

پر کردن و برش اشکال

مباحث این فصل:

❖ پر کردن اشکال

❖ رسم اشکال با قطر بیش از یک پیکسل

Replaceing pixel •

Moving Pen •

Filling Area Between Bound Areas •

Approximately By Tick Poly Line •

❖ برش اشکال

Cohen-suterland • الگوریتم

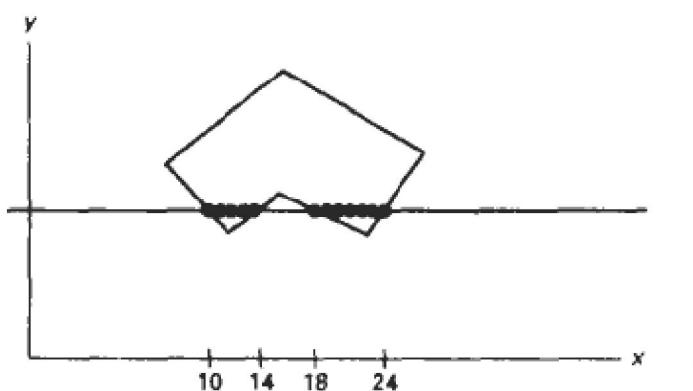
Cyrus-Beck • الگوریتم

پر کردن اشکال:

دو رهیافت اساسی برای پر کردن نواحی در سیستم های راست و وجود دارد. روش اول بر پایه اسکن خطوط و روش دوم بر پایه معادلات ریاضی میباشد. در این بخش ما به توضیح روش اول میپردازیم.

الگوریتم پر کردن چندضلعی نامنظم با استفاده از روش اسکن خطوط:

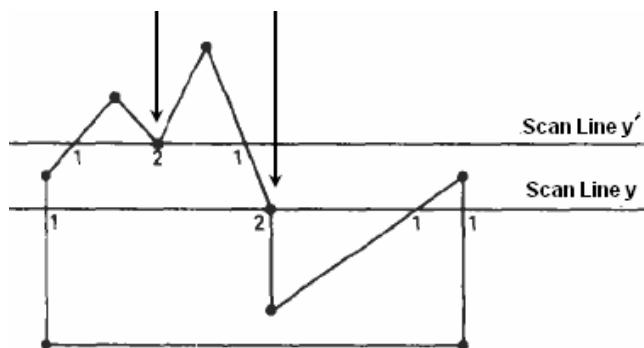
شکل زیر نحوه عملکرد این الگوریتم را برای رنگامیزی یک چندضلعی نامنظم نشان میدهد.



برای هر خط اسکن شده که از چندضلعی عبور کرده است، الگوریتم نقاط تقاطع خط اسکن شده با چندضلعی را بدست می آورد. سپس این نقاط تقاطع از چپ به راست مرتب میشوند و در نهایت مکانهای بین هر جفت از این نقاط در میانگیر قاب با رنگ مشخص مقاداردهی میشوند.

در همین مثال قبلی چهار نقطه تقاطع توسط الگوریتم تعیین میشوند و پیکسلهای داخلی از $x=10$ تا $x=14$ و از $x=18$ تا $x=24$ در میانگیر قاب با رنگ مشخص مقاداردهی میشوند.

اما دقت کنید که اگر نقاط تقاطع خط اسکن با چندضلعی، رئوس چندضلعی باشد در این صورت نیاز به یک سری دستکاری نقاط تقاطع داریم. برای مثال شکل زیر را در نظر بگیرید. در این صورت چگونه میتوان تابعه داخل چندضلعی را تشخیص داد.



در اینجا خط اسکن y هیچ مشکلی همراه ندارد اما خط اسکن y' در نقطه 2 دارای مشکل است. برای حل مشکل میتوان بدینگونه عمل کرد: برای هر نقطه تقاطع، قبل و بعد نقطه را نگاه میکنیم. اگر به طور یکنواخت نزولی یا صعودی

بود که نقطه را به عنوان نقطه تقاطع در نظر میگیریم و در غیراین صورت نقطه را به صورت دو نقطه تقاطع در یک مکان در نظر میگیریم.

برای پیاده سازی این روش نیاز به محاسبات فراوان میباشد. ولی مزیت آن نسبت به الگوریتم معادلات ریاضی توانایی در پر کردن اشکال چندضلعی نامنظم است.

رسم اشکال با قطر بیش از یک پیکسل:

جهت رسم اشکال با قطر با بیش از یک پیکسل چهار روش وجود دارد:
: Replaceing pixel

در این روش فرض میشود که قطر شکل یک پیکسل است و شکل رسم میشود. به کمک هر پیکسل میتوان پیکسل های اطراف آن را تعیین نمود. مشکلات این روش عبارتند از :

- 1) ضخامت یکسانی روی خطوط افقی، عمودی و مایل وجود ندارد
- 2) یکنواختی روی اشیاء وجود ندارد.
- 3) برای اشکال غیر از خط محاسبات زیادی باید انجام شود.

روش دوم : Moving Pen

در این روش یک قلم به شکل کادر مستطیل شکل در نظر گرفته میشود که مرکز یا گوشه کادر مستطیل شکل روی یک پیکسل از خط مرزی حرکت میکند. برای رسم خط این روش بهترین نتیجه را خواهد داد.

روش سوم : Filling Area Between Bound Areas

در این روش ابتدا مرز درونی و بیرونی اشیاء تعیین شده و سپس فاصله بین این دو مرز پر خواهد شد. در این روش باید پیکسل های مرز بیرونی را با محاسبات ریاضی بدست آوریم. همچنین بدست آوردن نقاط بین مرزها نیز مشکل است.

روش چهارم : Approximately By Tick Poly Line

با استفاده از تعدادی خط پیوسته که معمولاً دارای طولی کوتاه می باشند، شکل را تقریب میزنند. این روش کارایی چندانی ندارد.

عملیات برش^۸ :

به طور کلی هر رویه، که بخشی از تصویر که داخل یا خارج یک محدوده معین است را مشخص میکند، الگوریتم برش^۹ نامیده میشود. ناحیه ای که عمل برش بر روی آن انجام میشود پنجره برش^{۱۰} نامیده میشود.

عملیات برش معمولاً برای انتخاب بخشی از تصویر، خوش نمازای خطوط و یا مرزهای اشیاء، و همچنین عملیات مربوط به نقاشی و ترسیم (مانند کپی، انتقال و یا حذف ناحیه ای مشخص) کاربرد دارد.

⁸ Clipping Operation

⁹ Clipping Algorithm

¹⁰ Clipping Window

بسته به نوع کاربرد، پنجره برش میتواند یک چندضلعی نامتقارن و یا یک منحنی بسته باشد. در اینجا تنها برش را با پنجره برش مستطیل شکل انجام خواهیم داد.

در این فصل ما برش نقطه و برش خط را بررسی میکنیم. روالهای برش خط و برش ناحیه از اجزاء استاندارد بسته های گرافیکی هستند اما همه این بسته ها دارای روال مخصوص برای برش منحنی و اشکال کانونی(مثل دایره و بیضی) نیستند و ممکن است این اشکال را بصورت خط مستقیم در نظر بگیرند و سپس با استفاده از روال های برش خط و ناحیه به هدف خود برسند.

برش نقطه (Point Clipping)

فرض کنید پنجره برش ما یک مستطیل در موقعیت استاندارد باشد. ما یک نقطه را برای نمایش ذخیره میکنیم اگر معادلات زیر برقرار باشد:

$$\begin{cases} xW_{\min} \leq x \leq xW_{\max} \\ yW_{\min} \leq y \leq yW_{\max} \end{cases}$$

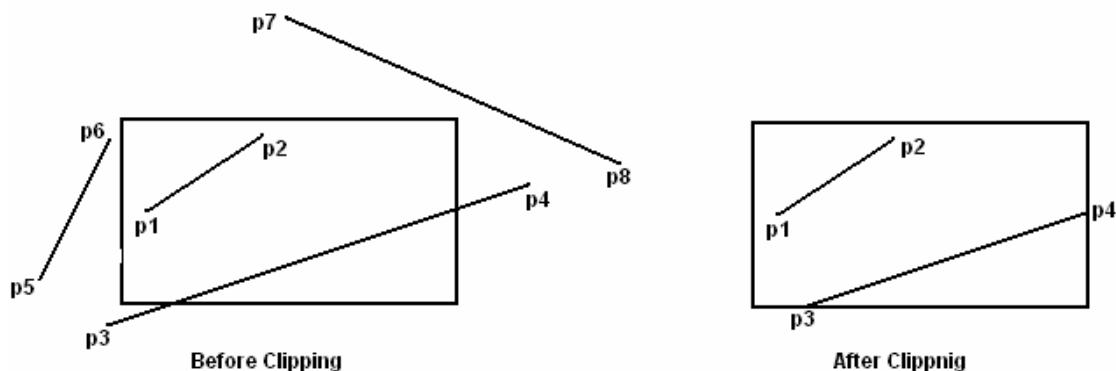
که در آن رئوس پنجره برش مستطیل شکل عبارتند از :

$$(xW_{\min}, xW_{\max}, yW_{\min}, yW_{\max})$$

در صورتیکه حتی یکی از چهار معادله فوق نادرست باشد نقطه حذف خواهد شد(برای نمایش ذخیره نمیشود). با اینکه رویه برش نقطه کمتر از رویه های برش خط و ناحیه کاربرد دارد، با این حال بعضی از کاربردها به این رویه نیازمندند. (مانند صحنه های مربوط به انفجار)

برش خط (Line Clipping)

شكل زیر رابطه بین موقعیت خطوط و موقعیت پنجره برش را برای عملیات برش نشان میدهد.



ابتدا بررسی میکنیم که آیا خط به طور کامل داخل پنجره برش قرار دارد یا نه. اگر قرار نداشت بررسی میکنیم آیا خط کاملا خارج پنجره برش قرار دارد یا نه. در نهایت اگر خط کاملا در داخل یا خارج ناحیه برش نبود با انجام یک سری معادلات تشخیص میدهیم کدام بخش از خط، داخل و کدام بخش خارج پنجره برش قرار دارد. در نهایت هدف ما بدست آوردن الگوریتمی کارآمد برای حذف خطوط خارج از ناحیه با انجام کمترین محاسبات است.

الگوریتم های کارامدی برای برش خط وجود دارد که ما در اینجا مهمترین آنها را بررسی میکنیم. بعضی از این الگوریتمها مختص اشکال دو بعدی می باشند و بعضی دیگر به سادگی به اشکال سه بعدی تعمیم می یابند.

الگوریتم Cohen – Sutherland

این الگوریتم از الگوریتمهای قدیمی است که بدلیل تعداد کم محاسبات از سرعت اجرای بالایی برخوردار است. در این روش به هر یک از دو سر انتهایی خط یک کد دو دویی چهار رقمی نسبت داده می شود که این کد موقعیت خط را نسبت به مرزهای مستطیل برش مشخص می کند. ناحیه ها به صورت زیر کد گذاری می شوند.

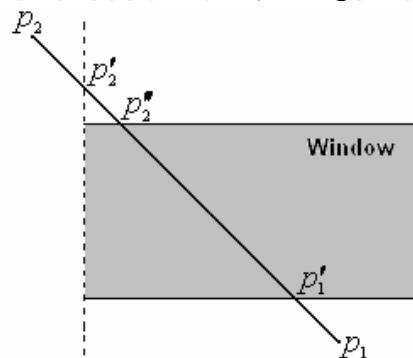
1001	1000	1010	Bit 4	Bit 3	Bit 2	Bit 1
0001	window 0000	0010	if $x < x_{\min}$	Bit1=1		
			if $x > x_{\max}$	Bit2=1		
0101	0100	0110	if $y < y_{\min}$	Bit3=1		
			if $y > y_{\max}$	Bit4=1		

با این کد گذاری میتوان برای تشخیص داد.

الف) خط داخل مستطیل برش است: زمانیکه کد مربوط به انتهای هر دو سر خط برابر 0000 باشد. این خطوط بدون انجام محاسبات ریاضی پذیرفته می شوند. برای تست این شرط میتوان کد مربوط به دو انتهای خط را با هم OR کرد اگر نتیجه 0000 بود خط تماماً داخل ناحیه برش است.

ب) خط خارج ناحیه برش است: زمانیکه حداقل یک بیت 1 در یک مکان مشخص در کد هر دو انتهای خط یافت شود. این خطوط بدون انجام محاسبات ریاضی حذف می شوند. برای تست این شرط میتوان کد مربوط به دو انتهای خط را با هم AND کرد اگر نتیجه چیزی غیر از 0000 بود خط تماماً خارج ناحیه برش است.

ج) بخشی از خط داخل و بخشی از خط خارج ناحیه برش است: هر حالتی به جز حالات بالا. در این حالت برای اینکه موقعیت خط را نسبت به مستطیل تشخیص دهیم نیاز به یک سری محاسبات ریاضی داریم. برای درک عملکرد این الگوریتم، مثالی را تشریح می کنیم. مثال شکل زیر را در نظر بگیرید:



با کد گذاری انتهای خط مشخص می شود خط به طور کامل داخل یا خارج مستطیل برش قرار ندارد و در شرایط اول و دوم صدق نمی کند، برای برش خط نیاز به یک سری محاسبات ریاضی داریم.

برای پیدا کردن نقاط تقاطع میتوان از فرمول شیب خط استفاده کرد. برای خطی با نقاط انتهایی (x_1, y_1) و (x_2, y_2) فرمول خط به صورت زیر است:

$$y = y_1 + m(x - x_1)$$

مقدار m که بصورت زیر محاسبه میشود:

$$m = (y_2 - y_1) / (x_2 - x_1)$$

هر کجا مقدار X برابر با یکی از مقادیر xW_{\max} یا xW_{\min} قرار گیرد، آنگاه مقدار متناظر با y بدست میآید. این شرط ها در زیر آورده شده است. برای نقاط انتهایی شرایط زیر را بررسی میکنیم.

$$x_p \geq xW_{\min} \quad \text{else} \quad P' : y = mxW_{\min} + b$$

$$x_p \leq xW_{\max} \quad \text{else} \quad P' : y = mxW_{\max} + b$$

$$y_p \geq yW_{\min} \quad \text{else} \quad P' : yW_{\min} = mx + b$$

$$y_p \leq yW_{\max} \quad \text{else} \quad P' : yW_{\max} = mx + b$$

ابتدا p_1 را با چهار ضلع مستطیل برش مقایسه میکنیم. نتیجه میگیریم که p_1 در پایین مستطیل قرار دارد. بنابراین p'_1 را که نقطه تقاطع خط با مستطیل برش است را پیدا میکنیم. این کار برای نقطه p_2 نیز انجام میشود. مزیت این الگوریتم محاسبات کم و عیب آن کار کردن تنها با ناحیه برش مستطیل شکل است.

الگوریتم : Cyrus – Beck

این الگوریتم یکی از الگوریتم های سریع برای برش خط است که بر پایه معادلات پارامتری خط و بوسیله Cyrus و Beck توسعه یافته است.

معادله خط در دستگاه دکارتی به سه صورت نوشته میشود:

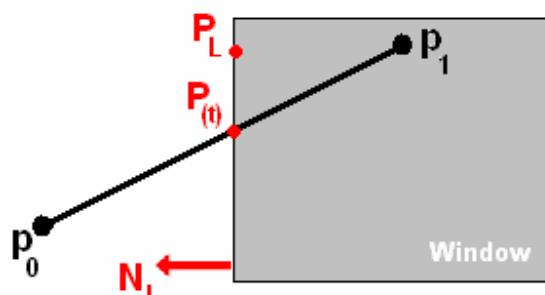
الف) معادله خط بر اساس شیب و عرض از مبدأ آن: $y = mx + b$

ب) معادله ضمنی خط: $Ax + By + C = 0$

ج) معادله پارامتری خط: در این فرم خط بر اساس دو نقطه ابتدایی و انتهایی آن نوشته میشود. $P(t) = P_0 + (P_1 - P_0)t$ که در آن P خود یک بردار به صورت $[x, y]$ است. بنابراین میتوان معادله خط را به صورت دو معادله زیر هم نوشت:

$$\begin{cases} x = x_0 + (x_1 - x_0)t \\ y = y_0 + (y_1 - y_0)t \end{cases}$$

برای درک این الگوریتم مراحل زیر را گام به گام انجام میدهیم:



1) معادله پارامتری خط را مینویسیم.

2) برای هر لبه (ضلع) مستطیل برش دو پارامتر زیر را بدست می‌آوریم:

a. بردار عمود بر لبه N_L که میتواند یکی از مقادیر زیر را اختیار کند:

$$(1,0), (0,1), (-1,0), (0,-1)$$

b. یک نقطه بر روی لبه P_L

3) با توجه به اینکه بردارهای N_L و $P(t) - P_L$ بر هم عمودند داریم :

$$N_L \bullet (P(t) - P_L) = 0$$

4) مقدار $(P(t)$ را در فرمول بالا جایگزین می‌کنیم و سپس مقدار t را از آن بصورت زیر بدست می‌آوریم:

$$t = \frac{N_L \bullet (P_0 - P_L)}{-N_L \bullet (P_1 - P_0)}$$

5) مقدار t را از رابطه بالا برای تمام نقاط مشترک خط و لبه‌های مستطیل برش بدست می‌آوریم.

6) بر اساس مقدار t داریم:

a. تمام $0 < t < 1$ یا $t > 1$ رد می‌شوند.

b. نقاط اشتراکی باقیمانده را رده بندی می‌کنیم.

i. نقاط مستعد برای داخل بودن (PE)

ii. نقاط مستعد برای خارج بودن (PL)

7) در این مرحله کمترین مقدار PL و بیشترین مقدار PE را پیدا و مقدار t آنها را در معادله پارامتری خط جایگزین کرده و خط بین نقاط حاصل را به عنوان برشی از خط اولیه رسم می‌کنیم.