

Tema 6

Realisme

- Introducció al realisme
- Ocultacions
- Il·luminació

Introducció al realisme

- **Tècniques de realisme:** tècniques que proporcionen una major impressió de realisme de manera que permet crear una clara percepció visual de l'escena.
 - **Visualització tridimensional:** especificant relacions espacials entre els objectes.
 - Projeccions (ortogràfica, perspectiva)
 - Ocultacions
 - **Il·luminació:** tècniques que tracten diferents aspectes d'interacció de la llum entre els objectes de l'escena.
 - Il·luminació
 - Textures
 - Ombrejat
 - Transparències
 - **Síntesi interactiva d'imatges:** tècniques de manipulació d'imatges 3D i sistemes interactius.
 - Estèreo
 - Desenfocament
 - Animació
 - Realitat virtual

Ocultaciones

Ocultacions

- **Ocultacions**
determinar les entitats visible d'una escena segons el punt de vista.
 - **Característiques**
 - Algorisme senzills de concepte.
 - Alt cost de càlcul.
 - **Entitats**
 - Línies (*hidden line elimination*)
 - Superfícies (*hidden surface elimination*)
 - **Classificació**: segons les entitats que tractem.
 - Espai Objecte (*Object precision - Object space*)
 - Espai Imatge (*Image precision - Image space*)

Espai objecte

Comparació entre els objectes en l'escena (món) eliminat les parts no visibles.

```
per tot objecte de l'escena fer  
• determinar les parts de l'objecte que són visibles respecte  
del punt de vista.  
• pintar les parts de l'objecte amb el color apropiat  
fi per
```

- **Característiques**
 - Algorisme de línies ocultes (*hidden lines*)
 - Treballen en l'escena tridimensional amb tota la precisió (no hi ha *aliasing*).
 - Càlcul de visibilitat d'un objecte depèn dels altres objectes (no paral·lelisme).
 - Primers algorismes desenvolupats per a pantalles de refresc.
 - La complexitat augmenta significativament si augmentem el nombre d'entitats a l'escena.

Espai imatge

Analitza la visibilitat sobre cada pixel de la imatge.

```
per tot pixel de la imatge fer  
• calcular el raig antiprojectiu corresponent al pixel  
• trobar l'objecte més proper al punt de vista que talla  
aquest raig  
• pintar el pixel amb el color d'aquest objecte  
fi per
```

- **Característiques**

- Algorisme de cares ocultes (*hidden surface*)
- Treballen a resolució de pantalla (hi ha *aliasing*).
- Càlcul de visibilitat d'un pixel independent dels altres pixels (paral·lelisme).
- Representació d'àrees sòlides per dispositius *raster*.
- La complexitat no augmenta significativament si augmentem el nombre d'entitats a l'escena.

Millores de l'eficiència

- Les ocultacions són algorismes d'alt cost computacional.
- Farem us de tècniques per reduir el nombre de càlculs minimitzant el temps de resposta d'aquest algorismes
- **Optimitzacions:**
 - Ordenacions
 - Coherència
 - Envoltants (*bounding box, bounding volume*)
 - Tests de visibilitat (*back face culling*)

Milliores de l'eficiència

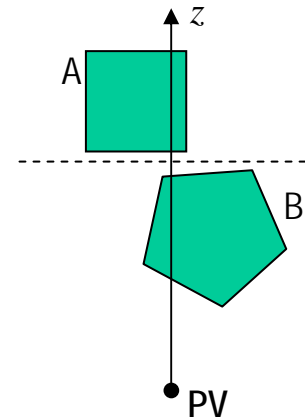
- **Ordenació**
 - Ordenació per profunditat (z) per decidir la precedència dels objectes segons el punt de vista.
 - Algorismes clàssics d'ordenació.
- **Coherència**
 - Els càlculs i les comparacions per cada element (objecte, superfície, aresta, vèrtex) de forma independent representa un alt cost de càlcul.
 - Aplicar **coherència** implica:
 - Aprofitar els càlculs realitzats i la regularitat de l'escena per expandir els càlculs de forma incremental en un entorn restringit.
 - Controlar les condicions que trenquen aquesta regularitat.

Millores de l'eficiència

- Tipus de coherència:

- **Coherència d'objecte:**

- Si dos objectes A, B són separables per un pla perpendicular a z , tota superfície de A comparada amb qualsevol superfície de B manté la mateixa relació de profunditat que els objectes.
 - Eliminem els càlculs de comparació entre les superfícies dels dos objectes.

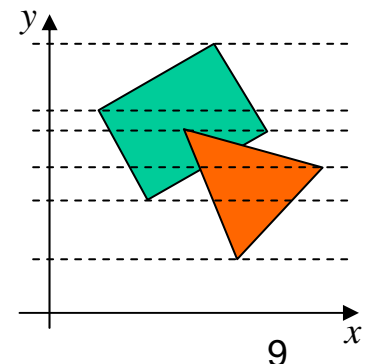


- **Coherència de superfícies:**

- Les propietats de les superfícies varien suaument.
 - Càlculs sobre una part de la superfície són ampliables incrementalment sobre altres parts de la superfície.

- **Coherència de línia d'escombrat (*scanline coherence*):**

- El conjunt de segments visibles per una línia d'escombrat varia mínimament respecte del conjunt de segments de la línia d'escombrat prèvia.



Milliores de l'eficiència

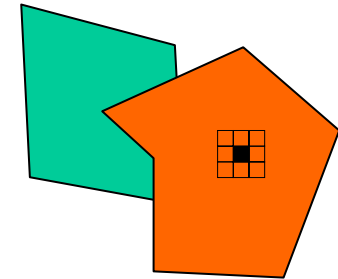
- Tipus de coherència (cont.):

- Coherència per àrea

- El conjunt de pixels adjacents a un pixel donat és probable que siguin de la mateixa cara visible.
 - Aquesta coherència no es verifica en els pixels de frontera entre cares.

- Coherència de *frame*

- Per moviments del punt de vista petits, els valors dels pixels en un *frame* d'una animació és similar als del *frame* anterior.
 - La coherència es trenca espacialment en els contorns de l'escena.
 - La regularitat es trenca temporalment en els canvis bruscos de punt de vista (càmera).



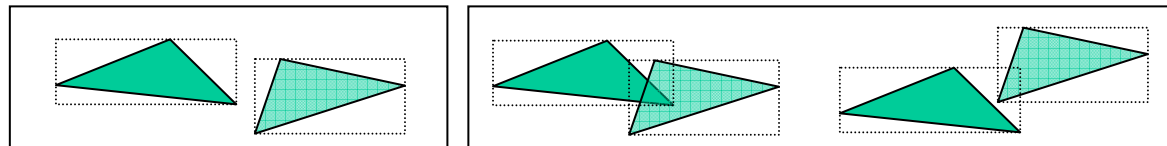
Milliores de l'eficiència

- **Envolvents**

- Definició d'una forma més senzilla (envolvent) associada a cada entitat de l'escena per minimitzar els càlculs de comparació entre objectes.

Comparació entre objectes → Comparació entre envolvents

- Criteris de comparació:
 - No existeix superposició d'envolvents \Rightarrow No existeix intersecció entre objectes.
 - Existeix superposició d'envolvents \Rightarrow Verificar si existeix interseccions.



- Tipus d'envolvents:
 - Unidimensionals (normalment z)
 - Bidimensionals (normalment x, y)
 - Tridimensionals
- Formes usuals d'envolvents:
 - Rectangulars
 - Poliedriques
 - Cilíndriques
 - Esfèriques

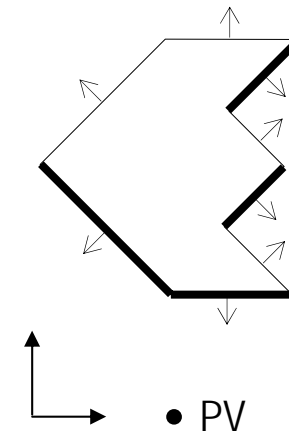
Milllores de l'eficiència

- **Test de visibilitat**
 - Tests de baix cost de càlcul que permet eliminar superfícies ocultes.
 - Test que decideixen la visibilitat total d'una cara (tests booleans).
 - Domini
 - Pel domini d'objectes convexes els test de visibilitat són suficients com a algorismes d'ocultacions.
 - Per objectes còncaus els test de visibilitat permeten filtrar un nombre considerable de cares pels algorismes d'ocultacions.
 - Restriccions:
 - normals a les cares en un únic sentit: cap a fora de l'objecte o cap a l'interior.
 - objectes tancats.
 - Tipus de test:
 - Test de visibilitat per pla.
 - Test de visibilitat per angle.

Millores de l'eficiència

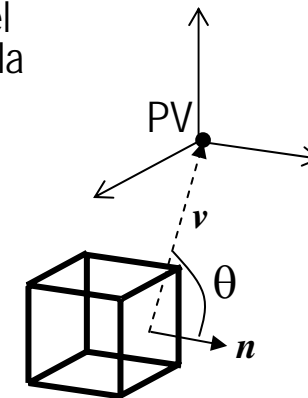
- **Test de visibilitat per pla**

- Tota funció $f(x,y,z) = Ax + By + Cz + D = 0$ subdivideix l'espai 3D en dos subespais.
 - » Subespai favorable al sentit del vector normal ($f(x,y,z) > 0$).
 - » Subespai contrari al sentit del vector normal ($f(x,y,z) < 0$).
- Sigui (A_1, B_1, C_1) la normal de la cara S_1 i (x_v, y_v, z_v) el punt de vista de l'escena.
 - » $A_1 x_v + B_1 y_v + C_1 z_v + D_1 < 0$ cara S_1 NO visible.
 - » $A_1 x_v + B_1 y_v + C_1 z_v + D_1 > 0$ cara S_1 visible.



- **Test de visibilitat per angle**

- Sigui $\mathbf{n} = (A_1, B_1, C_1)$ la normal a la cara S_1 , sigui (x_v, y_v, z_v) el punt de vista de l'escena, i sigui \mathbf{v} un vector que uneix un punt de la cara amb el punt de vista.
- Trobem l'angle $\theta = \arccos \left(\frac{\mathbf{n} \cdot \mathbf{v}}{|\mathbf{v}|} \right)$
 - » $\theta > \pi/2$ cara S_1 NO visible.
 - » $\theta < \pi/2$ cara S_1 visible.
 - » $\theta = \pi/2$ projecció de la cara en una línia.



Taxonomia dels algorismes

- **Espai objecte**
 - Roberts (Aresta-Volum)
 - Appel, Loutrel (Aresta-Aresta)
- **Prioritat per llista**
 - Newel, Newel & Sancha (*Depth Sort Algorithm*)
 - Schumacker (*Clusters*)
 - Arbres BSP (*Binary Space Partition*)
 - *Octree*
- **Espai imatge**
 - *Z-buffer*
 - *Scan Line*
 - Subdivisió per àrea
 - Warnock
 - Weiler & Atherton
 - *Visible Surface Ray Tracing*

Algorismes de prioritat per llista

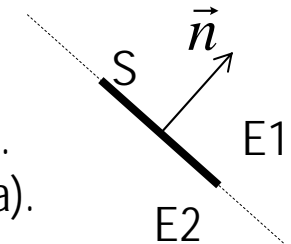
- Algorismes mixtes espai-objecte, espai-imatge.
- També anomenats com **algorismes del pintor**, doncs simulen l'acció d'un pintor al sobreposar els diferents colors (objectes) sobre la tela (pantalla).
- **Passos**
 - Ordenació de les superfícies per profunditat pels diferents punts de vista possibles (espai objecte)
 - Visualització ordenada de les superfícies (espai imatge)
- **Estratègies de visualització**
 - ***Back-to-front***
 - Dibuixar de la cara més llunyana fins a la més propera en profunditat (z).
 - Inconvenient: redundància (repintem en moltes zones de la imatge).
 - ***Front-to-back***
 - Dibuixar de la cara més propera fins a la més llunyana en profunditat (z).
 - Incloure un *buffer* de màscara (*buffer mask*, BM). *Buffer* que representa en cada bit un pixel de pantalla.
 - $BM(x,y)=1$: pixel ja pintat.
 - $BM(x,y)=0$: pixel no pintat.
 - Per cada cara, dibuixar pixels sempre que el bit corresponent en BM sigui 0, posant-los immediatament a 1.

Arbres BSP (*Binary Space Partition*)

- Mètode eficient per calcular relacions de visibilitat entre polígons 3D des d'un punt de vista arbitrari.
- Algorisme basat en un procés previ costós en càlcul per definir la partició espacial.
- Dóna un bon comportament en aplicacions on es canvia molt el punt de vista mantenint els elements fixes en l'escena (exploracions d'un entorn 3D).

Com funciona

- Tota superfície S talla l'espai en dos subespais
 - FRONT ($E1(S)$), a favor del vector normal (subespai de davant).
 - BACK ($E2(S)$), en contra del vector normal (subespai de darrera).
- Generem un arbre de partició espacial (BSP-Tree).

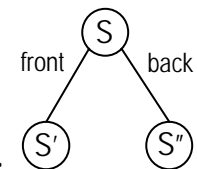


- Nodes: superfícies de l'escena

- Arcs: dos tipus

- Front (f). Superfícies situades a $E1(S)$, on S és la superfície del node pare.

- Back (f). Superfícies situades a $E2(S)$, on S és la superfície del node pare.



Si una superfície està als dos subespais s'ha de partir i tractar ambdues per separat. Arribarem a la profunditat d'arbre necessària fins a tenir un sol polígon.

Arbres BSP (*Binary Space Partition*)

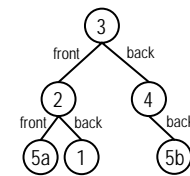
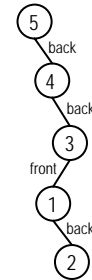
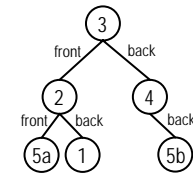
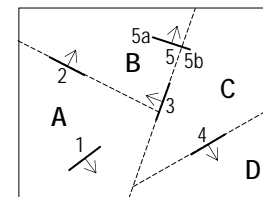
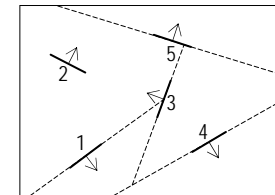
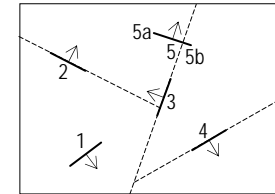
- **Passos:**

- 1.- **Obtenir l'arbre de partició espacial**

- Representació ambigua. Diferents arbres per la mateixa escena.
- Característiques desitjades d'un arbre:
 - Balancejat
 - Minimitzar el nombre d'interseccions
- No hi ha un mètode per trobar el arbre BSP òptim per una escena.

- 2.- Donat un punt de vista fer el **recorregut de l'arbre** per obtenir la llista de superfícies a visualitzar.

- Accedir al node (superfície S)
- Si el punt de vista està situat a $E1(S)$ [$E2(S)$]
 - 1.- Visualitzar superfícies $E2(S)$ [$E1(S)$] : baixar un nivell.
 - 2.- Visualitzar S
 - 3.- Visualitzar superfícies $E1(S)$ [$E2(S)$] : baixar un nivell.



- BACK → FRONT
- A: 4, 5b, 3, 5a, 2, 1
 - B: 4, 5b, 3, 1, 2, 5a
 - C: 1, 2, 5a, 3, 4, 5b
 - D: 5a, 2, 1, 3, 5b, 4

Algorisme d'espai imatge

- Determinen la visibilitat a nivell de pixels de pantalla.
- **Classificació d'algorismes:** segons grups de pixels a considerar.
 - Unidimensionals (pixel a pixel)
 - *Buffer* de profunditat (*Z-buffer*)
 - *Visible surface ray tracing*
 - Bidimensional (segments de línia d'escombrat)
 - Línia d'escombrat (*Scan Line*)
 - Tridimensional (subdivisió per àrea)
 - Warnock
 - Weiler & Atherton

Buffer de profunditat (Z-buffer)

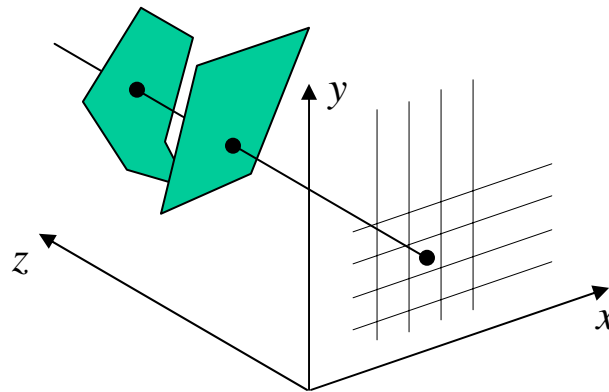
- Primer algorisme d'espai imatge.
- Dues estructures de dades:
 - **Frame buffer (F)**. La posició $f(x, y)$ emmagatzema el color de la cara visible del pixel (x, y) . Inicialitzat a 0 (negre). Resultat: **imatge final**
 - **Z-buffer (Z)**. La posició $z(x, y)$ emmagatzema la profunditat de la cara més propera pel pixel (x, y) . Inicialitzat a ∞ . Resultat: **mapa de profunditats**
- Per cada pixel es miraren totes les superfícies i decideix quina és la més propera per profunditat.
- **Algorisme**

```
per tot polígon pi en l'escena fer
  per tot pixel(x,y)interior al polígon pi fer
    pz = valor z de pi en (x,y)
    si pz < z(x,y) llavors
      z(x,y)=pz
      f(x,y)=color de pi en (x,y)
    fi si
  fi per
fi per
```

Buffer de profunditat (Z-buffer)

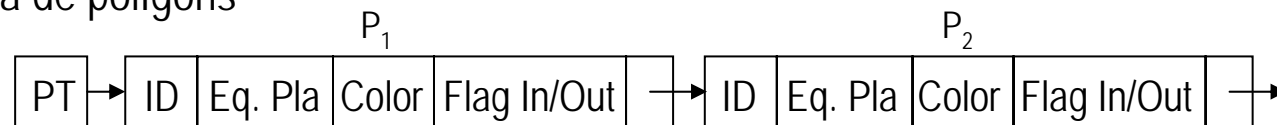
- Propietats

- Estructures de dades senzilles però voluminoses. Possibilitat de restringir els càlculs per línia d'escombrat minimitzant memòria.
- Inconvenient: *aliasing*
- Possibilitat de guardar *buffers* de profunditat ja calculats per aplicar l'algorisme sobre objectes mòbils. Guardem el *Z-buffer* del fons.
- *O-buffer* (Objecte-*buffer*). *Buffer* on es guarda per cada pixel les diferents profunditats de les superfícies que defineixen cada objecte. Útils per algorismes de realisme (transparències, ombrejat), retall, operacions booleanes.

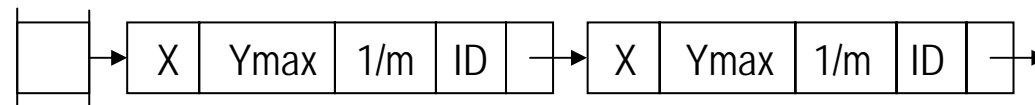


Línia d'escombrat (*Scan Line*)

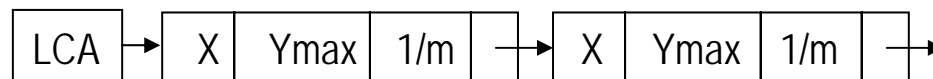
- Càlcul de visibilitat segons els segments de línia d'escombrat (*span*).
- Admet diferents tipus de coherència:
 - coherència per línia d'escombrat.
 - coherència per aresta
- Extensió de l'algorisme Y-X per ocultacions.
- Utilitzem tres taules
 - Llista de polígons



- Taula de costats



- Llista de costats actius



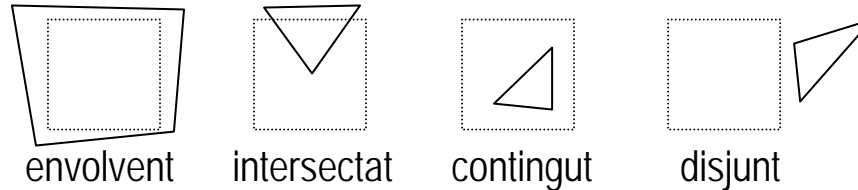
Subdivisió per àrea

- Algorismes de *divide & conquer* sobre regions de pantalla.
- Subdivisions recursives d'àrees en el pla de projecció simplificant l'anàlisi de l'escena.
- Algorismes sensibles a la complexitat de l'escena.
- Propietats
 - Coherència per àrea
 - Tècniques anti*aliasing*: nivells de subdivisió a nivell de subpixel.
- Algorismes
 - Warnock (*quadtrees*)
 - Weiler & Atherton

Subdivisió per àrea

- **Warnock**

- Subdivisió del pla de projecció en particions quadrades (*quadrees*).
- Tot polígon de l'escena es relaciona amb un element d'un quadree d'una de les següents formes:



- **Algorisme**

- per cada element del quadree, retallar i classificar els polígons sobre el quadree.
- els polígons d'interès del quadree són: envolvents, intersectats, continguts.

Estrategies:

- Tots els polígons d'interès són disjunts → pintem amb color de fons
- Existeix un únic polígon d'interès de tipus intersectat o contingut → primer dibuixem amb color de fons i després la part del polígon d'interès.
- Existeix un únic polígon d'interès envolvent → pintem amb color del polígon.
- Existeix més d'un polígon d'interès però un envolvent per davant. Dos casos:
 - » linealment separable → pintem amb color de l'envolvent.
 - » linealment no separables → SUBDIVIDIR
- Existeix més d'un polígon d'interès però cap envolvent per davant → SUBDIVIDIR

Visible surface ray tracing

- Per cada pixel tracem el raig que passa pel centre de projecció i la posició del pixel i mirem dins de l'escena els punts de tall d'aquest raig amb els objectes de l'escena. Agafem el punt d'intersecció més proper al centre de projecció.
- **Algorisme**

```
seleccionar centre de projecció (CP) i finestra
per tot pixel(pi) de la finestra fer
    determinar raig (ri) entre CP i pi
    per tot objecte(oj) en l'escena fer
        si oj intersecta ri i és més proper que actual llavors
            actual[pi] = punt d'intersecció
            pintem la intensitat de l'objecte oj a pi
            f(x,y)=color de pi en (x,y)
        fi si
    fi per
fi per
```

- Aplicacions
 - Ocultacions
 - Transparències, ombres
 - Visualització de models CSG

II.luminació

Il·luminació

- **Il·luminació**

assignar intensitats de llum als objectes de l'escena en funció de:

- Orientació de la superfície
 - Posició de la superfície
 - Característiques de la superfície
 - Fons de llum
- **Models d'il·luminació:** consideren els diferents factors que influeixen en la intensitat final d'un punt. Són models de tipus físic, que intenten explicar el que passa simulant el comportament de la llum en una escena.
Simplificació del problema.
 - **Algorismes d'il·luminació (*shading algorithms*):** determinen els models d'il·luminació que hem d'aplicar i els seus arguments.
 - Algorismes incrementals (il·luminació local)
 - Algorismes d'il·luminació global

Factors òptics

- **Propietats òptiques dels objectes**
 - **Superfície:** brillant / mat
 - **Material:** opac / translúcid / transparent
- **Tipus de reflexions**
 - **Difusa:** reflexions de baixa intensitat a totes direccions. Característica de superfícies mates i opaques.
 - **Especular:** reflexions d'alta intensitat i puntuals. Característiques de superfícies brillants.
- **Fonts de llum**
 - **Fonts emissores:** focus de llum.
 - **Puntuals:** fonts de llum petites en comparació als objectes de l'escena.
 - **direccionals:** fonts de llum a l'infinit o molt allunyades.
 - **Distribuïdes:** fonts de mida considerable respecte dels objectes de l'escena.
 - **Fonts reflectores:** objectes de l'escena (il·luminació global)

Models d'il·luminació

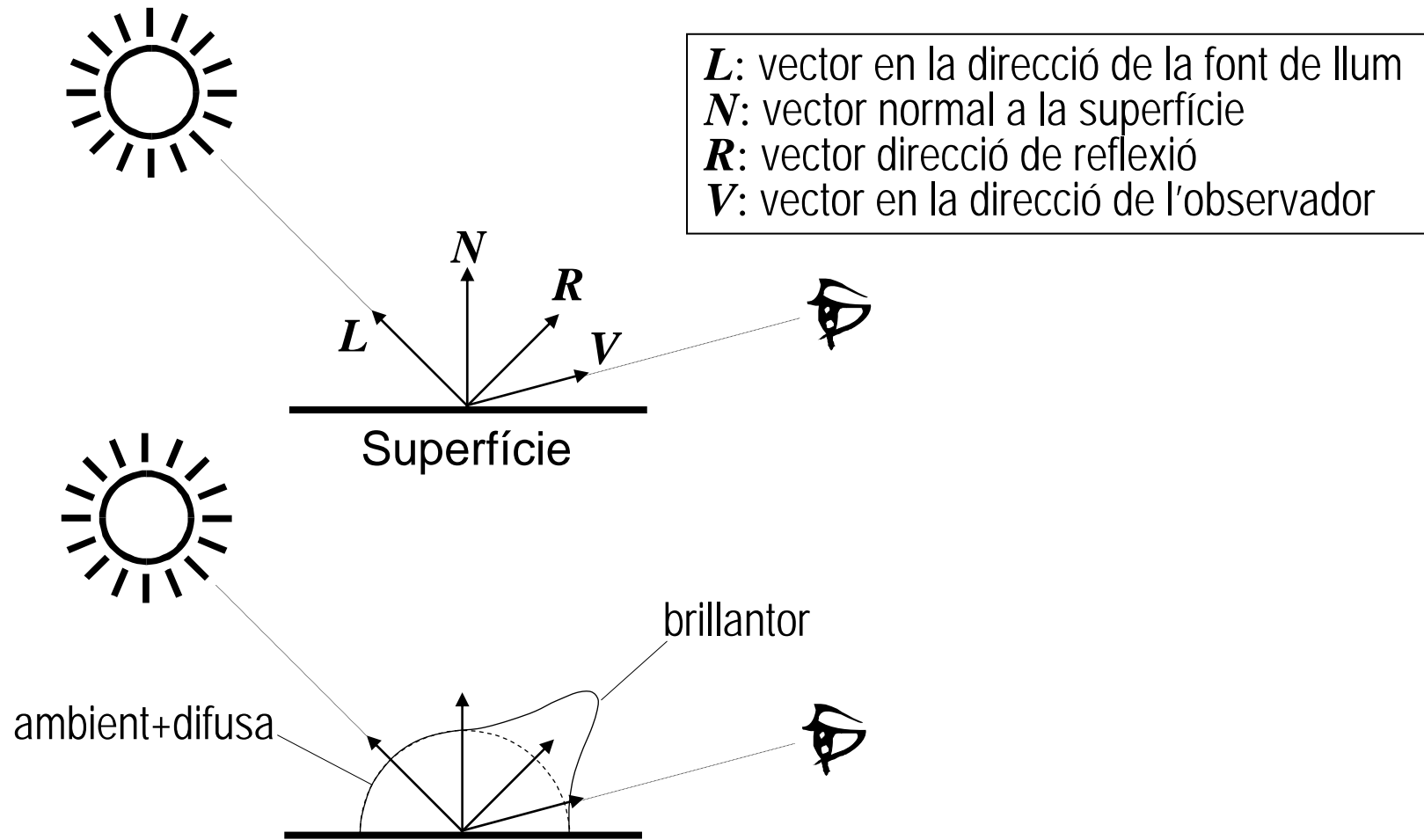
- Estudi de les diferents radiacions lumíniques que poden incidir en un punt de l'escena i diferents efectes que tenen lloc en aquest procés.
- Models utilitzats en algorismes d'il·luminació en major o menor mesura.

Tipus de models d'il·luminació

- Llum ambient *
- Reflexió difusa *
- Atenuació de la font de llum *
- Reflexió especular *
- Múltiples fonts de llum
- Fonts de llum direccionals

* termes de l'equació d'il·luminació

Geometria



Llum ambient

- Deguda a que cada objecte reflexa una intensitat de llum que incideix sobre altres objectes. El conjunt d'aquestes reflexions l'anomenem **Llum ambient**. La modelització d'aquest fenomen és complexa però es pot aproximar en molts casos per una intensitat constant en totes les direccions.
- Model més simple de il·luminació que dóna lloc a la següent equació d'il·luminació:

$$I = I_a k_a$$

on I : és la intensitat del punt que estem considerant

I_a : és la intensitat de llum ambient (constant per tota la escena)

k_a : és el coeficient de reflectivitat ambient ($k_a \in [0,1]$). Valor empíric associat al material.

Reflexió difusa

- Intensitat reflexada per l'objecte sense una direcció privilegiada. Utilitzem la llei de Lambert per modelitzar-la ($I=I_o \cos\theta$ on θ és l'angle d'incidència).

Reflexió lambertiana

- Model teòric per superfícies mats ideals.
- La il·luminació només depèn de l'angle d'incidència (angle entre \mathbf{N} normal i \mathbf{L} direcció de la font de llum) i ho fa amb el cosinus de l'angle.
- Afegint aquest terme de difusa al d'ambient que ja teníem ens queda:

$$I = I_a k_a + I_p k_d (\mathbf{N} \cdot \mathbf{L})$$

on I_p : és la intensitat de la font de llum.

k_d : és el coeficient de reflexió difusa ($k_d \in [0,1]$), depèn del material.

\mathbf{N} : vector normal a la superfície (normalitzat).

\mathbf{L} : vector direcció de la font de llum (normalitzat). Si la font és direccional (llunyana) \mathbf{L} és constant per tota l'escena.

Reflexió especular

- Característica de superfícies brillants. Zona brillant localitzada que pren el color de la font de llum.
- La reflexió especular té la direcció de **R** (direcció de reflexió) i decreix progressivament al voltant d'aquesta direcció
- La intensitat observada dependrà de la posició del observador (**V**).
- El model que té en compte tots els termes rep el nom de

Model d'il·luminació de Phong

$$I = I_a k_a + f^{at} I_p [k_d(\mathbf{N} \cdot \mathbf{L}) + k_s(\mathbf{R} \cdot \mathbf{V})^n]$$

- on
- f_{at} : factor d'atenuació que té en compte la distància entre la font i la superfície.
 - k_s : és el coeficient de reflexió especular ($k_d \in [0,1]$), depèn del material.
 - R**: vector direcció de reflexió (normalitzat).
 - V**: vector en la direcció de l'observador.
 - n : exponent per tenir en compte la brillantor ($n \in [1,500]$), depèn del material.

Múltiples fons i color

Podem estendre el model d'il·luminació de Phong al cas de múltiples fons i que l'escena tingui color.

- Si tenim més d'una font (M) hem de **sumar** les contribucions que fan cadascuna d'aquestes fons.
- Per tenir en compte el color farem els valors que intervenen en el model depenguin de la **longitud d'ona** i realitzarem el càlcul per algunes longituds d'ona (de 3 a 9). També separarem per un costat els coeficients de reflexió que dependran únicament del material i per altre el color propi de cada objecte (kO_λ)

La equació resultant és:

$$I_\lambda = I_{a\lambda} k_a O_{a\lambda} + \sum_i^M f_i^{at} I_{p\lambda i} [k_d O_{d\lambda} (\vec{N}\vec{L}_i) + k_s O_{s\lambda} (\vec{R}_i\vec{V})^n]$$

Algorismes incrementals

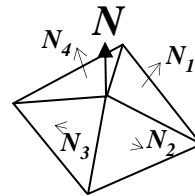
- Algorismes d'il·luminació per malles poligonals que simulen una suavització de la superfície tenint en compte la informació dels polígons adjacents.
- Algorismes amb cost de càlcul baix.
- Algorismes basats en interpolació segons línia d'escombrat de pantalla (algorismes d'espai imatge).

Algorismes:

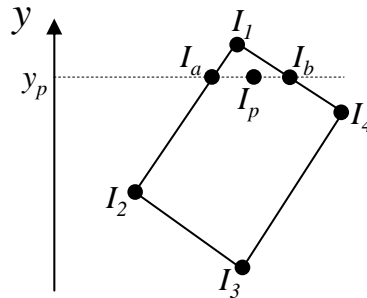
- **Gouraud** (interpolació d'intensitats).
- **Phong** (interpolació de vectors normals).

Algorisme de Gouraud

- Interpolació d'intensitats sobre cada polígon mitjançant els valors d'intensitat obtinguts en els vèrtexs.
- Passos:
 - Calcular vectors normals en els vèrtex com a mitja dels vectors normals de les facetes que incideixen en el vèrtex.



- Calcular la intensitat del vèrtex (I_i) mitjançant l'equació d'il·luminació i els models d'il·luminació considerats.
- Interpolació d'intensitats mitjançant les intensitats en els vèrtex. Es pot optimitzar aplicant coherències.



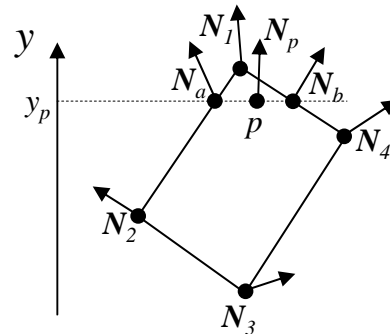
$$I_a = I_1 + (I_2 - I_1) \frac{y_p - y_1}{y_2 - y_1}$$

$$I_b = I_4 + (I_1 - I_4) \frac{y_p - y_4}{y_1 - y_4}$$

$$I_p = I_a + (I_b - I_a) \frac{x_p - x_a}{x_b - x_a}$$

Algorisme de Phong

- Interpolació de **vectors normals** sobre el polígon mitjançant els vectors normals dels vèrtexs.
- Intensitat calculada separatament per cada punt.
- Simplificació **L**, **R** constants.
- Passos:
 - Calcular vectors normals en els vèrtexs com a mitja dels vectors normals de les facetes que incideixen en el vèrtex.
 - Interpolació dels vectors normals mitjançant els dels vèrtexs. Es pot optimitzar aplicant coherències.



$$\vec{N}_a = \vec{N}_1 + (\vec{N}_2 - \vec{N}_1) \frac{y_p - y_1}{y_2 - y_1}$$

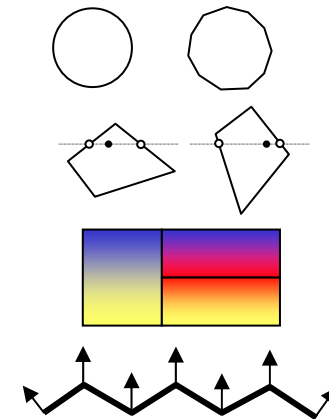
$$\vec{N}_b = \vec{N}_4 + (\vec{N}_1 - \vec{N}_4) \frac{y_p - y_4}{y_1 - y_4}$$

$$\vec{N}_p = \vec{N}_a + (\vec{N}_b - \vec{N}_a) \frac{x_p - x_a}{x_b - x_a}$$

- Calcular la intensitat en el punt mitjançant l'equació d'il·luminació i els models d'il·luminació considerats.

Avantatges i inconvenients

- Gouraud
 - ↑ Baix cost en càlcul
 - ↓ No considera intensitats especulars dels punt interiors. L'atenuació especular que es proporcional a $\cos^n \alpha$ s'interpola de forma lineal.
- Phong
 - ↑ Té en compte la reflexió especular en punts interiors
 - ↑ La interpolació del vector normal es una bona aproximació de la curvatura de la superfície que ha estat aproximada per facetes.
 - ↓ S'ha de normalitzar el vector interpolat. Alt cost en càlcul.
- Inconvenients dels algorismes incrementals
 - ↓ Silueta poligonal: les aproximacions per facetat no s'ajusten a les superfícies corbes inicials.
 - ↓ Dependència de l'orientació: els resultats d'interpolació depenen de l'orientació del polígon en pantalla.
 - ↓ Compartició de vèrtex: si un vèrtex existeix per uns polígons però no per altres tenim diferents valors d'interpolació.
 - ↓ Vectors normals ponderats poc representatius: es pot donar el cas que es compensin.



Algorisme d'il·luminació global

- La intensitat d'un punt en l'escena depèn:
 - Llum incident que prové de fons de llum (il·luminació local).
 - Llum reflectida d'altres superfícies (il·luminació global).

Il·luminació global

té en compte les contribucions d'intensitats de llum de superfícies d'altres objectes que hi ha a l'escena

Efectes:

- Ombres
- Transparències
- Reflexos (mirall)
- Llum ambient

Algorismes:

- Traçat de raigs (*ray tracing*): dependent del punt de vista.
- Radiositat (*radiosity*): independent del punt de vista.

Ray tracing

- Determina la intensitat d'un pixel segons el punt de l'escena més proper
 - Local: considera únicament els raigs que provenen de les fonts de llum.
 - Global: té en compte reflexió, refracció (transmissió) de les superfícies.

Algorisme:

- Per cada pixel tracem un raig que intersecta l'escena en un primer punt p .
- Calculem el valor de la intensitat local en aquest punt i tracem dos altres raigs: reflexat (R) i el refractat o transmès (T). $I_p = I_{\text{local}} + I_R + I_T$
- Aquest raigs intersectaran novament l'escena en altres punts que donaran lloc a noves intensitats i així successivament de manera recursiva fins :
 - un nombre determinat de nivells
 - la intensitat que aporten les noves contribucions és mínima
 - no hi ha tall amb l'escena
- Avantatges i inconvenients
 - ↑ Es pot paral·lelitzar, càlcul d'intensitat en cada pixel independent.
 - ↑ Té en compte difusa i especular.
 - ↑ Permet fer ocultacions, ombres, transparències, reflexions (miralls).
 - ↓ Alt cost.
 - ↓ *Aliasing*

Radiosity

- Algorisme que assumeix la conservació d'energia lumínica sobre un entorn tancat.
- Radiositat: quantitat d'energia emesa per una superfície es la suma de:
 - Energia emesa per la superfície
 - Energia reflexada que prové d'altres superfícies
- Donada una escena que està composta per un conjunt de superfícies (*patch*). Cadascun d'aquests *patches* té una radiositat definida per:

$$B_i = E_i + \sigma_i \sum_j B_j F_{j-i} \frac{A_j}{A_i}$$

on B_i, B_j són les radiositats dels patches

E_i és la llum emesa pel patch

σ_i és la reflectivitat del patch

F_{j-i} és el factor de forma que té en compte la geometria

A_i, A_j són les àrees dels patches i, j

Coneixem $E_i, \sigma_i, F_{j-i}, A_i$ i tenim com a incògnites B_i . Resolem aquest gran sistema d'equacions i obtenim els valors de radiositat de cadascun dels *patches* de l'escena.