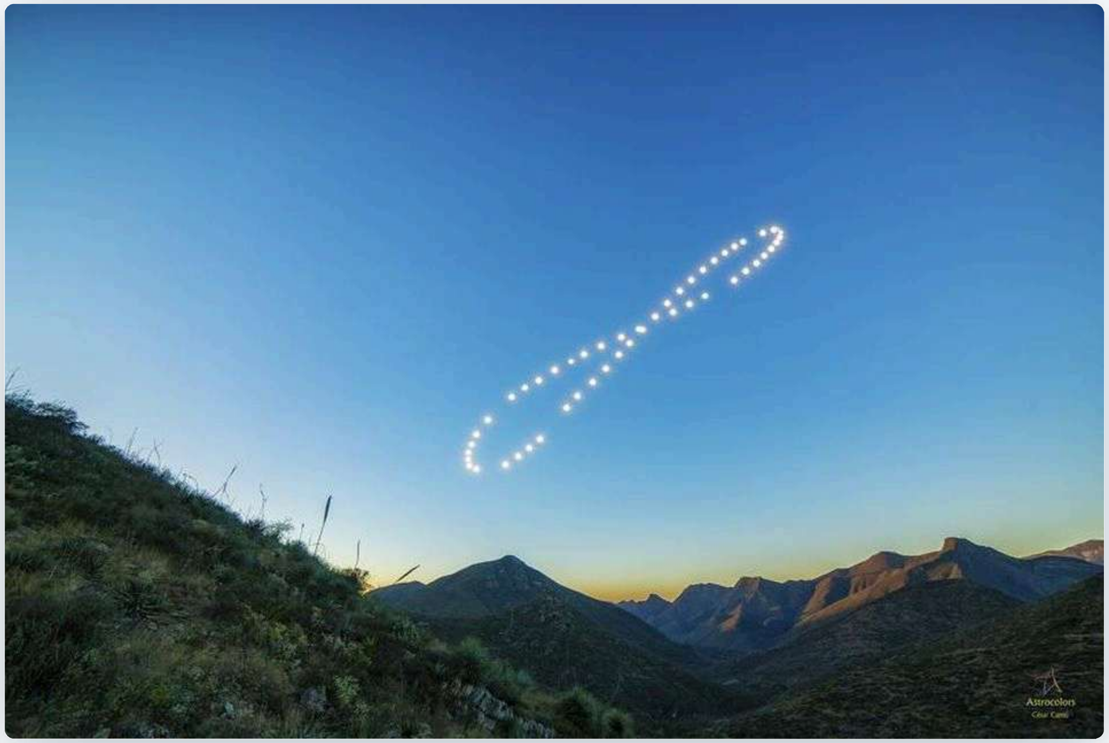
Omar Shehata ↩

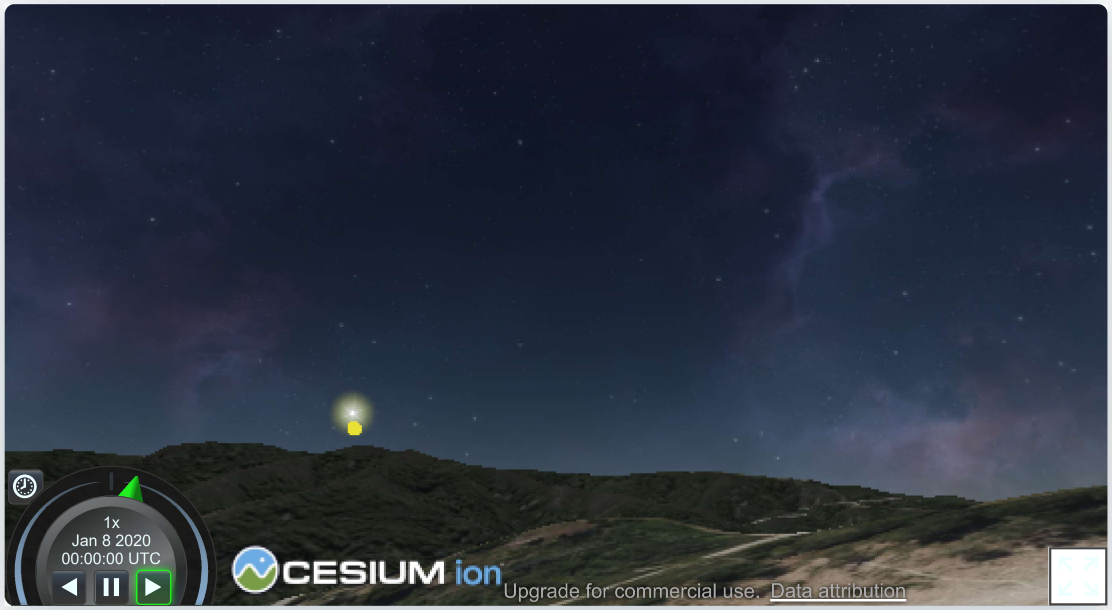
How to visualize the Analemma

From Omar's notebook.

An analemma is the shape the sun makes when you take a picture of it every day, at the same time and at the same location, for a year.


From Forbes.com. CÉSAR CANTÚ / ASTROCOLORS

I was curious if I could simulate this in software. The answer is yes! Below is what it looks like in San Francisco. Click and drag in the sky to look around.



Reset the viewer.

This was *really* fun because it was almost like *discovering* this phenomenon. The libraries I used were not designed to simulate this particular thing: it was a globe engine + a model of the sun's position over time.

The creators of these tools may not have necessarily been aware of this phenomenon at all. It just happens as a result of the position of the sun relative to the Earth throughout the year. It's like using software to discover something new about the universe.

How to recreate this visualization yourself

I'll show you how to run this code yourself so you have a sandbox to experiment with before we talk about theory.

Code snippet you can run/fork in your browser:
https://repl.it/@OmarShehata2/analemma#script.js

To modify the location:

1. Change the longitude/latitude/height on line 23.

You can enter these in degrees using `Cesium.Cartesian3.fromDegrees(37.757,-122.507, 150);`.

2. Modify the heading/pitch on lines 28-29.

I've found this easier done interactively. The `window.viewer` is exposed for this reason. Just open the example in a new tab (using a url like this: analemma.omarshehata2.repl.co) and you can paste the `viewer.camera.flyTo` call in the console to tweak the parameters in real time.

3. Modify the hour the sun is captured every day on line 73.

This is important because depending on where you are in the world, you want to set the hour to some time when the sun is visible. Hour values are `0-23`.

You can move freely in this viewer, anywhere in the world, and then find the sun. The analemma drawing will reset so you can see it. It always draws based on the location of the camera in the given frame.

Final tip: if you find a really nice vantage point while exploring, you can capture the lon/lat using `viewer.camera.positionCartographic` which gives you the longitude, latitude, and height.

The theory behind it

To create this visualization we're using CesiumJS as the globe engine. It also has a function for computing the sun's position relative to the Earth (in ECEF coordinates) given a datetime:

```
// Get the sun's position right now.
```

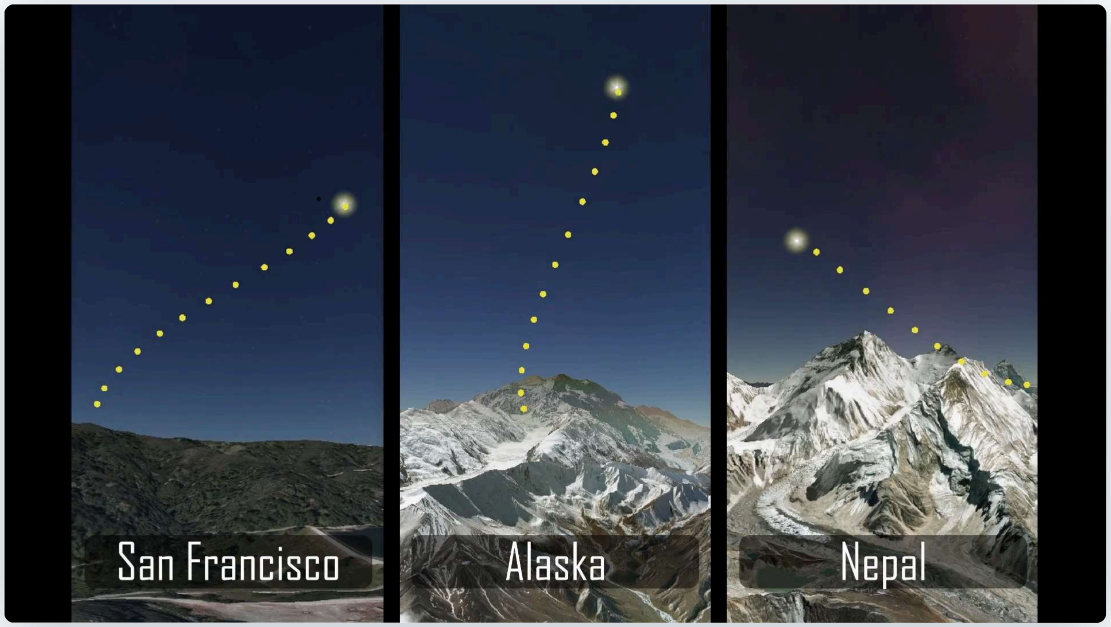
```
const time = Cesium.JulianDate.fromIso8601(new Date().toISOString());
const sunPosition = Cesium.Simon1994PlanetaryPositions.computeSunPositionInE
arthInertialFrame(time);
```

To plot the sun's location in the sky, we treat the sky as a 2D canvas: subtract the position of the sun from the camera to get the direction. Normalize that, then scale it by some arbitrary amount. Then put a yellow dot at that location.

This means moving the camera breaks this drawing (just like it does in real life!) You can alternatively save the canvas as an image every X frames, and then stitch together the pictures, which is how you'd do it in the real world.

The final piece: move the clock by one day every frame, keeping it at the same hour. If the Earth's orbit were perfectly circular, the sun would appear in the same location every day at the same hour.

Which hour should you choose? It turns out not to matter. The shape is always going to be exactly the same, it just moves the whole thing around in the sky. I find that picking a time such that the tip of the analemma is right at the horizon during sunset or sunrise is particularly mesmerizing.



A discontinuity in the universe?

If you implement exactly what I described above, this is what you might see:

Why does this jarring discontinuity happen? I figured it was daylight savings, but it was surprising because nothing in the functions I was using was explicitly handling timezones. This same function should work the same way regardless of where in the world you're standing (what about places that don't change the clock?)

I fixed it by just adding/subtracting an hour on March 8th and November 1st. But there's more questions to answer here: does this come from CesiumJS? From JavaScript's date objects? From the ISO 8601 spec?

In any case, it was pretty cool to learn that this bug is a physical manifestation of daylight savings.

1 Comment   1 Login ▼

G

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

♡ Share   Best Newest Oldest

Omar Shehata Mod

5 years ago

There's some really good questions in the Reddit discussion here:
https://www.reddit.com/r/As...

My favorites are:

* Why is the Nepal one mirrored from the SF one? I think it's because I did one at sunrise and one at sunset.
* Does this change every year, given that the Earth wobbles? I'm not sure! I think it should be different, if not year-to-year then perhaps with thousands of years? If the solar modal here doesn't handle it you could plug in one that does

0 0 Reply