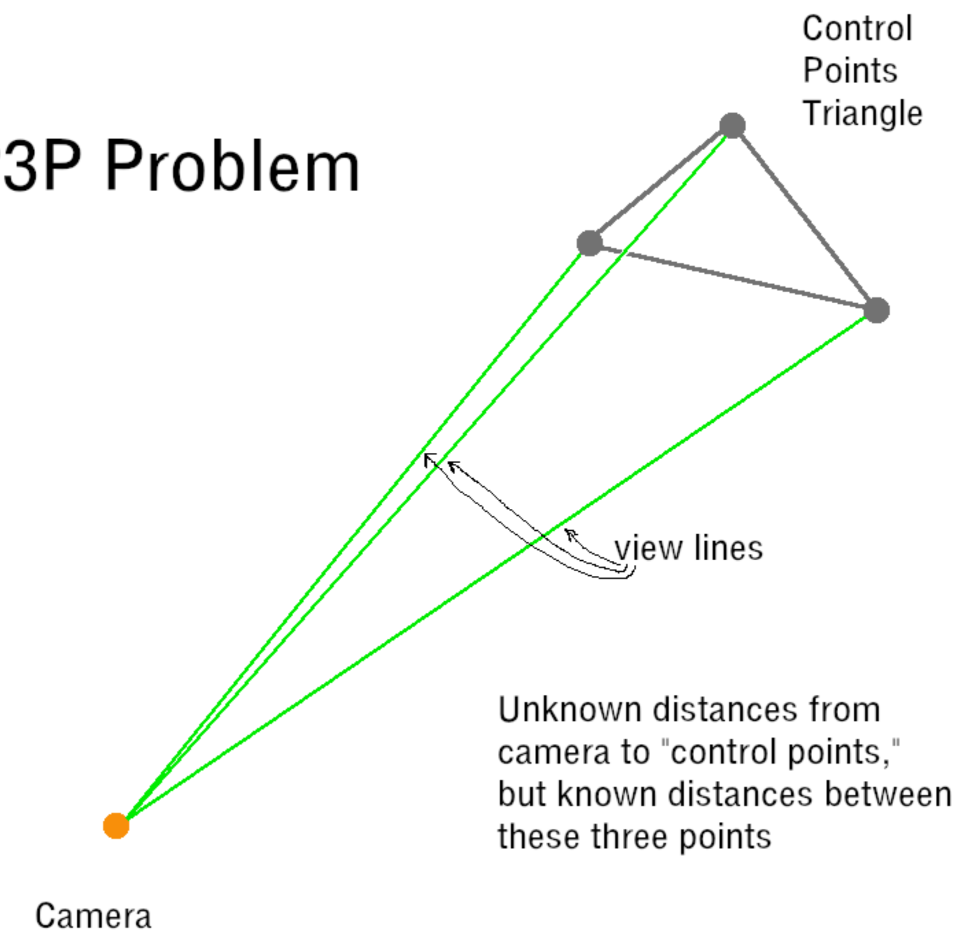
Elliptic Curves and the Perspective 3-Point Problem

A talk for the 2019 MAA Iowa Section Meeting

Michael Q. Rieck, Ph. D.
Drake University

- Soon after the invention of the camera, a mathematician named Johann August Grunert posed and solved a mathematical problem that became foundational to the area of "photogrammetry," and later of use in the area of "camera tracking"
- The problem has come to be known as the "Perspective 3-Point Pose Problem," or the "Perspective 3-Point Problem," or just the "P3P Problem"
- The idea is simple: given a photograph that contains the images of three known points in space, is it possible to determine the location of the camera?
- The three known points have come to be called the "control points," and of course, the distances between these points is known
- However, the ***distances from the camera to the control points are unknown***, and determining these is the tricky part of the P3P problem
- The triangle whose vertices are the control points is called the "control points triangle," and the plane containing it is called the "control points plane"
- A line connecting the camera position to a control point is called a "view line"

P3P Problem


Control
Points
Triangle
view lines
Unknown distances from
camera to "control points,"
but known distances between
these three points
Camera

- It is assumed that the ***angles between pairs of view lines are known***
- This is reasonable when dealing with a simple "pinhole camera" model
- Using the Law of Cosines, Grunert approached the P3P Problem as a system of three quadratic equations in the three unknown distances
- His solution, as well as a number of later solutions, reduces the system to a single polynomial equation in just one unknown, with coefficients that depend on the known parameters of the P3P Problem (triangle side lengths, etc.)
- In Grunert's case, the unknown is the square of the ratio of two of the unknown distances, and the polynomial has degree four, *i.e.* a ***quartic polynomial***
- Since a quartic polynomial can have up to four positive real roots, for given values of the P3P Problem parameters, there may be up to four solutions to the P3P Problem, *i.e.,* up to ***four possible positions of the camera***

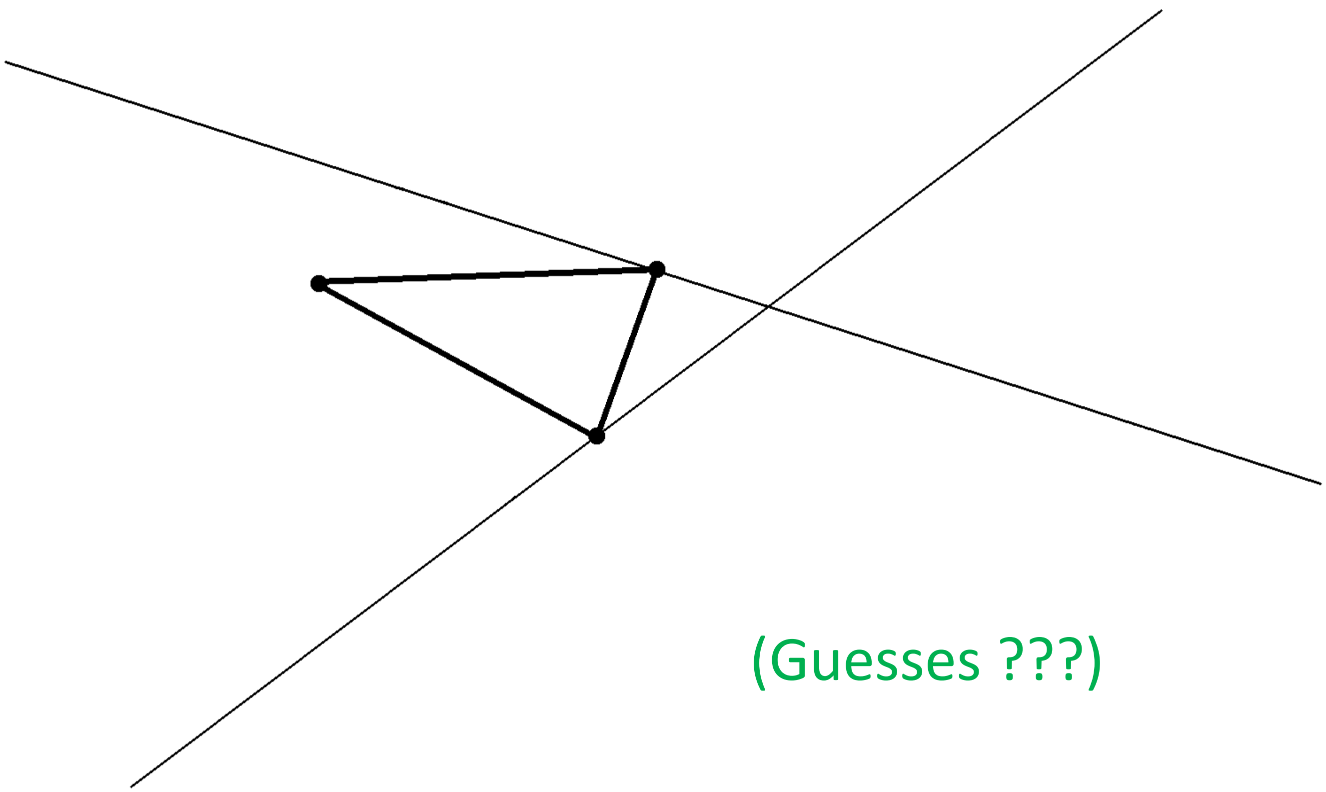
- In the 20th century, the P3P Problem was shown to be equivalent to finding the intersection of two quadratic curves in the real projective plane
- This observation has been exploited in the solution methods of Finsterwalder, Grafarend, and more recently, Persson and Norberg
- Wikipedia claims that the method of Persson and Norberg, which is known as "Lambda Twist," "achieved state of the art performance in 2018 with a 50 fold increase in speed and a 400 fold decrease in numerical failures"
- I was recently able to recast the P3P Problem as an equivalent problem involving ***four great circles*** on the unit sphere, $x^2 + y^2 + z^2 = 1$
- This led me to develop a method that focuses first of determining the ***directions of the sidelines*** of the control points triangle, rather than the distance from the camera to the control points

- My method is noticeably more accurate than Lambda Twist, by ***at least one order of magnitude***, in most cases, but it is about 60% slower
- I implemented C code for testing both methods, which is available at

    `github.com/mqrieck/EllipticCurveP3P`

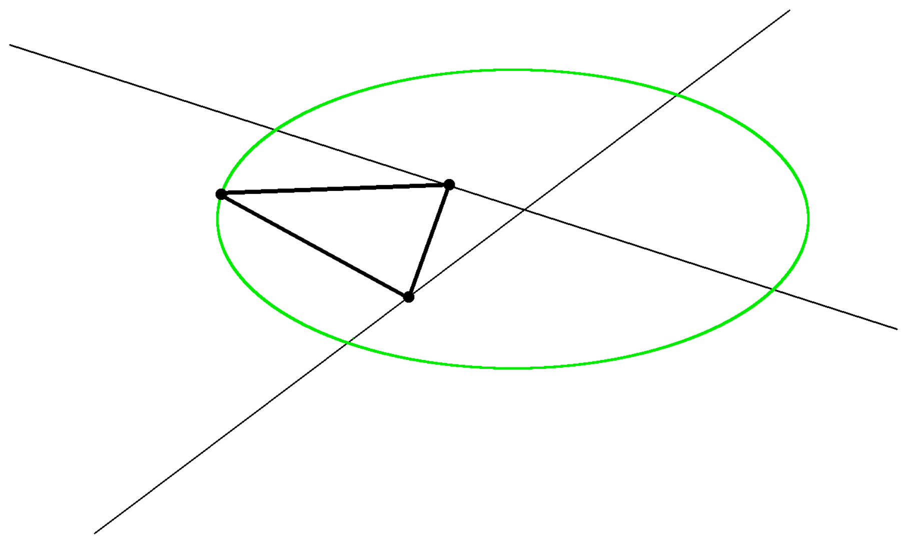
- My student, Pawel Barnas, was helpful in developing test programs in Python that tested the C code I wrote; he ran tests and collected data too
- The spherical interpretation of the problem is rather interesting in its own right, and my approach to solving it relies on a spherical analogue of a classical “slide puzzle” in the plane
- Let us first look at this classical problem

- *Problem:* Assuming that a triangle is allowed to move around on the plane, but with two of its vertices constrained to stay on two different non-parallel lines, describe the curve followed by the third vertex

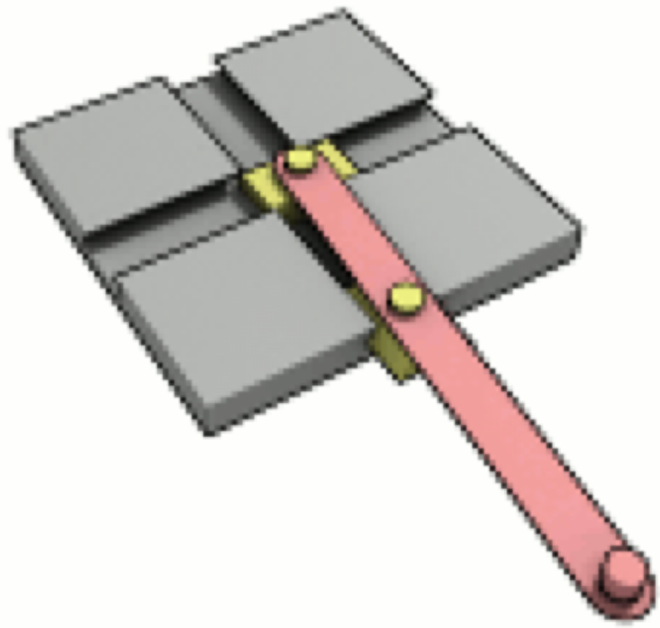


(Guesses ???)

- *Solution:* The third point will trace out an ellipse



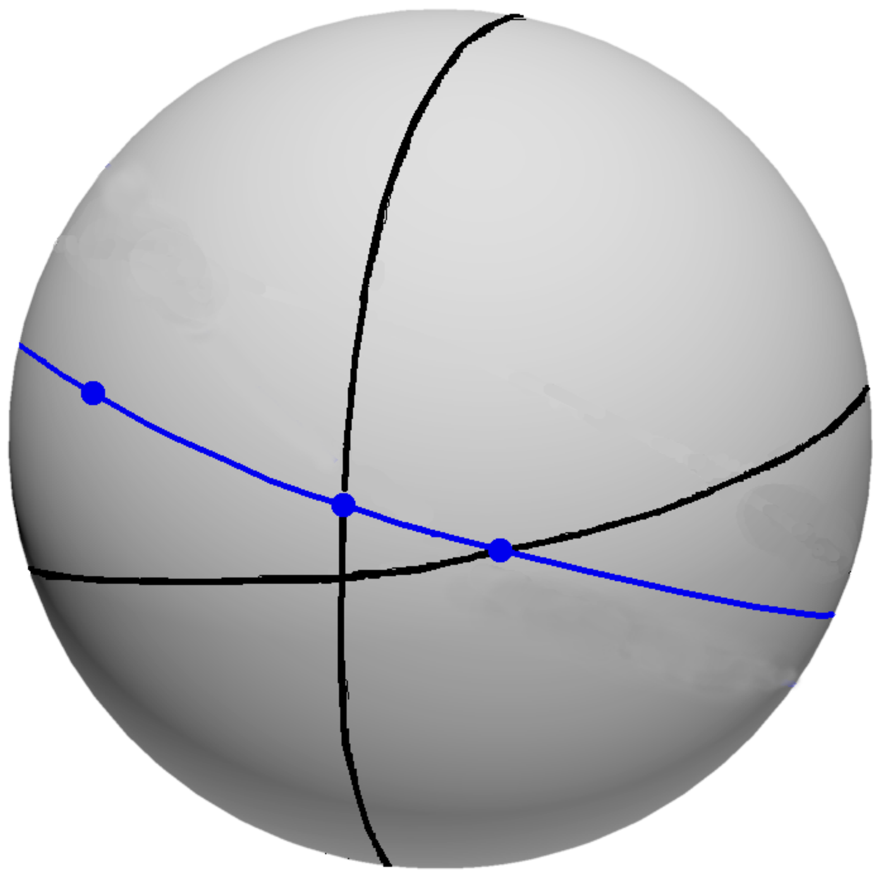
- Even if the triangle is degenerate (interior angles now being 0 or π), so we are now basically sliding a line segment, the curve traced will still be an ellipse
- This is the basis of an ellipse-drawing tool known as the "Trammel of Archimedes"



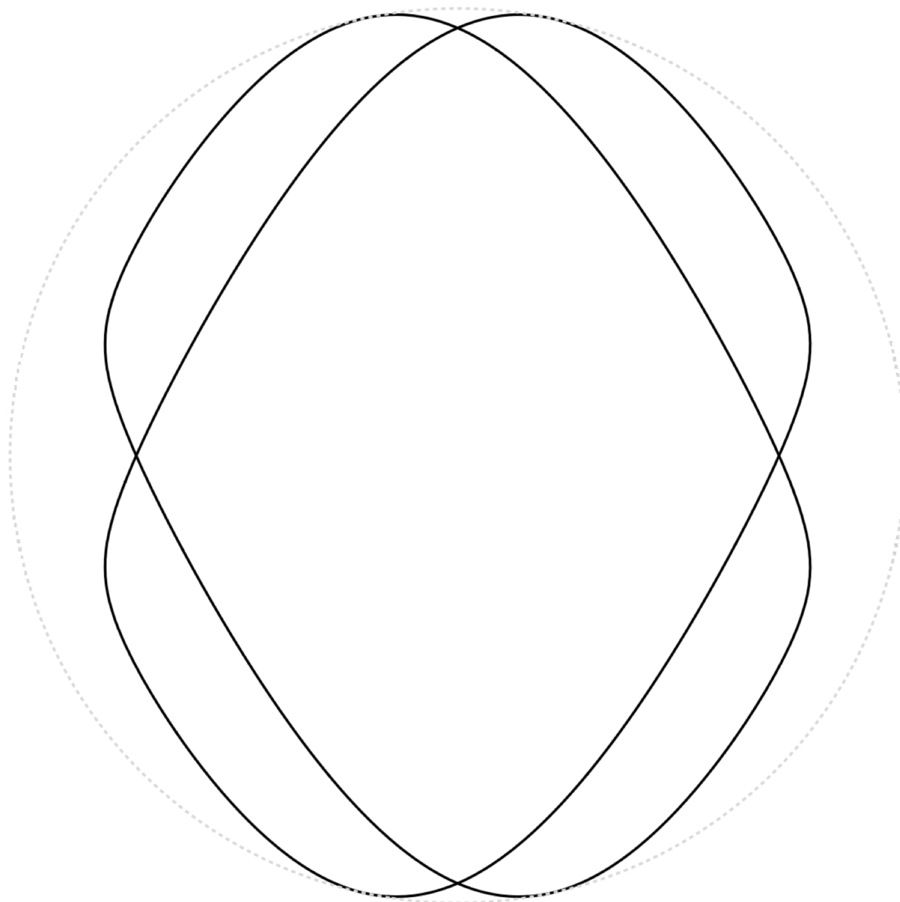
https://en.wikipedia.org/wiki/Trammel_of_Archimedes

- In my spherical analogue, lines are replaced with great circles, and line segments are replaced with arcs of great circles
- We will only need to be concerned with the spherical analogue of the triangle-sliding problem in the special case where the triangle is degenerate
- Thus, the three moving points will lie on a moving great circle, but these points must maintain their separation distances from each other, as this great circle moves around the sphere
- Two of these three points are also constrained to lie on fixed great circles
- Again, we are interested in the path traced out by the third point while sliding the moving circle, subject to the two constraints
- The resulting curve will no longer be described by a quadratic equation, but rather, it requires a quartic (fourth degree) equation
- The next figure shows the setup, but does not show the curve traced by the third moving point

A Sliding Problem on the Sphere



Here is the projection onto a plane of two of the possible quartic curves on the sphere:



Without loss of generality, we can rotate the sphere and arrange for the quartic equation to only involve the *x* and *y* coordinates (not *z*) and some constant parameters that will not be explained here (unless asked):

$$
\begin{aligned}
& [1-(\mu_0^2-\nu_0^2)^2]\,[\nu_0^2\,x^2-\mu_0^2\,y^2]^2 \\
+ \quad & 4\,(1-\alpha_1^2-\alpha_2^2)\,\mu_0^2\,\nu_0^2\,(\mu_0^2-\nu_0^2)(\nu_0^2\,x^2-\mu_0^2\,y^2) \\
- \quad & 4\,(\alpha_1^2+\alpha_2^2)\,\mu_0^2\,\nu_0^2\,(\nu_0^2\,x^2+\mu_0^2\,y^2) \\
+ \quad & 8\,(\alpha_2^2-\alpha_1^2)\,\mu_0^3\,\nu_0^3\,x\,y \\
- \quad & 4[1-(\alpha_1+\alpha_2)^2][1-(\alpha_1-\alpha_2)^2]\,\mu_0^4\,\nu_0^4 \;=\; 0.
\end{aligned}
$$

- The sliding great circle problem is then easily translated *from* a problem on the unit sphere *to* a problem in the real projective plane
- Here we use the standard model of the real projective plane, identifying its "points" with the lines through the origin in 3-dimensional Cartesian space
- Each such line intersects the unit sphere in two antipodal points, and so we may also identify such a pair of points on the sphere with a single "point" in the projective plane
- Under this identification, great circles on the unit sphere are identified with "lines" in the projective plane
- The quartic curve on the sphere that resulted from the sliding great circle problem then corresponds to a quartic curve in the projective plane
- It can be shown that this curve has two singularities, where the curve is self-crossing
- It can also be shown that it has "genus one" (definition omitted unless asked)

- Furthermore, by making a suitable projective transformation, an affine version of the resulting curve has an equation in the following form:

$$a^2x^2y^2 - b^2x^2 - c^2y^2 + d^2 - 2e^2xy = 0$$

- Via a simple birational transformation, the above can then be put into the Legendre form $\xi^2 = (1 - \omega^2)(1 - \kappa^2\omega^2)$, where ξ and ω are variables, and κ is a constant
- This is a classical "elliptic curve"
- Specifically, take

$\delta = (ad+bc+e^2)\,(ad+bc-e^2)\,(ad-bc+e^2)\,(ad-bc-e^2),$

$\rho = (a^2d^2 + b^2c^2 - e^4 + \delta^{1/2}) / 2a^2b^2,$

$\kappa = ab\,\rho / cd,$

$\omega = x / \rho^{1/2},$

$\xi = [\,(a^2x^2 - c^2)y - e^2x\,] / cd$

- The family of quartic curves described at the top of the previously slide also includes curves used in cryptography such as "Edwards curves"
- The *j*-invariant, an important and well-known invariant of isomorphic elliptic curves, can be computed for the curve on the previous slide:

$$j = \frac{16(a^4b^4\rho^4 + 14a^2b^2c^2d^2\rho^2 + c^4d^4)^3}{a^2b^2c^2d^2\,\rho^2\,(a^2b^2\rho^2 - c^2d^2)^4}$$

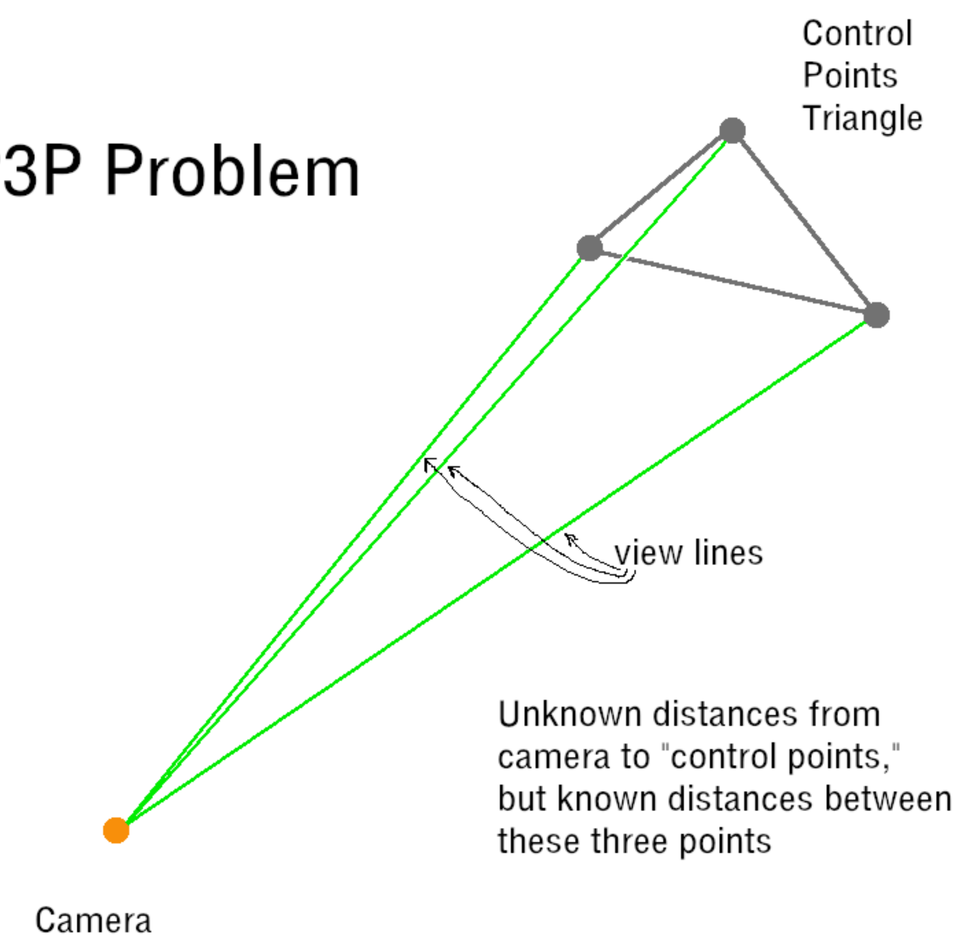
$$= \frac{16(a^4d^4 + b^4c^4 + e^8 + 14a^2b^2c^2d^2 - 2a^2d^2e^2 - 2b^2c^2e^4)^3}{a^2b^2c^2d^2\,\delta^2}$$

- The terminology “elliptic curve” is a little strange, and inconsistently used in the literature
- *It is **not** an ellipse!* (The name is an unfortunate historical accident)
- The most generous definition is an algebraic curve of genus one
- The family of curves considered here fits this definition
- In any case, these curves are birationally equivalent to non-singular cubic curves
- Such non-singular cubic curves constitute a more restricted class of curves that is often used to give a more restricted definition for “elliptic curves”

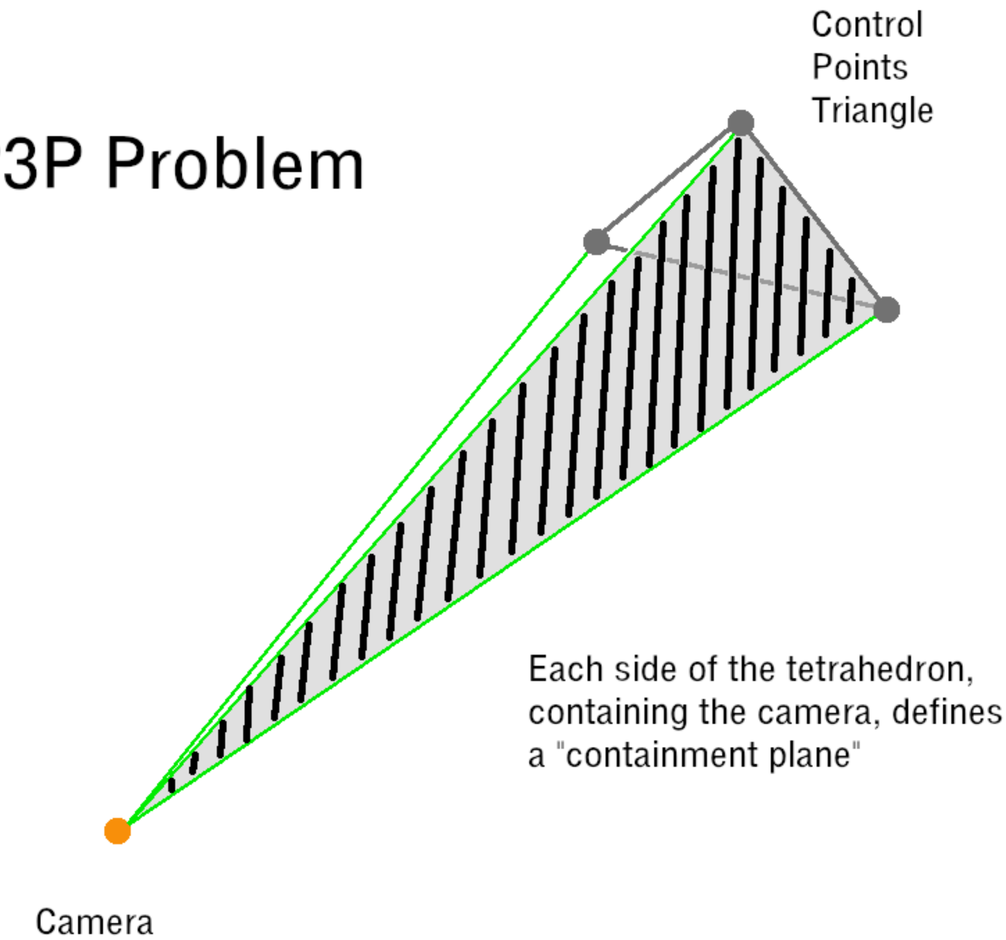
- There are even more restricted definitions
- Often one is also required to specify a “rational point” on the curve, in order to impose an abelian group structure on the curve
- Such additional structure will not be of concern here, though perhaps it should be!
- Let us now return to the P3P problem, and work with a Cartesian coordinate system that places the camera at the origin, and that agrees with the actual distances

- Locating the control points in this “camera-centric” coordinate system is sufficient for solving the P3P problem, where the determination of the camera’s position in physical space is the objective
- The essential thing is to find the distances between the camera and the control points, after which, everything becomes simple
- So, working in the camera-centric coordinate system, assume, *w.l.o.g.*, that the view lines are known, *a priori*
- It will be helpful to refer to any plane containing two view lines as a “containment plane,” and these are then also presumed to be known

P3P Problem

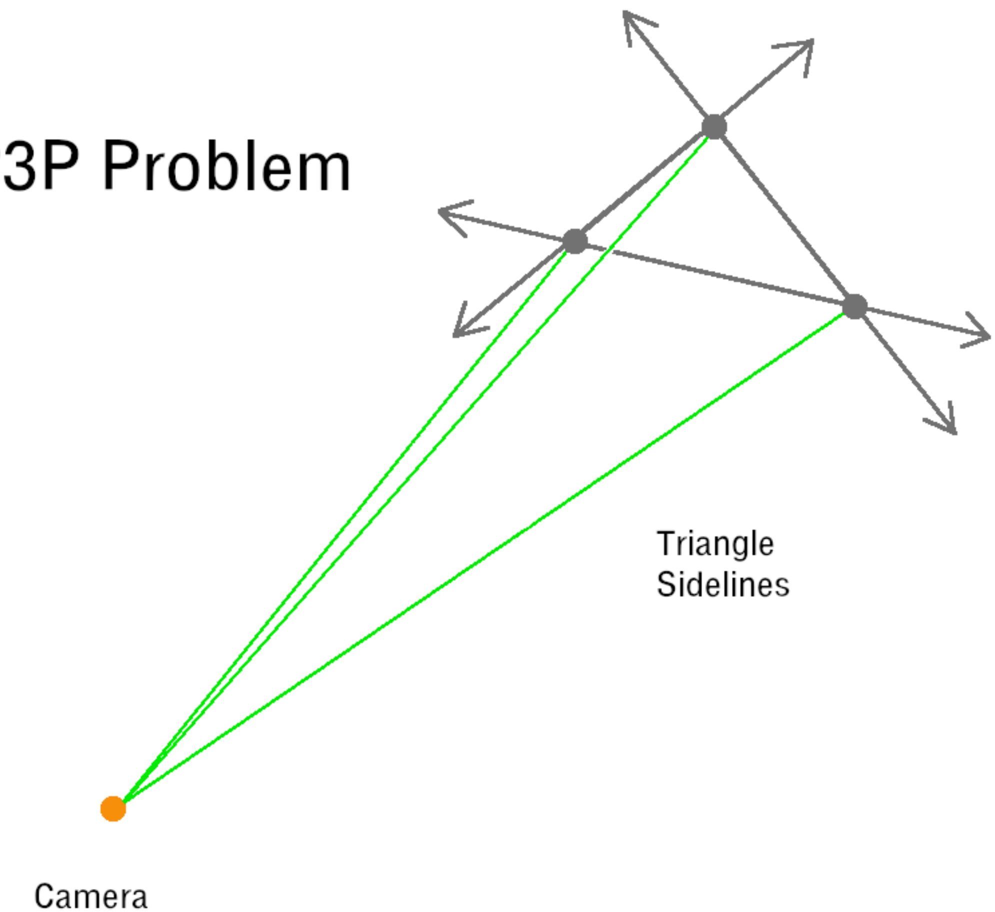

Control
Points
Triangle
view lines
Unknown distances from camera to "control points," but known distances between these three points
Camera

P3P Problem

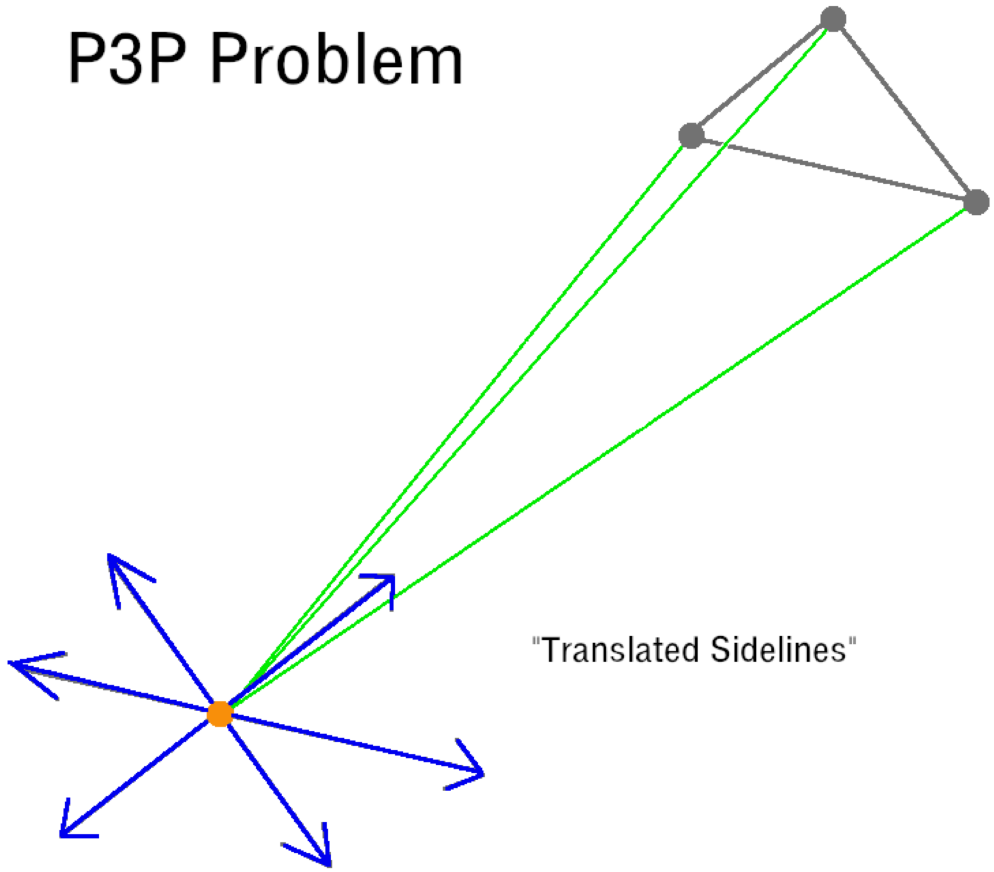

Control
Points
Triangle
Each side of the tetrahedron,
containing the camera, defines
a "containment plane"
Camera

- Of course, the distances between the control points are known, and so we may assume that the interior angles of the control points triangle are also known
- Although the positions of the control points are unknown, we can imagine parallel translating the three sidelines of the triangle, so that the translated lines become incident with the origin (the camera position)
- Accordingly, these three unknown lines will be called the “translated sidelines”

P3P Problem



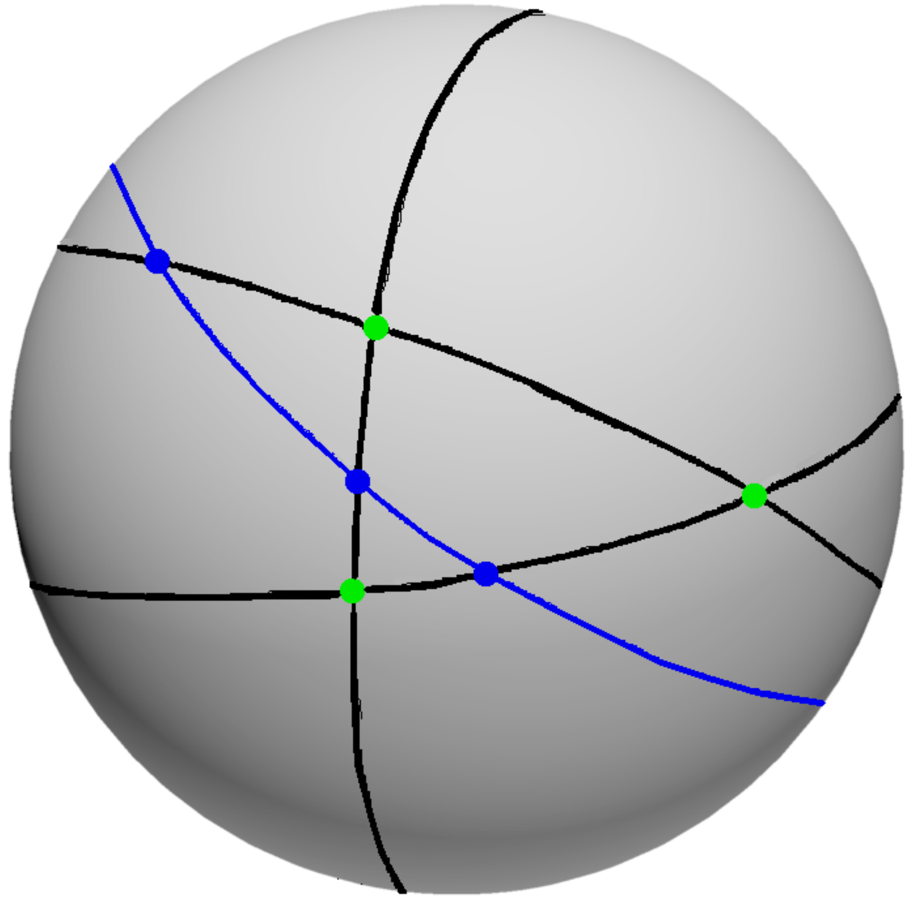
P3P Problem



- Restrict attention now to lines and planes that go through the origin
- We have the three known *view lines*
- We have the three unknown *translated sidelines*
- We have the three known *containment planes*
- Each containment plane contains two view lines and one translated sideline
- We also have the unknown plane containing the translated sidelines
- Now, notice what happens when these geometric objects are intersected with the unit sphere

- Each view line corresponds to a pair of known antipodal points on the sphere (*green points* in the upcoming slides)
- Any two of these pairs gives four points lying on the known great circle that corresponds to a known containment plane (*black circles* in upcoming slides)
- Each translated sideline corresponds to a pair of unknown antipodal points on the sphere (*blue points* in upcoming slides)
- These points, for all three translated sidelines, lie on an unknown great circle (*blue circle*) that corresponds to the plane containing the three translated triangle sidelines

Spherical Interpretation of the P3P Problem

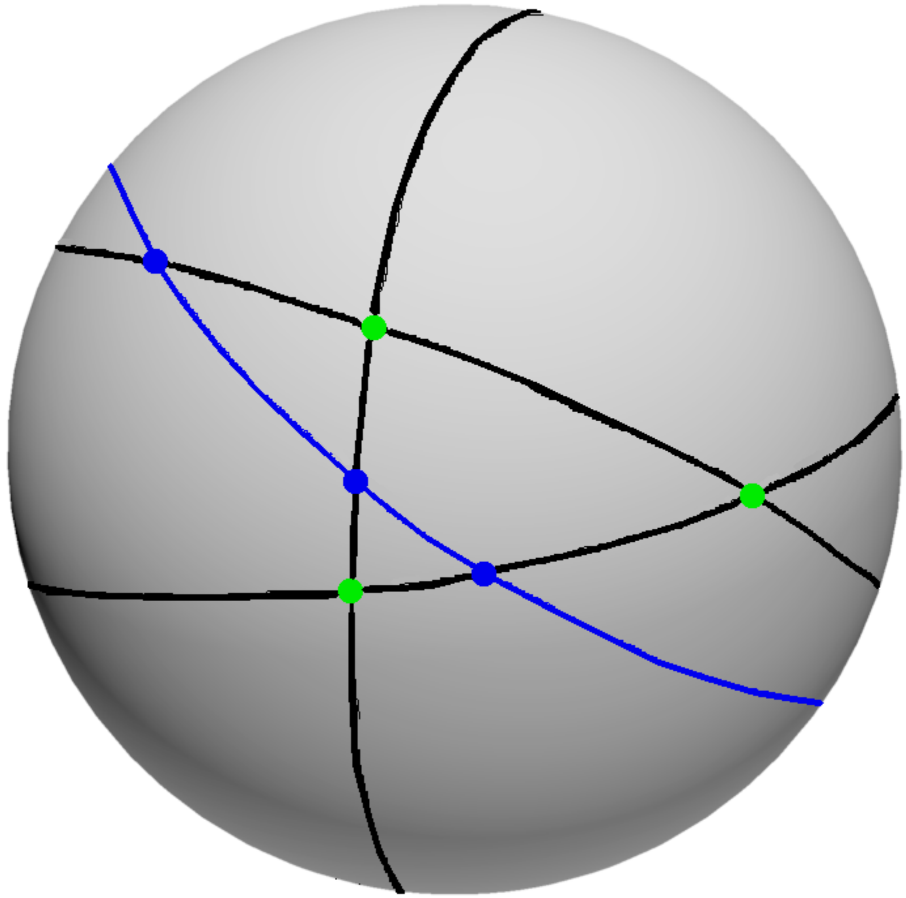


Green points correspond to view lines;
Blue points correspond to translated sidelines

- A fact that might not be obvious at first is that the distances between the blue points (corresponding to the translated sidelines) along the great circle containing these points, are actually known
- *Why?*
- These distances are just the angles between the sidelines of the control points triangle, *i. e.* the interior/exterior angles of this triangle
- The upshot of all this is that the P3P problem can now be ***reformulated*** as follows

- Given three fixed great circles on the unit sphere (the black circles), find three pairs of antipodal points (the blue dots) such that
  - each pair of points lies on a corresponding fixed great circle
  - all of these points also lie on a common great circle (the blue circle)
  - the distances between these points on this latter great circle have certain prescribed values

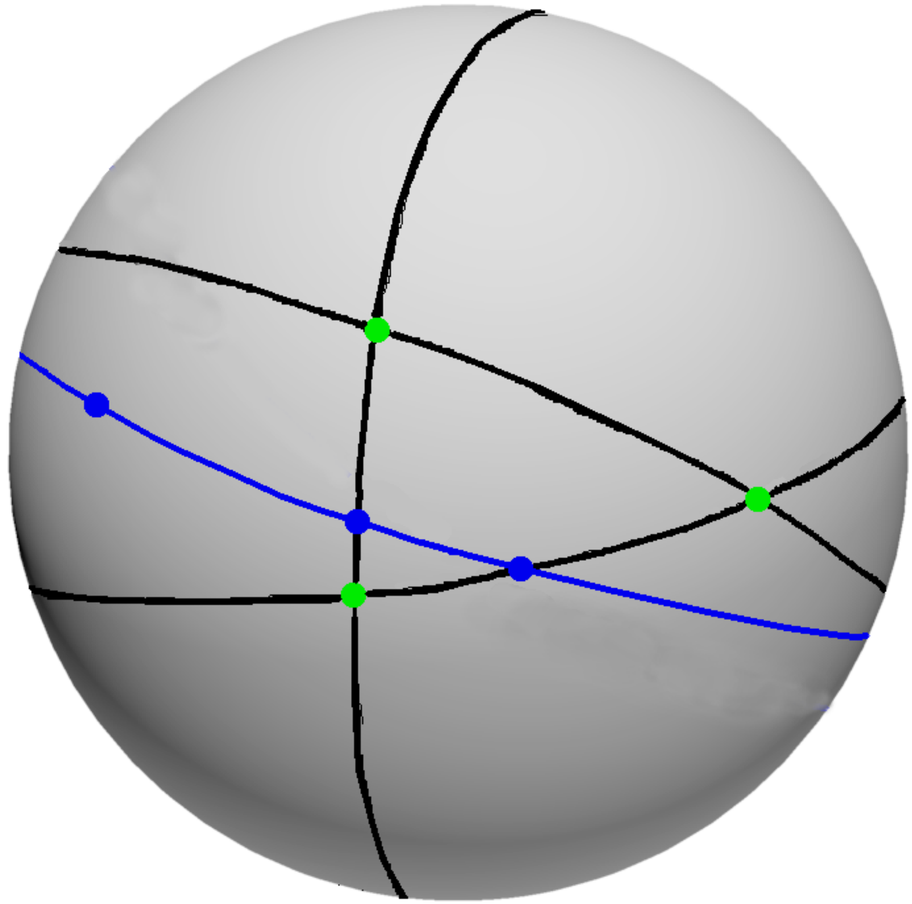
Spherical Interpretation of the P3P Problem



Green points correspond to view lines;
Blue points correspond to translated sidelines

- Solving this means identifying possible translated sidelines (for some P3P solution) and therefore also identifying the plane containing all of them
- Such a plane is parallel to the plane containing the possible positions of the control points (for some P3P solution)
- By solving a linear equation, it becomes easy to locate this latter plane, and thereby obtain one of the solutions to the P3P problem
- Now, let us go back to solving the problem on the unit sphere
- If we relax one of the constraints in the problem, by no longer requiring the third pair of antipodal points to lie on the third fixed great circle, then the (blue) great circle that contains the six (blue) points becomes free to slide around

- Indeed, we are now looking at the sliding problem considered earlier, where the path of one of the points in the third pair of antipodal points was the issue
- As already asserted, this path describes a certain quartic curve
- So, we simply need to ask where this quartic curve intersects the third fixed great circle, and thereby solve the P3P problem



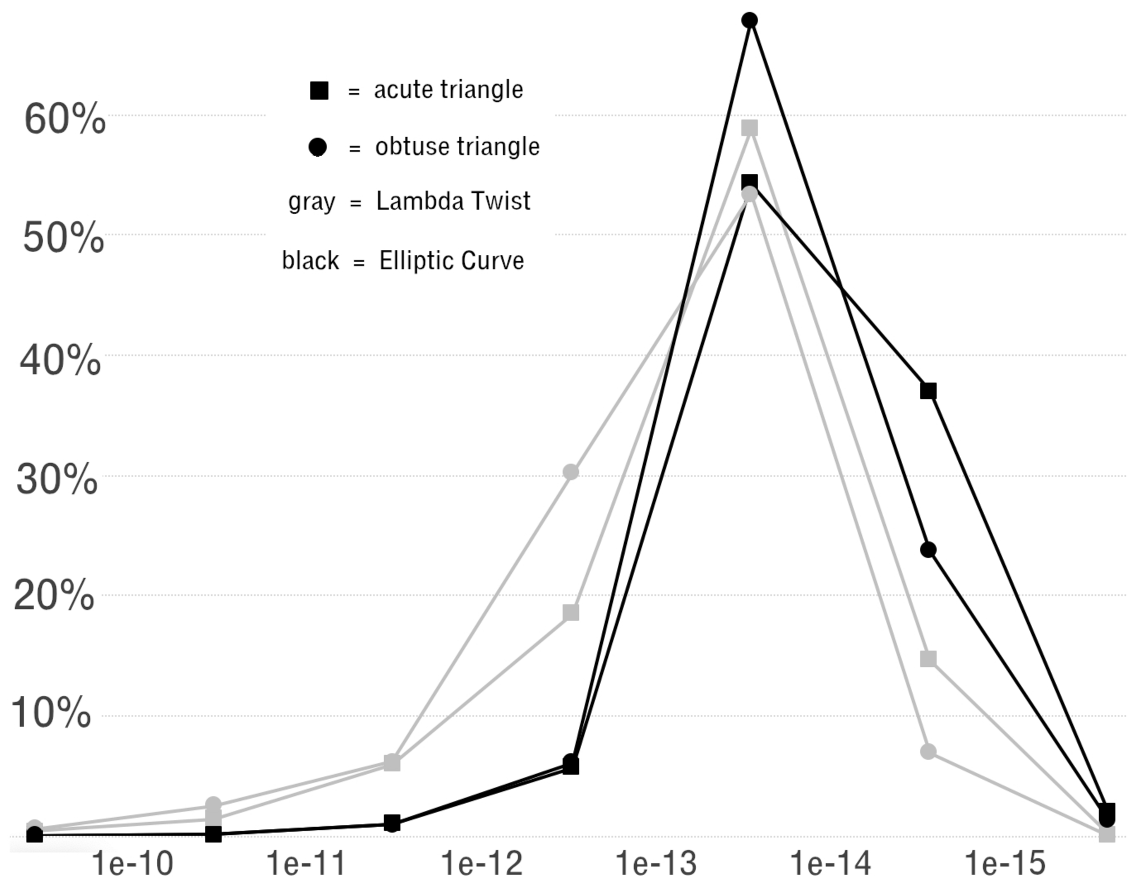
One of the constraints is relaxed and blue circle is allowed to move

- When this question is translated from the unit sphere to the real projective plane, we find that we are simply looking for the ***intersection of a (genus-one) quartic curve and a line!***
- That is easy to solve
- This thinking in the projective plane serves as the basis for the P3P-solver algorithm that was implemented and tested against the Lambda Twist algorithm
- I will close by briefly discussing the results of that testing
- The "attack angle" made a big difference
- By this, it is meant the angle between the line through the origin and the circumcenter of the triangle, and the line normal to the triangle, through its circumcenter

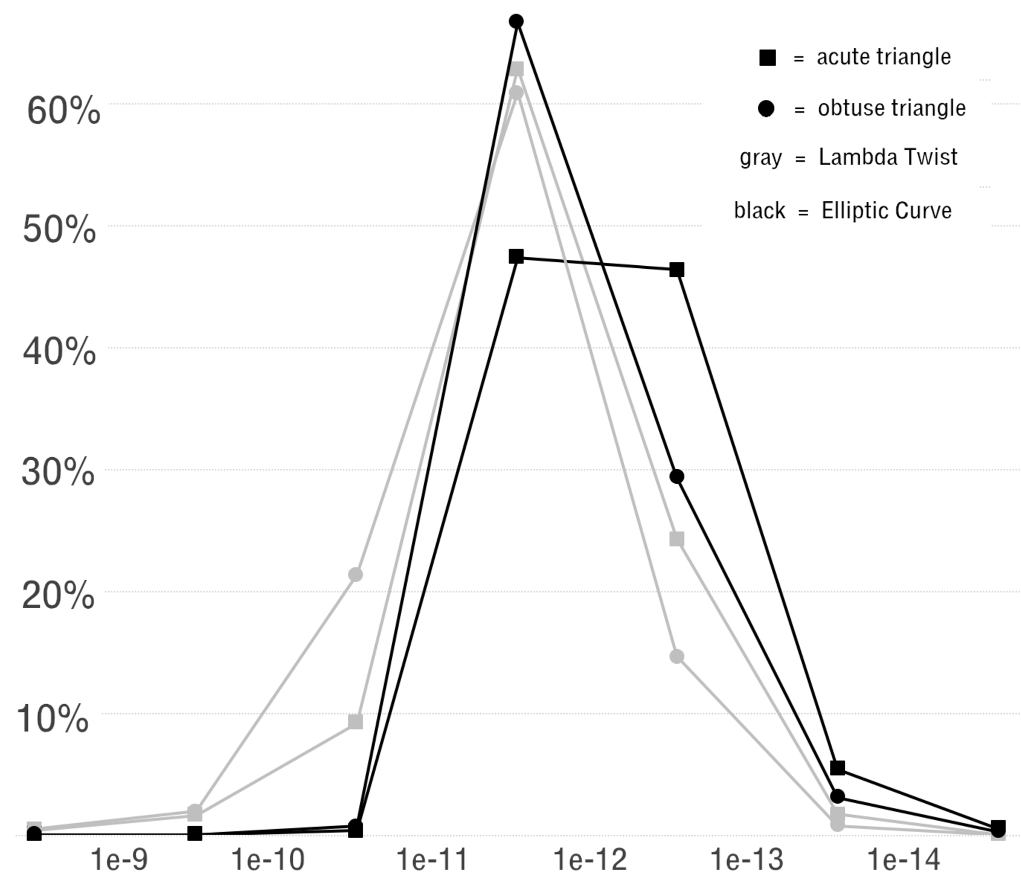
- The simulations involved taking a given triangle with vertices on the unit circle in the *xy*-plane, randomly rotating it in 3 dimensions about the origin, randomly lifting it along the positive *z* direction, and then randomly rotating about the origin again
- Then the vertices of the triangle, treated as control points, were observed from the origin, and the P3P-solver algorithms were applied to estimate the distances to the control points
- The first rotation was restricted to produce a restricted range of attack angles
- The lifting along the positive *z* direction was also restricted to a certain range of elevations

- The data collected and reported in the next few slides are for two different triangles, an obtuse triangle and an acute triangle (fairly close to equilateral)
- For each of the plots in these slides, ten million random trials were used
- Average errors and their standard deviations, minimum and maximum errors, and execution times were recorded
- As long as this attack angle was not close to a right angle, the method based on elliptic curves had an average error that was at least one order of magnitude better than that of the Lambda Twist method
- The minimum error, taken over a large number of trials, was similarly better
- The execution time was about 60 percent worse

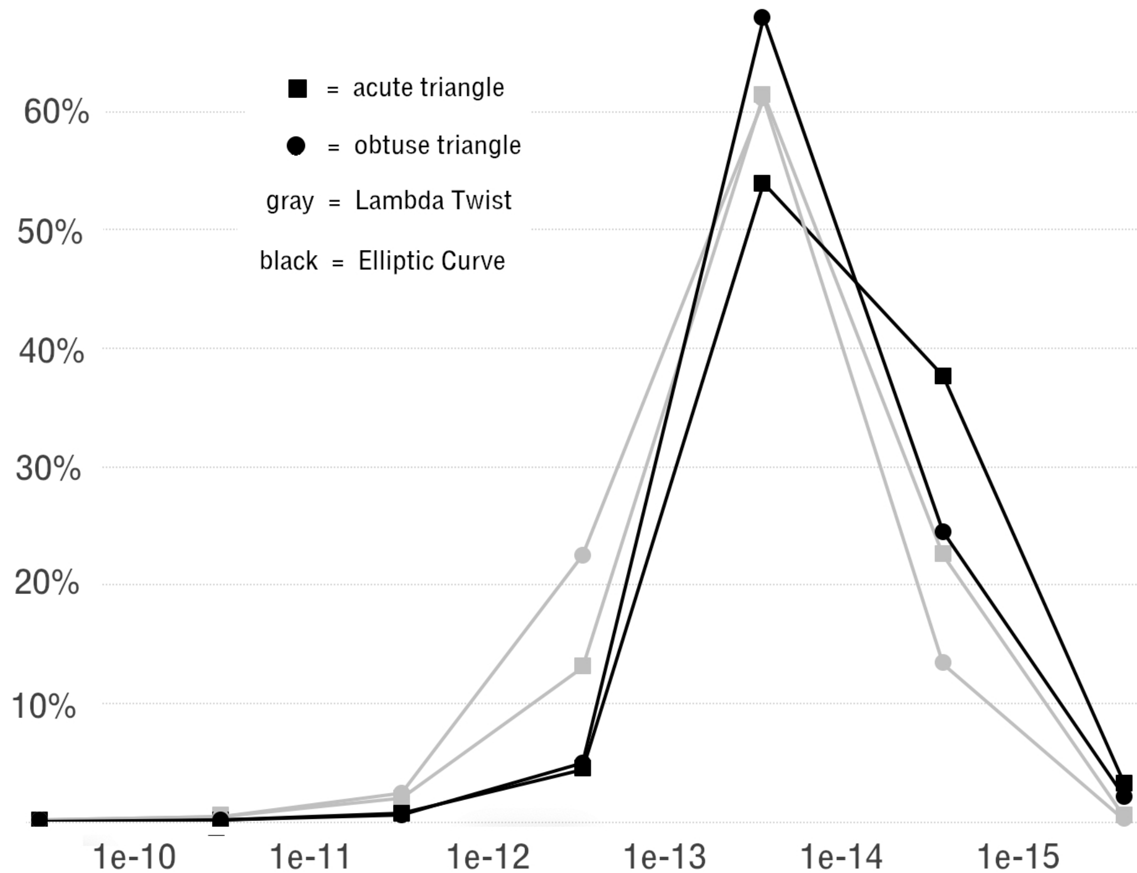
- Average error distributions when attack angle was between 0 and 30 degrees, and elevation of triangle was between 10 and 20 (for two different triangles):



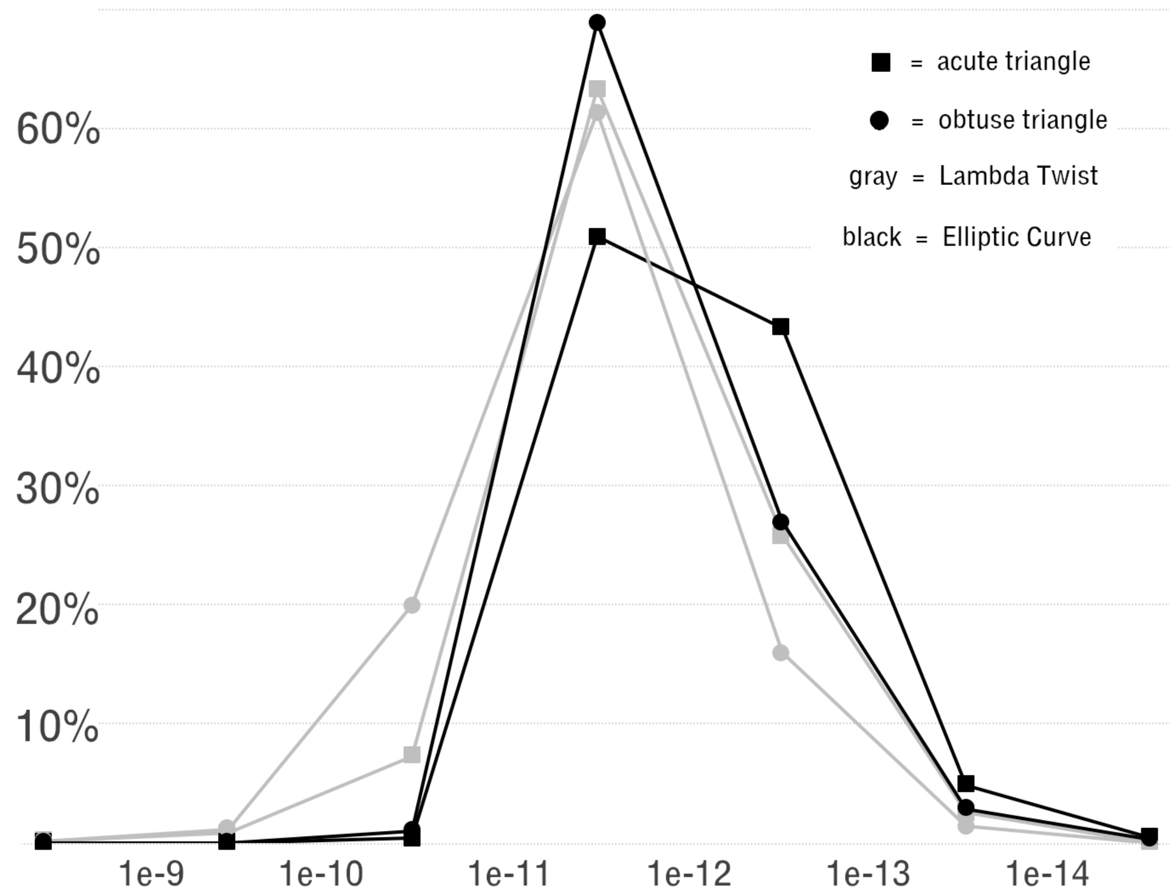
- Average error distributions when attack angle was between 0 and 30 degrees, and elevation of triangle was between 100 and 200 (for two different triangles):



- Average error distributions when attack angle was between 30 and 60 degrees, and elevation of triangle was between 10 and 20 (for two different triangles):



- Average error distributions when attack angle was between 30 and 60 degrees, and elevation of triangle was between 100 and 200 (for two different triangles):



REFERENCES

Apostle, T. M. and Mnatsakanian, M. A. (2009). A new look at the co-called Trammel of Archimedes. *Amer. Math. Monthly*, 116(2): 115–133.

Bernstein, J. D., Birkner, P. and Peters, C. (2008). Twisted Edwards Curves. *IACR Cryptology ePrint Archive*, DOI:10.1007/978-3-540-68164-9_26.

Edwards, H. M. (2007). A normal form for elliptic curves. *Bull. Amer. Math. Soc.*, 44(3): 393–422.

Gao, X. S., Hou, X. R. and Tang, J. L. (2003). Complete solution classification for the perspective-three-point problem. *Pattern Anal. Mach. Intell. IEEE Trans.*, 25 (8): 930-943.

Grunert, J. A. (1841). Das pothenotische problem in erweiterter gestalt nebst über seine anwendungen in der geodäsie. *Grunerts Archiv für Mathematik und Physik*, 1: 238–248.

Haralick, R. M., Lee, C.-N., Ottenberg, K. and Nölle, N. (1994). Review and analysis of solutions of the three point perspective pose estimation problem. *J. Computer Vision*, 13(3): 331–356.

Izmestiev, I. (2017). Spherical and hyperbolic conics. *arXiv:1702.06860v1 [math.MG].*

Ke, T. and Roumeliotis, S. (2017). An efficient algebraic solution to the perspective-three-point problem. *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*: 4618–4626.

Kneip, L., Scaramuzza, D. and Siegwart, R. (2011). A novel parameterization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. *Proc. 2011 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*: 2969-2976.

Persson, M. and Nordberg, K. (2018). Lambda Twist: an accurate fast robust perspective three point (P3P) solver. *Proc. 15th Europ. Conf. Computer Vision (ECCV)*, part 4: 334–349.

Wang, P., Xu, G., Wang, Z. and Yuehua, C. (2018). An efficient solution to the perspective-three-point pose problem. *Computer Vision and Image Understanding*, 166 (C): 81-87.

Wetzel, J. E. (2010). An ancient elliptic locus. *Amer. Math. Monthly*, 117(2): 161–167.

Thank You

Questions?