تفريغ مقرر بيانيات الحاسب| BCG601

Slash team

برنامج الهندسة المعلوماتية – الجـــــامعة الافتراضية السورية

تفريغ الطالب/ـــة:	ميادة ميرو
المدرّس المحاضر:	د. فاتن مشتا
عنوان المحاضرة	**Geometric modeling**
رقم المحاضرة	10
تاريخ بث المحاضرة	2022/12/31

GEOMETRIC MODELING

مقدمة:

المشهد لدينا قد يتكون من عدة أغراض أو model فكيف يتم رسم هذه الأشكال هندسياً؟

أي شكل من هذه الأشكال نقوم بتقسيمه الى عدة مضلعات (شيكة من المضلعات) للحصول على الشكل المطلوب تقريباً.

الأمور التي يجب الانتباه لها رياضياً:

1. ماهي طريقة تمثيل هذا الغرض.
2. ما مدى سهولة تغيير المشهد لتحويلة الى صورة render.
3. كم تكلفة عملية تخزين المشهد ونقله وحجم الذاكرة المطلوب.
4. سهولة رسمه سواء باليد أو بواسطة برامج معينة.
5. قدرته على التفاعل (بواسطة التحويلات الهندسية).
6. كمية التعقيد من العمليات الحسابية والمعادلات الرياضية.

يوجد عدة برامج تساعد في رسم الأغراض الموجودة في المشهد مثل CAD, CAM وتم تطويرها عدة مرات نتيجة لتطوير CAGP(computer aided geometric design)

وهي أداة هندسية تدعم أساس رياضي لوصف ومعالجة الأشكال الهندسية.

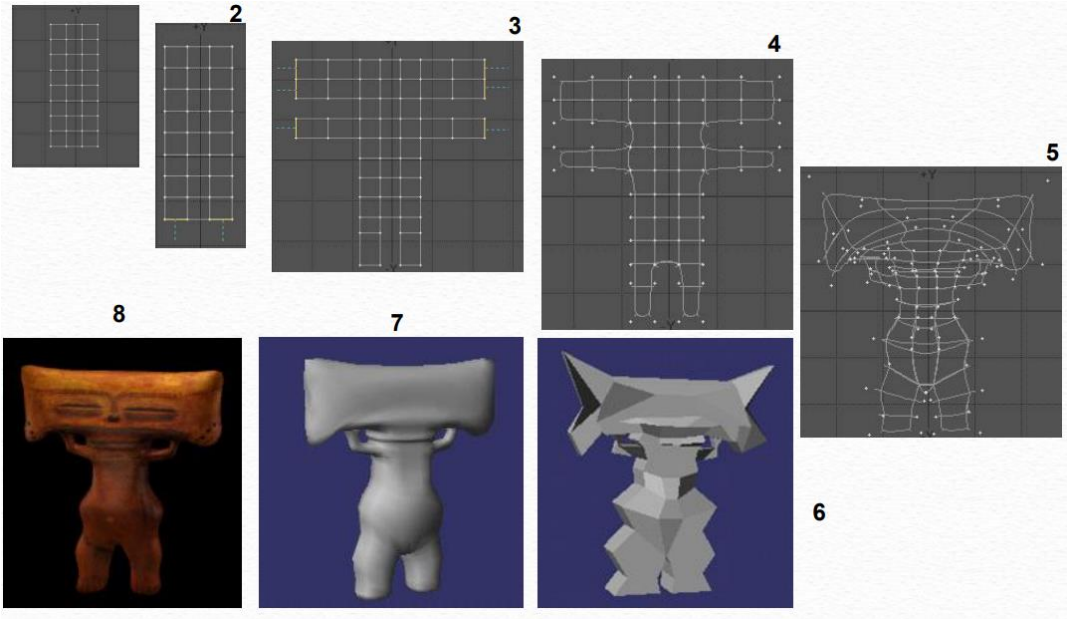
مصطلح Geometric modeling يشمل كل من

- Curve modeling المنحنيات
- Surface modeling السطوح

SURFACE MODELING

خطوات تصميم شكل ما تكون كالتالي:

يتم إضافة نقاط تحكم للحصول على الإنحناءات ومن ثم تطبق خوازمية محددة لتقليل التكسرات.



- غالبا نهتم فقط في سطح الغرض ولكن في بعض الأحيان نحتاج للاهتمام بالفراغ الذي يشغله الغرض والمكون الداخلي للغرض (مثل البنية الداخلية للعظام).
- يوجد طريقتين لتمثيل الغرض أو الmodel:

1. Parametric

المعادلات التي تأخذ وسيط نتنج كل النقاط التي تكون surface or volume بتغيير قيمة الوسيط.

2. Implicit

هي المعادلات الضمنية تخبرنا في حال كانت النقطة surface or volume

معادلة كرة ضمن الفراغ:

$$x^2 + y^2 + z^2 - 1 = 0$$

- التقنيات المستخدمة لتمثيل السطح الخارجي للغرض والتي تدعى polygon meshes (أي شكل يمكن تقسيمه لعدة مضلعات تسمة شبكة المضلعات) ستحتاج الى معادلات المستوي وطريقة حساب الناظم على المستوي.

POLYGON MESHES

يمكن تحويل أي غرض الى مجموعة من المضلعات

عملية ال render أي عملية التحويل الى مضلعات عملية تتم بسهولة وسرعة.

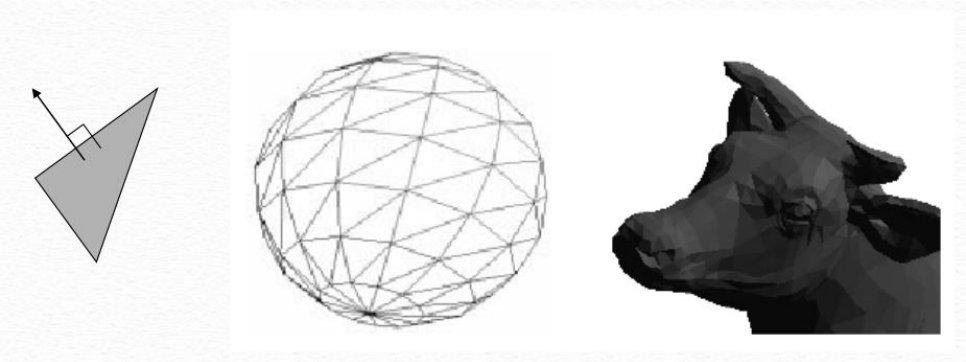
العديد من العمليات (مثل عمليات الهندسية) تتم بشكل سهل باستخدام المضلعات.

حجم الذاكرة اللازم لتمثيل شبكة المضلعات قليل.

مواصفات شبكة المضلعات polygon meshes

1. مترابطة: إذا كان أي رأسين بينهما حافة.
2. بسيطة: إذا لم تكن تحتوي على ثقوب.
3. مستوية: إذا كان كل مضلع هو مستوي وكل سطح هو مضلع مستوي.
4. محدودة: إذا كان رسم خط بين أي نقطتين من الشبكة يكون خط ينتمي لها.

مثال:



نقسم سطح الغرض الى مضلعات سواء مثلثات أو أشكال رباعية.

كلما زاد عدد المضلعات وصغر حجمها كلما زادت الدقة.

مثال:

في هذه الصورة نلاحظ أن حجم المضلعات كبير بالتالي الدقة منخفضة.

مثال:

بينما في هذه الصورة نلاحظ أن حجم المضلعات أصغر بالتالي الدقة أعلى والتفاصيل أوضح.

كيفية توصيف شبكة المضلعات POLYGON MESH

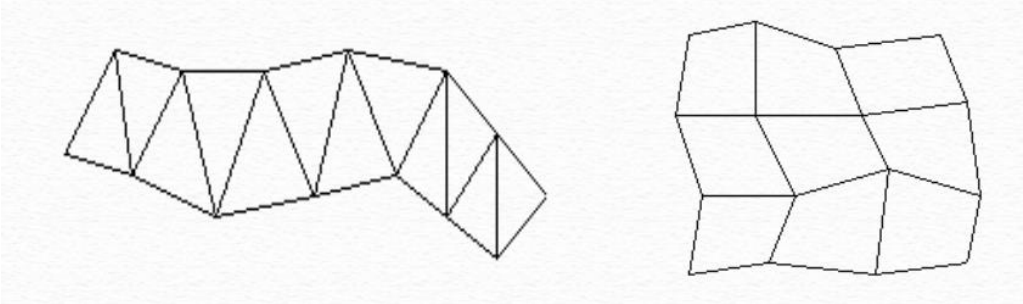
يتم توصيف شبكة المضلعات بالحوافأو الرؤوس أو الوجوة المتشكله (المضلعات)

الحافة: هي الي تربط بين رأسين من رؤوس المضلع.

الرأس: يكون مشترك على الأقل بين حافتين.

المضلع: هو تتالي مغلق من الحواف بحييث كل حافة تشترك على الأكثر بين مضلعين.

أنواع الشبكات الأكثر انتشاراً هي tringle strip و quadrilateral meshes



POLYGON DATA STRUCTURE

شبكة المضلعات تعرف بالرؤوس والحواف والسطوح وكل هذه العناصر يجب تخزينها للحصول على polygon meshes.

يوجد عدة طرق ولكننا نبحث عن الطريقة التي حجم تخزينها أقل وتمكننا من الحصول على حجم معلومات أكبر و العمليات الحسابية فيها أسرع.

POLYGON DATA STRUCTURE-1

نمثل شبكة المضلعات بالرؤوس حيث كل رأس له 3 إحداثيات

$$P=((x_1,y_1,z_1),(x_2,y_2,z_2),\dots\ ,(x_n,y_n,z_n))$$

تمثيلها في open gl

مكتبة سواء تحديد

```
struct Vertex {
        float coords[3];
}
struct Triangle {
        struct Vertex verts[3];
}
struct Triangle mesh[n];

glBegin(GL_TRIANGLES)
        for ( i = 0 ; i < n ; i++ )
        {
        glVertex3fv(mesh[i].verts[0]);
        glVertex3fv(mesh[i].verts[1]);
        glVertex3fv(mesh[i].verts[2]);
        }
glEnd();
```

منها open gl.

ملاحظة:

open gl تفرض علينا ترتيب الرؤوس مع عقارب الساعة أو عكسها لأهميتها في الوجه الأمامي والخلفي.

ميزاتها:

- سهلة في عملية القراءة والكتابة والنقل.
- الخرج مدعوم في أغلب برامج الرسم

عيوبها:

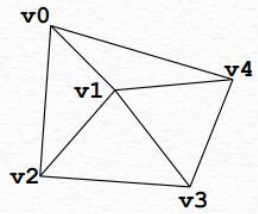
- لا تعطي معلومات عن الجوارات.
- لا تعطي معلومات عن الفتح والإغلاق.
- لا تعطي معلومات عن الرؤوس المشتركة والحواف المشتركة.
- تكرار الرؤوس المشتركة مما يسبب استهلاك اكبر للذاكرة.

POLYGON DATA STRUCTURE-2

هذه الطريقة توفر في استهلاك الذاكرة (مشكلة الطريقة السابقة)

بحيث نعرف الرؤوس التي تتألف منها الشبكة مرة واحدة.

ثم نعرف الوجوه ومضخة الرؤوس المشكلة لها.



vertices	v0	v1	v2	v3	v4

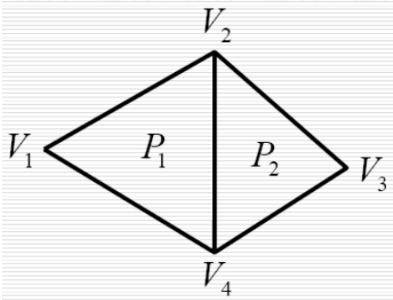
faces												
	0	2	1	0	1	4	1	2	3	1	3	4

مثال:

ميزاتها:

- معلومات الترابط سهل استنتاجها.
- تقلل من حجم التخزين في الذاكرة.
- نعرق الرأس مرة واحدة فقط.

$$V=((x_1,y_1,z_1),(x_2,y_2,z_2),\dots\ ,(x_n,y_n,z_n))$$



$$V=(V_1,V_2,V_3,V_4)$$
$$=((x_1,y_1,z_1),\dots,(x_4,y_4,z_4))$$
$$P_1=(1,2,4)$$
$$P_2=(4,2,3)$$

- عمليات الحساب والتلوين اسهل.

عيوبها:

- ما زالت المعلومات التي نحصل عليها قليلة وغير كافية.

POLYGON DATA STRUCTURE-3

وهذه هي الطريقة الأفضل.

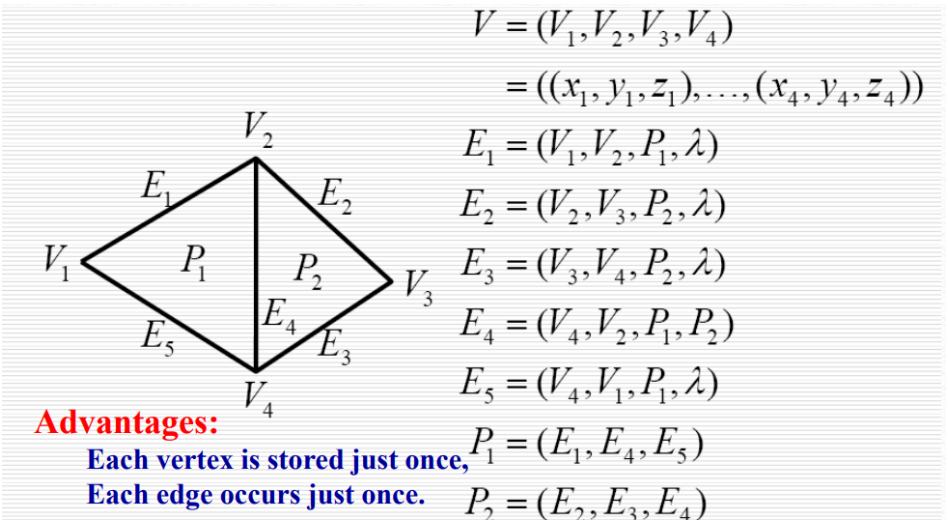
نعرف هنا ثلاثة قوائم قائمة رؤوس وقائمة حواف وقائمة أسطح أو مضلعات.

$$V=((x_1,y_1,z_1),(x_2,y_2,z_2),\dots\ ,(x_n,y_n,z_n))$$
$$E=((v_1,v_2,p_1,p_2),\dots,(v_{n-1},v_n,p_v,p_u))$$
$$P=(E_1,E_2\dots.E_n)$$

بحيث كل حافة لها رأسين والمضلعات المشتركة.

وقائمة المضلعات تعرف فقط بالحواف التي ضمنها.

مثال:



في هذه الطريقة أصبح لدينا معلومات أكثر ومع ذلك حافظنا على الهدر في الذاكرة.

ملاحظة: المعامل λ في قائمة الحواف للدلالة على أنه مشترك فقط في مضلع واحد.

PLAN EQUATION

لحساب معادلات المستوي لتحديد أي نقطة هل هي على المستوي أو أمامه أو خلفة

معادلة المستوي:

$$Ax + By + Cz + D = 0$$

حيث:

$$A = y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2)$$
$$B = z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2)$$
$$C = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)$$
$$D = -x_1(y_2z_3 - y_3z_2) - x_2(y_3z_1 - y_1z_3) - x_3(y_1z_2 - y_2z_1)$$

ملاحظة: المعادلات الرياضية غير مطلوب حفظها.

FRONT AND BACK POLYGON FACE

أي مستوي له وجهين وجه خلفي ووجه أمامي.

- Back face

هو الوجه الداخلي للمستوي.

- Front face

هو الوجه المرئي لنا أو للكاميرا.

لتحديد أي نقطة ما أنها أمام المستوي أو لا من خلال المعادلة

$$Ax + By + Cz + D \neq 0$$

if $Ax + By + Cz + D < 0$, the point (x, y, z) is behind the plane
if $Ax + By + Cz + D > 0$, the point (x, y, z) is in front of the plane

NORMAL VECTOR الناظم

يستخدم الناظم في عدة تطبيقات أهمها الإضاءة

وهو شعاع عامودي على السطح يتجه من الوجه الداخلي الى الوجه الأمامي

في open gl نحدد لكل مضلع ناظم أو لكل رأس.

في المنحنيات يكون لدينا ناظم مختلف لكل نقطة بينما في المستويات يكون الناظم هو نفسه لكل النقاط.

التعليمة المسؤولة عن تعريف الناظم في open gl هي `glNormal*()`

مثال:

Example 2-8 : Surface Normals at Vertices

```
glBegin (GL_POLYGON);
   glNormal3fv(n0);
   glVertex3fv(v0);
   glNormal3fv(n1);
   glVertex3fv(v1);
   glNormal3fv(n2);
   glVertex3fv(v2);
   glNormal3fv(n3);
   glVertex3fv(v3);
glEnd();
```

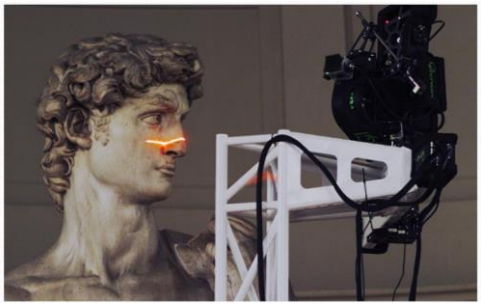
نلاحظ هنا أننا أخذنا لكل رأس ناظم.

حساب الناظم:

$$N = (V_2 - V_1) * (V_3 - V_1)$$

الجداء الخارجي لشعاعين يعطي الناظم عليهما لذلك ترتيب الرؤوس مهم.

MESHES FROM SCANNING



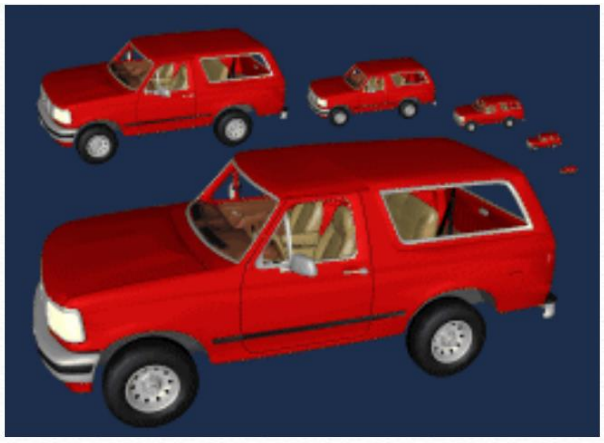
نستطيع توليد شبكة المضلعات بواسطة عملية scan من جهاز معين يعمل على

تقنيات الليزر .

LEVEL OF DETAIL

كيف نحصل على الدقة المناسبة وما هو حجم المضلع المناسب لها؟

يوجد توازن بين دقة التفاصيل و شروط الرؤية

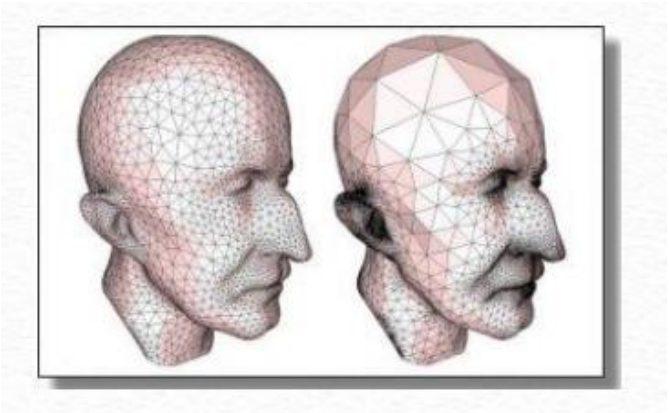


مثال:

كلما كانت السيارة أقرب أهتم بالتفاصيل أكثر .

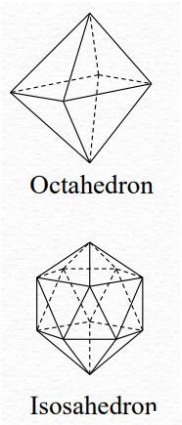
وكلما ابتعدت يقل الاهتمام بالتفاصيل.

ينقص التعقيد كلما ابتعدنا لانها كلما اقتربت تحتاج للعديد من العمليات الرياضية والتعقيد ووقت المعالجة.



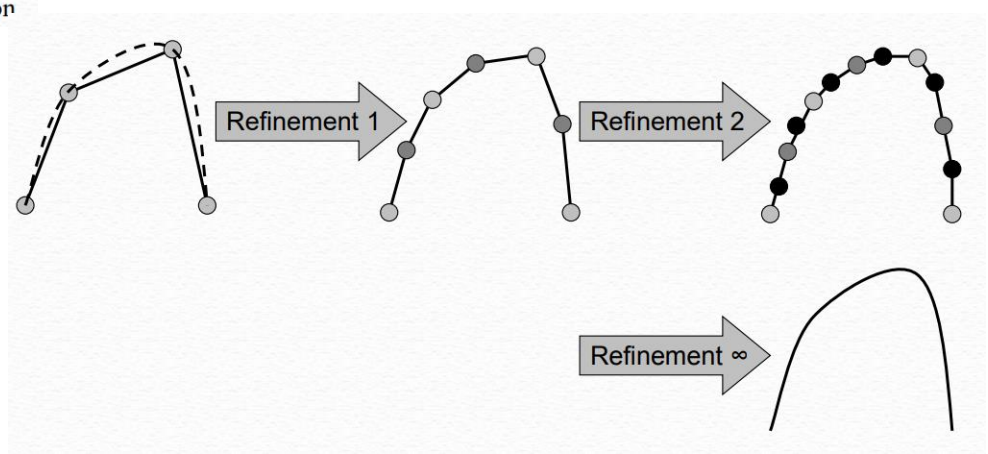
تحسين المودل IMPROVING THE MODEL

Subdivision method:

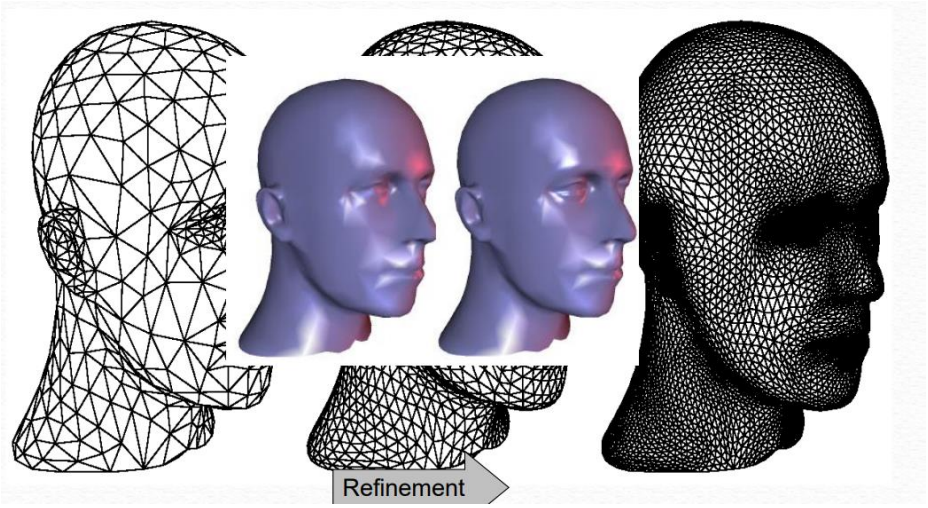


- ندخل رأس جديد في كل حافة.
- ندفع الرأس خارجاً لسطح الكرة التي تغلف السطوح.
- نقسم كل وجه مثلث الى 3 مثلثات باستخدام الرؤوس الجديدة.

مثال عن منحني لتقليل التكسرات:



مثال:



كلما زاد التعقيد توضحت التفاصيا أكثر ولكنها بالطبع تحتاج لمعالجة اكبر.

ملاحظة: كيفية زيادة الدقة رياضياً غير مطلوب, كما يوجد العديد من الأمثلة والصور عن هذه الفكرة في نهاية الملف.

نهاية المحاضرة العاشرة