Computational Geometry Lab:
TRIANGLES

John Burkardt
Information Technology Department
Virginia Tech
http://people.sc.fsu.edu/~jburkardt/presentations/cg_lab_triangles.pdf

December 23, 2010

1 Introduction to Computational Geometry

This is the first lab in a short series, which introduce topics from *Computational Geometry* and ask you to try to implement some simple algorithms associated with various tasks.

Many people need to use methods from computational geometry, but it's rare to find an organized course or textbook on the subject. And yet almost every scientific problem has a geometric aspect which must be discretized, simulated and analyzed. Moreover, even abstract problems which don't seem to have any geometric connection can often be understood using generalizations of ideas from the geometry of two and three dimensions.

Because it explores the ways of representing and analyzing surfaces, computational geometry is enormously important in computer-aided design and modeling, computer graphics and animation. But computational geometry is also a fundamental tool for the approximate solution of the partial differential equations that describe heat, fluid flow, chemical reactions, and other physical processes in which geometry obviously plays a role.

Although we can start by relying on axioms from Euclid and methods of measurement that were familiar to the ancient Egyptians, doing geometry on a computer requires learning some new ideas and tools. It is the purpose of these labs to introduce concepts of computational geometry in a way that allows the student to learn in part by programming the ideas that are presented.

Some of the fundamental ideas of computational geometry include *orientation*, *inclusion*, *distance*, *sampling*, *measure* (of area, of angle, of distance, of volume), *searching*, *connectedness*, *approximate integration*, and *boundaries*.

2 Introduction to this Lab

Most of the labs in this series will focus on problems involving two-dimensional regions. The first thing we will do to these regions is form a triangulation, that is, we will try to understand the complex region by thinking of it as a collection of simple triangles.

Of course triangles are simpler, but the point is that once we reach these shapes, we can actually compute facts about them, and these facts will tell us something about the original complicated shapes with which we began. It is relatively straightforward to compute many quantities associated with a triangle.

Some of the quantities we will look at, such as area, are obviously important. You probably know a formula for computing the area of a triangle, but do you know how to implement it in a computer? Other quantities such as the circumradius, are obscure enough that you may never even have heard of them. But we will consider a few of these more unusual quantities as well, since they will turn out to be useful in later computations.

3 Naming Things on a Triangle

It will be helpful if we can come up with a simple and short way of describing a triangle and various quantities associated with it.

Let's start by supposing we've chosen a name for a particular triangle, which might be simply **T1** or **Tri** or **Tex2**. Every triangle has three vertices. We could number them, but for now it might be more convenient to identify them by letter. So the vertices will be named **Va, Vb, Vc**. Sometimes, we may be considering more than one triangle, and so we will optionally specify the "full name" of a vertex by beginning with the name of the triangle that it belongs to. Thus, if we need to avoid ambiguity, we can more fully specify the second vertex of triangle **T1** as **T1.Vb**.

Now each vertex is a point in two dimensional space, so it has coordinates. We can specify a particular coordinate by appending **.x** or **.y** to the name of the vertex, as in **T1.Vb.x**, which refers to the x coordinate of the second vertex of triangle T1.

When we come across other properties of a triangle, such as area, we may use a similar notation, such as **Tri.Area**.

4 Defining a Triangle

We begin by assuming that we have defined a triangle **T**, which means that we know the x and y coordinates of its vertices **Va, Vb, Vc**. Notice that there are six different ways to list the three vertices; we will always choose to list the vertices in such a way that they move in a counterclockwise (CCW) path around the triangle. This is called choosing the **orientation** of a triangle. When we get to triangulations, we will see that choosing a consistent orientation for our triangles is vital to guarantee that some of our computations work correctly.

We can think of a representation of the information defining our triangle as a list of the vertices, with each vertex being a list of the coordinates. We will use curly braces to collect the elements of a list. Thus, the definition of a triangle might be thought of as:

```
T = {  T.Va,               T.Vb,               T.Vc }
  = { {T.Va.x, T.Va.y}, {T.Vb.x, T.Vb.y}, {T.Vc.x, T.Vc.y} }
```

For example, the example triangle Tex1 in Figure 1 can be described as

```
Tex1 = { {4,1}, {8,3}, {0,9} }
```

5 The Lengths of the Edges of a Triangle

Let's get some practice with our triangle notation by describing how to determine the lengths of the edges of the triangle. It will help us in our formulas if we give names to the edges. Since each edge is defined by two vertices of the triangle, appropriate names might be **Sab, Sbc, Sca**. (There is a different convention which names the edge using the capital letter whose lower case version is the opposite vertex.)

To find the length of edge **Sab**, we compute the *Euclidean length* of the vector from vertex **Va** to **Vb**, that is:

$$\text{Sab.length} = \sqrt{(Vb.x - Va.x)^2 + (Vb.y - Va.y)^2}$$

In particular, for triangle Tex1, we have

$$\text{Tex1.Sab.length} = \sqrt{(8-4)^2 + (3-1)^2} = \sqrt{16+4} = \sqrt{20} \approx 4.47$$

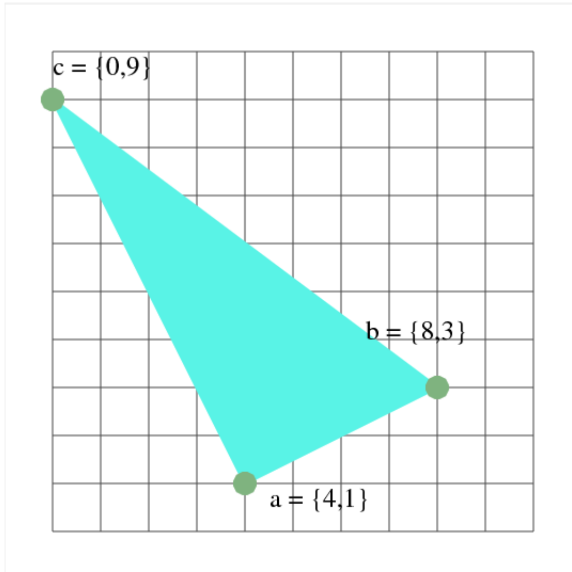

Figure 1: Tex1, Triangle Example 1.

6 The Angles of a Triangle

If you have the lengths of all the edges of a triangle, then you have enough information to compute the angles, using the law of cosines. (The law of sines is less useful, since it gives two choices for the angle!).

We need some new terminology here. Since an angle is associated with a node, we can represent the three angles of the triangle with the names {Aa, Ab, Ac}. In that case, we can write the law of cosines for angle **Ac** as

$$Sab^2 = Sbc^2 + Sca^2 - 2\,Sbc\,Sca\,\cos(Ac)$$

which can then be solved for the angle:

$$\mathrm{Ac} = \cos^{-1}\frac{Sbc^2 + Sca^2 - Sab^2}{2\,Sbc\,Sca}$$

The built in formulas for angle functions will usually work in radians. When printing such angles, it is often helpful to convert to the more familiar degree measurement, which can be done by multiplying by $\frac{180}{\pi} \approx 57.295$ degrees per radian.

In particular, for triangle Tex1, we have

$$\mathrm{Tex1.Ac} = \arccos\frac{10^2 + 80 - 20}{2 \cdot 10 \cdot \sqrt{80}} \approx 0.463 \text{ radians} \approx 26.5 \text{ degrees}$$

7 Program #1: Basic Triangle Measurements

Write a simple program that can read in the definition of a triangle, and which then computes:

- the lengths of the sides;
- the angles measured in radians;
- the angles, measured in degrees;

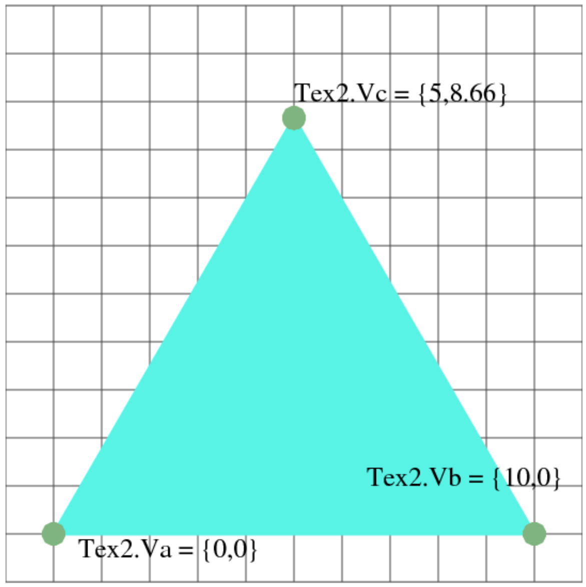

Figure 2: Tex2, Triangle Example 2.

- the sum of the angles, measured in degrees.

Test your program on triangle Tex1. Assume that the input is in a text file called "tex1.txt" of the form:

```
4.0  1.0
8.0  3.0
0.0  9.0
```

Repeat your tests on triangle Tex2, which has the definition:

```
 0.00  0.00
10.00  0.00
 5.00  8.66
```

Does your program detect something interesting about this triangle?

8 The Area of a Triangle

We have to thank Heron, an ancient Greek, for a simple formula for the area of a triangle which only involves the lengths of sides. Heron's formula is

$$\text{Area} = \frac{1}{4}\sqrt{(Sab + Sbc + Sca)(-Sab + Sbc + Sca)(Sab - Sbc + Sca)(Sab + Sbc - Sca)}$$

9 The Signed Area and the Orientation of a Triangle

You may remember from vector algebra something called the *cross product*. In the special case of a pair of planar vectors **u** and **v**, this could be regarded as computing the area of a parallelogram for which **u** and **v** form two of the sides. The formula for this scalar value is

$$\text{Cross Product}(u, v) = u \times v = u.x \cdot v.y - v.x \cdot u.y$$

This implies a formula for the area of a triangle:

$$\begin{aligned}\text{Signed Area } = \frac{1}{2} \quad (\quad & Va.x \cdot (Vb.y - Vc.y) \\ + \quad & Vb.x \cdot (Vc.y - Va.y) \\ + \quad & Vc.x \cdot (Va.y - Vb.y) \,)\end{aligned}$$

The area that is returned from this formula is “signed”; it is negative if the vertices are given in clockwise order.

This means that this formula also gives us the **orientation** of the triangle. That is, if the signed area is negative, then the vertices have been given in clockwise order.

10 The Signed Area as a Determinant

There is another way to represent the formula for the (signed) area of a triangle which we displayed above. The reason for rewriting the formula is in part to suggest how this formula for the area of a triangle will be extended to compute the volume of a tetrahedron, or the hypervolume of a hypersimplex. Given how powerful this generalization is, it is surprising that the formula is so simple:

$$\text{Signed Area } = \frac{1}{2}\begin{vmatrix} Va.x & Va.y & 1 \\ Vb.x & Vb.y & 1 \\ Vc.x & Vc.y & 1 \end{vmatrix}$$

Because of the properties of determinants, it is now easy to understand how the formula produces a signed area that is sensitive to the ordering of the vertices. The generalization to tetrahedral volume is almost obvious (the initial factor becomes $\frac{1}{6}$ and knowing that, it’s easy to guess why and to guess what happens in 4 dimensions!).

11 Program #2: Computing Area and Orientation

Write a simple program that reads a triangle and computes:

- the area of the triangle using Heron’s formula.
- the signed area of the triangle.
- the orientation of the triangle, “positive” or “negative”.

Test your program on triangle Tex1. Then switch the order of two of the vertices on triangle Tex1 and rerun your program. Which of your results stay the same?

12 The Centroid of a Triangle

The centroid of a triangle can be thought of as the center of mass. If you had a sheet of plastic in the shape of the triangle, then the sheet would balance at the centroid. Any line through the centroid divides the triangle into two pieces of equal area. The centroid can be found as the common intersection of the lines that connect each vertex with the midpoint of the opposite side.

The formula for the centroid is refreshingly easy, since it’s just the average of the three vertices:

$$\text{Centroid} = \frac{Va + Vb + Vc}{3}$$

Here, of course, we mean that both the x and y coordinates of the centroid are computed by averaging the corresponding coordinates of the vertices.

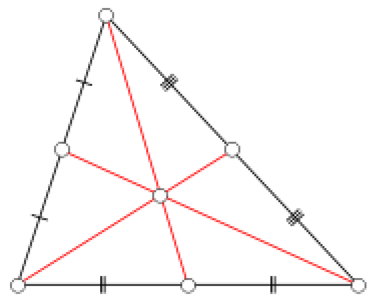

Figure 3: The Centroid of a Triangle

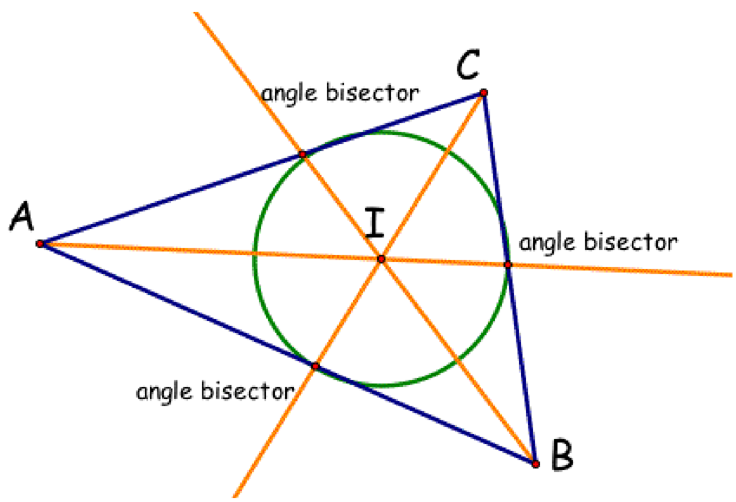

Figure 4: The Inscribed Circle of a Triangle

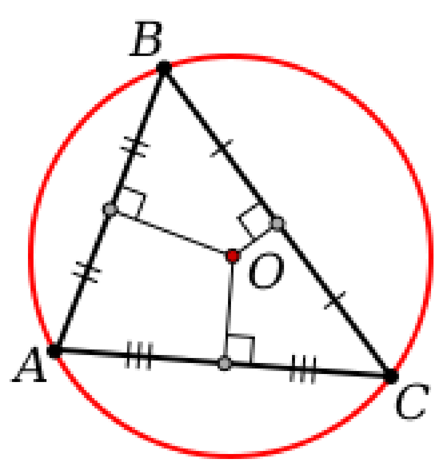

Figure 5: The Circumcircle of a Triangle

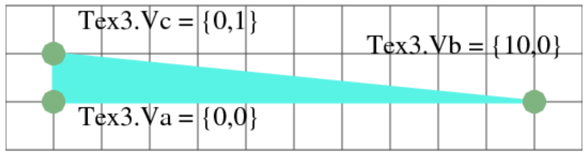

Figure 6: Tex3, Triangle Example 3.

13 The Inscribed and Circumscribed Circles of a Triangle

At a later point we will need to evaluate the "shape" of a triangle. Our evaluation will favor equilateral triangles, and penalize triangles whose sides vary greatly in length. One way to come up with a numerical measurement of shape is to compare the radiuses of the inscribed and circumscribed circles of a given triangle.

The inscribed circle is the largest circle that can be contained within a triangle; the center of this circle is found at the common intersection of the angle bisectors. The circumscribed circle is the smallest circle that contains the triangle; it does this by passing through all three vertices. Interestingly, the center of the circumscribed circle can lie outside of the triangle! Except for degenerate cases, these circles are unique.

Without further fuss, we will simply present formulas for computing the radiuses of these two circles for a given triangle, so that we may refer to these results when they are needed.

$$\text{Inradius} = \frac{1}{2}\sqrt{\frac{(-Sab+Sbc+Sca)(Sab-Sbc+Sca)(Sab+Sbc-Sca)}{Sab+Sbc+Sca}}$$

$$\text{Circumradius} = \frac{Sab\,Sbc\,Sca}{\sqrt{(Sab+Sbc+Sca)(-Sab+Sbc+Sca)(Sab-Sbc+Sca)(Sab+Sbc-Sca)}}$$

The attentive reader will notice a relationship between these definitions and Heron's area formula!

For an equilateral triangle, the circumscribed circle has exactly twice the radius of the inscribed circle. We can verify this by setting all sides equal to S, from which we get the relationship:

$$\text{Equilateral.Circumradius} = 2 \cdot \text{Equilateral.Inradius} = \frac{S}{\sqrt{3}}$$

which is the closest these two quantities can be. When it is time to develop a measure of "well-shaped" triangles, it is the ratio of inradius to circumradius that we will be examining.

14 Program #3: More Triangle Measurements

Write a program that reads a triangle and computes:

- the centroid;
- the radius of the inscribed circle (the "inradius");
- the radius of the circumscribed circle (the "circumradius");
- Q, which is 2 times the ratio of the inradius to the circumradius.

Test your program on triangles Tex1, Tex2 and Tex3. Triangle Tex3 is defined by

```
 0.0  0.0
10.0  0.0
 0.0  1.0
```

In theory, the value of Q will be between 0 and 1; It should be exactly 1 for an equilateral triangle, but will tend to 0 if the triangle sides vary greatly in length, Do your results agree with this claim?