- ALGORITMA GARIS

Algoritma garis adalah metode yang digunakan dalam grafika komputer untuk menggambar garis pada layar. Berikut ini adalah salah satu algoritma yang umum digunakan dalam garis grafika komputer:

1. Algoritma Bresenham:

Algoritma Bresenham adalah salah satu algoritma paling umum yang digunakan untuk menggambar garis. Algoritma ini menggunakan pendekatan inkremental untuk menentukan piksel mana yang harus diwarnai pada setiap langkah untuk menghasilkan garis yang tepat.

Berikut adalah langkah-langkah algoritma Bresenham untuk menggambar garis antara dua titik (x1, y1) dan (x2, y2):

- Hitung perbedaan antara koordinat x dan y titik akhir dan titik awal: dx = x2 - x1 dan dy = y2 - y1.

- Tentukan penambahan (increment) x dan y yang akan digunakan pada setiap langkah. Ini akan bergantung pada tanda dx dan dy.

- Hitung variabel pengambilan keputusan (decision variable) initial, biasanya menggunakan rumus decision = 2 \* dy - dx.

- Mulai dari titik awal (x1, y1) dan warnai piksel.

- Untuk setiap langkah, pilih piksel berikutnya dengan mengubah koordinat x dan y menggunakan penambahan yang telah ditentukan.

- Hitung nilai decision variable berikutnya dan sesuaikan koordinat x atau y jika diperlukan.

- Ulangi langkah-langkah 5-6 sampai mencapai titik akhir (x2, y2).

Algoritma Bresenham sangat efisien karena menghindari penggunaan operasi floating-point. Hal ini membuatnya cocok untuk penggunaan dalam sistem grafika komputer yang memiliki keterbatasan perangkat keras.

Ada juga algoritma garis lainnya seperti algoritma DDA (Digital Differential Analyzer) yang menggunakan pendekatan inkremental yang serupa, tetapi menggunakan operasi floating-point untuk menghitung perubahan koordinat dalam setiap langkah.

Salah satu algoritma garis yang umum digunakan dalam grafika komputer adalah Algoritma DDA (Digital Differential Analyzer). Algoritma ini menggunakan pendekatan inkremental untuk menggambar garis antara dua titik.

Berikut adalah langkah-langkah algoritma DDA untuk menggambar garis antara dua titik (x1, y1) dan (x2, y2):

1. Hitung perbedaan antara koordinat x dan y titik akhir dan titik awal: dx = x2 - x1 dan dy = y2 - y1.

2. Tentukan jumlah langkah yang akan diambil berdasarkan perbedaan antara dx dan dy. Jumlah langkah ini akan menjadi jumlah piksel yang harus diwarnai.

- Jika |dx| > |dy|, maka jumlah langkah = |dx|.

- Jika |dy| > |dx|, maka jumlah langkah = |dy|.

- Jika |dx| = |dy|, maka jumlah langkah = |dx| = |dy|.

3. Hitung perubahan inkremental dalam nilai x (deltax) dan y (deltay) untuk setiap langkah:

- deltax = dx / jumlah langkah.

- deltay = dy / jumlah langkah.

4. Mulai dari titik awal (x1, y1) dan warnai piksel. Lanjutkan ke titik berikutnya dengan menambahkan deltax ke x dan deltay ke y.

- x = x + deltax.

- y = y + deltay.

5. Ulangi langkah 4 sebanyak jumlah langkah yang ditentukan. Setiap langkah, warnai piksel dengan koordinat (x, y).

Algoritma DDA menggunakan operasi floating-point untuk menghitung perubahan koordinat dalam setiap langkah. Ini membuatnya lebih akurat dalam menghasilkan garis yang halus dibandingkan dengan algoritma yang hanya menggunakan operasi integer. Namun, karena operasi floating-point memakan waktu dan sumber daya komputasi yang lebih tinggi, algoritma DDA mungkin kurang efisien dalam implementasinya dibandingkan dengan algoritma inkremental lain seperti algoritma Bresenham.

- TRANSFORMASI

Transformasi dalam grafika komputer merujuk pada manipulasi objek grafis, seperti mengubah posisi, ukuran, rotasi, dan transformasi lainnya. Berikut adalah beberapa transformasi umum yang digunakan dalam grafika komputer:

1. Translasi:

Translasi adalah transformasi yang menggeser posisi objek dalam ruang. Objek dapat digeser ke arah vertikal, horizontal, atau diagonal. Translasi dilakukan dengan menambahkan atau mengurangkan nilai koordinat x dan y dari setiap titik objek.

2. Skala:

Skala adalah transformasi yang mengubah ukuran objek. Dalam skala, setiap titik objek dikalikan dengan faktor skala untuk mengubah jaraknya dari pusat koordinat. Jika faktor skala lebih besar dari 1, objek akan diperbesar, sedangkan jika faktor skala kurang dari 1, objek akan diperkecil.

3. Rotasi:

Rotasi adalah transformasi yang memutar objek sekitar pusat rotasi. Objek dapat diputar dalam arah searah jarum jam (positif) atau berlawanan arah jarum jam (negatif) sebesar sudut yang ditentukan. Rotasi sering diukur dalam derajat atau radian.

4. Pencerminan:

Pencerminan adalah transformasi yang membalik objek terhadap sumbu tertentu. Sumbu pencerminan dapat berupa sumbu x, y, atau garis lainnya. Dalam pencerminan, koordinat objek diubah menjadi nilai negatif pada sumbu yang dipilih.

5. Shearing (geser memanjang):

Shearing adalah transformasi yang menggeser objek dalam satu arah sedangkan arah lainnya tetap. Shearing terjadi ketika setiap titik dalam objek digeser dalam arah tertentu tergantung pada koordinatnya. Ini dapat menciptakan efek memanjang atau miring pada objek.

6. Transformasi perspektif:

Transformasi perspektif digunakan untuk menciptakan efek perspektif dalam grafika 3D. Transformasi ini melibatkan perhitungan koordinat objek dalam ruang 3D menjadi koordinat 2D pada layar dengan mempertimbangkan jarak dan sudut pandang pengamat.

Transformasi-transformasi ini dapat diterapkan pada setiap titik objek secara terpisah atau pada seluruh objek secara keseluruhan. Transformasi dapat dilakukan dengan menggunakan matriks transformasi atau dengan algoritma yang sesuai untuk setiap jenis transformasi.

- CLIPPING

Clipping (penyayatan) adalah proses dalam grafika komputer untuk memotong atau membatasi objek yang berada di luar jendela pandangan atau viewport. Ketika objek berada di luar jendela pandangan, hanya bagian objek yang terlihat atau berada di dalam jendela yang akan digambar, sementara bagian yang berada di luar akan dihilangkan atau tidak ditampilkan.

Clipping penting dalam grafika komputer karena memungkinkan kita untuk mengatur tampilan objek secara efisien dan mencegah gambar yang tidak perlu dihasilkan, yang dapat mengurangi kecepatan dan sumber daya yang diperlukan untuk rendering.

Ada beberapa algoritma yang umum digunakan dalam clipping, termasuk:

1. Clipping Cohen-Sutherland:

Algoritma Cohen-Sutherland adalah salah satu algoritma pemotongan garis yang paling umum digunakan. Algoritma ini membagi jendela pandangan menjadi sembilan bagian dengan menggunakan area kode, yang mewakili posisi relatif objek terhadap jendela. Setiap garis atau segmen garis akan diuji dengan area kode untuk menentukan apakah perpotongan dengan jendela terjadi. Jika tidak ada perpotongan, garis dapat dibuang. Jika ada perpotongan, garis akan dipotong sesuai dengan batas jendela.

2. Clipping Cyrus-Beck:

Algoritma Cyrus-Beck adalah algoritma pemotongan garis umum lainnya yang berbasis pada konsep vektor. Algoritma ini menghitung titik potong antara garis dan batas jendela dengan memanipulasi vektor normal dan vektor parameter.

3. Clipping Liang-Barsky:

Algoritma Liang-Barsky adalah algoritma lain yang digunakan untuk pemotongan garis. Algoritma ini bekerja dengan menghitung parameter yang menggambarkan perpotongan garis dengan batas jendela. Jika garis berada di luar jendela, parameter akan disesuaikan sehingga garis dipotong dengan batas jendela.

Selain itu, ada juga algoritma pemotongan lainnya yang digunakan untuk memotong objek dua dimensi, seperti pemotongan poligon Sutherland-Hodgman, pemotongan lingkaran atau elips dengan algoritma midpoint, dan lain-lain.

Pemotongan juga dapat dilakukan dalam tiga dimensi untuk objek yang berada di luar frustum pandangan kamera, menggunakan algoritma pemotongan tiga dimensi seperti frustum culling atau algoritma pemotongan perspektif.