

רינדור פוליגונים לפי מודל תאורה - Rendering Methods

כאשר אנחנו רוצים לצבוע משטח על המסך, איך קובעים את הצבע של כל פיקסל?
הצורה המהירה ביותר היא:

Flat Shading

לכל פוליגון נחשב את הנורמל, ונצבע את אותו הפוליגון באותו צבע.
זה בד"כ לא יוצא חלק...

Gouraud Shading

1. מחשבים לכל קודקוד מה הנורמל שלו. הנורמלים מוגדרים מראש בתוך ייצוג האובייקט בסצנה, או שאפשר לחשב אותם באמצעות ממוצע של כל המשטחים השכנים.
2. מחשבים את העצמה המוחזרת בכל קודקוד.
3. עושים אינטרפולציה בי-לינארית לכל נקודה לפי הפוליגונים:
למשל, אם יש פוליגון של 3 נקודות:

• עושים אינטרפולציה לינארית על ציר ה- u כדי למצוא נקודה I_4 על הקו P_1P_2 ו- I_5 על הקו P_3P_2 :

$$I_4 = uI_2 + (1 - u)I_1$$

$$I_5 = I_3u + (1 - u)I_2$$

• אינטרפולציה לינארית על ציר ה- v :

$$I = vI_4 - (i - v)I_5$$

השיטה הזאת מאפשרת לייצר משטח "מעוגל", ובמקום מעברים חדים בין הפוליגונים השינוי יהיה הדרגתי וחלק.

Phong Shading

כמו Gouraud, אבל הופכים את הסדר בין 2 ל-3:

1. מחשבים את הנורמל של כל קודקוד.
 2. עושים אינטרפולציה בי-לינארית על הנורמלים כדי לחשב את הנורמל בכל נקודה.
 3. מפעילים את מודל התאורה על כל נקודה לפי הנורמל שלה.
- השיטה הזאת יותר כבדה מ-Gouraud, אבל נותנת תוצאות יותר טובות שכן המעבר בין הצבעים הוא לא לינארי.

Global Illumination Effects

מודל התאורה מטפל באור שעובר ללא הפרעה - אבל איך מטפלים בצל? בהשתקפות? בשבירה של האור?

Transparency

איך מטפלים בשקיפות? אנחנו רוצים שאובייקט יהיה שקוף - שנוכל לראות את מה שנמצא מאחוריו:

$$I = (1 - K_1) I_1 + K_1 I_2$$

כאשר: K_1 השקיפות של האובייקט. אם $K_1 = 1$ האובייקט הוא שקוף - כלומר לא רואים אותו בכלל - ואם $K_1 = 0$ אז הוא אטום לחלוטין ולא רואים מה שיש מאחוריו. בד"כ במקום K נשתמש $\alpha = 1 - K$. האלפא היא האטימות של האובייקט.

I_1 התאורה בנקודה של האובייקט השקוף (למחצה)

I_2 התאורה בנקודה של האובייקט מאחוריו. אם גם האובייקט הזה שקוף אז צריך לחשב אותו בנפרד (עם I_3 ו K_2)

Ray Tracing

Ray Tracing הוא שיטת רינדור שפותרת כמה בעיות:

- איך לדעת אם משטח מוסתר (ולא צריך להציג אותו)
- מטפל טוב יותר בעניין הצללים - נותן החזרה אידיאלית ושבירה אידיאלית. אם רוצים החזרה ושבירה לא אידיאליות Ray Tracing לא יעבוד.
- מאפשר ציור ישיר של הרבה צורות גיאומטריות פשוטות. השיטה - לכל פיקסל ופיקסל מייצרים קרן ממיקומו בתוך העין ובודקים איפה היא פוגעת:
 - אם היא לא פוגעת בכלום, מחזירים את הרקע.
 - אם היא פוגעת במשטח, אפשר לחשב את עצמת התאורה.
 - אם המשטח מבריק, אפשר לחשב את זווית ההחזרה וליצור קרן חדשה שממשיכה את הקרן שלנו.איך עושים את reflection הזה? יש לנו את ווקטור כיוון הפגיעה V_{in} של הקרן ויש לנו את הנורמל N של המשטח. אנחנו רוצים לחשב את ווקטור כיוון ההחזרה V_{out} :

$$V_{out} = V_{in} - 2(V_{in} \cdot N) N$$

ככל שנאפשר יותר שבירות החישוב יהיה יותר כבד - אבל נקבל תמונה יותר ריאליסטית.

Refractions

שבירה מתרחשת כאשר יש הבדל פיזיקלי בין התכונות של החומרים השקופים (מקדם שבירה) לפי חוק Snell:

$$n_1 \sin \theta = n_2 \sin \varphi$$

כאשר: n_1 מקדם השבירה של החומר הראשון.
 θ זווית הפגיעה מהצד של החומר הראשון.
 n_2 מקדם השבירה של החומר השני.
 φ זווית ההחזרה מהצד של החומר השני.

האלגוריתם

- לכל פיקסל (x, y) בתמונה, ייצר את הקרן בעולם התלת מימדי
- חשב $\text{Image}(x, y) = \text{TraceRay}(\text{ray})$ לפי:
 - מצא את המשטח הראשון שהקרן פוגעת בו
 - אם אין - החזר את צבע הרקע
 - חשב את התאורה הישירה.
 - חשב את התאורה שמגיעה מההשתקפות
 - חשב את התאורה שמגיעה משבירה
 - שלב את כל עצמות התאורה לפי מודל shading
 - החזר את ערך התאורה

מציאת נקודת החיתוך

איך מוצאים נקודת חיתוך בין קרן למשטח?

יש לנו: • משטח $F(x, y, z) = 0$

$$\begin{aligned} R_x(t) &= O_x + tD_x \\ R_y(t) &= O_y + tD_y \\ R_z(t) &= O_z + tD_z \end{aligned} \quad \bullet \text{ ייצוג פרמטרי של הקרן } R(t) = O + tD \text{ כולומר:}$$

מציבים במשוואת המשטח:

$$F(O_x + tD_x, O_y + tD_y, O_z + tD_z) = 0$$

ומוצאים את הפתרון היחיד.

Radiosity

Ray Tracing לא תמיד מספיק:

- הוא לא מטפל בתאורה לא ישירה.
- הוא לא מטפל ב-Color Bleeding
- הוא לא מטפל ב-Soft Shadows - שטח מוצלל שכן מואר קצת מהאובייקטים מסביב.

כדי לפתור את הבעיה הזאת, לקחו חישובים מתחום אחר מפיסיקה - Radiosity (קרינה) - והשתמשו בצורה שפותרים את חישובי הקרינה שם כדי למצוא את התאורה בכל נקודה ונקודה במשטח. איך זה עובד? מחלקים את העולם למשטחים קטנים, וכל משטח כזה תורם לקרינה הכללית. הוא יכול לייצר קרינה בעצמו אם הוא מקור אור, וגם אם הוא לא מקור אור הוא מקבל קרינה מכל המשטחים האחרים וכל המשטחים האחרים משפיעים על הקרינה המתקבלת מאותו משטח:

$$B_i = E_i + \rho_i \sum_j F_{ij} B_j$$

כאשר:	B_i	כמות הקרינה שמשטח i פולט
	E_i	כמות הקרינה שהוא מייצר בעצמו (emitted)
	ρ_i	מקדם ההחזרה של המשטח
	F_{ij}	מקדם שאומר איך משטח j משפיע על משטח i

נשים לב ש- E_i ו- ρ_i ידועים, ואת F_{ij} אפשר לחשב לפי זוויות. אבל B_j תלויים בחישובי הנוסחה של המשטחים האחרים. לכן צריך לפתור מערכת משוואות:

$$\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix} + \begin{bmatrix} \rho_1 F_{11} & \rho_1 F_{12} & \cdots & \rho_1 F_{1n} \\ \rho_2 F_{21} & \rho_2 F_{22} & \cdots & \rho_2 F_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_n F_{n1} & \cdots & \cdots & \rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix}$$

ולפוט:

$$\begin{bmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1 - \rho_2 F_{22} & \cdots & -\rho_2 F_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_n F_{n1} & \cdots & \cdots & 1 - \rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

השיטה הזאת מאוד יקרה חישובית, ותלויה בכמה נפרק את העולם שלנו למשטחים קטנים. אפשר גם לעשות משטחים מסויימים גדולים ואחרים קטנים.