



بيانيات الحاسوب
Computer Graphics

جامعة
المنارة

HAMARA UNIVERSITY

Dr.-Eng. Samer Sulaiman

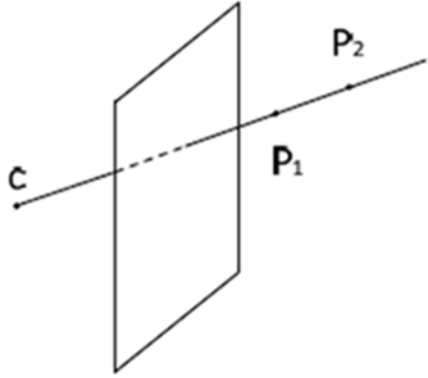
2022-2023

مفردات المنهاج



- أساسيات بيانيات الحاسوب
 - مقدمة: مفاهيم أساسية
 - التحويلات ثنائية البعد 2D
 - التحويلات ثلاثية البعد 3D
- خوارزميات بيانيات الحاسوب
 - الخوارزميات الهندسية
 - الخوارزميات النقطية الأساسية
 - **خوارزميات إزالة الأسطح المخفية**
- نماذج بيانيات الحاسوب
 - الألوان ونماذج الألوان
 - المنحنيات والأسطح: النمذجة الهندسية
 - نماذج الإضاءة والتظليل
 - عناصر التركيب والنمذجة الإجرائية

خوارزميات بيانيات الحاسوب



• إزالة الأسطح المخفية Hidden surface removal

- عرض مجموعة من المضلعات (الكائنات) ثلاثية الأبعاد

- بعضهم محجوب (على الأقل جزئياً) عن المراقب من قبل الآخرين

- عند تحديد مجال الإسقاط، يؤدي إلى الحصول على مستوى صورة الذي سيتم الإسقاط عليها ومجموعة من الخطوط التي يحدث الإسقاط على طولها

- يتم إسقاط جميع الخطوط ضمن مجال الإسقاط إلى نفس النقطة في مستوى الصورة

- بالنسبة للنقطتين P1 و P2 الواقعة على نفس الخط، سيتم اعتبار النقطة الأقرب لمستوى الصورة مرئية والنقطة الأخرى مخفية

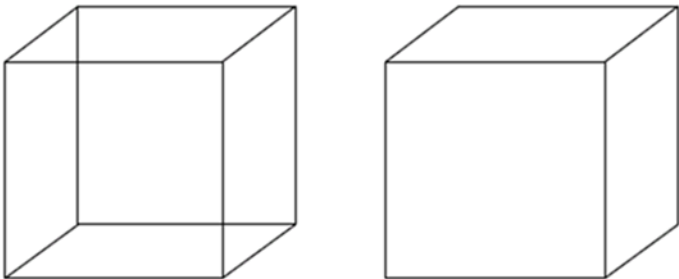
- مهمة إزالة السطح المخفي (HSR)، أو كما يطلق عليها أيضاً تحديد السطح المرئي) هي تحديد النقطة التي

- ستكون مرئية لكل بكسل من مستوى الصورة

- مثال: يتم عرض بعض حواف المضلعات فقط

- إزالة الحواف المخفية بواسطة المضلعات في حالة رسم المضلعات

- يؤدي إلى مهمة إزالة الحواف المخفية



خوارزميات بيانيات الحاسوب

• إزالة الأسطح المخفية Hidden surface removal

• في رسومات الحاسوب الحديثة:

- عادة ما تحتاج إلى عرض المضلعات والأشياء بأكملها
- تحديد الكائن الذي سيكون مرئيًا لكل بكسل
- مهمة سهلة لعدد قليل من الأشياء البسيطة
- تصبح معقدة للغاية ويستغرق وقتًا طويلاً:

- عن طريق زيادة تعقيد المشهد
- التعامل مع ملايين الأشياء

• بالنسبة لمثل هذه المشاهد ، تكون نسبة صغيرة فقط من جميع الكائنات مرئية حقًا

• يصبح فحص كل كائن (بالإضافة إلى فحص كل كائن مقابل كل كائن آخر) مكلفًا للغاية للتطبيقات في الزمن الحقيقي

• تعتمد جميع خوارزميات HSR على مبدأ الفرز

• تختلف حقًا في نوع الفرز المستخدم

• لا تهتم بالأشياء المخفية ولكن تريد تحديد الأشياء المرئية

• بالنسبة للخوارزميات الكلاسيكية ، يكون هذا الفرز واضحًا جدًا

• عادةً ما تكون الخوارزميات الحديثة مزيّجًا من عدة خوارزميات أبسط بما في ذلك التحسينات المختلفة والمؤشرات المكانية للتعامل مع المشاهد المعقدة في الزمن الحقيقي



خوارزميات بيانيات الحاسوب

• إزالة الأسطح المخفية Hidden surface removal

• التصنيف على أساس مكان التنفيذ:

• مجال الكائن Object space:

• عادة يتم مقارنة الكائنات مع بعضها البعض

• أكثر تعقيدًا $O(N^2)$ حيث N هو العدد الإجمالي للكائنات

• مجال الصورة Image space:

• مقارنة إسقاطات الكائنات لكل بكسل من مستوى الصورة

• تميل إلى أن تكون أبسط ولها تعقيد $O(NP)$ ، حيث P هو العدد الإجمالي للبكسل

• المفاهيم الأساسية:

• المضلعات الأمامية والخلفية:

• بالنسبة إلى متعدد السطوح المغلق:

• تحديد لكل وجه من الوجوه الناظم الذي يشير خارج متعدد السطوح، يسمى بالناظم الخارجي

• لا يحتوي متعدد السطوح المغلق على ثقب يمكن من خلالها رؤية داخله

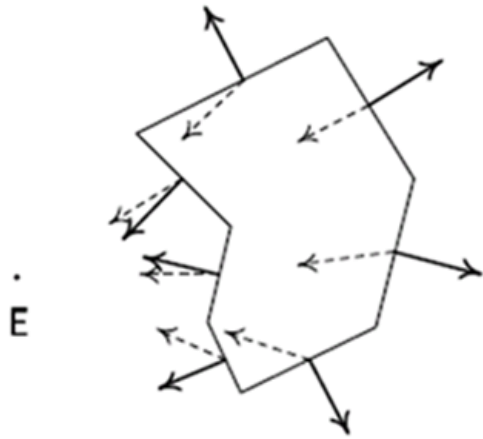
• الوجوه الوحيدة التي يمكن رؤيتها هي الوجوه التي تشير أشعة الناظم الخارجية في اتجاه المراقب

• تسمى هذه الوجوه بالوجوه الأمامية

• تسمى الوجوه التي تكون أشعة الناظم الخارجية بعكس اتجاه المراقب بالوجه الخلفي

• لا يمكن للمراقب رؤية أي من الوجوه الخلفية

• تصبح دائمًا مخفيًا عن طريق الوجوه الأمامية



•
E



خوارزميات بيانيات الحاسوب

• إزالة الأسطح المخفية Hidden surface removal

• المفاهيم الأساسية:

• المضلعات الأمامية والخلفية:

• بالنسبة إلى متعدد السطوح المغلق:

• لا يحتوي متعدد السطوح المغلق على ثقب يمكن من خلالها رؤية داخله

• بالنسبة إلى متعدد السطوح المحدب ، ستكون جميع أوجهه الأمامية مرئية تمامًا

• لن يخفي أي وجه أمامي وجهًا أماميًا آخر بسبب التحجب

• يتم حل مهمة الرؤية بسهولة

• بالنسبة لمتعدد السطوح غير المحدب أو لمجموعة من الأشكال المتعددة السطوح المحدبة

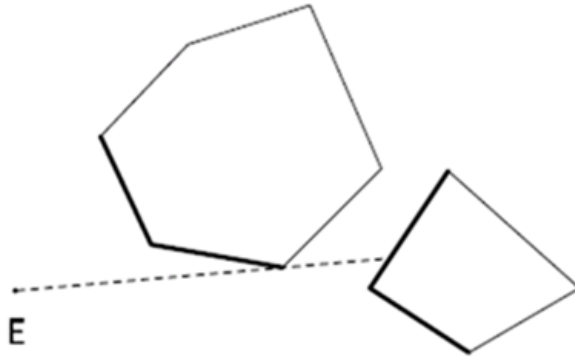
• يمكن لبعض المضلعات الأمامية إخفاء الأوجه الأمامية للمضلعات الأخرى عن المراقب

• تتبع شعاع من المراقب (تمامًا كما هو الحال بالنسبة للاختبار ثنائي الأبعاد للنقاط داخل المضلع)

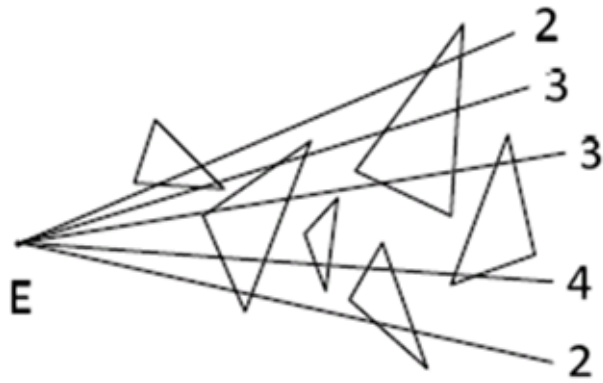
• عدد المرات التي يخترق فيها الشعاع المضلعات الأمامية يساوي عدد المرات التي يخترق فيها المضلعات الخلفية

• بشكل عام ، عدد المضلعات الخلفية هو حوالي نصف جميع المضلعات

• يؤدي رفضها (لأنها مخفية دائمًا عن المراقب) إلى تقليل عدد المضلعات المعالجة بمقدار ضعفين



خوارزميات بيانيات الحاسوب



• إزالة الأسطح المخفية Hidden surface removal

• المفاهيم الأساسية:

• تعقيد عمق:

• بالنسبة للخطوط التي تمر عبر بعض وحدات البكسل من مستوى الصورة:

• حساب عدد تقاطعاتها مع المضلعات الأمامية

• يسمى هذا الرقم تعقيد العمق للبكسل

• يتوافق تعقيد العمق بواحد أو صفر في الحالة البسيطة

• لا يوجد أكثر من كائن واحد يغطي هذا البكسل

• يصبح تحديد مدى الرؤية سهلاً للغاية

• عندما يزداد تعقيد العمق ، تصبح المشكلة أكثر صعوبة

• إذا كان عمق التعقيد يساوي 10

• وجود 10 كائنات تسقط على هذا البكسل

• واحد منهم فقط مرئي حقاً

• التسعة الأخرى هي مضيعة للموارد والمعالجة

• أخذ جميع وحدات البكسل في الصورة ليتم عرضها:

• تحديد متوسط تعقيد العمق للمشاهد

• اعتماداً على المشهد وأيضاً على موقع واتجاه المراقب

• التعامل مع المشاهد ذات التعقيد العالي للعمق :

• المهمة الأكثر أهمية ليست العثور على الأشياء المرئية ولكن رفض بأسرع ما يمكن معظم الأشياء غير المرئية

• يمكن معالجة الأشياء المتبقية بواسطة بعض الخوارزميات التقليدية

خوارزميات بيانيات الحاسوب

- إزالة الأسطح المخفية Hidden surface removal
- المفاهيم الأساسية:
 - التماسك Coherence:
 - يعد استخدام التماسك مفتاحًا لبناء خوارزميات HSR فعالة
 - أنواع التماسك:
 - تماسك الكائن Object coherence
 - كائنان قريبان يميلان إلى أن يكونا مرئيين (أو مخفيين) في وقت واحد
 - تماسك الوجه (السطح) Face coherence
 - إذا كان أحد وجوه الكائن مرئيًا (مخفيًا) ، فإن الوجوه المجاورة تكون مرئية (مخفية) أيضًا
 - تماسك الحواف Edge coherence
 - عادة ما تنتهي الحواف القريبة إلى نفس الوجه أو الوجوه القريبة من بعضها البعض وعادة ما يكون لها نفس الرؤية
 - تماسك المنطقة (بكسل) Area (pixel) coherence
 - إذا كان بعض البكسل يتوافق مع وجه / كائن معين ، فعادةً ما تتوافق وحدات البكسل القريبة أيضًا مع نفس الوجه / الكائن
 - تماسك الإطار (الزمن) Frame (time) coherence
 - عادةً ما تتغير مجموعة الوجوه / الكائنات المرئية قليلاً جدًا من إطار إلى إطار
 - كل هذه الأنواع من التماسك ليست قواعد صارمة
 - يجري الاستثناءات
 - تعكس اتجاهًا أو سلوكًا مشتركًا يمكن استخدامه للحصول على مكاسب في الأداء والتنفيذ

خوارزميات بيانيات الحاسوب

• إزالة الأسطح المخفية Hidden surface removal

• المفاهيم الأساسية:

• الحاجب Occluders:

- أي جسم يحجب بعض الأشياء الأخرى
- بفرض وجود كائن أو مضلع وموقع مراقب

- القدرة على بناء منطقة في مساحة ثلاثية الأبعاد يتم حجبا عن المراقب بواسطة هذا الكائن / المضلع
- يمكن التخلص من أي كائن موجود تمامًا داخل هذه المنطقة باعتباره مخفياً
- يمكن وجود عدة كائنات تقوم بمهمة الحجب:
- القدرة على الاستفادة من دمج الكائنات التي تقوم بالحجب

• يميل الاختبار باستخدام العديد من الكائنات الحاجبة مجتمعة إلى الحصول على نتائج أفضل من اختبار كل حاجب على حدى

• مثال:

• لا يمكن لأي من الحاجبين A و B إخفاء الكائن C.

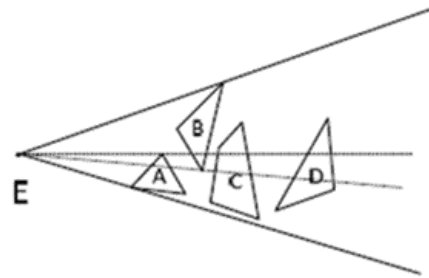
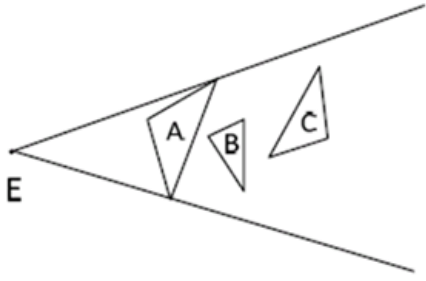
• معاً يمكن إخفاء الكائن C

• تقنية شائعة في الألعاب الحديثة:

• اختيار مجموعة من الكائنات الحاجبة (جعلها تبسيطاً للهندسة الحقيقية)

• استخدام هذه الكائنات لإخراج الهندسة المخفية

• إنشاء مثل هذه المضلعات التي تحتاج إلى استخدام مضلعات بسيطة (استخدامها بالاختبار يكون سهلاً)، والتي تعطي تقديراً مناسباً للرؤية



خوارزميات بيانيات الحاسوب

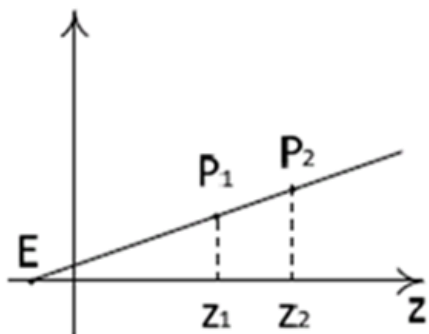


- إزالة الأسطح المخفية Hidden surface removal
- خوارزمية توجيه الشعاع Ray casting Algorithm:
 - تحديد ما هو مرئي بالضبط في كل بكسل من مستوى الصورة
 - مجرد بناء شعاع من موقع المراقب عبر البكسل
 - يتم فحص جميع الكائنات للتأكد من تقاطعها مع هذا الشعاع
 - يعطي أقرب تقاطع للمراقب نقطة مرئية للبكسل
 - فوائد الخوارزمية:
 - إنشاء صور واقعية:
 - تحديد الرؤية
 - حساب الظلال والانعكاسات والانكسارات والعديد من التأثيرات الأخرى
 - يمكن استخدام الكود التالي (بسيط)
 - تعقيد هذه الخوارزمية هو $O(NP)$ عندما تقوم بفحص كل كائن لكل شعاع
 - تعقيد الخوارزمية المحسنة (التي تعتمد عادةً على بنية الدليل المكاني المختلفة) هو $O(P \log N)$
 - القدرة على العمل مع البيانات غير المضلعة
 - يمكن معالجة أي كائن يمكن من أجله حساب تقاطع مع شعاع
- في مهام الزمن الحقيقي ، نادرًا ما يتم استخدام توجيه الأشعة لتحديد الرؤية ، ولكن غالبًا ما يتم استخدامه في مهام أخرى ، على سبيل المثال: عادة ما يتم تتبع رصاصة عبر المشهد للعثور على نقطة إصابة عن طريق خوارزمية توجيه الأشعة
- for all pixels :
 - for all objects :
 - compare distance

خوارزميات بيانيات الحاسوب

• إزالة الأسطح المخفية Hidden surface removal

• خوارزمية z-buffer:



- يتطلب مخزنًا مؤقتًا إضافيًا، حيث يتم الاحتفاظ بقيمة z للنقطة المسقطة لكل بكسل
- عادةً ما يتم الاحتفاظ بقيمة العمق كقيم بدون إشارة 16/24/32 بت unsigned integer
- يعتمد اختيار z على حقيقة أن قيمة z للنقطة بالنسبة لإسقاط المنظور ترتبط بالمسافة إلى المراقب (مركز الإسقاط)

- البحث عن نقطتين $P1$ و $P2$ على نفس خط الإسقاط أيهما أقرب إلى المراقب

- بمجرد مقارنة إحداثياتها z ، فإن النقطة ذات الإحداثي z الأصغر تكون أقرب إلى المراقب
- يمكن معاملة الإحداثي Z على أنه عمق النقطة

- الاحتفاظ بقيمة العمق لجميع وحدات البكسل في مخزن مؤقت خاص (مصنوفة ثنائية الأبعاد)، تسمى المخزن المؤقت z أو المخزن المؤقت للعمق

- في البداية، يتم تعيين عمق كل بكسل على القيمة القصوى التي يمكن تمثيلها بالبتات المتاحة

- على سبيل المثال: استخدام أعداد صحيحة بدون إشارة 32 بت للحفاظ على العمق، فإن القيمة القصوى ستكون $2^{32}-1$

- تدعم وحدات معالجة الرسومات الحديثة استخدام قيم النقطة العائمة للمخزن z -buffer

- لذلك يتم تحويل كل مضلع إلى صيغة نقطية ولكل بكسل ناتج يتم حساب إحداثيات z باستخدام الاستيفاء الثنائي الخطي bilinear interpolation

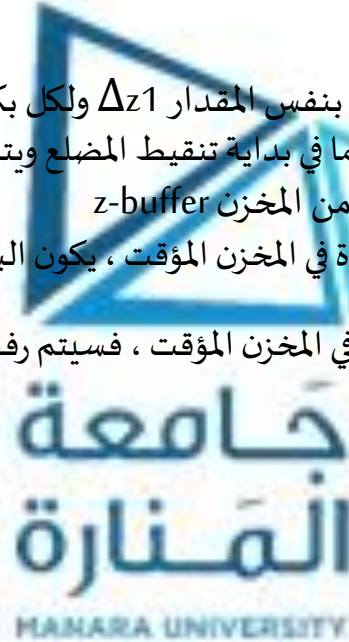
خوارزميات بيانيات الحاسوب

• إزالة الأسطح المخفية Hidden surface removal

• خوارزمية z-buffer:

- تنقيط مضلع لكل بكسلين (x, y) و $(x + 1, y)$ تختلف أعماقها بنفس المقدار Δz_1 ولكل بكسلين (x, y) و $(x, y + 1)$ تختلف أعماقها بالقيمة Δz_2
- تعتمد قيمة الدلتا السابقة على المضلع بحيث يتم حسابها في بداية تنقيط المضلع ويتطلب الحصول على عمق للبكسل التالي بإضافة واحدة فقط
- تتم مقارنة قيمة z المحسوبة للبكسل مع قيمة z لهذا البكسل من المخزن z-buffer
- إذا كانت قيمة z للبكسل الناتج أصغر من القيمة الموجودة في المخزن المؤقت ، يكون البكسل أقرب إلى المراقب ويتم كتابة قيمته z إلى z-buffer ويتم كتابة لونه في المخزن المؤقت للإطار
- إذا كانت قيمة z للبكسل أكبر أو تساوي القيمة الموجودة في المخزن المؤقت ، فسيتم رفض البكسل ببساطة
- باستخدام الكود التالي

- for all objects :
for all covered pixels :
compare z



- العمل في مساحة الصورة (مثل توجيه الشعاع)
- معالجة كل مضلع بدوره
- الاستفادة من التماسك باستخدام مخطط تدريجي لحساب قيم العمق
- الحصول على نتائج صحيحة بشكل مستقل عن ترتيب معالجة المضلعات
- إجراء فرز جذري حسب إحداثيات x و y ثم إجراء فرز z باستخدام مقارنة واحدة فقط لكل بكسل لكل مضلع

خوارزميات بيانيات الحاسوب

• إزالة الأسطح المخفية Hidden surface removal

• خوارزمية z-buffer:

• مقارنة بين خوارزمية depth buffer algorithm و ray casting algorithm:

• كلاهما يستخدم حلقة للمضلعات وحلقة لوحدات البكسل
• يتم تبديل هاتين الحلقتين

• الحلقة الخارجية لتوجيه الشعاع تعتمد على البكسل

• الحلقة الخارجية لمخزن العمق z-buffer تعتمد على المضلعات

• كلا الخوارزميات تدعم المعالجة التفرعية بسهولة

• بالنسبة لمخزن العمق ، تكون العملية التي يتم إجراؤها بالتوازي (مقارنة العمق) أبسط بكثير من عملية توجيه الأشعة (فحص جميع الكائنات بحثًا عن تقاطع مع شعاع)

• تعد خوارزمية المخزن المؤقت للعمق معيارًا واقعيًا تم استخدامه

• يجري تنفيذه في الأجهزة في جميع وحدات معالجة الرسومات لسنوات عديدة

• هناك تطبيقات للأجهزة التي تستخدم توجيه الأشعة

• ليست منتشرة ورخيصة

• يستخدم عادة للمهام المتخصصة مثل عرض الحجم

• المخزن المؤقت للعمق الجهاز سريع وفعال للغاية

• لا تزال تتطلب معالجة جميع المضلعات

• بالنسبة إلى المشاهد الكبيرة جدًا ، يمكن أن تصبح بطيئة ما لم يتم إجراء بعض التحسينات الأخرى



خوارزميات بيانيات الحاسوب

• إزالة الأسطح المخفية Hidden surface removal

• خوارزميات الأولوية Priority algorithms:

• تكمن عملية الفرز في مبدأ عمل جميع خوارزميات HSR

• تعتمد مجموعة الخوارزميات على الفرز الصريح للمضلعات

• تقوم هذه الطرق بفرز الكائنات (عادة المضلعات)

• ما يهم سوى تنقيطها ورسمها بالترتيب الفرز ، حيث سيتم الحصول على الصورة الصحيحة

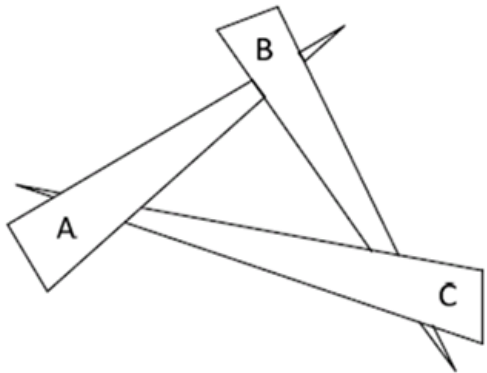
• عرض المضلعات البعيدة أولاً والمضلعات القريبة من المراقب الأخيرة (ترتيب من الخلف إلى

الأمام)

• بالنسبة لبعض مجموعات البيانات ، يعمل هذا النهج بشكل جيد

• مثال: رسم سطح $z = f(x, y)$

• تتطلب الحالات الأخرى تقسيم المضلعات ، وإلا فإن هذا الترتيب مستحيل



خوارزميات بيانيات الحاسوب



• إزالة الأسطح المخفية Hidden surface removal

• خوارزميات الأولوية Priority algorithms:

• خوارزمية الرسام Painter's algorithm:

• أبسط أنواع خوارزميات الأولوية:

• تعتمد على فرز جميع المضلعات وفقًا للتالي:

• الحد الأدنى من إحداثي z لرؤوسه

• إحداثي z لمتوسط رؤوسه

• بدء الرسم بأبعد الكائنات ثم ارسم أقرب الكائنات فوق الكائنات الأبعد

• تعمل بشكل جيد مع المضلع التقريبية للحلقة والأسطح الرياضية الأخرى

• ينتج عن بعض الحالات فشل في إنتاج صورة غير صحيحة

• استخدام الفرز الصريح في فضاء الكائنات

• إجراء المقارنة في فضاء الصور

• يتطلب كل إطار للرسوم المتحركة فرزًا واضحًا

خوارزميات بيانيات الحاسوب

• إزالة الأسطح المخفية Hidden surface removal

• خوارزميات الأولوية Priority algorithms:

• خوارزمية الرسام Painter's algorithm:

• أبسط أنواع خوارزميات الأولوية:

• امتلاك إمكانية الاستفادة من تماسك الإطار

• عادةً ما يتغير ترتيب الكائنات المصنفة قليلاً جداً من إطار إلى إطار ، لذلك من الممكن استخدام طرق الفرز التي يمكن أن تستفيد من هذا

• لا يتم إجراء تقسيم المضلع ، لذلك لا يمكن التعامل مع جميع الحالات بشكل صحيح
• استخدام أشجار تقسيم الفضاء الثنائية Binary Space Partitioning trees (BSP-trees)

• توفير الترتيب الصحيح لأي مجموعة من المضلعات وأي موقع مراقب

• بدلاً من فرز المضلعات ، يتم تنظيمها في شجرة ثنائية باستخدام الخاصية البسيطة

• إذا كان المضلعان A و B موجودان في نصفين مختلفين بالنسبة إلى مستوى معين π

• يكون المضلع الذي يقع في نفس نصف مساحة المراقب لا يمكن إخفاؤه بأي شكل من الأشكال بواسطة المضلع الموجود في نصف المساحة أخرى



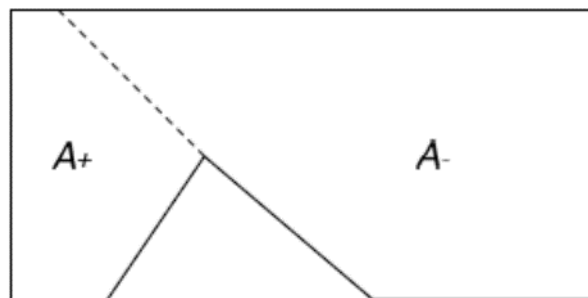
خوارزميات بيانيات الحاسوب

• إزالة الأسطح المخفية Hidden surface removal

• خوارزميات الأولوية Priority algorithms:

• استخدام أشجار تقسيم الفضاء الثنائية Binary Space Partitioning trees (BSP-trees)

- أحد أكثر التطبيقات المعروفة لأشجار BSP لإزالة الأسطح المخفية هو استخدامها في لعبة Doom الأسطورية First Person Shooter (FPS) بواسطة idSoftware. اللعبة ليست ثلاثية الأبعاد نقية ، بل ما يسمى D - 2.5 خريطة مسطحة ثنائية الأبعاد مقذوفة في البعد الثالث
- تم رسم جميع مستويات اللعبة كخرائط ثنائية الأبعاد وتم تخصيص ارتفاعاتها وموادها للحواف (التي أصبحت جدرانًا) والمضلعات (التي أصبحت أرضيات وسقوف)
- تم تجميع جميع الحواف المكونة للمستوى في مرحلة المعالجة المسبقة في شجرة ثنائية الأبعاد BSP
- إجراء بناء شجرة BSP هو مجرد معالجة متكررة لمجموعة الحواف التي تتوقف بمجرد أن تشكل الحواف جزءًا من حدود المضلع المحدب
- يتم ذلك لأنه إذا كانت الحواف جزءًا من حدود المضلع المحدب ، فلا داعي لمزيد من الطلبات ؛ علينا فقط إزالة الواجهات الخلفية
- تحتاج فقط إلى مستوى تقسيم واحد (جذر الشجرة) يشكل عقدتين ورقيتين $A+$ و $A-$



خوارزميات بيانيات الحاسوب

• إزالة الأسطح المخفية Hidden surface removal

• خوارزميات الأولوية Priority algorithms:

• استخدام أشجار تقسيم الفضاء الثنائية Binary Space Partitioning trees (BSP-trees)

• بناء شجرة BSP:

- يتم إجراء جميع التقسيمات المضلعة اللازمة لذلك لكل عقدة شجرية بها شجرتان فرعيتان تقعان في نصفي مستوي مختلفة
- الحصول على ترتيب صحيح من الخلف إلى الأمام لموقع مراقب معين
- نحتاج فقط إلى اجتياز الشجرة بحيث نختار في كل عقدة ترتيب اجتياز كلا الفرعين حتى نتمكن أولاً من اجتياز العقدة الفرعية المقابلة لنصف مستوي لا تحتوي على المراقب.
- تقوم أشجار BSP بتقسيم الشكل إلى أجزاء إذا تم تجاوز الشكل بواسطة مستوى التقسيم.
- بسبب ذلك ، يمكن أن يكون كل شكل في عقدة فرعية واحدة فقط.
- عادة ما يتم بناء الشجرة للمضلعات ، لذلك لكل مضلع لدينا مستوى يمكن استخدامه لتقسيم المضلعات الأخرى

خوارزميات بيانيات الحاسوب

• إزالة الأسطح المخفية Hidden surface removal

• خوارزميات الأولوية Priority algorithms:

• استخدام أشجار تقسيم الفضاء الثنائية Binary Space Partitioning trees (BSP-trees)

• بناء شجرة BSP:

• مبدأ العمل:

• وجود مجموعة من المضلعات A و B و C و D

• اختيار واحد منهم (مثل المضلع A)

• استخدام مستوى يحتوي على هذا المضلع لتصنيف جميع المضلعات الأخرى.

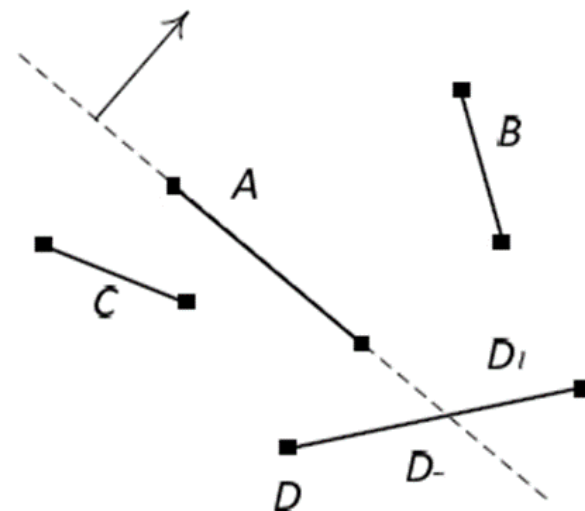
• لأي مضلع آخر هناك أربعة احتمالات فقط

• يقع المضلع بالكامل في نصف المنطقة الموجبة بالنسبة للمستوى (مثل المضلع B)

• يقع المضلع بالكامل في نصف المنطقة السالبة بالنسبة للمستوى (مثل المضلع C)

• يقع المضلع في المستوى نفسه

• يتم عبور المضلع بواسطة المستوى (مثل المضلع D).



خوارزميات بيانيات الحاسوب

• إزالة الأسطح المخفية Hidden surface removal

• خوارزميات الأولوية Priority algorithms:

• استخدام أشجار تقسيم الفضاء الثنائية (BSP-trees) Binary Space Partitioning trees

• بناء شجرة BSP:

• مبدأ العمل:

• وجود مجموعة من المضلعات A و B و C و D

• تعطينا الحالتان الأوليتان مجموعتين (قوائم) من المضلعات

• تحتوي القائمة الأولى على جميع المضلعات من نصف المنطقة الموجبة

• تحتوي القائمة الثانية على جميع المضلعات من نصف المنطقة السالبة.

• الحالة الأخيرة (يتم تجاوز المضلع بواسطة المستوى) ، تتطلب تقسيم المضلع بواسطة المستوى

• بعد تقسيم جميع أجزائه ، يمكن تصنيفها على أنها تقع في نصف المنطقة الموجبة أو السالبة

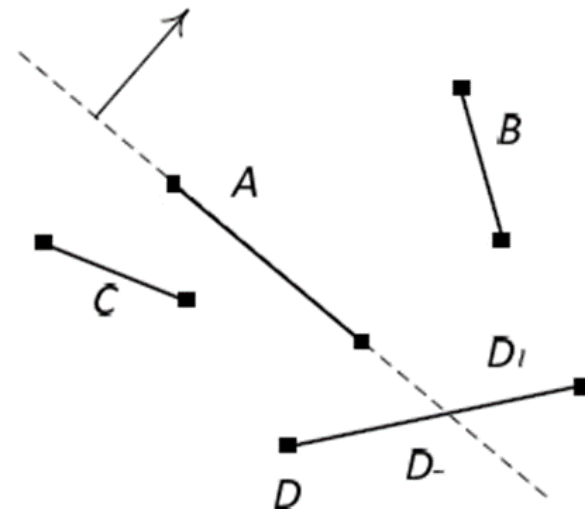
• وضع الأجزاء في القوائم المقابلة

• يمكن إضافة المضلعات الموجودة في المستوى نفسه إلى أي من هاتين القائمتين ، ولا يهم أي منهما.

• النتيجة: الحصول على قائمتين من المضلعات (D + ، B) و (D - ، C) ومستوى تقسيم (أو المضلع A الذي تم استخدامه للحصول على المستوى) بدلاً من القائمة الأولية (D ، C ، B ، A).

• أخذ المضلع A (مستواه) كجذر للشجرة الثنائية

• ستتوافق قوائم المضلعات التي تم الحصول عليها مع توابعها أو فروعها



خوارزميات بيانيات الحاسوب

• إزالة الأسطح المخفية Hidden surface removal

• خوارزميات الأولوية Priority algorithms:

• استخدام أشجار تقسيم الفضاء الثنائية Binary Space Partitioning trees (BSP-trees)

• بناء شجرة BSP:

• مبدأ العمل:

• وجود مجموعة من المضلعات A و B و C و D

• تطبيق نفس الإجراء على كلتا القائمتين (تتم معالجة كل قائمة بشكل مستقل عن الأخرى)

• الحصول على عقدتين إضافيتين وأربع قوائم مضلعات

• إن تكرار هذا الإجراء حتى لا تحتوي القوائم الناتجة على أكثر من مضلع واحد سينتهي في شجرة ثنائية.

• بالنسبة لمضلعات A و B و C و D، يمكن أن ينتهي بنا الأمر بشجرة

• (لاحظ أنه يمكن أن تكون هناك أشجار BSP أخرى لهذه المجموعة من المضلعات).

