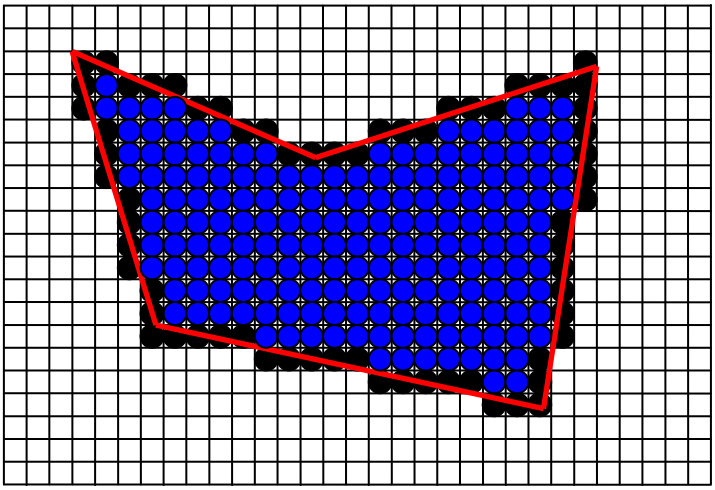
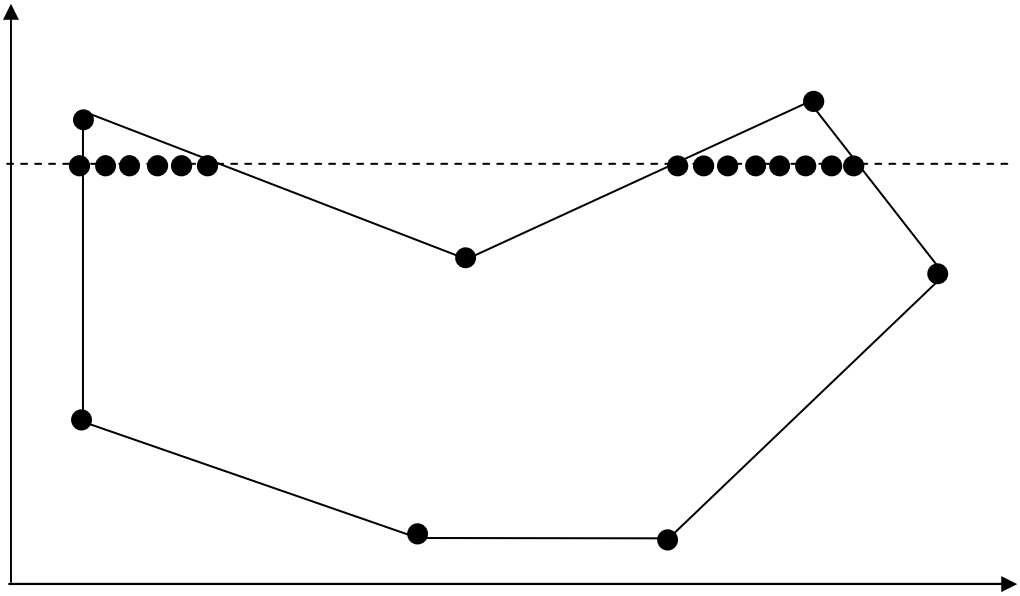
Generearea poligoanelor

Fie $P=[P_1P_2...P_n]$ un poligon pe ecran, unde coordonatele punctului $\mathbf{P_i}$ sunt $\mathbf{(x_i,y_i)}$, iar $P_1=P_n$.

Probleme:
1. Generarea (desenarea) interiorului poligonului P
2. Generarea (desenarea) conturului poligonului P - se generează fiecare segment

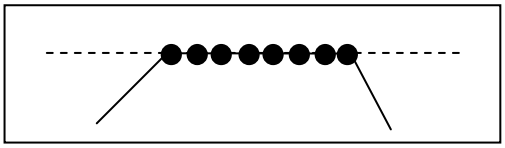


Metoda cea mai des folosita pentru generarea pixelilor este metoda de intersectare a poligonului cu toate liniile de baleiaj ale ecranului (toate liniile orizontale care intersectează poligonul). Algoritmul se numeşte *scan-line (algoritmul linie de baleiaj)*.

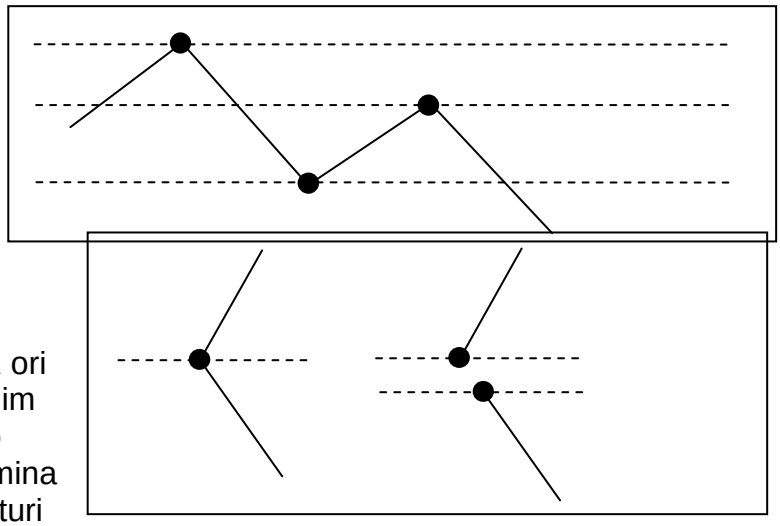


Fie $ymin=min\{y_1, y_2, ..., y_{n-1}\}$ şi $ymax=max\{y_1, y_2, ..., y_{n-1}\}$.
Se iau pe rând toate liniile de baleiaj cuprinse între ymin şi ymax. Pentru fiecare linie se parcurg paşii următori:
P1. Se determină intersecţiile cu toate laturile poligonului.

a. Dacă linia de baleiaj este paralelă cu latură, atunci nu se reţine niciun punct de intersecţie.



b. Dacă un punct apare de două ori (este un vârf al poligonului, deci linia de baleiaj intersectează în acel punct două laturi) şi acel punct este de minim sau maxim local, atunci se reţin ambele intersecţii.



c. Punctele care apar de două ori dar nu sunt puncte de minim sau maxim local se reţin o singură dată. Pentru a elimina o intersecţie, una dintre laturi se poate "scurta" cu 1 pixel pe verticală, şi în acest caz linia de baleiaj va intersecta o singură latură într-un astfel de vârf.

P2. Punctele de intersecţie se ordonează crescător după valoarea absciselor. Fie Q_1, Q_2, Q_3, ... aceste puncte după ordonare.

P3. Se reţin punctele (pixelii) între Q_1 şi Q_2, Q_3 şi Q_4, ...

Cel mai mult durează pasul P1. Pentru a scurta timpul de execuţie al algoritmului se pot folosi următoarele **observaţii**:

a. Dacă ecuaţia unei laturi neorizontale este: $y = mx + n$, atunci $x = \frac{1}{m}(y - n)$. Prin intersecţia cu linia de baleiaj $y = j$ se obţine $x_j = \frac{1}{m}(j - n)$, iar prin intersecţia cu linia următoare: $y = j + 1$ se obţine $x_{j+1} = \frac{1}{m}(j + 1 - n) = x_j + \frac{1}{m}$, care durează mai puţin decât efectuarea calculelor necesare obţinerii valorii prin intersecţia a două drepte.

b. La un moment dat nu toate laturile poligonului se intersectează cu linia de baleiaj. Ar fi util dacă o parte din laturi (cele care nu se intersectează) nu ar fi luate în considerare pentru linia respectivă de baleiaj.

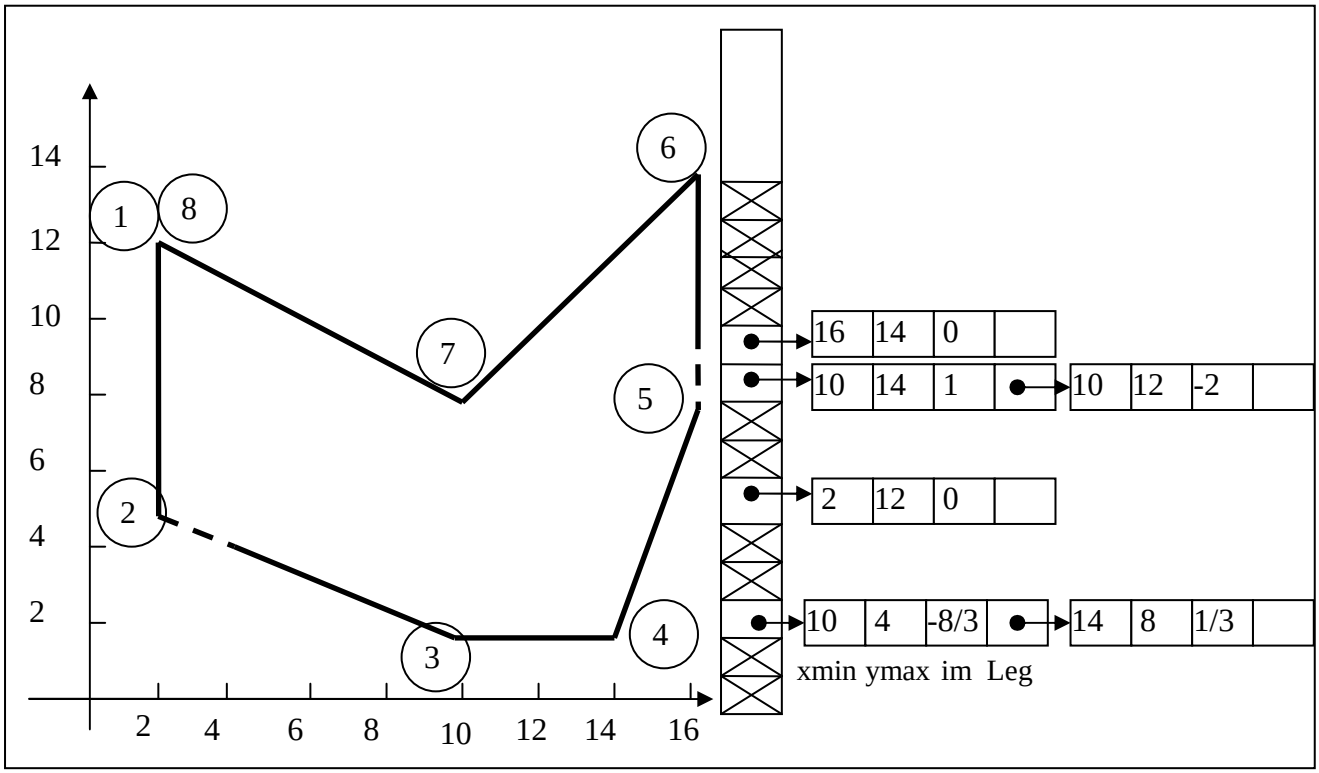
Pentru optimizarea algoritmului, datele se pot organiza sub forma următoare:

- Pentru fiecare latură a poligonului, delimitata de vârfurile (x_i, y_i) şi (x_{i+1}, y_{i+1}), care nu este orizontală, se determină următoarele date:
  - ymin = ordonata cea mai mică, dintre cele două ordonate (y_i şi y_{i+1});
  - xmin = abscisa corespunzătoare lui ymin;
  - ymax = ordonata cea mai mare;
  - im = valoarea 1/m, unde m este coeficientul unghiular al laturii

  O astfel de latură trebuie să fie luată în considerare la intersecţia cu linia de baleiaj începând cu ymin, nu va mai trebui luată în considerare atunci când linia de baleiaj depăşeşte valoarea ymax, iar la trecerea la următoarea linie de baleiaj abscisa unui punct de intersecţie dintre linia de baleiaj cu latura curentă se va modifica prin valoarea im.
- Toate datele de mai sus, cu aceeaşi valoare pentru ymin, se păstrează într-o listă înlănţuită. In cadrul listei elementele se pot păstra în ordinea crescătoare a valorilor

xmin. Datorită caracteristicilor comune a elementelor dintr-o astfel de listă înlănţuită (cu aceeaşi valoare pentru ymin), valoarea ymin se poate păstra a singură dată şi poate fi indicele şirului ce precizează adresa de început a primului element din lista înlănţuită corespunzătoare.

Pentru următorul poligon se obţin listele alăturate. Vom nota cu LM aceste liste înlănţuite (corespunzătoare muchiilor poligonului).



Obs. Pentru laturile 2-3 şi 5-6 s-a făcut scurtarea.

Pentru fiecare linie de baleiaj se va păstra, tot sub forma unei liste înlănţuite, cu aceeaşi structură ca mai sus, pe care o vom nota LL, datele despre muchiile ce intersectează această linie de baleiaj.

Algoritmul de mai sus se poate rescrie astfel:

1. Lista LL se iniţializează la o listă vidă
2. Fie y prima linie de baleiaj (cu cea mai mică valoare y) ce intersectează poligonul
3. Repetă paşii 3.1-3.5 până când lista LL şi listele LM devin vide.
   - 3.1. Datele din lista LM[y] se transferă în lista LL.
     In LL elementele se ordonează crescător după valorile xmin
   - 3.2. Pentru perechi de elemente: 1-2, 3-4, ... din lista LL se reţin valorile xmin şi se marchează pixelii, aflaţi pe linia de baleiaj y între aceste perechi de pixeli xmin.
   - 3.3. Din lista LL se elimină elementele pentru care y=ymax
   - 3.4. Pentru fiecare element din LL, valoarea xmin se înlocuieşte cu xmin+im, unde im este luat din acelaşi element cu xmin. După aceste modificări lista LL se ordonează crescător după valorile xmin.
   - 3.5. y:=y+1 (se trece la următoarea linie de baleiaj).