

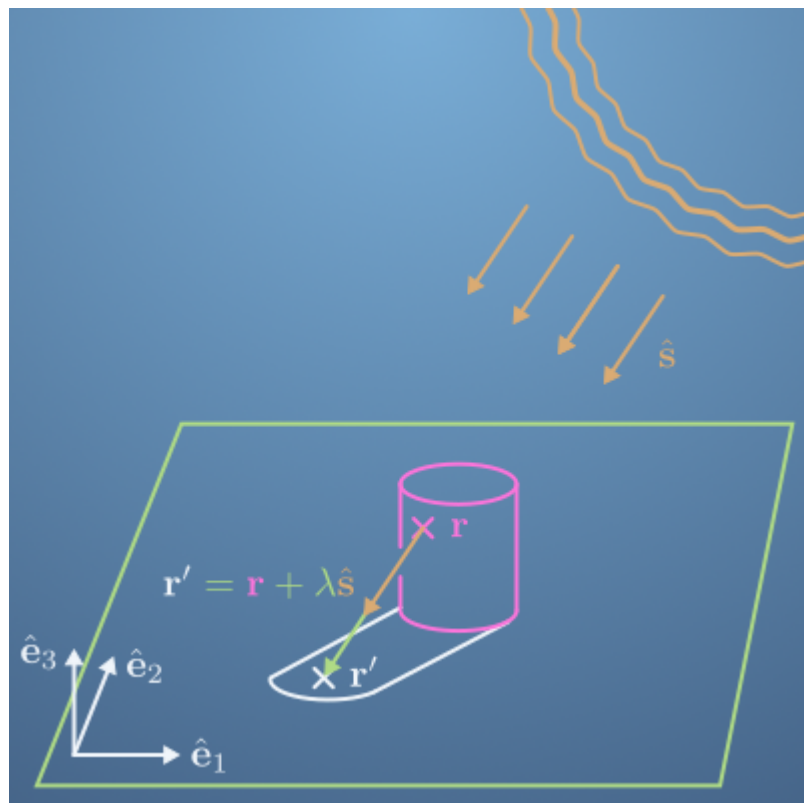
Master Guide: Linear Algebra - Shadow Projection & Einstein Summation

Dokumen ini adalah rangkuman komprehensif dari diskusi mengenai proyeksi bayangan, mulai dari intuisi fisik, penurunan rumus vektor, penerjemahan ke notasi Einstein, hingga implementasi matriks untuk komputasi grafis.

Catatan ini mencakup **Pertanyaan Kritis (Q&A)** yang muncul selama proses belajar untuk memastikan pemahaman fundamental yang kuat.

BAB 1: LOGIKA VEKTOR & SKENARIO FISIK (Soal No. 1)

1.1 Skenario Masalah: "Google Maps" Cahaya



Kita ingin mencari koordinat bayangan (\mathbf{r}') di lantai dari sebuah objek (\mathbf{r}) yang melayang di udara dan disinari matahari ($\hat{\mathbf{s}}$).

- **Aktor 1: \mathbf{r} (Posisi Benda)** → Titik asal keberangkatan.
- **Aktor 2: $\hat{\mathbf{s}}$ (Arah Sinar)** → Vektor satuan (panjang = 1) yang menunjukkan arah jalan.
- **Aktor 3: \mathbf{r}' (Posisi Bayangan)** → Titik tujuan di lantai.

1.2 Rumus Perjalanan (The Journey)

Secara vektor, perjalanan cahaya ditulis sebagai:

$$\mathbf{r}' = \mathbf{r} + \lambda \hat{\mathbf{s}}$$

Q: Apa itu λ (Lambda)? Kenapa harus ada?

A: \hat{s} itu cuma penunjuk arah (kompas), panjangnya cuma 1 langkah kecil. Kalau kita cuma jalan $\mathbf{r} + \hat{s}$, kita akan melayang di udara. λ (**Durasi/Skalar**) adalah angka pengali yang bilang: "*Ulangi langkah ini sebanyak λ kali sampai nabrak lantai.*"

Mari kita bedah pakai **ANGKA NYATA** biar langsung "klik" di otak.

1. Bedah Si \hat{s} (Vektor Arah)

Bayangkan sinar matahari nya datang miring.

Secara matematika, arah sinar (\hat{s}) itu punya 3 komponen: Kanan, Depan, dan Bawah.

Misalnya, kita pakai angka ngawur aja ya:

$$\hat{s} = \begin{bmatrix} 1 \\ 2 \\ -5 \end{bmatrix}$$

- **1:** Bergerak ke Kanan (Sumbu X)
 - **2:** Bergerak ke Depan (Sumbu Y)
 - **-5:** Bergerak ke Bawah (Sumbu Z / Vertikal)
-

2. Bedah Si s_3 (Komponen Vertikal)

Sekarang, apa itu s_3 ?

Di soal dibilang $s_3 = \hat{s} \cdot \hat{e}_3$.

Mari kita hitung pakai angka tadi:

$$s_3 = \begin{bmatrix} 1 \\ 2 \\ -5 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Cara hitungnya (Dot Product):

$$(1 \times 0) + (2 \times 0) + (-5 \times 1)$$

$$= 0 + 0 - 5$$

$$s_3 = -5$$

KESIMPULAN PENTING:

Lihat hasilnya! Hasilnya cuma angka -5.

Dia bukan kotak vektor lagi. Dia cuma Satu Angka (Skalar).

Jadi, s_3 itu artinya: "Kecepatan Turunnya doang berapa?" (Jawabannya: -5).

3. Bedah Si λ (Lambda)

Lambda itu Jumlah Langkah (Durasi). Ini juga cuma Angka (Skalar).

Misalnya, butuh 2 detik (atau 2 langkah) buat sampai lantai.

Jadi:

$$\lambda = 2$$

4. Bedah Hubungan λs_3

Nah, sekarang pertanyaan utamamu: "*Hubungan mereka apa?*"

Karena λ adalah angka (2), dan s_3 adalah angka (-5), maka:

$$\lambda s_3 = 2 \times (-5) = -10$$

Ini adalah **Perkalian Biasa** (Angka kali Angka). Bukan Dot Product.

Logika Ceritanya:

"Setiap 1 detik (s_3), ketinggian berkurang 5 meter. Kalau kita jalan selama 2 detik (λ), maka total ketinggian yang berkurang adalah 10 meter."

Ringkasan Visual

- \hat{s} (**Vektor**): Paket Komplit [Kanan, Depan, Bawah].
- s_3 (**Skalar**): Cuma ambil angka "Bawah"-nya saja. (Angka tunggal).
- λ (**Skalar**): Pengali / Durasi.
- λs_3 : Perkalian biasa. Artinya "Total Jarak Turun".

1.3 Syarat Pendaratan: Konsep "Altimeter" (\hat{e}_3)

Bagaimana matematika tahu kalau kita sudah sampai lantai? Definisi lantai adalah tempat di mana **Ketinggian** (z) = 0.

Kita menggunakan vektor basis vertikal $\hat{e}_3 = [0, 0, 1]$ sebagai alat ukur.

$$\mathbf{r}' \cdot \hat{e}_3 = 0$$

Q: Kenapa harus di-Dot Product dengan \hat{e}_3 ? Apa intuisinya? A: Dot product dengan basis ($\cdot \hat{e}_3$) berfungsi sebagai "**Saringan**" atau "**Proyeksi ke Tiang Bendera**".

- Kita tidak peduli posisi x (kiri-kanan) atau y (depan-belakang).
- Kita cuma mau tanya: "*Berapa tingginya?*"
- Dot product ini membuang info x dan y , menyisakan info z saja. Syarat sampai lantai adalah $z = 0$.

1.4 Menghitung Durasi Turun (λ)

Kita substitusi rumus perjalanan ke dalam saringan altimeter:

$$(\mathbf{r} + \lambda \hat{\mathbf{s}}) \cdot \hat{\mathbf{e}}_3 = 0$$

Pecah menjadi komponen-komponennya:

$$(\mathbf{r} \cdot \hat{\mathbf{e}}_3) + (\lambda \hat{\mathbf{s}} \cdot \hat{\mathbf{e}}_3) = 0$$

Di soal, $(\hat{\mathbf{s}} \cdot \hat{\mathbf{e}}_3)$ disingkat menjadi s_3 .

$$(\mathbf{r} \cdot \hat{\mathbf{e}}_3) + \lambda s_3 = 0$$

Q: Apa itu s_3 ? Apakah λ di-dot-kan dengan s_3 ? A:

- s_3 (**Kecepatan Turun**): Ini adalah Skalar (Angka tunggal). Didapat dari vektor $\hat{\mathbf{s}}$ yang diambil komponen z -nya saja (misal: -5 m/s).
- λs_3 (**Total Jarak Turun**): Ini adalah **Perkalian Biasa** (Angka \times Angka). Bukan dot product.
- Logikanya: (Tinggi Awal) + (Durasi \times Kecepatan Turun) = 0.

Hasil Akhir λ (Pindah Ruas): Kita pindahkan $(\mathbf{r} \cdot \hat{\mathbf{e}}_3)$ ke kanan (jadi negatif), lalu bagi dengan s_3 :

$$\lambda = -\frac{(\mathbf{r} \cdot \hat{\mathbf{e}}_3)}{s_3}$$

1.5 Rumus Vektor Final

Masukkan nilai λ kembali ke rumus perjalanan awal:

$$\mathbf{r}' = \mathbf{r} - \hat{\mathbf{s}} \frac{(\mathbf{r} \cdot \hat{\mathbf{e}}_3)}{s_3}$$

BAB 2: NOTASI INDEKS EINSTEIN (Soal No. 2)

Kita perlu mengubah rumus vektor di atas menjadi bentuk yang bisa diproses komputer per komponen (baris demi baris).

2.1 Konsep Dasar: Analogi Laci & Blender

A. Si i (Si Bos / Free Index)

- **Makna:** "Sekarang kita kerjakan Laci/Baris nomor berapa?"
- **Sifat:** Muncul sendirian di setiap suku. Tidak dijumlahkan.
- **Contoh:** r_i (Elemen vektor r pada baris i).

B. Si j (Si Kuli / Dummy Index)

- **Makna:** "Looping untuk menghitung total." (Khusus Dot Product).
- **Sifat:** Muncul **KEMBAR** (dua kali) dalam satu suku. Artinya: Kalikan lalu **JUMLAHKAN SEMUANYA** (Summation).
- **Contoh:** $a_j b_j$ artinya $(a_1 b_1 + a_2 b_2 + a_3 b_3)$.

Q: Kenapa Dot Product pakai j ? Kenapa gak pakai i aja? **A:** Kalau pakai i , komputer bingung! Huruf i sudah dipakai sebagai "Penunjuk Baris Utama" (Global Variable). Kalau kita pakai i lagi di dalam dot product (looping), terjadi **Tabrakan Variabel**. Kita butuh variabel baru (j) yang bekerja sebagai "Kuli" hitung cepat di dalam, tanpa mengganggu "Si Bos i " di luar.

2.2 Menerjemahkan Rumus Vektor ke Indeks

Rumus Vektor:

$$\mathbf{r}' = \mathbf{r} - \frac{\hat{s}}{s_3} (\mathbf{r} \cdot \hat{\mathbf{e}}_3)$$

Terjemahan:

1. Semua vektor bebas (\mathbf{r}' , \mathbf{r} , \hat{s}) diberi label i .
2. Dot product ($\mathbf{r} \cdot \hat{\mathbf{e}}_3$) diterjemahkan pakai j .

$$r'_i = r_i - \frac{s_i}{s_3} (r_j [\hat{\mathbf{e}}_3]_j)$$

BAB 3: DERIVASI MATRIKS & TRIK IDENTITAS

3.1 Masalah Faktorisasi (Si Jomblo)

Tujuan kita adalah membuat rumus: **Output = Matriks \times Input (r_j)**. Kita harus menarik keluar r_j .

Rumus saat ini:

$$r'_i = \underbrace{r_i}_{\text{Masalah}} - \underbrace{\frac{s_i [\hat{\mathbf{e}}_3]_j}{s_3} r_j}_{\text{Aman}}$$

Suku pertama (r_i) **JOMBLO**. Dia tidak punya r_j . Kita tidak bisa memfaktorkan r_j kalau ada suku yang tidak memilikinya.

3.2 Solusi: Trik "Ganti Baju" ($r_i = I_{ij}r_j$)

Kita gunakan Matriks Identitas (I) untuk memanipulasi bentuk tanpa mengubah nilai.

$$r_i = I_{ij}r_j$$

Q: Saya bener-bener gak ada gambaran, kenapa kok bisa gitu? A (Bukti Konkret):

Misal vektor $r = [2, 5, 9]$. Kita mau cari r_1 (isi laci 1, yaitu 2). Rumus $I_{1j}r_j$ artinya:

$(I_{11} \times r_1) + (I_{12} \times r_2) + (I_{13} \times r_3) = (1 \times 2) + (0 \times 5) + (0 \times 9) = 2$. **Terbukti:** Hasilnya balik lagi ke angka 2 (r_1). Trik ini sah!

3.3 Bentuk Akhir Matriks Transformasi

Substitusi r_i dengan $I_{ij}r_j$, lalu tarik r_j keluar:

$$r'_i = (I_{ij}r_j) - \frac{s_i[\hat{e}_3]_j}{s_3}r_j$$

$$r'_i = \left(I_{ij} - \frac{s_i[\hat{e}_3]_j}{s_3} \right) r_j$$

Isi dalam kurung adalah elemen Matriks A_{ij} . (Catatan: $[\hat{e}_3]_j$ sama dengan I_{3j}).

BAB 4: MEMBANGUN MATRIKS (Soal No. 3 & 4)

4.1 Logika Dimensi Matriks (2×3)

Q: Bukannya matriks itu input vs output? Kalau ada Z kenapa outputnya gak 3 baris? A:

- **Input (3 Kolom):** Benda asalnya 3D (x, y, z). Kita butuh 3 kolom untuk menampung x, y, z .
- **Output (2 Baris):** Soal minta bayangan di **LANTAI 2D** (x', y'). Kita buang sumbu z' karena nilainya pasti 0.
- Jadi dimensinya: **2 Baris (Output) \times 3 Kolom (Input)**.

4.2 Mengisi Nilai Matriks Row-by-Row

Rumus Elemen: $A_{ij} = I_{ij} - \frac{s_i[\hat{e}_3]_j}{s_3}$. Ingat: $[\hat{e}_3]_j$ hanya bernilai **1** saat $j = 3$ (kolom terakhir).

Baris 1 (Sumbu X', $i = 1$):

- Col 1 ($j = 1$): $1 - 0 = 1$
- Col 2 ($j = 2$): $0 - 0 = 0$
- Col 3 ($j = 3$): $0 - \frac{s_1(1)}{s_3} = -s_1/s_3$

Baris 2 (Sumbu Y', $i = 2$):

- Col 1 ($j = 1$): $0 - 0 = 0$

- Col 2 ($j = 2$): $1 - 0 = 1$
- Col 3 ($j = 3$): $0 - \frac{s_2(1)}{s_3} = -s_2/s_3 I_{23} - \frac{s_2(1)}{s_3} = -s_2/s_3$ **Baris 3 (Sumbu Z', $i = 3$) - Dibuang/Nol:**
 - Karena output di rantai, z' harus selalu **0**.
 - Secara matematis: $1 - \frac{s_3(1)}{s_3} = 1 - 1 = 0$.
 - Hasil Baris 3: $[0, 0, 0]$.

Matriks Akhir (2×3):

$$A = \begin{bmatrix} 1 & 0 & -s_1/s_3 \\ 0 & 1 & -s_2/s_3 \end{bmatrix}$$

BAB 5: FILOSOFI & APLIKASI (Soal No. 5 & 6)

5.1 Kenapa Repot-repot Bikin Matriks? (Linear Transformation)

Q: Kenapa kok mintanya gitu? Kenapa saya harus melihat ini sebagai transformasi linear? A:

1. **Batch Processing (Game Dev):** Objek 3D (seperti Mario) terdiri dari **JUTAAN titik**. Menghitung rumus vektor manual satu-satu itu lambat (CPU based).
2. **Mesin Otomatis (GPU):** Dengan membuat Matriks A **SEKALI SAJA**, kita bisa melempar matriks ini ke GPU. GPU bisa mengalikan matriks ini ke jutaan titik **SEKALIGUS** (Paralel).
3. **Efisiensi:** Kita merakit "Mesin Pembuat Bayangan" (A) di awal, lalu mesin itu bekerja otomatis untuk seluruh titik di game.

5.2 Eksekusi Hitungan

Data: Vektor Matahari $\hat{s} = [4/13, -3/13, -12/13]$.

- Konstanta Baris 1: $-s_1/s_3 = 1/3$.
 - Konstanta Baris 2: $-s_2/s_3 = -0.25$.
- $-\frac{-3/13}{-12/13} = -\frac{3}{12} = -0.25$ **A Konkret:** $-\frac{-3/13}{-12/13} = -\frac{3}{12} = -0.25$

Matriks A:

$$A = \begin{bmatrix} 1 & 0 & 1/3 \\ 0 & 1 & -0.25 \end{bmatrix}$$

Soal 5: Satu Titik

Input $\mathbf{r} = [6, 2, 3]$.

- $x' = 6 + (0) + (1/3 \times 3) = 6 + 1 = 7$
- $y' = 0 + 2 + (-0.25 \times 3) = 2 - 0.75 = 1.25$ Hasil: $\mathbf{rp} = [7, 1.25]$

Soal 6: Batch Processing (Banyak Titik)

Input Matriks R :

$$R = \begin{bmatrix} 5 & -1 & -3 & 7 \\ 4 & -4 & 1 & -2 \\ 9 & 3 & 0 & 12 \end{bmatrix}$$

Rumus Cepat:

- $x'_{baru} = x_{lama} + (\frac{1}{3}z_{lama})$
- $y'_{baru} = y_{lama} - (0.25z_{lama})$

Hasil Per Kolom:

1. **Kol 1:** $x = 5 + (3) = 8, y = 4 - (2.25) = 1.75$
2. **Kol 2:** $x = -1 + (1) = 0, y = -4 - (0.75) = -4.75$
3. **Kol 3:** $x = -3 + (0) = -3, y = 1 - (0) = 1$
4. **Kol 4:** $x = 7 + (4) = 11, y = -2 - (3) = -5$

Hasil Akhir:

$$Rp = \begin{bmatrix} 8, & 0, & -3, & 11 \\ 1.75, & -4.75, & 1, & -5 \end{bmatrix}$$