

Algoritma Klipping Garis yang Efisien dalam Ruang 2D

Mamatha Elliriki¹, Chandrasekhara Reddy², dan Krishna Anand³

¹Departemen Matematika, Universitas GITAM, India

²Departemen Matematika, Cambridge Institute of Technology-NC, India

³Departemen Ilmu Komputer, Institut Sains dan Teknologi Sreenidhi, India

Abstrak: Masalah clipping tampaknya cukup sederhana dari sudut pandang manusia karena dengan visualisasi sebuah garis dapat dengan mudah dilacak apakah benar-benar berada di dalam dan jika tidak bagian dari garis yang terletak di luar jendela. Namun, dari sudut pandang sistem, jumlah perhitungan dan perbandingan untuk garis dengan perhitungan floating point sangat besar yang pada gilirannya menambah kompleksitas yang melekat. Perlu meminimalkan jumlah komputasi sehingga mencapai peningkatan yang signifikan dalam hal efisiensi. Dalam karya ini, model matematis telah diusulkan untuk mengevaluasi titik persimpangan sehingga memotong garis yang mengandalkan perhitungan integral. Selain itu, tidak ada perhitungan lebih lanjut yang diperlukan untuk mengevaluasi titik persimpangan.

Kata kunci: Panjang orto, sistem grafik raster, klipping garis, titik persimpangan, kemiringan geometris, jendela persegi panjang.

Diterima 23 Juli 2015; diterima 1 Juni 2016

1. Perkenalan

Penggunaan klipping garis dalam berbagai aplikasi ilmiah tampaknya meningkat secara eksponensial. Meskipun konsepnya tampak sederhana pada awalnya, besarnya penggunaannya membuat penelitian ilmiah tentang hal yang sama menjadi kebutuhan yang mendesak. Dengan perspektif ini dalam pikiran, pekerjaan ini telah mengalihkan fokusnya untuk meningkatkan efisiensi algoritma klipping garis [4]. Hal ini pada gilirannya juga memberikan dorongan tambahan untuk melakukan perbaikan dalam pemotongan poligon yang juga telah disoroti pekerjaan ini.

Mungkin, algoritma klipping garis tampaknya menjadi algoritma terpenting kedua setelah algoritma menggambar garis di titik peringat [10] pandang dalam grafik. Secara umum, jendela klipping dalam sistem pemindaian raster biasanya dikaitkan dengan jendela persegi panjang. Namun, bentuk jendela termasuk lingkaran dan poligonal [16, 26] juga dapat dipertimbangkan. Jendela persegi mudah dipetakan ke layar komputer menggunakan transformasi yang disebut 'pemetaan jendela ke viewport' yang pada gilirannya terdiri dari translasi, penskalaan, dan rotasi.

Tujuan utama clipping dalam sistem grafis komputer dua Dimensi adalah untuk menghilangkan objek, garis atau segmen garis yang berada di luar jendela view port dan clipping of lines merupakan konsep dasar dalam proses visualisasi [9]. Objek dunia nyata dapat direpresentasikan relatif terhadap sistem koordinat dunia referensi [12]. Sulit untuk melihat semua objek di layar komputer secara bersamaan dalam satu tangkapan layar karena biasanya menempati banyak tempat di sistem koordinat dunia. Seperti dalam persepsi manusia dan kamera, satu tangkapan layar berisi gambar beberapa objek, yang sebagian mungkin terpotong. Di

mengidentifikasi objek atau bagian dari objek yang perlu ditampilkan pada satu tangkapan layar, algoritma klipping perlu dijalankan pada setiap objek untuk menentukan apakah itu akan muncul sepenuhnya atau sebagian di layar atau akan terpotong. Algoritma klipping [7, 11, 23, 24, 27] biasanya memiliki parameter yang menggambarkan 'jendela klipping' di dunia 2D atau 'volume klipping' di dunia 3D selain dari representasi objek yang diuji untuk klipping.

Sejumlah besar aplikasi komputer dan seluler membutuhkan perangkat keras yang kuat ditambah dengan algoritma yang efisien dan cepat [3, 13, 19] untuk keberhasilan penggunaannya. Selain itu, pemilihan alat yang tepat membantu dalam penggunaan aplikasi dan memberikan tingkat kecanggihan tambahan yang muncul dalam animasi dan antarmuka pengguna grafis [1]. Selain itu, implementasi perangkat keras memberikan arti khusus untuk pemrosesan yang cepat dan solusi yang akurat.

Sisa dari makalah ini disusun sebagai berikut. Bagian 2, secara singkat menjelaskan tentang algoritma yang ada untuk memotong garis di jendela persegi panjang. Bagian 3 menyajikan analisis masalah dan berbagai kemungkinan garis yang ada dalam sistem koordinat dua dimensi. Rumus komputasi diturunkan di bagian 4 dan rumus ini sangat menyederhanakan pekerjaan ekstraksi garis yang perlu disimpan dalam sistem raster. Secara rinci studi dan pengembangan algoritma dijelaskan untuk memotong garis dari garis asli di bagian 5. Pada bagian 6 berbagai algoritma dijelaskan untuk pemanfaatan sumber daya yang optimal untuk kasus yang berbeda. Bagian 7 menyimpulkan analisis kinerja dan perbandingan dengan algoritma sebelumnya.

2. Algoritma yang Ada

Penelitian telah menunjukkan bahwa, komputasi dengan setiap persimpangan antara garis dan tepi jendela membutuhkan lebih banyak siklus untuk Central Processing Unit (CPU) karena menggabungkan operasi floating point. Beberapa algoritma diperkenalkan ke garis klip terhadap jendela persegi panjang [2, 15, 21] dan non-persegi panjang [12, 28]. Pengurangan jumlah simpang akan berdampak pada peningkatan efisiensi. Dalam makalah ini, algoritma yang efisien diusulkan untuk meminimalkan jumlah persimpangan sebanyak mungkin.

Saat ini, dua jenis pendekatan tersedia untuk memotong garis sehubungan dengan jendela kliping. Salah satu metodologi ini adalah untuk menangani pendekatan logis [18, 20, 25] di mana setiap segmen garis dipotong secara identik dengan membagi wilayah dengan kode logis. Algoritma Cohen Sutherland (CS) berada di bawah pendekatan ini dan berurusan dengan pengkodean titik akhir segmen garis. Algoritma ini ditemukan sangat efektif ketika seluruh baris ditemukan berada di dalam jendela. Untuk garis yang benar-benar di luar, perhitungan yang tidak perlu perlu dilakukan. Kompleksitas komputasi telah ditemukan meningkat ketika garis ditemukan sebagian di dalam jendela [8]. Pendekatan kedua berkaitan dengan parameterisasi [15, 29]. Algoritma Cyrus Beck [5] dan Liang Barsky (LB) [15] termasuk dalam kategori kedua dan algoritma yang dikembangkan dengan bantuan persamaan parametrik. Metode-metode ini diskemakan berdasarkan perilaku segmen garis. Namun, telah ditemukan tidak efisien dalam banyak skenario.

Sebuah versi modifikasi untuk kliping telah dicoba oleh Nicholl Lee Nicholl (NLN) [21] di mana mereka merancang algoritma parametrik hibrida menggunakan kedua teknik. Keuntungan utama dari pendekatan hybrid ini berkaitan dengan minimalisasi operasi yang memakan waktu seperti perkalian dan pembagian [14]. Mereka mampu mengurangi jumlah perbandingan dengan dua pertiga sehubungan dengan algoritma Cohen-Sutherland dan setengahnya sehubungan dengan algoritma Liang dan Barsky [15]. Namun, waktu yang digunakan dalam pembuatan region sangat penting karena algoritme awalnya perlu menentukan posisi titik akhir. Selain itu, jumlah upaya yang dilakukan untuk menerapkan transformasi geometris dan membandingkan kemiringan terletak di sisi lain yang lebih tinggi dan menyimpan perhitungan ini untuk pengujian lebih lanjut membutuhkan lebih banyak memori.

Semua algoritme yang dijelaskan di sini tidak memenuhi persyaratan kecepatan dari kebutuhan sehari-hari. Menjaga sudut pandang ini dalam perspektif, dibutuhkan algoritma yang jauh lebih cepat [17, 22, 30] untuk jendela persegi panjang dan ruang poligon di dunia saat ini. Namun, diamati bahwa untuk beberapa skenario, tingkat kerumitan ditemukan meningkat pesat yang pada gilirannya menyebabkan hilangnya efisiensi yang serius. Untuk memperbaiki kekurangan yang melekat pada ini:

metode, algoritma kliping baru untuk jendela kompleks telah diusulkan oleh Skala [24, 25] dan Day telah mengusulkan algoritma jendela kecil dua dimensi [6]. Penggunaan pencarian biner dalam algoritma ini telah menyebabkan penurunan tingkat kerumitan yang cukup besar. Namun, gain yang diamati tidak dapat direplikasi pada pertimbangan jendela persegi panjang.

Prosedur langsung untuk memotong garis adalah mengevaluasi semua titik persimpangan garis dengan tepi yang terletak di jendela persegi panjang. Diamati bahwa masalah kliping lebih sulit daripada mengevaluasi titik persimpangan dari garis tak terbatas. Dalam metode kliping, sangat penting untuk mengenali apakah titik persimpangan untuk garis terletak di dalam atau di tepi atau di luar. Metode yang sama berlaku untuk keempat tepi jendela persegi panjang. Titik perpotongan dari segmen garis yang terletak di luar tepi atau di luar titik akhir tidak ada artinya pada garis simpan.

3. Analisis Komputasi dan Pengembangan Algoritma

Untuk menghitung titik-titik perpotongan suatu garis, perlu dilambangkan ruas garis yang dihubungkan oleh dua titik akhir ($x_{Mulailah}$, $kamu_{Mulailah}$) dan (x_{akhir} , $kamu_{akhir}$) dan parameter jendela kliping persegi panjang adalah (x_{kiri} , $kamu_{bawah}$) dan (x_{Baik} , $kamu_{puncak}$); koordinat sudut kiri bawah dan kanan atas dinyatakan dalam satuan dunia nyata. Parameter viewport layar diberikan dalam piksel di ruang gambar. Tanpa kehilangan keumuman, ($x_{Mulailah}$, $kamu_{Mulailah}$) dapat dianggap sebagai titik awal yang dekat dengan x -sumbu dan (x_{akhir} , $kamu_{akhir}$) adalah titik akhir segmen garis lainnya. Dari sini, dapat diamati bahwa jika kemiringan garis lurusnya positif, maka $x_{Mulailah} < x_{akhir}$. Jika tidak, $x_{Mulailah} > x_{akhir}$. Tapi untuk semua lereng, $kamu_{akhir}$ selalu lebih besar dari atau sama dengan $kamu_{Mulailah}$ yaitu, $kamu_{Mulailah} \leq kamu_{akhir}$.

Informasi ini disorot dalam Gambar 9 dan 12. Ketidaksetaraan ini berguna untuk mengurangi jumlah perbandingan dalam pengujian di dalam/di luar. Sekarang, garis dapat diputuskan dengan salah satu tes berikut.

Penerimaan Lengkap: Seluruh baris terletak di dalam jendela kliping. Ini menyiratkan bahwa seluruh baris harus disimpan karena tidak ada pemotongan yang terjadi. Perbandingan yang ditunjukkan dalam Persamaan (1) dianggap cukup. Ini memberikan dorongan untuk kriteria penerimaan ini seperti yang ditunjukkan pada Gambar 2.

$$x_{kiri} \leq x_{Mulailah}, x_{akhir} \leq x_{Baik} \text{ \& } kamu_{bawah} \leq kamu_{Mulailah}, kamu_{akhir} \leq kamu_{puncak} \quad (1)$$

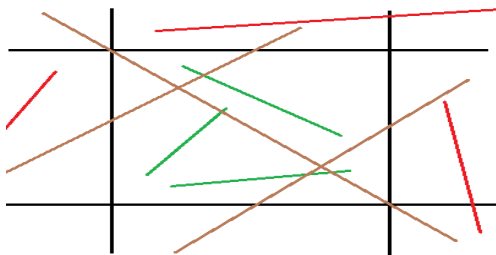
Penolakan Lengkap: Seluruh garis terletak di luar dan pada satu sisi jendela kliping dan tidak akan memotong sisi-sisi yang tegak lurus, sehingga garis dapat ditolak. Perbandingan yang ditunjukkan pada Persamaan (2) menyoroti kasus seperti yang ditunjukkan pada Gambar 3.

$$\{x_{Mulailah}, x_{akhir}\} \cdot x_{kiri} \text{ \& } \{x_{Mulailah}, x_{akhir}\} \cdot x_{Baik} \text{ \& } kamu_{bawah} \leq kamu_{akhir} \text{ \& } kamu_{bawah} \text{ \& } kamu_{Mulailah} \text{ \& } kamu_{puncak} \leq kamu_{Mulailah} \quad (2)$$

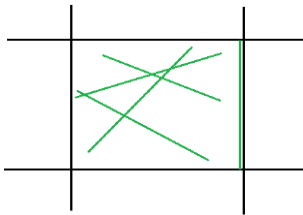
Penerimaan Sebagian (Kasus 1): Kedua titik akhir ditempatkan di luar jendela persegi panjang dan

garis memotong tepi tegak lurus dari jendela kliping. Dalam hal ini, jika beberapa bagian dari garis berada di dalam jendela kliping maka bagian dari segmen garis tersebut harus diekstraksi bersama dengan titik akhir yang terletak di jendela kliping. Kasus ini adalah yang paling kompleks dan disajikan secara rinci di bagian 5.1. dan 5.2.

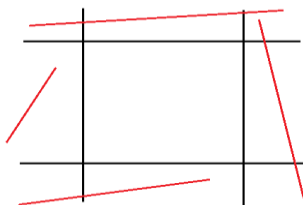
Penerimaan Sebagian (Kasus 2): Satu titik ujung berada di dalam jendela sementara ujung lainnya berada di luar. Sangat penting untuk mengetahui titik perpotongan garis dengan salah satu dari empat tepi jendela kliping dan garis harus ditarik dari titik dalam ke titik perpotongan. Situasi ini dianggap sebagai bagian dari yang sebelumnya dan dibahas di bagian 5.3. sebagai kasus khusus.



Gambar 1. Segmen garis umum yang akan diuji untuk dijepit di jendela persegi panjang.



Gambar 2. Kumpulan segmen garis terletak sepenuhnya di dalam jendela persegi panjang (Uji 1).



Gambar 3. Segmen garis yang ditolak sepenuhnya yang terletak seluruhnya di luar jendela kliping (Uji 2).

4. Rumus Panjang Ortho untuk Titik Persimpangan

Pembentukan rumus komputasi diinginkan untuk pengembangan algoritma dan dianggap sebagai pekerjaan signifikan yang dikembangkan menggunakan fundamental geometri. Seperti disebutkan sebelumnya, $(x_{Mulailah}, kamu_{Mulailah})$ dan $(x_{Akhir}, kamu_{Akhir})$ adalah titik akhir suatu garis dan x_{tepi} adalah dimensi tepi kiri atau kanan dari jendela persegi panjang. Demikian pula, $kamu_{tepi}$ adalah dimensi tepi atas atau bawah. Definisikan, lebih lanjut $x = x_{Akhir} - x_{Mulailah}$ dan $y = kamu_{Akhir} - kamu_{Mulailah}$ yang adalah jarak dalam hal x arah dan y arah masing-masing dengan titik akhir garis. Dari sini, jika kemiringan positif maka produk dari x dan y akan s

lebih besar dari atau sama dengan nol ($x * y \geq 0$) tanda produk lain adalah negatif. Dengan bantuan dimensi kliping dan titik akhir segmen garis, titik perpotongan dapat dengan mudah dievaluasi dan rumus-rumus ini disajikan pada bagian selanjutnya dari bagian ini.

4.1. Perpotongan Ruas Garis dengan $x = x_{tepi}$ (y tidak diketahui)

Biarkan segmen garis dengan titik akhir $P(x_{Mulailah}, kamu_{Mulailah})$ dan $Q(x_{Akhir}, kamu_{Akhir})$ dan berpotongan $x = x_{tepi}$ pada intinya $C(x_{tepi}, R_y)$. R_{kamu} didefinisikan sebagai panjang orto, adalah jarak tegak lurus dari sumbu x titik potong dan berperan strategis dalam menentukan ruas garis. Dari geometri, kemiringan garis lurus untuk dua titik sembarang adalah sama [8, 10, 11]. yaitu,

$$\text{kemiringan}\{P(x, y), Q(x_y)\} = \text{kemiringan}\{C(x_{tepi}, R_y), Q(x_y)\} \quad (3)$$

$$\frac{kamu_{Akhir} - kamu_{Mulailah}}{x_{Akhir} - x_{Mulailah}} = \frac{kamu_{Akhir} - R_{kamu}}{x_{Akhir} - x_{tepi}} \quad (4)$$

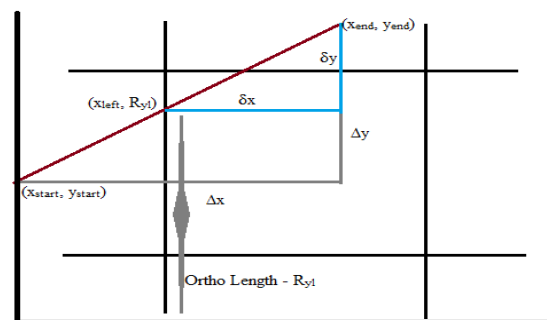
$$\frac{kamu_{kamu}}{x} = \frac{kamu}{x} \quad (5)$$

dimana $x, \Delta y$ dan $x = x_{Akhir} - x_{tepi}$ adalah konstanta yang diketahui.

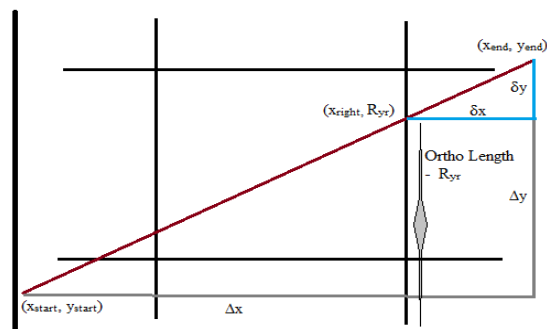
$kamu_{kamu}$ adalah konstanta untuk dievaluasi.

Dari Persamaan (5), panjang orto dapat dihitung sebagai:

$$R_{kamu} = kamu_{Akhir} - \frac{kamu}{x} \quad (6)$$



Gambar 4. Ortho length Ryl dalam representasi sistem koordinat 2D.



Gambar 5. Tampilan grafis panjang orto Ryr digunakan untuk mengevaluasi titik potong tepi kanan.

Persamaan (6) berlaku bahkan untuk menukar titik akhir segmen garis. Sekarang, jika segmen garis

memotong tepi kiri jendela kliping pada titik (X_{kiri}, R_l) , (Gambar 4) maka Persamaan (6) menjadi:

$$R_{kamu} = \frac{kamu}{x} \text{ dimana } x = X_{akhir} - X_{kiri} \quad (7)$$

Analisis serupa dapat diberikan untuk segmen garis yang memotong tepi kanan jendela persegi panjang. Membiarkan R_{tahun} menjadi jarak orto dari sumbu x ke titik potong garis dengan tepi kanan jendela kliping seperti yang disajikan pada Gambar 5. Maka Persamaan (6) menjadi

$$R_{tahun} = \frac{kamu}{x} \text{ dimana } x = X_{akhir} - X_{Baik} \quad (8)$$

4.2. Perpotongan ruas garis dengan $y=y_{edge}$ (δx tidak diketahui)

Bagian 4.1. difokuskan pada segmen garis kemiringan positif. Dapat diamati bahwa eksperimen yang sesuai dapat dilakukan dengan kemiringan negatif juga. Sebuah kasus dipertimbangkan di mana garis PQ berpotongan dengan garis lurus $y = y_{tepi}$ pada intinya $D(R_x, kamu_{tepi})$ dimana R_x adalah jarak orto dari $(0, y_{tepi})$ untuk $(R_x, kamu_{tepi})$. Persamaan (9) berkaitan dengan perumusan kemiringan. Persamaan (10) dan (11) dapat diturunkan dari persamaan kemiringan (9).

$$\text{kemiringan}(P(x_{Mulallah}, kamu_{Mulallah}), Q(x_{Mulallah}, kamu_{Mulallah})) = \frac{kamu_{Mulallah} - kamu_{Mulallah}}{x_{Mulallah} - x_{Mulallah}} \quad (10)$$

$$\frac{kamu_{Mulallah} - kamu_{Mulallah}}{x_{Mulallah} - x_{Mulallah}} = \frac{x}{kamu} \quad (11)$$

x dan x_{akhir} R_x tidak diketahui konstanta untuk dievaluasi; x , y dan $kamu$ diketahui konstanta dapat dievaluasi dengan parameter yang diberikan.

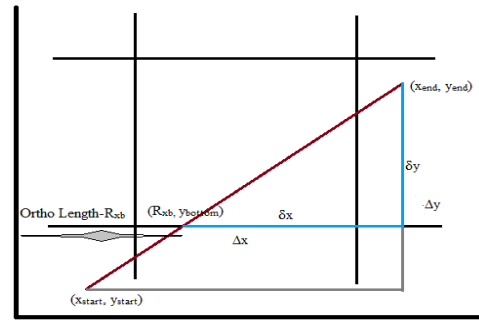
Dari Persamaan (11), panjang orto dihitung. Jarak antara sumbu y dan titik perpotongan pada jendela kliping dapat dihitung seperti yang ditunjukkan pada: Persamaan (12).

$$R_x = x_{akhir} - \frac{x}{kamu} \quad (12)$$

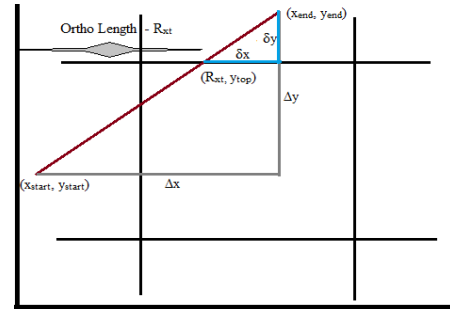
R_{xb} dan R_{xt} adalah panjang orto seperti yang disebutkan pada Gambar 6, dan 7 masing-masing adalah jarak dari sumbu y ke tepi bawah dan atas jendela kliping. Persamaan (13) dan (14) akan diperoleh dari persamaan (12).

$$R_{xb} = x_{akhir} - \frac{x}{kamu} \text{ di mana? kamu } kamu_{Mulallah} \quad (13)$$

$$R_{xt} = x_{akhir} - \frac{x}{kamu} \text{ di mana? kamu } kamu_{Mulallah} \quad (14)$$



Gambar 6. Evaluasi panjang orto R_{xb} untuk ruas garis yang menarik tepi bawah.



Gambar 7. Bentuk grafik orto panjang R_{xt} dalam sistem koordinat 2D.

Persamaan (7), (8), (13), dan (14), mewakili rumus jarak persimpangan untuk garis yang memotong jendela persegi panjang. Rumus ini memberikan indikasi yang tepat tentang keberadaan garis di dalam jendela persegi panjang. Besaran x , y , x dan y dapat dengan mudah dievaluasi dengan bantuan titik-titik tertentu dari segmen garis dan dimensi jendela kliping. Rumus kuantitas dimensi panjang orto mengungkapkan kesetaraan bahkan jika titik akhir dipertukarkan.

Dari sini diketahui bahwa untuk semua kemiringan ruas garis, evaluasi jarak dari sumbu x ke titik potong garis dengan tepi kiri dan kanan jendela kliping memiliki rumus yang sama dengan variasi $smallx$ dan y yang kecil. Rumus ini selanjutnya dapat disederhanakan dengan menggunakan fungsi roundup dan floor dalam bentuk integral sebagai:

Untuk tepi kiri $x = x_{kiri}$ dari jendela kliping

$$R_{kamu} = \frac{kamu}{x} \text{ dimana } x = x_{kiri} \quad (15)$$

$$R_{kamu} = \frac{2 \text{ kamu } x}{2 x} \text{ dimana } x = x_{kiri} \quad (16)$$

Mengikuti prosedur serupa untuk tepi ght $x = x_{Baik}$ dari jendela kliping, kami memiliki:

$$R_{tahun} = \frac{2 \text{ kamu } x}{2 x} \text{ dimana } x = x_{Baik} \quad (17)$$

Untuk semua kemiringan segmen garis, evaluasi jarak integral dari sumbu y titik potong garis dengan tepi bawah dan atas jendela kliping adalah:

Untuk tepi bawah $y = y_{bawah}$ dari jendela kliping

$$R_{xb} \quad x_{akhir} \quad \frac{2 \times kamu \quad kamu}{2 \quad kamu} \quad \text{dimana } kamu \quad kamu_{akhir} \quad kamu_{bawah} \quad (18)$$

Untuk tepi atas $y = y_{puncak}$ dari jendela kliping

$$R_{xt} \quad x_{akhir} \quad \frac{2 \times kamu \quad kamu}{2 \quad kamu} \quad \text{dimana } kamu \quad kamu_{akhir} \quad kamu_{atas} \quad (19)$$

Dimana $f()$ adalah fungsi lantai yang didefinisikan sebagai bagian bilangan bulat dari fungsi $f(\omega)$.

Persamaan ini, dalam C compiler menghitung sepertiga kali lebih cepat dari perhitungan floating point. Sebagian besar garis dihilangkan dengan uji pertidaksamaan. Semua baris yang tersisa dikembangkan dengan bantuan ortho panjang R_{aku} , R_{tahun} , R_{xb} , dan R_{xt} dan ini adalah faktor keputusan untuk memilih apakah garis itu di dalam atau di luar dari jendela kliping. Jika uji pertidaksamaan (1) dan (2) gagal untuk suatu garis maka kemungkinan lain untuk garis tersebut adalah kedua titik ujung garis berada di luar yang memotong sisi tegak lurus jendela kliping atau salah satu ujung garis berada di dalam. Sekarang mari kita menangani masalah pertama sebagai dua kasus khusus dengan bantuan kemiringan garis. Untuk sembarang kemiringan garis dapat positif atau negatif dan dapat ditentukan sebagai positif jika pertidaksamaan $x * y > 0$ atau dalam bentuk yang mudah $x_{akhir} \times y_{mulallah}$ berlaku baik yang merupakan perhitungan nilai integral, jika tidak maka negatif. Bagian berikut membahas studi terperinci untuk garis yang menerima sebagian atau sepenuhnya menolak.

5. Tes Kliping Garis

Sebagian besar garis memerlukan beberapa tes untuk menentukan bagian garis yang tersisa di dalam jendela kliping (Gambar 1, 2, dan 3). Semua ortholengths R_{aku} , R_{tahun} , R_{xb} dan R_{xt} dari suatu garis dievaluasi terlebih dahulu dan perpotongan yang sesuai telah dihitung. Jika semua titik potong dengan panjang orto berada di luar jendela maka seluruh garis ditolak. Untuk garis yang sebagian berada di dalam jendela kliping, tepat dua titik berada di tepi jendela dan dua titik berada di luar. Dalam hal ini, garis dengan dua titik ujung yang berada di dalam jendela digambar dan bagian garis yang tersisa dibuang.

Evaluasi panjang orto dan keempat titik persimpangan dapat dihindari dalam kasus garis tak terlihat melalui uji perbandingan sehingga menghemat tenaga dan waktu. Semua garis yang tidak terlihat dapat dihilangkan dengan melakukan maksimal dua perbandingan. Namun, paling banyak empat perbandingan diperlukan untuk garis yang terlihat sebagian. Untuk perhitungan lebih lanjut, seperti yang dinyatakan sebelumnya, titik pertama dekat dengan sumbu x dan titik akhir garis berada di atas titik pertama. Dari sini, jika $x * y > 0$, maka kemiringannya positif dan ditarik garis dari kiri ke kanan dan/atau diagonal di atas dalam sistem koordinat, sehingga $x_{akhir} \times y_{mulallah}$ dan $kamu_{akhir} \times y_{mulallah}$. Kebalikannya berlaku untuk lereng negatif. Dari sini dapat dicapai bahwa, terlepas dari semua kemiringan, penulis telah mengambil dua titik akhir sedemikian rupa sehingga $kamu_{akhir}$ selalu lebih besar atau sama dengan $kamu_{mulallah}$ ($kamu_{akhir} \times y_{mulallah}$).

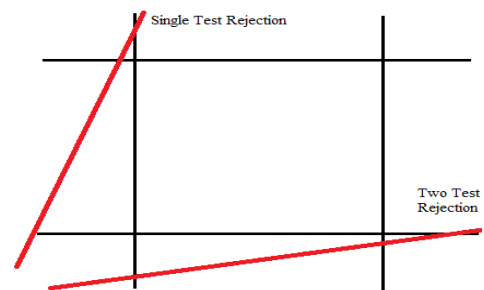
Batasan pertidaksamaan ini digunakan untuk mengurangi perbandingan untuk garis yang terlihat sebagian dan garis yang ditolak seluruhnya.

Prosedur untuk garis dengan kemiringan positif adalah sebagai berikut. Jika $R_{say} > kamu_{puncak}$, maka seluruh baris berada di luar jendela kliping dan dapat ditolak (Gambar 8). Ini dapat divisualisasikan menggunakan analogi sederhana. Ryl adalah jarak dari sumbu x ke titik perpotongan garis dengan tepi kiri jendela dan kemiringan garis positif. Oleh karena itu, titik-titik garis dari titik menatap ke (x_{kiri}, R_{yrl}) jatuh ke kiri jendela kliping dan di luar. Menurut properti kemiringan, sering digunakan dalam algoritma NLN, jika kemiringan antara titik pertama dan titik persimpangan lebih besar dari kemiringan antara titik pertama dan puncak kiri atas jendela maka seluruh garis berada di luar, garis adalah ditolak. Jika $R_{tahun} < kamu_{bawah}$ kemudian garis memotong tepi kanan di bawah jendela kliping dan lewat di luar, karenanya ditolak (Gambar 8). Dalam hal ini, garis dengan titik $(x_{say}, kamu_{say})$ untuk semua $kamu_{say} < R_{tahun}$ di bawah jendela kliping dan bagian yang tersisa dari garis adalah sisi kanan ke jendela.

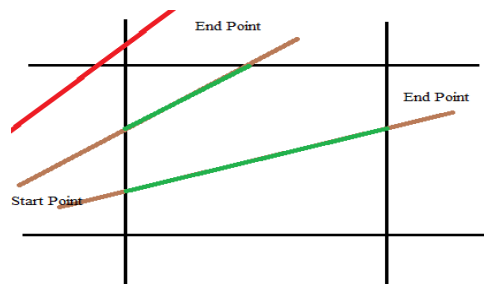
Analisis serupa dapat dilakukan untuk lereng negatif (Gambar 11). Untuk semua kasus yang tersisa, bagian dari garis berada di dalam jendela kliping dan prosedur selanjutnya digunakan untuk menyimpan bagian dari segmen garis.

5.1. Garis lurus dengan Lereng Positif

Mulanya, R_{kamu} telah dihitung dan diperiksa jangkauannya. Jika $kamu_{bawah} < kamu_{puncak}$, kemudian garis memasuki jendela kliping pada titik (x_{kiri}, R_{say}) .

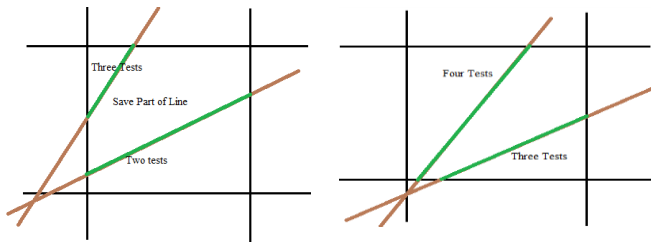


Gambar 8. Segmen garis dengan kemiringan positif.



Gambar 9. Ruas garis dengan kemiringan positif.

Selanjutnya, titik akhir berikutnya harus dihitung. Ini dapat dievaluasi dengan mudah menggunakan rumus panjang orto R_{tahun} atau R_{xt} . Ini pada gilirannya mungkin terletak di tepi kanan atau atas jendela kliping. Gambar 9, dan 10 menyortir aspek-aspek yang disebutkan.



Gambar 10. Garis segmen masuk ke jendela kliping dengan kemungkinan kasus yang berbeda.

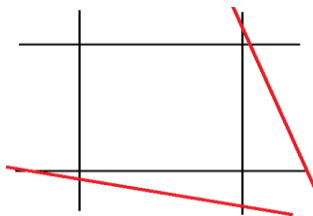
5.2. Garis Lurus dengan Kemiringan Negatif

Garis dengan kemiringan negatif telah ditemukan analog dengan kemiringan positif. Pada kasus ini $x_{akhir} < x_{mulailah}$. Nilai R_{kamu} dan R_{tahun} yang telah dihitung sebelumnya telah disimpan untuk perhitungan lebih lanjut. Prosedur berikut membantu dalam: penentuan keberadaan bagian garis yang ada di dalam jendela.

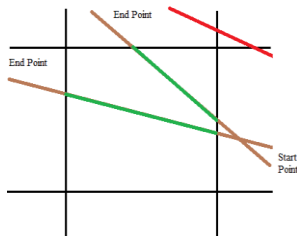
Jika $kamu_{bawah} < R_{tahun} < y_{puncak}$, kemudian garis memotong tepi kanan jendela kliping pada titik (x_{baik}, R_{kamu}) . Titik akhir berikutnya yang dievaluasi mungkin terletak di tepi kiri atau atas jendela kliping. Ini dapat dievaluasi menggunakan rumus panjang orto R_{yl} atau R_{xt} .

Jika $R_{kamu} < y_{puncak}$, maka titik akhir terakhir adalah (x_{kiri}, R_{yl}) lain $(R_{xt}, kamu_{puncak})$ dan bagian dari garis yang terlihat dapat dengan mudah digambar menggunakan dua titik akhir ini.

Jika $R_{tahun} < kamu_{bawah}$ kemudian garis masuk ke dalam jendela pada titik $(R_{xb}, kamu_{bawah})$ dengan memotong tepi bawah. Jika $R_{kamu} \leq kamu_{atas}$, titik potong berikutnya adalah (x_{kiri}, R_{sayu}) . Kalau tidak, intinya adalah $(R_{xt}, kamu_{puncak})$.

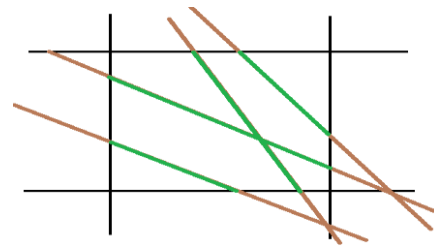


Gambar 11. Segmen garis yang dibuang dengan kemiringan negatif.



Gambar 12. Bagian dari segmen garis di dalam jendela kliping.

Jika $R_{tahun} \leq kamu_{atas}$, maka titik akhir kedua adalah (x_{baik}, R_{tahun}) . Kalau tidak, itu $(R_{xt}, kamu_{puncak})$ seperti yang disorot pada Gambar 12. Garis yang terlihat sebagian pada gilirannya dapat digambar menggunakan dua titik akhir ini.



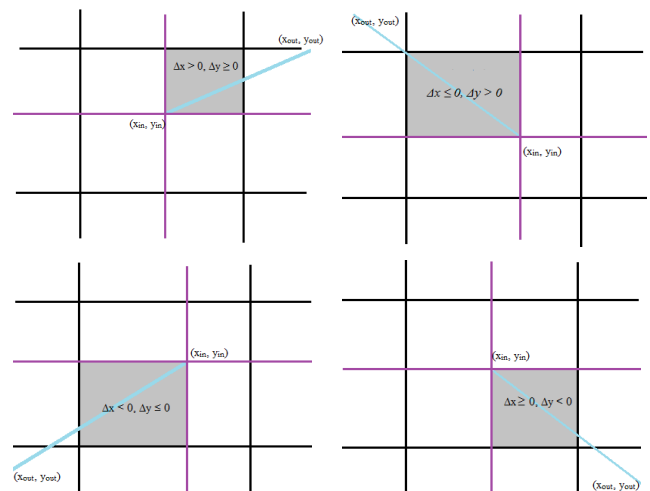
Gambar 13. Segmen garis membutuhkan jumlah pengujian yang lebih banyak.

Jika $R_{aku} < kamu_{bawah}$ lalu baris masuk ke dalam jendela di titik (R_{xb}, y_{bottom}) dengan memotong tepi bawah dan titik berikutnya adalah (x_{baik}, R_{tahun}) jika $R_{tahun} \leq kamu_{puncak}$ lain $(R_{xt}, kamu_{puncak})$.

5.3. Satu Garis Tepi ada Di Dalam Jendela Kliping

Dengan modifikasi kecil seperti yang dinyatakan sebelumnya, biarkan $(x_{di}, kamu_{di})$ menjadi salah satu titik akhir dari garis yang terletak di dalam jendela kliping dan $(x_{di luar}, kamu_{di luar})$ menjadi titik akhir lainnya yang terletak di luar jendela. Karena beberapa bagian dari garis terletak di dalam jendela, kemungkinan penolakan total berhenti ada. Untuk mengekstrak bagian dari garis, mendefinisikan kembali x $x_{di luar}$ x_{di} , $kamu$ $kamu_{di luar}$ y . Jendela kliping sekarang dibagi menjadi empat wilayah berdasarkan lokasi titik yang terletak di dalam jendela. Wilayah-wilayah ini disajikan pada Gambar (14) dan dapat diidentifikasi sebagai:

1. Daerah pojok kanan atas: $x > 0, y > 0$
2. Daerah pojok kiri atas: $x < 0, y > 0$
3. Daerah pojok kiri bawah: $x < 0, y < 0$
4. Daerah pojok kanan bawah: $x > 0, y < 0$



Gambar 14. Berbagai kemungkinan garis lurus keluar dari jendela kliping ke wilayah luar.

Bagian wilayah untuk mengevaluasi titik akhir lain yang terletak di tepi akan ditentukan dengan bantuan nilai positif dan negatif dari x dan y . Berbagai kendala yang perlu diperiksa dalam setiap kasus telah disorot dengan jelas dalam berbagai bagian Gambar 14.

Seperti yang didefinisikan sebelumnya, panjang orto adalah jarak, dari sumbu koordinat ke titik persimpangan antara tepi jendela kliping dan garis.

$$R_x \times \frac{2 \times \text{kamu} - \text{kamu}}{2 \text{ kamu}} \text{ dimana kamu} \frac{\text{kamu}}{\text{di luar kamu}} \text{ e} \quad (20)$$

$$R_{\text{kamu}} \frac{\text{kamu}}{\text{di luar}} \times \frac{2 \text{ kamu} \times x}{2 x} \text{ dimana } x \times \frac{x}{\text{di luar}} \times \text{tepi} \quad (21)$$

x dan y menunjukkan jarak horizontal dan vertikal dari titik yang terletak di luar jendela ke tepi persimpangan yang sesuai dari jendela kliping. Setelah mengidentifikasi daerah dengan bantuan kondisi pertidaksamaan, panjang orto yang sesuai R_x dan/atau R_{kamu} telah dievaluasi. Untuk garis yang terletak di daerah pojok kanan atas, titik terakhir adalah $(x_{\text{Baik}}, y_{\text{y}})$ jika $R_{\text{kamu}} \leq \text{kamu}_{\text{puncak}}$. Jika tidak, itu $(R_x, \text{kamu}_{\text{puncak}})$. Prosedur serupa dapat diterapkan untuk menghitung titik akhir lain dari sebuah garis untuk semua wilayah yang tersisa. Dalam hal ini, itu bisa dengan mudah mengamati bahwa maksimal dua perhitungan R_x dan R_{kamu} perlu dilakukan. Dalam skenario kasus terbaik, hanya satu di antara R_x dan R_{kamu} perlu dievaluasi.

6. Algoritma yang Diusulkan

Untuk mempertimbangkan berbagai kasus kliping garis yang disorot di bagian 5, satu set tiga algoritma matematika telah dirancang. Pilihan set algoritma yang akan diterapkan tergantung pada kendala puas dengan garis. Besaran R_{x1} , R_{x2} , R_{y1} , dan R_{y2} adalah panjang orto dan dihitung menggunakan rumus formula dijelaskan sebelumnya.

Algoritma 1: Pemilihan algoritma dan titik akhir untuk garis lurus

Dapatkan kliping tepi jendela x_{kiri} , x_{Baik} , $\text{kamu}_{\text{bawah}}$ dan $\text{kamu}_{\text{puncak}}$
Kumpulkan titik akhir segmen garis (x_1, kamu_1) , (x_2, kamu_2)

Jika $(y_1 - \text{kamu}_2)$

{

Tetapkan $(x_{\text{Mulailah}}, \text{kamu}_{\text{Mulailah}})$ (x_1, kamu_1)

Tetapkan $(x_{\text{akhir}}, \text{kamu}_{\text{akhir}})$ (x_2, kamu_2)

}

Lain

{

Tetapkan $(x_{\text{Mulailah}}, \text{kamu}_{\text{Mulailah}})$ (x_2, kamu_2)

Tetapkan $(x_{\text{akhir}}, \text{kamu}_{\text{akhir}})$ (x_1, kamu_1)

}

Simpan seluruh baris jika memenuhi pertidaksamaan {

$x_{\text{kiri}} \{x_{\text{Mulailah}}, x_{\text{akhir}}\} x_{\text{Baik}}$

$\text{kamu}_{\text{bawah}} y_{\text{Mulailah}} \text{ dan } \text{kamu}_{\text{akhir}} y_{\text{puncak}}$

}

Tolak seluruh baris jika memenuhi pertidaksamaan {

$\text{kamu}_{\text{akhir}} - \text{kamu}_{\text{bawah}} \text{ atau } \text{kamu}_{\text{mulai}} - \text{kamu}_{\text{puncak}}$

$\{x_{\text{Mulailah}}, x_{\text{akhir}}\} < x_{\text{kiri}} \text{ atau } \{x_{\text{Mulailah}}, x_{\text{akhir}}\} > x_{\text{Baik}}$

}

Jika kedua titik ujung berada di luar jendela persegi panjang

{ Terapkan algoritma kedua }

Lain

{ Terapkan algoritma ketiga }

Algoritma 2: Kedua Titik Akhir berada di Luar Jendela

Evaluasi R_{x1} , R_{x2} , R_{y1} dan R_{y2} pada waktu yang tepat

Kasus I: $x_{\text{Mulailah}} x_{\text{akhir}}$ (Kemiringan garis positif) Jika $(R$

$\text{saya} > \text{kamu}_{\text{puncak}})$

{

Tolak baris

}

Lain Jika $(R_{\text{kamu}} y_{\text{bawah}})$

{

Tetapkan titik pertama $(x_{\text{kiri}}, y_{\text{y}})$

Hitung titik berikutnya $(x_{\text{Baik}}, R_{\text{tahun}})$ atau $(R_{\text{xt}}, \text{kamu}_{\text{puncak}})$

}

Lain jika $(R_{\text{tahun}} < \text{kamu}_{\text{bawah}})$

{

Tolak baris

}

Lain

{

Tetapkan titik pertama $(R_{\text{db}}, \text{kamu}_{\text{bawah}})$

Jika $(R_{\text{tahun}} y_{\text{puncak}})$

{ Tetapkan titik terakhir $(x_{\text{Baik}}, R_{\text{tahun}})$ }

Lain

{ Tetapkan titik terakhir $(R_{\text{xt}}, \text{kamu}_{\text{puncak}})$ }

}

Kasus II: $x_{\text{mulai}} > x_{\text{akhir}}$ (Kemiringan garis negatif) Jika $(R_{\text{tahun}}$

$> \text{kamu}_{\text{puncak}})$

{

Tolak baris

}

Lain Jika $(R_{\text{tahun}} y_{\text{bawah}})$

{

Tetapkan titik pertama $(x_{\text{Baik}}, R_{\text{tahun}})$

Hitung titik terakhir $(x_{\text{kiri}}, y_{\text{y}})$ atau $(R_{\text{xt}}, \text{kamu}_{\text{puncak}})$

}

Lain jika $(R_{\text{aku}} < \text{kamu}_{\text{bawah}})$

{

Tolak baris

}

Lain

{

Tetapkan titik pertama $(R_{\text{db}}, \text{kamu}_{\text{bawah}})$

Jika $(R_{\text{kamu}} y_{\text{atas}})$

{

Tetapkan titik terakhir $(x_{\text{kiri}}, y_{\text{y}})$

}

Lain

{

Tetapkan titik terakhir $(R_{\text{xt}}, \text{kamu}_{\text{puncak}})$

}

}

Algoritma 3: Satu Titik Akhir Garis Di Dalam Jendela Persegi Panjang

Tetapkan $(x_{\text{di}}, \text{kamu}_{\text{di}})$ sebagai titik dalam Set $(x_{\text{di}} \text{ luar}, \text{kamu}_{\text{di}} \text{ luar})$ sebagai titik luar garis di daerah kliping

Simpan (Xdi, kamuDi)

hitung titik akhir lainnya dengan prosedur berikut

Jika ($x > 0$ && $y \geq 0$ && R_{tahun} yatas)

{

Tetapkan titik akhir lainnya (Xbaik, Rtahun)

}

Lain

{

Tetapkan titik akhir lainnya (Rxt, kamuPuncak)

}

Jika ($x \geq 0$ && $y > 0$ && R_{xt} Xkiri)

{

Tetapkan titik akhir lainnya (Rxt, kamuPuncak)

}

Lain

{

Tetapkan titik lain (Xkiri, Ryl)

}

Jika ($x < 0$ && $y \geq 0$ && R_{kamu} ybawah)

{

Tetapkan titik lain (Xkiri, Ryl)

}

Lain

{

Tetapkan titik lain (Rxb, kamuBawah)

}

Jika ($x \geq 0$ && $y < 0$ && R_{xb} Xbaik)

{

Tetapkan titik lain (Rxb, kamuBawah)

}

Lain

{

Tetapkan titik lain (Xbaik, Rtahun)

}

Setelah titik akhir dihitung dan jika bagian dari segmen garis berada di dalam, maka bagian dari garis dapat dengan mudah ditarik yang terletak di dalam jendela kliping. Pemilihan titik akhir untuk garis dalam algoritma ketiga sedikit berbeda dan dipilih sedemikian rupa sehingga satu titik berada di dalam jendela persegi panjang dan titik lain berada di luar di mana saja dalam sistem koordinat.

7. Analisis Kinerja

Algoritma yang diusulkan telah dibandingkan dengan algoritma yang ada di industri grafis. Berbagai kesimpulan dapat disimpulkan.

Telah diamati bahwa algoritma yang diusulkan sebagian besar bergantung pada perbandingan yang hanya membutuhkan satu siklus CPU selain dari beberapa perhitungan aritmatika.

Tidak seperti metode lain, tidak ada perhitungan yang diperlukan untuk mengevaluasi titik akhir dari garis tabungan yang terletak di dalam jendela kliping.

Semua algoritma yang ada saat ini menggunakan perhitungan floating point sedangkan algoritma yang diusulkan hanya bergantung pada nilai integral. Diamati bahwa perhitungan integral sepertiga kali lebih cepat daripada perhitungan floating point.

Kompleksitas waktu tidak berubah dengan ukuran jendela.

Algoritma LB mungkin lurus ke depan tetapi membutuhkan sejumlah besar perhitungan (perkalian dan pembagian) dibandingkan dengan metode lain.

CS tampaknya merupakan algoritma yang paling sederhana dan tertua tetapi jumlah perhitungan tumbuh secara eksponensial untuk titik-titik yang terletak di tepi. Selain itu, ada ambiguitas dalam pilihan atas atau bawah atau samping daerah.

Algoritma NLN adalah dikembangkan berdasarkan kemiringan perbandingan bahwa adalah disimpan untuk lebih lanjut perhitungan. Oleh karena itu, sejumlah besar memori diperlukan untuk menangani tingkat penyimpanan.

Dalam algoritma NLN, transformasi geometrik digunakan. Hal ini pada gilirannya mengarah pada kebutuhan akan pengetahuan di area tersebut pada awalnya untuk pemahaman dan kemudian untuk implementasi selanjutnya. Dalam algoritma NLN, transformasi geometri pertama harus diterapkan untuk memutar lokasi titik dan tiga lereng dibandingkan. Namun, dalam algoritma yang diusulkan, pertama R_{tahun} dibandingkan dengan pertidaksamaan $y_{bawah} \leq R_{tahun} y_{puncak}$ dan titik dipilih sebagai (Xbaik, Rtahun).

Perhitungan numerik dievaluasi dengan tiga

kasus yang berbeda.

1. Garis kemiringan positif.
2. Garis kemiringan negatif.
3. Bagian dari garis di dalam jendela selain dari garis yang diterima dan ditolak sepenuhnya.

Dimensi jendela persegi panjang, dalam analisis ini diambil sebagai (25, 25) dan (75, 75). yaitu, tepi jendela kliping adalah $x = 25$, $x = 75$, $y = 25$ dan $y = 75$. Hasil disajikan dalam tabel.

Tabel 1 menyajikan hasil untuk garis kemiringan positif, di mana garis-garis ini dapat dilewati jendela kliping baik dengan memotong tepi kliping kiri atau bawah. Hasil numerik untuk garis kemiringan negatif disajikan pada Tabel 2 dan bagian terpotong dari garis dan titik ujungnya dihitung. Tabel 3 menangani hasil untuk garis, di mana satu titik berada di dalam jendela kliping dan titik akhir lainnya berada di luar. Berdasarkan pengujian algoritme, beberapa baris ditolak dan di beberapa baris lainnya sebagian dari baris terpotong.

Tabel 1. Segmen garis lereng positif.

S. Tidak Ada Poin Pertama	Poin Terakhir	R_{kamu}	R_{tahun}	R_{xt}	Keputusan	Simpan Poin	
						Pertama Titik	Terakhir Titik
1	(15, 50)	(35, 150)	100	----	Ditolak	10	20
2	(00, 05)	(175, 40)	Ditolak			--	--
3	(20, 35)	(100,75)	38	63	----		(75, 63)
4	(15, 30)	(85, 100)	40	90	----		(60, 75)
5	(10, 10)	(110, 60)	18	43	40		(75,43)
6	(20, 05)	(65, 95)	15	115	30		(55, 75)

Tabel 2. Segmen garis kemiringan negatif.

S. Tidak Ada Poin Pertama	Poin Terakhir	R _{titik}	R _{titik}	R _{titik}	Keputusan	Simpan Poin	
						Poin pertama	Terakhir Titik
1	(100, 30)	(65, 100)	80	---	Ditolak	---	---
2	(110, 05)	(00, 27)	Ditolak	---	---	---	---
3	(90, 10)	(10, 90)	25	75	---	---	---
4	(90, 20)	(50, 100)	50	150	---	---	---
5	(80, 00)	(00, 80)	55	55	---	---	---
6	(75, 00)	(20, 110)	0	100	---	---	---

Tabel 3. Satu sisi ruas garis berada di dalam jendela kliping.

S. Tidak Ada Poin Pertama	Poin Terakhir	R _{titik}	R _{titik}	Wilayah	Simpan Poin	
					Di Titik	Keluar Titik
1	(50, 60)	(150, 80)	65	---	saya	(50, 60) (75, 65)
2	(40, 60)	(70, 90)	95	55	saya	(40, 60) (55, 75)
3	(50, 40)	(00, 90)	65	---	II	(50, 40) (25, 65)
4	(50, 50)	(20, 110)	100	38	II	(50, 50) (38, 75)
5	(50, 60)	(20, 30)	35	---	ARU ARU ARU	(50, 60) (25, 35)
6	(50, 50)	(30, 10)	00	38	ARU ARU ARU	(50, 50) (38, 25)
7	(40, 60)	(90, 35)	43	---	IV	(40, 60) (75, 43)
8	(40, 50)	(80, 10)	15	65	IV	(40, 50) (65, 75)

Referensi

- [1] Abdullah M., Pathan A., dan Al Shaikhli I., "Pendekatan Berbasis Web dan Perangkat Lunak Memadukan Jejaring Sosial untuk Pembelajaran Bahasa Arab Al-Qur'an Online," *Jurnal Teknologi Informasi Arab Internasional*, jilid 14, tidak. 1, hal. 80-90, 2017.
- [2] Andreev R. dan Sofianska E., "Algoritma Baru untuk Kliping Garis Dua Dimensi," *Komputer dan Grafik*, jilid 15, tidak. 4, hlm. 519-526, 1991.
- [3] Bui D. dan Skala V., "Algoritma Cepat untuk Memotong Garis dan Segmen Garis di E₂," *Komputer Visual*, jilid 14, tidak. 1, hlm. 31-38, 1998.
- [4] Chen X., Yong J., Wang G., Paul J., dan Xu G., "Menghitung Jarak Minimum antara Titik dan Kurva NURBS," *Desain dengan bantuan komputer*, jilid 40, tidak. 10, hlm. 1051-1054, 2008.
- [5] Cyrus M. dan Beck J., "Kliping Dua dan Tiga Dimensi Umum," *Komputer dan Grafik*, jilid 3, tidak. 1, hlm. 23-28, 1978.
- [6] Hari J., "Algoritma Kliping Garis Dua Dimensi Baru untuk Jendela Kecil," *Forum Grafik Komputer*, jilid 11, tidak. 4, hlm. 241-245, 1992.
- [7] Hari J., "Algoritma untuk Memotong Garis dalam Objek dan Ruang Gambar," *Komputer dan Grafik*, jilid 16, tidak. 4, hlm. 421-426, 1993.
- [8] Dévai F., "Analisis Algoritma Nicholl-Lee-Nicholl," *dalam Prosiding Konferensi Internasional tentang Ilmu Komputasi dan Aplikasinya*, Singapura, hal. 726-736, 2005.
- [9] Foley J. dan Van Dam A., *Dasar-dasar Grafik Komputer Interaktif*, MA: Addison Wesley, 1982.
- [10] Bruto M. dan Pfister H., *Grafik Berbasis Titik*, Morgan Kaufmann, 2011.
- [11] Hearn D. dan Baker P., *Grafik Komputer*, Prentice Hall, 1997.
- [12] Huang W., "Algoritma Kliping Garis Transformasi Affine untuk Poligon," *dalam Prosiding Konferensi Internasional tentang Intelijen Komputasi*, Nanning, hlm. 55-60, 2013.
- [13] Hughes J. dan Foley J., *Grafik Komputer: Prinsip dan Praktik*, pendidikan pearson, 2013.
- [14] Klette R. dan Rosenfeld A., *Geometri Digital: Metode Geometri untuk Analisis Gambar Digital*, Morgan Kaufmann, 2004.
- [15] Liang Y. dan Barsky B., "Konsep dan Metode Baru untuk Kliping Garis," *Transaksi ACM pada Grafik*, jilid 3, tidak. 1, hlm. 1-22, 1984.
- [16] Liu Y., Wang X., Bao S., Gomboši M., dan alik B., "Algoritma untuk Kliping Poligon, dan untuk Menentukan Persimpangan dan Persatuan Poligon," *Komputer dan Geosains*, jilid 33, tidak. 5, hlm. 589-598, 2007.
- [17] Lu G., Wu X., dan Peng Q., "Algoritma Kliping Garis yang Efisien Berdasarkan Penolakan Garis Adaptif," *Komputer dan Grafik*, jilid 26, tidak. 3, hlm. 409-415, 2002.
- [18] Mamatha E., Reddy C., dan Anand S., "Komputasi Titik Fokus dan Homogen Transformasi Geometris untuk Kurva Linear," *Perspektif dalam Sains*, jilid 8, hlm. 19-21, 2016.
- [19] Mamatha E., Reddy C., dan Prasad K., "Plotting Pikel Digital Anti Aliased untuk Garis Pemindaian Raster Menggunakan Evaluasi Area," *dalam Proceedings of Emerging Research in Computing, Information, Communication and Applications*, Singapura, hlm. 461-468, 2016.
- [20] Kumar M., Mamatha V., Reddy C., Mukesh V., dan Reddy R., "Penyembunyikan Data dengan Gambar Reversibel Berbasis Ganda Menggunakan Teknik Sudoku," *dalam Prosiding Konferensi Internasional tentang Kemajuan Komputasi, Komunikasi dan Informatika*, Udupi, hal. 2166-2172, 2017.
- [21] Nicholl T., Lee D., dan Nicholl R., "Algoritma Baru yang Efisien untuk Kliping Garis 2-D: Pengembangan dan Analisisnya," *ACM SIGGRAPH Grafik Komputer*, jilid 21, tidak. 4, hal.253-262, 1987.
- [22] Reddy C., Janani B., Narayanan S., dan Mamatha E., "Memperoleh Deskripsi untuk Gambar Sederhana menggunakan Teknik Realisasi Permukaan dan Pemrosesan Bahasa Alami," *Jurnal Sains dan Teknologi India*, jilid 9, tidak. 22, hlm. 1-7, 2016.
- [23] Rogers D. dan Rybak L., "Pada Algoritma Kliping Garis Umum yang Efisien," *Grafik Komputer dan Aplikasi*, jilid 5, tidak. 1, hal.82-86, 1985.
- [24] Schneider P. dan Eberly D., *Alat Geometris untuk Grafik Komputer*, Morgan Kaufmann, 2002.
- [25] Skala V., "Pendekatan Baru pada Kliping Garis dan Segmen Garis Secara Homogen

- Koordinat," *Komputer Visual*, jilid 21, tidak. 11, hlm. 905-914, 2005.
- [26] Slater M. dan Barsky B., "Garis 2D dan Kliping Poligon Berdasarkan Subdivisi Ruang," *Komputer Visual*, jilid 10, tidak. 7, hlm. 407-422, 1994.
- [27] Wang J., Cui C., dan Gao j., "Efisien Algoritma untuk Operasi Kliping Berdasarkan Jerat Trapesium dan Garis Sapu Teknik," *Kemajuan dalam Perangkat Lunak Rekayasa*, jilid 47, tidak. 1, hlm. 72-79, 2012.
- [28] Weng L., Daman D., dan Rahim M., "Pengecoran Bayangan dengan Penyangga Stensil untuk Rendering Real-Time," *Jurnal Teknologi Informasi Arab Internasional*, jilid 5, tidak. 4, hal. 102-110, 2008.
- [29] Wu Q., Huang X., dan Han Y., "Algoritma kliping untuk Segmen Parabola terhadap Jendela Sirkular," *Komputer dan Grafik*, jilid 30, tidak. 4, hlm. 540-560, 2006.
- [30] Zhang M. dan Sabharwal C., "Implementasi Efisien Algoritma Kliping Garis Parametrik dan Poligon," *dalam Prosiding Simposium ACM tentang Komputasi Terapan*, Madrid, hal. 796-800, 2002.



Mamatha Elliriki aku s bekerja di Universitas GITAM, Bengaluru Kampus. Dia saat ini sedang mengerjakan model grafis, digital geometri, kinerja berbagai sistem pemrosesan. Dia menerbitkan banyak artikel penelitian di

jurnal peer review dan diindeks dan konferensi internasional.



Chandrasekhara reddy memiliki menerbitkan beberapa makalah ilmiah di jurnal dan konferensi bereputasi. Dia bekerja sebagai profesor di Institut Teknologi Cambridge-Kampus Utara, Bangalore. Bidang penelitiannya saat ini meliputi

analisis kinerja model Markov.



Krishna Anand aku s saat ini bekerja sebagai profesor dalam Departemen Ilmu Komputer di Institut Sains dan Teknologi Sreenidhi, Hyderabad. Dia memiliki pengalaman mengajar yang kaya di berbagai universitas selain industri the

pengalaman. Bidang minatnya meliputi Simulasi, Komputasi Lunak, Grafik, dan Sistem Pakar.