Abstract

This thesis studies an algorithms for the detection of circles which are produced by particles travelling through the RICH detectors generating Cherenkov radiation. At the beginning there is an introduction to the linear hough transform and then to the hough transform to detect circles. The first case considered is when there is only one unknown (radius) and the center is known. The second case is with two unknowns where the radius is known but the center unknown ((x, y) coordinates of the center are the two unknown parameters). Lastly there is the case where all the parameters are unkown. For this case there is on the one hand the traditional approach studied with the conventional accumulator space and on the other hand a new approach is developped and studied. This approach works on the basis that each circle is defined by 3 points. Once there are three points the radius and the center can be caculated.

1 Introduction

1.1 LHC - Large Hadron Collider

The Large Hadron Collider (LHC) is a proton-proton collider. It is the largest and highest-energy particle accelerator in the world. It was built by the European Organisation for Nuclear Research from 1998 to 2008. It aims to test the predictions of different theories in high-energy particle physics, and in particular for the search of the Higgs boson (which has been confirmed this year) and signs for new physics beyong the Standard Model of particle physics. The LHC lies in a tunnel 27 km in circumference and 100 m below the surface of the French-Swiss border near Geneva. The LHC was built in collaboration with over 10000 scientists and engineers from over 100 countries. The accelerator has been running with a center of mass energy $\sqrt{s} = 13$ TeV since 20 May 2015.

The core physics programm for the period 2014-2015 includes:

- <COMMENT: PROGRAM>

1.2 LHCb

The LHCb is one of the four big experiments conducted at the LHC (ATLAS, CMS and ALICE being the other 3). The aim of this experiment is the study of decays of particles containing b and $\bar{b}$ particles (B-Mesons). During collisions these particles don't fly in all direction but rather stay close to the beam pipe. This is reflected in the design of the LHCb detector which is a foward arm spectrometer 20 meters long with subdetectors along the beam pipe as seen in figure 1.2.

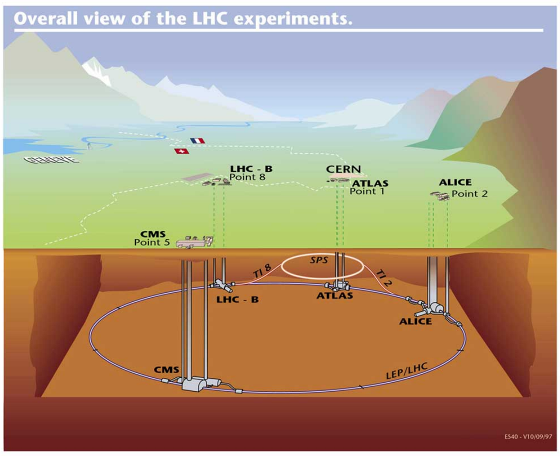


FIG. 1.1 – The LHC ring with its 4 experiments: ATLAS, CMS, Alice and LHCb

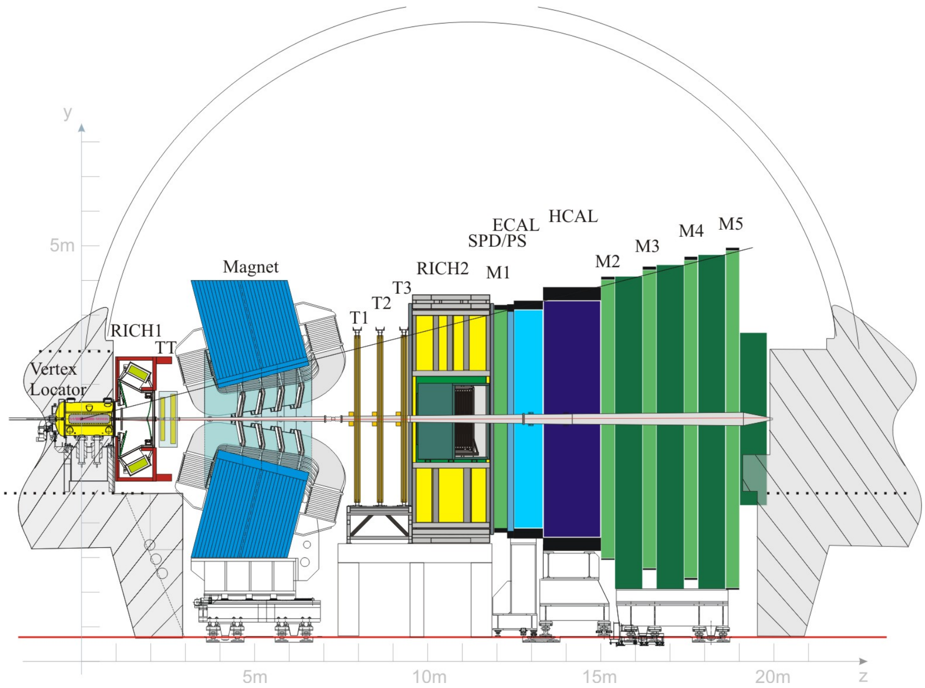


FIG. 1.2 – LHCb Detector: RICH1 before the magnet and RICH2 after the magnet with silicon strip detector in between and muon and calorimeter at the end.

A quick overview of the detector parts [1].

VELO The VErtex LOcator is where the beams collide and $b/\bar{b}$ are produced. The VELO measures the distance between the photon collision point and the point where B particles decay. Thus the B particles are not measured directly but inferred from the separation of these two points.

RICH The RICH detectors are bult for particle identification. One detector on each side of the magnet are used to cover different momentum ranges. RICH detectors work by measuring emissions of Cherenkov radiation which is emmited if a particle travels faster than the speed of light through a certain medium (often compared to breaking the sound barrier).

Magnet A particle normally moves in a straight line but entering a magnetic field causes the path of charged particles to curve according to the Lorentz force

$$\mathbf{F} = q\left(\mathbf{E} + (v \times \mathbf{B})\right)$$

thus allowing to determine the charge of the particle. Also the track curvature can be used to infer the momentum of the particle.

Tracking System The tracking system is based on 4 large tracking stations, each covering about 40 m^2. It is used to determine the position where the particle passed the detector. In the silicon detector a charge gets deposited on a strip which defines the position and in the gas-filled tubes of the outer tracker a passing particle ionizes the gas molecules producing electrons.

Calorimeters They are designed to stop particles and measing the amount of energy lost while coming to a halt. The design of the stations is sandwich like. One metal plate and one plastic plate. A particle hitting the metal plate causes a secondary shower which induces a UV light in the plastic plate. The energy lost is proportional to the amount of light emmitted. It is the main way of identifying particles with no charge (e.g. photons, neutrons).

Muon system Muons are present in the final stage of B decays and thus important for the LHCb experiment. There are 5 rectangular stations increasing in size at the end of the detector. The total area covered by these stations is about $435\,\mathrm{m}^2$. Each station is filled with a combination of 3 gases. Passing muons react with this mixture and wire electrodes detect the result.

1.2.1 Particle Identification

An important requirement at LHCb is the particle identification. This is handled by CALO, Muon and RICH sub-detectors. The Calorimeters beside measuring energies and positions of electrons, photons and hadrons also provide indentification of said particles. The Muon system identifies muons to a very high level of purity which is esential for many J/Ψ's in their final states.

Hadron identification is very important for decays where the final states of interest are purey hadronic. The LHCb RICH system provides. It is composed of two detectors. One positioned upstream of the dipole magnet and the other one positioned downstream of the dipole magnet. The optics is arranged similarly in both sub-detectors: spherical focusing mirrors project the Cherenkov photons onto a series of flag mirrors which then reflect them onto a series of photon detector arrays, located outisde the detector acceptance [4].

2 Theory

2.1 Cherenkov-Radiation

The speed of light in vacuum, **c**, is a universal physical constant. According to Einstein's special theory of relativity, c is the maximum speed at which all matter (or information) in the universe can travel. The speed at which light propagates in a medium, however, can be significantly less can c.

Cherenkov radiation results when a charged particle travels through a dielectric medium with a speed greather than the speed of light through said medium. Moreover, the velocity that must be exceeded is the phase velocity (v_{Phase} or short v_{P}) and not the group velocity $v_{\text{Group}} = \frac{\partial\omega}{\partial k}$.

$$v_{\text{P}} = \frac{\lambda}{T} \quad \text{or} \quad \frac{\omega}{k}$$

As a charged particle travels through the medium, it disrupts the local electromagnetic field. If the particle travels slowly then the disturbance elastically relaxes to the mechinal equilibrium as the particle passes. However, if the particle travels fast enough, the limited response speed of the medium means that a disturbance is left in the wake of the particle, and the energy in this disturbance radiates as coherent shockwave.

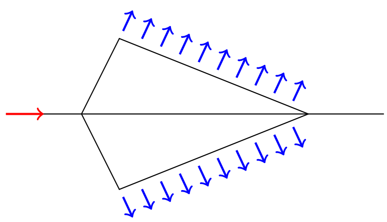


FIG. 2.1 – Cherenkov radiation

$$\begin{aligned}
x_p &= v_p \cdot t = \beta c t \\
x_{\text{em}} &= v_{\text{em}} \cdot t = \frac{c}{n} t \\
\cos\theta &= \frac{x_{\text{p}}}{x_{\text{em}}} = \frac{\frac{c}{n} t}{\beta c t} = \frac{1}{n\beta}
\end{aligned}$$

which is independent from the angle θ.

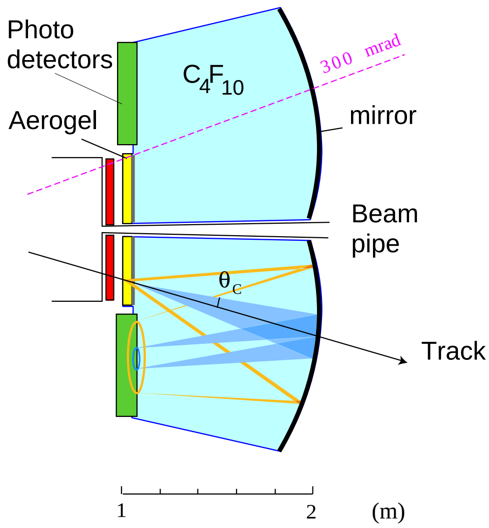


FIG. 2.2 – RICH-1 detector [3].

2.2 RICH Detector

Particle identification is a fundamental requirement at the LHCb experiment. Meaningful CP-violation measurements are only possible if hadron identification is available hence the ability to distinguish between kaons and pions is essential. The LHCb experiment is unique in the sense that its hadronic particle identification is handled only by the RICH sub-detectors. This means – as mentioned before – that the RICH has to cover a wide range of momentum (1-100 GeV/c).

The RICH-1 in front of the magnet covers a lower momentum range from 1-60 GeV/c. It is composed of 5 cm thick aerogel tiles arranged around the beam pipe. The aerogel with $n = 1.03$ is suited for the lowest momentum tracks. Directly behind the aerogel is circa 1 m of C_4F_{10} which covers the intermediate region of momentum. For the highest momentum tracks, gaseous CF_4 is used in the RICH-2.

There is a strong corelation between the polar angle and momentum of the tracks. Tracks with a wider angle often have lower momentum. That is why RICH-1 with the aerogel is located before the dipole magnet so tracks with low momentum will be covered before they swept out of the acceptenace by the magnet.

Both sub-detectors are located in low magnetic field regions to keep the tracks straight while they pass through the radiators.

2.3 Hough Transform

The Hough-Transform is a feature extraction technique used in image analysis, computer vision and digital image processing.

the purpose is to find imperfect instances of objects within a certain cass of shapes by a voting procedure. This voting procedure is carried out in a parameter space from which object candidates are obtained as local maxima in a so called accumlator space that is explicitly constructed by the algorithm for computing the Hough-Transform.

Initially the Hough-Transform was concerned with finding straight lines but has been extended to identifying positions of arbitrary shapes, such as circles and ellipses.

2.3.1 Linear Hough Transform

A linear function is normally defined as the following:

$$f(x) = m \cdot x + b$$

where m is the slope of the line and b the intercept. For the Hough-Transform however, this representation is not ideal. For a vertical line m would go to infinity which gives us an unbound transform space for m. For this reason Duda and Hart suggested the ρ-θ parametrization [2].

$$r = x \cos\theta + y \sin\theta$$

where r is the distance from the origin to the closest point in the line and θ is the angle between the x-axis and the line connecting the origin with that closest point.

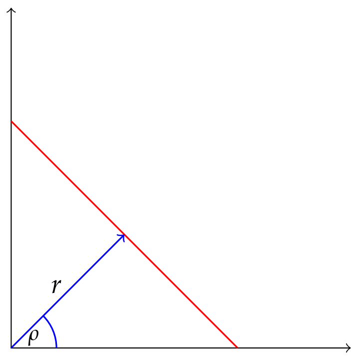
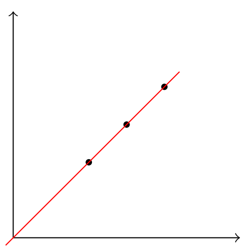


FIG. 2.3 – ρ-θ parametrisation

This means given a single point in the plane, the set of all lines going through this point form a sinusoidal curve in ρ-θ space. Another point that lies on the same straight line in the plane will produce a sinusoidal curve that intersects with the other at (ρ-θ) and so do all the points lying on the same straight line.

Example of a Linear HT



TAB. 2.1 – Angle vs Distance

Angle	Distance
0	2.334

2.3.2 Circle Hough Transform

For this thesis we are interested in circle detection so we need to adapt our linear Hough Transform in order to find circles. In a two dimensional space, a circle can be described by:

$$(x - c_x)^2 + (y - c_y)^2 = r^2 \tag{2.1}$$

Where (c_x, c_y) is the center of the circle and r the radius. The possible parameters for the parameters space are now c_x, c_y and r. This means if we know the center of the circle the parameter space is one-dimensional and if we know the radius of the circle our parameter space is two-dimensional and of course if we know nothing the parameter space is three-dimensional.

3 Methods

3.1 Conventional Hough-Transforms

In the following subsections we discuss the conventional Hough-Transforms for the case of a one, two and three dimensional parameter space. These methods were mainly considered to get an idea what was possible with the conventional hough transform. For closer study the method of choice was the combinatorial approach discussed in depth in section 3.2.

3.1.1 1D: Known Center - Find Radius

In this case the center(s) of the circle(s) is/are known so only the radius is missing. For the radius there is an array with a minimum value and increasing by a defined stepsize to the maximum possible radius value. For the example in this thesis the minimum is 0, the maximum radius is 1 and stepsize equal to 0.001. Introducing following scoring function $\eta(r)$ allows to find the right radius.

$$\eta(r) = (c_x - x)^2 + (c_y - y) - r^2 \tag{3.1}$$

and using this in a gauss distribution

$$w(\eta) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(\frac{-\eta^2}{2\sigma^2}\right) \tag{3.2}$$

Thi is of course just the circle equation with c_x, c_y being the center of the circle, x, y are the data points and r the radius. So f a lot of the data points have the same distance r from the circle center there will be a high score for this particular radius. The index forthe highest score can then be used to find the corresponding radius.

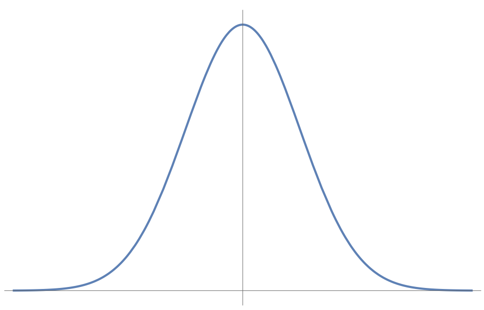


FIG. 3.1 – Using the probability density function of the normal distribution to calculate the score of a point in order to have a well defined maximum if a point lies directly on the circle and $\eta(r) = 0$.

```
DMENSION = 1001
r= linspace(0,1,DIMENSION)
for c in centers:
  scores = zeros(DIMENSION)
  for x,y in allPoints:
    s = 2*BIN_WIDTH
    eta = (c_x-x)**2 + (c_y-y)**2 - r**2
    scores += 1. / ( sqrt( 2 * Pi ) * s )
               * exp( -( eta ** 2 )
                / ( 2 * s ** 2 ) )

  index = max(scores)
  circle = {}
  circle['center'] = c
  circle['radius'] = r[index]
```

FIG. 3.2 – Pseudo code for the 1D Hough Transform. r is an array of length 1001 so η will also be an array of length 1001. Scores is where the score for each iteration is stored. For each point the score is computed and added to the scores array and at the end the index with the highest score is the index we need to get the radius.

Runtime

The runtime of this algorithm is of $\mathcal{O}(n)$ where n is the dimension of the radius array.

3.1.2 2D: Known Radius - Find Center

Now the radius is known and the x and y coordinate of the center (c_x, c_y) are unknown. Now instead of a one dimension the accumulator is 2 dimensional. The range of that space is simply the dimension of the plane which for this thesis is generally $[-0.5, 0.5]$. Size of the bins is 0.001. So if the plane was 1 m wide the accuracy of the accumulator is up to 1 mm. As in the one dimensional case we use the scoring function 3.1 in combination with the weight function 3.2.

Runtime

The runtime of this algorithm is $\mathcal{O}(n^2)$ where n is the dimension of the histogram. The calculation of the weight has to be done for each data point of the 2D histogram. So in a 1000×1000 histogram with 400 data points we calculate 400'000'000 times the weight of a grid point. Reducing the dimensions of the histogram weakens the accuracy of the whole algorithm but can speed up the calculations considerably. With a 1000×1000 histogram the resolution in each space dimension is 1 mm. The RICH Technical Design Report states the resolution of the HPD is 2.5 mm $\times$ 2.5 mm.

The need (not entirely true) to calculate the weight for each grid point and data point means that there is a loop over data points and two loops for the x and y coordinate of the grid. To improve upon that there is the possibility of array broadcasting.

Array broadcasting

Consider following one dimensional arrays where x is a 1D histogram binning entries from 1 to 4 and same for y.

$$x = [1,2,3,4]$$
$$y = [1,2,3,4]$$

Now all combinations between an element of x and y represent a 2D grid $((1,1),(1,2),...)$. So to iterate through all those grid points one would have to create 2 for-loops iterating through x and y

```
def funcB():
  a = np.random.randn(100)
  b = np.random.randn(100)
  for x in a:
    for y in b:
      print (1-x)^2 + (2-y)^2 - 9
```

This is not only slow but also doesn't look too nice if there are even more loops. Broadcasting now turns this one dimensional array of length n into an n by n ma-

```
  DIMENSION = 1001
  xbins = linspace(-0.5,0.5,DIMENSION)
  ybins = linspace(-0.5,0.5,DIMENSION)
  x, y = broadcast_arrays( xbins[..., newaxis],
                           ybins[newaxis,...] )

  for r in Radiuses:
    weights = zeros( (DIMENSION,DIMENSION) )
    for xd,yd in allPoints:
      s = 2*BIN_WIDTH
      eta = (xd-x)**2 + (yd-y)**2 - r**2
      weights += 1. / ( sqrt( 2 * pi ) * s )
                 * exp( -( eta ** 2 )
                 / ( 2 * s ** 2 ) )
    i, j = argmax(weights)
    removeUsedPoints()
    circle['Center'] = (xbins[i], ybins[j])
    circle['Radius'] = r
```

FIG. 3.3 – Pseudo code for the 2D Hough Transform. xbins and ybins are arrays of length 1001. Here we use array broadcasting in order to avoid for loops and the weights can be evaluated in one line. This means that the x and y variables have dimension (1001,1001) but they don't take up that much memory. The x variable for example just broadcasts its value from the first row down to all the other rows and for y it broadcasts the first column to all the other columns. Weights variable is a 1001 by 1001 matrix. Again the entry with the highest score is the candidate for a possible circle center and if found stores in a final variable called circle.

trix

$$x = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}$$

and

$$y = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

And with this the loops can be omitted:

```
def funcA():
  a = np.random.randn(100)
  b = np.random.randn(100)
  x,y = np.broadcast_arrays(a[...,np.newaxis],
                            b[np.newaxis,...])
  print (1-x)^2 + (2-y)^2 - 9
```

In this case this prints a 4 by 4 array with the function evualted for each combination of entries of x and y

$$\begin{bmatrix} -8 & -9 & -8 & -5 \\ -7 & -8 & -7 & -4 \\ -4 & -5 & -4 & -1 \\ 1 & 0 & 1 & 4 \end{bmatrix}$$

A runtime comparison shows

```
In [3]: %timeit funcA()
10000 loops, best of 3: 76.8 us per loop

In [4]: %timeit funcB()
100 loops, best of 3: 7.99 ms per loop
```

So the version with broadcasting is 100 times than the double loop. And the memory consumption is moderate since they broadcasted entries aren't new memory locations but just refer to the initial array.

Optimizations

It was mentioned before that for each data point the weight for the whole grid has to be calculated, that is not true. In the 2D case each grid point is a potential center for a circle so if a grid point is further away than a maximum radius that is a constraint given by physics this calculation could be skipped. This could probably be done even smarter with the use of some mesh so only points in the surrounding face of the mesh are considered.

3.1.3 Simple Example of 2 Circles Without Noise

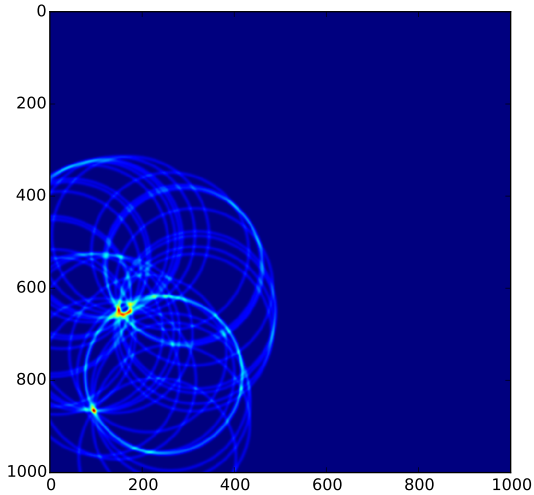


FIG. 3.4 – 2D weight matrix in the first iteration of the Hough Transform algorithm. The circles have similar radiuses (0.170 and 0.158) which explains one clear maximum in the bottom left and a smeared one a bit to the top right.

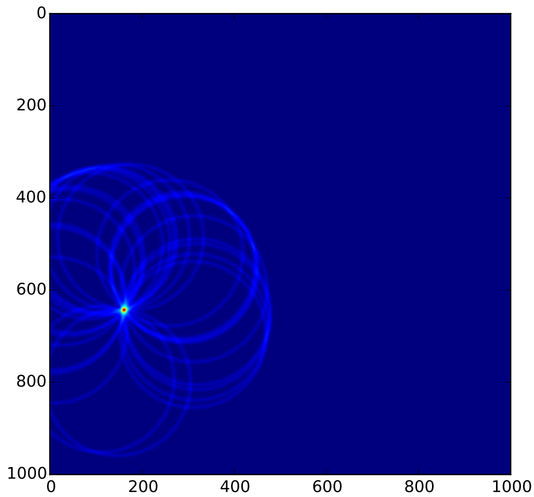


FIG. 3.5 – 2D weight matrix in the second iteration. Points that satisfied the condition being less than a certain ϵ away from the radius found in the first iteration are removed leaving (hopefully) only points available that belong to the second circles.

3.1.4 3D: Nothing is Known - Find Everything

3.2 Combinatorial approach

A circle is uniquely defined by 3 points and radius and center can be calculated. If there are 15 points lying on the same circle there are 455 possible combinations of triplets. Calculating the center and radius for these 455 triplets should result in the same center and same radius for all the triplets (floating point inaccuracy not considered). Having one background hit in addition to the 15 circle hits increases the triplet number to 560. The triplets with points solely consisting of points on the circle still have the same center and radius but the new combinations that now include a background hit will vary and it is unlikely that any two triplets including the background point will have the same center and radius. Here is an overview of the algorithm used for this thesis.

1. Build all possible triples of points given the data points
2. For all the point triples calculate the center and the radius of the potential circle
3. Due to constraints in the radius many of the circles with a radius bigger than a certain threshold will be dropped.
4. Create a histogram with the radius distribution. Peaks in the radius distribution hint to a circle.
5. Scan the radius histogram for peaks and look at the center point histogram for the given radius of a peak. If there is also a peak in the center point histogram the set of the points of the triples lie on a circle with a radius and center given by the histogram peaks.

3.2.1 Generating the triples

For generating the triples the built-in function `itertools.combinations()` of python is used. It takes a list as input

3.2.2 Calculating the Circle given 3 points

Let (A, B, C) be a triple of points in a 2D plane and a, b, c the length of the sides opposite to the respective corner. The semiperimeter is defined as

$$s = \frac{a+b+c}{2} \tag{3.3}$$

using this we can calculate the radius R of the circumcircle of triangle $\overline{ABC}$:

$$R = \frac{abc}{4\sqrt{s(a+b-s)(a+c-s)(b+c-s)}} \tag{3.4}$$

We have $\lambda_1, \lambda_2, \lambda_3$ as the barycenteric coordinates of the circumcenter:

$$\lambda_1 = a^2 \cdot (b^2 + c^2 - a^2) \tag{3.5}$$

$$\lambda_2 = b^2 \cdot (a^2 + c^2 - b^2) \tag{3.6}$$

$$\lambda_3 = c^2 \cdot (a^2 + b^2 - c^2) \tag{3.7}$$

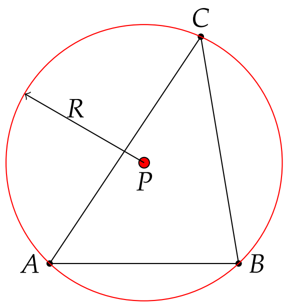


FIG. 3.6 – The circumradius (R) and the circumcenter (P) of a circle defined by three points.

Multiplying a matrix consisting of the column vectors of A, B, C with a column vector of $\lambda_1, \lambda_2, \lambda_3$ and dividing the resulting vector by the sum of the barycentric coordinates (for noramlization) leads to the circumcenter of the triangle $\overline{ABC}$

$$\begin{pmatrix} A_x & B_x & C_x \\ A_y & B_y & C_y \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = \boldsymbol{P'} \tag{3.8}$$

$$\frac{\boldsymbol{P'}}{\lambda_1 + \lambda_2 + \lambda_3} = \boldsymbol{P} \tag{3.9}$$

3.2.3 Drawback

There is also a drawback with this method: the combinatorics blow up with a high number of data points $\binom{N}{3}$. So for example with 200 data points (circle data and background) the number of triplets is

$$\binom{200}{3} = 1313400$$

and for 300:

$$\binom{300}{3} = 4455100$$

So the runtime of the algorithm is roughly in the order of $\mathcal{O}(N^3)$ which can be easily seen when taking the upper bound of $\binom{N}{k} \leq \frac{N^k}{k!}$ and since $k = 3$ then means $\frac{N^3}{3!}$ see figure 3.7.

Improvement of speed

As seen before this approach scales with $\binom{N}{3}$. This means that with 500 data points we have to create around 20 million triplets which are used for calculating radiuses and centers of circles. To qualify as a circle the algorithm needs a threshold to decide if a candidate is a circle or not. If a circle has 8 data points then there are 120 different ways to combine 3 different points with each other to get the same circle. With this a threshold can be defined for finding circles. This of course means that

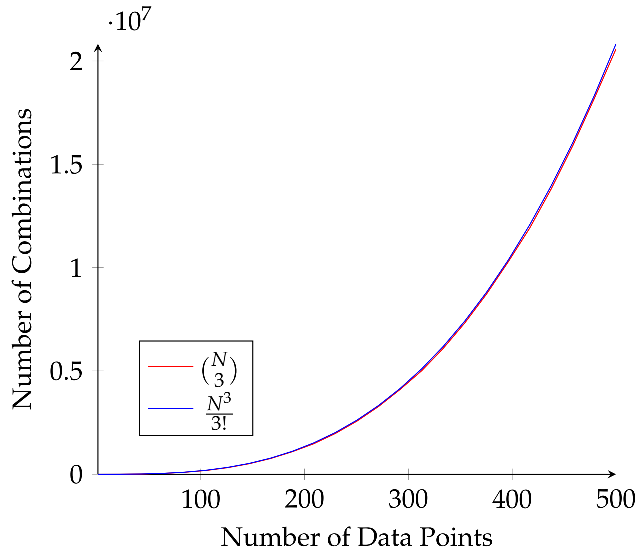


FIG. 3.7 – Scaling of the algorithm. Binomial Growth with $\binom{N}{3}$ compared with the approximation $\frac{N^3}{3!}$. So the algorithm runs in the order of $\mathcal{O}(n^3)$

circles with less than 8 points will normally never be found unless another point that actually doesn't belong to the circle lies on the circle and contributes to the radius and center histogram pushing the circle above the threshold.

Now one way to improve the speed of the algorithm is splitting the original data set randomly into two lists. For each of these lists all the possible combinations of triplets is generated again and combined in one total list.

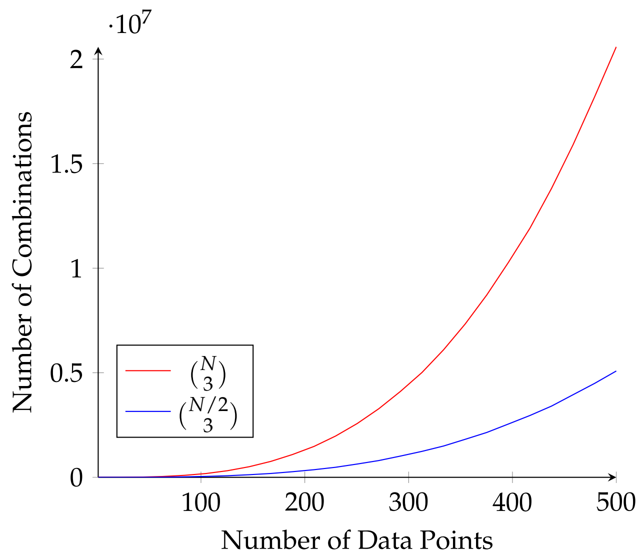


FIG. 3.8 – Number of combinations with $\binom{N}{3}$ compared to the number of combinations generated from $\binom{N/2}{3}$

3.2.4 Possible Optimisation: Average Radius of Random Circles In a Unit Square

An interesting property of calculating the radius of triplets generated from points that are distributed uniformely is that they always obey a certain shape.

First we calculate the expected area of a triangle formed by three points randomly chosen from the unit square. Let $A = (a_1, a_2)$, $B = (b_1, b_2)$, $C = (c_1, c_2)$ be the vertices of the random triangle T. We consider the case where $a_2 > b_2 > c_2$ which takes $\frac{1}{6}$ of the total "Volume". Fix a_2, b_2, c_2 for the moment and we can write.

$$b_2 = (1-t)a_2 + tc_2, \qquad 0 \leq t \leq 1.$$

The side AC of T intersects the horizontal level $y = b_2$ at the point $S = (s, b_2)$ with

$$s = s(a_1, c_1, t) = (1-t)a_1 + tc_1 \tag{3.10}$$

The area X of T is then given by

$$X = \frac{1}{2}|b_1 - s|(c_2 - a_2)$$

We now start integrating with respect to our six variables. The innermost integral is with respect to b_1 and gives

$$\begin{aligned} X_1 &:= \int_0^1 X \mathrm{d}b_1 = \frac{1}{2}(c_2 - a_2)\left(\int_0^s (s - b_1)db_1 + \int_s^1 (b_1 - s)db_1\right) \\ &= \frac{1}{4}(c_2 - a_2)(1 + 2s + 2s^2) \end{aligned}$$

Next we integrate over b_2:

$$X_2 := \frac{1}{4}\int_0^1 \int_{a_2}^1 (c_2 - a_2)^2\, dc_2\, da_2 \times \int_0^1 \int_0^1 \int_0^1 (1 - 2s + 2s^2)\, dt\, dc_1\, da_1$$

This gives

$$X_3 = \frac{1}{4} \cdot \frac{1}{12} \cdot \frac{11}{18} = \frac{11}{6 \cdot 144}$$

But generalizing our assumption at the beginning $a_2 < b_2 < c_2$ we multiply this result by 6 and obtain then $\frac{11}{144}$.

4 Results

5 Conclusions

Bibliography

[1] LHCb Collaboration. *The LHCb Detector*. Online. URL: http://lhcb-public.web.cern.ch/lhcb-public/en/Detector/Detector-en.html.

[2] Richard O. Duda and Peter E. Hart. "Use of the Hough Transformation to Detect Lines and Curves in Pictures". In: *Commun. ACM* 15.1 (Jan. 1972), pp. 11–15. ISSN: 0001-0782. DOI: 10.1145/361237.361242. URL: http://doi.acm.org/10.1145/361237.361242.

[3] RICH LHCb. "Technical Design Report". In: *CERN/LHCC* 37 (2000), p. 2000.

[4] A Powell. *Particle identification at LHCb*. Tech. rep. 2011.