Algoritmi per la triangolazione di poligoni nel piano

Simone Marini
Istituto per la Matematica Applicata e Tecnologie Informatiche
Consiglio Nazionale delle Ricerche
simone@ima.ge.cnr.it

27 Settembre 2002

Introduzione

Generalmente un poligono può essere molto complesso: può essere composto da alcune miliaia di vertici, può essere concavo o convesso e può contenere buchi. Per questi motivi vi è spesso la necessità di decomporre il poligono in componenti più semplici che possono essere facilmente e velocimente gestite. I poligoni piani possono essere decomposti in diversi modi, ad esempio, in triangoli, in parti convesse, in trapezzoidi oppure in poligoni star-shaped.
Il calcolo della triangolazione di un poligono piano è un algoritmo fondamentale nella geometria comptazionale. Ad esempio l'algoritmo di Kirkpatrick per la ricerca di un punto all'intenro un poligono presuppone che il poligono sia triangolato, anche alcuni algoritmi per il calcolo dell'itersezione tra poligoni presuppongono che i poligoni siano triangolati.

Ogni poligono semplice ha una triangolazione. Questo dipende dal fatto che ogni poligono ha almeno una diagonale se ha almeno un vertice convesso. Da questa osservazione si può concludere che:

- ogni poligono con n=4 vertici ha una diagonale;
- ogni poligono con n vetici può essere partizionato in triangoli aggiungiendo le possibili diagonali.

Esistono molti modi per triangolare un poligono di n vertici, tutti questi modi hanno però in comune il numero di diagonali: n - 3 ed il numero di triangoli n - 2.
Per alcune applicazioni è essenziale che il minimo angolo interno di ogni triangolo sia massimizzato, in questo modo viene evitata la produzione di triangoli allungati, questo tipo di triangolazione viene detta di Delaunay. Gli algoritmi basati sulla triangolazione di Delaunay producono triangolazioni di qualità elevata.

1 Algoritmi basati sull'inserimento di diagonali

1.1 Approccio Diretto

Un algoritmo basato sull'esistenza di almeno una diagonale all'interno di un poligono con almeno 4 vertici, si può riassumere nel seguente modo:

1. Trova una diagnale. Tale diagonale divide il pligono in due poligoni
2. Applica ricorsivamente l'algoritmo su entrambi i poligoni ottenuti al passo 1
3. L'algoritmo termina quando non è più possibile trovare una diagonale, cioè quando il poigono è stato completamente suddiviso in triangoli.

La ricerca della diagonale al punto 1 dell'algoritmo, consiste nel scegliere una coppia di vertici non consecutivi del poligono, quindi considerare tale coppia gli estremi della possibile diagonale s. Se esiste un lato del poligono, i cui estremi non incidono negli estremi di s, ma che interseca s, allora s non è una diagonale del poligono. Se non esiste un lato del poligono che interseca s, a meno dei suoi vertici estremi, allora s è una diagonale del poligono. Nella

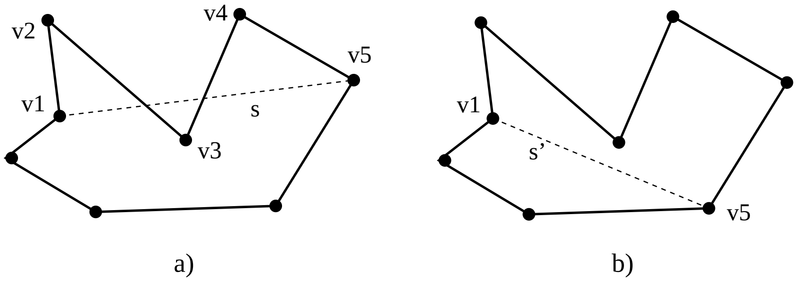

Figura 1: a) Il segmento s non è una diagonale del poligono. b) Il segmento s' è una diagonale del poligono.

figura 1, viene considerato dapprima il segmento s, figura 1a), tale segmento non è una diagonale del poligono in quanto viene intersecato dai lati $v2 - v3$ e $v3 - v4$ mentre in 1b) il segmento s' rappresenta una diagonale del poligono in quanto non interseca nessun lato se non quelli incidenti nei suoi estremi.
Questo algoritmo calcola tutte le possibili diagonali, e per ogni diagonale verifica la sua intersezione con tutti i lati del poligono e le diagonali già trovate, pertanto la sua complessità nel caso peggiore è $O(n^4)$.

1.2 Ear Searching Methods

Il vertice v_i di un poligono P, è detto principale se nessun altro vertice del poligono giace all'interno del triangolo v_{i-1}, v_i, v_{i+1}, oppure, in altre parole, se nessun altro vertice di P giace all'interno della diagonale v_{i-1}, v_{i+1}. Un vertice principale v_i di un poligono semplice P è detto $e$ar. Nella figura 2a), il vertice v_i é un $e$ar, mentre in figura 2b) la diagonale v_{i-1}, v_{i+1} non è contenuta nella nel poligono, quindi il vertice v_i non è un $e$ar. Si dimostra che un qualsiasi poligono semplice, che non sia un triangolo ha almeno 2 $e$ar che non si sovrapongono tra loro, dove due $e$ar v_i e v_j non si sovrappongono se $[v_{i-1}, v_i, v_{i+1}] \cap [v_{j-1}, v_j, v_{j+1}] = \varnothing$.
L'approccio diretto al calcolo degli $e$ar produce una triangolazione del poli-

gono iniziale, con un algoritmo che ha una complessità tempo di $O(n^3)$. Tale algoritmo è stato proposto da Meisters [1].

Nel 1990 Elgindi, Everett e Toussaint svilupparono un algoritmo per la ricerca degli *e*ar in tempo $O(n^2)$ [2]. Esso è basato sulla seguente osservazione: un good-subpolygon P_1 di un poligono semplice P, è un sottopoligono il cui boundary differisce da quello di P al più per un lato. Il good-subpolygon così ottenuto ha almeno un ear. L'idea base dell'algoritmo si riassume nei seguenti passi:

1. dividere il poligono iniziale P in due sottopoligoni, tali che, uno dei due sia un good-subpolygon con al più $\lfloor \frac{n}{2} \rfloor + 1$ vertici;
2. il passo precedente viene ricorsivamente applicato ai due poligoni ottenuti;
3. Il processo di ricorsione termina quando non è più possibile trovare un good-subpolygon, cioè quando il poligono è stato completamente triangolato.

Nel caso peggiore l'algoritmo ha complessitá $O(n^2)$ per un poligono con n

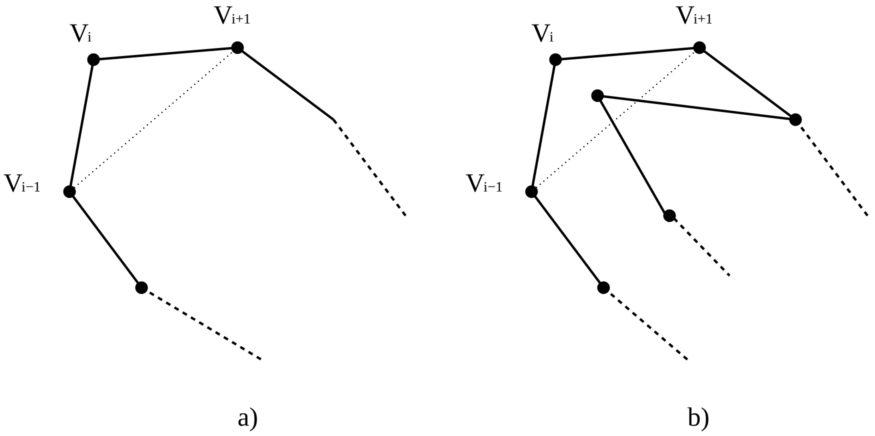

Figura 2: a) i vertici v_{i-1}, v_i, v_{i+1} formano un ear. b) la diagonale v_{i-1}, v_{i+1} non è completamente contenuta nel poligono.

vertici, in quanto la divisione del poligono iniziale, in due sotto poligoni ha complessità tempo $O(n)$ ed il secondo passo viene ripetuto al più n volte.

1.3 Divide et Impera

Garey, Jonson, Preparata and Tarjan proposero un algoritmo divide et impera la cui complessità tempo per la prima volta fu minore di $O(n^2)$ [3]. In particolare il loro algoritmo opera in tempo $O(n\ log\ n)$ ed è composto da due passi. Il primo passo decompone il poligono semplice in sottopoligoni monotoni, in tempo $O(n\ log\ n)$, il secondo passo triangola i sottopoligoni ottenuti in tempo lineare.

Un algoritmo con complessità tempo $O(n\ log^*\ n)$ più semplice di quello sviluppato da Garey, Jonson, Preparata and Tarjan, è l'algoritmo sviluppato da Seidel [4]. L'idea base dell'algoritmo di Seidel è definita in tre passi:

1. Il poligono viene suddiviso in trapezzoidi;
2. ai trapezzoidi viene aggiunta una diagonale;
3. le diagonali introdotte al passo precedente suddividono il poligono in sottopoligoni che possono essere triangolati in tempo lineare.

Il poligono viene suddiviso in trapezzoidi in tempo $O(n\ log\ n)$ (figura 3a)) e le diagonali vengono introdotte in tempo costante scegliendo due vertici del trapezzoide che non giaciono sullo stesso lato del trapezzoide figura 3b). Il passo 3 dell'algoritmo viene effettuato in tempo lineare poiché i sottopoligoni prodotti al passo 2 sono formati da due catene monotone di lati, di cui una delle due è formata da un solo lato, figura 3c).

Chazelle e Incerpi [5], svilupparono un algoritmo la cui complessità tempo dipende dalla forma del poligono. L'algoritmo costruisce una triangolazione del poligono in tempo $O(n\ log^*\ s)$ dove $s < n$ e n è il numero di vertici del poligono. La quantità s misura la "sinuosità" del poligono, cioè quante volte

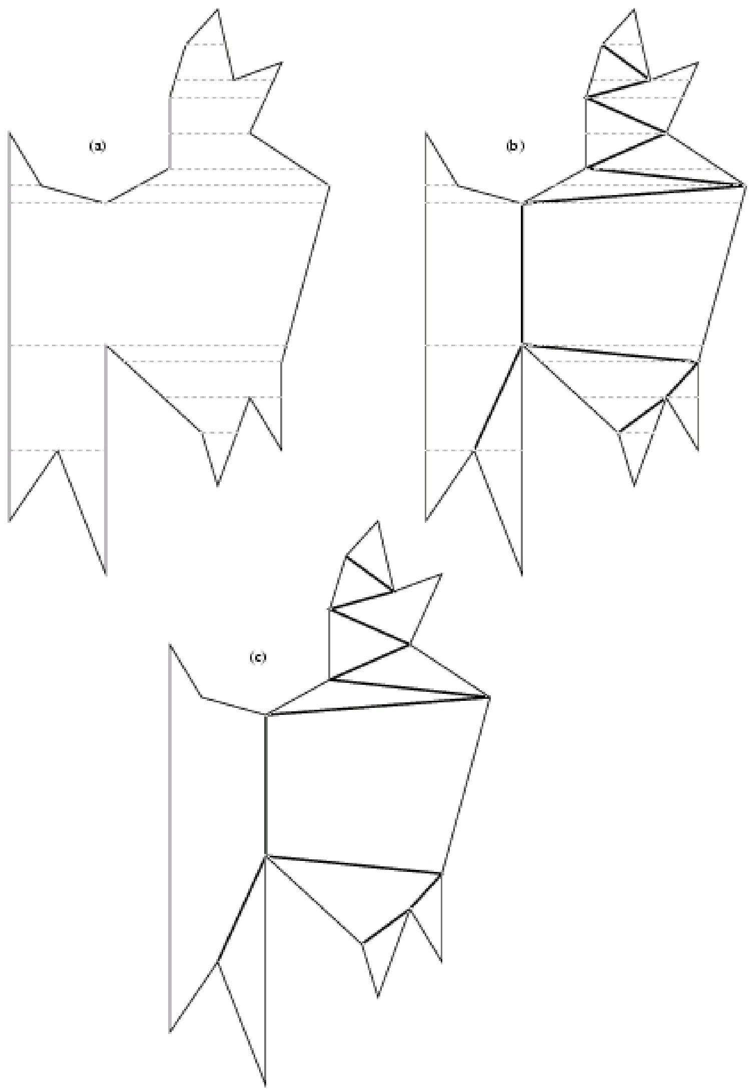

Figura 3: a) Poligono decomposto in sottopoligoni trapezzoidali. b)Introduzione delle diagonali. c) Decomposizione del poligono in pezzi facili da triangolare.

il boundary del poligono cambia direzione formando “tratti di spirali” orientati diversamente. In generale la quantità s è molto piccola anche in poligoni molto complessi e non dipende dal numero di vertici del poligono. L'idea base per l'analisi della “sinuosità” del poligono, si basa sui cambiamenti di orientamento della linea $L[v_i, v_{i+1}]$, che passa attraverso i vertici v_i e v_{i+1} dove $0 < i < n$. Ogni volta che L passa per la posizione verticale girando in senso orario (antiorario) un contatore viene decrementato. L descrive una spirale (anti-spirale) se il contatore non viene mai decrementato (incrementato) due volte in una sucessione. Un anuova catena di vertici viene creata quando la precedente cessa di evolversi come spirale o anti spirale.

Chazelle nel 1991 propose un algoritmo in grado di triangolare un poligono semplice in tempo $O(n)$ nel caso peggiore [6]. La triangolazione viene prodotta utilizzand una struttura chiamata visibility-map che a sua volta è la generalizzazione di una decomposizione in trapezzoidi del poligono.
L'algoritmo proposto imita l'algoritmo merge sort. Il poligono di n vertici viene diviso in catene di $\frac{n}{2}$ vertici, e queste, in catene di $\frac{n}{4}$ vertici e così via. La visibility-map di una catena è ottenuta facendo il merge delle visibility-map delle reltive sotto catene. In questo modo, l'algoritmo ha complessità $O(n\ log\ n)$. Chazelle migliora l'algoritmo dividendolo in due fasi. Nella prima fase viene calcolata una versione grossolana delle visibility-map. Queste visibility-map sono abbastanza grossolane da poterne fare il merge in tempo lineare. Nella seconda fase le visibility-map, vengono raffinate in tempo lineare. La triangolazione è infine prodotta dalla decomposizione in trapezzoidi ottenuta dalle visibility-map.

2 Algoritmi basati sulla triangolazione di Delaunay

La triangolazione di un poligono convesso puó essere ottenuta come triangolazione di Delaunay, considerando i vertici del poligono, come un insieme di

punti del piano (figura 4a)). Se il poligono non è convesso i suoi lati vanno considerati come vincoli per la triangolazione.
Sia V un insieme dei punti del piano e L un insieme di segmenti, la cui intesezione è vuota, ed aventi come estremi vertici di V. La coppia $G = (V, L)$ definisce il grafo dei vincoli.
Due vertici P_i e P_j sono detti mutuamente visibili se il segmento P_i, P_j non interseca nessun vincolo oppure se appartiene ad L. Il grafo di visibilità di G è la coppia $G_v = (V_v, E_v)$ dove $V_v = V$ e $E_v = \{(P_iP_j|P_i,\ P_j \in V_v$ e $P_i,\ P_j)$ sono mutuamente visibili rispetto a $L\}$, figura 4b). Un arco appartenente ad

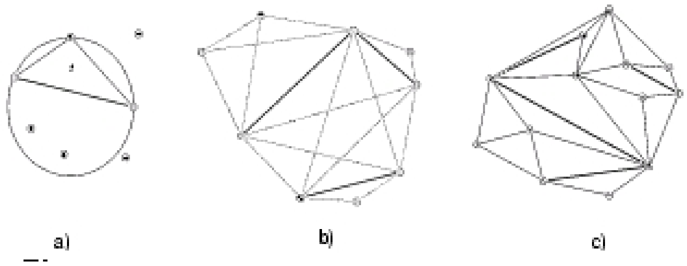

Figura 4: a) Proprietà del cerchio vuoto. b) Mappa di visibilità. c) Triangolazione di Delaunay vincolata.

E_v unisce due punti mutuamente visibili rispetto ad L. Una triangolazione vincolata di V rispetto ad L è il grafo $T = (V; L) = (V_t, E_t))$ dove $V_t = V$ e E_t è il sotto insieme massimale di $E_v \cup L$ tale che $E_l \subseteq E_t$ e l'intersezione tra gli archi di E_t è vuota tranne che nei loro estremi.
Una triangolazione di Delaunay vincolata (CDT) $T = (V; L)$ rispetto ad un insieme di vertici V e ad un insieme di segmenti L è una triangolazione di V, in cui per ogni triangolo t appartenente a T, la circonfrenza che inscrive t non contiene nessun altro vertice appartenente a T che sia visibile dai tre vertici di t (figura 4c)). In altre parole un triangolo t, appartenente ad una triangolazione vincolata T, è di Delaunay se non esistono altri vertici di T interni alla circonferenza che inscrive t, che siano visibili dai tre vertici di t.
La triangolazione di un poligono semplice può essere generalmente calcolata

come segue:

1. Calcolare la triangolazione vincolata di Delaunay poligono semplice rispetto all'insieme di vertici del poligono e con vincoli gli archi del poligono;
2. Rimuovere i triangoli che sono all'esterno del poligono.

Gli algoritmi per il calcolo della triangolazione di Delaunay vincolata, possono essere divisi in due gruppi: algoritmi per il calcolo della CDT che hanno come grafo dei vincoli un poligono semplice, oppure algoritmi per il calcolo della CDT con grafo dei vincoli generale.

2.1 Algoritmi per il calcolo della triangolazione vincolata di Delaunay per poligoni semplici

Un algoritmo con complessità tempo $O(n^2)$ il cui approccio è basato sulla visibilità dei vertici, è descritto in De Floriani, Falcidieno, Pienovi [7]. L'algoritmo calcola il grafo di visibilità dei vertici del poligono semplice Q in tempo $O(n^2)$ ed il diagramma di Voronoi dei vertici di Q in tempo $O(n\ log\ n)$. La triangolazione di Delaunay del poligono viene calcolata unendo ogni vertice P di Q ai vertici che sono visibili da P e che sono Voronoi-neighbours di P.

Un algoritmo che triangola il poligono in tempo $O(n\ log\ n)$ è stato proposto da Lee e Lin. L'algoritmo è basato sull'osservazione che due vertici t_1 e t_2, appartenenti al poligono P tali che il segmento t_1t_2 sia completamente interno a P, possono essere trovati in in tempo lineare. Il segmento t_1t_2 suddivide P in due sotto poligoni P_l e P_r. Tali sottopoligoni vengono ricorsivamente suddivisi e per ogni sottopoligono viene calcolata la triangolazione vincolata di Delaunay T_l e T_r. La triangolazione del poligono iniziale T viene ottenuta dal merging di T_l e T_r.

2.2 Algoritmi per il calcolo della triangolazione vincolata di Delaunay per grafi dei vincoli generali

Un algoritmo che necessita di una fase di preprocessing è quello proposto da Boissonnat [8]. La fase di preprocessing ha lo scopo di trasformare il porblema della triangolazione di poligono semplice e piano, in un problema per la triangolazione di Delunay per un insieme di punti generici.
L'idea base è quella di modificare l'insieme dei vertici del poliogno aggiungendo dei punti che stanno sui segmenti che definiscono i vincoli, in questo modo la triangolazione ottenuta contiene necessariamente i vincoli. Il vincolo e è un arco di Delunay se la circonferenza avente come diametro e non interseca nessun altro vincolo. Se la circonferenza associata ad e interseca qualche altro vincolo allora e viene splittatto in modo che le circonferenze associate ai sotto segmenti ottenuti non intersechino altri vincoli. Se due vincoli si intersecano nei loro punti estremi, viene aggiunto un nuovo punto in entrambe i segmenti. La circonferenza che inscrive il triangolo formato dai due nuovi punti e dal punto che i due segmenti hanno in comune, non interseca nessun altro vincolo.
Questo algoritmo ha complessità tempo $O(n\ log\ n)$ e genera al più $O(n)$ nuovi punti.

Il metodo proposto da Boissonnat assume che i punti ed i vincoli siano predefiniti all'inizio del processo di triangolazione. Il metodo proposto da De Floriani e Puppo [9], risolve il problema CDT aggiornando in modo incrementale la triangolazione, aggiungendo nuovi punti e nuovi vincoli.
Viene calcolata una triangolazione iniziale, ad esempio un triangolo oppure un rettangolo (diviso in due triangoli) contenente l'intero dominio ed in modo incrementale vengono aggiunti i nuovi punti ed i vincoli. Quando l'ultimo punto o vincolo è stato aggiunto il triangolo (rettangolo) iniziale viene eliminato.
L'algoritmo ha complessità $O(ln^2)$ dove l è il numero di vincoli ed n il numero di vertici della triangolazione.

2.3 Proprietá di alcuni degli algoritmi citati

In questa sezione gli algoritmi proposti da Meister, descritto nella sezione 1.2, Seidel, descritto nella sezione 1.2 e l'algoritmo di De Floriani Puppo, descritto nella sezione 2.2 vengono confrontati al fine di stabilire la qualità della triangolazione prodotta e della possibilità di utilizzo con poligoni non semplici.

In figura 5 viene mostrato un esempio di come un poligono semplice viene

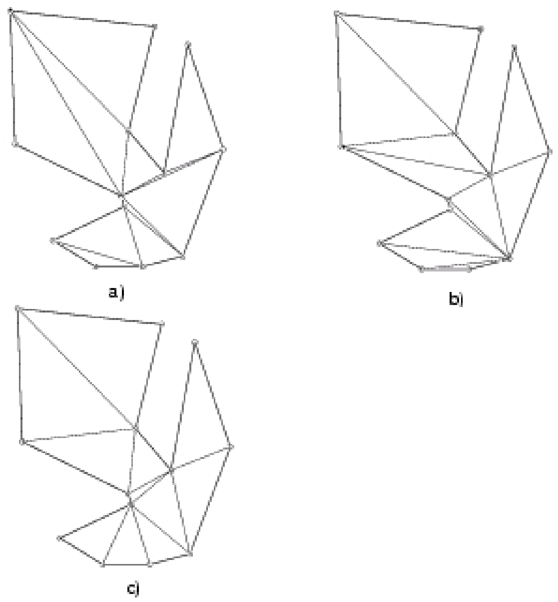

Figura 5: a) Ear cutting. b) Algoritmo di Seidel. c) Triangolazione vincolata di Delaunay.

triangolato utilizzando i tre algoritmi.

In figura 5a) possiamo vedere che Ear cutting calcola la triangolazione di minore qualità, cioè quella con triangoli più allungati. Una triangolazione di qualità migliore viene calcolata dall'algoritmo di Seidel 5b). Questi risultati sono dovuti al fatto che in entrambe i casi, gli algoritmi non hanno meccanismi per il controllo della qualità dei triangoli generati. Il risultato con qualità migliore viene ottenuto dall'algoritmo per il calcolo della triangolazione vincolata di Delaunay, figura 5c), anche se alcuni triangoli appaiono ancora allungati.

A parte Ear cutting, gli algoritmi trattati nella sezione 1 possosno essere estesi al trattamento di poligoni non semplici. Gli algoritmi che calcolano la triangolazione di Delaunay vincolata sono particolarmente adatti al trattamento di poligoni non semplici, in quanto il boundary dei buchi può essere trattato come un insieme di vincoli.

Riferimenti bibliografici

[1] Meisters G.H., *P*oligons have hears, American Mathematical Monthly. Vol 82, June/July, 1975.

[2] ElGindy H., Everett H., Toussaint G. T. *S*licing an Ear in Linear Time, Pattern Recognition Letters. Vol 14, 1993.

[3] Garey M.R., Jhonson D.S., Preparata F.P., Tarjan R.E., *T*riangulating a simple polygon, Inform. Process., Lett. 7, 1978.

[4] Seidel R. *A* simple and fast incremental and randomized algorithm for computing trapezoidal decompositions and for triangulating polygons, Computational geometry: Theory and applications, Vol 1, No. 1, 1991.

[5] Chazelle B., Incerpi J., *T*riangulation and shape complexity, ACM Transaction on Graphics, Vol 3, 1984.

[6] Chazelle B., *T*riangulating a simple poligon in linear time, Disc. Comp. Geom. 6, 1991.

[7] De Floriani L., Falcidieno B., Pienovi C., *D*elaunay-based representation of surfaces defined over arbitrary-shaped domains, Comput. Vision Graphic Image Process. 32, 1985.

[8] Boissonnat J.D. *S*hape reconstruction from planar cross section, Comput. Vision Graphic Image Process. Vol 44, 1988.

[9] De Floriani L, Puppo E. *A*n On-Line Algorithm for Costrained Delaunay Triangulation, Graphical Models and Image Processing. Vol 54, 3, 1992.