

Daftar Isi

I	Konsep Dasar Grafik Komputer	1
1	Pendahuluan	5
1.1	Sejarah Perkembangan	6
1.2	Graphics Rendering Pipeline	7
1.3	Rendering Primitives	9
1.4	Algorithms	9
1.5	Application Programming Interfaces	10
2	Devices dan Device Independence	11
2.1	Calligraphic dan Raster Devices	11
2.2	Cara Kerja Monitor	11
3	Clipping	15
3.1	Outcode Clipping Garis	16
3.1.1	Algoritma Cohen Sutherland	17
3.2	Cyrus-Beck/Liang-Barsky Parametric Line Clipping-1	17
4	Geometri	19
4.1	Ruang Vektor	19
4.2	Ruang Affine	21
4.3	ruang Euclid	21
4.4	Ruang Koordinat	22
4.5	Ruang Affine dan Transformasi	22
4.6	Transformasi Geometri	22
4.6.1	Translasi	22
4.6.2	Skala dari Titik Pusat (0,0,0)	23
4.6.3	Rotasi	24
4.6.4	Komposisi Transformasi	24

DAFTAR ISI

4.7 Transformasi 3 Dimensi 25

Daftar Gambar

1.1	Model 3D Human sebagai awal Grafik Komputer	5
1.2	Sketchpad ditemukan oleh Sutherland	6
1.3	Model Pipelined Rendering	8
2.1	Beam Monitor	13
3.1	Segmen Garis	15
3.2	Pembagian 9 daerah clipping	16
3.3	Algoritma Sutherland Line Clipping	17
3.4	Menghitung Irisan	18
4.1	Translasi $T(2,2,0)$	22
4.2	Skala $S(3,2)$	23
4.3	Rotasi sejauh θ derajat denga pusat di $(0,0,0)$	24
4.4	Komposisi Transformasi untuk Rotasi sejauh θ derajat denga pusat di $P(x,y)$	25

Bagian I

Konsep Dasar Grafik Komputer

Tujuan Instruksional Khusus:

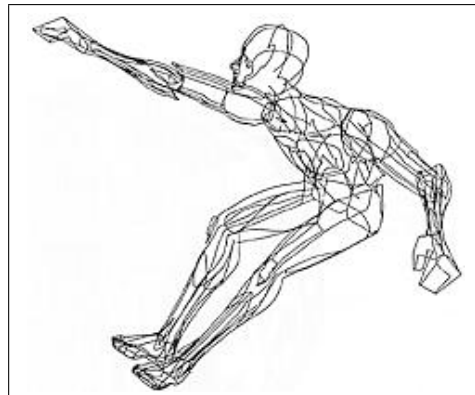
- Memahami konsep dasar pengertian Citra
- Memahami Sistem Visual Manusia
- Mengetahui Jenis Citra
- Mengetahui Proses Pengolahan Citra dan Implementasi Pengolahan Citra
- Mengetahui Konsep Pewarnaan

Bab 1

Pendahuluan

Grafik komputer adalah suatu proses pembuatan, penyimpanan dan manipulasi model dan citra. Model berasal dari beberapa bidang seperti fisik, matematik, artistik dan bahkan struktur abstrak.

Istilah "grafik komputer" ditemukan pada tahun 1960 oleh William Fetter yang pada waktu itu beliau akan menjelaskan suatu metoda baru di perusahaan Boeing yaitu pembentukan disain model cockpit dengan menggunakan pen plotter dan referensi model tubuh manusia 3 Dimensi (seperti gambar 1.1).



Gambar 1.1: Model 3D Human sebagai awal Grafik Komputer

Pengguna mengendalikan isi, struktur dan kemunculan objek dan menampilkan citra melalui suatu komponen dasar visual feedback dari suatu sistem masukan grafik interaktif (mis.: mouse, tablet dan stylus, peralatan force feedback, scanner, live video stream, dll) serta penyimpanan dan menampilkan/sebagai

output (mis.: layar, printer berbasis kertas, perekam video, non-linear editor, dll). Sistem interaktif grafik pertama, Sketchpad (gambar 1.2), ditemukan oleh Ivan Sutherland di MIT pada tahun 1963 sebagai hasil disertasi Phd-nya.



Gambar 1.2: Sketchpad ditemukan oleh Sutherland

1.1 Sejarah Perkembangan

Awal tahun 60-an dimulainya model animasi dengan menampilkan simulasi efek fisik. Pada tahun 1961, Edward Zajac menyajikan suatu model simulasi satelit dengan menggunakan teknologi Grafik Komputer. Pada tahun 1963 ditemukannya Sutherland (MIT), Sketchpad (manipulasi langsung, CAD), Alat untuk menampilkan Calligraphic (vector), penemuan mouse oleh Douglas Englebert. Tahun 1968 ditemukan Evans & Sutherland. Journal SIGGRAPH pertama diterbitkan pada tahun 1969.

Perkembangan teknologi grafik komputer pada akhir tahun 60-an hingga 70-an adalah:

- 1970: Pierre Bézier mengembangkan kurva Bézier.
- 1971: ditemukan Gouraud Shading,
- 1972: Ditayangkannya film Westworld, sebagai film pertama yang menggunakan animasi komputer.
- 1974: Ed Catmull develops z-buffer (Utah). First Computer Animated Short, Hunger: Keyframe animation and morphing

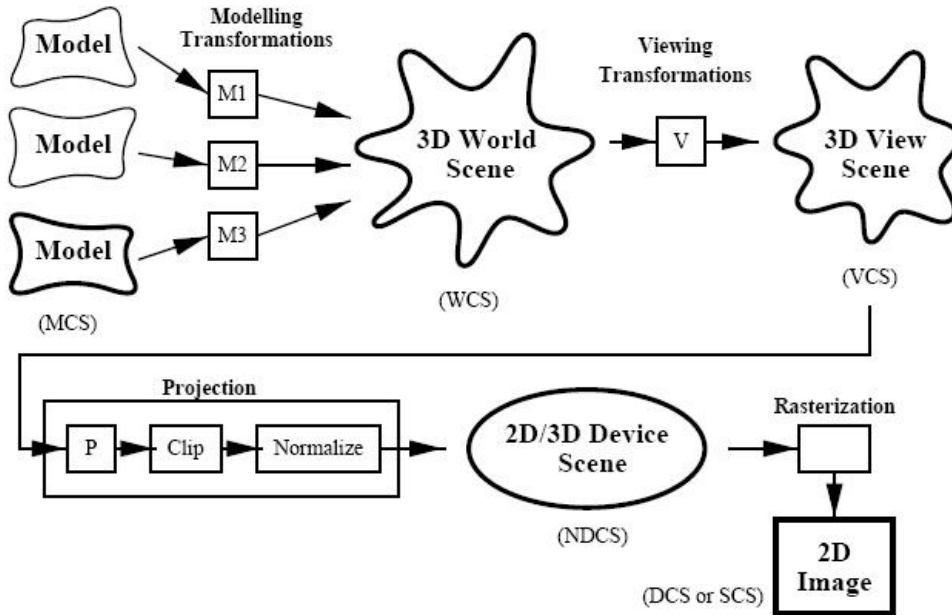
1.2. GRAPHICS RENDERING PIPELINE

- 1976: Jim Blinn mengembangkan texture dan bump mapping.
- 1977: Film terkenal Star Wars menggunakan grafik komputer
- 1979: Turner Whitted mengembangkan algoritma ray tracing. untuk pesawat Death Star.
- Pertengahan tahun 70-an hingga 80-an: Pengembangan Quest for realism radiosity sebagai main-stream aplikasi realtime.
- 1982: Mengembangkan teknologi grafik komputer untuk menampilkan partikel.
- 1984: Grafik Komputer digunakan untuk menggantikan model fisik pada film The Last Star Fighter.
- 1986: Pertama kalinya Film hasil produksi grafik komputer dijadikan sebagai nominasi dalam Academy Award: Luxo Jr. (Pixar).
- 1989: Film Tin Toy (Pixar) memenangkan Academy Award.
- 1995: Diproduksi film Toy Story (Pixar dan Disney) sebagai film 3D animasi panjang pertama.

Pada akhir tahun 90-an, ditemukannya teknologi visualisasi interaktif untuk ilmu pengetahuan dan kedokteran, artistic rendering, image based rendering, path tracing, photon maps, dll. Pada tahun 2000 ditemukannya teknologi perangkat keras untuk real-time photorealistic rendering.

1.2 Graphics Rendering Pipeline

Rendering adalah proses konversi dari scene ke suatu citra (gambar). Scene terdiri dari model dalam ruang sistem 3 dimensi. Model terdiri dari model primitif yang didukung oleh sistem render. Model dapat dibuat oleh tangan ataupun perangkat lunak 3D modelling. Citra/gambar dapat ditampilkan di layar monitor, dicetak di printer laser, atau ditulis ke suatu raster memory. Proses dari "model" ke "scene" dan ke "citra" dipecah menjadi proses yang lebih kecil lagi dan disebut sebagai *graphics pipeline*. Bagian dari pipeline proses sering diimplementasikan ke dalam perangkat keras grafik untuk meningkatkan kecepatan. Dukungan percepatan perangkat keras dimulai untuk mendukung programmable pipelines.



Gambar 1.3: Model Pipelined Rendering

Dalam suatu programmable pipeline, beberapa tahapan pipeline berada di unit, sehingga kita dapat mengimplementasikan model penyinaran sendiri ataupun transformasi geometrik. Programmable graphics accelerator dikenal sebagai GPU adalah merupakan teknik untuk mempercepat proses menampilkan grafik. Forward projection pipeline dasar dapat dilihat pada gambar 1.3

Setiap tahapan memperbaiki scene, dan mengkonversi model primitif 3D objek pada ruang model ke primitive di ruang device, dimana objek-objek tersebut dikonversikan ke pixel (*rasterized*). Beberapa sistem koordinat yang digunakan adalah:

- Memodelkan Sistem Koordinat (Modelling Coordinate System **MCS**).
- World Coordinate System **WCS**.
- VCS: Viewer Coordinate System.
- NDCS: Normalized Device Coordinate System.

1.3. RENDERING PRIMITIVES

- DCS or SCS: Device Coordinate System or equivalently the Screen Coordinate System.

Transformasi antara dua sistem koordinat menggunakan operasi perkalian matriks. Informasi tambahan juga dapat digunakan seperti pencahayaan dan bayangan serta beberapa model dasar dapat dihilangkan (disembunyikan) ataupun dimodifikasi (clipping).

1.3 Rendering Primitives

Models are composed of, or can be converted to, a large number of geometric primitives. Typical rendering primitives directly supported in hardware include:

- Points (single pixels)
- Line Segments
- Polygons (perhaps only convex polygons or triangles).

Modelling primitives include these, but also

- Piecewise polynomial (spline) curves
- Piecewise polynomial (spline) surfaces
- Implicit surfaces (quadrics, blobbies, etc)

Software renderer may support modelling primitives directly, or may convert them into polygonal or linear approximations for hardware rendering.

1.4 Algorithms

A number of basic algorithms are needed:

- Transformation: Convert representations of models/primitives from one coordinate system to another.
- Clipping/Hidden Surface Removal: Remove primitives and parts of primitives that are not visible on the display.

- Rasterization: Convert a projected screen-space primitive to a set of pixels.
- Picking: Select a 3D object by clicking an input device over a pixel location.
- Shading and Illumination: Simulate the interaction of light with a scene.
- Animation: Simulate movement by rendering a sequence of frames.

1.5 Application Programming Interfaces

X11: 2D rasterization.

PostScript, PDF: 2D transformations, 2D rasterization

Phigs+, GL, OpenGL, Direct3D: 3D pipeline

APIs provide access to rendering hardware via a conceptual model. APIs hide which graphics algorithms are or are not implemented in hardware by simulating missing pieces in software. For 3D interactive applications, we might modify the scene or a model directly or just the viewing information. After each modification, usually the image needs to be regenerated. We need to interface to input devices in an event-driven, asynchronous, and device independent fashion. APIs and toolkits are also defined for this task; we will be using X11 through Tcl/Tk.

Bab 2

Devices dan Device Independence

2.1 Calligraphic dan Raster Devices

Calligraphic Display Devices draw polygon and line segments directly:

- Plotters
- Direct Beam Control CRTs
- Laser Light Projection Systems

Raster Display Devices represent an image as a regular grid of samples.

- Each sample is usually called a pixel
- Both are short for picture element.
- Rendering requires rasterization algorithms to quickly determine a sampled representation of geometric primitives.

2.2 Cara Kerja Monitor

Raster Cathode Ray Tubes (CRTs) most common display device , yang dapat:

- Capable of high resolution.

- Good color fidelity.
- High contrast (100:1).
- High update rates.

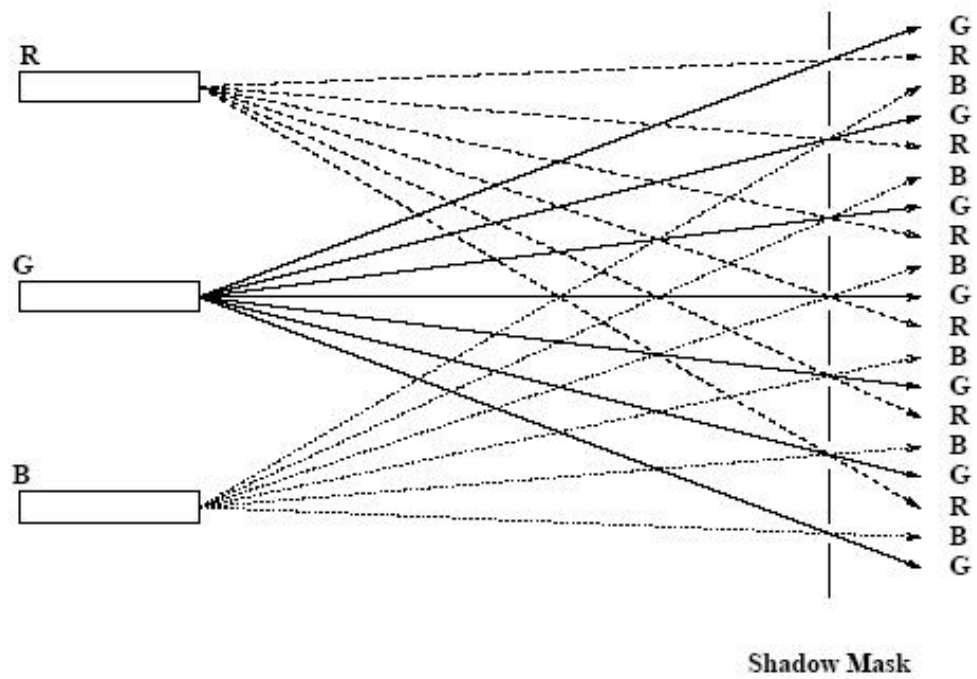
Electron beam scanned in regular pattern of horizontal scanlines. Raster images stored in a frame buffer. Frame buffers composed of VRAM (video RAM). VRAM is dual-ported memory capable of:

- Random access
- Simultaneous high-speed serial output: built-in serial shift register can output entire *scanline* at high rate synchronized to pixel clock.

Intensity of electron beam modified by the pixel value. Burst-mode DRAM replacing VRAM in many systems.

Colour CRTs have three different colours of phosphor and three independent electron guns. Shadow Masks allow each gun to irradiate only one colour of phosphor (gambar 2.1).

2.2. CARA KERJA MONITOR



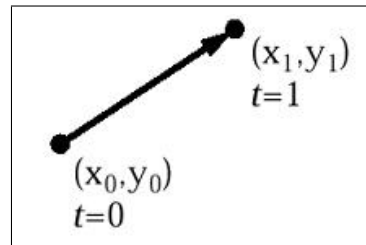
Gambar 2.1: Beam Monitor

BAB 2. DEVICES DAN DEVICE INDEPENDENCE

Bab 3

Clipping

Clipping adalah proses mengabaikan bagian objek yang berada di luar daerah yang akan ditampilkan (diluar window). Jenis clipping terdiri dari: clipping titik, clipping garis. Clipping titik adalah menghilangkan titik-titik yang berada di luar window, yaitu dengan menentukan apakah suatu titik berada dalam satu daerah atau tidak. Clipping garis menghilangkan bagian segmen garis yang berada di luar window. Segmen garis dapat melewati daerah batas. Bentuk parameter segmen garis adalah: 3.1



Gambar 3.1: Segmen Garis

$$\begin{aligned} X &= x_0 + t(x_1 - x_0) \\ Y &= y_0 + t(y_1 - y_0) \\ &0 \leq t \leq 1 \\ P(t) &= P_0 + t(P_1 - P_0) \end{aligned}$$

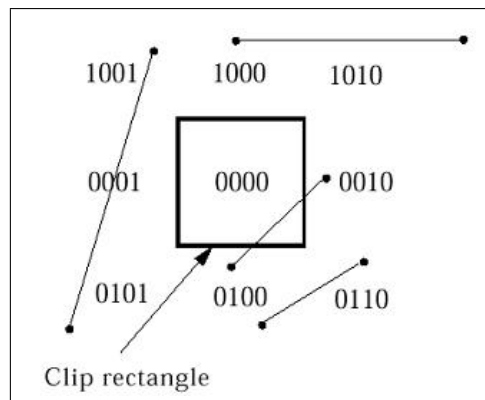
P_0 dan P_1 bukan merupakan berada pada titik yang sama. Untuk $t \in \mathfrak{R}$, $L(t)$ mendefinisikan suatu bentuk garis yang tak berhingga, sedangkan untuk $t \in [0, 1]$, $L(t)$ mendefinisikan suatu segmen garis dari P_0 ke P_1 . Persamaan

tersebut baik untuk membangkitkan titik-titik pada suatu garis tetapi tidak untuk pengujian apakah suatu titik berada pada atau di luar suatu garis.

3.1 Outcode Clipping Garis

Membagi bidang ke dalam sembilan daerah seperti pada gambar 3.2. Kesembilan daerah tersebut ditunjukkan oleh 4 bit outcodes, yaitu:

- Bit pertama: merupakan daerah luar setengah bidang (halfplane) dari bagian sisi atas (top edge), di atas (top edge)
- Bit kedua: merupakan daerah luar setengah bidang (halfplane) dari sisi bawah (bottom edge), di bawah bottom edge
- Bit ke tiga: merupakan daerah luar setengah bidang (halfplane) dari sisi kanan (right edge), ke kanan dari (right edge)
- Fourth bit: merupakan daerah luar setengah bidang (halfplane) dari sisi kiri (left edge), ke sebelah kiri dari (left edge)



Gambar 3.2: Pembagian 9 daerah clipping

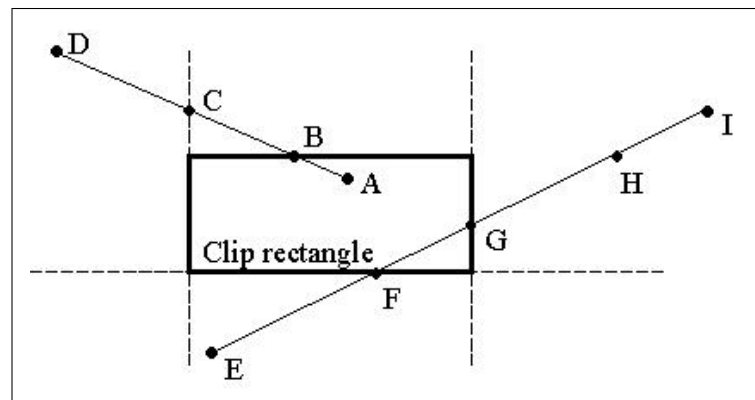
Garis dengan $OC_0 = 0$ dan $OC_1 = 0$ secara trivial diterima. Sedangkan garis yang berada seluruhnya di halfplane pada outside suatu sisi dapat secara trivial ditolak, yaitu $OC_0 \neq 0$ dan $OC_1 \neq 0$ (mereka membagi (share) suatu "outside" bit).

3.2. CYRUS-BECK/LIANG-BARSKY PARAMETRIC LINE CLIPPING-1

3.1.1 Algoritma Cohen Sutherland

Apabila suatu garis bukan trivial diterima maupun bukan ditolak, maka perlu melakukan proses divide and conquer, yaitu membagi garis ke dua segmen, maka dapat menjadi T/A atau T/R atau kedua segmen:

- use a clip edge to cut the line
- use outcodes to choose edge that is crossed: if outcodes differ in the edges bit, endpoints must straddle that edge
- pick an order for checking edges: top bottom right left
- compute the intersection point; the clip edge fixes either x or y, can be substituted into the line equation
- iterate for the newly shortened line "extra" clips may happen, e.g., E-I at H (Gambar 3.3)

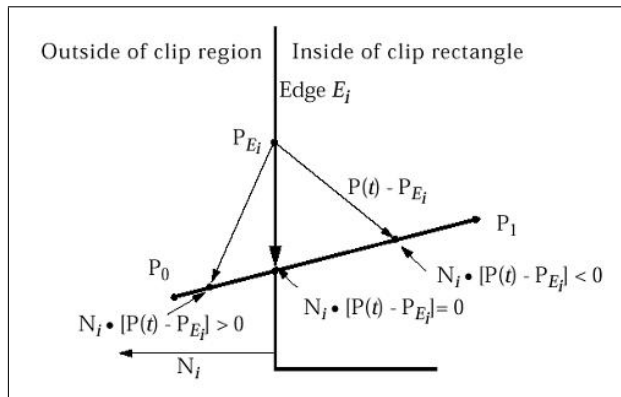


Gambar 3.3: Algoritma Sutherland Line Clipping

3.2 Cyrus-Beck/Liang-Barsky Parametric Line Clipping-1

Menggunakan formulasi parametrik garis $P(t) = P_0 + t(P_1 - P_0)$ dan mencari empat t s untuk empat sisi clip, kemudian memutuskan bentuk mana yang merupakan irisan dan menghitung titik (x, y) untuk irisan tersebut (≤ 2).

Untuk setiap titik P_{E_i} pada sisi E_i ,



Gambar 3.4: Menghitung Irisan

Bab 4

Geometri

Geometry provides a mathematical foundation for much of computer graphics:

- Geometric Spaces: Vector, Affine, Euclidean, Cartesian, Projective.
- Affine Geometry
- Affine Transformations
- Perspective
- Projective Transformations
- Matrix Representation of Transformations
- Viewing Transformations
- Quaternions dan Orientation

4.1 Ruang Vektor

Definisi:

- Himpunan Vektor V .
- Terdapat dua operasi untuk setiap vektor $\vec{u}, \vec{v} \in V$:
 - Penjumlahan: $\vec{u} + \vec{v} \in V$

BAB 4. GEOMETRI

- Perkalian Skalar: $\alpha \vec{u} \in V$ dimana α merupakan anggota dari Field F (misalkan bilangan Real).

Axioma:

- Addition Commutes: $\vec{u} + \vec{v} = \vec{v} + \vec{u}$
- Addition Associates: $(\vec{u} + \vec{v}) + \vec{w} = \vec{u} + (\vec{v} + \vec{w})$
- Scalar Multiplication Distributes: $\alpha(\vec{u} + \vec{v}) = \alpha \vec{u} + \alpha \vec{v}$
- Unique Zero Element: $\vec{0} + \vec{u} = \vec{u}$
- Field Unit Element: $1 \vec{u} = \vec{u}$

Span:

- Diberikan $\mathcal{B} = \{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n\}$.
- \mathcal{B} span \mathcal{V} jika dan hanya jika setiap $\vec{v} \in \mathcal{V}$ dapat ditulis $\vec{v} = \sum_{i=1}^n \alpha_i \vec{v}_i$
- Dinotasikan sebagai $\langle \mathcal{B} \rangle = \mathcal{V}$
- $\sum_{i=1}^n \alpha_i \vec{v}_i$ adalah kombinasi linier dari vektor di \mathcal{B}

Basis:

- Sembarang himpunan spanning minimal adalah suatu basis
- Semua basis berukuran sama.

Dimensi:

- Sejumlah vektor dalam suatu basis
- Dalam hal ini yang akan dibahas adalah ruang dimensi 2 dan dimensi 3

4.2 Ruang Affine

Definisi:

Himpunan vektor-vektor \mathcal{V} dan titik-titik \mathcal{P} . Vektor \mathcal{V} membentuk ruang vektor dan titik-titik dapat dikombinasikan dengan vektor untuk membentuk titik baru $P + \vec{v} \Rightarrow Q$ dimana $P, Q \in \mathcal{P}$ dan $\vec{v} \in \mathcal{V}$

Frame adalah suatu perluasan affine dari suatu basis. Merupakan suatu basis vektor ditambah suatu titik \mathcal{O} (titik pusat), yaitu $F = \vec{v}_1, \vec{v}_2, \dots, \vec{v}_n, \mathcal{O}$. Dimensi pada ruang affine adalah sama dengan dimensi dari ruang vektor.

4.3 ruang Euclid

Ruang Metrik adalah sembarang ruang dengan suatu distance (jarak) $d(P, Q)$ yang didefinisikan pada anggotanya. Suatu metric $d(P, Q)$ harus memenuhi aksioma berikut:

1. $d(P, Q) \geq 0$
2. $d(P, Q) = 0$ jika dan hanya jika $P = Q$
3. $d(P, Q) = d(Q, P)$
4. $d(P, Q) \leq d(P, R) + d(R, Q)$

Pada ruang Euclid metric berbasiskan pada dot (inner) product: $d^2(P, Q) = (P - Q) \cdot (P - Q)$

Dot product didefinisikan pada vektor, distance metric didefinisikan pada titik.

Sifat dari dot product:

$$\begin{aligned}(\vec{u} + \vec{v}) \cdot \vec{w} &= \vec{u} \cdot \vec{w} + \vec{v} \cdot \vec{w} \\ \alpha(\vec{u} \cdot \vec{v}) &= (\alpha \vec{u}) \cdot \vec{v} \\ &= \vec{u} \cdot (\alpha \vec{v}) \\ \vec{u} \cdot \vec{v} &= \vec{v} \cdot \vec{u}\end{aligned}$$

Norm :

$$|\vec{u}| = \sqrt{\vec{u} \cdot \vec{u}}$$

Sudut :

$$\cos(\angle \vec{u} \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{|\vec{u}| |\vec{v}|}$$

Tegak Lurus :

$$\vec{u} \cdot \vec{v} = 0 \Rightarrow \vec{u} \perp \vec{v}$$

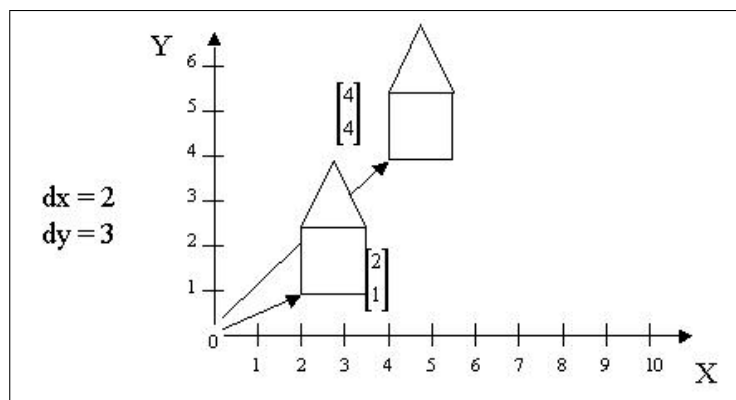
4.4 Ruang Koordinat

4.5 Ruang Affine dan Transformasi

4.6 Transformasi Geometri

Transformasi geometrik adalah memindahkan suatu objek secara geometri dengan cara mengalikan posisi objek dengan matriks transformasi.

4.6.1 Translasi



Gambar 4.1: Translasi $T(2,2,0)$

Translasi dispesifikasikan oleh suatu vektor $[\Delta x, \Delta y, 0]^T$ dimana 0 menyatakan titik pusat $(0,0,0)$ (gambar 4.1). Beberapa sifat dari proses translasi adalah:

- Suatu titik $[x, y, 1]^T$ akan dipetakan ke $[x + \Delta x, y + \Delta y, 1]^T$

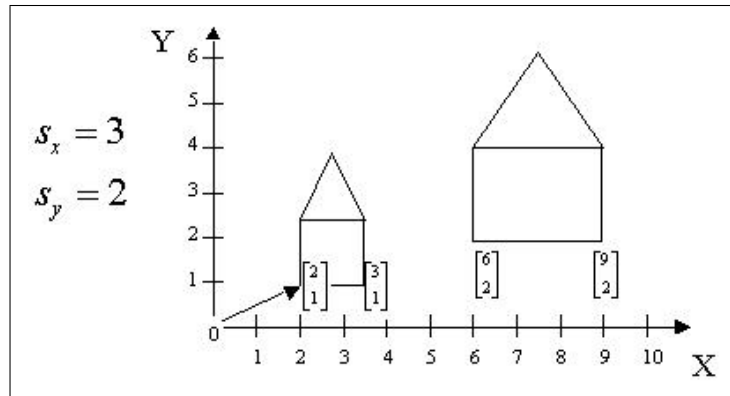
4.6. TRANSFORMASI GEOMETRI

- Suatu vektor tidak akan berubah setelah ditranslasi.
- Translasi adalah bukan suatu transformasi linear.

Bentuk matriks translasi adalah:

$$\overbrace{T(\Delta x, \Delta y)}^{\quad} \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + \Delta x \\ y + \Delta y \\ 1 \end{bmatrix}$$

4.6.2 Skala dari Titik P:usat (0,0,0)



Gambar 4.2: Skala S(3,2)

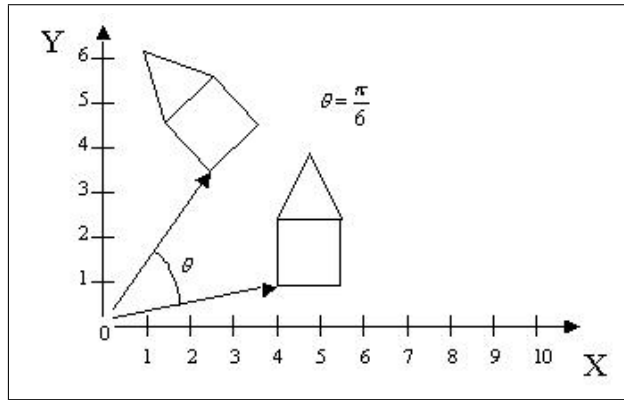
Melakukan skala suatu objek berdasarkan pada titik pusat (0,0,0). Dispefifikasikan oleh faktor $s_x, s_y \in \mathcal{R}$ (Gamabr 4.2). Sifat dari transformasi skala adalah:

- Diaplikasikan ke titik ataupun vektor dan bersifat linier
- Memetakan titik $[s_x, s_y, 1]^T$ ke titik $[s_x x, s_y y, 1]^T$
- Memetakan suatu vektor $[x, y, 0]^T$ ke vektor $[s_x x, s_y y, 0]^T$

Bentuk matriks skala adalah:

$$\overbrace{S(s_x, s_y)}^{\quad} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \text{ or } 1 \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \\ 0 \text{ or } 1 \end{bmatrix}$$

4.6.3 Rotasi



Gambar 4.3: Rotasi sejauh θ derajat dengan pusat di $(0,0,0)$

Rotasi adalah proses memutar objek dengan poros titik $(0,0,0)$ berlawanan arah jarum jam sejauh θ derajat (Gambar 4.3).

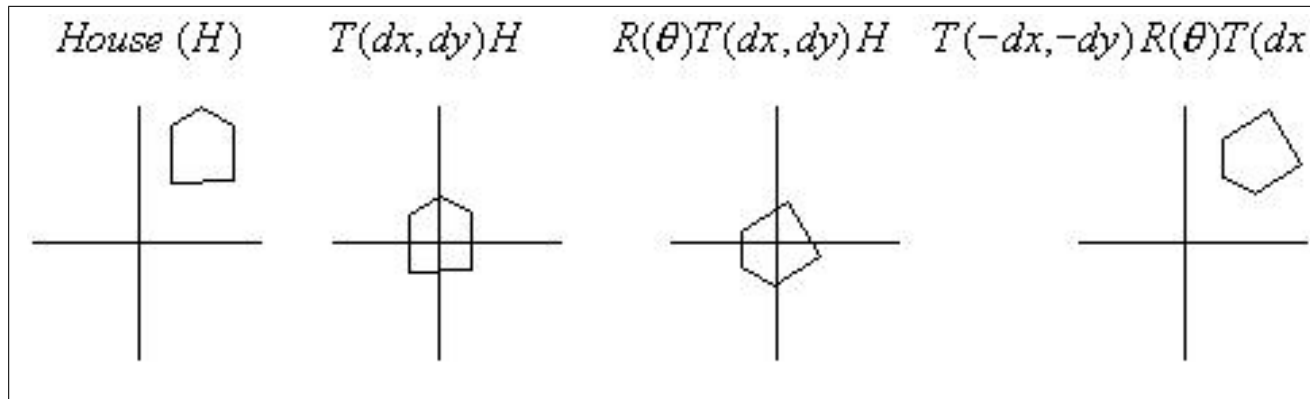
Bentuk matriks rotasi adalah:

$$\overbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{R(\theta)} \begin{bmatrix} x \\ y \\ 0 \text{ or } 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 0 \text{ atau } 1 \end{bmatrix}$$

4.6.4 Komposisi Transformasi

Komposisi transformasi adalah sederetan transformasi yang dilakukan pada suatu objek untuk menghasilkan suatu transformasi khusus. Berikut adalah contoh transformasi melakukan rotasi pada titik pusat suatu objek $P[x, y, 1]^T$ seperti pada gambar 4.4. Pertama objek ditranslasikan ke titik pusat $(0,0,0)$ dan kemudian dirotasi sejauh θ derajat dan kemudian ditranslasikan kembali sejauh translasi pertama tapi dengan arah kebalikannya. Urutan dari transformasi komposisi adalah penting, karena mempengaruhi hasil matriks transformasi komposisi. Pembentukan matriks transformasi komposisi contoh di atas adalah sebagai berikut:

4.7. TRANSFORMASI 3 DIMENSI



Gambar 4.4: Komposisi Transformasi untuk Rotasi sejauh θ derajat dengan pusat di $P(x,y)$

$$\begin{aligned}
 & T(x_0, y_0) \circ R(\theta) \circ T(-x_0, -y_0) \\
 = & \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \circ \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} \\
 = & \begin{bmatrix} \cos \theta & -\sin \theta & x_0(1 - \cos(\theta)) + y_0 \sin(\theta) \\ \sin \theta & \cos \theta & y_0(1 - \cos(\theta)) - x_0 \sin(\theta) \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

4.7 Transformasi 3 Dimensi

Transformasi untuk koordinat 3 dimensi. Prinsip dasar adalah sama dengan transformasi 2 dimensi, hanya ditambahkan satu sumbu koordinat, yaitu z . Perbedaan yang ada adalah pada transformasi rotasi, dimana dilakukan rotasi terhadap 3 sumbu.

Translasi $T(\Delta x, \Delta y, \Delta z)$

$$T(\Delta x, \Delta y, \Delta z) = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Skala $S(s_x, s_y, s_z)$

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotasi pada masing-masing sumbu x, y, z sejauh θ pada titik pusat $(0,0,0)$
 $R_x(\theta), R_y(\theta), R_z(\theta)$.

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$