

6.801/6.866: Machine Vision, Lecture 23

Professor Berthold Horn, Ryan Sander, Tadayuki Yoshitake
MIT Department of Electrical Engineering and Computer Science
Fall 2020

These lecture summaries are designed to be a review of the lecture. Though I do my best to include all main topics from the lecture, the lectures will have more elaborated explanations than these notes. Therefore, if you're looking for the most rigorous review and treatment of these topics, we encourage you to rewatch the lecture videos. With that said, we hope these summaries are beneficial for your learning. If you have any feedback for these lecture summaries, please submit it [here](#).

1 Lecture 23: Gaussian Image and Extended Gaussian Image, Solids of Revolution, Direction Histograms, Regular Polyhedra

In this lecture, we cover more on the Gaussian images and extended Gaussian images (EGI), as well as integral Gaussian images, some properties of EGI, and applications of these frameworks.

1.1 Gaussian Curvature and Gaussian Images in 3D

Let's start with defining what a Gaussian image is:

Gaussian Image: A correspondence between points on an image and corresponding points on the unit sphere based off of equality of the surface normal vectors.

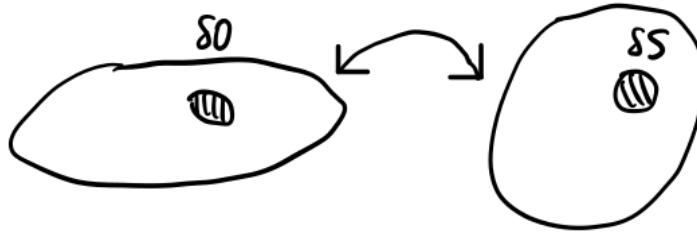


Figure 1: Mapping from object space to the Gaussian sphere using correspondences between surface normal vectors. Recall that this method can be used for tasks such as image recognition and image alignment.

Recall that we defined the **Gaussian curvature** of an (infinitesimal) patch on an object as a scalar that corresponds to the ratio of the areas of the object and the Gaussian sphere we map the object to, as we take the limit of the patch of area on the object to be zero:

$$\begin{aligned}\kappa_{\text{Gaussian}} &= \lim_{\delta \rightarrow 0} \frac{\delta S}{\delta O} \\ &= \text{The ratio of areas of the image and the object}\end{aligned}$$

Rather than use Gaussian curvature, however, we will elect to use the inverse of this quantity (which is given by density) $\frac{1}{\kappa}$, which roughly corresponds to the amount of surface that contains that surface normal.

We can also consider the integral of Gaussian curvature, and the role this can play for dealing with object discontinuities:

1.1.1 Gaussian Integral Curvature

The integral of Gaussian curvature is given by:

$$\iint_O K dO = \iint_O \frac{dS}{dO} dO = \iint_O dS = S \text{ (The area of a corresponding patch on the sphere)}$$

Note: One nice feature of **integral curvature**: This can be applied when curvature itself cannot be applied, e.g. at discontinuities such as edges and corners.

We can also apply the integral over the Gaussian sphere:

$$\iint_S \frac{1}{K} dS = \iint_S \frac{dO}{dS} dS = \iint_S dO = O \text{ (The area of a corresponding path on the object)}$$

This second derivation is used more often in practice.

1.1.2 How Do We Use Integral Gaussian Curvature?

As we briefly mentioned above, integral Gaussian curvature can be used for “smoothing out” discontinuities and providing continuity when there is none guaranteed by the curvature function itself. Discontinuities can be caused by features such as corners, edges, and other abrupt changes in surface orientation.

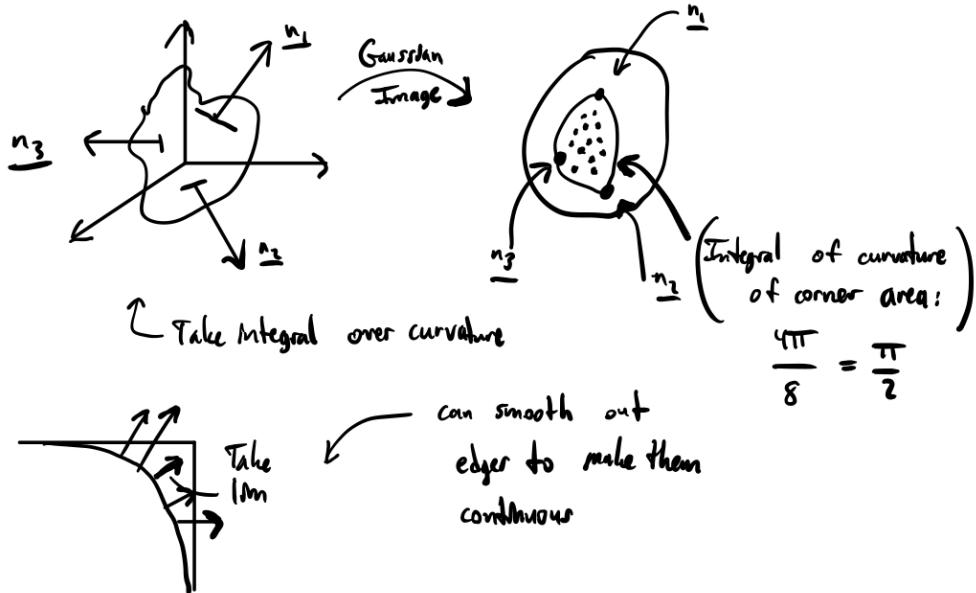


Figure 2: An application in which integral Gaussian curvature can be used in lieu of the curvature function. The smoothing properties of integration enable for the mitigation of discontinuity affects.

Next, let's consider the different forms our mass density distribution G can take.

1.1.3 Can We Have Any Distribution of G on the Sphere?

It is important to understand if we need to impose any restrictions on our mass density distribution. Let's analyze this in both the discrete and continuous cases. The results from the discrete case extend to the continuous case.

- **Discrete Case:** Consider an object that we have discretized to a set of polygon facets, and consider that we have a camera that observes along an optical axis \hat{v} , and a mirror image of this camera that observes along the optical axis $-\hat{v}$. Note that any surface normals that are not parallel to these optical axes will have foreshortening effects imposed.

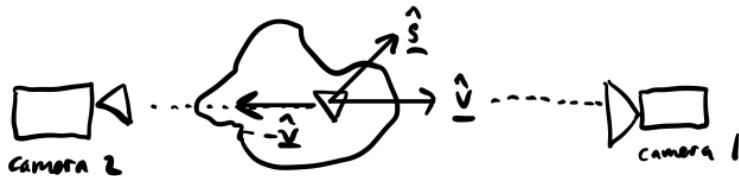


Figure 3: Problem setup in the discrete case. Consider our object parameterized by a discrete number of finite polygon facets, and a camera and its mirror image observing the different surface normals of these facets from two opposite vantage points.

First, consider only taking the facets with positive dot product (i.e. 90 degrees or less) away from the vector $\hat{\mathbf{v}}$, and sum the product of these facet area multiplied by normal and the unit vector corresponding to the optical axis $\hat{\mathbf{v}}$:

$$\sum_{\hat{\mathbf{s}}_i \cdot \hat{\mathbf{v}} \geq 0} A_i \hat{\mathbf{s}}_i \cdot \hat{\mathbf{v}}$$

Now, consider the other facets, that is, those for which we take the facets with positive dot product (i.e. 90 degrees or less) away from the vector $-\hat{\mathbf{v}}$, which is 180 degrees rotated from $\hat{\mathbf{v}}$, and applying the same operation:

$$\sum_{\hat{\mathbf{s}}_i \cdot (-\hat{\mathbf{v}}) \geq 0} A_i \hat{\mathbf{s}}_i \cdot (-\hat{\mathbf{v}})$$

We must have equality between these two sums:

$$\sum_{\hat{\mathbf{s}}_i \cdot \hat{\mathbf{v}} \geq 0} A_i \hat{\mathbf{s}}_i \cdot \hat{\mathbf{v}} = \sum_{\hat{\mathbf{s}}_i \cdot (-\hat{\mathbf{v}}) \geq 0} A_i \hat{\mathbf{s}}_i \cdot (-\hat{\mathbf{v}})$$

And therefore, if we combine the sums to the same side:

$$\begin{aligned} & \sum_{\hat{\mathbf{s}}_i \cdot \hat{\mathbf{v}} \geq 0} A_i \hat{\mathbf{s}}_i \cdot \hat{\mathbf{v}} - \left(\sum_{\hat{\mathbf{s}}_i \cdot (-\hat{\mathbf{v}}) \geq 0} A_i \hat{\mathbf{s}}_i \cdot (-\hat{\mathbf{v}}) \right) = \mathbf{0} \\ & \sum_{\hat{\mathbf{s}}_i \cdot \hat{\mathbf{v}} \geq 0} A_i \hat{\mathbf{s}}_i \cdot \hat{\mathbf{v}} - \left(- \sum_{\hat{\mathbf{s}}_i \cdot (-\hat{\mathbf{v}}) \geq 0} A_i \hat{\mathbf{s}}_i \cdot \hat{\mathbf{v}} \right) = \mathbf{0} \\ & \sum_{\hat{\mathbf{s}}_i \cdot \hat{\mathbf{v}} \geq 0} A_i \hat{\mathbf{s}}_i \cdot \hat{\mathbf{v}} + \sum_{\hat{\mathbf{s}}_i \cdot (-\hat{\mathbf{v}}) \geq 0} A_i \hat{\mathbf{s}}_i \cdot \hat{\mathbf{v}} = \mathbf{0} \\ & \sum_{\{\hat{\mathbf{s}}_i \cdot \hat{\mathbf{v}} \geq 0\} \cup \{\hat{\mathbf{s}}_i \cdot (-\hat{\mathbf{v}}) \geq 0\}} A_i \hat{\mathbf{s}}_i \cdot \hat{\mathbf{v}} = \mathbf{0} \\ & \sum_i A_i \hat{\mathbf{s}}_i \cdot \hat{\mathbf{v}} = \mathbf{0} \\ & (\sum_i A_i \hat{\mathbf{s}}_i) \cdot \hat{\mathbf{v}} = \mathbf{0} \quad \forall \hat{\mathbf{v}} \end{aligned}$$

Therefore, from this it must be the case that $\sum_i A_i \hat{\mathbf{s}}_i = \mathbf{0}$, i.e. that the center of mass of the mapping must be at the center of the sphere. This also means that the Gaussian density G must have a “zero-mean” mass distribution for any closed object. This does not generally hold for any non-closed objects.

- **Continuous Case:** This same result holds in the continuous case - the mean must be at the center of the sphere for closed objects. Mathematically:

$$\iint_S G(\hat{\mathbf{s}}) \hat{\mathbf{s}} dS = \mathbf{0}$$

Therefore, for both the discrete and continuous cases, we can think of the Extended Gaussian Image (EGI) as a mass distribution over the sphere that objects are mapped to according to their surface normals.

Next, let's look at some discrete applications. Continuous EGI can be useful as well, especially if we are using an analytical model (in which case we can compute the exact EGI).

1.2 Examples of EGI in 3D

Let's start with some examples of EGI in 3D.

1.2.1 Sphere: EGI

We have seen this before for the circle, and can simply generalize this from 2D to 3D:

- **Curvature:** $\kappa = R^2$
- **Density:** $G = \frac{1}{K} = \frac{1}{R^2}$

We saw that because the curvature and density are the same everywhere, that this imposes limitations on the effectiveness of this surface for recognition and alignment tasks. Let's next consider something with more complicated structure.

1.2.2 Ellipsoid: EGI

Recall our two main ways to parameterize ellipses/ellipsoids: implicit (the more common form) and “squashed circle/sphere” (in this case, the more useful form):

- **Implicit parameterization:** $\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 = 1$
- **“Sphere Squashing” parameterization:** These are parameterized by θ and ϕ :

$$\begin{aligned}x &= a \cos \theta \cos \phi \\y &= b \sin \theta \cos \phi \\z &= c \sin \phi\end{aligned}$$

We will elect to use the second of these representations because it allows for easier and more efficient generation of points by parameterizing θ and ϕ and stepping these variables.

As we did last time for the 2D ellipse, we can express these parametric coordinates as a vector parameterized by θ and ϕ :

$$\mathbf{r} = (a \cos \theta \cos \phi, b \sin \theta \cos \phi, c \sin \phi)^T$$

And similarly to what we did before, to compute the **surface normal** and the **curvature**, we can first find (in this case, two) tangent vectors, which we can find by taking the derivative of \mathbf{r} with respect to θ and ϕ :

$$\begin{aligned}\mathbf{r}_\theta &\triangleq \frac{d\mathbf{r}}{d\theta} = (-a \sin \theta \cos \phi, b \cos \theta \cos \phi, 0)^T \\\mathbf{r}_\phi &\triangleq \frac{d\mathbf{r}}{d\phi} = (-a \cos \theta \sin \phi, -b \sin \theta \sin \phi, c \cos \phi)^T\end{aligned}$$

Then we can compute the normal vector by taking the cross product of these two tangent vectors of the ellipsoid. We do not carry out this computation, but this cross product gives:

$$\begin{aligned}\mathbf{n} &= \mathbf{r}_\theta \times \mathbf{r}_\phi = \cos \phi (b \cos \theta \cos \phi, a \sin \theta \cos \phi, a \sin \phi)^T \\&= (b \cos \theta \cos \phi, a \sin \theta \cos \phi, a \sin \phi)^T\end{aligned}$$

Where we have dropped the $\cos \phi$ factor because we will need to normalize this vector anyway. Note that this vector takes a similar structure to that of our original parametric vector \mathbf{r} .

With our surface normals computed, we are now equipped to match surface normals on the object to the surface normals on the unit sphere. To obtain **curvature**, our other quantity of interest, we need to differentiate again. We will also change our parameterization from $(\theta, \phi) \rightarrow (\xi, \eta)$, since (θ, ϕ) parameterize the object in its own space, and we want to parameterize the object in unit sphere space (ξ, η) .

$$\begin{aligned}\hat{\mathbf{n}} &= (\cos \xi \cos \eta, \sin \xi \cos \eta, \sin \eta)^T \\\bar{\mathbf{s}} &= (a \cos \xi \cos \eta, b \sin \xi \cos \eta, \sin \eta)^T \\\kappa &= \left(\frac{\mathbf{s} \cdot \mathbf{s}}{abc} \right)^2 \\G &= \frac{1}{K} = \left(\frac{abc}{\mathbf{s} \cdot \mathbf{s}} \right)^2\end{aligned}$$

With the surface normals and curvature now computed, we can use the distribution G on the sphere for our desired tasks of **recognition** and **alignment**!

Related, let us look at the **extrema** of curvature for our ellipsoid. These are given as points along the axes of the ellipsoids:

1. $\left(\frac{bc}{a}\right)^2$
2. $\left(\frac{ac}{b}\right)^2$
3. $\left(\frac{ab}{c}\right)^2$

For these extrema, there will always be one maxima, one minima, and one saddle point. The mirror images of these extrema (which account for the mirrors of each of these three extrema above) exhibit identical behavior and reside, geometrically, on the other side of the sphere we map points to.

How Well Does this Ellipsoid Satisfy Our Desired Representation Properties?

- **Translation invariance:** As we saw with EGI last time, this representation is robust to translation invariance.
- **Rotation “Equivariance”:** This property works as desired as well - rotations of the ellipsoid correspond to **equivariant rotations** of the EGI of this ellipse.

1.3 EGI with Solids of Revolution

We saw above that the EGI for an ellipsoid is more mathematically-involved, and is, on the other end, “too simple” in the case of a sphere. Are there geometric shapes that lend themselves well for an “intermediate representation”? It turns out there are, and these are the **solids of revolution**. These include:

- Cylinders
- Spheres
- Cones
- Hyperboloids of one and two sheets

How do we compute the EGI of solids of revolution? We can use **generators** that produce these objects to help.

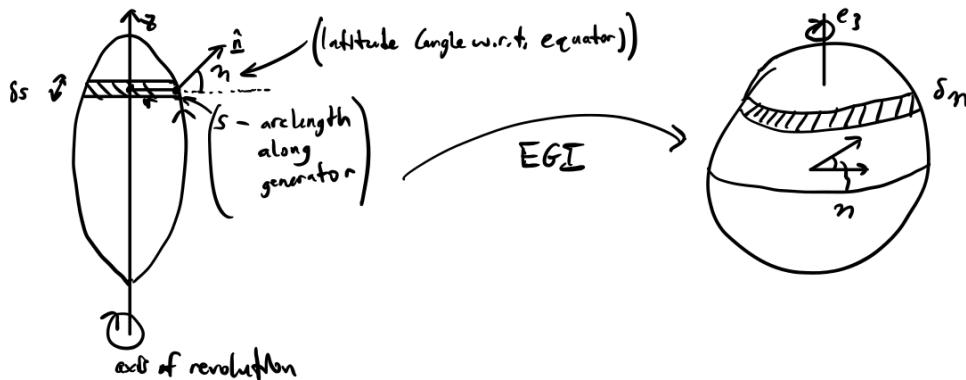


Figure 4: EGI representation of a generalized solid of revolution. Note that bands in the object domain correspond to bands in the sphere domain.

As we can see from the figure, the bands “map” into each other! These solids of revolution are symmetric in both the object and transform space. Let’s look at constructing infinitesimal areas so we can then compute Gaussian curvature κ and density G :

- **Area of object band:** $\delta O = 2\pi r \delta s$
- **Area of sphere band:** $\delta S = 2\pi \cos(\eta) \delta n$

Then we can compute the curvature as:

$$\begin{aligned}\kappa &= \frac{\delta S}{\delta O} = \frac{2\pi \cos(\eta) \delta \eta}{2\pi r \delta s} = \frac{\cos(\eta) \delta \eta}{r \delta s} \\ G &= \frac{1}{\kappa} = \frac{\delta O}{\delta S} = r \sec(\eta) \frac{\delta s}{\delta \eta}\end{aligned}$$

Then in the limit of $\delta \rightarrow 0$, our curvature and density become:

$$\begin{aligned}\kappa &= \lim_{\delta \rightarrow 0} \frac{\delta S}{\delta O} = \frac{\cos \eta}{r} \frac{d\eta}{ds} \quad (\text{Where } \frac{d\eta}{ds} \text{ is the rate of change of surface normal direction along the arc, i.e. curvature}) \\ G &= \lim_{\delta \rightarrow 0} \frac{\delta O}{\delta S} = r \sec \eta \frac{ds}{d\eta} \quad (\text{Where } \frac{ds}{d\eta} \text{ is the rate of change of the arc length w.r.t. angle}) \\ \kappa &= \frac{\cos \eta}{r}\end{aligned}$$

1.4 Gaussian Curvature

Using the figure below, we can read off the trigonometric terms from this diagram to compute the curvature κ in a different way:

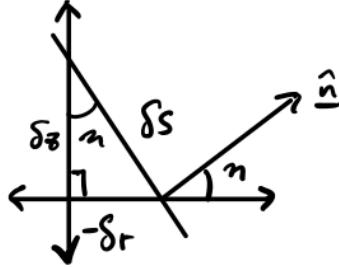


Figure 5: Our infinitesimal trigonometric setup that we can use to calculate Gaussian curvature.

We can read off the trigonometric terms as:

$$\sin \eta = \frac{-\delta r}{\delta s} = -r_s \text{ (First derivative)}$$

$$\cos \eta \frac{d\eta}{ds} = \frac{d}{ds}(-r_s) = -r_{ss} \text{ (Second derivative)}$$

Then the curvature for the solid of revolution is given by:

$$\kappa = \frac{-r_{ss}}{r}$$

Let's look at some examples to see how this framework applies:

1.4.1 Example: Sphere

Take the following cross-section of a sphere in 2D:

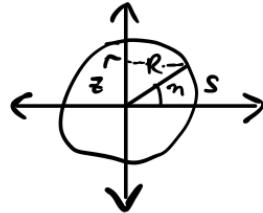


Figure 6: Cross-section of a 3D sphere in 2D. We can apply our framework above to derive our same result for Gaussian curvature.

Mathematically:

$$r = R \cos \left(\frac{S}{R} \right)$$

$$r_{ss} = R \left(-\frac{1}{R^2} \right) \cos \left(\frac{S}{R} \right) = -\frac{1}{R} \cos \left(\frac{S}{R} \right)$$

Then curvature is given by:

$$\kappa = \frac{-r_{ss}}{r} = \frac{-\left(-\frac{1}{R}\right) \cos \left(\frac{S}{R}\right)}{R \cos \left(\frac{S}{R}\right)} = \frac{1}{R^2}$$

Alternatively, rather than expressing this in terms of R and S , we can also express as a function of z (with respect to the diagram

above). Let's look at some of the quantities we need for this approach:

$$\begin{aligned}\tan \eta &= -\frac{dr}{dz} = -r_z \text{ (First derivative)} \\ \sec^2 \frac{d\eta}{ds} &= \frac{d}{ds}(-r_z) = -r_{zz} \frac{dz}{ds} \text{ (By Chain Rule)} \\ \sec^2 \eta &= 1 + \tan^2 \eta \\ \cos \eta &= \frac{dz}{d\eta} = -z_s\end{aligned}$$

Putting all of these equations together:

$$\begin{aligned}\kappa &= \frac{-r_{zz}}{r(1+z^2)^2} \\ r &= \sqrt{R^2 - Z^2} \\ r_z &= -\frac{Z}{\sqrt{R^2 - Z^2}} \\ r_{zz} &= -\frac{R^2}{(R^2 - Z^2)^{\frac{3}{2}}} \\ 1 + r_z^2 &= \frac{R^2}{R^2 - Z^2} \\ \text{Then: } \kappa &= \frac{-r_{zz}}{r(1+z^2)^2} = \frac{1}{R^2}\end{aligned}$$

One particular EGI, and the applications associated with it, is of strong interest - the **torus**.

1.4.2 EGI Example Torus

Geometrically, the torus can be visualized as a cross-section in the diagram below, with small radius ρ and large radius R .

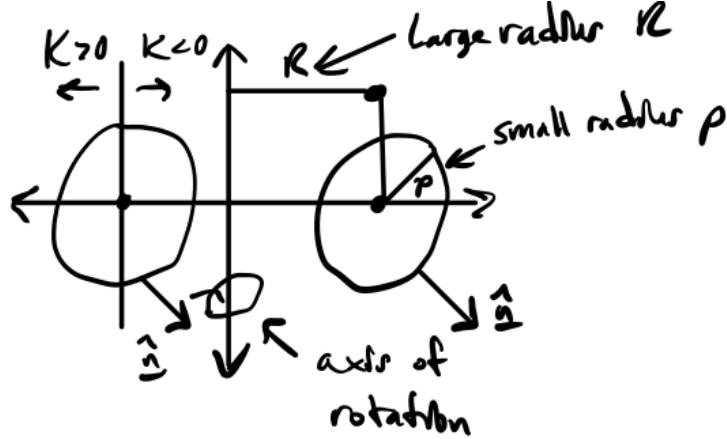


Figure 7: Geometric cross-section of the torus, along with its associated parameters and their significance that describe the torus.

What problems might we have with this object? It is **non-convex**. This Gaussian image may not be invertible when objects are non-convex. We lose uniqueness of mapping, as well as some other EGI properties, by using this non-convex object.

For the torus, the mapping from the object to the sphere is **not invertible** - each point in the EGI maps to two points on the object, which means the **surface normal** of the object is not unique.

The torus is convex into the board. Convex shapes/objects have non-negative curvature, and above we have a saddle point with negative curvature.

Keeping these potential issues in mind, let's press on. Mathematically:

$$\begin{aligned} r &= R + \rho \cos \eta \\ &= R + \rho \cos \left(\frac{s}{\rho} \right) \end{aligned}$$

From here, we can take the second derivative r_{ss} :

$$r_{ss} = \frac{d^2r}{ds^2} = -\frac{1}{\rho} \cos \left(\frac{s}{\rho} \right)$$

Combining/substituting to solve for curvature κ :

$$\kappa = \frac{-r_{ss}}{r} = \frac{1}{\rho} \frac{\cos \left(\frac{s}{\rho} \right)}{R + \rho \cos \left(\frac{s}{\rho} \right)}$$

We will also repeat this calculation for the corresponding surface normal on the other side. We will see that this results in different sign and magnitude:

$$\begin{aligned} r &= R - \rho \cos \eta \\ &= R - \rho \cos \left(\frac{s}{\rho} \right) \end{aligned}$$

From here, we can take the second derivative r_{ss} :

$$r_{ss} = \frac{d^2r}{ds^2} = \frac{1}{\rho} \cos \left(\frac{s}{\rho} \right)$$

Combining/substituting to solve for curvature κ :

$$\kappa = \frac{-r_{ss}}{r} = -\frac{1}{\rho} \frac{\cos \left(\frac{s}{\rho} \right)}{R + \rho \cos \left(\frac{s}{\rho} \right)}$$

Since we have two different Gaussian curvatures, and therefore two separate densities for the same surface normal, what do we do? Let's discuss two potential approaches below.

- Approach 1: Adding Densities:** For any non-convex object in which we have multiple points with the same **surface normals**, we can simply add these points together in density:

$$G = \frac{1}{\kappa_+} + \frac{1}{\kappa_-} = 2\rho^2$$

Even though this approach is able to cancel many terms, it creates a constant density, which means we will not be able to solve alignment/orientation problems. Let's try a different approach.

- Approach 2: Subtracting Densities:** In this case, let us try better taking into account local curvature by subtracting instead of adding densities, which will produce a non-constant combined density:

$$G = \frac{1}{\kappa_+} - \frac{1}{\kappa_-} = 2R\rho \sec(\eta)$$

This result is non-constant (better for solving alignment/orientation problems), but we should note that because of the secant term, it has a singularity at the pole. We can conceptualize this singularity intuitively by embedding a sphere within an infinite cylinder, and noting that as we approach the singularity, the mapped location climbs higher and higher on the cylinder.

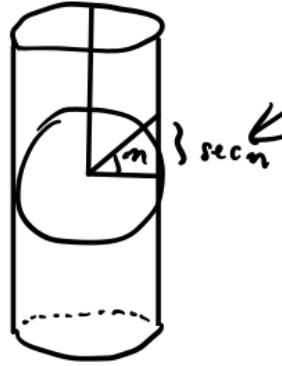


Figure 8: Embedding our sphere within a cylinder geometrically illustrates the singularity we observe for $\sec \eta$ as $\eta \rightarrow \frac{\pi}{2}$.

Note: Our **alignment** and tasks are done in sphere space, and because of this, we do not need to reconstruct the original objects after applying the EGI mapping.

1.4.3 Analyzing the Density Distribution of the Sphere (For Torus)

Whereas we had a direct mapping from “bands” to “bands” in the previous case, in this case our “bands” map to “crescents” on the sphere when computing the EGI of the torus.

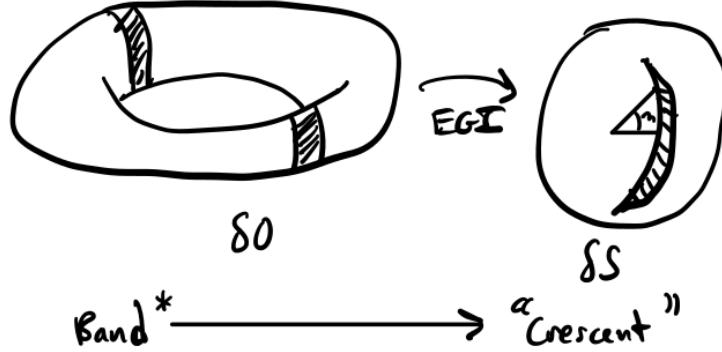


Figure 9: Computing the EGI of a torus, and demonstrating how we do not have the same “band to band” correspondence/mapping that we had in the previous case, suggesting that this mapping may not be invertible.

Note: On a torus, the band is slightly narrower on the inner side. Can balance out this asymmetry by using the band on the opposite side.

We can also consider the area of a torus:

$$\begin{aligned} A_{\text{torus}} &= (2\pi\rho)(2\pi R) \\ &= 4\pi^2 R\rho \\ &= \text{“A circle of circles”} \end{aligned}$$

As a result of how this area is structured, two “donuts” (tori) of different shape/structure but with the same area correspond/map to the same EGI, since EGI captures a ratio of areas. This loss of uniqueness is due to the fact that this torus representation still remains non-convex - in a sense this is the “price we pay” for using a non-convex representation.

Some other issues to be mindful of with a torus:

- It is hard to extract all surface normals since it is impossible to image all surfaces of a torus with just a single camera/camera pose. This phenomenon is also due to the non-convexity of the torus.

1.5 How Can We Compute EGI Numerically?

How can we compute EGI numerically through applications/frameworks such as data or with discretized ? We can look at the facets/pixels of objects.

Let us first suppose we have a triangular facet:

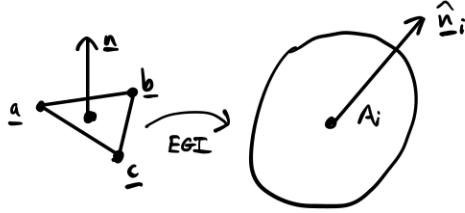


Figure 10: A triangular facet, which we can imagine we use to compute our EGI numerically.

As we have done before, we can find the **surface normal** and **area** of this facet to proceed with our EGI computation:

- **Surface Normal:** This can be computed simply by taking the cross product between any of the two edges of the triangular facet, e.g.

$$\begin{aligned} \mathbf{n} &= (\mathbf{b} - \mathbf{a}) \times (\mathbf{c} - \mathbf{b}) \\ &= (\mathbf{a} \times \mathbf{b}) + (\mathbf{b} \times \mathbf{c}) + (\mathbf{c} \times \mathbf{a}) \end{aligned}$$

- **Area:** This can be computed by recalling the area of a triangular ($\frac{1}{2}$ base \times height):

$$\begin{aligned} A &= \frac{1}{2}(\mathbf{b} - \mathbf{a}) \cdot (\mathbf{c} - \mathbf{a}) \\ &= \frac{1}{6}((\mathbf{a} \cdot \mathbf{a} + \mathbf{b} \cdot \mathbf{b} + \mathbf{c} \cdot \mathbf{c}) - (\mathbf{a} \cdot \mathbf{b} + \mathbf{b} \cdot \mathbf{c} + \mathbf{c} \cdot \mathbf{a})) \end{aligned}$$

We can repeat this area and surface normal computation on all of our facets/elements to compute the mass distribution over the sphere by adding different facets from all over surface. We add rather than subtract components to ensure we have a non-constant EGI, and therefore have an EGI representation that is suitable for alignment tasks.

1.5.1 Direction Histograms

Direction histograms are another way of thinking about how we build an EGI. These are used in different applications outside of machine vision as well, e.g. finding direction histograms in (i) muscle fibers, (ii) neural images, (iii) analyzing blood vessels.

Typically, these **direction histograms** can be used to ascertain what the most common orientations are amongst a set of orientations.

For histograms in higher dimensions, is subdividing the region into squares/cubes the most effective approach? As it turns out, no, and this is because the regions are not round. We would prefer to fill in the plane with disks.

To contrast, suppose the tessellation is with triangles - you are now combining things that are pretty far away, compared to a square. Hexagons alleviate this problem by minimizing distance from the center of each cell to the vertices, while also preventing overlapping/not filled in regions. **Other notes:**

- We also need to take “randomness” into account, i.e. when points lie very close to the boundaries between different bins. We can account for this phenomena by constructing and counting not only the original grids, but additionally, shifted/offset grids that adjust the intervals, and consequently the counts in each grid cell.

The only issue with this approach is that this “solution” scales poorly with dimensionality: As we increase dimensions, we need to take more grids (e.g. for 2D, we need our (i) Original plane, (ii) Shifted x , (iii) Shifted y , and (iv) Shifted x and shifted y). However, in light of this, this is a common solution approach for mitigating “randomness” in 2D binning.

1.5.2 Desired Properties of Dividing Up the Sphere/Tessellations

To build an optimal representation, we need to understand what our desired optimal properties of our tessellations will be:

1. **Equal areas of cells/facets**
2. **Equal shapes of cells/facets**

3. “Rounded” shapes of cells/facets
4. Regular pattern
5. **Allows for easy “binning”:** As an example a brute force way to do binning for spheres is to compute the dot product with all surface normals and take the largest of these. This is very impractical and computationally-inefficient because it requires us to step through all data each time. For this reason, it is important we consider implementations/representations that make binning efficient and straightforward.
6. **Alignment of rotation:** The idea with this is that we need to bring one object into alignment with the other in order to do effective recognition, one of the key tasks we cited.

For instance, with a dodecahedron, our **orientation/direction histogram** is represented by 12 numbers (which correspond to the number of faces on the dodecahedron):

$$[A_1, A_2, \dots, A_{12}]$$

When we bring this object into alignment during, for instance, a recognition task, we merely only need to permute the order of these 12 numbers - all information is preserved and there is no loss from situations such as overlapping. This is an advantage of having alignment of rotation.

$$[A_7, A_4, \dots, A_8]$$

Platonic and **Archimedean** solids are the representations we will use for these direction histograms.