بيانات حاسوبية

المحاضرة السادسة

د. غيث ابراهيم بلال

Attributes of Graphics primitives

الصفات المميزة للرسومات الأولية

• عرض الخط line width:

ان كل ما سبق مناقشته كان يتم الرسم فيه بخطوط عرضها pixel واحد

لنناقش الآن الطرق التي يتم من خلالها الرسم بسماكة معينة.

1. مضاعفة البكسلات Replicating pixel:

يمكننا تمديد الخوارزميات (المعطاة في المحاضرات السابقة) ببساطة لرسم عدة خطوط متجاورة بجانب بعضها البعض بدلاً من رسم خط منفرد.

(أي الخوارزميات التي أخذناها سابقاً نستطيع بسهولة إعادة تطبيقها من أجل شروط جديدة لتنتج لدينا رسومات جديدة بسماكة معينة).

قاعدة:

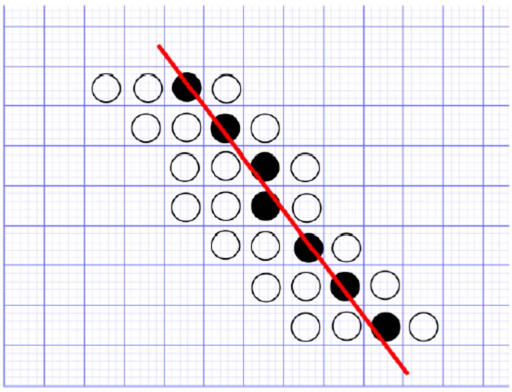
يمكننا إعادة الرسم او (التكرار) على الاعمدة او على الاسطر.

ويتم التكرار على الاعمدة عندما يكون ميل المستقيم محصور بين 1 و 1- بينما يتم على الاسطر في الحالات الأخرى.

أي نستطيع مقارنة Δx و Δy لكل من نهايتي المستقيم.

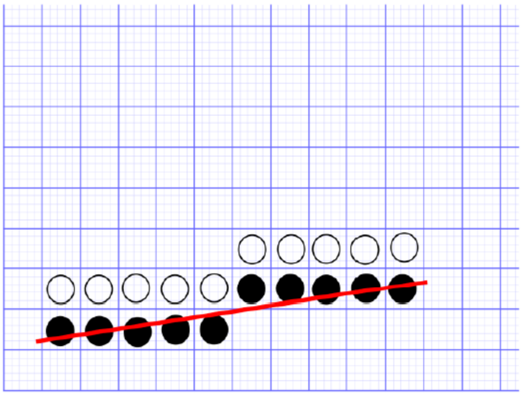
إذا كان $|\Delta x| \geq |\Delta y|$ فإن الرسم (او التمديد او التكرار) يتم على الاعمدة وعكس ذلك التمديد يكون على الأسطر (الصفوف).

أمثلة:



ميل المستقيم $|m|>1$

يمدد بشكل افقي



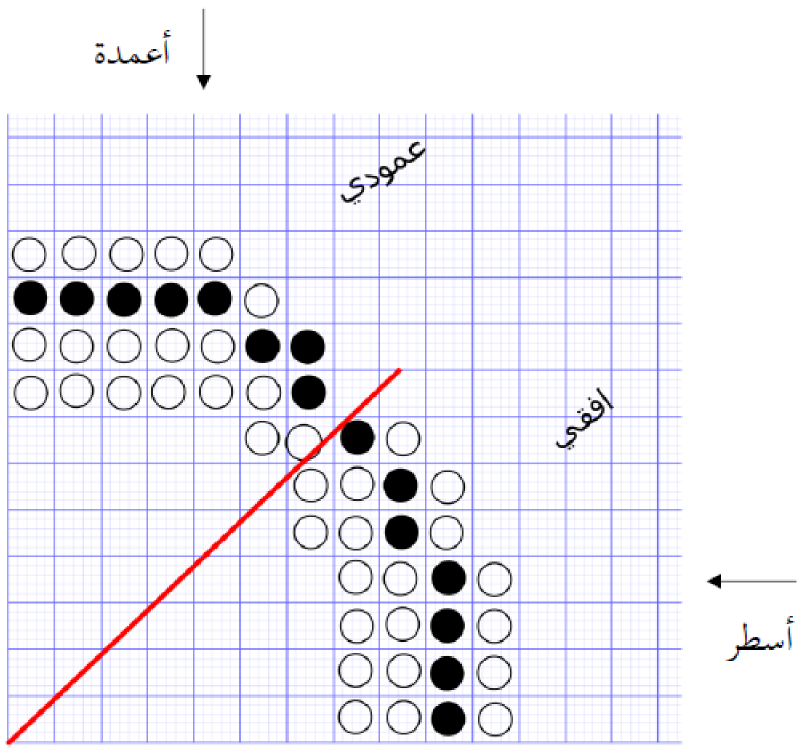
ميل المستقيم $|m|\leq 1$

يمدد بشكل عمودي

تطبق الطريقة السابقة على الخط المنحني: (او قوس من الدائرة) وذلك لأجل الحصول على الخط المنحني وبسماكات متنوعة و يتم عن الطريق التكرار او التمديد افقياً او عمودياً.

فإذا كان مقدار الميل اقل او يساوي الواحد نرسم او نكرر الرسم على الخطوة (spans) العمودية وإذا كان الميل أكبر من الواحد فالرسم يكون افقي ومنه نستنتج ان التكرار عمودي في الثمن الثاني من الدائرة وافقي في الثمن الأول.

مثلاً: السماكة تساوي 4، من $x = 0$ الى $y = x$ نستخدم التكرار العمودي ، ثم نستخدم الامتداد او التحديد الافقي في الثمن الثاني.



هذه الطريقة تضم القطوع والدوائر وباقي المنحنيات.

2. The moving pen:

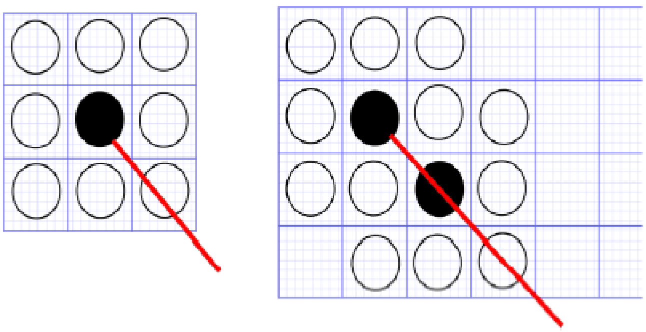
هنا نعتمد قلم بشكل مربع يتحرك على طول الخط الأصلي المرسوم بسماكة pixel واحد حيث تعمل هذه الطريقة بشكل معقول مع الخطوط المستقيمة.

اذ يحدد قناع البكسلات الذي يخزن كمصفوفة لمواقع البكسلات التي تتوضع على طول المسار المستقيم.

حيث يوضح الشكل التالي قلم مربع الشكل:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

ويمكن ان نحصل على القناع التالي عن طريق تحريك المركز (او زاوية واحدة) للقناع على طول المسار.



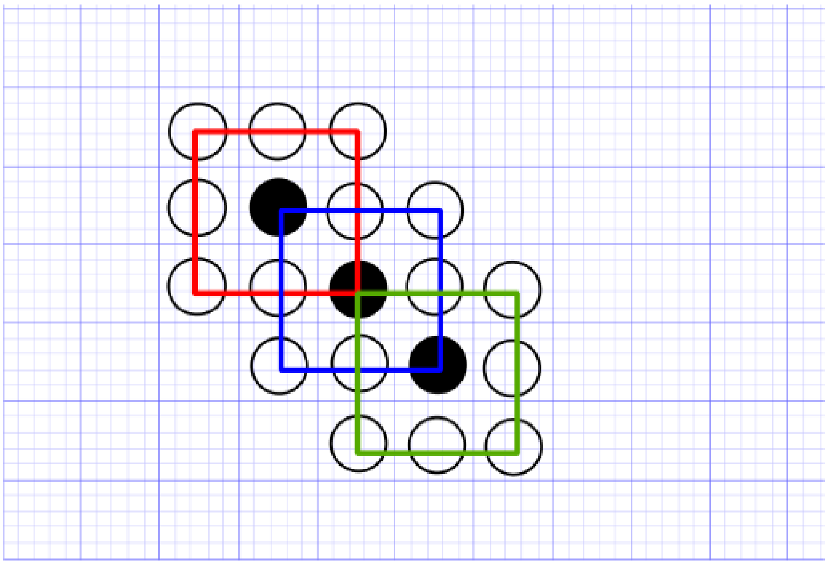
ملاحظة 1: لتجنب توضع البكسلات أكثر من مرة في ذاكرة Buffer نستطيع ببساطة (تكديس) المسافات الافقية المتولدة عند كل موقع من القناع مع المحافظة على كل x لموقع البداية والنهاية لكل نقطة من نقاط المستقيم الممسوح المتقاطعة مع خط الشبكة.

ملاحظة 2: الخطوط المنتجة بهذه الطريقة يمكن ان تعرض بسماكات مختلفة بتغيير حجم القناع (mask) او المصفوفة.

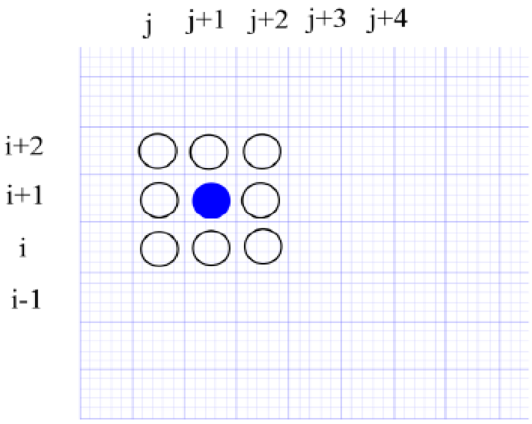
مثلاً: خط القلم السابق يمكن ان يرفع بتحويل المصفوفة الى 2×2 او تزداد سماكته بوضعها 4×4

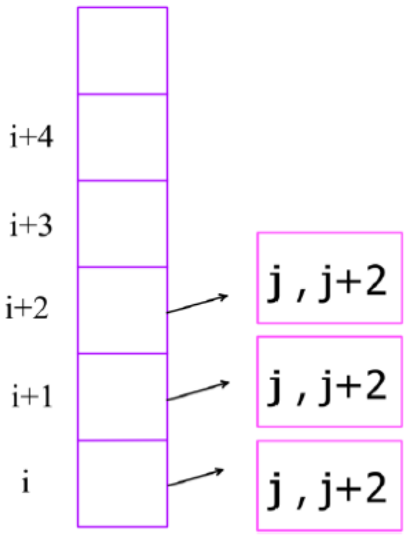
تطبق على المستقيمات والمنحنيات بنفس الأسلوب السابق

ملاحظة : تتحرك بكسل بكسل مع اخذ القناع على طول المسار مع تحديد سماكة الخط.

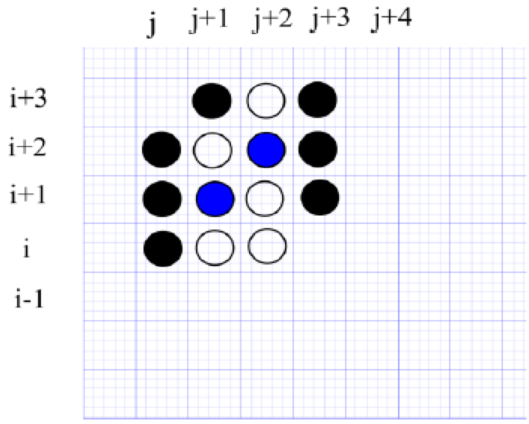


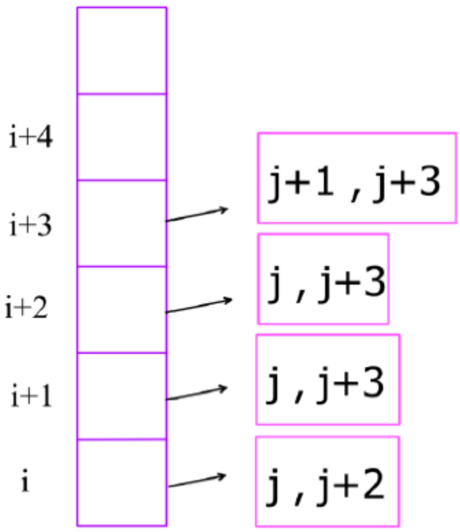
الحركة 1





الحركة 2





3. Filling Areas Between Boundaries:

هنا نقوم برسم خطين متجاورين (متوازيين) للخط الأصلي يبعدان عنه بمقدار نصف السماكة المطلوبة ثم تملأ المسافة بين هذين الخطين بالعرض المطلوب او السماكة المطلوبة.

أو يمكننا اعتبار الخط الأصلي هو الحد الخارجي ورسم حد اخر داخلي يبعد عنه بمقدار السماكة المطلوبة ثم نقوم بملء هذه المساحة مما يسمح بمعالجة مشكلة السماكة الزوجية ولكن ذلك يجعل الرسم مزاح قليلاً (تقريباً نصف السماكة) تطبق على جميع الخطوط المستقيمة والمنحنية (بما فيها الدائرية والقطوع).

4. Line style:

ان شكل الخط يتضمن الخط المستمر أو المصمت أو غير المنقطع.

(معروف رسمه خلال المحاضرات السابقة)

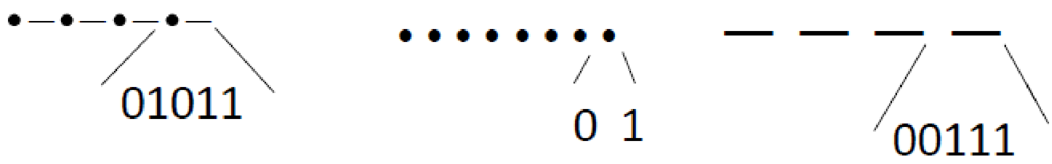
الخط المتقطع او المنقط يمكن ان نعدل الخوارزمية لرسم المستقيم لأجل الحصول على هكذا خطوط بتحديد طول وتباعد الأقسام المستمرة على طول الخط . حيث يوصف الطول والفراغات الداخلية بواسطة قناع البكسل وله نموذج لمصفوفة اعداد ثابتة.

مثال: 11111000: يمثل خط متقطع بمستقيم طوله 5 بكسلات وثلاث فراغات داخلية.

يمكن ان تكون البكسلات التي تحمل الرقم 1 ملونة باي لون واضح والفراغات الداخلية تكون بلون الخلفية او الأرضية

اذن يعبر العدد 1 عن مكان وجود خط بينما يعبر الصفر عن مكان انقطاع الخط.

أمثلة:



سيئات هذه الطرق:

1. تسبب الخوارزمية الأولى (مضاعفة البكسلات) فجوات واضحة عند تلاقي المستقيمات مع بعضها البعض بزوايا معينة كما أنها تسبب وجود بعض البكسلات الناقصة عند عملية الانتقال من التمديد (التكرار) على الاعمدة الى التمديد (التكرار) على الاسطر.

2. ان الخطوط العمودية والافقية تتمتع بعرض مغاير للخطوط ذات الزوايا حيث تكون الخطوط الافقية والعمودية أكثر سماكة من الخطوط المائلة.

3. إضافة إلى مشكلة كون سماكة الخط زوجية مما سيجبرنا على اختيار احدى الجهات من الخط المرسوم لتكون أكثر سماكة من الجهة الأخرى.

أما الطريقة الثانية : حركة القلم the moving pen:

- تعطي سماكة أكبر عند البداية والنهاية للخط.
- الخطوط الافقية والعمودية في هذه الطريقة اقل سماكة من الخطوط المائلة (عكس الحالة الأولى).

Filling: ملئ المنطقة او التعبئة:

ناقشنا سابقاً آلية رسم الاشكال الأساسية ولكن ماذا لو أردنا تعبئة هذه الاشكال بألوان معينة؟

تعبئة المنطقة: هي عملية تلوين مساحة او منطقة من صورة محددة.

حيث توصف المنطقة بدلالة بكسلات حدودها او بكامل البكسلات التي تؤلف المنطقة.

وبالتالي لدينا الطرق التالية لتعبئة الاشكال:

1. خوارزمية تعبئة الحدود أو ملئ الحدود boundary:

هي خوارزمية تكرارية تبدأ ببكسل أولي يسمى البذرة داخل المنطقة وتتحقق الخوارزمية فيما إذا كان هذا البكسل هو بكسل حدود أو بكسل تمت تعبئته فإذا كان الجواب لا، يملأ البكسل ويكرر نفس العمل بالنسبة إلى أي بكسل مجاور كبذرة جديدة.

تعمل هذه الخوارزمية بشكل جيد على المناطق ذات الشكل الاعتباطي بملاحقة وملء البكسلات التي ليست على الحدود والموصولة بالبذرة وهذا يتطلب وقتاً وذاكرة لكي ينفذ وذلك بسبب وجود استدعاءات متكررة كثيرة خصوصاً إذا كان حجم المنطقة كبيراً جداً.

ويمكن ادخال بعض التغيرات للحد من عدد الاستدعاءات المتكررة.

مثلاً:

يمكننا أولاً ملئ البكسلات التي على يسار وعلى يمين البذرة على نفس خط المسح إلى أن نشمل جميع بكسلات الحدود ثم ننتقل إلى الخط الثاني أو المستقيم الافقي الثاني.

حيث تكتب الخوارزمية بالشكل الاجرائي التالي:

- ندخل نقطة (البذرة) مع اللون المراد تعبئته ولون الحد.
- إذا كانت النقطة غير ملونة باللون المطلوب و ليست من لون الحد تملأ النقطة.
- ثم نستدعي لأجل نقطة جديدة $(x+1,y)$ مع نفس الألوان للحد والمنطقة، وأيضاً نستدعي من أجل النقاط:

$$(x-1,y) \quad (x,y+1) \quad (x,y-1)$$

```
BoundaryFill(int x,int y, fill_color, boundary_color){
 int color;
 getPixel(x,y,color)
 if(color != boundary_color && color != fill_color){
   setPixel(x,y,fill_color);
   BoundaryFill(x+1,y,fill_color,boundary_color);
   BoundaryFill(x-1,y,fill_color,boundary_color);
   BoundaryFill(x,y+1,fill_color,boundary_color);
   BoundaryFill(x,y-1,fill_color,boundary_color);   } }
```

2.خوارزمية Flood fill:

تبدأ أيضاً هذه الخوارزمية ببذرة داخل المنطقة وتتحقق فيما إذا كان للبكسل لون المنطقة الأصلي.

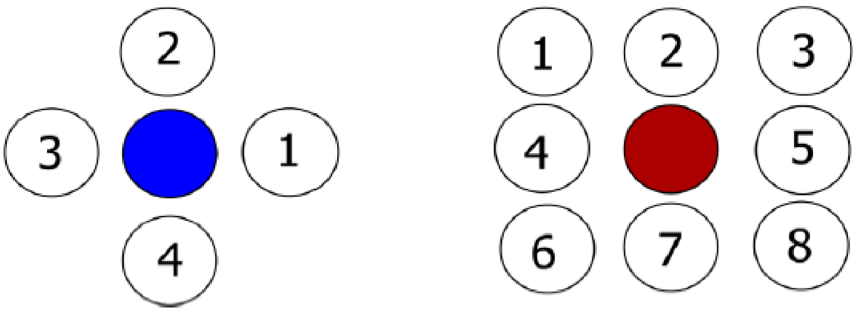
إذا كان الجواب نعم ، يملأ البكسل الجديد.

ثم نستخدم كل جيران البكسل كبذرة جديدة في استدعاء متكرر ويمكن تطبيق نفس التحسين السابق على هذه الخوارزمية ، وكذلك هي فعالة مع المناطق غير النظامية.

توضيح:

حيث نقوم بتلوين نقطة ما من داخل الشكل المطلوب تعبئته ثم الانتقال إلى النقاط الأربعة المجاورة لها أو النقاط الثمانية المجاورة لها وتعبئتهم.

ملاحظة: تظهر لدينا بعض المشاكل عندما نستخدم الوصل الرباعي لذلك نعتمد على الوصل الثماني.

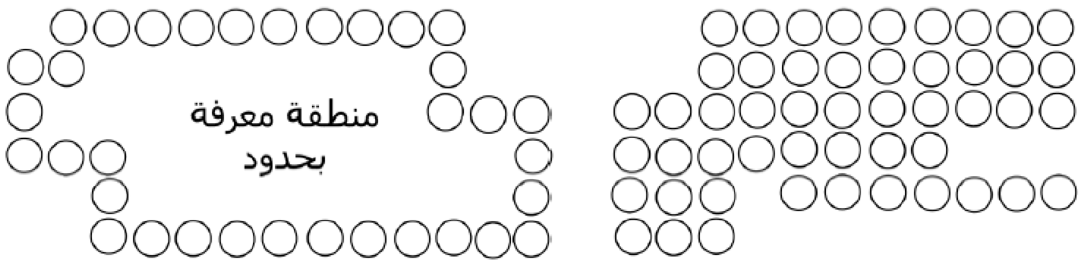


حيث تكتب الخوارزمية بالشكل الاجرائي التالي:

- ندخل نقطة (بكسل) مع لون التعبئة العام.
- إذا كان البكسل باللون العام يملأ.
- ثم نستدعي لأجل البكسلات مع تكرار نفس العمل:

$$(x+1,y)\ (x-1,y)\ (x,y+1)\ (x,y-1)$$

الشكل التالي يوضح ذلك:



Fill boundary

منطقة معرفة داخلياً
نستخدم Flood

```
FloodFill(int x, y, fill_color, Original_color){
 int color;
 getPixel(x,y,color)
 if(color == Original_color){
   setPixel(x,y,fill_color);
   FloodFill(x+1,y,fill_color,Original_color);
   FloodFill(x-1,y,fill_color,Original_color);
   FloodFill(x,y+1,fill_color,Original_color);
   FloodFill(x,y-1,fill_color,Original_color);
 }
}
```