Input:
Points of Interest (list of points)
Solar Vectors (list of vectors)
Environment (single mesh)
Distances (list of floats)

Pseudocode:

```
#intersect rays with environment
For each PoI:
        sum of intersections = 0
        For each Solar Vector:
                Draw reverse Solar Vector from PoI
                Intersect with environment
                if it intersects:
                        sum = sum + 1
        sums.append(sum)

#normalize the sums (0.0 to 1.0)
highestvalue = max(sums)
for each sum in sums:
        normalizedvalue = sum/highestvalue
        normalizedvalues.append(normalizedvalue)

#making sure the heart of the building receives no shadow
for each normalizedvalue:
        if distance > 11000:
                normalizedvalue = 1
        if 7000 < distance < 11000:
                normalizedvalue = 0.9
        if 6000 < distance < 7000:
                normalizedvalue = 0.7
        shadowvalues.append(normalizedvalue)
```

Output:
list of floats indicating how much shadow each voxel receives