether the trap is empty or not we just need to check if there are ions detected. Detecting an ion cloud is slightly more difficult, given that our ion detection method is not robust against ion clouds.

4.2.1 Ion cloud

To detect an ion cloud (see Fig. 4.4) we compute the contours of the image. The ion cloud can be then differentiated from crystallized ions (or no ions) by looking at the number of contours in the image and the area of the biggest contour.

More specifically, we first crop our input image into the region of interest (see Fig. 4.1). Then we apply a Gaussian filter to reduce the noise in the image - in this case we don't need the DoG algorithm that we used in Chapter 4.1.1, given that we can afford loosing some spatial information. Afterwards we convert it into a binary image and use the contour detection algorithm mentioned by Suzuki and Abe [11] to find the image contours. The detection of an ion cloud is accomplished by comparing the area of the largest contour with a predefined thresholding value.

4.2.2 Contrast detection

Another powerful tool which lets us briefly manipulate ion states and extract additional information is based on switching on and off the beam of one of the lasers used in the trap setup.

Looking back at the energy level structure shown in Chapter 2.1, we recap how our repump laser at 650nm is used to excite the ion out of the long-lived $D_{3/2}$ state. Consequently, if we turn off the 650nm laser, after some small delay, all ions will end up falling in this (new) dark state and they will stop emitting light. By doing this, we can measure

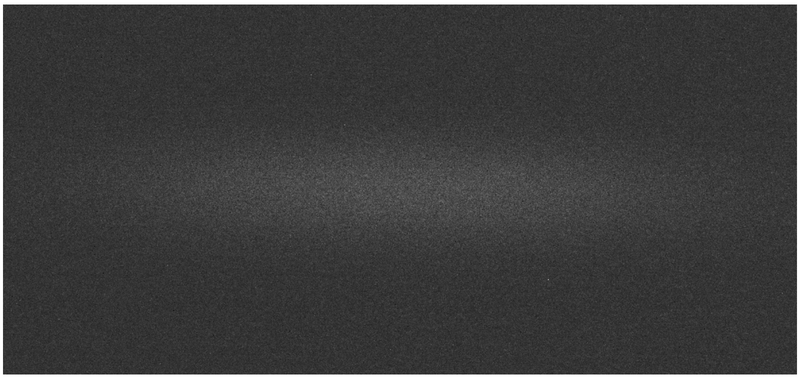

Figure 4.4: Image with an ion cloud.

the intensity contrast between an image with the laser on and another image with the laser off. We then create a boolean *contrast* variable by comparing the aforementioned value with a predefined threshold. With this new variable we can now define 5 different trap states:

1. Crystallized ions: If *contrast* is True and we detect ions in the image.
2. Empty trap: If *contrast* is False and we don't detect ions in the image.
3. Ion cloud: If we detect an ion cloud using the method in Chapter 4.2.1.
4. Bad detection: If *contrast* is False and we detect ions in the image and vice-versa.
5. Connection error: If the connection to the API could not be established.

4.3 Dim ions

Sometimes one of the trapped particles emits less light than the other ones (see Fig. 4.5). Given that our barium oven is not isotopic-pure, when loading new ions it is possible that, by natural abundance, a different isotope than ^{138}Ba is loaded. These "bad" isotopes have frequency shifts, resulting in less emitted light. It may also happen that, by light-assisted chemical reactions, a molecule is loaded into the trap, producing a similar result.

To detect a dim ion, we need to first measure its intensity. To do so we sum the intensity of all the pixels inside a circle of radius 15 around each ion and we save the maximum value obtained. An ion is classified as a "dim ion" when its intensity is lower than $0.55 \cdot I_{max}$.

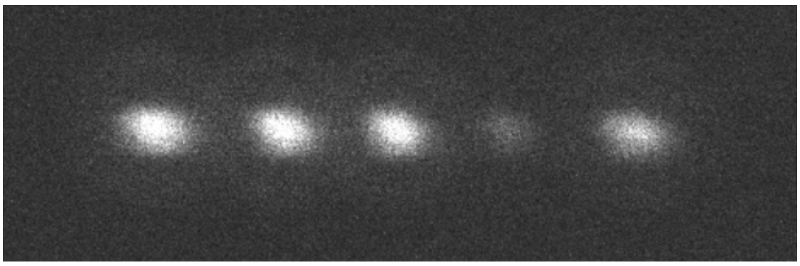

Figure 4.5: Image with 4 bright ions and 1 dim ion.

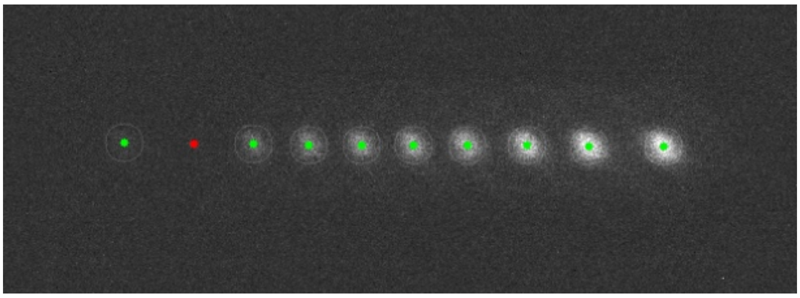

Figure 4.6: Image with the detection of 9 bright ions (green) and 1 dark ion (red). The leftmost ions' are darker due to the intensity gradient on the ions. The area we use to measure each ion's intensity is plotted in white.

4.3.1 Ion intensity gradient

Even though sometimes it is unnoticeable, there exists an intensity gradient on the ions, as seen in Figure 4.6. We must then not confuse the ions affected by this gradient as dim ions. Consequently, we first need to compensate this gradient before classifying any dim ion. We do this by first measuring the intensities of columns of height 30 centered on each visible ion (we only use values of one column to have a well defined x position for each intensity). Using these values together with the x coordinate of each column, we can apply a least-squares linear fit to it and calculate its slope. We then use the calculated slope to compensate the intensity gradient (see Fig. 4.7).

4.4 Ion chain linearity

When calculating the equilibrium positions of ions in a linear trap (Chapter 2.2.3) we neglected the motion along the y and z axes (assuming they are strongly bound in these directions). However, if a large number of ions are stored in the trap, the transverse vibrations can become unstable, and the ions will adopt a zigzag configuration [12] (see Fig. 4.8).

To detect this zigzag effect we apply a least-squares linear fit to the bright ion coordinates and calculate the sum of squared residuals. We then compare this residual value to a predefined threshold to decide whether the ions are in a zigzag configuration or not.

4.5 Reorder rate

Another useful attribute of the trapped ions is its reorder rate. Ions tend to shuffle inside the ion trap. This becomes noticeable when an ion decays into a dark state (after absorbing a 455nm photon), because then it starts jumping back and forth of the ion string. It is interesting to measure the rate of this aforementioned event, because it is an

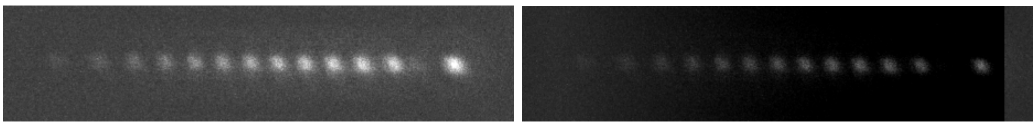

Figure 4.7: Left: Image containing trapped ions with noticeable intensity gradient. Right: Same image after compensating the gradient.

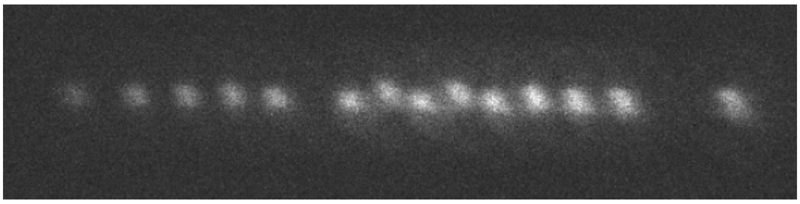

Figure 4.8: Image with ions in a zigzag configuration.

indicator of the stability of the ions inside the trap. A high reorder rate might predict the escape of all the trapped ions, as seen in Chapter 6.5.

To measure the reorder rate we must first be able to differentiate two sets of ion coordinates. This is accomplished by saving, for each ion, its coordinates and its class (bright=0, dim=1, dark=2). We then compare the values of two consecutive images, checking both the coordinates and the classes. If they somehow differ, we assume the ions have shuffled. We finally compute its moving average by summing up the number of events that occurred in the last 60 seconds.

4.6 Spot intensity, size and roundness

Eventually, it might also be interesting to output the intensity, size and roundness of the brightest ion (ion with maximum value in its center pixel). This data can be used to optimize the parameters of the trap to, for example, minimize the spot size and maximize the spot intensity and the roundness.

The spot intensity is calculated using the same technique as in Chapter 4.3. We basically sum the intensity of all the pixels inside a circle of radius 15 around the ion.

To find the spot size, we first use the row values where the brightest ion is located. Since all the other ions should also be located on this row, when plotting it we should see as many peaks as bright ions are in the image (see Fig. 4.9). We then smooth the values of the 1D signal by applying a convolution of a ”Hamming” window [13] with the signal. At last we measure the peak width at half height of the highest peak and output its value.

The roundness is computed using a principal component analysis (PCA) [14] of the pixels of the brightest ion. More specifically, we crop the image around the brightest ion into a 50x50 image centered on the ion. We then get the coordinates of all the pixels with a greater value than half the maximum value of the cropped image, trying to keep

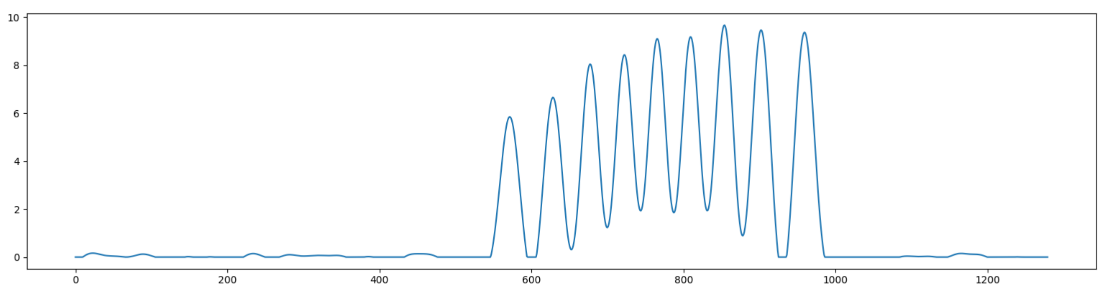

Figure 4.9: Plot of the smoothened row values where the brightest ion is located.

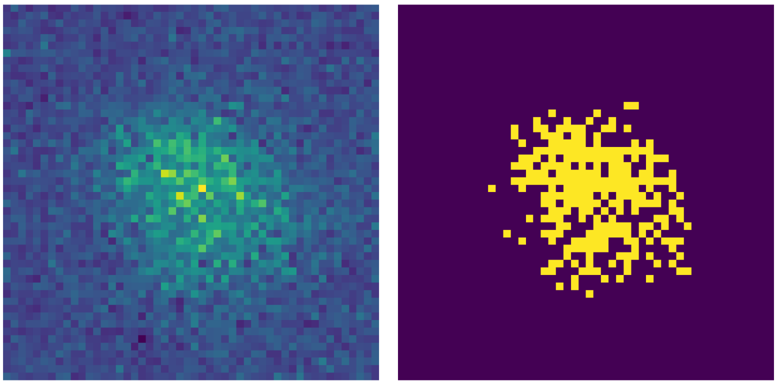

Figure 4.10: Left: Cropped image of the brightest ion. Right: Binary image containing the pixels with value greater than half the maximum value.

only the pixels that correspond to the ion (see Fig. 4.10). We use these coordinates to compute the variance along its two principal components and output the fraction of the variance of the second component divided by the variance of the first one. This fraction can be viewed as a ”roundness” measure given that, if the ion is completely round, both variances will have the same value and the output will tend to 1. If the ion is an ellipse with high eccentricity, the second component will have a very small variance compared to the first one, and the output will tend to 0.

4.7 Background light

We can measure the properties of the background light through turning off and on the beam of the 650nm laser (like in Chapter 4.2.2). More specifically, while the laser is off, we measure the mean and standard deviation of the image intensity. These values give us information about the amount of room light entering the trap and characterize the shot noise in the image.

Chapter 5

Frontend

5.1 Web application

As mentioned in the abstract, we used Flask (a Python API) to design a simple web application that displays the measured image features in real-time.

This is accomplished by first running the image analysis code in the background, and saving all current outputs in shared-memory (updated approximately every 0.2 seconds). Then in our web application code we fetch and display these values (see Fig. 5.1), updating them every second.

5.2 Grafana

Until now we only saved the current outputs by overwriting any past values obtained. However, it may be useful to keep the outputs instead of overwriting them, to check up on past events. This is accomplished using InfluxDB, which is a time-series database

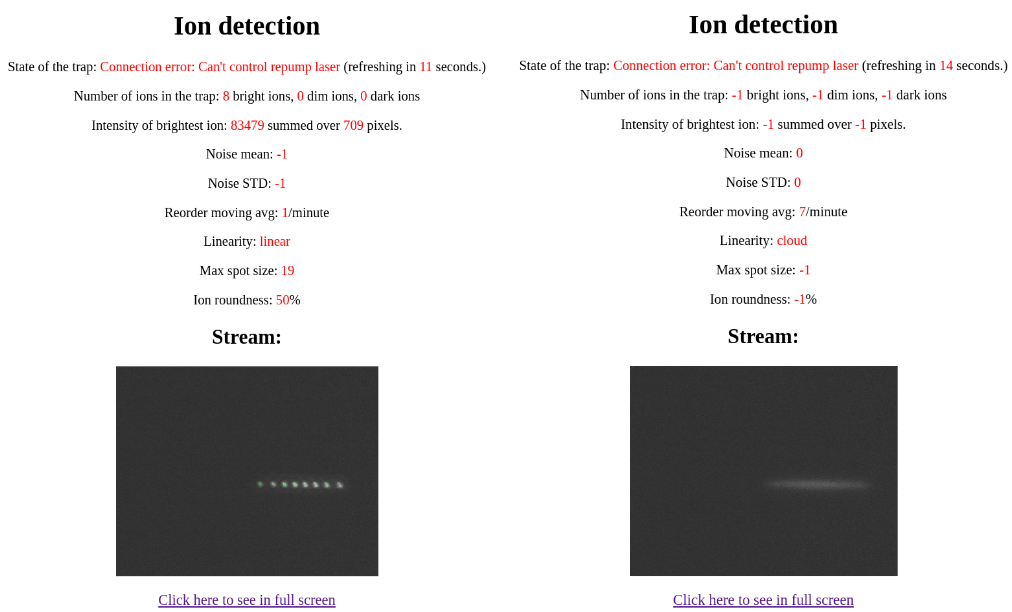

Figure 5.1: Screenshots of our web application. Left: The trap contains 8 bright ions. Right: The trap contains an ion cloud.

developed by InfluxData. We then use Grafana, an open source visualization tool, to show in graphs our database (see Fig. 5.2).

Figure 5.2: Screenshot of our time-series database displayed on Grafana.

Chapter 6

Results

In this chapter we discuss the performance of our code in detecting and measuring all the mentioned features in Chapter 4. Even though we cannot use any performance evaluation method (such as F1-score) because we don't have any labeled dataset, we show some examples and explore the limitations of each feature analysis. In some cases we also describe some interesting results that we have seen.

6.1 Ion coordinates

Thanks to the DoG algorithm mentioned in Chapter 4.1.1, bright ions are detected with a very high precision even if they are really dim, as seen in Figure 6.1. The precision depends highly on the threshold value used to discard any local maxima generated by noise. Lowering the threshold will make the algorithm capable of detecting dimmer ions, but the probability of a false positive detection due to noise will be higher.

The detection of dark ions is also accomplished with very high precision and robustness. Moreover, in some cases, it correctly detects dark ions that are hardly distinguishable by the viewer (see left image of Fig. 6.2). The accuracy of a dark ion's detection can be easily verified when a bright ion occupies the previous dark ion's position (see Fig. 6.2).

As mentioned in Chapter 4.1.2, the detection of dark ions is completely dependent on the scaling factor, which can be calibrated when there are only bright ions in the trap. One can thus expect that the detection will depend on the precision of the aforementioned scaling factor. In addition, given that we find the dark ions' positions by choosing the number of ions that will best fit the positions of the detected bright ions, in very few cases it may happen that a wrong number of ions (usually higher than the correct number) produces the smallest fit error, and is thus wrongly selected.

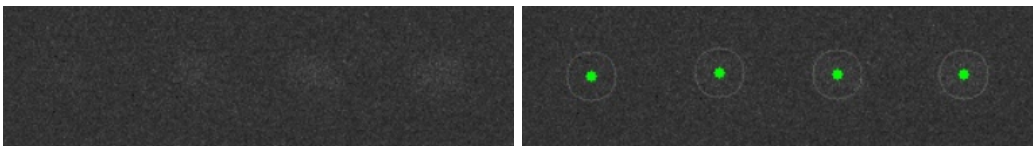

Figure 6.1: Left: Cropped input image with 4 barely visible ions. Right: Same image with the ion coordinates plotted in green and the area used to measure the intensity plotted in white.

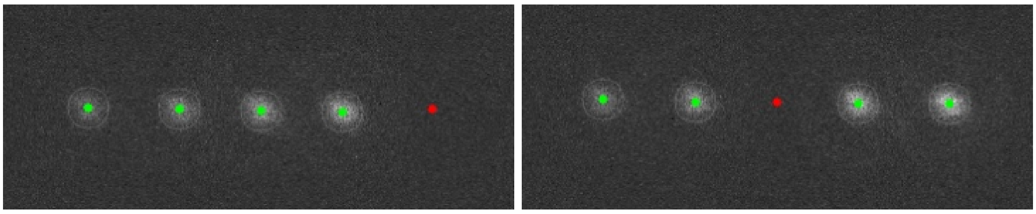

Figure 6.2: Detection of 4 bright ions (green) and 1 dark ion (red) for two different position configurations of the same group of trapped ions.

6.2 Trap state

In Chapter 4.2 we defined 5 possible states: crystallized ions, empty trap, ion cloud, bad detection and connection error.

The ”crystallized ions”, ”empty trap” and ”bad detection” states are subjected to a good ion detection and a good contrast measurement. As long as bright ions are always detected and a good contrast threshold value is set (to decide whether we have an intensity contrast when turning off the 650nm laser), the ”crystallized ions” and ”empty trap” states will be correctly selected. One possible unfortunate case is having only 1 dark ion in the trap, which will incorrectly yield an ”empty trap” state. We can nonetheless accept this mistake, considering that not even a human would be able to correctly identify this dark ion.

The ”connection error” state is displayed when we can't connect to the API, and thus cannot be subject to error.

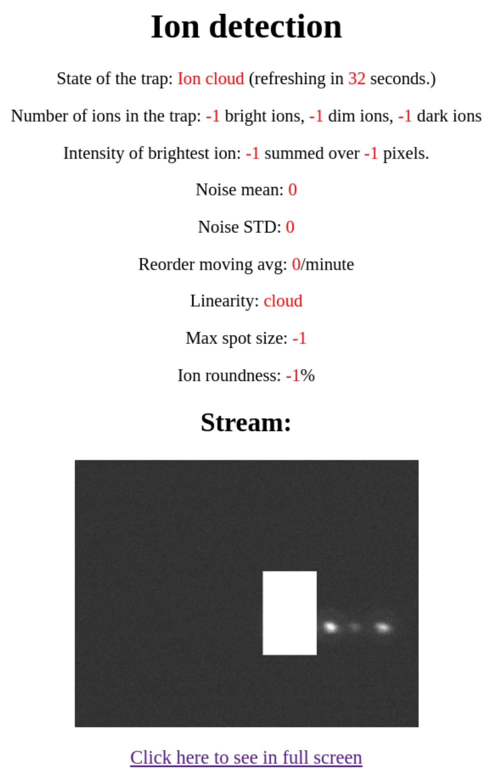

Figure 6.3: Screenshot of the outputs of our image analysis of a simulation of room light entering the trap.

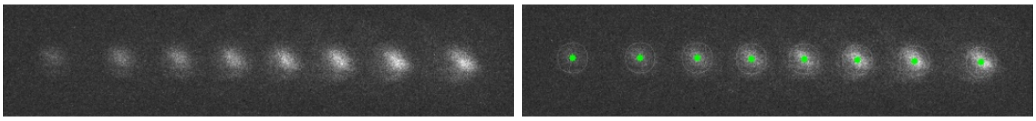

Figure 6.4: Left: Cropped image of trapped ions with a visible intensity gradient. Right: Detection and classification of all ions as bright ions (green). There is 0 dim ions (yellow).

Last but not least, by empirically evaluating the ion cloud detection performance, we can infer that the "ion cloud" state is correctly selected (such as in Fig. 5.1) with high precision. There is however one possible drawback of our detection method. By recapping what we discussed in Chapter 4.2.1, the ion cloud detection is made by measuring the area of the biggest contour in the image. With this in mind, in the rare case when room light enters the trap (through the eyepiece) and covers up some of the ions (but there are some ions still visible), our method would incorrectly classify the trap state as an ion cloud. We simulated this event by whitening a big fraction of our image, as seen in Figure 6.3.

6.3 Dim ions

As mentioned in Chapter 4.3, the detection of dim ions is done by comparing the intensity of all ions with the intensity of the brightest ion. This technique requires a threshold value for classification, which is set to $0.55 \cdot I_{max}$. We also developed a method to compensate the ion intensity gradient, which would otherwise affect the result.

The performance of our procedure depends of course on the value of the intensity threshold. On the one hand, a low threshold will result in a high precision but a low recall (some dim ions may be classified as bright ions). On the other hand, increasing the threshold will yield a high recall but with low precision (probably all dim ions will be correctly classified, but some of bright ions may be also classified as dim ions). We also tested our intensity gradient compensation, which works satisfactorily as seen in Figure 6.4.

One drawback of our method is that dim ions can only be detected as long as there is a bright ion in our trap. If all ions are "bad" isotopes, the intensity of the brightest ion will of course be very similar to all the other intensities and thus all ions will be classified as "bright ions".

6.4 Ion chain linearity

For detecting whether we have a zigzag configuration or not we apply a linear fit to the bright ion coordinates and compare the error to a threshold value, as mentioned in

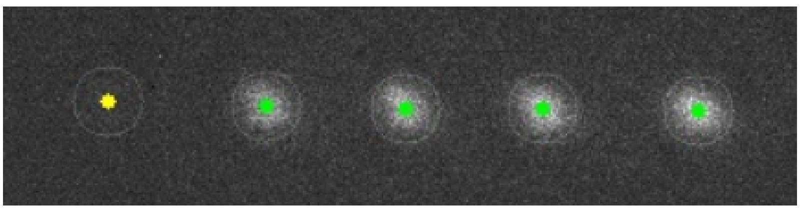

Figure 6.5: Image with the detection of 4 bright ions (green) and 1 dim ion (yellow).

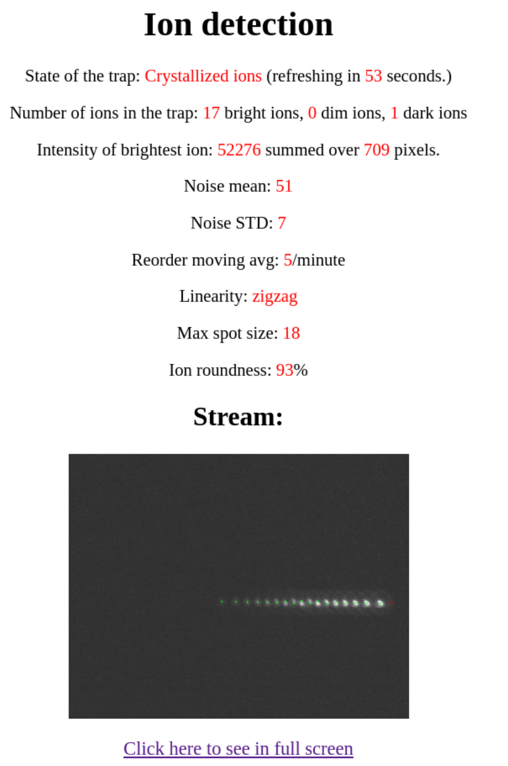

Figure 6.6: Detection of the zigzag configuration.

Chapter 4.4.

After selecting a suitable threshold, the zigzag detection seems to work flawlessly (see Fig. 6.6).

6.5 Reorder rate

The accuracy of the reorder rate measurement is completely related to the precision of the ion coordinates and type detection (bright, dim, dark).

It is important to mention that our method of detecting dim ions might incorrectly increase the reorder rate. More specifically, when the intensity of an ion is very close to the intensity threshold used in Chapter 4.3, the ion's detected class might oscillate between "bright" and "dim", simulating a reorder event and causing the reorder rate to increase.

Additionally, thanks to Grafana we can observe in Figure 6.7 the outcome of one ion falling into a dark state. At first we have 5 bright ions and 1 dark ion, and we have a very low reorder rate. Then around 15:24 one bright ion falls into a dark state, reducing the amount of ions inside the cooling cycle. Consequently, the crystal heats up and its shuffling rate increases (as seen in the figure). When the ion leaves the dark state the crystal recovers the initial dynamics.

Finally, we can look at another interesting example. In Figure 6.8 we see the importance of measuring the reorder rate. We see how, due to unknown circumstances, the value starts increasing rapidly around 05/16 10:00. The rapid increase of the reorder rate is an indicator that something is not working as expected, and may cause all ions to escape

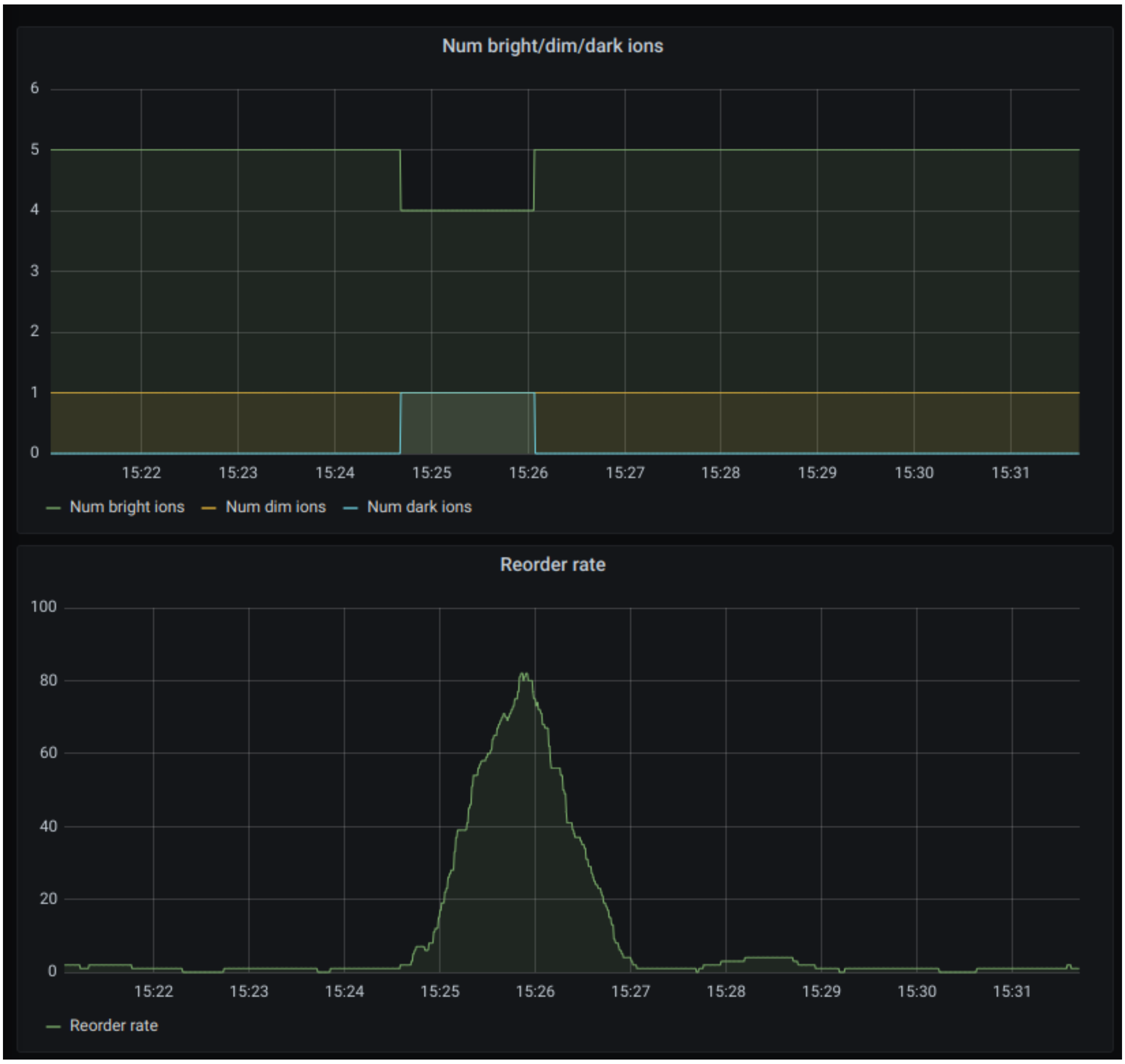

Figure 6.7: Graph showing an increase of the reorder rate due to an ion falling into a dark state.

the trap, which in this case happens shortly after 05/16 12:00. Moreover, we also observe an unusual number of dark ions which most probably means that the scaling factor is not well calibrated (see Chapter 6.1).

6.6 Spot intensity, size and roundness

As mentioned in Chapter 4.6, our goal was to use the spot intensity, size and roundness data to optimize the parameters of the trap. Thus what is important is not measuring the true values (which would also be hard to define) but the correlation between the measured values and the true values.

One way to check that everything is working properly is to make sure that these values are practically constant when the ions are stable. Figure 6.9 shows a graph of the intensity, size and roundness of the brightest ion of a string of 7 trapped ions. First of all, we see how these values are almost constant before 14:00. Secondly, we notice a sudden change around 14:00, which would seem at first that our method is not robust enough. However, we finally discovered that this alteration took place because the lights

Figure 6.8: Graph showing a rapid increase of the reorder rate and escape of all the ions.

of the room where the trap is located were switched on around that time. Most probably the room light entered the trap and interacted with the trapped ions, changing its state. Thanks to this small incident we know that the measured values are somehow correlated to the actual state of the ions.

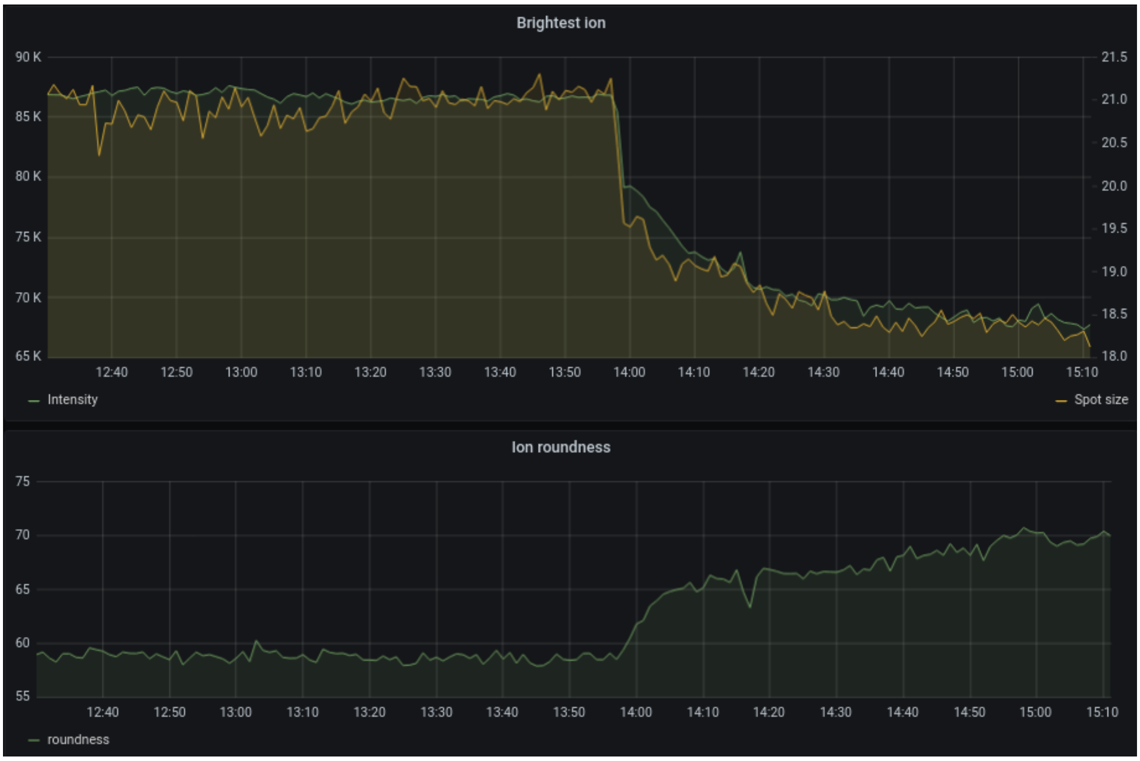

Figure 6.9: Graph showing the spot intensity, size and roundness of the brightest ion.

Chapter 7

Project Structure

The files inside this project's folder are structured as follows:

```
/.
├── data
│   ├── mask_dist.csv
│   └── mask.csv
├── templates
│   └── index.html
├── app.py
├── calibrate.py
├── settings.py
├── tools.py
├── README.md
└── requirements.txt
```

The files *mask.csv* and *mask_dist.csv* inside the folder *data* contain the theoretical values of the ion positions inside a linear trap, as explained on the second paragraph of Chapter 4.1.2. The file *index.html* inside *templates* contains the html code for the web application. The web app is generated using the Flask code inside the file *app.py*. This file is also responsible of running the image analysis in the background. The calibration of the scaling factor is done in *calibrate.py* (see third paragraph of Chapter 4.1.2). The file *settings.py* contains all the needed constants for our code such as thresholds, region of interest, time-series database information and more. All the needed functions and classes are defined in *tools.py*. This file also contains a class called $Main()$, which creates a background thread and constantly runs the image analysis in there. We finally have the *README.md* file which provides a small insight into the code usage and the *requirements.txt* file which contains all the needed libraries to run the code.

Chapter 8

Conclusions

This project was aimed to build a system for analysing images of trapped ions to extract the most relevant features for the completely unattended operation of the trap. Our work delivers on this goal and presents a solution that

- is capable of analyzing images in real-time, computing its features at a frequency of approximately $4s^{-1}$,
- is able to measure features that are imperceptible to the human eye, such as the position of dark and dim ions and ion roundness,
- is easily adaptable to a wide set of different setups,
- offers a simple and lucid web app interface for debugging,
- provides graphs displaying a time series database of some of the measured features.

We also claim that our image analysis tool will make possible the following achievements:

- Unsupervised loading of ions, thanks to the precise detection of the number of ions inside the trap.
- Automation of laser frequency tuning, by virtue of the measurement of the ion brightness and reorder rate.
- Automation of micromotion compensation, thanks to our spot size and roundness estimation.
- Detection of anomalies such as excessive axial potential, by detecting if the ions have a zigzag configuration, and having too many ions or wrong laser frequencies, via ion cloud detection.

Appendix A

Mathematical description of Doppler cooling

Doppler cooling is a technique for reducing the motional energy of small particles (typically atoms or ions) due to momentum transfer from atoms to light during off-resonant scattering. For simplicity, we consider only the interaction between a single-frequency laser with a two-level atom. We will follow a similar derivation as the one done in [15].

We start with the equation of the photon-atom interaction Hamiltonian in the dipole approximation

$$\hat{H}_{int} = \vec{d} \cdot \vec{E}(\vec{R}_0) \tag{A.1}$$

Where $\vec{d}$ is the dipole vector and R_0 is the position of the center of charge of the atom.

The electric field of the traveling wave can be written as

$$E(z) = \frac{E_0}{2}(e^{i(\vec{k}\hat{R}-\omega_L t)} + e^{-i(\vec{k}\hat{R}-\omega_L t)}) \tag{A.2}$$

and the dipole vector as

$$\vec{d} = \mu_{eg}(\sigma_+ + \sigma_-) \tag{A.3}$$

After moving to an interaction picture and applying a rotating-wave approximation we end up with the Hamiltonian

$$H_I = -\frac{\hbar\delta_L}{2}\sigma_z + \frac{\hbar\Omega(\hat{R})}{2}(\sigma_+ e^{i\vec{k}\hat{R}} + \sigma_- e^{-i\vec{k}\hat{R}}) \tag{A.4}$$

where $\delta_L = \omega_{eg} - \omega_L$ is the detuning of the laser and $\Omega = \frac{\mu_{eg}E_0}{\hbar}$ is the rabi frequency.

Using the Ehrenfest theorem we can derive an expression for the force resulting from the optical potential

$$\vec{F} = \left\langle \frac{d\hat{P}}{dt} \right\rangle = \left\langle \frac{i}{\hbar}[\hat{H},\hat{P}] \right\rangle = \left\langle -\vec{\nabla} H_I \right\rangle \tag{A.5}$$

Inserting Eq. A.4 into Eq. A.5 we obtain

$$\vec{F} = \left\langle -\frac{\hbar}{2}\vec{\nabla}\Omega(\hat{R})(\sigma_+ e^{i\vec{k}\hat{R}} + \sigma_- e^{-i\vec{k}\hat{R}}) + \frac{\hbar\Omega(\hat{R})i\vec{k}}{2}(\sigma_+ e^{i\vec{k}\hat{R}} - \sigma_- e^{-i\vec{k}\hat{R}}) \right\rangle \tag{A.6}$$

To help us with the calculations, we now make use of two approximations: we will make a mean-value approximation by replacing the operators by its mean values, we will then make a separation of timescales approximation, by considering that the timescale to reach the internal steady-state is small compared to any significant changes in $\vec{R}$ and v. Therefore we can use the steady-state of the internal state operators.

By using the Liouville-von Neumann equation for the density operator we can derive a set of optical Bloch equations. We can write them into Bloch vector components defined as

$$u_m = \langle\sigma_+\rangle e^{i\vec{k}\vec{R}_m} + \langle\sigma_-\rangle e^{-i\vec{k}\vec{R}_m} \tag{A.7}$$

$$v_m = i(\langle\sigma_+\rangle e^{i\vec{k}\vec{R}_m} - \langle\sigma_-\rangle e^{-i\vec{k}\vec{R}_m}) \tag{A.8}$$

$$\omega = \langle\sigma_{ee}\rangle - \langle\sigma_{gg}\rangle \tag{A.9}$$

We can then calculate the steady-state solutions to these

$$u_m^S = \frac{\delta\Omega}{\left(\frac{\Gamma}{2}\right)^2 + \delta^2 + \frac{\Omega}{2}^2} \tag{A.10}$$

$$v_m^S = \frac{\Gamma\Omega/2}{\left(\frac{\Gamma}{2}\right)^2 + \delta^2 + \frac{\Omega}{2}^2} \tag{A.11}$$

$$w_m^S = -\frac{\left(\frac{\Gamma}{2}\right)^2 + \delta^2}{\left(\frac{\Gamma}{2}\right)^2 + \delta^2 + \frac{\Omega}{2}^2} \tag{A.12}$$

where $\delta = \delta_L - \vec{k}\vec{v}$.

Inserting these operators into Eq. A.6 we get

$$F = -\frac{\hbar}{2}u_m^S \nabla\Omega + v_m^S \frac{\hbar\vec{k}\Omega(\vec{R}_m)}{2} \tag{A.13}$$

From this equation two forces can be derived, namely the dipole force and the scattering force. The dipole force (first term) arises due to the excitation of the dipole of the atom in phase with the laser drive. This term is dominant for very large detunings ($\delta >> \Omega, \Gamma$). In this limit we find

$$\vec{F}_{dipole} \approx -\frac{\hbar}{2}\vec{\nabla}\left(\frac{\Omega^2}{4\delta}\right) \propto \vec{\nabla}(I) \tag{A.14}$$

which means that the dipole force points towards the direction of maximum intensity. Consequently, this force can be used to trap atoms.

The scattering force (second term) can be written

$$\vec{F}_s = \hbar\vec{k}\Gamma\rho_{ee} = \hbar\vec{k}R_{scatt} \tag{A.15}$$

For velocities close to zero, we can expand the force with respect to the components of the velocity as

$$\vec{F}_s(\vec{v}) = \vec{F}_s(\vec{v}=0) + \eta\vec{k}|\vec{v}|cos(\theta) \tag{A.16}$$

where η is the damping parameter

$$\eta = -\hbar|\vec{k}|^2 \left(\frac{\partial R_{scatt}}{\partial \delta}\right)_{\vec{v}=0} \tag{A.17}$$

The first term of Eq. A.16 produces a steady force on the atoms, which must be countered by external forces. The second term is a velocity dependent force, the sign of which depends on the laser detuning δ_L. In the 1-dimensional case, where the laser points along the $\hat{x}$ axis, the force is in the same direction as the motion, and $\theta = 0$

$$\vec{F}_x = \vec{F}_x(\vec{v}=0) + \eta v_x \tag{A.18}$$

We see that if $\delta_L < 0$ we get a friction term, for which the force is always in the opposite direction to the velocity of the atom. This slows the atom down, producing an effect known as Doppler cooling.

This effect occurs because the atom tends to scatter more photons when moving towards the laser. When absorbing a photon, the atom gets a momentum kick in the direction of the laser beam. The re-emission of the photon, however, is in probabilistic direction, giving rise to net momentum transfer on the ions similar to radiation pressure. Given that there is a preference for absorbing photons which have momentum opposite to that of the initial velocity, on average the atom slows down.

Bibliography

[1] Monroe group. Barium. [Online]. [Accessed 3 June 2021]. Available from: https://mike.depalatis.net/ionptable/barium/.

[2] A. Kriesch. Quadrupole ion trap, 2006. [Online]. [Accessed 3 June 2021]. Available from: https://en.wikipedia.org/wiki/Quadrupole_ion_trap.

[3] Joseph N. Tan, J. J. Bollinger, B. Jelenkovic, and D. J. Wineland. Long-range order in laser-cooled, atomic-ion wigner crystals observed by bragg scattering. *Phys. Rev. Lett.*, 75:4198–4201, Dec 1995.

[4] Richard C. Thompson. Ion coulomb crystals. *Contemporary Physics*, page 1–17, Jan 2015.

[5] (2005) Plasma Crystallization. In: Charged Particle Traps. Springer Series on Atomic, Optical, and Plasma Physics, vol 37. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-26576-7_18.

[6] E. L. Pollock and J. P. Hansen. Statistical mechanics of dense ionized matter. ii. equilibrium properties and melting transition of the crystallized one-component plasma. *Phys. Rev. A*, 8:3110–3122, Dec 1973.

[7] D.F.V. James. Quantum dynamics of cold trapped ions with application to quantum computation. *Applied Physics B: Lasers and Optics*, 66(2):181–190, Feb 1998.

[8] Wikipedia. Difference of gaussians. [Online]. [Accessed 28 June 2021]. Available from: https://en.wikipedia.org/wiki/Difference_of_Gaussians.

[9] Hui Kong, Hatice Cinar Akakin, and Sanjay Sarma. A generalized laplacian of gaussian filter for blob detection and its applications. *Cybernetics, IEEE Transactions on*, 43:1719–1733, 01 2013.

[10] D. Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character. Royal Society (Great Britain)*, 207:215–217, 02 1980.

[11] S. Suzuki and K. Abe. Topological structural analysis of digitized binary images by border following. *Comput. Vis. Graph. Image Process.*, 30:32–46, 1985.

[12] J. P. Schiffer. Phase transitions in anisotropically confined ionic crystals. *Phys. Rev. Lett.*, 70:818–821, Feb 1993.

[13] Wikipedia. Hamming window. [Online]. [Accessed 30 June 2021]. Available from: https://en.wikipedia.org/wiki/Window_function#Hamming_window.

[14] Wikipedia. Principal component analysis. [Online]. [Accessed 30 June 2021]. Available from: https://en.wikipedia.org/wiki/Principal_component_analysis.

[15] J. Home. Quantum optics. ETH Zürich lecture notes, 2020.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

ION TRAP IMAGE ANALYSIS

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
MITTJANS-COMA	ALBERT

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Barcelona, June 10, 2021	A Mitjans

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.