CS380 Introduction to Computer Graphics
Homework #3

20160042 Inyong Koo

Describe the following terms with respect to computer graphics.

1. Point

   Point is one of the fundamental geometric objects. A point indicates only location.

2. Affine sum

   In an affine space, the addition of two vectors, the multiplication of a vector by a scalar, and the addition of a vector and a point are defined, the addition of two aribitrary points and the multiplication of a point by a scalar are not. However, we can express point P upon $\overline{RQ}$ using following equation with latter two operations. We call such operation the '**Affine sum**'
   $P = \alpha_1 R + \alpha_2 Q$ where $\alpha_1 + \alpha_2 = 1$

3. Convex hull

   The **convex hull** is the set of points formed by the affine sum of n points. $P = \alpha_1 P_1 + \alpha_2 P_2 + \cdots + \alpha_3 P_3$ where $\alpha_1 + \alpha_2 + \cdots + \alpha_n = 1$, under additional restriction $\alpha_i \geq 0, i = 1, 2, \cdots, n$
   Geometrically, the convex hull is the set of points that we form by stretching a tight-fitting surface over the given set of points – shrink-wrapping of the points.

4. Barycentric coordinate

   We can express a plane by expressing coordinate of a point in the plane using two nonparallel vectors. Using concept of affine sum, it means a point on a plane can be expressed using three points, not on a single line, of the plane. $T = \alpha P + \beta Q + \gamma R$ where $\alpha + \beta + \gamma = 1$. The representation of a point by (α, β, γ) is called its **barycentric coordinates** representation.

5. Tessellation

   Since triangular polygons are always flat, either the modeling system is designed to always produce triangles, or the graphics system provides a method to divide an arbitrary polygon into triangular polygons. We call this division into triangles, **tessellation**.

6. CSG (Constructed solid geometry)

   Constructive solid geometry (CSG) is the major exception to triangle mesh approximation approach to provide curved objects. In CSG system, we build objects from a small set of volumetric objects through a set of operations such as union and intersection. It is an excellent approach for modelling, but rendering CSG models is more difficult than is rendering surface-based polygonal models.

7. Homogenous coordinates

   Common three-dimensional representation of coordinate can cause confusion between representation of a point and a vector. **Homogenous coordinates** avoid this difficulty by using a four-dimensional representation for both points and vectors in three dimensions.
   Specified by (v_1, v_2, v_3, P_0), we can represent a point by $P = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 + P_0$. P is then represented by $(\alpha_1, \alpha_2, \alpha_3, 1)$ In the same frame, a vector can be represented by $w = \delta_1 v_1 + \delta_2 v_2 + \delta_3 v_3$. W is then represented by $(\delta_1, \delta_2, \delta_3, 0)$.

8. Frames (compared to coordinates)

   A **frame** is determined by an origin and basis vectors. Instead of using only basis vectors as in coordinates, notation of a frame allows us to avoid the difficulties caused by vectors having magnitude and direction but no fixed position. We are also able to represent points and vectors in a manner that will allow us to use matrix representations but that maintains a distinction between the two geometric types.

9. Normalized device coordinates

   Normalized device coordinates is a coordinate system occurred during the pipeline, after the clip coordinates are divided by the w component, called perspective division, into three-dimensional representation. The normalized device coordinates later take into account the viewport, and create window coordinates.

10. Affine transformations

    Affine transformations are the transformation takes place in affine space, homogenous coordinates. Transformation is a linear function that has 12 degrees of freedom. Line is conserved in affine transform so if we want to transform a line segment, we need only to transform the endpoints. Rotation, translation, scaling and shearing are the affine transformations.

11. Rigid-body transformation

    Rigid-body transformations are transformations that does not alter the shape or volume of an object, but only the object's location and orientation. (Rotation and translation)

12. Shearing

    Shearing is one of the affine transformations that changes shape of the object, but not its volume. Another affine transformation that is not rigid-body transformation is scaling, which changes volume of the object but not its shape.

13. Uniform variables

    When we send vertex attributes to a shader, these attributes can be different for each vertex in a primitive. We may also want parameters that will remain the same for all vertices in a primitive or equivalently for all the vertices that are displayed when we execute a function. Such variables are called uniform variables.

14. Quaternion

    Quaternions are an extension of complex numbers that provide an alternative method for describing and manipulating rotations.

15. Zero vector

    Zero vector 0 is a vector that satifies $u + \mathbf{0} = u$ for $\forall u \in V$.

16. Basis

    If a vector space has dimension n, any set of n linearly independent vectors form a **basis**.
    If $v_1, v_2, \cdots, v_n$ is a basis for V, any vector v can be expressed uniquely in terms of the basis vectors as $v = \beta_1 v_1 + \beta_2 v_2 + \cdots + \beta_n v_n$.