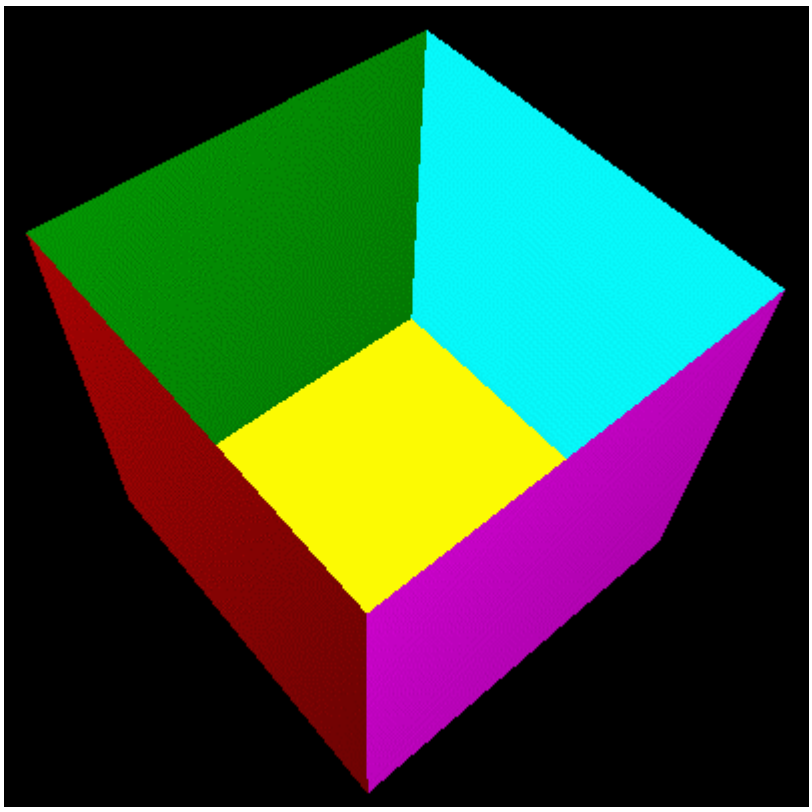
KHÖ MAT KHUAT

HIDDEN SURFACE REMOVAL



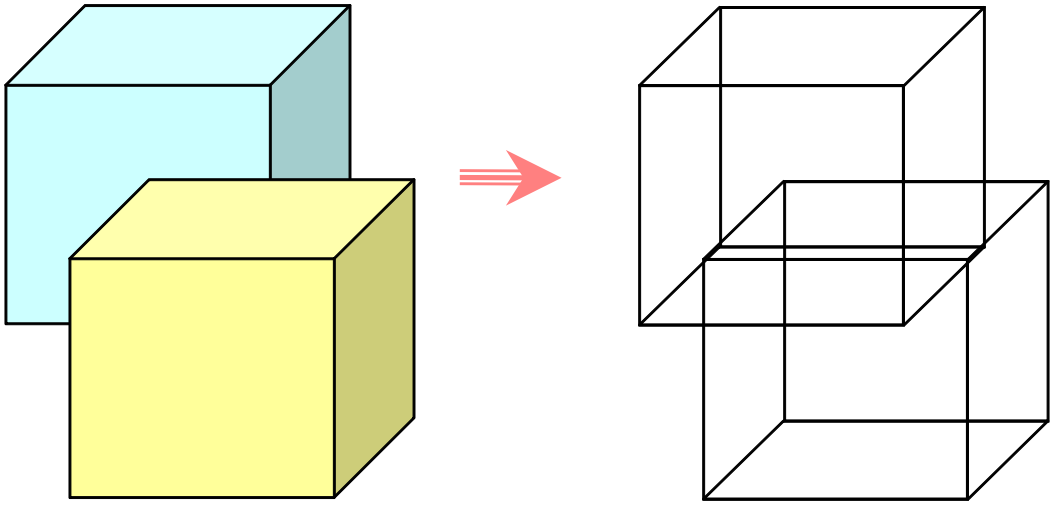
Cac dang khac nhau cua van ñe khö mat khuat

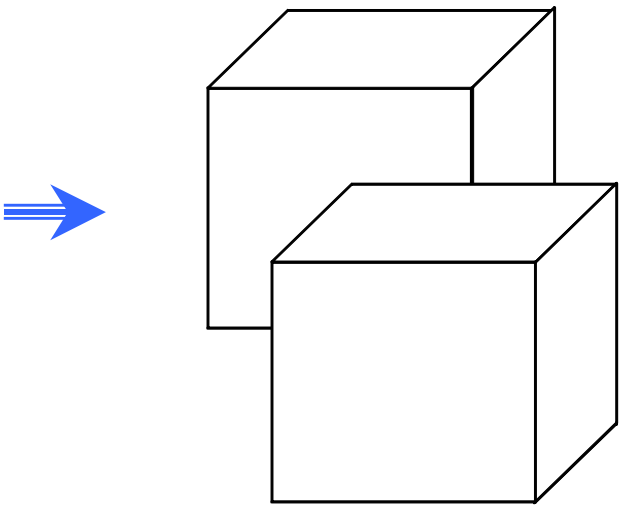
Cac thuat toan khö mat khuat (HSR)

- Back-face detection
- Painter's algorithm
- Ray casting
- Z-buffer
- Scan-line
- Area subdivision

Dan nhap

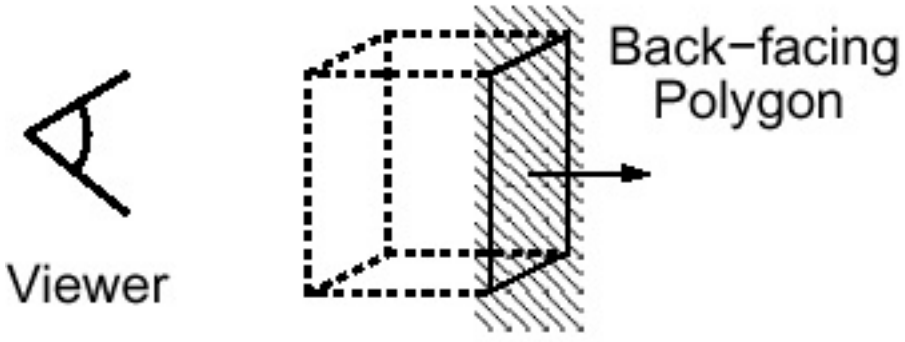
- Khi chieu canh cua ta tö khong gian 3 chieu xuong khong gian 2 chieu (screen space) doc theo truc z, cac ñiem nam tren cung mot tia chieu se co chung mot anh.
- Van ñe la khi hien thò, ta phai chon mau thích hôp cho ñiem nay. Mau ño phai la mau cua ñoi töông ma ta that sö thay ñöôc (gan ta nhat) chö khong phai ñoi töông bò che khuat (bôi ñoi töông khac).
- Khi muon co hình anh that ta khong the khong khö mat khuat (xem ví du ben döôi)



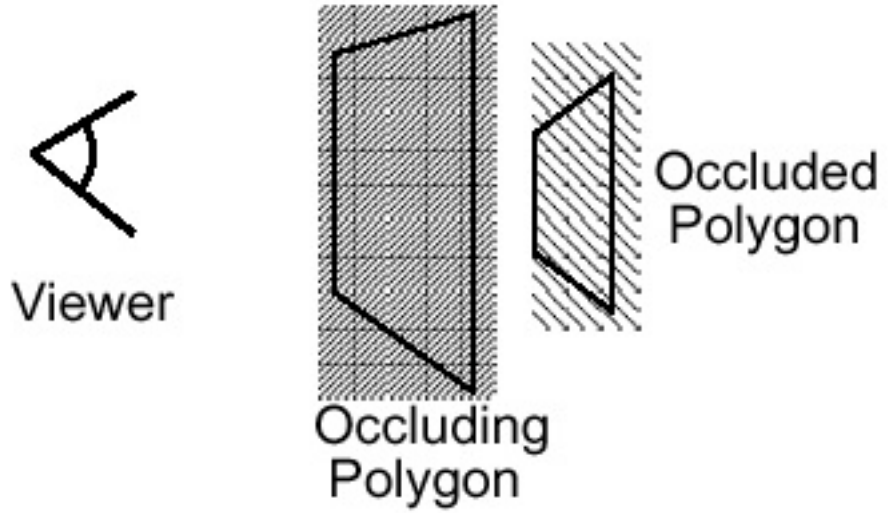


Cac dang khac nhau cua van ñe khö mat khuat

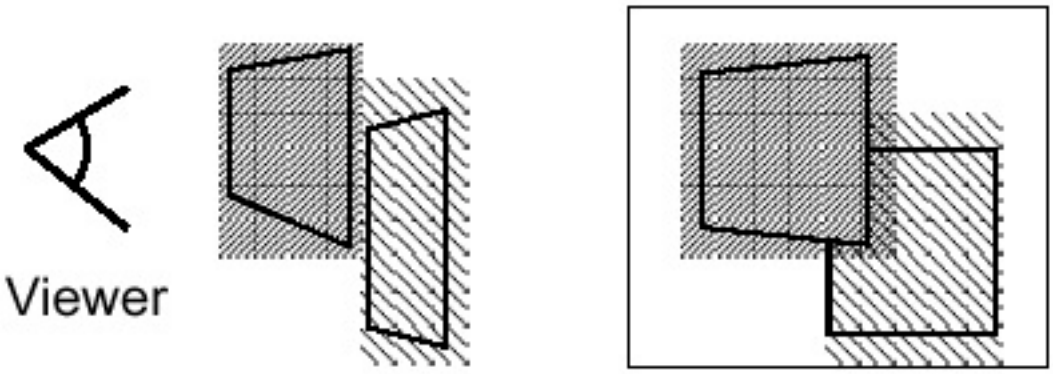
- Cac mat co the quay löng lai vôi ngöôi quan sat (Back-face)



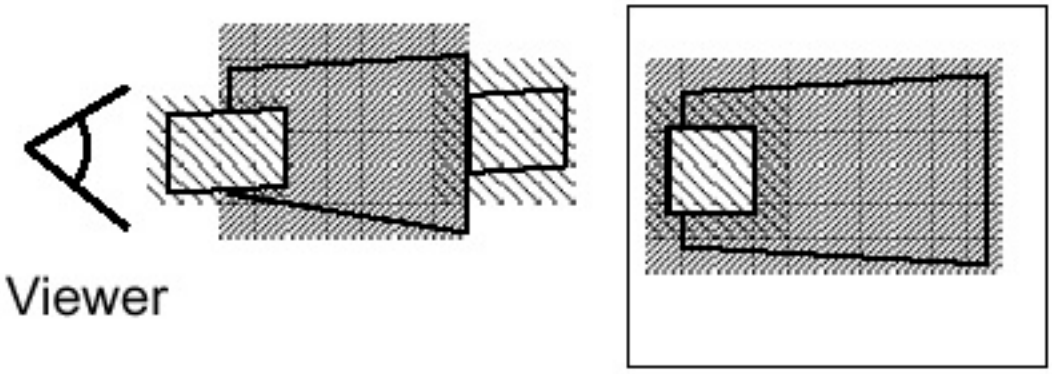
- Cac mat co the bò che bôi cac mat khac



- Cac mat co the chong len nhau

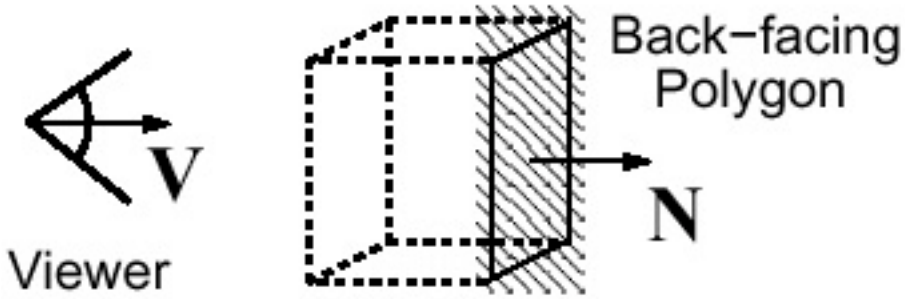


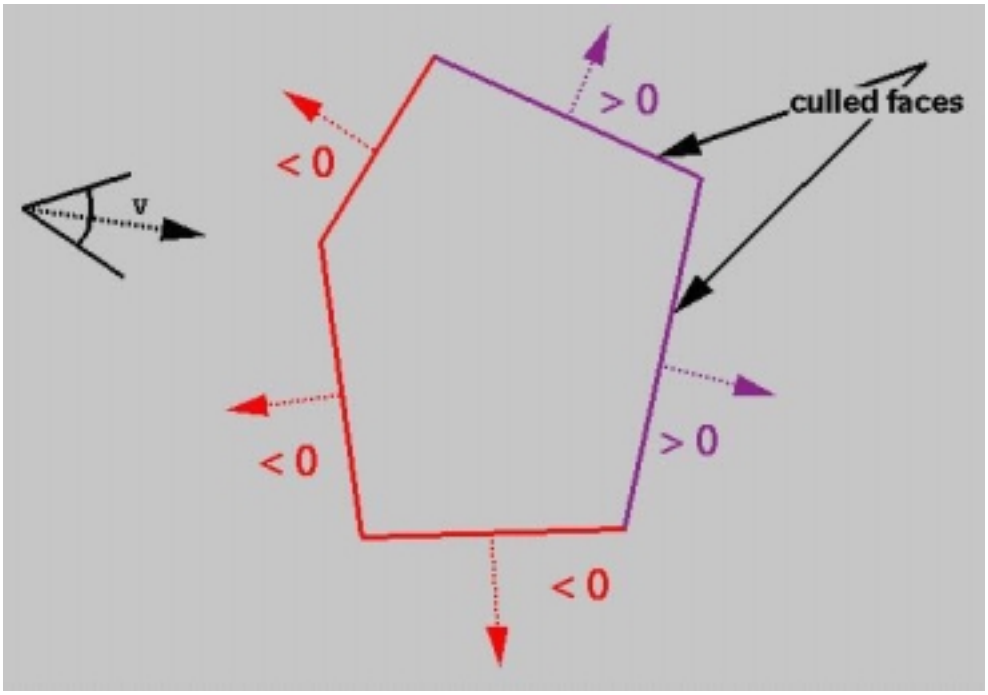
- Caùc maët coù theå caét nhau



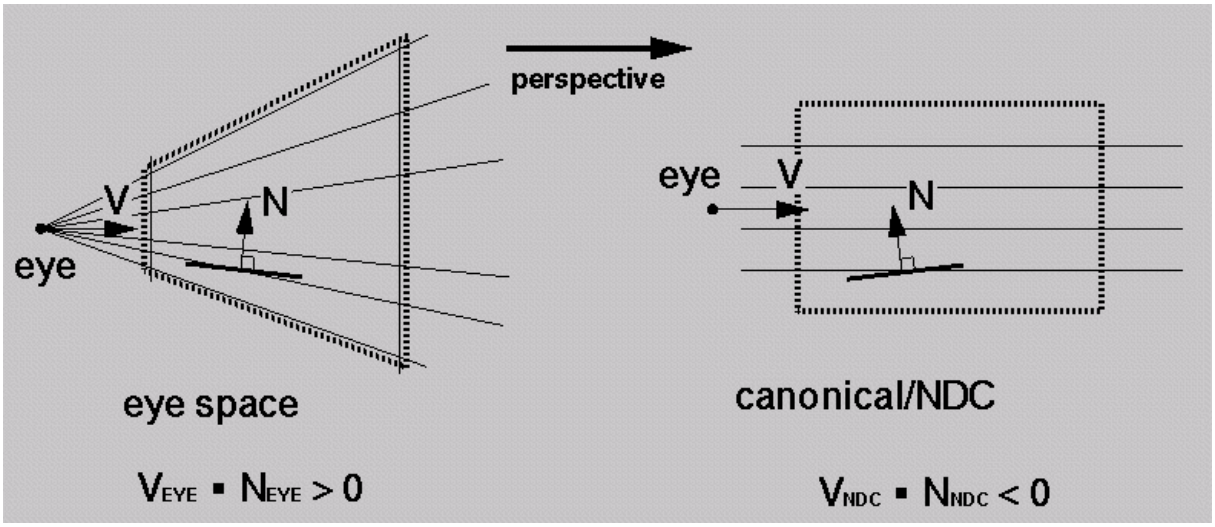
Back-face detection

- Khoâng hieån thò caùc maët höôùng ra töø vò trí quan saùt
- Moät polygon quay löng laïi viewer neáu $V \bullet N > 0$.

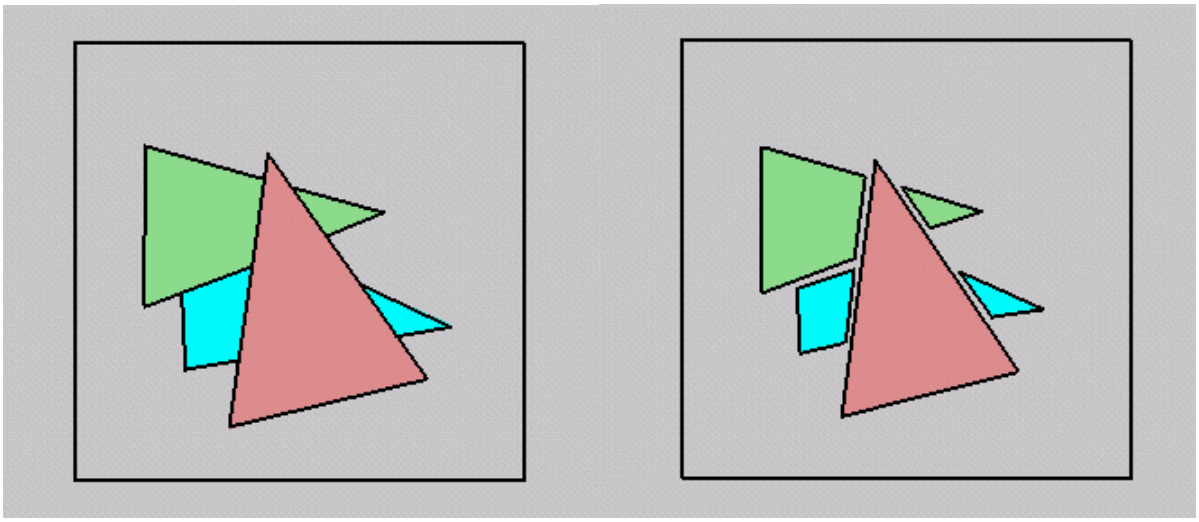




- Ta coù theå aùp duïng pheùp "NORMAL TEST" treân ñeå kieåm tra vôùi caùc pheùp chieáu khaùc nhau ?

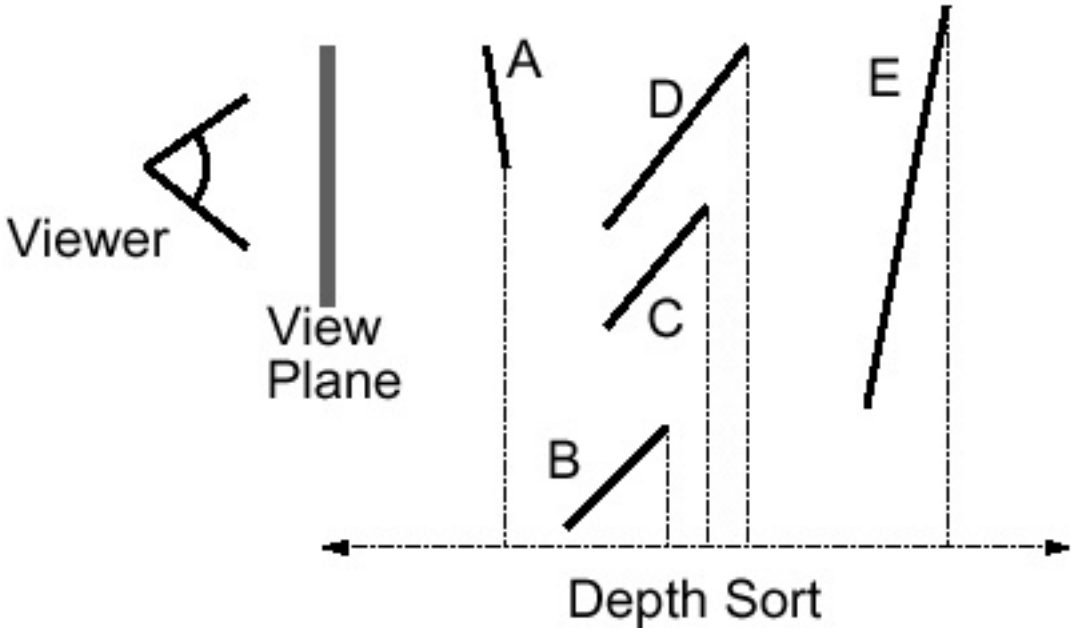


- Khi naøo ta phaûi aùp duïng pheùp back-face culling ?
- Chi phí cho coâng vieäc naøy treân n polygon laø bao nhieâu ?
- Giaû quyeát xong baøi toaùn back-face culling ta ñaõ giaû quyeát xong baøi toaùn HSR chöa ?
- Dó nhieân laø chöa. Trong raát nhieàu caûnh caùc maët chöa leân nhau. Ta phaûi giaû quyeát baèng caùch khaùc.

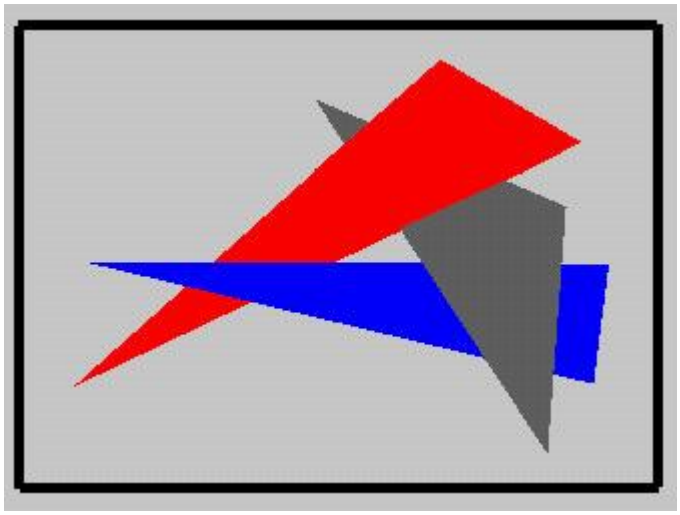


Depth sorting

- Con goi la Painter's algorithm
- Sap xep cac mat theo thöù töï töø xa ñeán gaàn (giam dan theo ño sau) theo vò trí sau nhat cua moi mat.
- Scan convert töng mat theo thöù töï nay.



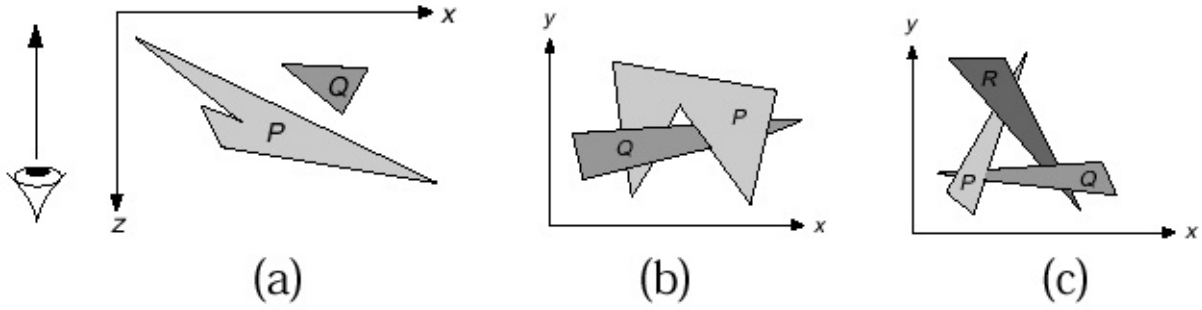
- Tuy nhieân, khong phai bao giô ta cung co the sap xep theo ño sau (xem hình döôi)



- Giai quyeát vaán ñeà nay nhö the nao ?

Xöû lyù caùc vöôøng maùc khi tính ño sau

- Khi saép xeáp caùc maët theo ño sau, coù nhieàu tình huoáng xaùc ñònh raát khoù.



(a) (b) (c)

- Thuaät toaùn saép xeáp theo ño sau coù theå caûi ñaët nhö sau:

  1. Khôûi ñoäng vieäc saép xeáp theo vò trí z nhoû nhaát (xa)

  2. Giaûi quyeát caùc mô hoà:

  (a) So saùnh theo toaï ño X

  (b) So saùnh theo toaï ño Y

  (c) Kieåm tra P coù hoaøn toaøn naèm veà 1 phía cuûa Q ?

  (d) Kieåm tra Q coù hoaøn toaøn naèm veà 1 phía cuûa P ?

  (e) So saùnh hình chieáu leân X-Y (Polygon Intersection)

  (f) **Hoaùn vò hoaëc taùch caùc polygon**

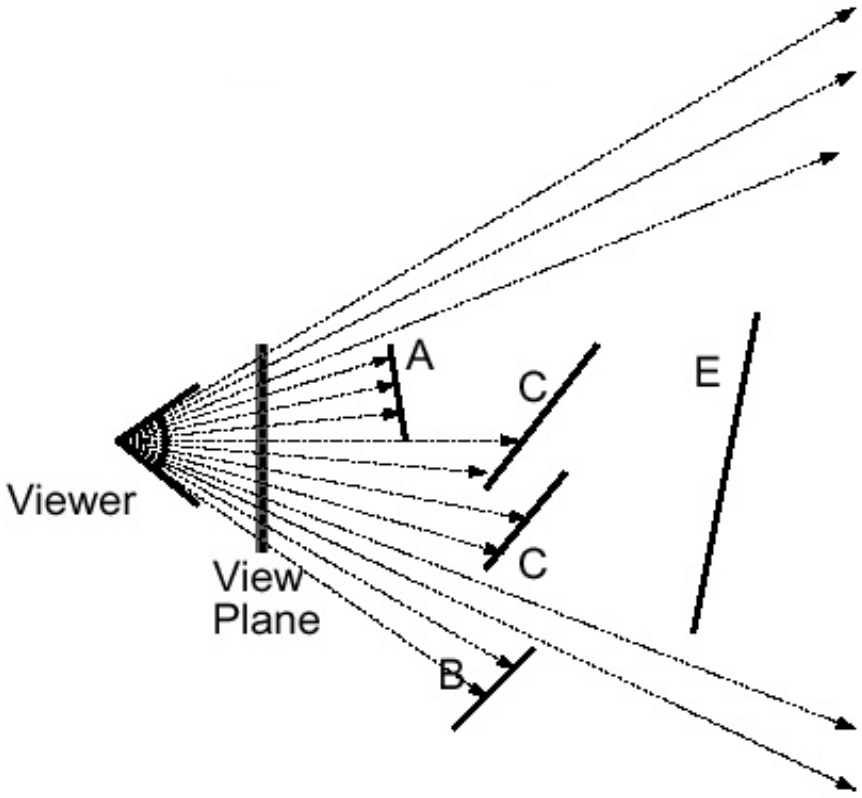
  3. Scan convert töø xa ñeán gaàn.

- Moät soá löu yù veà Painter's Algorithm

  - Coù ño phöùc taïp O(nlogn)
  - Caùc polygon caét nhau phaûi ñöôïc chia thaønh caùc polygon con.
  - Phaûi tính toaùn treân moãi pixel cuûa moãi polygon.
  - Vieäc xaùc ñònh ño sau cuûa caùc maët khoâng ñôn giaûn

Ray casting

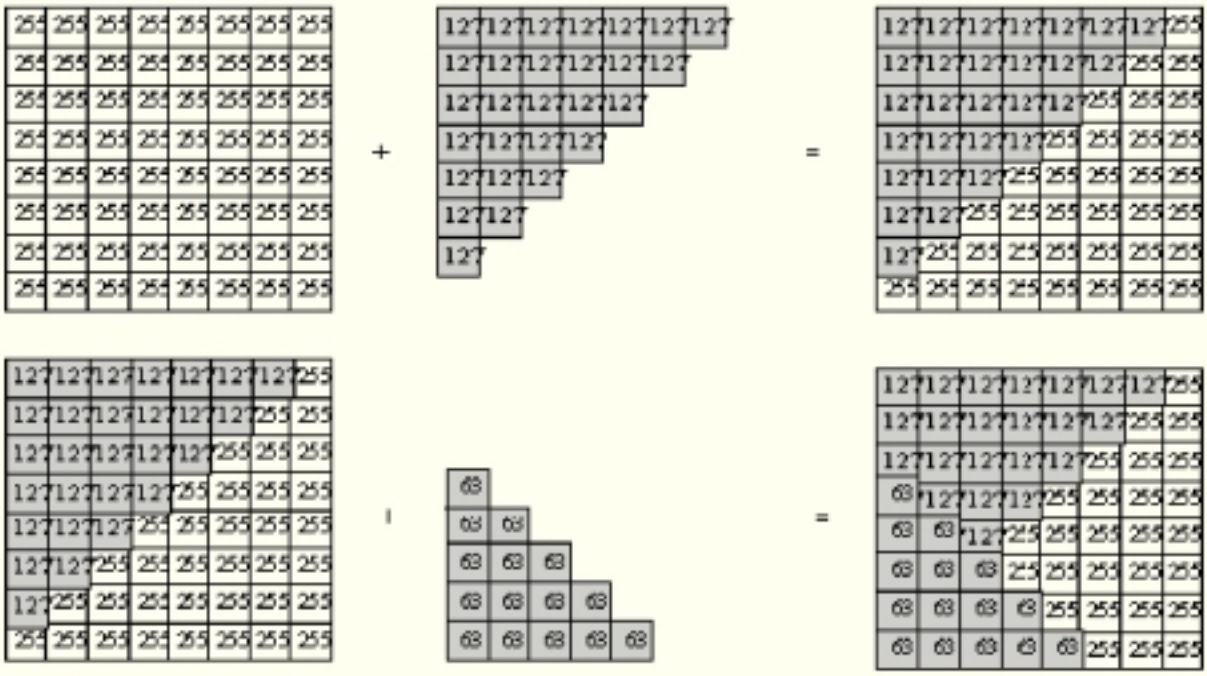
- Tö view point phong cac tia ñen moi ñiem tren view plane.
- Xac ñònh mat gan nhat cat cac tia nay.



- Mot so löu y ve Ray casting
  - Co ño phöc tap O(plogn) vôi p la so pixel tren VP
  - Ñôn gian ve mat khai niem nhöng khong pho dung

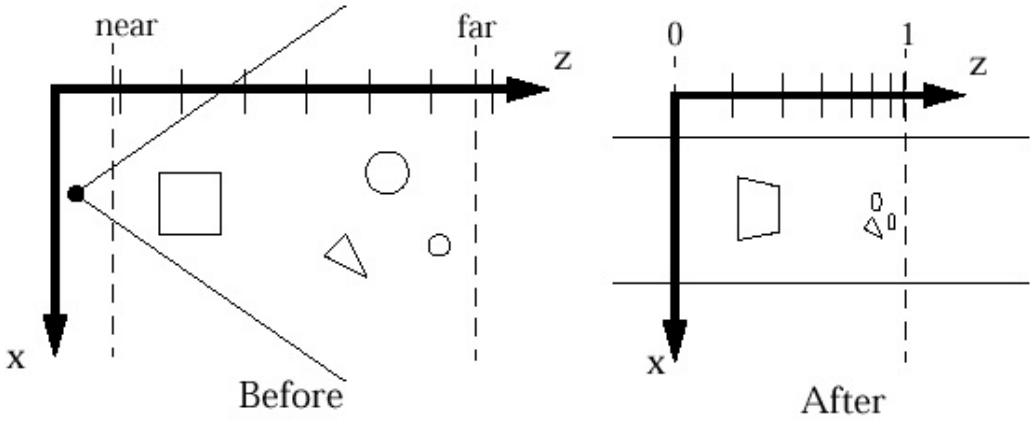
Z-Buffer

- Xay döng 2 buffer:
  - Intensity Buffer: löu mau cac pixel (init bang mau nen)
  - Depth (“Z”) Buffer: löu ño sau (init bang ño sau max).
- “Ve” töng polygon:
  - Neu ño sau cua ñiem tren polygon nho hôn ño sau töông öng ñang löu trong Z-Buffer thì cap nhat lai Z-Buffer va Intensity Buffer.



- Cac öu ñiem cua Z-Buffer
  - Thích hôp cai ñat tren phan cöng.
  - Ta co the scan-convert cac polygon theo thö tö bat ky.
  - Moi lan ta chæ phai xet mot polygon
  - Cho phep tong hôp nhieu canh vôi nhau hoac bo sung cac ñoi töông môi vao mot canh phöc tap.
  - Co the ap dung vôi cac mat cong, cac mat khong co dang ña giac.

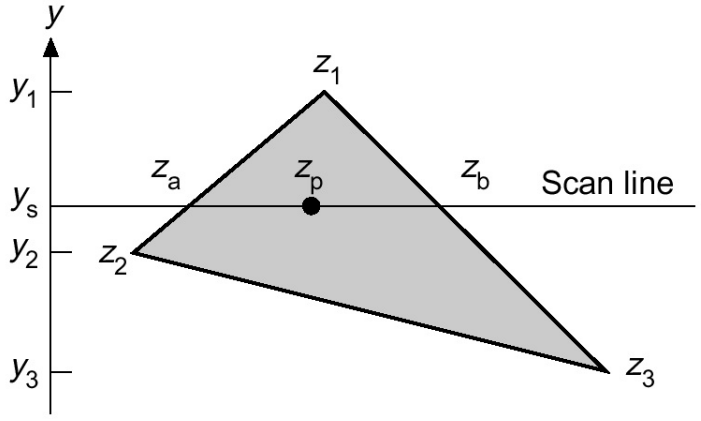
- Cac nhöôc ñiem cua Z-Buffer
  - Ñoi hoi bo nhô rat lôn
  - Co the mat chính xac khi chuan hoa trong qua trình tính ño sau.



  - Khong thöc hien ñöôc phep xö ly anti-alias
  - Phai scan-convert tat ca cac ñoi töông.

Lam the nao ñe tính toan Z-Buffer hieu qua

- Lay y töông cua phep to mau polygon (theo thuat toan scanline) khi tính giao ñiem cua scanline vôi cac canh cua polygon.
- Ta co the thöc hien töông tö ñe tính ño sau cho töng ñiem tren polygon:



$$z_a = z_1 - (z_1 - z_2)\frac{y_1 - y_s}{y_1 - y_2}$$

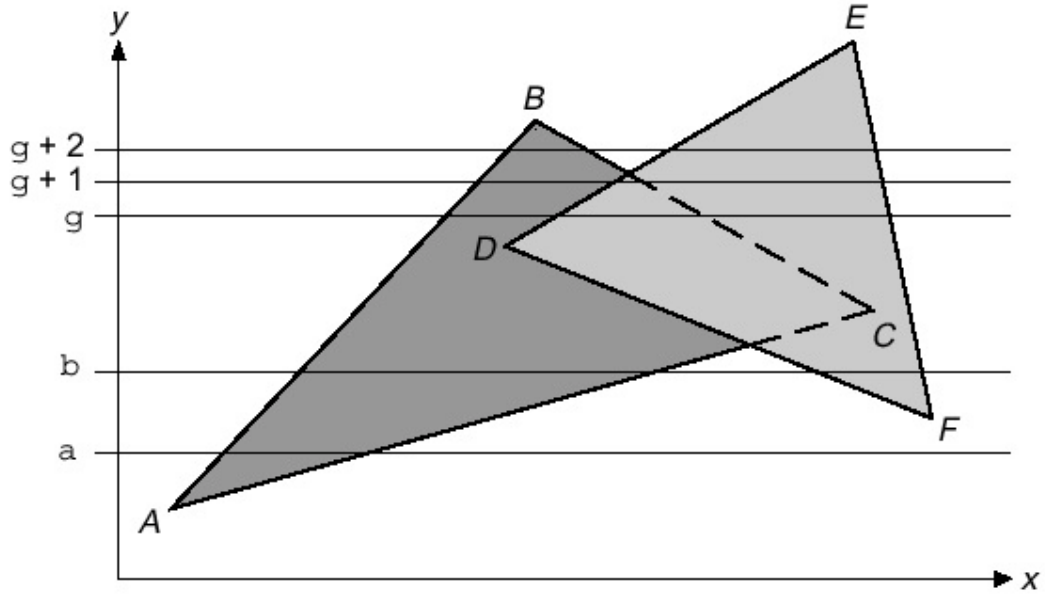
$$z_b = z_1 - (z_1 - z_3)\frac{y_1 - y_s}{y_1 - y_3}$$

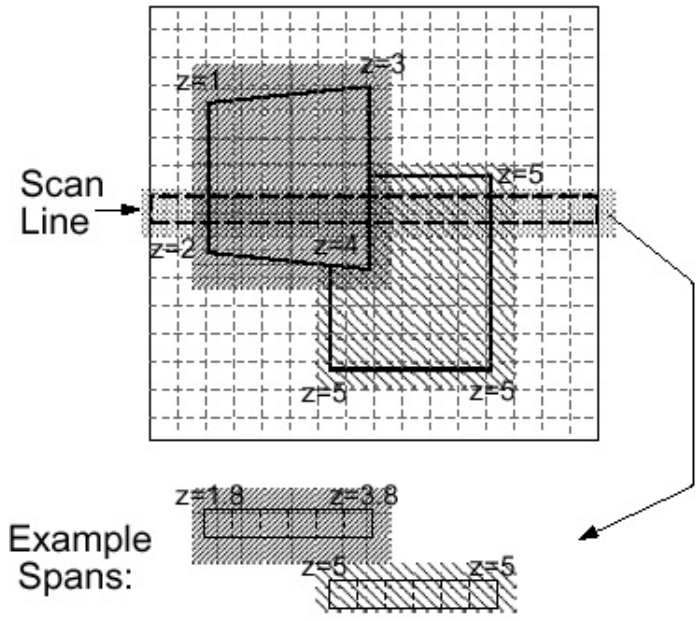
$$z_p = z_b - (z_b - z_a)\frac{x_b - x_p}{x_b - x_a}$$

- Khi ña co z_a va z_b vôi moi canh, ta co the tính z_p tuan tö

Scan-line

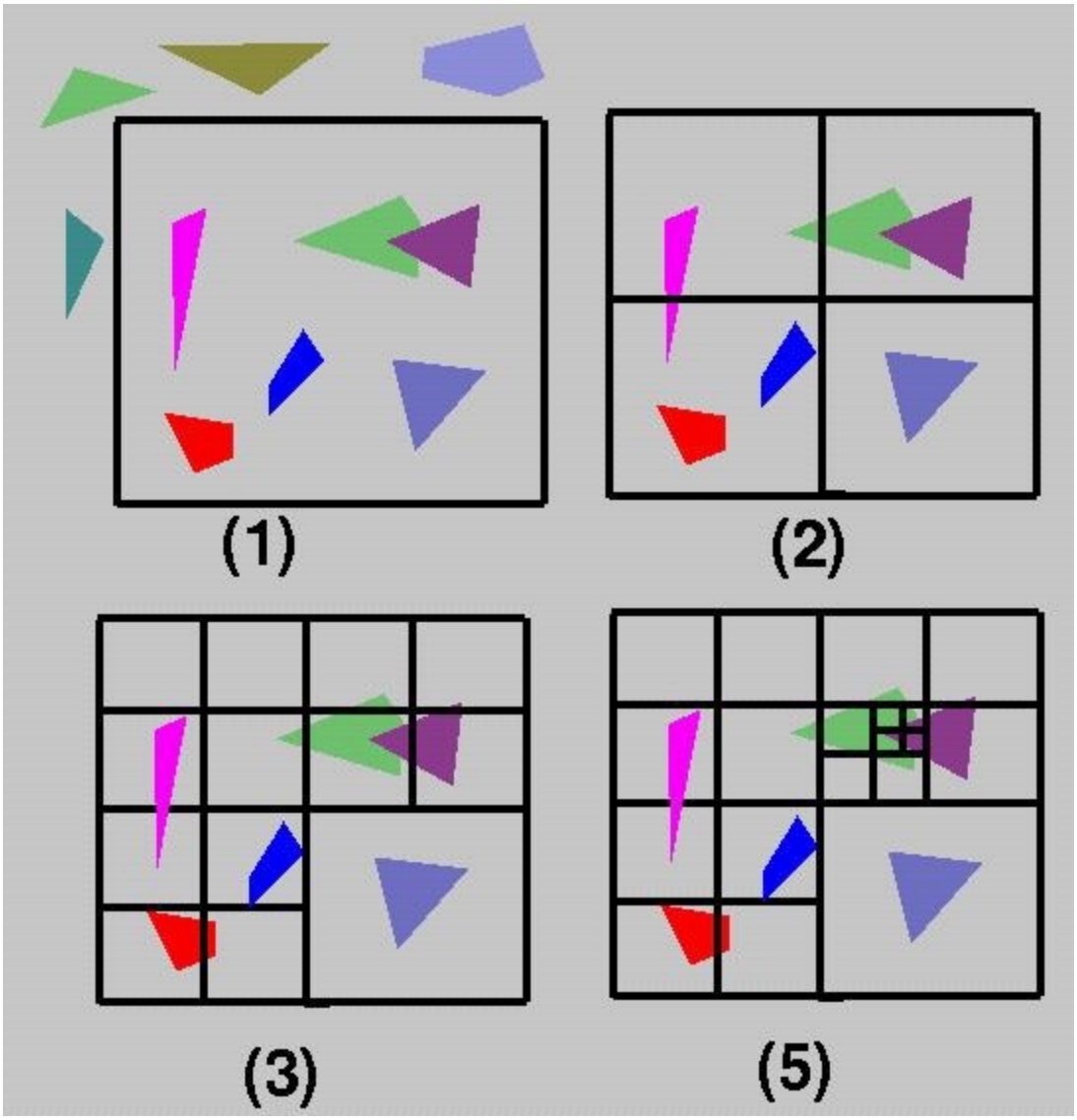
- Mô rong tö töông cua thuat toan to mau scanline.
- Quet scanline doc theo VP.
- Vôi moi scanline xac ñònh cac ñoan öng vôi cac mat:
  - Xac ñònh cac giao ñiem cua scanline vôi cac ñöông bien.
  - Sap xep cac giao ñiem theo thö tö tang dan cua x.
  - Vôi moi ñoan to bang 1 mau (cua mat gan nhat).





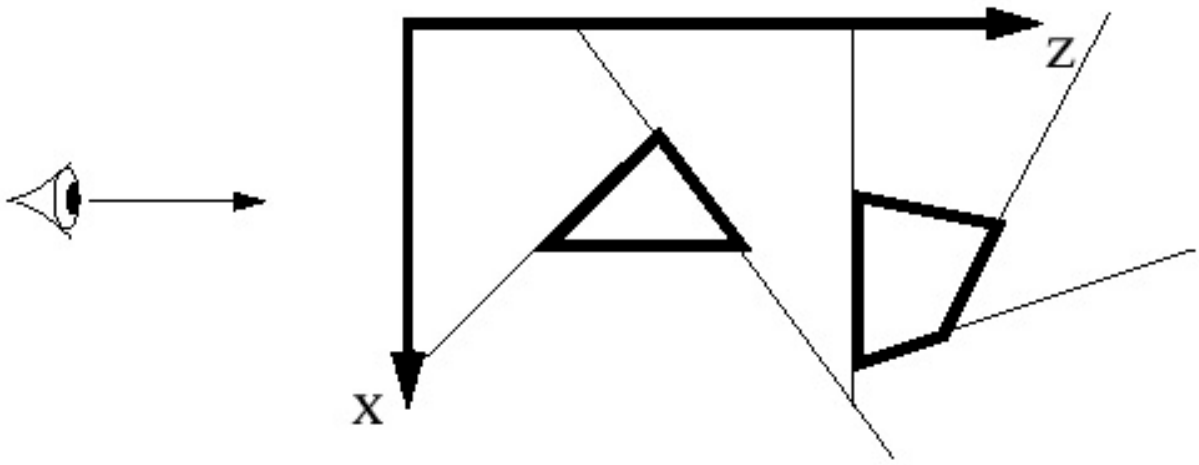
Warnock's Algorithm

- Bat ñau vôi vung la toan bo viewport
- To mot vung neu:
  - Khong co mat nao giao vôi no: mau nen.
  - Chæ co duy nhat 1 mat giao vôi no: ñôn gian
  - Co mot mat che khuat tat ca cac mat khac trong vung.
- Ngöôc lai: chia nho vung lam 4, tiep tuc qui trình vôi töng vung con.

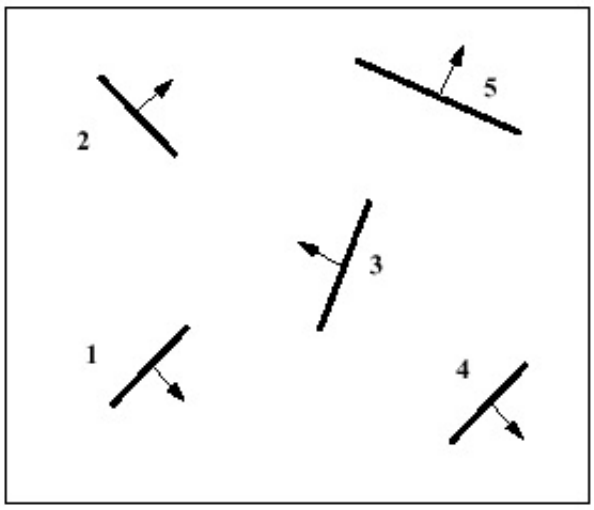


BSP Algorithm

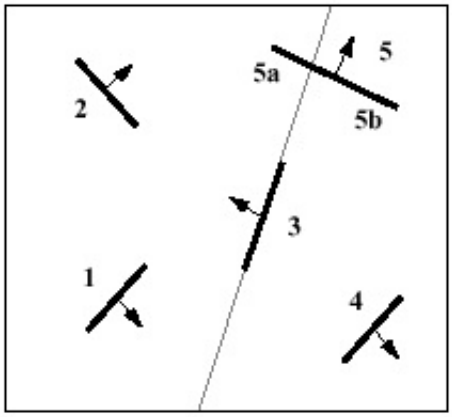
- BSP: Binary Space Partitioning
- Thuat toan cung cap mot qui trình chia nho khong gian va xac ñònh thöù töï ve cac ñoi töôïng.

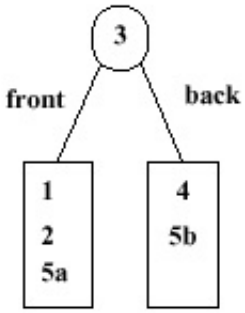


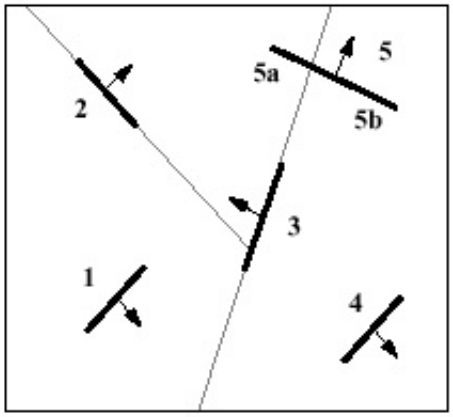
- Dung chien löôïc chia ñeå trò:
  - Ñeå hien thò cac polygon ñung, ta hien thò cac polygon ô phía "xa" tröôùc sau ño môi hien thò cac polygon ô phía "gan". (Gan la phía chöa View point)
  - Ñeå hien thò ñung cac polygon ô cung mot phía, ta chon mot polygon bat ky lam chuan chia va xöû ly ñeä qui.
- Xet mot ví du:

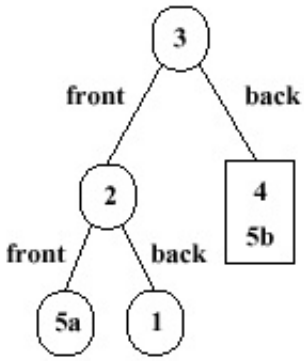


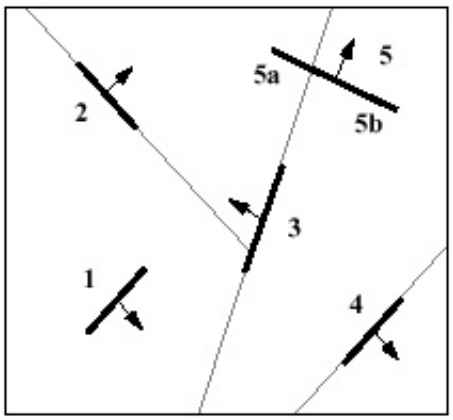
- Bat ñau tö mat so 3:

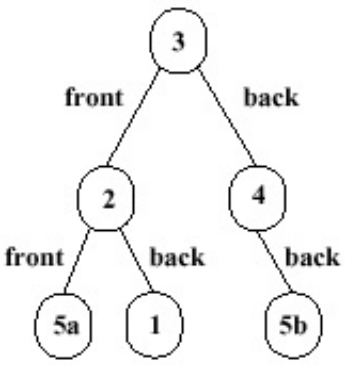




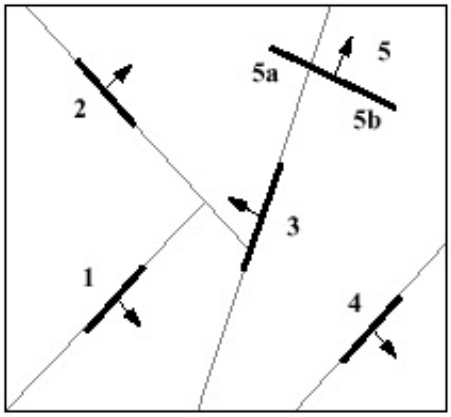


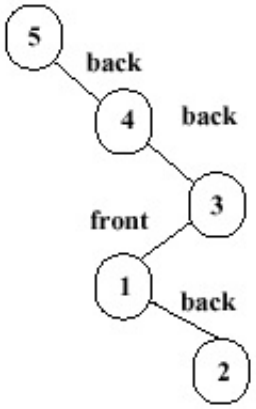






- Neu bat ñau tö mat so 5 ta co ket qua khac:





```
void BSP_displayTree(BSP_tree* tree)
{
   if ( tree is not empty )
      if ( viewer is in front of root ) {
      BSP_displayTree(tree->backChild);
      displayPolygon(tree->root);
      BSP_displayTree(tree->frontChild)
      }
      else {
         BSP_displayTree(tree->frontChild);
         /* ignore next line if back-face culling desired */
         displayPolygon(tree->root);
         BSP_displayTree(tree->backChild)
      }
}
```

Ket luan

- Hidden surface algorithms
  - Back-face detection
  - Depth sort
  - Ray casting
  - Z-buffer
  - Scan-line
  - Area subdivision (Warnock's)
  - BSP
- Hardware
  - Z-buffer
- Software
  - Depth sort
  - Scan-line