Nalezení jádra polygonální oblasti

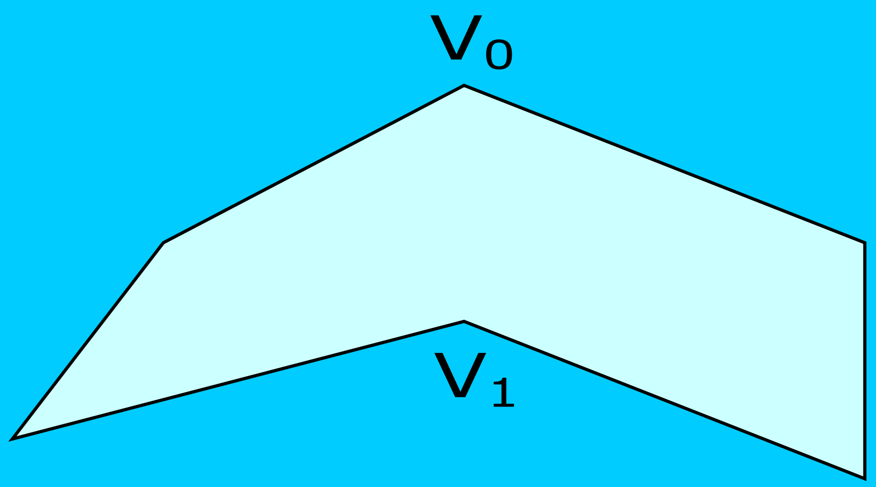
Zpracoval: Šafránek David

Slovník pojmů

hrana	edge	e
vrchol	vertex	v
polorovina	half-plane	H
polygon	polygon	P
jádro	kernel	K
ve směru hod. ručiček	clockwise	
protisměru hod. ručiček	counterclockwise	
Hvězdicovitý polygon	star-shaped , star polygon polygon s neprázdným jádrem	

Terminologie vrcholů

- v_0 konvexní, vypouklý (convex)
- vnitřní úhel < 180



- v_1 konkávní, vydutý, introspektivní (reflex)
- vnitřní úhel > 180

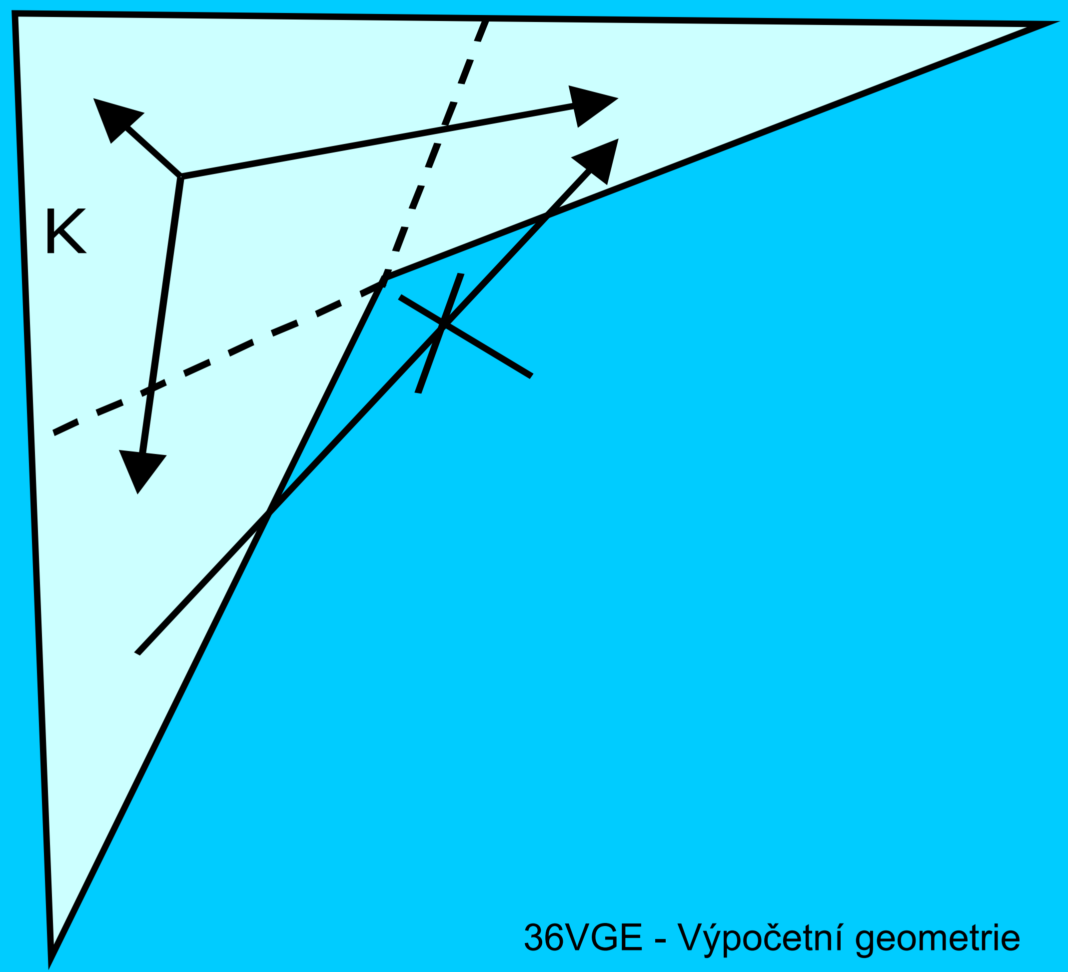
Problém

- máme polygon s N vrcholy v rovině, zkonstruujeme jeho jádro

Definice

- množina bodů odkud jsou všechny body polygonu přímo viditelné, aniž by výhled křížila hrana

Grafická ukázka definice



Odborná definice

Jádro je průnik N polorovin. Každá hrana polygonu P určuje polorovinu, ve které jádro musí ležet. Průnik vnitřních (levých - postupujeme-li proti směru hodinových ručiček) polorovin.

Pozice bodu vůči orientované přímce

- vlevo nebo vpravo
- a, b, c určuje přímku
- x a y určuje bod
- řešíme rovnici $w = a \cdot x + b \cdot y + c$
- $w=0$: leží na přímce
- $w < 0$: leží v pravé polorovině
- $w > 0$: leží v levé polorovině
- při zvolení opačného normálového vektoru (a, b) se změní w znaménko

Algoritmus

- Algoritmus v pořadí prochází vrcholy P a konstruuje konvexní polygony K_i. Každý z těchto polygonů může i nemusí být ohraničený.
- Polygon K_{i+1} je tvořen průnikem H_n $n=0..i+1$, a $i=0..N-2$ (N je počet vrcholů/hran počátečního polygonu).
- Takto zkonstruované K_i je společný průnik polorovin ležících vlevo od hran P.

Důsledky algoritmu

- Tento výsledek má zřejmé důsledky:
- 1) K_{N-1}=K(P) - aplikujeme na všechny hrany
- 2) K_1 nadmnožinou K_2 - $K_{i+1} = K_i$ "and" H_i

Základní algoritmus

- počítáme průnik poloroviny s polygonem
- výpočetně náročné

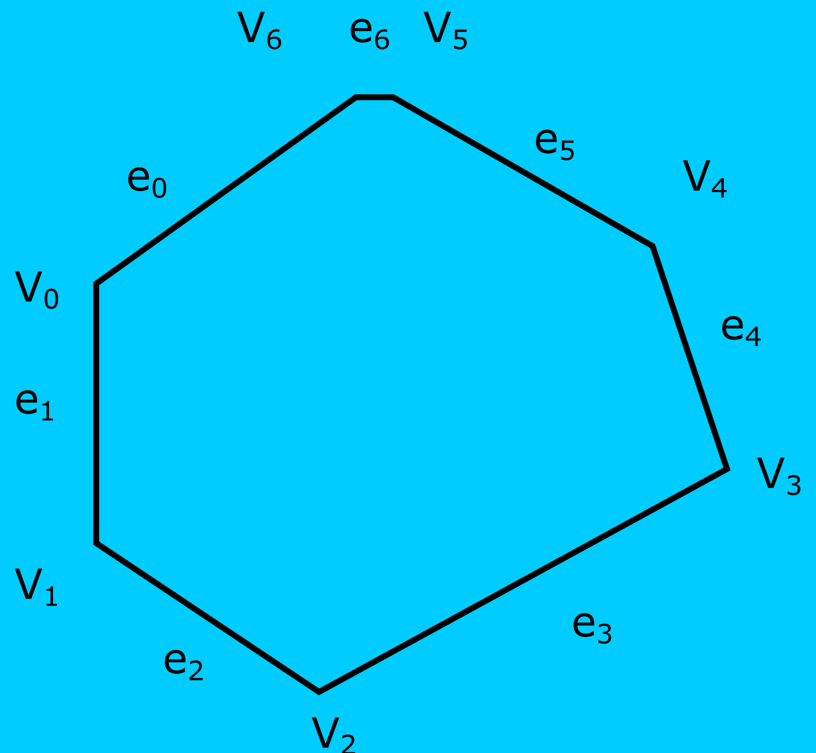
Chytrý algoritmus

- zrychlení pomocí vrcholů F a L, při průniku s polorovinou vymezují jen část jádra, tudíž neprohledáváme celé dosud nalezené jádro

Vstup

- polygon P - uspořádané vrcholy (hrany)
- e_i následované v_i

N=7



Výstup

- když bude K(P) neprázdný, pak výstup bude také ve formě posloupnosti vrcholů (hran)

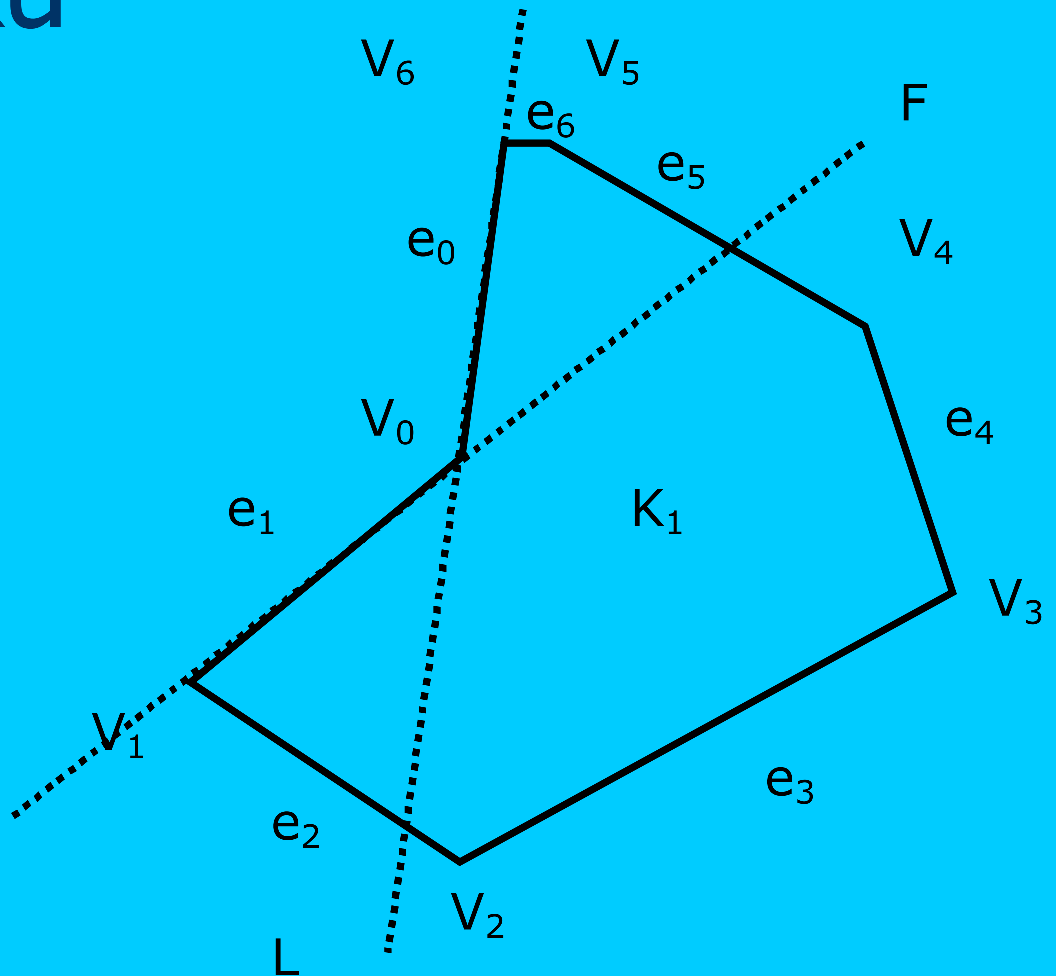
Triviální případ

- N < 3: není polygon
- N = 3: jádro je totožné s polygonem

Úvodní krok

- Nalezení konkávního vrcholu a vygenerování výchozího neuzavřeného jádra K_1. Není-li vrchol nalezen, pak polygon P je konvexní, jedná se o triviální případ - jádro je totožné s polygonem K(P)=P. Tudíž připustíme, že v_0 je konkávní vrchol.
- Určení F_1 jako bod v nekonečnu na přímce e_1 ve směru v_0 a L_1 obdobně na přímce e_0 ve směru v_0.

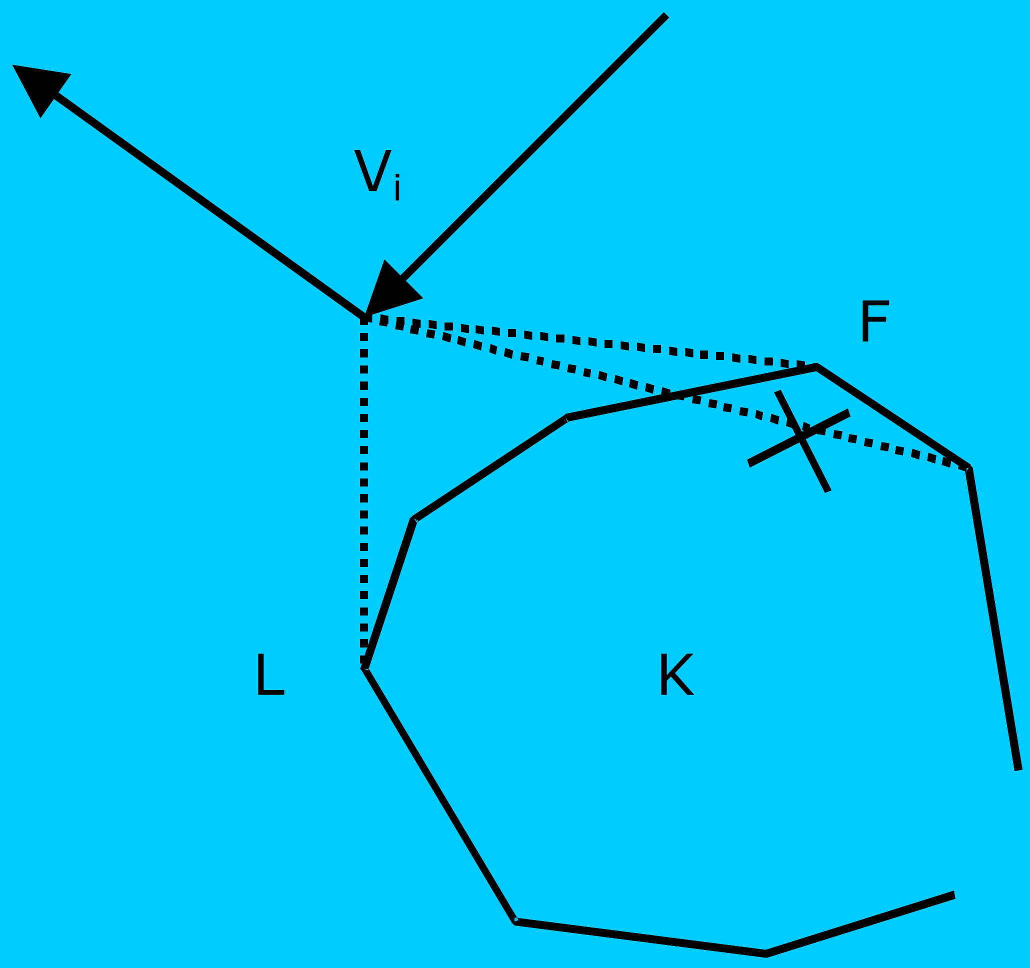
Grafická zobrazení prvního kroku



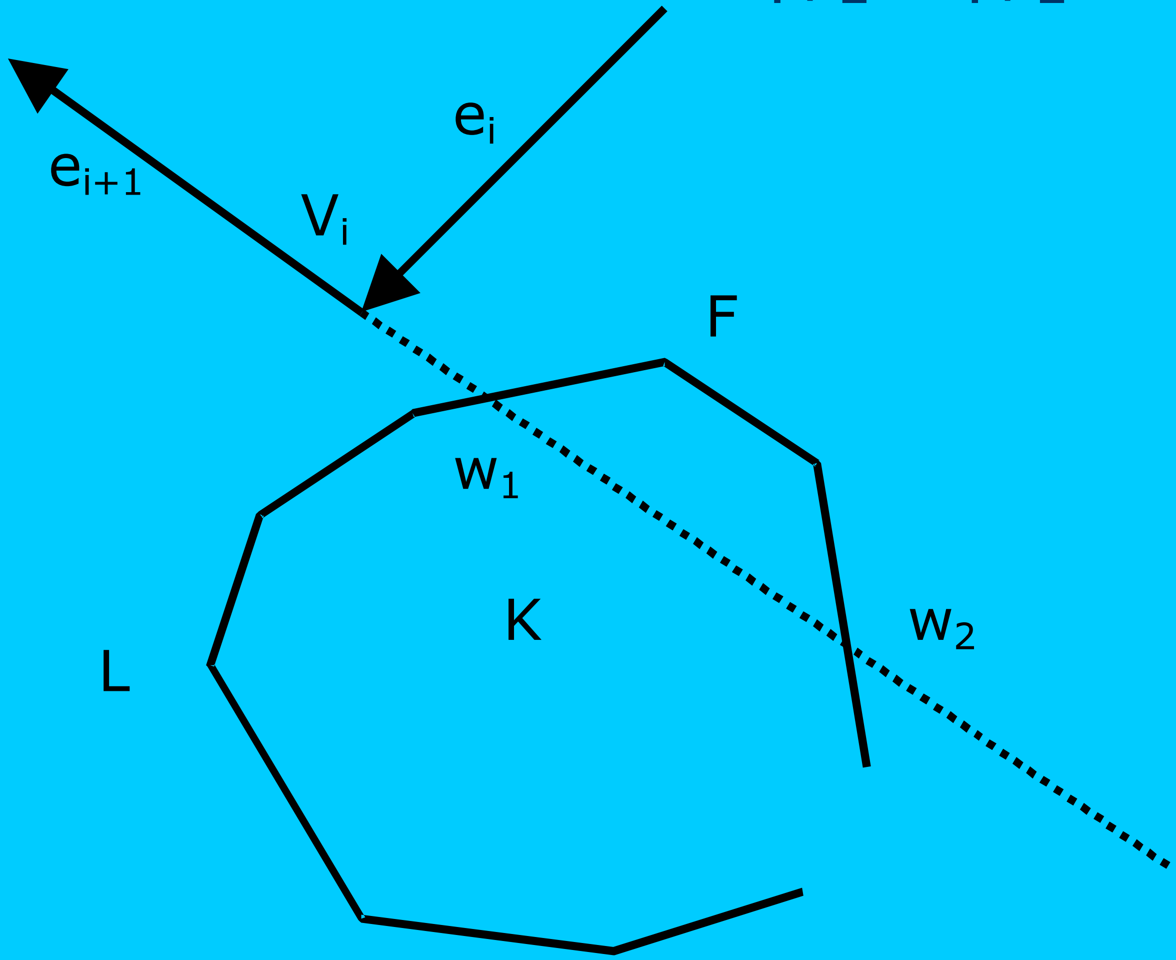
Hlavní krok

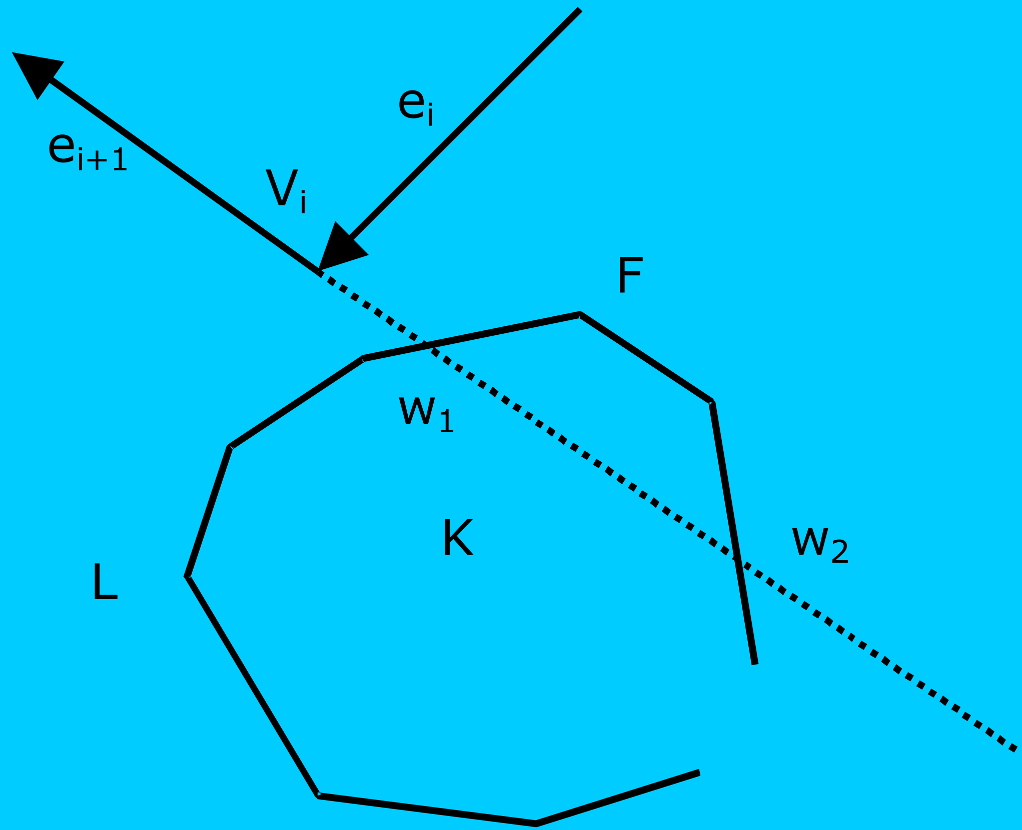
- máme-li K_i, pak zhotovíme K_{i+1} tak, že vezmeme další vrchol v pořadí (v_i) a spočteme K_{i+1}
- dále upravíme F_i a L_i tak, že spojnice s aktuálním bodem v_i polygonu neprotíná žádné jeho hrany
- detailní popis algoritmu obsahuje značné větvení (konvexní | konkávní vrchol, F (L) leží vpravo | vlevo)

Podmínka pro F a L

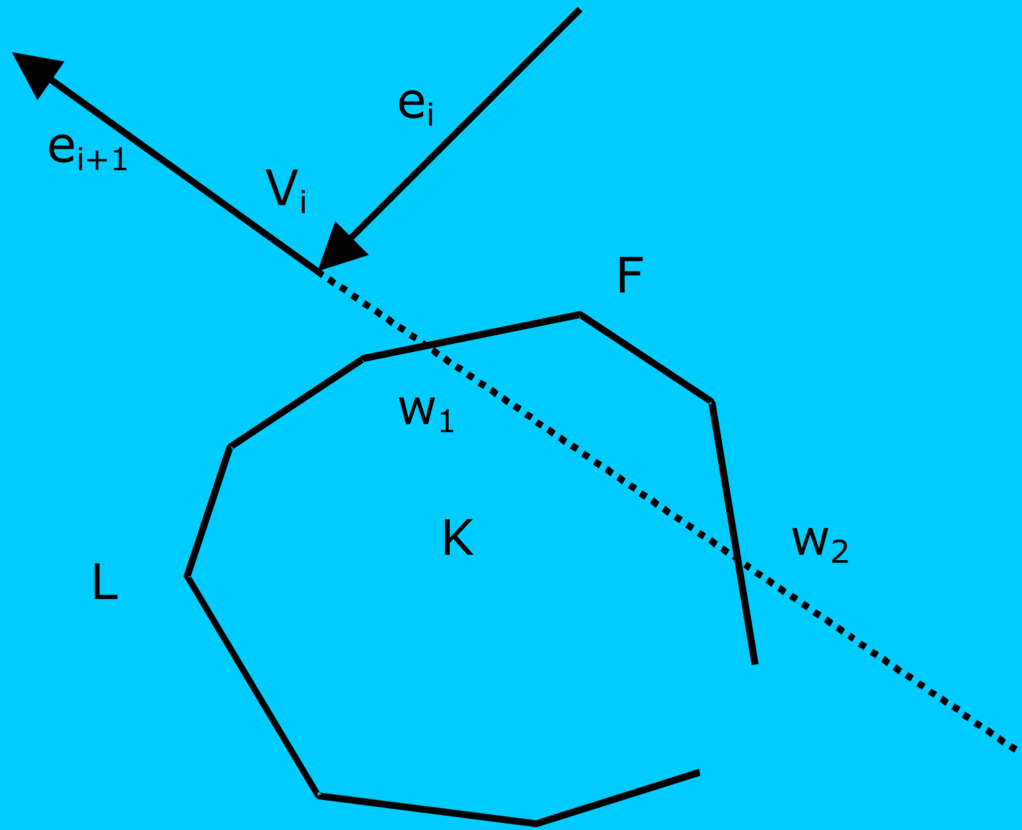


1.1 Konkávní vrchol, F leží vpravo od od e_{i+1} v_{i+1}

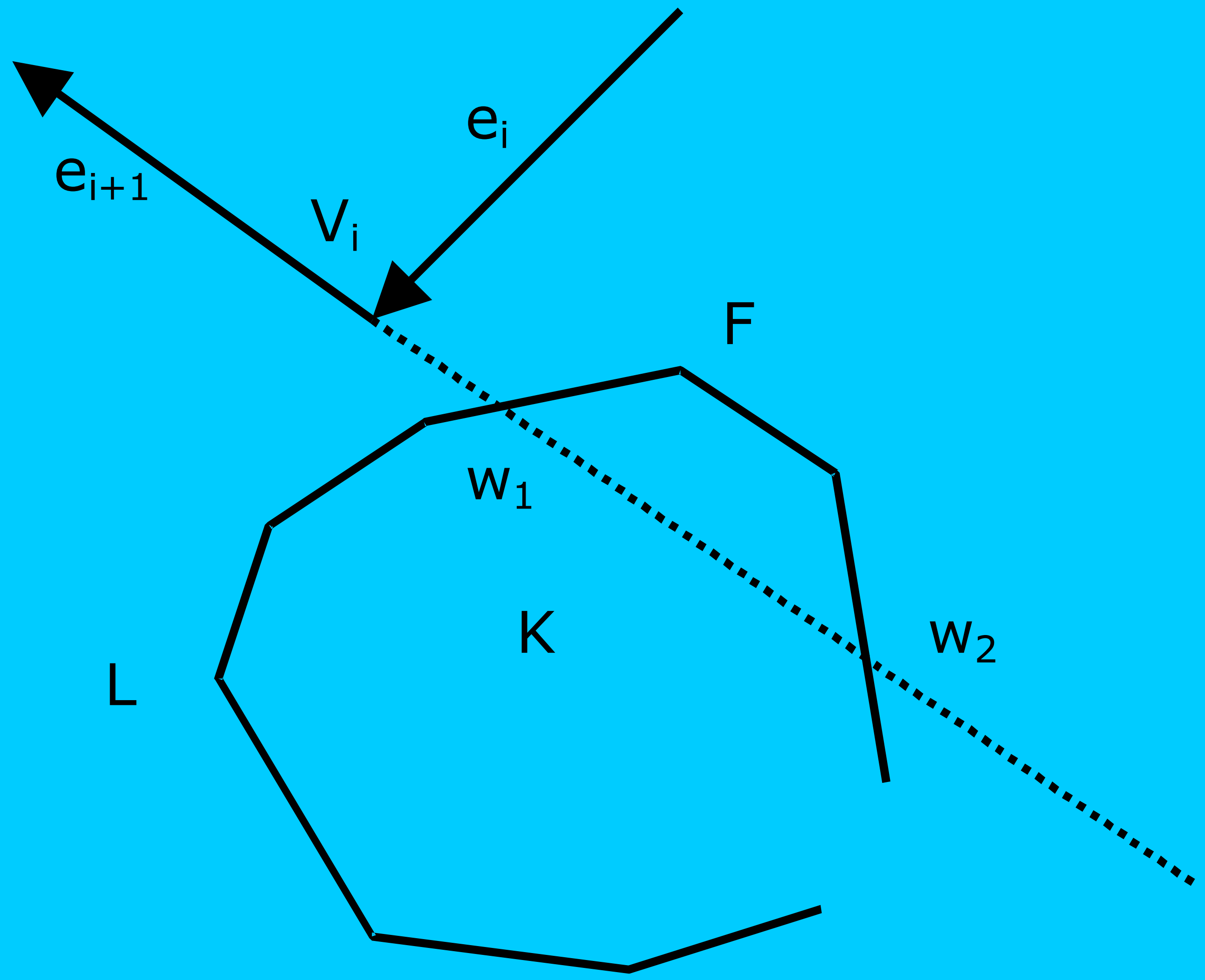




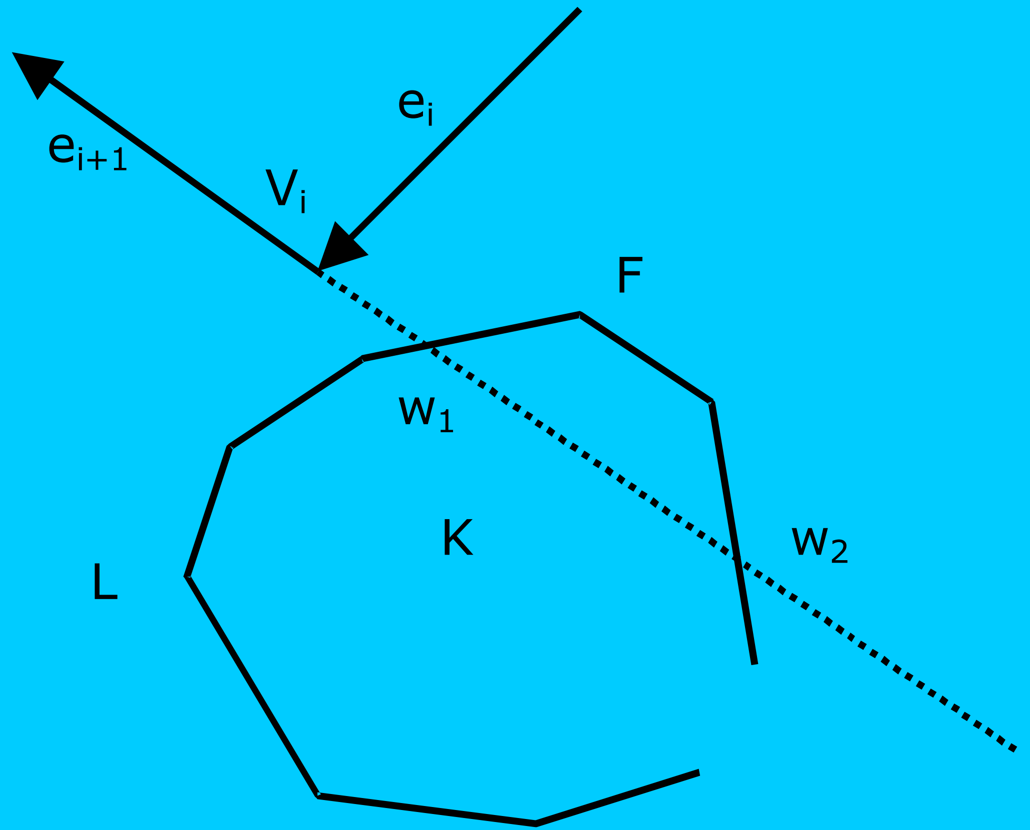
- procházíme K proti směru hodinových ručiček od F dokud nenalezneme průsečík (w_1) s e_{i+1} v_{i+1}, dojdeme-li až k L, pal je K(P)=Ø



- procházíme K po směru hodinových ručiček od F dokud nenalezneme průsečík (w_2) s e_{i+1} v_{i+1}, ořízneme K podle hrany w_1 w_2

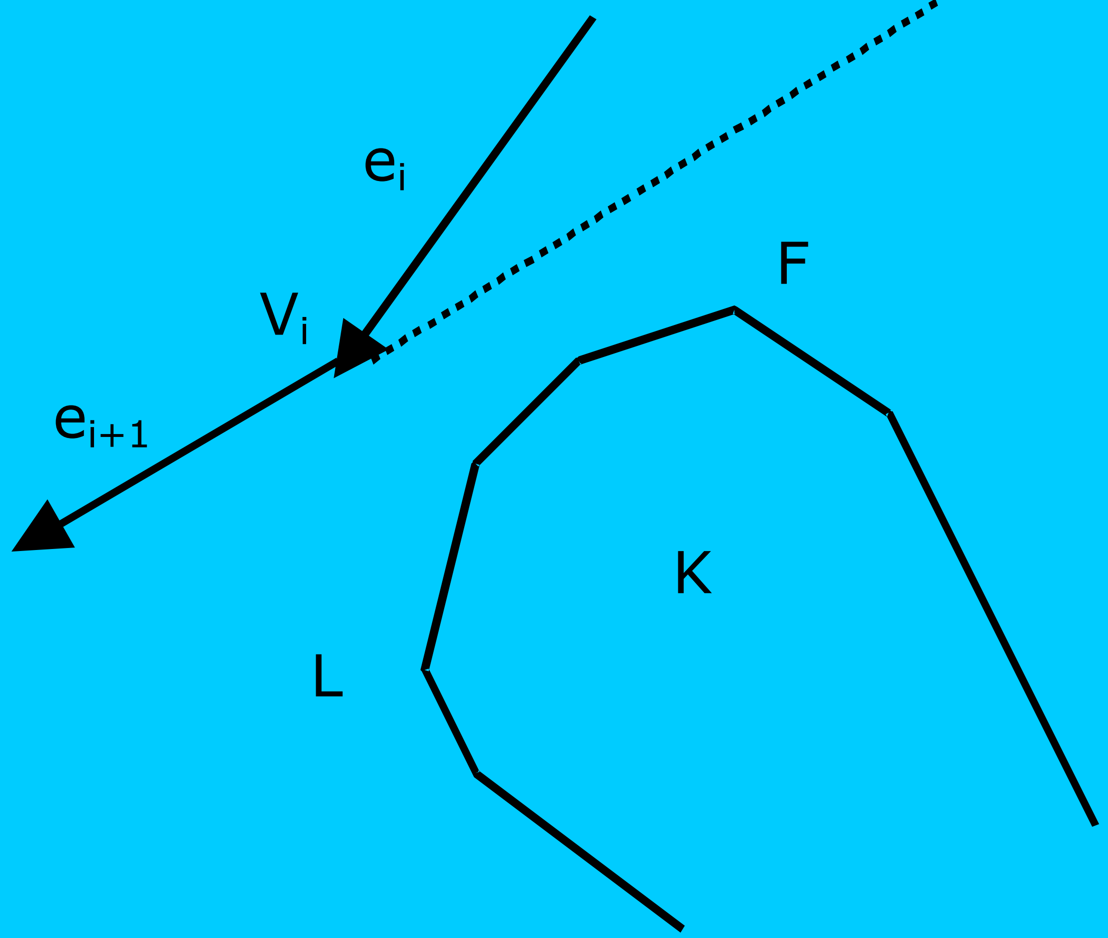


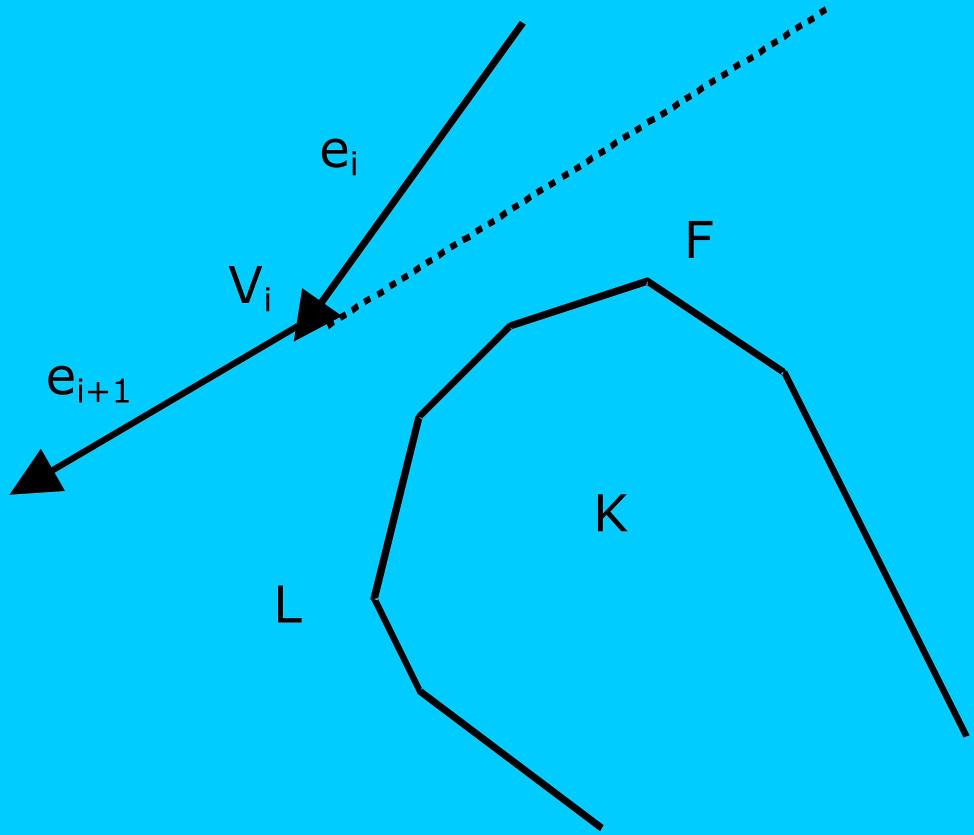
- F: je na konci hrany w_1 w_2



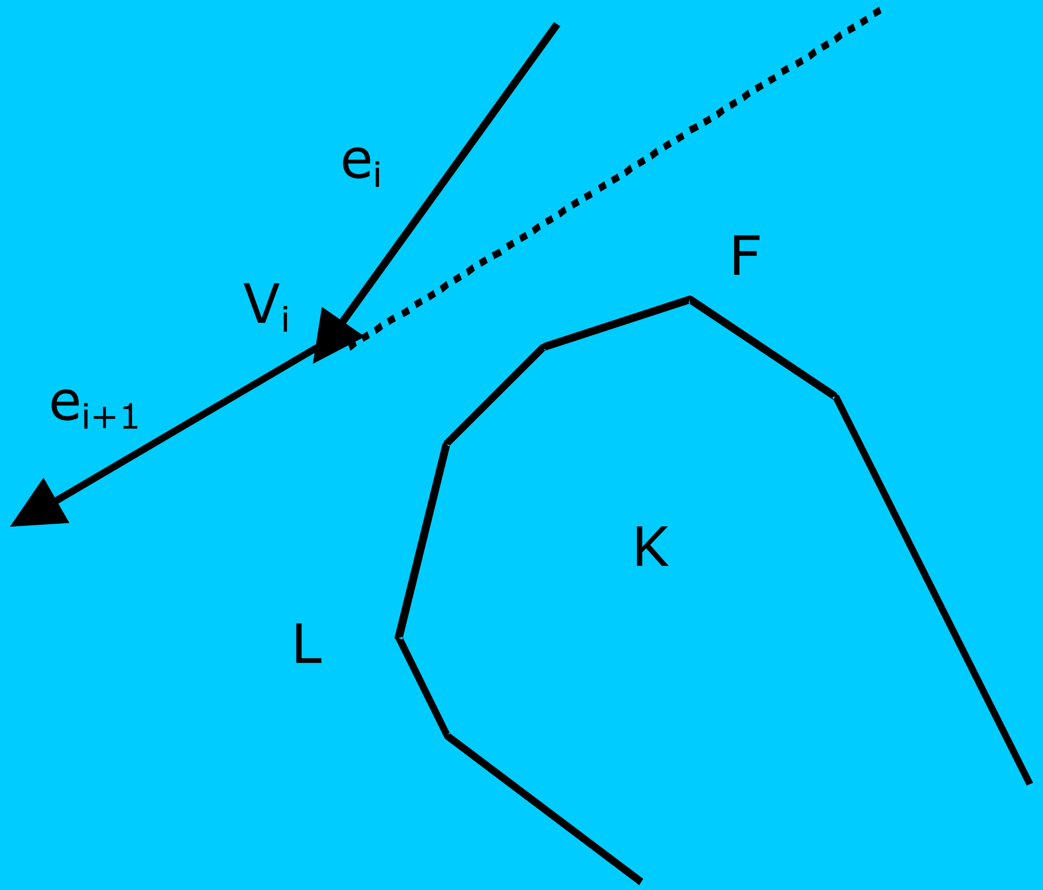
- L: procházíme K proti směru hodinových ručiček od L dokud nenajdeme vrchol w, takový, že následující vrchol leží vpravo od $v_{i+1}(v_{i+1}w)$, pak L_{i+1}=w jinak zůstává

1.2 Konkávní vrchol, F leží vlevo od od e_{i+1} v_{i+1}



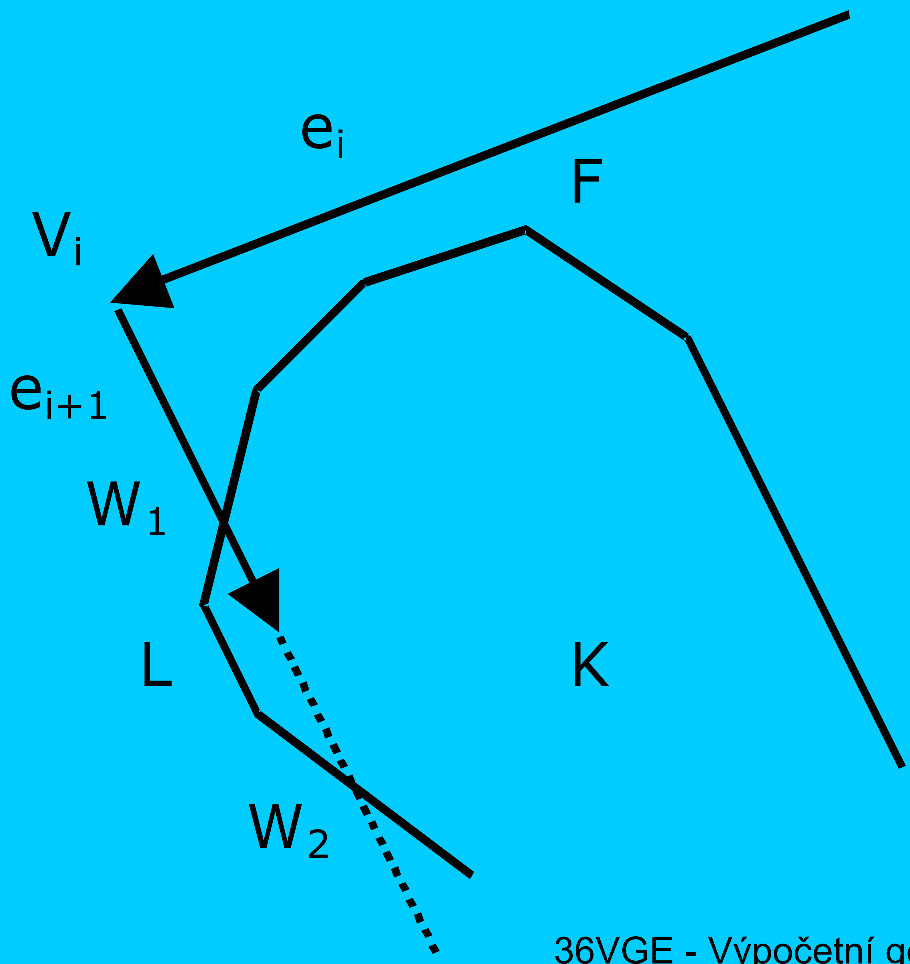


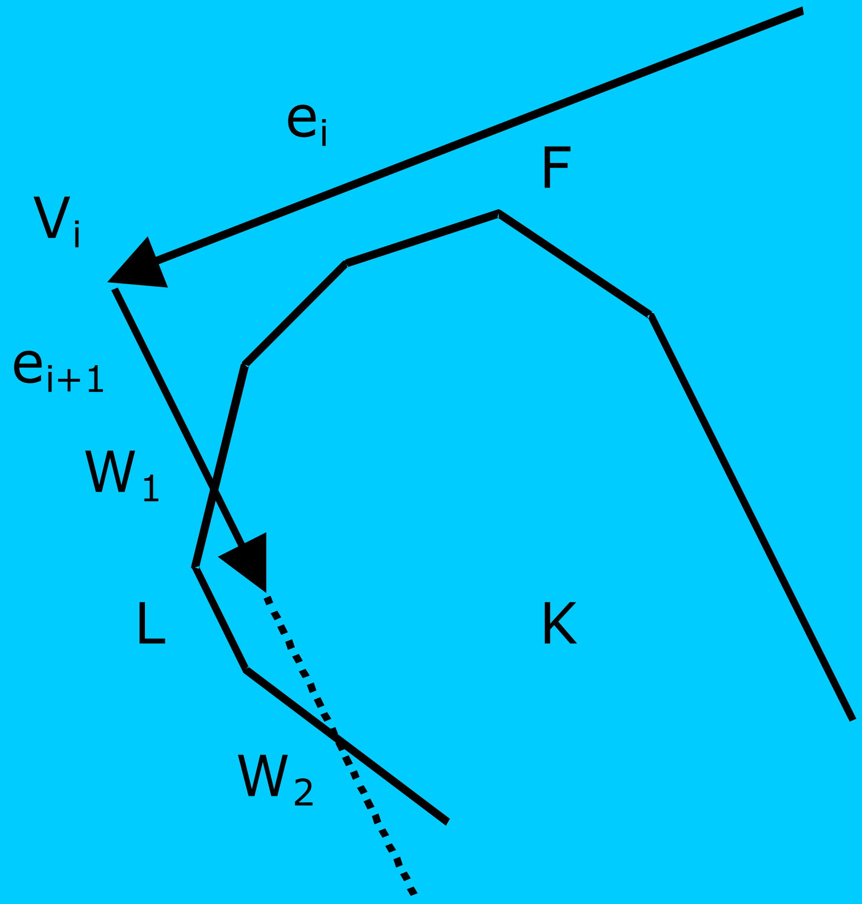
- K zůstává stejné.
- F: procházíme K protisměru hodinových ručiček od F dokud nenajdeme vrchol w, takový, že následující vrchol leží vpravo od $v_{i+1}(v_{i+1}w)$, pak F_{i+1}=w jinak zůstává



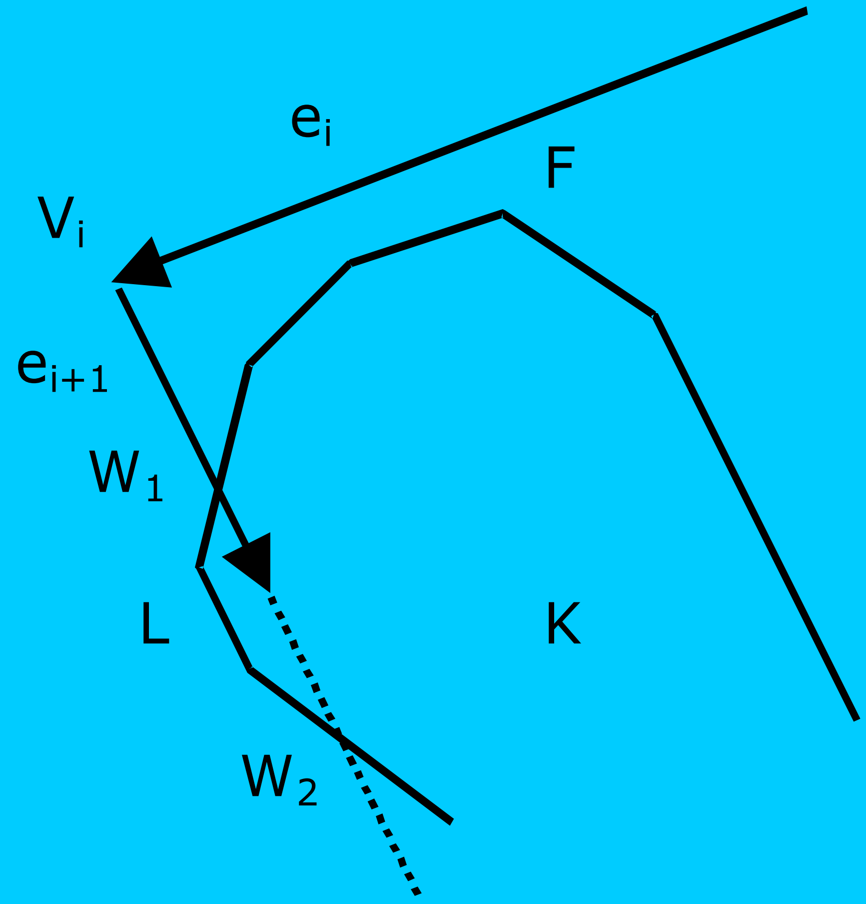
- L: stejně jako v případě 1.1
- 1.1: procházíme K proti směru hodinových ručiček od L dokud nenajdeme vrchol w, takový, že následující vrchol leží vpravo od $v_{i+1}(v_{i+1}w)$, pak $L_{i+1}=w$ jinak zůstává

2.1 L leží na nebo vpravo od e_{i+1} v_{i+1}



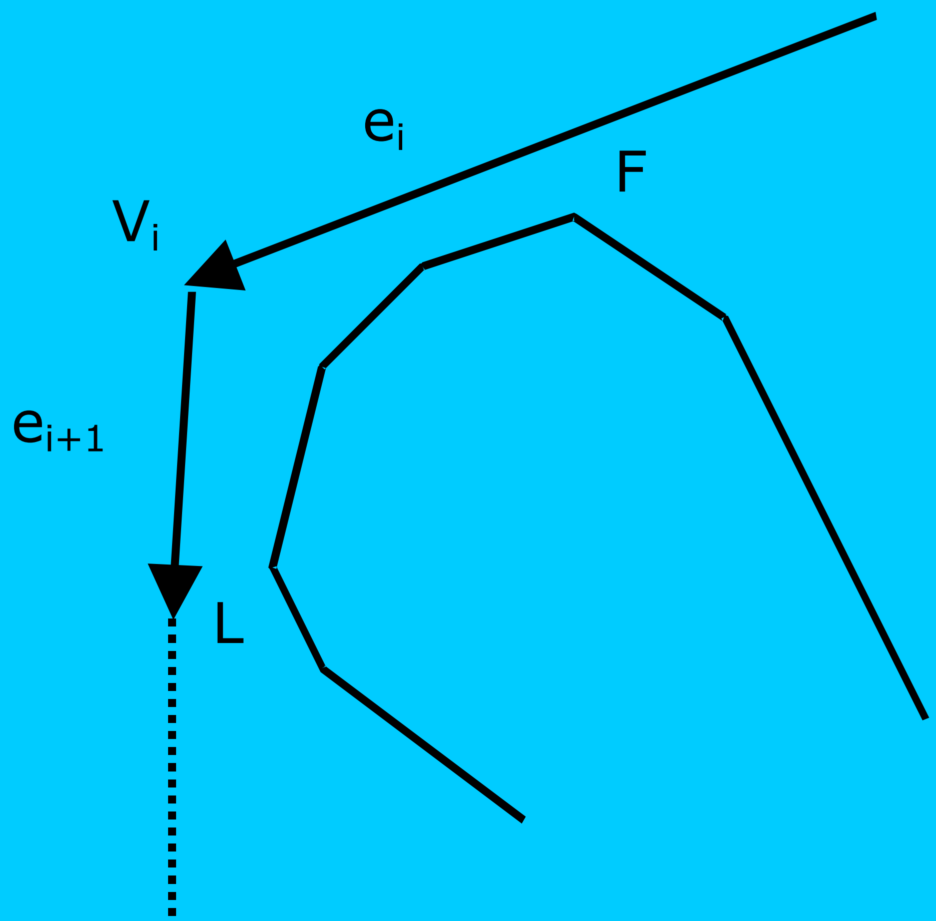


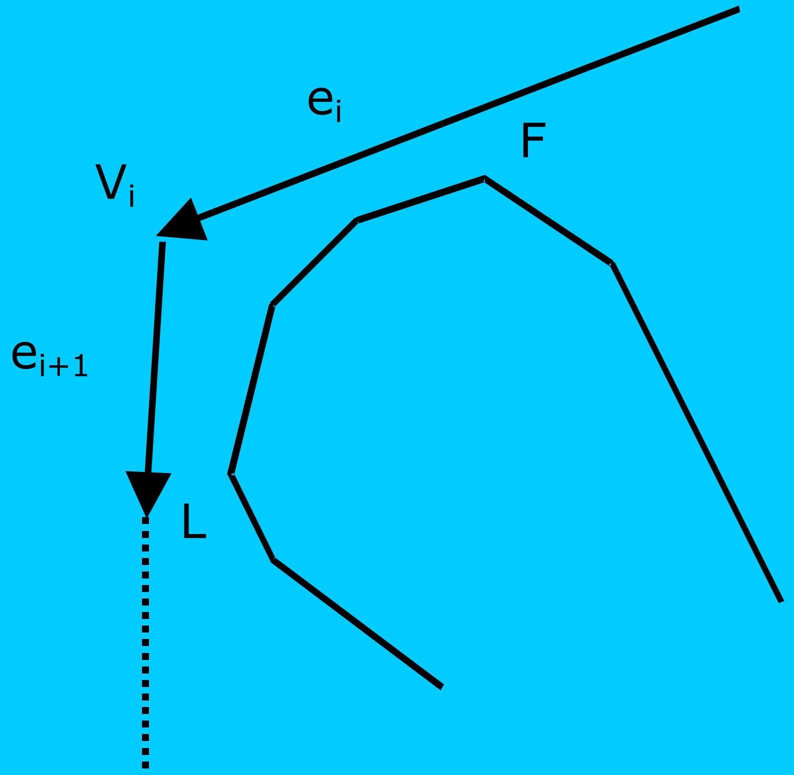
- procházíme K po směru hodinových ručiček od L dokud nenalezneme průsečík (w_1) s e_{i+1} v_{i+1}, dojdeme-li až k L je $K(P)=\varnothing$



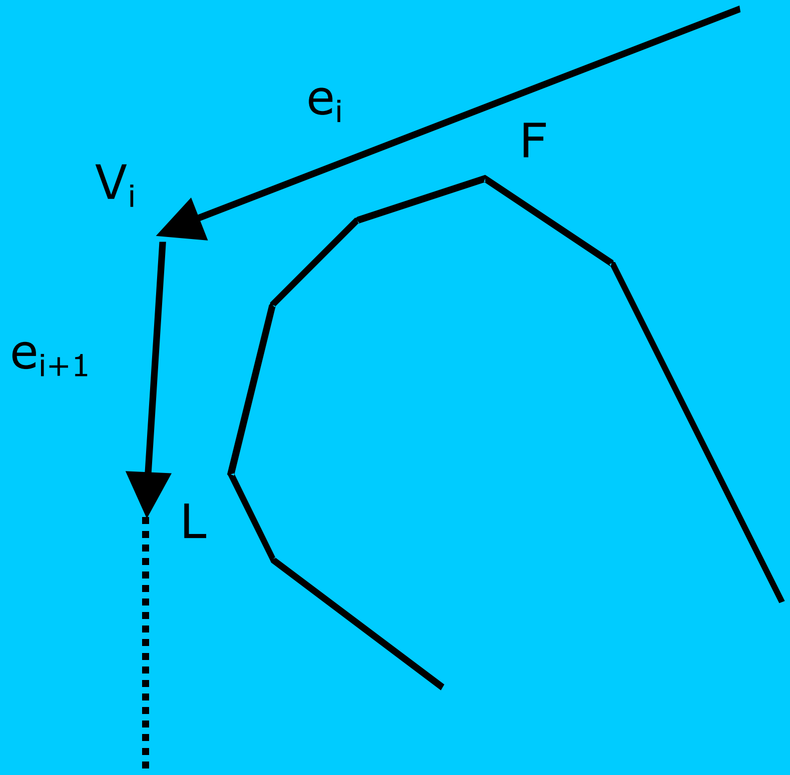
- procházíme K proti směru hodinových ručiček od L dokud nenalezneme průsečík (w_2) s v_i e_{i+1}. Provedeme oříznutí hranou w_1 w_2

2.2 L leží vlevo od e_{i+1} v_{i+1}





- K zůstává stejné.
- F: stejně jako v případě 1.2
- 1.2 procházíme K protisměru hodinových ručiček od F dokud nenajdeme vrchol w, takový, že následující vrchol leží vpravo od $v_{i+1}(v_{i+1}w)$, pak F_{i+1}=w jinak zůstává

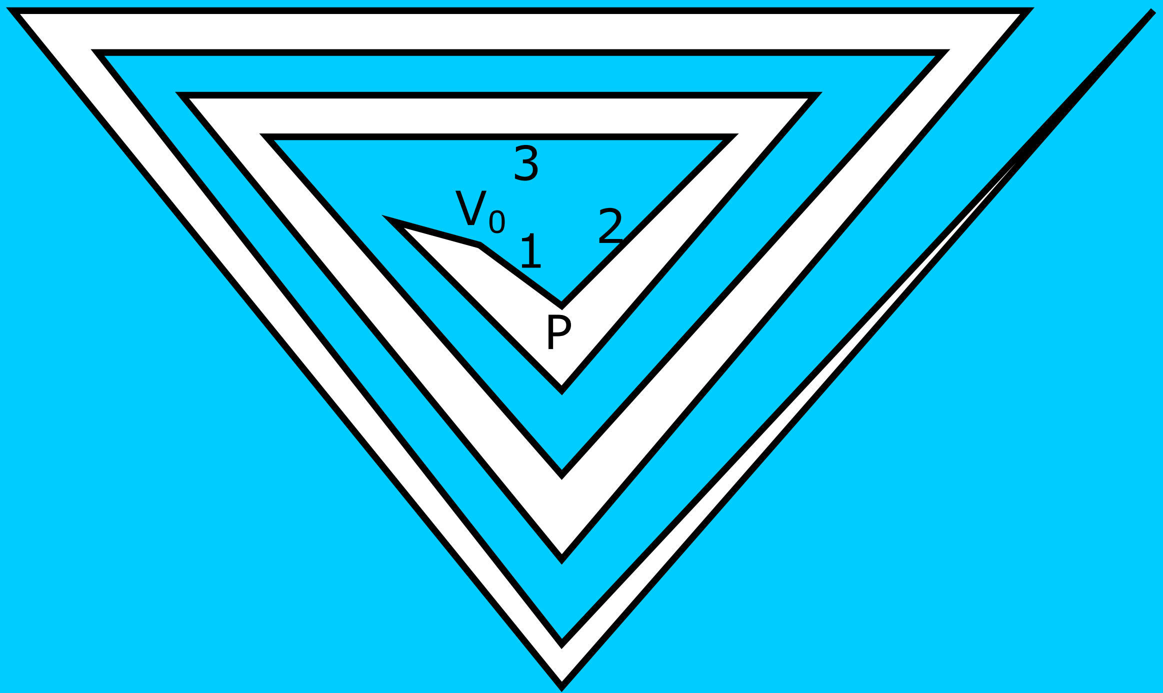


- L: je-li K ohraničené tak jako 1.1 jinak zůstává
- 1.1: procházíme K proti směru hodinových ručiček od L dokud nenajdeme vrchol w, takový, že následující vrchol leží vpravo od $v_{i+1}(v_{i+1}w)$, pak L_{i+1}=w jinak zůstává

Složitost

- paměťová složitost O(n)
- jádro polygonální oblasti může být zkonstruováno v optimálním čase O(n)
- nejhorší složitost nastane pro následující instanci - polygon bez jádra, kde má daný algoritmus časovou složitost $O(N^2)$

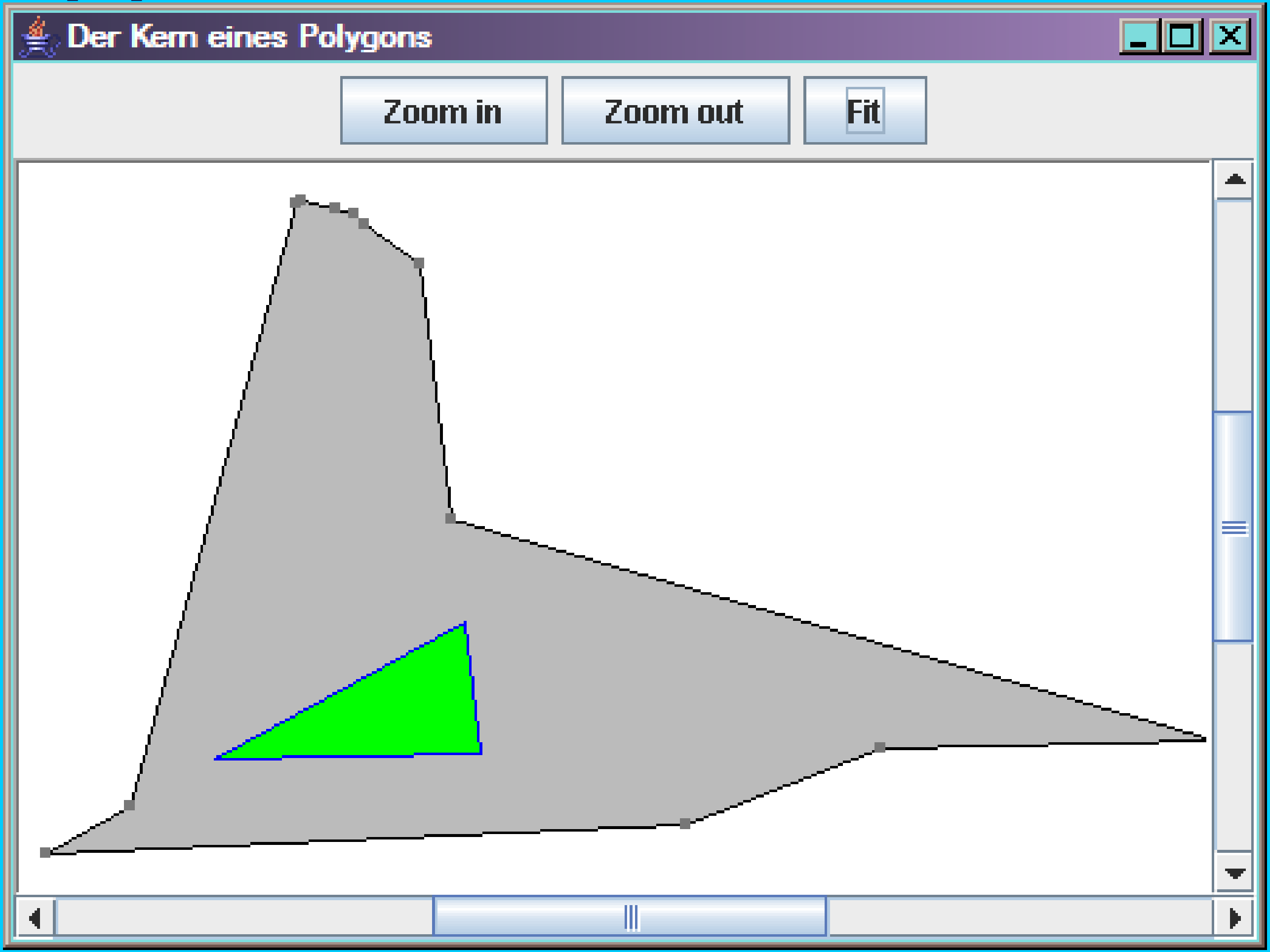
Nejhorší případ



Řešení

- ukončit algoritmus jakmile obteče v úhlu větším než 3π kterýkoliv bod částečného jádra K

Applet



- http://web.informatik.uni-bonn.de/I/GeomLab/VisPolygon/index.html.en

Použití

- zjištění, zda je polygon hvězdicovitý
- umístění všesměrového vysílače

Další algoritmy

- **Competitive strategy** (Závodní strategie)
- nalezení bodu jádra polygonální oblasti - maximálně 5.333x delší než nejkratší cesta, předpokládá rozhled 360° - využití v robotice

Zobecnění do více prostorů

- při přechodu do trojrozměrného prostoru je třeba přímky nahradit rovinami, a najít uspořádání všech trojúhelníků (tesselace) v objektu

Kde získat další informace

- Franco P. Preparata: Computational Geometry
- http://www.pi6.fernuni-hagen.de/publ/tr211.pdf

Dotazy

Děkuji za pozornost