Inleiding Computational Geometry

Simpele polygonen

Een *polygoon* of veelhoek kan worden gerepresenteerd als een *linked list* van hoekpunten of zijden.

We spreken van een *simpel polygoon* wanneer twee zijden z_i en z_j elkaar enkel snijden als $i = j \pm 1$ plus de eerste en de laatste zijde en dan enkel in de hoekpunten. De zijden moeten een enkel gesloten pad vormen. Bij voorkeur elimineer je gestrekte hoeken. Simpele polygonen gedragen zich dus als de veelhoeken uit de meetkundeles.

Als de zijden van het polygoon een gesloten pad vormen, dan verdelen ze het vlak in twee delen: het eindige interieur en het oneindige exterieur. (dit wordt de Jordan Curve stelling genoemd).

Binnen of buiten?

Gegeven een simpel polygoon S en een punt P. Hoe test je nu of P in het interieur of exterieur ligt?

Trek een halve lijn vanuit P die niet parallel aan een van de zijden loopt en de zijden niet in een hoekpunt van S snijdt. Dan zal de lijn iedere keer dat ze een zijde snijdt overgaan van interieur naar exterieur of omgekeerd om in het exterieur te eindigen.

Als het aantal snijdingen oneven is dan ligt P in het interieur; is het even, dan ligt P in het exterieur.

Binnen of buiten?

De *winding number* methode werkt handiger:

Trek een halve lijn vanuit P die niet parallel aan een van de zijden loopt en de zijden niet in een hoekpunt van S snijdt. Loop alle zijden af en test of ze die halve lijn snijden. Als punt P nu links van de zijde ligt, tel 1 op bij het winding number, als het rechts ligt trek je 1 af.

Als het resultaat ±1 is, dan ligt P binnen de polygoon, als het 0 is ligt het erbuiten, en hogere waarden betekenen een niet-simpel polygoon.

Diagonalen

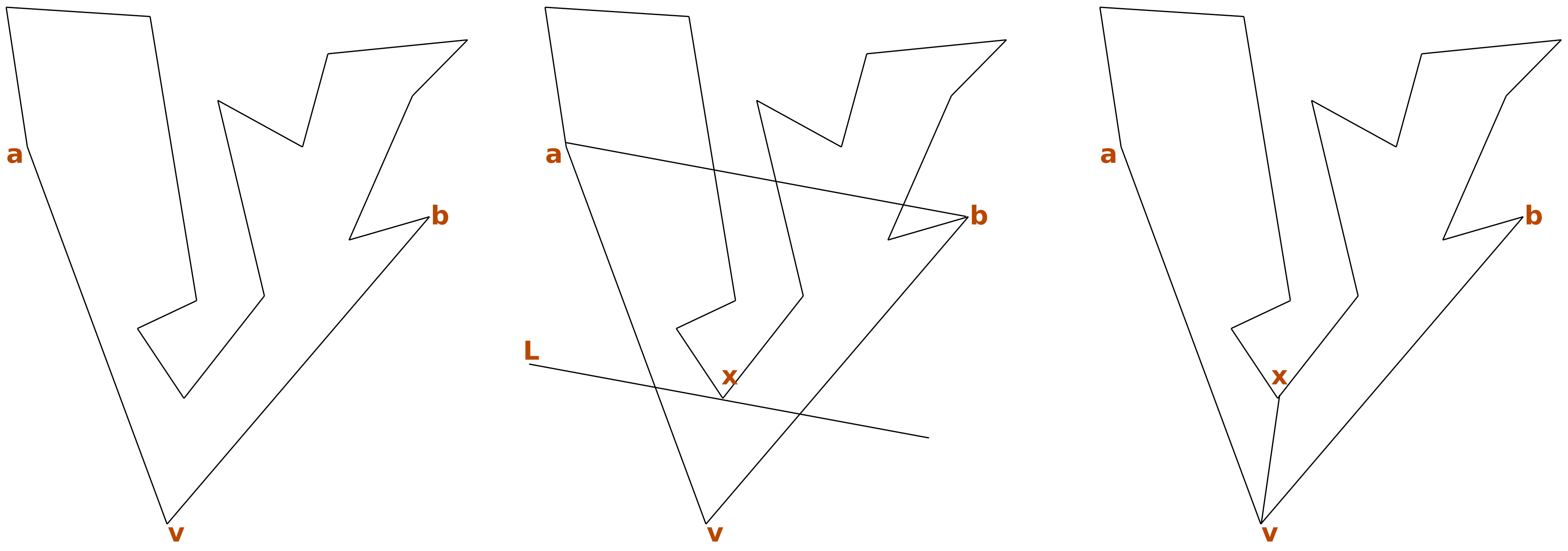
Een diagonaal van een veelhoek is een lijnstuk dat twee hoeken verbindt en afgezien van de uiteinden in het interieur van de veelhoek ligt.

Stelling: elke veelhoek met meer dan drie hoeken heeft een diagonaal.

hcc! programmeren

Inleiding Computational Geometry

Diagonalen



Diagonalen

Stelling: elke veelhoek met meer dan drie hoeken heeft een diagonaal.

Zie figuur vorige sheet. Laat *v* de onderste hoek zijn van de polygoon, of de meest rechtse daarvan. Laat *a* en *b* de hoekpunten zijn die aan *v* grenzen. Als het lijnstuk *ab* geheel in het interieur ligt, dan is het een diagonaal.

Zo niet, trek een lijn *L* evenwijdig aan *ab* en schuif die omhoog vanaf *v* totdat ze een ander hoekpunt, *x*, raakt. De driehoek gevormd door *L* en zijden *av* en *bv* bevat geen andere hoekpunten. Dan moet lijnstuk *vx* een diagonaal zijn.

Inleiding Computational Geometry

Triangulatie

Stelling: elke veelhoek heeft een triangulatie (opdeling in driehoeken).

Als het aantal hoeken *n* van polygoon *S* gelijk is aan drie, dan gaat de stelling op.

Neem nu aan dat $n > 3$ en dat de stelling geldt voor alle veelhoeken met minder dan *n* hoeken. Volgens de vorige stelling heeft S een diagonaal die ze verdeelt in veelhoeken S_1 en S_2. Die hebben minder dan *n* hoeken, dus volgens de inductiehypothese kunnen ze worden getrianguleerd. Volgens de Jordan curve stelling behoort het interieur van S_1 tot het exterieur van S_2 en omgekeerd, daarom vormen de triangulatie van S_1 en S_2 samen een triangulatie van *S*.

Inleiding Computational Geometry

Eigenschappen van triangulaties

Stelling: elke triangulatie van een veelhoek S met n hoeken bestaat uit n – 2 driehoeken en n - 3 diagonalen.

De stelling is triviaal geldig voor $n = 3$. Neem voor de inductie aan dat $n > 3$ en dat de stelling opgaat voor alle polygonen met minder dan n hoeken.

Kies nu een diagonaal d die hoekpunten a en b verbindt en de veelhoek verdeelt in S_1 en S_2 met resp. n_1 en n_2 hoeken. Omdat a en b in zowel S_1 als S_2 voorkomen, concluderen we dat $n_1 + n_2 = n + 2$. Aangezien S_1 $n_1 - 2$ driehoeken bevat en S_2 $n_2 - 2$ stuks, bezit S samen

$$(n_1 - 2) + (n_2 - 2) = (n_1 + n_2) - 4 = (n + 2) - 2 = n - 2$$

driehoeken. Analoog vinden we voor S $n - 3$ diagonalen.

Toepassingen van triangulaties

Stelling: De som van de inwendige hoeken van een simpel polygoon met *n* hoeken bedraagt $\pi \times (n - 2)$.

Volgens de vorige stelling kan een simpel polygoon worden getrianguleerd tot $(n - 2)$ driehoeken waarvan de hoeken samen $\pi \times (n - 2)$ radialen bedragen. Die hoeken vormen samen de inwendige hoeken van het polygoon.

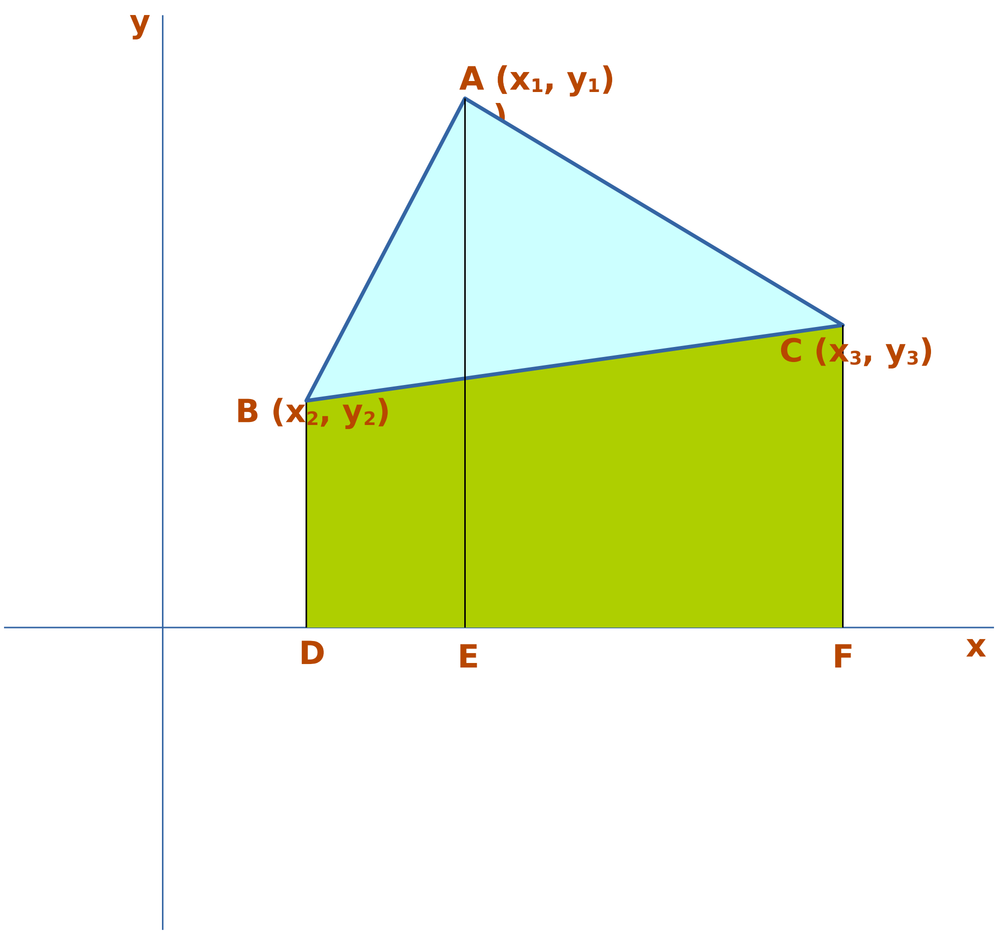
Inleiding Computational Geometry

Toepassingen van triangulaties

Stelling: De hoeken van een simpel polygoon vormen samen een bocht van 2 π. (een hoekpunt *p* vormt een *bocht* van π minus de inwendige hoek bij *p*).

Uit bovenstaande definitie volgt dat de totale bocht = $n \times \pi - \pi \times (n - 2) = 2\pi$.

De oppervlakte van een driehoek



Stelling A: De oppervlakte van een driehoek is gelijk aan de basis maal de halve hoogte.

Stelling B: de oppervlakte van een trapezium is gelijk aan de halve hoogte maal de som van de bases (parallelle zijden).

Bewijs: zie appendix

Inleiding Computational Geometry

De oppervlakte van een driehoek

Nu willen we de oppervlakte van de Δ ABC uitrekenen in termen van de coördinaten van de hoekpunten. We hebben een drietal loodlijnen neergelaten op de x-as die trapezia vormen.

Opp(Δ ABC) = Opp(BAED) + Opp(ACFE) – Opp(BCFD)

Opp(BAED) = ½ ×(BD + AE) × DE = ½ × $(y_2 + y_1) \times (x_1 - x_2)$

Opp(ACFE) = ½ ×(AE + CF) × EF = ½ × $(y_1 + y_3) \times (x_3 - x_1)$

Opp(BCFD) = ½ ×(BD + CF) × DF = ½ × $(y_2 + y_3) \times (x_3 - x_2)$

Opp(Δ ABC) = ½ ×$[(y_2 + y_1) \times (x_1 - x_2)]$ + ½ × $[(y_1 + y_3) \times (x_3 - x_1)]$ - ½ × $[(y_2 + y_3) \times (x_3 - x_2)]$

Opp(Δ ABC) = ½ ×$|x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)|$

hcc programmeren

Inleiding Computational Geometry

De oppervlakte van een veelhoek

Stelling: gegeven een simpele polygoon S met hoeken (x_i,y_i). De oppervlakte van S wordt dan gegeven door:

$$\mathrm{Opp}(S) = ½ \times |\Sigma(x_i y_{i-1} - x_{i-1} y_i)|$$

Voorbeeld: een vierhoek, die wordt verdeeld in twee driehoeken door diagonaal BD.

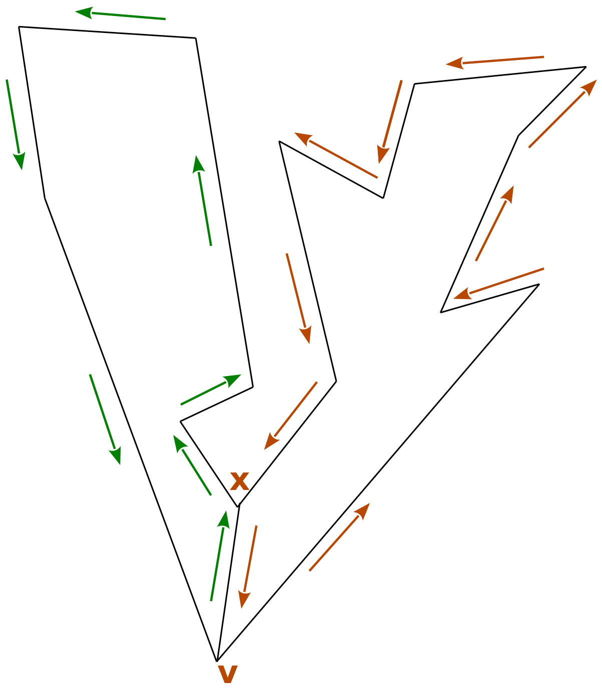
$$\mathrm{Opp}(\Delta\ ABD) = ½ \times |x_2y_1 - x_1y_2 + x_4y_2 - x_2y_4 + x_1y_4 - x_4y_1|$$

$$\mathrm{Opp}(\Delta\ BCD) = ½ \times |x_3y_2 - x_2y_3 + x_4y_3 - x_3y_4 + x_2y_4 - x_4y_2|$$

$$\mathrm{Opp}(ABCD) = ½|x_2y_1 - x_1y_2 + x_3y_2 - x_2y_3 + x_4y_3 - x_3y_4 + x_1y_4 - x_4y_1|$$

Inleiding Computational Geometry

De oppervlakte van een veelhoek (visualisatie)



Visualisatie van bewijs:

We tekenen de afzonderlijke termen als een pijl langs de corresponderende zijde.

Zoals we al zagen kan onze polygoon door de diagonaal xv worden gesplitst in twee veelhoeken. De groene resp. rode pijltjes geven de oppervlaktes van S_1 en S_2 aan.

Het is nu gemakkelijk te zien hoe de pijltjes langs de diagonaal elkaar opheffen en het resultaat de oppervlakte van de grote veelhoek is.

Inleiding Computational Geometry

Oren in triangulaties

Definitie: in een triangulatie vormt een drietal aangrenzende hoekpunten *a, b* en *c* een *oor* als *ac* een diagonaal is. Punt *b* heet dan de *punt* van het oor.

Stelling: elke veelhoek met meer dan drie hoeken heeft minstens twee oren.

Een vierhoek heeft een diagonaal, die ze in twee driehoeken verdeelt; beide zijn oren.

Kijk naar de polygoon uit de vorige sheet, die door diagonaal *vx* is verdeeld in veelhoeken S_1 en S_2, die volgens de inductiehypothese elk minstens twee oren bezitten. Daarvan verliest elk er door de samenvoeging maximaal een oor met punt in *v* of *x*. Volgens Bartjens zal de resterende polygoon dus minstens twee oren overhouden.

Inleiding Computational Geometry

Een algoritme voor triangulatie

Gezocht: een algoritme om een simpel polygoon te trianguleren.

Oplossing: zoek de hoekpunten af totdat je een oor hebt en snijd dat af, zodat het een van de driehoeken vormt.

Herhaal de procedure met de resterende hoeken, totdat je slechts een driehoek overhoudt, die eveneens tot de oplossing behoort.

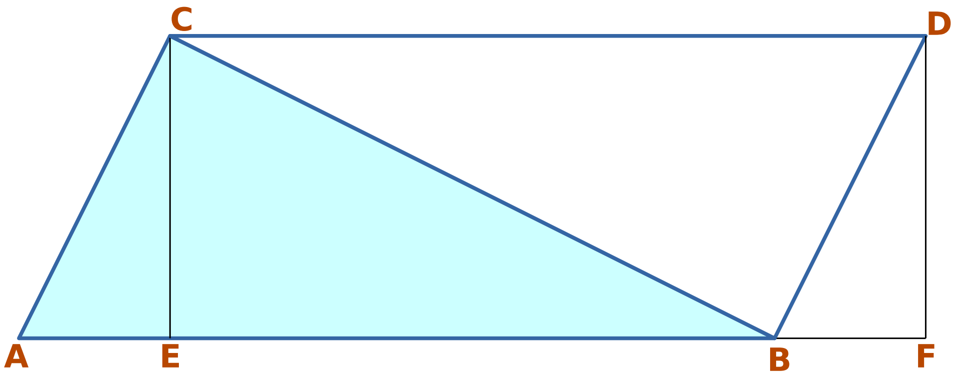
Volgens dit principe wordt het probleem opgelost in $O(n^2)$ rekentijd.

Bronvermelding

- https://press.princeton.edu/books/hardcover/9780691145532/discrete-and-computational-geometry
- https://www.cuemath.com/geometry/area-of-triangle-in-coordinate-geometry/
- https://en.wikipedia.org/wiki/Polygon_triangulation
- http://profs.ic.uff.br/~anselmo/cursos/CGI/slidesNovos/Inclusion%20of%20a%20Point%20in%20a%20Polygon.pdf

Inleiding Computational Geometry

Appendix A: Oppervlakte van een driehoek

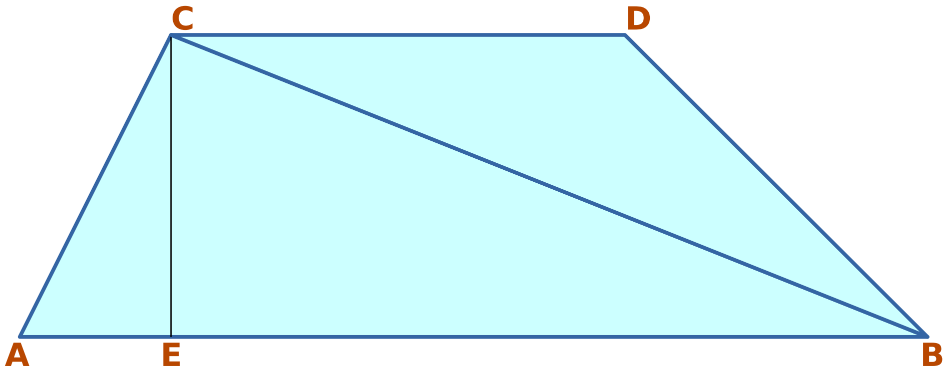


Gegeven driehoek ABC. Voeg nu de congruente driehoek BDC toe. Dan moet AC // BD en AB // CD; dus moet parallellogram ABDC 2 maal de oppervlakte van ABC hebben.

Laat nu loodlijn CE neer op zijde AB. Als je driehoek AEC verwijdert en driehoek BFD toevoegt, ontstaat een rechthoek met dezelfde oppervlakte $||AB|| \times ||CE||$. Dus moet gelden dat

$$\text{Opp}(ABC) = \tfrac{1}{2} \times ||AB|| \times ||CE||$$

Appendix B: Oppervlakte van een trapezium



Gegeven trapezium ABDC. Dit kan worden verdeeld door diagonaal BC in een tweetal driehoeken.

Volgens stelling A geldt dat

$$\text{Opp}(ABDC) = \text{Opp}(ABC) + \text{Opp}(BDC)$$

$$\text{Opp}(ABCD) = \tfrac{1}{2} \times \|CE\| \times (\|AB\| + \|CD\|)$$

einde van dit onderwerp