Lecture Notes for CMU 15-462: Computer Graphics

Emulie Chhor

May 13, 2021

Introduction

Courses Notes for CMU 15-462: Computer Graphics by Keenan Crane

1 Course Overview

Overview

1. What is Computer Graphics
2. What do we learn in Computer Graphics
3. What is Modelization
4. What is Rendering
5. Projects throughout the course

1.1 What is Computer Graphics

The goal of computer graphics is to simulate sensory information by turning digital information into sensory stimuli (visual, sound, texture). Its "inverse" would be the field of computer vision, whose goal is to translate sensory information in our environment into digital information.

1.2 What do we learn in Computer Graphics

There are two phases to learning computer graphics:

1. Theory: What are the techniques used to simulate sensory stimuli
2. Systems: How can we simulate those stimuli efficiently

Theory

1. Representation: How do we encode shape and motion?

2. Sampling and Aliasing: How can we acquire and reproduce signal
3. Numerical Methods: How can we modify signal
4. Radiometry and Light Transport: How can we reproduce light behavior
5. Perception: What can we learn from psychology and physiology to enhance human perception

Systems

1. Parallel and Heterogenous Processing
2. Graphics-Specific Programming Languages

1.3 What is Modelization

The first concept to understand in computer graphics is that we can produce simuli in two steps:

1. Modelization: How can we describe object to a computer
2. Rasterization: How will the computer translate our model into sensory stimuli

Usually, modelization is done by

1. Listing a bunch of points
2. Linking these points with a list of edges

However, we often have to perform some transformations to accurately represent the object. For instance, we need to find a way to represent 3D object onto a 2D shape.

How to convert 3D points into 2D shapes

One solution is to use perspective to our advantage. We know that object further from us are smaller, so we can use the "pinhole camera" from optical physics to realize that we have rectangular triangle, which allow us to use pythagoras to form a ration between the original points onto our 2D shape. If we have a 3D point (x,y,z) that we want to translate onto a 2D shape (u,v), we

1. substract the camera from the 3D point
2. Divide (x,y) by z to get our 2D point (u,v)

1.4 What is Rendering

To display our points on the computer, we first have to understand how the computer screen works. A computer screen is made of pixels, which are little square that contains a value which tells us how much green, red and blue should be in the square.

Often, we talk about raster display because we take a continuous object and represent it with a discrete mapping (pixels grid)

Which Rasterization Techniques to use: diamond rule

Since our shapes aren't alway rectangular, we must decide which pixels to turn on and off. There exist a bunch of rasterization techniques, but we often make up our own based on our needs:

1. Turn on all pixels that the vector touches
2. Diamond Rules: Turn pixel only if it passes through the diamond shape inside the pixel

How to color the pixel

Like with rasterization, there are several ways to color each pixel, but we often like to color it using the ratio of the vector inside the pixel: the more a vector is inside a pixel, the more we make the color thicker

How does the computer find vector: incremental rasterization

A naive solution to find which pixels to color would be to check each pixel individually, which takes $O(n^2)$ for O(n) pixels. A better method would be to use incremental rasterization. We

1. Calculate the slope of the line and add it to the y-value to find the pixels inside the x-range of the vector
2. Compute the pixel value
3. Draw

1.5 Projects throughout the course

The course consists of building our own library

1. Rasterization
2. Geometric Modeling
3. Photorealistic Rendering
4. Motion and Animation

2 Linear Algebra

Overview

1. What is Linear Algebra
2. Geometric and algebraic interpretation of Vector Spaces
3. Defining what is a linear map

4. Span, Basis and Orthonormal Basis
5. System of Linear Equations
6. Matrices in Linear Algebra

2.1 What is Linear Algebra

L'algèbre linéaire est la branche de mathématique qui étudie les espaces vectoriel. Il est important de maitriser ces notions puisqu'elles permettent d'abstraire certains concepts géométriques algébriquement, ce qui nous permet de simplifier nos calculations.

2.2 Geometric and algebraic interpretation of Vector Spaces

Qu'est-ce qu'un vecteur

Géométriquement, on peut visualiser un vecteur comme une flèche ayant une direction et une longuer. Ce vecteur se situe dans un espace qu'on nomme espace vectoriel. Dans cet espace, ce vecteur possède certaines propriétés qu'on nomme axiomes.

Pourquoi étudier les espaces vectoriels

Habituellement, l'interprétation géométrique et algébrique des espaces vectoriels se fait assez bien. On veut donc identifier ces propriétés afin de pouvoir généraliser les espaces vectoriels non pas uniquement sur des vecteurs, mais sur des structures plus compliquées tel que des polynômes, fonctions, matrices, etc.

Ainsi, puisqu'on sait qu'un vecteur est caractérisé par sa norme et sa direction, on veut pouvoir généraliser ces notions et interpréter géométriquement ces objets.

2.2.1 Axiomes d'un espace vectoriel

Un espace vectoriel possèdent les 10 propriétés suivantes (axiomes):

1.
2.
3.
4.
5.
6.
7.
8.

9.

Dans des cours de maths plus avancés, on verra que les espaces vectoriels possèdent ces caractéristiques puisque ce sont des fields/ring

De plus, il est utile d'interpréter ces propriétés géométriquement et algébriquement. C'est donc un bon exercices de voir pourquoi, par exemple, l'addition de vecteurs est commutatif géométriquement et algébriquement.

2.2.2 Interprétation géométrique et algébrique de la norme

Cette sous section focusse sur les propriétés de la norme d'un objet dans un espace vectoriel. Ici, on se penchera sur la norme d'un vecteur et d'une fonction, norme qu'on nomme L^2. Notre but sera donc de

1. Calculer la norme d'une vecteur et d'une fonction
2. Interpréter géométriquement et algébriquement les propriétés de la norme d'un vecteur et d'une fonction

Calcul de la Norme

La norme d'un vecteur est sa longueur, alors que la norme L^2 d'une fonction est l'aire sous sa courbe. Notons qu'on peut interpréter une intégrale comme une somme (penser à Riemann)

1. Norme d'un vecteur: $|u| = \sqrt{(}\sum_{i=1}^{n} u_i^2)$
2. Norme d'une fonction (L^2) norm: $||f|| := \sqrt{(}\int_0^1 f(x)^2\,dx)$

2.2.3 Interprétation géométrique et algébrique des propriétés de la norme

1. $|u| \geq 0$
2. $|u| = 0 \Longleftrightarrow u = 0$
3. $|cu| = |c||u|$
4.
5. Inégalité du triangle: $|u| + |v| \geq |u + v|$

2.3 Interprétation géométrique et algébrique du produit scalaire

Le produit scalaire mesure si 2 vecteurs (ou fonction) se dirige vers la même direction (leur alignement). Encore une fois, on veut

1. Calculer le produit scalaire de deux vecteurs/fonctions
2. Interpréter géométriquement et algébriquement les propriétés du produit scalaire

2.3.1 Calcul du produit scalaire

1. Produit Scalaire d'un vecteur: $< u, v >:= \sum_{i=1}^{n} u_i v_i$
2. Produit Scalaire d'une fonction: $<< f, g >>:= \int_0^1 f(x)g(x)\, dx$

2.3.2 Interprétation géométrique et algébrique des propriétés du produit scalaire

1. Symmétrique: $< u, v >=< v, u >$
2. Projection and scaling
3. If we scale the vector, the scalar product is also scaled: $< cu, v >= c < u, v >$
4. a vector is aligned with itself: $< u, u >= 1$ et $< u, u >\geq 0$

Edges Detection using Derivatives

Parfois, la norme et le produit vectoriel d'une image ou d'un singal peuvent être trompeur. Par exemple, si on interpréte la norme d'une image comme sa "brillance", alors il se pourrait que notre algorithme nous indique qu'il y ait plus d'information dans une image d'un ciel que dans une photo d'un chien. Pour ce faire, on peut redéfinir notre définition de la norme comme étant la dérivée, et ainsi détecter les edges.

2.4 Defining what is a linear map

2.4.1 Pourquoi travailler avec des transformations linéaires

On veut travailler avec des transormations linéaires parce que

1. travailler avec des transformations non-linéaires est un problème NP, très difficile à computer
2. Les transformations usuelles (rotations, translation, ..) sont des transformations linéaires
3. Puisque l'addition et la multiplication sont définies de la même façon avant et après transformations, on peut faire des approximations de Taylor: on peut approximer des fonctions transcendantes avec des fonctions polynômiales.

Plus tard, on verra qu'on peut interpréter les matrices comme des transformations linéaires.

Is it a linear function

On peut déterminer si une transformation est linéaire de 2 façons: algébriquement et géométriquement.

Géométriquement:

1. L'origine part à la même place
2. Les lignes restent des lignes

Algébriquement:

1. l'addition est définie de la même façon avant et après transformation:
2. la multiplication est définie de la même façon avant et après transformation:

Ainsi, la fonction $f(x) = ax + b$ n'est pas une transformation linéaire, puique géométriquement, elle ne mappe pas les vecteurs à la même origine, et algébriquement, l'addition n'est pas définie de la même façon avant et après transformation. Intuitivement, c'est comme si on disait que l'ordre dans lequel on performe nos transformation change le output, ce qui n'a pas de sens.

2.5 Span, Basis and Orthonormal Basis

Span (Espace générateur)

On peut interpréter un espace générateur de la façon suivante: c'est l'ensemble de vecteurs qu'on peut obtenir en faisant la combinaison linéaire de vecteurs d'un ensemble.

La notion de span est importante, car on veut pouvoir déterminer si on notre ensemble de vecteurs est assez grand pour générer les vecteurs de l'espace

Basis

La base d'un espace vectoriel est simplement l'ensemble des vecteurs nécessaires pour construire tous les vecteurs d'un espace. On veut pouvoir générer tous les vecteurs, sans en avoir trop. La base doit être:

1. Linéairement indépendant: on a un nombre minimal de vecteurs pouvant générer ie il existe une seule façon de construire le vecteur
2. Générateur: on peut faire une combinaison linéaire

Il faut noter que lorqu'on additionne, multiplie des vecteurs, il faut toujours s'assurer qu'on travaille dans la même base.

Orthonormal Basis

En computer graphics, on veut travailler avec des bases orthonormales, car la longueur reste la même

1. Comment déterminer si la base est orthogonale? les paires des vecteurs sont orthogonales entre elles
2. Comment transformer une base quelqconque en base orthogonale? Gram-Schmidt ou QR Decomposition

Fourier Transform

Les fouriers transformations et les fourier decomposition sont utilisés pour décomposer les signaux sur des bases sinusoidales $cos(nx), sin(mx)$ en projettant les fonctions (se répètent sur des période 2 pi) sur différentes fréquences.

2.6 System of Linear Equations

Une autre utilisation de l'algèbre linéaire est p/r à la résolution de système linéaire. On veut résoudre les systèmes linéaire, car ils nous disent si on a des intersections (points, droites, plans, ...)

Types de solutions

Encore une fois, on peut interpréter les solutions à un SEL algébriquement et géométriquement.

1. Aucune solution
   - Algébriquement: contradiction [0 0 .. 0 — k]
   - Géométriquement: il n'y a pas d'intersections
2. Solution Unique
   - Algébriquement: toutes les variables sont associées à un pivot
   - Géométriquement: il y a une seule intersection point/droite
3. Infinité de solutions
   - Algébriquement: variable libre
   - Géométriquement: superposition des droites/plans

Notons que les SEL sont aussi associés à l'équation Ax=b, qui nous permet de trouver la préimage x de b, c-à-d le vecteur original avant d'avoir appliquer la matrice de transformation A sur le vecteur b

2.7 Matrices in Linear Algebra

On a peu parler des matrices, car elles ont plusieurs interprétations géométriques, ce qui peut nous mélanger dans la compréhension.

TODO

3 Vector Calculus

NEXT