Copilot

The `Slerp` function from `scipy.spatial.transform` is used to interpolate between rotations. It stands for Spherical Linear intERPolation. Given a set of keyframe rotations and corresponding times, `Slerp` computes the interpolated rotations for any time within the keyframe times.

In your code, you've defined key rotations (`key_rot`) at certain key times (`key_times`) and then created a `Slerp` object. You can use this `Slerp` object to get the interpolated rotation at any time within `key_times`.

Now, if you want to get the angles (theta, phi) corresponding to these interpolated rotations, you can use the `as_euler` method of the `Rotation` class. This method returns the Euler angles corresponding to the rotation.

Here's how you can modify your `draw_ellipsoid` function to print the (theta, phi) angles for each interpolated rotation:

```
def draw_ellipsoid(t, dims, ax):
    rotation = slerp(t)
    theta, phi, _ = rotation.as_euler('ZYX', degrees=True)
    print(f"Time: {t}, Theta: {theta}, Phi: {phi}")
    # rest of your code...
```

In this code, `rotation.as_euler('ZYX', degrees=True)` returns the Euler angles corresponding to the rotation, in degrees. The 'ZYX' argument specifies the order of the rotations. Since we're interested in the 'Z' and 'Y' rotations (which correspond to theta and phi, respectively), we ignore the 'X' rotation.

Please note that the angles returned by `as_euler` are in the range [-180, 180] for theta and [-90, 90] for phi. If you want them in the range [0, 360] and [0, 180], respectively, you can add 360 to negative angles.

I hope this helps! Let me know if you have any other questions. 😊