مخططات الحاسبة

كلية التربية للعلوم الصرفة

قسم علوم الحاسوب

المرحلة الثالثة

اعداد

أ.د. اسراء محمد خضر

م.م. عمار عادل احمد & م. كنار محمد سامي

2025-2024

محاضرة 1: مقدمة

Computer Graphics: علم يتعلق بتكوين الصـــور والأفلام المتحركة باســـتخدام الحاســـوب. وهذا يتضـمن استخدام الجزء المادي والبرمجيات لتكوين هذه الصـور ويسـتخدم في تطبيقات متعددة. تطور هذا العلم في السـنوات 10–20 الماضـية. كما يمكن تعريفه بأنه نوع من علوم الحاسـوب الذي يعالج النظرية والتكنولوجيا لتأليف صـورة على الحاسـوب بحيث يمكن أن تصـف صـورة بسـيطة لتخطيطات معينة على الحاســـوب آو تصـــف عدة أشـــكال متداخلة ومعقدة. وكلا التعريفين يدلان بان البرمجيات التي تطورت لتكوين هذه الرسوميات هي الأنظمة الثنائية والثلاثية الأبعاد. تطور هذا العلم بواسطة العالم ايفان سوذرلاند سنة 1960 وهو احد المؤسسين الذي استخدم التحويلات الثنائية والثلاثية الأبعاد وكذلك معادلات الخط المستقيم .

مجالات استخدام مخططات الحاسوب:

رسومات الحاسوب لها تطبيقات مختلفة ، بعضها المدرجة أدناه

1. **واجهات مستخدم رسومات الكمبيوتر واجهات المستخدم الرسومية – نموذج رسومي موجه بالماوس يسمح للمستخدم بالتفاعل مع الحاسوب.**
2. **رسومات عرض الأعمال – "الصورة تستحق ألف كلمة".**
3. **رسم الخرائط – رسم الخرائط.**
4. **خرائط الطقس – خرائط في الوقت الحقيقي ، تمثيلات رمزية.**
5. **التصوير الساتلي – الصور الجيوديسية.**
6. **تحسين الصورة – زيادة حدة الصور غير الواضحة.**
7. **التصوير الطبي – التصوير بالرنين المغناطيسي ، التصوير المقطعي المحوسب ، إلخ – غير جراحي الفحص الداخلي.**
8. **الرسومات الهندسية – الميكانيكية والكهربائية والمدنية وغيرها استبدال مخططات الماضي.**
9. **أسلوب الطباعة – استخدام صور الشخصيات في النشر استبدال النوع الصعب من الماضي.**

10. الهندسة المعمارية – خطط البناء ، الرسومات الخارجية .

11. الفن – توفر أجهزة الكمبيوتر وسيلة جديدة للفنانين.

12– التدريب – محاكيات الطيران ، والتعليمات بمساعدة الكمبيوتر ، إلخ.

Photo: هي أي صورة ملتقطة بكاميرا رقمية او عادية او اجهزة الموبايل.

Picture: وهي التسمية القديمة للـ Image.

Image: وهو المصطلح الذي يندرج تحته كل الصور التي تصدر من التلفاز والقادمة من الاقمار الصناعية والصور الثابتة والصور المتحركة في الافلام.

Pixel: هو اصغر جزء رسومي يظهر على الشاشة وبشكل نقطة وله احداثيين (x,y)، وأن عدد نقاط الصورة تخزن في ذاكرة الحاسوب بشكل مصفوفة وأن عدد النقاط الكلية في الشاشة هي التي تحدد دقة الشاشة (Resolution) مثل 256×256 ، 600×800.

مكونات الصورة Geometric Primitive:

هي عناصر أولية هندسية تكون اجزاء من الرسوميات مثل:

PolyLine: هو مجموعة متسلسلة من الخطوط المستقيمة المتصلة نهاية بنهاية يدعى الخط المستقيم حافة، ونهايات الخطوط بالروؤس، وأبسط مثال لذلك الخط المستقيم وهو عبارة عن مجموعة من النقاط المضيئة المرتبطة بنهايتين (x_1,y_1) و (x_2,y_2). يدعى الخط المستقيم بالمغلق اذا انتهى من نقطة البداية ويدعى البسيط اذا لم يتقاطع.

Text: يوجد نوعين من اجهزة الرسوميات:

Text mode: يستخدم دوال ثابتة لكتابة الحروف والارقام والرموز على الشاشة.

Graphic mode: يرسم الحروف باسلوب أوفر واجمل في العرض على الشاشة.

Filled region: هو اسلوب مليء شكل هندسي باللون او أي نموذج متكرر أو نموذج بلون معين مثل رسم اشكال متداخلة مع بعضها.

–تنسيق الاحداثيات

1. تم استخدام مواصفات الإحداثيات الديكارتية في الكمبيوتر الرسومات.

2. أي نوع آخر من القيم الإحداثي للصورة يجب أن يكون تحويلها إلى تنسيق ديكارتي قبل إعطاء مدخلات للرسومات صفقة.

3. عملية التحويل من وصف إحداثيات العالم يشار إلى المشهد على شاشة جهاز إخراج واحد أو أكثر باسم "إحداثيات الجهاز" أو "إحداثيات الشاشة".

4. بشكل عام ، يقوم النظام الرسومي أولاً بتحويل إحداثيات العالم الموقف لتطبيع إحداثيات الجهاز. في النطاق من 0 إلى 1 قبل التحويل النهائي إلى إحداثيات جهاز معين.

- أجهزة العرض

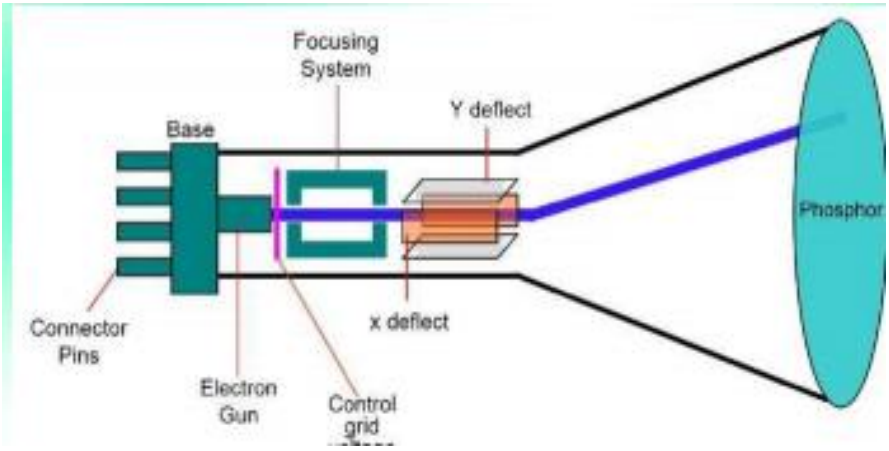
تُعرف أجهزة العرض أيضًا باسم أجهزة الإخراج. الأكثر استخداما جهاز الإخراج في نظام الرسومات هو شاشة فيديو.

شاشة CRT

1-أنبوب أشعة الكاثود

تحتوي على أنبوب زجاجي ، ذو طرف كبير يمثل الشاشة مطلية من الداخل بطبقة الفوسفور ، بينما يحتوي الطرف الآخر بشكل أساسي على البندقية الإلكترونية وملفات الانحراف. تحتوي البندقية الإلكترونية على أجزاء كما يلي:

1. عنصر التسخين 2. كاثود 3.شبكة التحكم 4. أنود التسارع 5.التركيز على الشبكة.



يعتمد CRT على المفهوم الفيزيائي بأن الفوسفور ينتج ضوءًا إذا كانت إضرابها بالإلكترونات لها سرعة ولحظة تؤثر على لإلكترون الفوسفور السرعة وحرة لإعطاء الضوء. CRT هو ملف أنبوب زجاجي مفرغ مع شاشة داخلية مطلية بالفوسفور.

1. يطلق مسدس الإلكترون حزمة من الإلكترونات (أشعة الكاثود).

2. يمر شعاع الإلكترون من خلال أنظمة التركيز والانحراف التي توجهها نحو مواضع محددة على المغلفة بالفوسفور شاشة.
3. عندما يضرب الشعاع الشاشة، ينبعث الفوسفور نقطة صغيرة من ضوء في كل موضع تلامسه شعاع الإلكترون.
4. يعيد رسم الصورة عن طريق توجيه شعاع الإلكترون مرة أخرى فوق نقاط الشاشة نفسها بسرعة. هناك نوعان من التقنيات المستخدمة لإنتاج الصور على شاشة CRT

1- **Vector scan/Random Scan Display**

شاشة المسح المتجه (Vector Scan Display) هي نوع من تقنيات العرض حيث تقوم برسم الصور مباشرةً كخطوط أو متجهات بدلاً من شبكة بكسلات. بدلاً من مسح الشاشة نمطياً صفًا بعد صف، تتحرك شاشة المتجه بالشعاع الإلكتروني لرسم الأشكال مثل الخطوط، المنحنيات، والنقاط بناءً على وصفات رياضية.

الخصائص الرئيسية لشاشات المسح المتجه:

- رسم الخطوط: تقوم الشاشة بإنشاء الصور عن طريق رسم خطوط بين الإحداثيات المحددة، وهو مثالي لعرض النماذج الهيكلية (wireframes)، والنصوص، والرسومات القائمة على الخطوط.

- لا توجد شبكة بكسلات: على عكس شاشات الرستر التي تعتمد على شبكة بكسلات، ترسم شاشات المتجه الأشكال بدقة على مسارات سلسة، مما ينتج غالباً صورًا أكثر وضوحًا للرسومات الخطية.

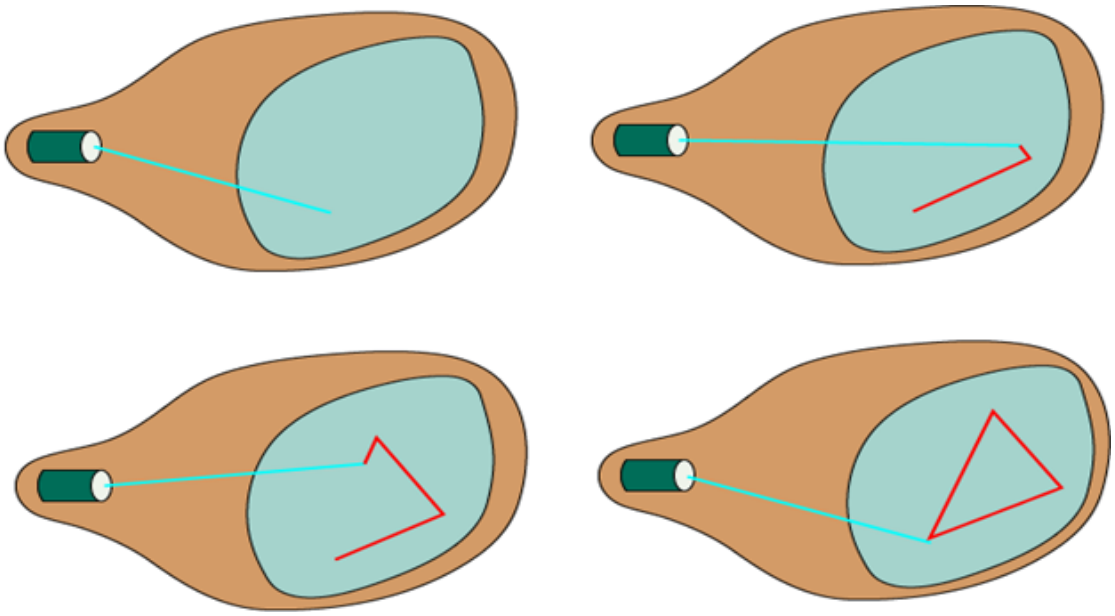
- كفاءة لبعض أنواع الرسومات: نظرًا لأن النظام يرسم الخطوط الضرورية فقط، فإن شاشات المتجه فعالة للغاية في التطبيقات التي تتكون فيها الصور بشكل رئيسي من الرسومات الخطية أو الأشكال البسيطة.

تطبيقات شاشات المسح المتجه:
1. أجهزة الأوسيلوسكوب: كانت الاستخدامات المبكرة لشاشات المتجه في أجهزة الأوسيلوسكوب، حيث كانت الموجات الممثلة على الشاشة تُعرض كخطوط.

2. ألعاب الفيديو المبكرة: تم استخدام شاشات المتجه في الألعاب مثل *Asteroids* وBattlezone، حيث كانت الرسومات الهيكلية مناسبة جدًا لهذه التكنولوجيا.

3. التصميم بمساعدة الحاسوب (CAD): استخدمت الأنظمة المبكرة للتصميم بمساعدة الحاسوب شاشات المتجه لعرض الرسومات الهندسية الدقيقة والنماذج الهيكلية.

4. محاكيات الطيران: تم استخدام الرسومات المتجهية أيضًا في محاكيات الطيران، حيث كانت دقة رسم الخطوط أمرًا بالغ الأهمية.



2-**Raster Display**

شاشة الرستر (Raster Display) هي نوع من تقنيات العرض المرئي التي تمثل الصور على شكل شبكة أو مصفوفة من البكسلات. تعمل عن طريق مسح شعاع إلكتروني (في شاشات CRT) أو التحكم في البكسلات الفردية (في الشاشات المسطحة الحديثة) صفًا بعد صف، بدءًا من أعلى الشاشة والانتقال أفقيًا. بمجرد الانتهاء من صف واحد، ينتقل النظام إلى الصف التالي حتى يتم عرض الصورة بالكامل.

الجوانب الرئيسية لشاشات الرستر تشمل:

– المسح الرستري: عملية مسح شعاع إلكتروني أو تفعيل صفوف البكسلات عبر الشاشة من اليسار إلى اليمين، ومن أعلى إلى أسفل.

– تمثيل البكسل: يتم تقسيم الشاشة إلى شبكة من البكسلات، بحيث يمثل كل بكسل نقطة ضوء يمكن أن يكون لها لون وسطوع معين.

– ذاكرة الإطار (Frame Buffer): يتم تخزين الصورة التي سيتم عرضها في ذاكرة مؤقتة، حيث يتوافق كل موقع في الذاكرة مع بكسل على الشاشة.

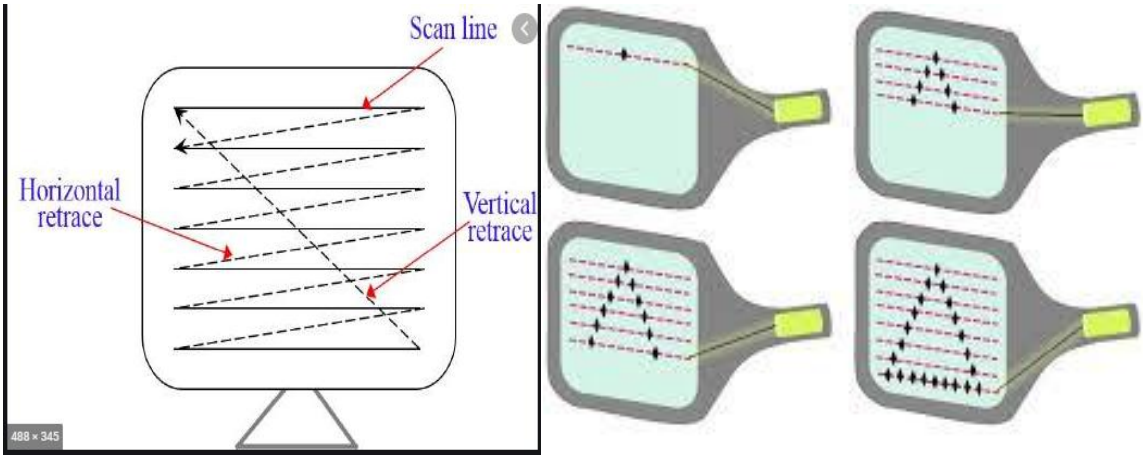
– معدل التحديث: يتم إعادة رسم الصورة على الشاشة عدة مرات في الثانية (عادةً 60 هرتز أو أكثر) لمنع الوميض وتوفير حركة سلسة.

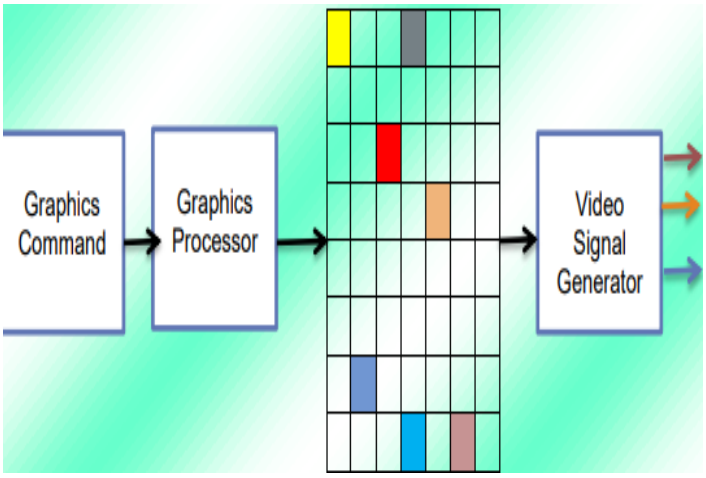
أنواع شاشات الرستر:

1. أنبوب الأشعة المهبطية (CRT): يستخدم أشعة إلكترونية ومواد فسفورية لإضاءة البكسلات على الشاشة.
2. شاشة الكريستال السائل (LCD): تستخدم إضاءة خلفية وكريستالات سائلة لتعديل الضوء وإنشاء الصورة.
3. شاشة LED: مشابهة لـ LCD لكنها تستخدم إضاءة خلفية LED، أو أحيانًا تستخدم LEDs كعناصر البكسل الفعلية.

شاشات الرستر هي أكثر أنواع الشاشات استخدامًا اليوم في أجهزة التلفاز، شاشات الكمبيوتر، والأجهزة المحمولة.يتم تمرير شعاع الإلكترون عبر الشاشة صفًا واحدًا في كل مرة من أعلى إلى أسفل. كما هي يتم تخزين تعريف الصورة في منطقة ذاكرة تسمى المخزن المؤقت للإطار Frame Buffer . هذا الإطار العازلة يخزن قيم الشدة لجميع نقاط الشاشة. كل نقطة شاشة تسمى بكسل (صورة عنصر).
في أنظمة الأسود والأبيض ، يُطلق على المخزن المؤقت للإطار الذي يخزن قيم البكسل أ نقطية Bitmap. كل إدخال في الصورة النقطية عبارة عن بيانات 1 بت التي تحدد تشغيل (1) وإيقاف تشغيل (0) شدة البكسل.
في أنظمة الألوان ، يُطلق على المخزن المؤقت للإطار الذي يخزن قيم وحدات البكسل اسم خريطة بكسل (على الرغم من أن العديد من مكتبات الرسومات في الوقت الحاضر تسميها على أنها صورة نقطية أيضًا). كل إدخال في الخريطة البيكسيلية تحتل عددًا من البتات لتمثيل لون البكسل. للحصول على شاشة ملونة

حقيقية ، فإن ملف عدد البتات لكل إدخال هو 24 (8 بت لكل قناة حمراء / خضراء / زرقاء ، كل قناة 28 = 256 مستويات قيمة الشدة ، أي. 256 إعدادًا للجهد لكل من مسدسات الإلكترون الأحمر / الأخضر / الأزرق).





<u>-مزايا عرض CRT</u>

1. منخفضة التكلفة.
2. انخفاض الوزن.
3. استجابة ودقة عالية.
4. زاوية رؤية واسعة.
5. شاشة مسطحة مع صورة واضحة وألوان واضحة ودقة عالية.

6- تقنية مصدق عليها وإصلاحات سهلة ومكلفة.

- عيوب شاشة CRT

1. حجم كبير ، وزن ضخم مع حجم الشاشة.
2. الطاقة المستهلكة.
3. أشعة كهرومغناطيسية تحيط بالشاشة.
4. معظم شاشة CRT تعمل مع الإشارة التناظرية ونادرًا ما تعمل مع الإشارة الرقمية.

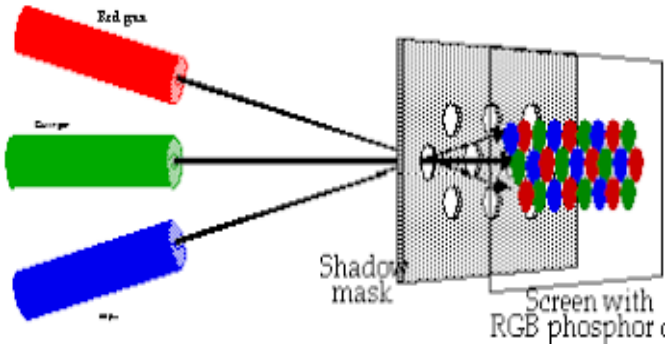
المحاضرة 2

الشاشات الملونة والشاشات المسطحة

1-شاشات CRT الملونة

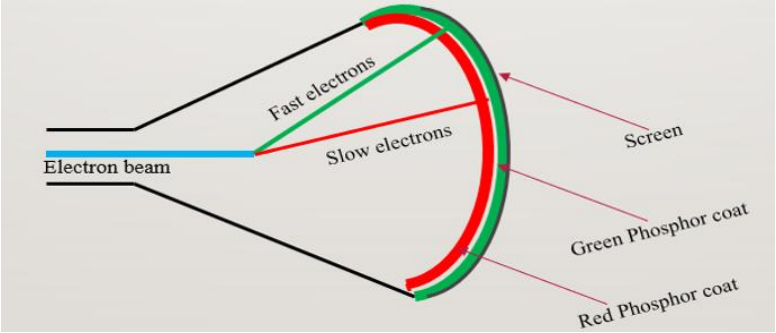
تعرض شاشات CRT صورًا ملونة باستخدام مجموعة من الفوسفورات التي تنبعث منها إضاءة ملونة مختلفة. الأحمر والأخضر والأزرق. تحتوي على 3 مسدس الكتروني يتوافق مع كل لون. ينتج نطاقًا من الألوان عن طريق الجمع بين الضوء المنبعث من الفوسفورات المختلفة بسرعات مختلفة. هناك طريقتان أساسيتان لعرض الألوان:

1- تقنية اختراق الشعاع / نمط الشريط. 2- تقنية قناع الظل / قناع دلتا.



1-تقنية اختراق الشعاع Beam Penetration

1. **تستخدم هذه التقنية مع شاشات المسح العشوائية.**
2. **إن CRT الملون مطلي بطبقتين من الفوسفور عادة باللونين الأحمر والأخضر. الطبقة الخارجية من الأحمر والطبقة الداخلية من الفوسفور الأخضر.**
3. **إنها تقنية منخفضة التكلفة لإنتاج الألوان في شاشات المسح العشوائية.**
4. **يمكنه عرض الألوان الأحمر، الأخضر، البرتقالي والأصفر فقط.**



2- تقنية قناع الظل Shadow Mask

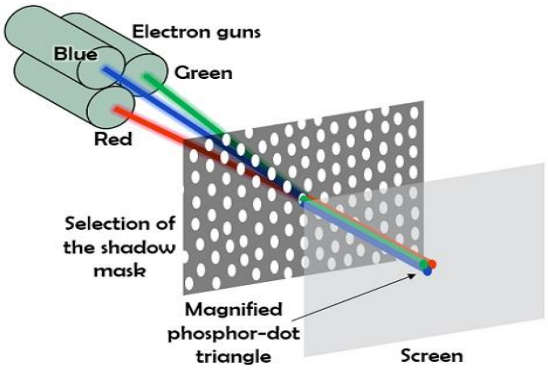
1. تنتج مجموعة واسعة من الألوان مقارنة بتقنية اختراق الشعاع.

2. تستخدم هذه التقنية بشكل عام في شاشات المسح النقطي. بما في ذلك التلفزيون الملون.

3. في هذه التقنية ، يحتوي CRT على ثلاث نقاط لونية فسفورية في كل موضع بكسل. نقطة للأحمر وأخرى للأخضر وأخرى للضوء الأزرق. يُعرف هذا باسم Dot Triangle.

4. شبكة قناع الظل تقع خلف الشاشة المطلية بالفوسفور. تتكون شبكة قناع الظل من سلسلة من الثقوب المحاذاة مع نمط نقطة الفوسفور.

5. عن طريق تغيير شدة الحزم الإلكترونية الثلاثة يمكننا الحصول على ألوان مختلفة في قناع الظل CRT.



Shadow Masking in Colour CRT

-مزايا شاشات قناع الظل

1. إنتاج صور دقيقة 2. إنتاج مشاهد الظلال والألوان المختلفة.

2-سلبيات شاشات قناع الظل

1. دقة منخفضة 2. غالية الثمن 3. شعاع إلكترون موجه إلى الشاشة بأكملها

• <u>الفرق بين طريقة اختراق الشعاع وطريقة قناع الظل.</u>

الميزة	طريقة اختراق الشعاع	طريقة قناع الظل

الاستخدام	يستخدم مع Random Scan System لعرض الالوان	يستخدم مع Raster Scan System لعرض الالوان
الالوان	يمكن أن تعرض أربعة ألوان فقط ، أي الأحمر والأخضر والبرتقالي والأصفر	يمكنه عرض ملايين الألوان
اعتمادية اللون	اعتمادية اللون تتوفر ألوان أقل لأن الألوان في Beam Penetration تعتمد على سرعة شعاع الإلكترون	الملايين من الألوان متوفرة لأن الألوان في Shadow Mask تعتمد على نوع الشعاع.
التكلفة	التكلفة أقل تكلفة مقارنة بقناع الظل	إنه أغلى من الطرق الأخرى
جودة الصورة	جودة الصورة ليست جيدة ، أي ضعيفة مع طريقة اختراق الشعاع	يمنحك Shadow Mask الواقعية في الصورة مع تأثير الظل وملايين الألوان
الدقة	دقة عالية	دقة منخفضة

1. MDA (محول العرض أحادي اللون): مراقب مع عرض النص وأنواع الألوان ، مثل الأحمر أو الأخضر أو الأزرق. بدقة 720 × 350

2. CGA (محول رسومات الألوان): شاشة ملونة صنعتها شركة IBM لأول مرة في عام 1981 ، وهي تتعرف على النصوص والرسومات. مخطط الدقة 320 × 200 بكسل. الحرف لديه كثافة بكسل 7 × 9 نقاط.

3. EGA (محول الرسومات المحسن): شاشة ملونة ، تم تصنيعها عام 1981 ، بالتزامن مع إصدار IBM PC AT. مخطط الدقة 640 × 340 بكسل. الحرف لديه كثافة بكسل 7 × 9 نقاط.

4. VGA (صفيف رسومات الشاشة): شاشة ملونة ، تم تصنيعها بواسطة شركة IBM في عام 1986 جنبًا إلى جنب مع قرار IBM 256 ، بها 720 × 400 بكسل. كثافة الحرف 9 × 16 نقطة.

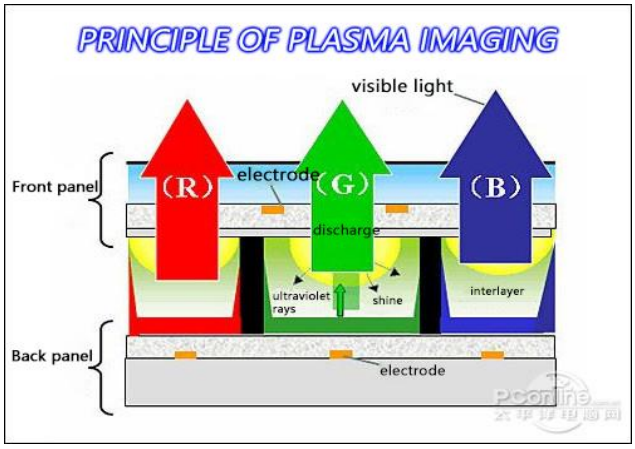
2-الشاشة المسطحة

تم استخدام مصطلح الشاشة المسطحة مؤخرًا والذي يشير إلى فئة من أجهزة الفيديو التي قللت من حيث الحجم والوزن ومتطلبات الطاقة مقارنةً بشاشات CRT ، نظرًا لأن الشاشة المسطحة أرق من شاشات CRT ، فيمكننا تعليقها على الجدران أو ارتداء في اليد. نظرًا لأنه يمكننا الكتابة على بعض شاشات العرض المسطحة ، فستتوفر قريبًا كمفكرات جيب. يتم تصنيف اللوحة المسطحة إلى نوعين:

1. الشاشات الباعثة: العرض أو الباعث المنبعث عبارة عن أجهزة تحول الطاقة الكهربائية إلى ضوء. مثل لوحة البلازما والصمامات الثنائية الباعثة للضوء.

2. الشاشات غير الباعثة: تستخدم شاشات العرض غير المنبعثة أو غير الباعثة تأثيرات بصرية لتحويل ضوء الشمس أو الضوء من مصدر آخر إلى أنماط رسومية. مثل شاشات الكريستال السائل (شاشة الكريستال السائل).

1- شاشات البلازما



1. يتم تصنيعه من خلال آلاف الأنبوب الزجاجي المفرغ والمملوء بمزيج من غازات (Ne، He، Xe).
2. تم تلوين الأنابيب بالفوسفور الأحمر والأخضر والأزرق وتوضع بين الطبقة الزجاجية الأمامية والنهائية. يتم توصيل كل أنبوب من طرفيه ، الواجهة الأمامية للعرض والنهاية للتوجيه. الغاز عبارة عن أيون من خلال الجهد الكهربائي الموجه إلى كلا الطرفين لذلك يحدث تفريغ كهربائي في الخلية. ويؤثر ذلك على أن يكون الغاز أيونًا ويتحول إلى بلازما وتنبعث الأشعة فوق البنفسجية.
3. يتم تخزين تعريف الصورة في مخزن مؤقت للتحديث ويتم تطبيق جهد إطلاق النار لتحديث مواضع البكسل ، 60 مرة في الثانية.
4. تُستخدم الطرق الحالية المتناوبة لتوفير تطبيق أسرع لجهود إطلاق النار وبالتالي عروض أكثر إشراقًا.

❖ مزايا عرض البلازما

1. مُصنَّع بحجم ضخم يسمح باستخدامه في المسرح والملاعب والمتنزهات.
2. وزن خفيف وسمك صغير مما يساعد المستخدم على تعليقه على الحائط.
3. شاشة مسطحة مع صورة واضحة وألوان واضحة ودقة عالية.
4. بدون وجود أشعة كهرومغناطيسية تحيط بالشاشة ، لذلك يمكن استخدام نظام الصوت بدون تأثير الأشعة.
5. زاوية رؤية واضحة تقارب 160.

❖ عيوب شاشة البلازما

1. لا يمكن تصنيع شاشة البلازما بحجم صغير لأنه من المستحيل القضاء على المسافة بين البكسل كما يجب أن يكون الزجاج الأنبوبي سميكًا لأسباب فنية.
2. شاشة البلازما لها عمر افتراضي لأن طبقة الفوسفور المطلية بالأنبوب الداخلي تتحلل بفعل اصطدام جزيئات الغاز.

2-شاشة الصمام الثنائي الباعث للضوء العضوي (OLED / LED)

1. OLED شاشة جديدة للغاية تعتمد على تقنية حديثة جدًا تسمى الصمام الثنائي العضوي للضوء. ينتج الصمام الثنائي من مادة عضوية رفيعة جدًا تعرف بالبوليمرات العضوية ، وعنصرها الرئيسي الكربون بسمك يبلغ حوالي 500 نانومتر.
2. في هذا العرض ، يتم ترتيب مصفوفة من الصمام الثنائي الباعث للضوء متعدد الألوان لتشكيل موضع البكسل في الشاشة. ويتم تخزين تعريف الصورة في المخزن المؤقت للتحديث.
3. على غرار تحديث خط المسح الضوئي ، تتم قراءة معلومات CRT من المخزن المؤقت للتحديث وتحويلها إلى مستويات الجهد التي يتم تطبيقها على الثنائيات لإنتاج نمط الضوء على الشاشة. يتم تصنيف OLED إلى العديد من الفئات أهمها ما يلي:

- مصفوفة سلبية OLED: تستخدم لتصنيع الشاشة الصغيرة مثل شاشة الجوال واللوحة ولعبة الفيديو.
- Active matrix OLED: يستخدم لتصنيع شاشة كبيرة مثل شاشة التلفاز والكمبيوتر.
- شاشة OLED الشفافة: وهي شفافة تستخدم لتصنيع حاجب الريح للسيارة والطائرة.
- OLED قابل للطي: إنه مقاوم للكسر لذا يستخدم مع سطح المختبر.
- أبيض OLED: يستخدم مع أنظمة الإضاءة الحديثة لخفة وزنه وقلة استهلاكه للطاقة ووميضه العالي للغاية.

❖ مزايا OLED

1. الطبقات البلاستيكية العضوية في OLED أرق وأخف وزنًا وأكثر مرونة.
2. المصابيح أكثر إشراقًا من المصابيح.
3. لا تتطلب OLEDs إضاءة خلفية بينما تولد OLED الضوء بنفسها. نظرًا لأن OLEDs لا تتطلب إضاءة خلفية ، فإنها تستهلك طاقة أقل بكثير وهذا مهم بشكل خاص للأجهزة التي تعمل بالبطاريات مثل الهواتف المحمولة.
4. OLEDs أسهل في الإنتاج ويمكن تصنيعها بأحجام أكبر. نظرًا لأن OLEDs عبارة عن مواد بلاستيكية بشكل أساسي ، فيمكن تحويلها إلى صفائح رفيعة وكبيرة.
5. تتمتع شاشات OLED بمجال رؤية كبير ، حوالي 170 درجة. لديهم نطاق رؤية أوسع بكثير.

❖ عيوب OLED

1- مدى الحياة - بينما تتمتع أفلام OLED باللونين الأحمر والأخضر بعمر أطول (46.000 إلى 230.000 ساعة) ، فإن المواد العضوية الزرقاء تتمتع حاليًا بأعمار أقصر بكثير (تصل إلى حوالي 14000 ساعة.

2- عمليات التصنيع مكلفة الآن. 3- يمكن أن يؤدي الماء إلى إتلاف OLEDs بسهولة.

2- شاشة الكريستال السائل (LIQUID CRYSTAL DISPLAY LCD)

1. يستخدم بشكل عام في نظام صغير مثل الآلة الحاسبة والكمبيوتر المحمول.
2. ينتج هذا الجهاز غير المنبعث صورة عن طريق تمرير الضوء المستقطب من المحيط أو من مصدر ضوء داخلي عبر مادة الكريستال السائل التي يمكن محاذاة إما لحجب الضوء أو نقله.
3. يحدد تقاطع موصلين موضع البكسل.
4. في حالة التشغيل ON يكون الضوء المستقطب الذي يمر عبر المادة ملتويًا بحيث يمر عبر المستقطب المعاكس. في حالة إيقاف التشغيل ، ستنعكس مرة أخرى نحو المصدر.
5. يشار إلى هذا النوع من الأجهزة المسطحة باسم شاشة LCD ذات المصفوفة السلبية.
6. في المصفوفة النشطة ، يتم استخدام ترانزستورات LCD عند كل نقطة شبكة (س ، ص).
7. يتسبب الترانزستور في تغيير البلور لحالته بسرعة وأيضًا للتحكم في درجة تغير الحالة.
8. يمكن أن يكون الترانزستور أيضًا بمثابة ذاكرة حتى يتم تغييرها. لذا فإن الترانزستور يجعل الخلية تعمل طوال الوقت لإعطاء شاشة أكثر إشراقًا ، فسيكون ذلك إذا كان يجب تحديثها بشكل دوري.

❖ مزايا شاشة LCD

1. تكلفة منخفضة. 2. الوزن المنخفض. 3. صغر الحجم 4. انخفاض استهلاك الطاقة.
5. شاشة مسطحة مع صورة واضحة وألوان واضحة ودقة عالية. 6. دون الأشعة الكهرومغناطيسية تحيط الشاشة.
7. شاشة كبيرة يمكن تعليقها على الحائط. 8. زاوية الرؤية: تصل إلى 165 درجة ، الصورة تعاني من الجانب.

- عيوب شاشة LCD

1. تم إصلاح نسبة العرض إلى الارتفاع والقرار. 2. تباين أقل من CRTs بسبب ضعف مستوى اللون الأسود.
3. تؤدي أوقات الاستجابة البطيئة وتحويل معدل المسح الضوئي إلى تشوهات شديدة في الحركة وتدهور للصورة لنقل الصور أو تغييرها بسرعة.

LCD

المحاضرة 3: أنواع شاشات العرض ثلاثية الأبعاد

شاشات العرض الثلاثية الأبعاد (3D Display Screens) هي نوع من الشاشات التي تعرض الصور والفيديوهات بطريقة تعطي إحساسًا بالعمق، مما يجعل المشاهد يرى الصور وكأنها ثلاثية الأبعاد. تختلف هذه الشاشات عن الشاشات الثنائية الأبعاد التقليدية في أنها تعرض صورتين مختلفتين قليلاً، واحدة لكل عين، وهذا الاختلاف في الزاوية يسمح للدماغ بتكوين إحساس بالعمق.

أنواع شاشات العرض الثلاثية الأبعاد

1- **الشاشات المجسمة (Stereoscopic 3D Displays)** هي النوع الأكثر شيوعاً من شاشات العرض الثلاثية الأبعاد، وتعمل على إنشاء تأثير ثلاثي الأبعاد من خلال عرض صورتين مختلفتين قليلاً، واحدة

لكل عين. هذه الشاشات تعتمد على حقيقة أن كل عين ترى الأشياء من زاوية مختلفة قليلاً، مما يساعد الدماغ في تفسير العمق.

- كيفية العمل

الشاشات المجسمة تعرض صورتين مختلفتين، واحدة موجهة للعين اليسرى وأخرى للعين اليمنى. الفرق الطفيف بين الصورتين (المعروف باسم المنظر الثنائي) يخلق انطباعاً بالعمق عند دمجهما في الدماغ. لكي يتم فصل الصورتين بشكل صحيح وتوجيه كل واحدة للعين المناسبة، يتم استخدام نظارات خاصة.

- التقنيات المستخدمة في الشاشات المجسمة

1. **نظارات الأناجليف (Anaglyph 3D Glasses):**

- كيفية العمل: هذه النظارات تستخدم عدسات ملونة (عادةً حمراء وزرقاء أو حمراء وخضراء) لتصفية الصورتين المعروضتين على الشاشة بحيث يتم توجيه كل صورة إلى العين المناسبة.
- المزايا: رخيصة وسهلة الإنتاج.
- العيوب: جودة الألوان تكون محدودة؛ قد تكون الصور باهتة وغير واقعية.

2. نظارات الاستقطاب (Polarized 3D Glasses):

– كيفية العمل: في هذه التقنية، يتم استخدام شاشة تعرض صورتين مستقطبتين بشكل مختلف (إحدى الصور مستقطبة أفقياً والأخرى عمودياً)، وتقوم النظارات المستقطبة بتوجيه الصورة المناسبة لكل عين.

– المزايا: جودة صورة أفضل مقارنة بالأناجليف، وتوفر ألوان أكثر واقعية.

– العيوب: يتطلب شاشة خاصة ووجود نظارات خاصة، وقد تتأثر جودة الصورة بزاوية المشاهدة.

3. **نظارات المصراع النشط (Active Shutter 3D Glasses)**:

– كيفية العمل: تعمل هذه التقنية بالتزامن مع الشاشة، حيث تقوم النظارات بإغلاق وفتح عدساتها بالتناوب مع كل إطار من الصورة المعروضة على الشاشة. هذه العملية تحدث بسرعة فائقة بحيث ترى كل عين الصورة المخصصة لها فقط.

– المزايا: جودة صورة عالية جداً مع دقة وألوان ممتازة.

– العيوب: تتطلب نظارات إلكترونية تحتاج إلى شحن، وغالباً ما تكون هذه النظارات أكثر تكلفة.

– **التطبيقات**

1. الأفلام ثلاثية الأبعاد: في السينما المنزلية ودور السينما التجارية.
2. ألعاب الفيديو: لتوفير تجربة غامرة وواقعية.
3. العروض التقديمية والمؤتمرات: لعرض محتوى ثلاثي الأبعاد.

4. التعليم: في تعليم العلوم والهندسة، حيث يمكن عرض النماذج ثلاثية الأبعاد بشكل أكثر تفصيلاً.

2- الشاشات المجسمة الذاتية (Autostereoscopic 3D Displays) هي نوع من شاشات العرض الثلاثية الأبعاد التي توفر تجربة مشاهدة ثلاثية الأبعاد دون الحاجة إلى ارتداء نظارات خاصة. هذا النوع من الشاشات يحل مشكلة عدم الراحة التي تسببها النظارات في الشاشات المجسمة التقليدية.

- كيفية العمل

تعمل الشاشات المجسمة الذاتية باستخدام تقنيات خاصة تقوم بتوجيه الصور المختلفة إلى كل عين بشكل منفصل، بحيث يتم إنشاء تأثير ثلاثي الأبعاد دون الحاجة لنظارات. التقنية الأساسية هي عرض صورتين أو أكثر من زوايا مختلفة على الشاشة، وتوجيه كل صورة إلى العين المناسبة.

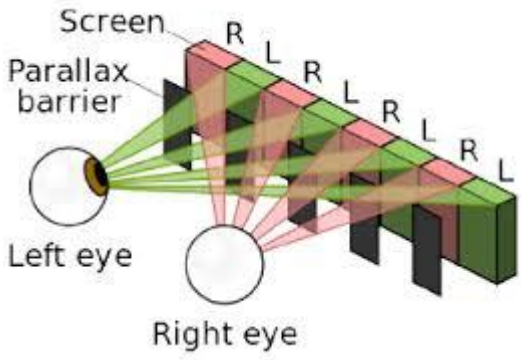
- التقنيات المستخدمة في الشاشات المجسمة الذاتية

1. حواجز الحواجز البصرية (Parallax Barrier):

- كيفية العمل: تتكون هذه التقنية من حاجز يحتوي على فتحات دقيقة توضع أمام الشاشة. عندما تنظر إلى الشاشة، تقوم الفتحات بتوجيه الصور المختلفة إلى كل عين بشكل منفصل، مما يخلق تأثيراً ثلاثي الأبعاد.

- المزايا: بسيطة من حيث المبدأ ولا تحتاج إلى أجهزة إضافية للمشاهد.

- العيوب: زاوية المشاهدة محدودة جداً، مما يعني أنه يجب على المشاهدين الوقوف أو الجلوس في موقع معين للحصول على التأثير ثلاثي الأبعاد.



2. العدسات متعددة الأوجه (Lenticular Lenses):

- كيفية العمل: تستخدم هذه التقنية مجموعة من العدسات الصغيرة الموضوعة أمام الشاشة والتي تقوم بكسر الضوء لتوجيه الصور المختلفة إلى كل عين. يتم عرض الصور من زوايا متعددة بحيث يمكن للمشاهد رؤية تأثير ثلاثي الأبعاد من زوايا مختلفة.

– المزايا: توفر زاوية مشاهدة أوسع مقارنة بحواجز الحواجز البصرية، وتجربة مشاهدة أكثر راحة.

– العيوب: يمكن أن تكون التكلفة أعلى، كما أن جودة الصورة قد تتأثر إذا لم يكن المشاهد في الموقع المثالي.

3. تقنيات الرؤية المتعددة (Multiple Viewpoint Technology):

– كيفية العمل: هذه التقنية تعرض صورًا متعددة من زوايا مختلفة على الشاشة، وتستخدم خوارزميات متقدمة لتوجيه كل صورة إلى عين المشاهد بناءً على موقعه بالنسبة للشاشة.

– المزايا: مرونة عالية وزاوية مشاهدة واسعة جداً، وتوفر تجربة أكثر طبيعية.

– العيوب: هذه التقنية تحتاج إلى معالجة معقدة وقد تكون مكلفة، كما أن بعض النسخ منها قد تحتاج إلى كاميرات أو مستشعرات لتتبع حركة المستخدم.

– التطبيقات

1. الهواتف الذكية: بعض الهواتف تستخدم الشاشات المجسمة الذاتية لتوفير تجربة ثلاثية الأبعاد دون الحاجة إلى نظارات.
2. شاشات الألعاب: بعض الشاشات المصممة للألعاب تستخدم هذه التقنية لتوفير تجربة لعب غامرة.
3. العروض التجارية والمعارض: تستخدم هذه الشاشات لجذب الانتباه من خلال عرض محتوى ثلاثي الأبعاد تفاعلي.
4. الواقع الافتراضي والواقع المعزز: تستخدم لتوفير تأثير ثلاثي الأبعاد بدون الحاجة إلى نظارات.

3–الشاشات الحجمية (Volumetric Displays) هي نوع من شاشات العرض التي تعرض صورًا ثلاثية الأبعاد حقيقية يمكن رؤيتها من زوايا مختلفة، مما يسمح للمشاهدين برؤية الكائنات الثلاثية الأبعاد بشكل كامل كما لو كانت موجودة فعليًا في الفضاء. على عكس الشاشات المجسمة التي تعتمد على خداع الدماغ لإدراك العمق، فإن الشاشات الحجمية تقدم صورًا مجسمة حقيقية في ثلاثة أبعاد.

– كيفية العمل

الشاشات الحجمية تعمل من خلال إنشاء صور في حجم ثلاثي الأبعاد باستخدام تقنيات مثل عرض الضوء على وسط مادي أو خلق صور في الفضاء باستخدام الليزر أو مصفوفات LED. المشاهدون يمكنهم رؤية الصور من جميع الزوايا، مما يعني أنهم يمكنهم التحرك حول الشاشة لرؤية الكائن من جوانب مختلفة.

– التقنيات المستخدمة في الشاشات الحجمية

1. الشاشات الحجمية القائمة على الـ LED:

– كيفية العمل: تستخدم هذه التقنية مصفوفة من مصابيح LED مرتبة بشكل ثلاثي الأبعاد أو على سطح دوار. عندما تدور المصفوفة بسرعة عالية، يتم عرض الصور بشكل متسلسل بحيث تُكون صورة ثلاثية الأبعاد مجسمة يمكن رؤيتها من جميع الزوايا.

– المزايا: توفر زاوية مشاهدة 360 درجة وتتيح التفاعل مع الصور من زوايا متعددة.

– العيوب: حجم العرض يمكن أن يكون محدودًا، كما أن التفاصيل قد تكون أقل من التقنيات الأخرى.

2. **الشاشات الهولوجرافية (Holographic Displays):**

– كيفية العمل: تعتمد هذه التقنية على استخدام الليزر لخلق نمط تداخل ضوئي في وسط معين (مثل الهواء أو مواد شفافة) والذي يعيد تكوين الصورة الثلاثية الأبعاد.

– المزايا: تقدم صورًا ثلاثية الأبعاد معقدة وعالية الجودة يمكن رؤيتها من أي زاوية.

– العيوب: هذه التقنية معقدة جدًا ومكلفة وتتطلب تجهيزات دقيقة.

3. الشاشات الحجمية التفاعلية:

– كيفية العمل: هذه الشاشات تستخدم تقنيات مثل الكاميرات وأجهزة الاستشعار للتفاعل مع المستخدمين، مما يسمح لهم بالتفاعل مع الصور الثلاثية الأبعاد في الوقت الحقيقي. يمكن للمستخدمين إدارة أو تعديل الكائنات الظاهرة على الشاشة بالحركة أو الإيماءات.

– المزايا: تتيح التفاعل المباشر مع الصور الثلاثية الأبعاد، مما يوفر تجربة غامرة وتفاعلية.

- العيوب: تتطلب تجهيزات وأجهزة إضافية، مما يزيد من التكلفة والتعقيد.

- **التطبيقات**

1. الطب: تُستخدم هذه الشاشات في التصوير الطبي لعرض الأعضاء والهياكل المعقدة بشكل ثلاثي الأبعاد، مما يساعد الأطباء على التخطيط للجراحات أو تشخيص الحالات.
2. التصميم الهندسي: تسمح بعرض النماذج المعمارية أو التصاميم الصناعية في شكل ثلاثي الأبعاد كامل يمكن فحصه من جميع الزوايا.
3. الترفيه والألعاب: تقدم تجربة مشاهدة متقدمة في المتاحف أو المعارض التفاعلية، حيث يمكن للمشاهدين رؤية المعروضات من جميع الزوايا.
4. التعليم: تُستخدم في الفصول الدراسية لعرض المفاهيم العلمية بشكل ثلاثي الأبعاد، مما يساعد على تحسين فهم الطلاب.

4-الشاشات الهولوغرافية (Holographic Displays) هي نوع متقدم من شاشات العرض الثلاثية الأبعاد التي تستخدم تقنيات تعتمد على تداخل الضوء لإنشاء صور ثلاثية الأبعاد يمكن رؤيتها من زوايا مختلفة، دون الحاجة إلى نظارات خاصة. هذه التقنية قادرة على عرض صور مجسمة حقيقية تظهر وكأنها معلقة في الفضاء.

- **كيفية العمل**

الشاشات الهولوغرافية تعتمد على تقنية الهولوغرافيا، التي تعمل عبر تسجيل وإعادة بناء الموجات الضوئية المنعكسة من كائن ما. يتم ذلك عن طريق تسجيل أنماط تداخل الضوء بين شعاع الليزر المنعكس عن الكائن وشعاع مرجعي. هذا النمط (الهولوجرام) يمكن إضاءته بعد ذلك باستخدام مصدر ضوء مناسب لإعادة تكوين الصورة ثلاثية الأبعاد في الفضاء.

- **أنواع الشاشات الهولوغرافية وتقنياتها**

1. الهولوغرام التقليدي:

- كيفية العمل: يتم تسجيل نمط التداخل الضوئي على وسط حساس للضوء (مثل فيلم أو لوحة فوتوغرافية). عند إضاءة الهولوجرام بضوء ليزر مماثل، يتم إعادة إنشاء صورة ثلاثية الأبعاد للكائن.
- المزايا: هذه الطريقة تُنتج صوراً ثلاثية الأبعاد دقيقة وحقيقية.
- العيوب: تتطلب عملية إنتاج معقدة واستخدام ليزر لتسجيل وعرض الصور.

2. **الشاشات الهولوغرافية الرقمية (Digital Holography):**

– كيفية العمل: تستخدم أجهزة العرض الرقمية مثل شاشات LCD أو DLP لعرض الهولوجرامات، والتي يتم إنشاؤها باستخدام برمجيات الكمبيوتر. هذا يسمح بعرض صور متحركة ثلاثية الأبعاد.

– المزايا: مرونة عالية، حيث يمكن تغيير المحتوى الهولوغرافي بسهولة، وإمكانية عرض الفيديوهات الهولوغرافية.

– العيوب: تتطلب معالجة بيانات كبيرة ومعقدة، وقد تكون دقة الصورة محدودة بسبب قيود تقنية العرض.

3. الشاشات الهولوغرافية الحجمية:

– كيفية العمل: تعتمد على استخدام وسط مادي، مثل جزيئات الهواء أو مواد شفافة، التي يمكن التحكم فيها لإعادة إنشاء صورة ثلاثية الأبعاد حقيقية في الفضاء.

– المزايا: توفر تجربة مشاهدة ثلاثية الأبعاد كاملة يمكن رؤيتها من جميع الزوايا.

– العيوب: تقنية معقدة ومكلفة، وغالباً ما يكون حجم الصورة محدودًا.

4. الهولوجرامات الميدانية الضوئية (Light Field Displays):

– كيفية العمل: تُنشئ هذه التقنية صورة ثلاثية الأبعاد عن طريق عرض آلاف من الصور الفردية المتراكبة من زوايا مختلفة، مما يتيح رؤية الصورة من زوايا متعددة.

– المزايا: يمكن أن تعرض صورًا ثلاثية الأبعاد مع عمق وتفاصيل عالية.

– العيوب: تتطلب قدرة حوسبة عالية جدًا وبيانات ضخمة.

– **التطبيقات**

1. الطب: تُستخدم لعرض صور ثلاثية الأبعاد للأعضاء والهياكل البيولوجية، مما يساعد الأطباء في التخطيط للجراحات.
2. التعليم: تُستخدم لعرض مفاهيم علمية معقدة بشكل ثلاثي الأبعاد.
3. الترفيه والإعلانات: تستخدم في الحفلات الموسيقية، المتاحف، والعروض الترويجية لخلق تأثير بصري مميز.
4. الاجتماعات الافتراضية: تُستخدم في إنشاء صور ثلاثية الأبعاد للمشاركين عن بعد، مما يحسن من تفاعل الاجتماعات الافتراضية.

محاضرة 4 :خوارزميات رسم الخط المستقيم

1- استخدام معادلة المستقيم (ميل ونقطة):

تستخدم هذه الطريقة عند عدم وجود نظرية في الحل. وتعرف معادلة المستقيم بـ $y = mx + b$.

ويعرف الميل بانه مقدار انحدار الخط المستقيم ويحسب بـ $m = \frac{\Delta y}{\Delta x}$

في البدء يجب استخراج مقدار الميل من معادلته باستخدام نقطتي البداية والنهاية للمستقيم

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$

ثم تعويض قيمة الميل m في معادلة المستقيم مع احدى النقاط لاستخراج قيمة الثابت b:

$$y_1 = mx_1 + b$$

$$\therefore b = y_1 - mx_1 \quad \text{أو} \quad b = y_2 - mx_2$$

ثم تطبيق معادلة المستقيم باستخدام m و b ونقطة البداية للمستقيم لاستخراج باقي نقاط المستقيم بحيث تحسب y من $y = mx + b$ وأن $x_1 \le x \le x_2$ بزيادة 1.

مثال: ارسم الخط المستقيم باستخدام معادلة المستقيم للنقطتين P1(1,2), P2(5,6)؟

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{6-2}{5-1} = \frac{4}{4} = 1$$

$b = y_1 - mx_1 = 2 - 1 * 1 = 1$

X	Y = mx + b	Point
1	Y = 1*1+1=2	(1,2)
2	Y = 1*2+1=3	(2,3)
3	Y = 1*3+1=4	(3,4)
4	Y = 1*4+1=5	(4,5)
5	Y = 1*5+1=6	(5,6)

2-خوارزمية برزنهام لرسم الخط المستقيم:

تعتمد هذه الخوارزمية على الارقام الصحيحة فقط وكذلك الاستفادة من خاصية الزيادة بواحد.

الطريقة: يحسب الميل اولاً من $m = \frac{\Delta y}{\Delta x}$ وتثبت النقطة الاولى.

- **if |m| <1**

اذا كان الميل أقل من 1 تكون $x_1 \le x \le x_2$ بزيادة 1، ويتم حساب y من:-

$d_i = 2dy-dx$

if $d_i \ge 0$ then $y_i = y_{i-1} + 1$

$d_{i+1} = d_i + 2(dy-dx)$

if $d_i < 0$ then $y_i = y_{i-1}$

$d_{i+1} = d_i + 2(dy)$

- **if |m| >1**

اذا كان الميل أكبر من 1، نقلب القوانين السابقة بحيث كل x تصبح y وكل y تصبح x.

أي أن $y_1 \le y \le y_2$ بزيادة 1، ويتم حساب x من:

$d_i = 2dx-dy$

if $d_i \ge 0$ then $x_i = x_{i-1} + 1$

$d_{i+1} = d_i + 2(dx-dy)$

if $d_i < 0$ then $x_i = x_{i-1}$

$d_{i+1} = d_i + 2(dx)$

- **if |m| =1**

اذا كان الميل يساوي 1، فان الحل سيكون صحيحاً بأي طريقة من الطريقتين السابقتين.

مثال 1 : استخدم خوارزمية برزنهام لرسم الخط المستقيم للنقطتين P1(5,8) , P2(9,11)؟

$$m = \frac{\Delta y}{\Delta x} = \frac{11-8}{9-5} = \frac{3}{4} = 0.75 < 1$$

$x_1 \le x \le x_2$

بزيادة 1 وحساب y

x	y
5	8
6	9
7	10
8	10
9	11

$d_i = 2dy-dx = 2 * 3 - 4 = 2 > 0$

$y_i = y_{i-1} + 1 = 8 + 1 = 9$

$d_{i+1} = d_i + 2(dy-dx) = 2 + 2\ (3 - 4) = 0$

$y_i = y_{i-1} + 1 = 9 + 1 = 10$

$d_{i+1} = d_i + 2(dy-dx) = 0 + 2\ (3 - 4) = -2 < 0$

$y_i = y_{i-1} = 10$

$d_{i+1} = d_i + 2(dy) = -2 + 2 * 3 = 4 > 0$

$y_i = y_{i-1} + 1 = 10 + 1 = 11$

مثال 2: استخدم خوارزمية برزنهام لرسم الخط المستقيم للنقطتين P1(5,2) , P2(9,7)؟

$$m = \frac{\Delta y}{\Delta x} = \frac{7-2}{9-5} = \frac{5}{4} = 1.25 > 1$$

x

تقلب القوانين و $y_1 \le y \le y_2$ وحساب 1

x	y
5	2
6	3
7	4
7	5
8	6
9	7

$= 2dx-dy = 2 * 4 - 5 = 3 > 0$

$x_i = x_{i-1} + 1 = 5 + 1 = 6$

$d_{i+1} = d_i + 2(dx-dy) = 3 + 2\,(4 - 5) = 1 > 0$

$= x_{i-1} + 1 = 6 + 1 = 7$

$d_{i+1} = d_i + 2(dx-dy) = 1 + 2\,(4 - 5) = -1 < 0$

$x_i = x_{i-1} = 7$

$d_{i+1} = d_i + 2(dx) = -1 + 2 * 4 = 7 > 0$

$x_i = x_{i-1} + 1 = 7 + 1 = 8$

$d_{i+1} = d_i + 2(dx-dy) = 7 + 2\,(4 - 5) = 5 > 0$

$x_i = x_{i-1} + 1 = 8 + 1 = 9$

مثال 3: استخدم خوارزمية برزنهام لرسم الخط المستقيم للنقطتين (P1(-5,2 , P2(-9,7؟

$$m = \frac{\Delta y}{\Delta x} = \frac{7-2}{-9+5} = \frac{5}{-4} = |-1.25| > 1$$

بزيادة x وحساب 1 $y_1 \le y \le y_2$ تقلب القوانين وفي حالة تناقص x مع ملاحظة

NOTE: $dx = |dx|$

x	y
–5	2
–6	3
–7	4
–7	5
–8	6
–9	7

$dy_i = 2dx-dy = 2 * 4 - 5 = 3 > 0$

$x_i = x_{i-1} - 1 = -5 - 1 = -6$

$d_{i+1} = d_i + 2(dx-dy) = 3 + 2\,(4 - 5) = 1 > 0$

$x_i = x_{i-1} - 1 = -6 - 1 = -7$

$d_{i+1} = d_i + 2(dx-dy) = 1 + 2\,(4 - 5) = -1 < 0$

$x_i = x_{i-1} = -7$

$d_{i+1} = d_i + 2(dx) = -1 + 2 * 4 = 7 > 0$

$x_i = x_{i-1} - 1 = -7 - 1 = -8$

$d_{i+1} = d_i + 2(dx-dy) = 7 + 2\,(4 - 5) = 5 > 0$

$x_i = x_{i-1} - 1 = -8 - 1 = -9$

H.W.: استخدم خوارزمية برزنهام لرسم الخط المستقيم للنقطتين (P1(2,2 , P2(7,7؟

المحاضرة 5 خوارزميات الخط المستقيم

3-خوارزمية DDA (Digital Differential Analyzer):

اســاس هذه النظرية هو ان نحســب احداثي بدلالة الاحداثي الاخر اعتماداً على قيمة الميل. ولها الصــفات التالية:

1- تعتمد بصورة كاملة على الميل والاتجاهات.

2- تفشل في حساب دقة النهايات للخط المستقيم.

3- تعاني من انه يجب أن تتم الحسابات باستخدام رياضيات الفارزة العشرية.

- **if $|m| \le 1$** **الحالــة الاولــى**

تكون $x_1 \le x \le x_2$ بزيادة 1، ونستخرج قيمة y حسب التالي:

بما ان الزيادة على الاحداثي السيني هي 1، أي أن الفرق بين أي قيمتين من قيم x هي:

$$\Delta x = 1$$

$$m = \frac{\Delta y}{\Delta x}$$

$$\Delta y = m\Delta x$$

$$\Delta y = m$$

$$y_{i+1} - y_i = m$$

$$\therefore y_{i+1} = y_i + m$$

ملاحظـــة:

If $|m| \le 1$ and $y_1 < y_2$ (تزايد) then $y_{i+1} = y_i + m$

If $|m| \le 1$ and $y_1 > y_2$ (تناقص) then $y_{i+1} = y_i - m$

مثال 1: ارسم الخط المستقيم بين النقطتين P1(5,7) , P2(8,9) باستخدام DDA؟

$$m = \frac{\Delta y}{\Delta x} = \frac{9-7}{8-5} = \frac{2}{3} = 0.6 < 1$$

$5 \le x \le 8$ بزيادة 1

$y_1 < y_2$ (تزايد) $\Rightarrow 7<8$

نستخدم $y_{i+1} = y_i + m$

x	$y_{i+1} = y_i + m$	point
5	7	(5,7)
6	y=7+0.6=7.6	(6,8)
7	y=7.6+0.6=8.2	(7,8)
8	y=8.2+0.6=8.8	(8,9)

مثال 2: ارسم الخط المستقيم بين النقطتين P1(5,7) , P2(8,5) باستخدام DDA؟

$$m = \frac{\Delta y}{\Delta x} = \frac{5-7}{8-5} = \frac{-2}{3} = -0.6$$

$|m| < 1$

$5 \le x \le 8$ بزيادة 1

$y_1 > y_2$ (تناقص) $\Rightarrow 7>5$

نستخدم $y_{i+1} = y_i - m$

x	$y_{i+1} = y_i - m$	point
5	7	(5,7)
6	y=7-0.6=6.4	(6,6)
7	y=6.4-0.6=5.8	(7,6)
8	y=5.8-0.6=5.2	(8,5)

مثال 3: ارسم الخط المستقيم بين النقطتين P1(8,10) , P2(2,6) باستخدام DDA؟

$$m = \frac{\Delta y}{\Delta x} = \frac{6-10}{2-8} = \frac{-4}{-6} = 0.6 < 1$$

$8 \ge x \ge 2$ بتناقص 1

$y_1 > y_2$ (تناقص) $\Rightarrow 10>6$

نستخدم $y_{i+1} = y_i - m$

x	$y_{i+1} = y_i - m$	point
8	10	(8,10)
7	y =10-0.6=9.4	(7,9)
6	y =9.4-0.6=8.8	(6,9)
5	y =8.8-0.6=8.2	(5,8)
4	y =8.2-0.6=7.6	(4,8)
3	y =7.6-0.6=7	(3,7)
2	y =7-0.6=6.4	(2,6)

- **if |m| >1** **الحالــة الثانيــة**

تكون $y_1 \le y \le y_2$ بزيادة 1، ونستخرج قيمة x حسب التالي:

بما ان الزيادة على الاحداثي الصادي هي 1، أي أن الفرق بين أي قيمتين من قيم y هي:

$$\Delta y = 1$$

$$m = \frac{\Delta y}{\Delta x}$$

$$m\Delta x = 1$$

$$m(x_{i+1} - x_i) = 1$$

$$x_{i+1} - x_i = \frac{1}{m}$$

$$\therefore x_{i+1} = x_i + \frac{1}{m}$$

ملاحظـــة:

If |m| > 1 and $x_1 < x_2$ (تزايد) then $x_{i+1} = x_i + \frac{1}{m}$

If |m| > 1 and $x_1 > x_2$ (تناقص) then $x_{i+1} = x_i - \frac{1}{m}$

مثال 1: ارسم الخط المستقيم بين النقطتين P1(3,2) , P2(8,10) باستخدام DDA؟

$$m = \frac{\Delta y}{\Delta x} = \frac{10-2}{8-3} = \frac{8}{5} = 1.6 > 1$$

$$\frac{1}{m} = \frac{5}{8} = 0.625$$

$2 \le y \le 10$ بزيادة 1

$x_1 < x_2$ (تزايد) $\Rightarrow 3<8$

نستخدم $x_{i+1} = x_i + \frac{1}{m}$

y	$x_{i+1} = x_i + \frac{1}{m}$	point
2	3	(3,2)
3	x =3+0.625=3.625	(4,3)
4	x =3.625+0.625=4.25	(4,4)
5	x =4.25+0.625=4.875	(5,5)
6	x =4.875+0.625=5.5	(6,6)
7	x =5.5+0.625=6.125	(6,7)
8	x =6.125+0.625=6.75	(7,8)
9	x =6.75+0.625=7.375	(7,9)
10	x =7.375+0.625=8	(8,10)

مثال 2: ارسم الخط المستقيم بين النقطتين P1(5,8) , P2(9,5) باستخدام DDA؟

$$m = \frac{\Delta y}{\Delta x} = \frac{5-8}{9-5} = \frac{-3}{4} = |-0.75| < 1$$

$5 \le x \le 9$ بزيادة 1

x	$y_{i+1} = y_i - m$	point
5	8	(5,8)
6	y =8-0.75=7.25	(6,7)
7	y =7.25-0.75=6.5	(7,7)
8	y =6.5-0.75=5.75	(8,6)
9	y =5.75-0.75=5	(9,5)

$y_1 > y_2$ (تناقص) $\Rightarrow 8 > 5$

نستخدم $y_{i+1} = y_i - m$

مثال 3: ارسم الخط المستقيم بين النقطتين P1(0,0) , P2(5,5) باستخدام DDA؟

M=5/5=1

$5 \geq x \geq 0$ بتزايد 1

$Y2 > y_1$ (تزايد) $\Rightarrow 5>0$

نستخدم $y_{i+1} = y_i + m$

x	$y_{i+1} = y_i + m$	point
0	0	(0,0)
1	y =0+1=1	(1,1)
2	y =1+1=2	(2,2)
3	y=2+1=3	(3,3)
4	y=3+1=4	(4,4)
5	y=4+1=5	(5,5)

H.W.- ارسم قطعة مستقيمة من النقطة (2 ، 4) إلى (9 ، 9) باستخدام خوارزمية DDA.

H.W.–ارسم قطعة مستقيمة تربط (20 ، 10) و (25 ، 14) باستخدام خط برزنهام.

المحاضرة 6 :خوارزميات رسم الدائرة

1) الطريقة العامة باستخدام معادلة الدائرة.
2) الاحداثيات القطبية.
3) طريقة برزنهام.
4) خوارزمية إنشاء دائرة نقطة المنتصف

1– الطريقة العامة لرسم الدائرة: رسم المركز (x_c , y_c) ورسم نصف القطر r.

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

ولرسم نقاط الدائرة يجب استخراج قيم y من المعادلة العامة السابقة:

$$(y-y_c)^2 = r^2 - (x-x_c)^2$$
$$y - y_c = \mp\sqrt{r^2 - (x-x_c)^2}$$
$$\therefore y = y_c \mp \sqrt{r^2 - (x-x_c)^2}$$

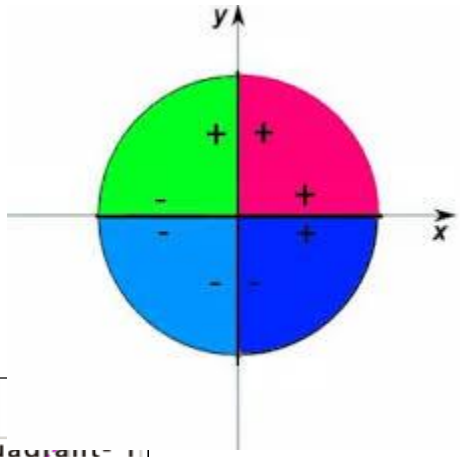
لهذه الطريقة مساوئ:

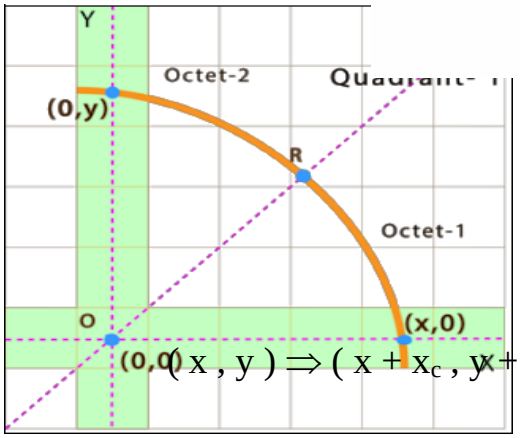
- لحساب قيم y في رسم كل نقطة يتطلب حسابات معقدة في المعادلة مثل الجذر التربيعي والضرب.
- تظهر نقاط المحيط متباعدة لذا يستخدم الاسلوب الثاني في رسم الدائرة.

2-الاحداثيات القطبية: أهملت هذه الطريقة بسبب الوقت المستغرق في حساب الدوال القطبية لرسم كل نقطة من نقاط الدائرة، عليه تم اعتماد الاسلوب الثالث.

3-خوارزمية برزنهام لرسم الدائرة:

ترسم الدائرة باستخدام احداثيات المركز (0,0) أو أحد ارباع الدائرة في الرسم. نفرض ان الدائرة ترسم من نقطة الاصل لتكون حالة عامة، لذا نقوم برسم نقاط ثمن الدائرة، ويعوض في باقي الاجزاء بالاعتماد على مبدأ التماثل، فتضاف x_c , y_c الى باقي الاثمان وكما يلي:





$$(x, y) \Rightarrow (x + x_c, y + y_c)$$

الربع الاول $(y , x) \Rightarrow (y + x_c , x + y_c)$

الربع الثاني $(-x , y) \Rightarrow (-x + x_c , y + y_c)$
$(-y , x) \Rightarrow (-y + x_c , x + y_c)$

الربع الثالث $(-x , -y) \Rightarrow (-x + x_c , -y + y_c)$
$(-y , -x) \Rightarrow (-y + x_c , -x + y_c)$

الربع الرابع $(x , -y) \Rightarrow (x + x_c , -y + y_c)$
$(y , -x) \Rightarrow (y + x_c , -x + y_c)$

الطريقـة:

$(x_1 , y_1) = (0 , r)$ اختيار اول نقطة للرسم وهي

$d_i = 3 - 2r$ حساب اول معامل وهو

if $d_i < 0$ then $\left.\begin{matrix} x_{i+1} = x_i + 1 \\ y_{i+1} = y_i \end{matrix}\right\} (x_i + 1, y_i) \Rightarrow (x++, y)$

$d_{i+1} = d_i + 4 * x_{i+1} + 6$ حساب المعامل الجديد

else $\left.\begin{matrix} x_{i+1} = x_i + 1 \\ y_{i+1} = y_i - 1 \end{matrix}\right\} (x_i + 1, y_i - 1) \Rightarrow (x++, y--)$

$d_{i+1} = d_i + 4 (x_{i+1} - y_{i+1}) + 10$ حساب المعامل الجديد

تكرر الخطوة الاخيرة في حسـاب معامل جديد ومقارنته مع الصـفر الى ان نصـل الى نقطة التوقف وهي أن $x_i \geq y_i$.

مثال: استخدم خوارزمية برزنهام لرسم الدائرة المعرفة بالمعادلة:

$$(x-10)^2 + (y+7)^2 = 16$$

الحل: $x_c = 10$, $y_c = -7$, $r = 4$ وسيكون رسم الدائرة في الربع الرابع:

$(x_1 , y_1) = (0 , r) = (0 , 4)$

$d_i = 3 - 2r = 3 - 2*4 = -5 < 0$

$(x++ , y) \Rightarrow (1,4)$

$d_{i+1} = d_i + 4*x + 6 = -5 + 4*1 + 6 = 5 > 0$

$(x++ , y--) \Rightarrow (2,3)$

$d_{i+1} = d_i + 4(x-y) + 10 = 5 + 4(2-3) + 10 = 11 > 0$

x	y
0	4
1	4
2	3
3	2

Stop

$(x{+}{+}\ ,\ y{-}{-}) \Rightarrow (3,2)$

stop because $x \geq y$ $\quad (3,2)$

هنا تم ايجاد الثمن النموذجي للبدء بالحل فنقوم بالاعتماد عليه وعلى مبدأ التماثل لايجاد الاثمان الكلية وكما يلي:

$x + x_c$	$y + y_c$	$y + x_c$	$x + y_c$	
0+10= 10	4-7= -3	4+10= 14	0-7= -7	الربع الاول
1+10= 11	4-7= -3	4+10= 14	1-7= -6	
2+10= 12	3-7= -4	3+10= 13	2-7= -5	
3+10= 13	2-7= -5	2+10= 12	3-7= -4	
$-x + x_c$	$y + y_c$	$-y + x_c$	$x + y_c$	
-0+10= 10	4-7= -3	-4+10= 6	0-7= -7	الربع الثاني
-1+10= 9	4-7= -3	-4+10= 6	1-7= -6	
-2+10= 8	3-7= -4	-3+10= 7	2-7= -5	
-3+10= 7	2-7= -5	-2+10= 8	3-7= -4	
$-x + x_c$	$-y + y_c$	$-y + x_c$	$-x + y_c$	
-0+10= 10	-4-7= -11	-4+10= 6	-0-7= -7	الربع الثالث
-1+10= 9	-4-7= -11	-4+10= 6	-1-7= -8	
-2+10= 8	-3-7= -10	-3+10= 7	-2-7= -9	
-3+10= 7	-2-7= -9	-2+10= 8	-3-7= -10	
$x + x_c$	$-y + y_c$	$y + x_c$	$-x + y_c$	
0+10= 10	-4-7= -11	4+10= 14	-0-7= -7	الربع الرابع
1+10= 11	-4-7= -11	4+10= 14	-1-7= -8	
2+10= 12	-3-7= -10	3+10= 13	-2-7= -9	
3+10= 13	-2-7= -9	2+10= 12	-3-7= -10	

H.W.: استخدم خوارزمية برزنهام لرسم الدائرة $(x-10)^2 + (y-7)^2 = 4$

H.W.: استخدم خوارزمية برزنهام لرسم الدائرة $x^2 + y^2 = 9$

البرنامج الفرعي لرسم دائرة باستخدام خوارزمية برزنهام:

```
void CIR ( int x , int y , int r )
{
  int x1 , y1 , d ;
  x1 = 0 ;
  y1 = r ;
```

```
  d = abs(3 – 2 * r) ;
  while ( x1 <= y1 ) {
            x1 ++ ;
            putpixel ( x1 + x , y1 + y , RED) ;
            putpixel ( -x1 + x , y1 + y , RED) ;
            putpixel ( x1 + x , -y1 + y , RED) ;
            putpixel ( -x1 + x , -y1 + y , RED) ;
            putpixel ( y1 + x , x1 + y , RED) ;
            putpixel ( -y1 + x , x1 + y , RED) ;
            putpixel ( y1 + x , -x1 + y , RED) ;
            putpixel ( -y1 + x , -x1 + y , RED) ;
            if ( d < 0 )  d = d + 4 * x1 + 6 ;
                  else { d = d + 4 * (x1 – y1) + 10 ;
                           y1 -- ;
                         }
              }
}
```

Basic Structure of a C-graphics program:

```
#include<stdio.h>
#include<graphics.h>//must be included for every graphics program
#include<dos.h> //for including delay function.

void main()

{

int gd=DETECT,gm; //gd=detects best available graphics driver, gm =graphics mode.

initgraph(&gd,&gm,"C:\\TurboC3\\BGI");// for initializing graph mode

// above 2 steps are must for every graphics program.

//declaration of any variables must be done before calling  initgraph() function.

// next write code for producing requiring design or drawing object line(100,100,200,200);//draws a line

segment.

getch();
}
```

Basic Structure of a C-graphics program:

```
#include<stdio.h>
#include<graphics.h>//must be included for every graphics program
#include<conio.h>
#include<dos.h> //for including delay function.
void main()
{
int gd=DETECT,gm; //gd=detects best available graphics driver, gm =graphics mode.
initgraph(&gd,&gm,"C:\\TurboC3\\BGI");// for initializing graph mode
// above 2 steps are must for every graphics program.
//declaration of any variables must be done before calling  initgraph() function.
// next write code for producing requiring design or drawing object
line(100,100,200,200);//draws a line segment.
getch();
}
```

Computer Graphics

Experiment – 1

Digital Differential Analyzer Algorithm

1 : Digital differential analyzer (DDA) is used for linear interpolation of variables over an interval between given start, end points and for rasterization of lines, triangles and polygons. Using DDA Algorithm, Write a C-Program to draw a line segment between two given points?

Aim: To implement DDA Algorithm for drawing a line segment between two given end points A (x1, y1) and B(x2, y2).

Description: DDA algorithm is an incremental scan conversion method. Here we perform calculations at each step using the results from the preceding step. The characteristic of the DDA algorithm is to take unit steps along one coordinate and compute the corresponding values along the other coordinate. The unit steps are always along the coordinate of greatest change, e.g. if dx = 10 and dy = 5, then we would take unit steps along x and compute the steps along y.

In DDA we need to consider two cases;

One is slope of the line less than or equal to one (|m| ≤1) and slope of the line greater than one (m| > 1).

1) When **|m| ≤ 1** means y2-y1 = x2-x1 or y2-y1 <x2-x1.In both these cases we assume x to be the major axis. Therefore we sample x axis at unit intervals and find the y values corresponding to each x value. We have the slope equation as

$$\Delta y = m \Delta x$$

$$y2-y1 = m (x2-x1)$$

In general terms we can say that $y_{i+1} - y_i = m(x_{i+1} - x_i)$. But here $\Delta x = 1$; therefore the equation reduces to $y_{i+1} = y_i + m = y_i + dy/dx$.

2) When **| m| > 1** means y2-y1> x2-x1 and therefore we assume y to be the major axis. Here we sample y axis at unit intervals and find the x values corresponding to each y value. We have the slope equation as

$$\Delta y = m \Delta x$$

$$y2-y1 = m (x2-x1)$$

Algorithm:

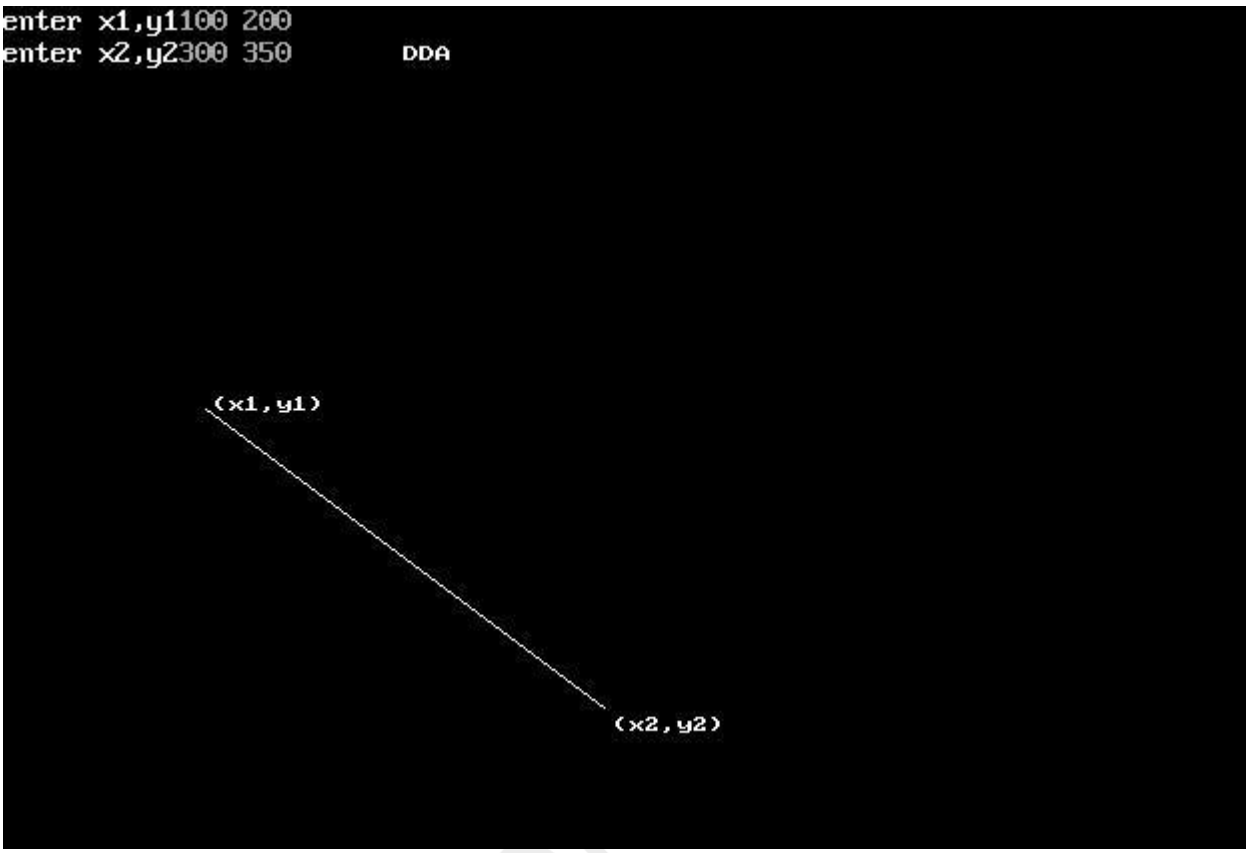
1. Start.

2. Declare variables x,y,x1,y1,x2,y2,k,dx,dy,s,xi,yi and also
   declare gdriver=DETECT, mode.
3. Initialize the graphic mode with the path location in TurboC3 folder.
4. Input the two line end-points and store the left end-points in (x1,y1).
5. Load (x1, y1) into the frame buffer; that is, plot the first point. put x=x1,y=y1.
6. Calculate dx=x2-x1 and dy=y2-y1.
7. If abs (dx) > abs (dy), do s=abs(dx).
8. Otherwise s= abs(dy).
9. Then xi=dx/s and yi=dy/s.
10. Start from k=0 and continuing till k<s,the points will be
    i. x=x+xi.
    ii. Y=y+yi.
11. Plot pixels using putpixel at points (x,y) in specified colour.
12. Close Graph and stop.

Program:

```
#include<stdio.h>
#include<graphics.h
> #include<math.h>
float round(float a);
void main()
{
        int gd=DETECT,gm;
        // gd=graphics driver (detects best graphics driver and assigns it as default, gm=graphics
mode
.       int x1,y1,x2,y2,steps,k;
        float
        xincr,yincr,x,y,dx,dy;
        printf("enter x1,y1");
        scanf("%d%d",&x1,&y
        1); printf("enter
        x2,y2");
        scanf("%d%d",&x2,&y
        2);
        initgraph(&gd,&gm,"c:\\turboc3\\BGI");//initializes the
        graph dx=x2-x1;
        dy=y2-y1;
        if(abs(dx)>abs(dy))
                steps=abs(dx);
        else
                steps=abs(dy);
        xincr=dx/step
        s;
        yincr=dy/step
        s; x=x1;
        y=y1;
        for(k=1;k<=steps;k++)
        {
                delay(100);//for seeing the line drawing process slowly.
                x+=xincr;
                y+=yincr;
                putpixel(round(x),round(y),WHITE);
        }
        outtextxy(200,20,"DDA"); // for printing text at desired screen
        location. outtextxy(x1+5,y1-5,"(x1,y1)");
        outtextxy(x2+5,y2+5,"(x2,y2
        )"); getch();
        closegraph();   // closes the graph and comes back to previous graphic mode.
}
float round(float a)
{
        int
        b=a+0.5;
        return b;
}
```

Output:



Computer

Experiment 2

Bresenham's Line Drawing Algorithm

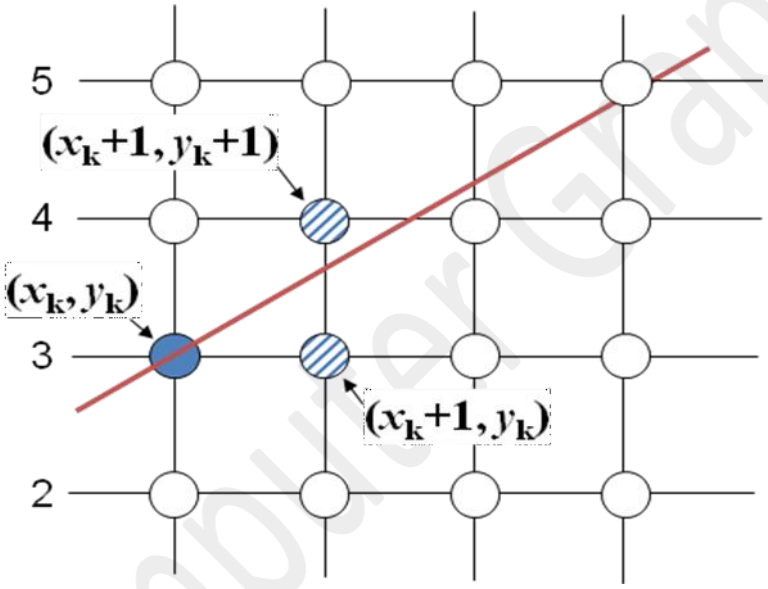
2: Bresenham's line algorithm is an algorithm which determines which order to form a close approximation to a straight line between two given points. Write a C program for determining pixel activation list between two given points in order to draw line segment using bresenham's Line drawing algorithm?

Aim: To implement Bresenham's line drawing algorithm for drawing a line segment between two given endpoints A (x1, y2) and B(x2, y2).

Description:

Basic Concept:

- Move across the *x* axis in unit intervals and at each step choose between two different *y* coordinates



- For example, from position (2, 3) we have to choose between (3, 3) and (3, 4). We would like the point that is closer to the original line
- So we have to take decision to choose next point. So next pixels are selected based on the value of decision parameter p. The equations are given in below algorithm.

Computer Graphics

Algorithm:

BRESENHAM'S LINE DRAWING ALGORITHM

1. Input the two line end-points, storing the left end-point in (x_0, y_0)
2. Plot the point (x_0, y_0)
3. Calculate the constants Δx, Δy, $2\Delta y$, and $(2\Delta y - 2\Delta x)$ and get the first value for the decision parameter as:

$$p_0 = 2\Delta y - \Delta x$$

4. At each x_k along the line, starting at $k = 0$, perform the following test. If $p_k < 0$, the next point to plot is (x_k+1, y_k) and:

$$p_{k+1} = p_k + 2\Delta y$$

Otherwise, the next point to plot is (x_k+1, y_k+1) and:

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$

5. Repeat step 4 $(\Delta x - 1)$ times

NOTE: The algorithm and derivation above assumes slopes are less than 1. For other slopes we need to adjust the algorithm slightly

Computer Graphics

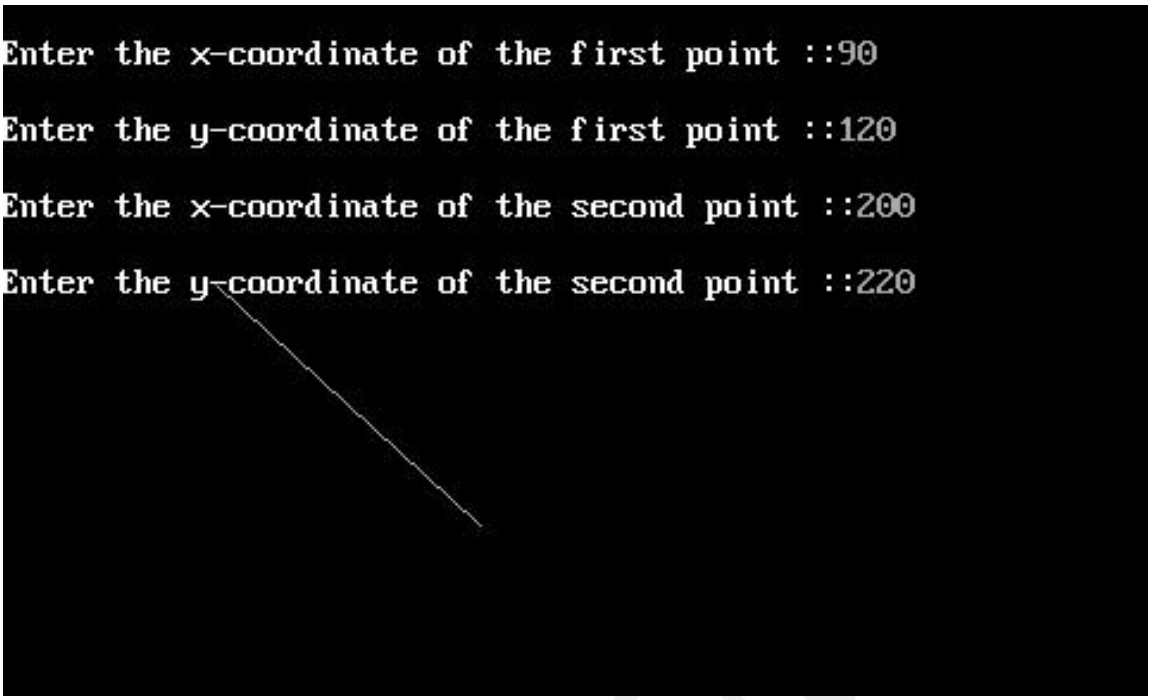
Program:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
        int
        x,y,x1,y1,x2,y2,p,dx,dy;
        int gd=DETECT,gm;
        initgraph(&gd,&gm,"C:\\TurboC3\\BGI");
        printf("\nEnter the x-coordinate of the first point ::");
        scanf("%d",&x1);
        printf("\nEnter the y-coordinate of the first point ::");
        scanf("%d",&y1);
        printf("\nEnter the x-coordinate of the second point ::");
        scanf("%d",&x2);
        printf("\nEnter the y-coordinate of the second point ::");
        scanf("%d",&y2);
        x=x1;
        y=y1;
        dx=x2-
        x1;
        dy=y2-
        y1;
        putpixel(x,y,2);
        p=(2*dy-dx);
        while(x<=x2)
        {
          if(p<0
          )
          {       x=x+1;
                  p=p+2*dy
                  ;
          }
          else
          {

                  x=x+1
                  ;
                  y=y+1
                  ;
                  p=p+(2*dy)-(2*dx);
           }
          putpixel(x,y,7);
        }
        getch();
        closegraph();
}
```

Computer Graphics

Output:



Computer Gr

Experiment- 5

Creating various types of texts and fonts

5: Using different graphics functions available for text formatting in C-Language, Write a C program for displaying text in different sizes, different colors, different font styles?

Aim: To write a C-program for displaying text in different sizes, different colors and different font styles by using graphics functions.

Description:

The following graphics functions are available for text formatting in C.

1. **Outtext()**------- for displaying text on output screen
2. **Outtextxy()**---- for displaying text on output screen at location (x,y).
3. **Settextstyle() --** used for specifying font style, font display direction(vertical or horizontal), font size.
4. **Setcolor() ------** setcolor function is used to change the current drawing color.e.g. setcolor(RED) or setcolor(4) changes the current drawing color to RED. Remember that default drawing color is WHITE.

Outtext: outtext function displays text at current position.
Eg: outtext("To display text at a particular position on the screen use outtextxy");

Outtextxy(): outtextxy function display text or string at a specified point(x,y) on the screen.
Eg: outtextxy(100, 100, "Outtextxy function");--displays the message "Outtextxy function"
At screen location of (100, 100).

Settextstyle(): Settextstyle function is used to change the way in which text appears, using it we can modify the size of text, change direction of text and change the font of text.
Eg: settextstyle(font,direction,charsize);--settextstyle(TRIPLEX_FONT,HORIZ_DIR,2);
Font –font style- it may be font name or integer value from 0- 9.

Different fonts are:

DEFAULT_FONT,
TRIPLEX_FONT,
SMALL_FONT,
SANS_SERIF_FONT,
GOTHIC_FONT,
SCRIPT_FONT,
SIMPLEX_FONT,
TRIPLEX_SCR_FONT,
COMPLEX_FONT,
EUROPEAN_FONT,
BOLD_FONT

Directions are two.

1. Horizontal direction(HORIZ_DIR or 0)
2. Vertical direction(VERT_DIR or 1)

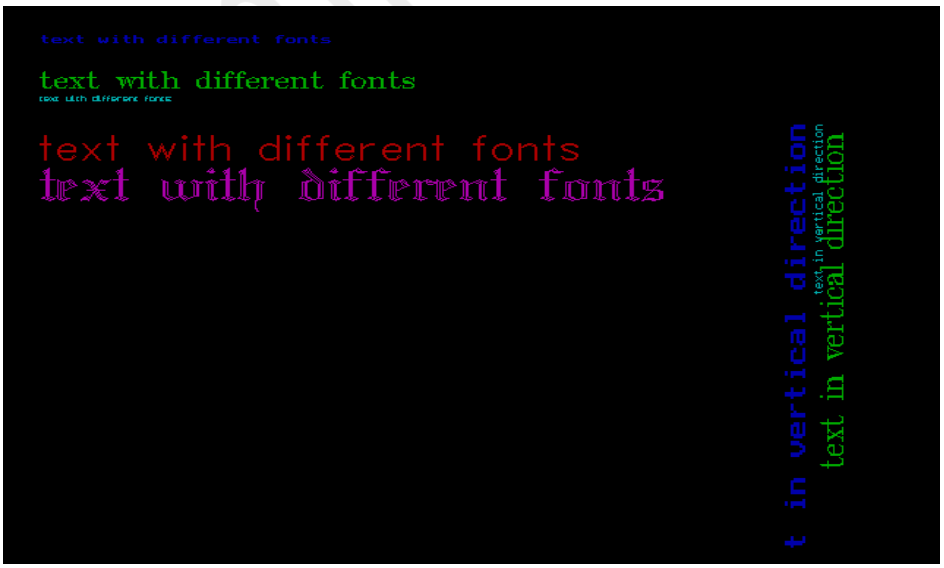
Character size(0-9)

Computer Graphics

Program:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
        int gd=DETECT,gm,x=25,y=25,font=10;
        initgraph(&gd,&gm,"C:\\turboC3\\BGI");
        for(font=0;font<=4;font++)
        {
            settextstyle(font,HORIZ_DIR,font+1);// sets font type, font direction, size
            setcolor(font+1); // sets color for text.
            outtextxy(x,y,"text with different fonts"); // prints message on screen at (x,y)
            y=y+25;
        }
        for(font=0;font<=2;font++)
        {
            settextstyle(font,VERT_DIR,font+2
            ); setcolor(font+1);
            x=250
            ;
            y=100
            ;
            outtextxy(x,y,"text in vertical direction");
            y=y+25;
        }
getch();
closegraph();
}
```

Output:



Computer Graphics

Experiment- 6

Creating two dimensional objects

6: Using certain graphic functions available in C-language for drawing lines, rectangles & circles, Write a C-Program which generates pixel activation list for drawing the following simple two dimensional objects (Circle, Ellipse…..).

i) House ii) Car iii) Fish iv) man?

Aim: To write a C-program for creating simple two dimensional shape of house, car, fish, man using lines, circles etc.

Description:

The following graphics functions are available for creating two dimensional shapes in C.

line
circle
ellipse
rectangle
drawpoly

line: line function is used to draw a line from a point(x1,y1) to point(x2,y2) i.e. (x1,y1) and (x2,y2) are end points of the line.
Declaration: - line(x1, y1, x2, y2); ---line (100,200,300,400);

Circle: Circle function is used to draw a circle with center (x, y) and third parameter specifies the radius of the circle.
Declaration: circle(x, y, r)—circle (100, 200, 25) ;(25 is radius of circle, (100,100) is center of circle).

Ellipse: Ellipse is used to draw an ellipse (x, y) are coordinates of center of the ellipse, startangle is the starting angle, endangle is the ending angle, and fifth and sixth parameters specifies the X and Y radius of the ellipse. To draw a complete ellipse strangles and end angle should be 0 and 360 respectively.
Usage: ellipse(x, y, startangle, endangle, xradius, yradius);--ellipse(100,200,0,360,25,45);((100,200) is center of ellipse, 0 is start angle, 360 is end angle, 25 is x-axis radius, 45 is radius circle).

Rectangle: rectangle function is used to draw a rectangle. Coordinates of left top and right bottom corner are required to draw the rectangle. left specifies the X-coordinate of top left corner, top specifies the Y-coordinate of top left corner, right specifies the X-coordinate of right bottom corner, bottom specifies the Y-coordinate of right bottom corner.

Syntax: rectangle(left,top,right,bottom);--rectangle(100,200,300,400);

Computer Graphics

Drawpoly: Drawpoly function is used to draw polygons i.e. triangle, rectangle, pentagon, hexagon etc.

Syntax: drawpoly(num,points);--num indicates number of vertices of polygon. Num = num+1.

Example: we will draw a triangle using drawpoly, consider for example the array :-
int points[] = { 320, 150, 420, 300, 250, 300, 320, 150};

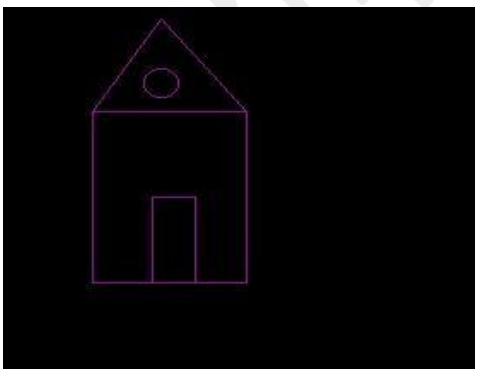
points array contains coordinates of triangle which are (320, 150), (420, 300) and (250, 300). Note that last point(320, 150) in array is same as first.

Number of vertices are denoted by num. for any polygon, number of vertices are (num+1). For triangle, number of vertices are 4.

i) Program for creating House shape:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
        int                             gd=DETECT,gm;
        initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
        setcolor(5);
        rectangle(60,80,150,200);
        rectangle(95,140,120,200);
        line(60,80,100,15);
        line(100,15,150,80);
        circle(100,60,10);
        getch();
        closegraph();
}
```

Output:

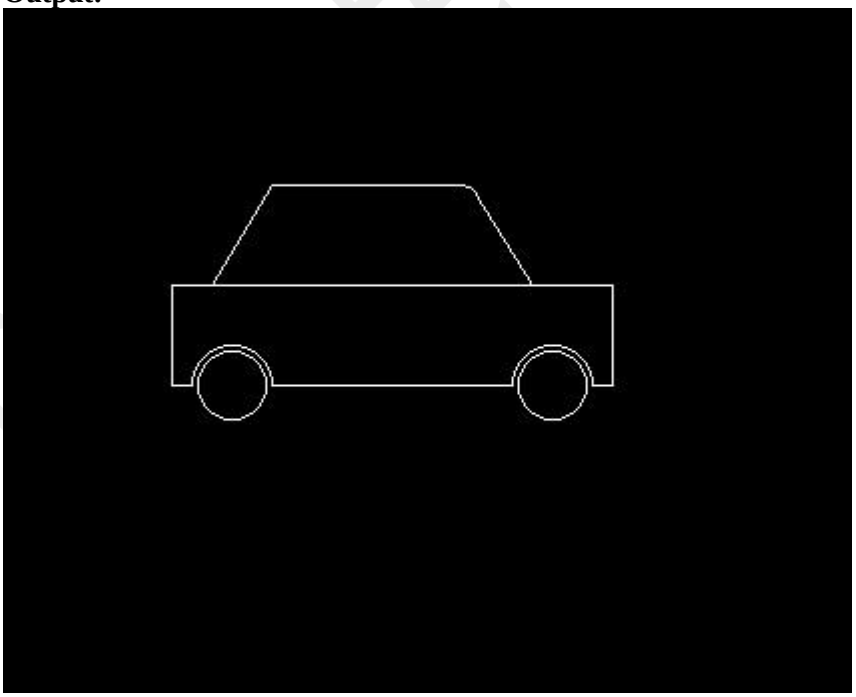


Computer Graphics

ii) Program for creating simple car shape:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>
void main()
{
   int gd = DETECT, gm;
   initgraph(&gd,  &gm,  "C:\\TurboC3\\BGI");
   cleardevice();
   line( 150, 100,  242, 100);
   ellipse(242, 105, 0, 90, 10, 5);
   line(150, 100,  120, 150);
   line(252, 105,  280, 150);
   line(100, 150,  320, 150);
   line(100, 150,  100, 200);
   line(320, 150,  320, 200);
   line(100, 200,  110, 200);
   line( 320, 200,   310, 200);
   arc(130, 200, 0, 180, 20);
   arc( 290, 200, 0, 180, 20);
   line( 270, 200,  150, 200);
   circle(130, 200, 17);
   circle(290,   200,   17);
   getch();
}
```

Output:



Computer Graphics

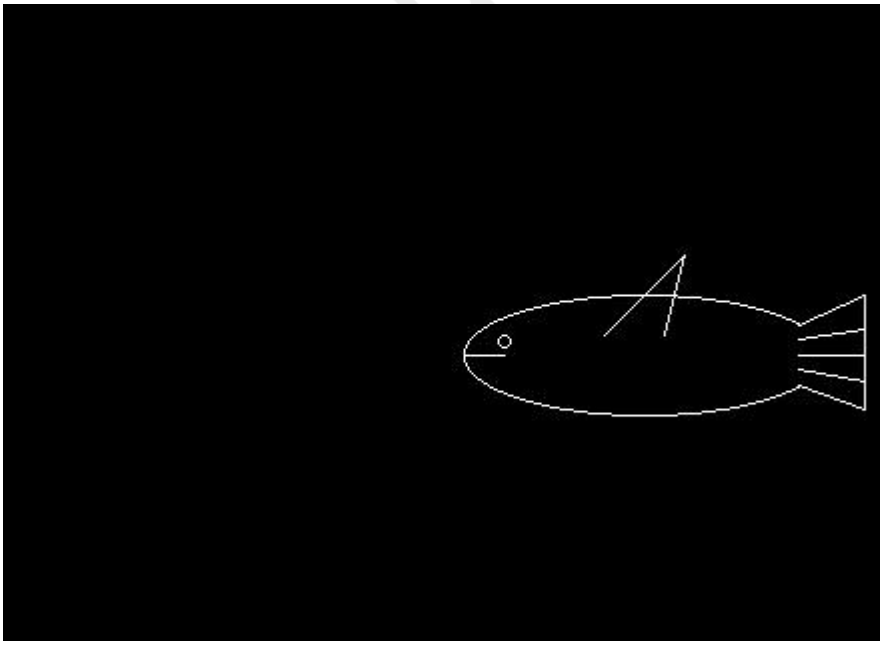
iii) Program for creating fish:

```
#include<stdlib.h>
#include<conio.h>
#include<dos.h>
#include<graphics.h>
#include<ctype.h>

void main()
{
        int gd=DETECT,gm;
        initgraph(&gd,&gm,"C:\\TurboC3\\BGI");
        cleardevice();
        ellipse(520,200,30,330,90,30);
      circle(450,193,3);
      line(430,200,450,200);
      line(597,185,630,170);
      line(597,215,630,227);
      line(630,170,630,227);
      line(597,200,630,200);
      line(597,192,630,187);
      line(597,207,630,213);
      line(500,190,540,150);
      line(530,190,540,150);

    getch();
}
```

Output:



Computer Graphics