Abstract. What’s the hardest math I’ve ever used in real life? Me, myself, directly — not by using a cellphone or a GPS device that somebody else designed? And in “real life” — not while studying or teaching mathematics?

I use addition and subtraction daily, adding up bills or calculating change. I use percentages often, though mostly it is just “add 15 percents”. I seldom use multiplication and division: when I buy in bulk, or when I need to know how many miles I need to replace my kitchen floor. I’ve used powers twice in my life, doing calculations related to mortgages.

I’ve used a tiny bit of geometry and algebra for a tiny bit of non-math-related computer graphics I’ve played with. And for a long time, that was all. In my talk I will tell you how recently a math topic discovered only in the 1800s made a brief and modest appearance in my non-mathematical life.

Goal. Find the least-blur path to go from Mona’s left eye to Mona’s right eye in fixed time. Alternatively, fix your blur-tolerance, and find the fastest path to do the same. For fixed blur, our camera moves at a speed proportional to its distance from the image plane:

How go fastest?

The Mona Plane
**The Hardest Math I’ve Ever Really Used, 2**

**Fermat’s Principle**

\[ c \sim 300,000 \]

\[ c \sim 250,000 \]

**The Brachistochrone**

\[ \frac{mgh}{\frac{1}{2}mv^2} = \sqrt{\frac{0}{10}} \]

\[ \sqrt{\frac{10}{20}} \]

\[ \sqrt{\frac{30}{40}} \]

\[ \sqrt{\frac{40}{50}} \]

**Bernoulli on Newton.** “I recognize the lion by his paw”.

**The Least Action Principle.** Everywhere in physics, a system goes from \( A \) to \( B \) along the path of least action.

With small print for quantum mechanics.

**The Happy Segway Principle**

A Segway is happy iff both its wheels are happy or unhappy.

**The Happy Camera-carrying Segways above the Mona Plane**

**The Lobachevsky Plane**

Two parallels through one point

**The Actual Code**

```plaintext
p3.y = p2.y + b*x3p;
x = p1.x-p2.x; y = p1.y-p2.y;
d1 = p1.d; d2 = p2.d;
norm = sqrt(x*x + y*y);
a = x/norm; b = y/norm;
xis = a*x + b*y;
x0 = (xis + (d1*d1-d2*d2)/xis)/2;
x = x0 + (a+d1-d2)/2/a;
xpp = cos(theta3); d3pp = sin(theta3);
x3p = x0 + r*x3pp; p3.d = r*d3pp;
p3.x = p2.x + a*x3p;
```