

Aproximació facetada de superfícies paramètriques retallades

Marc Vigo Anglada

Tesi doctoral

Directors: Pere Brunet Crosa
Núria Pla Garcia

Secció d'Informàtica Gràfica
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

Agost de 1998

Avaluació de la reparametrització

Els algorismes de triangulació que hem desenvolupat treballen gairebé íntegrament en espai paramètric i per tant depenen de la forma com estiguin parametritzats els pedaços del sòlid original. D'aquest problema ja ha ens n'hem ocupat i de fet hem desenvolupat un preprocessament, la reparametrització, per intentar solucionar-lo i un postprocessament de millora de la forma en espai imatge que ens confirmarà que, com a mínim en alguns casos, les triangulacions obtingudes podrien ser millors tenint en compte la forma dels triangles. Comprovar que aquest inconvenient existeix en realitat és ben senzill: tant sols cal aplicar els algorismes de triangulació amb i sense la reparametrització sobre algun objecte. Per exemple, la figura 6.26 mostra un exemple molt il·lustratiu. Aquest mateix objecte ens serveix per comprovar que pot ser molt convenient aplicar la reparametrització als pedaços que componen els objectes, si mes no en alguns casos concrets, tal com havíem previst a la secció 3.5.

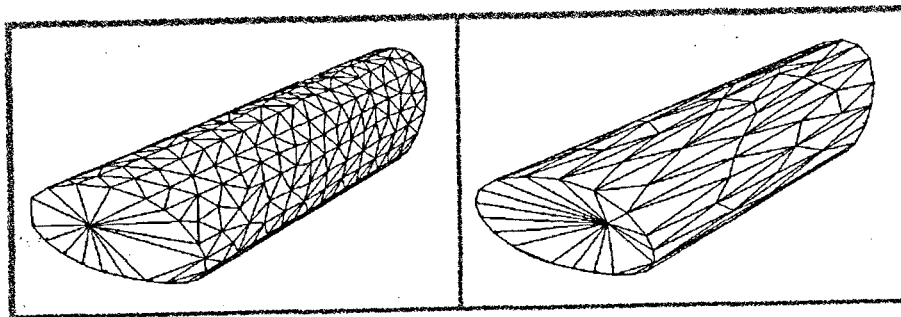
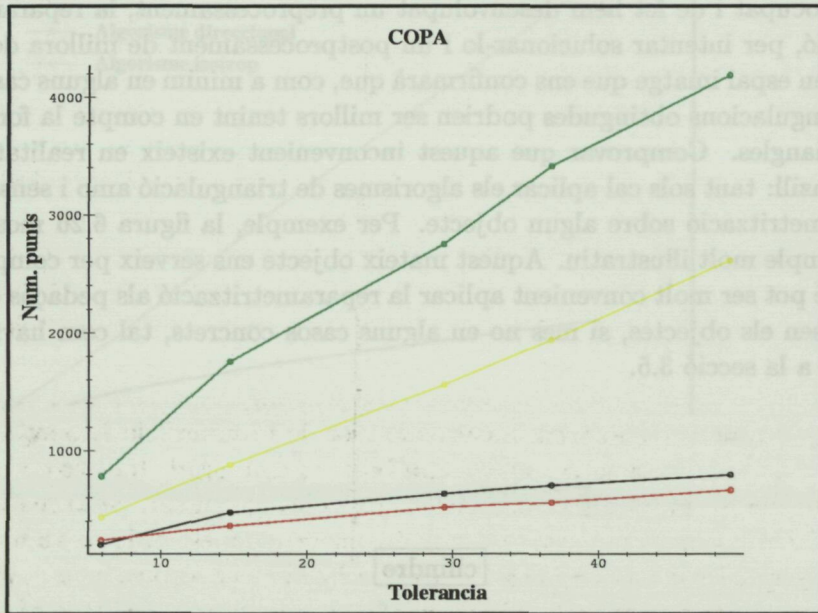


Figura 6.26: Exemple que mostra clarament fins a quin punt pot arribar a dependre la triangulació obtinguda de la parametrització dels pedaços. A l'esquerra, el resultat de l'algorisme isòtrop aplicat sobre la superfície 3; a la dreta, el mateix objecte triangulat fent servir també l'algorisme isòtrop però aplicant prèviament la reparametrització dels pedaços que el componen.

Efectivament, hem pogut veure que la reparametrització pot arribar a ser molt útil si es volen obtenir triangles de forma bona en aquells casos en què les distàncies en espai paramètric i en espai imatge no es corresponen, tal com havíem dit. Això no obstant, com que es tracta d'una reparametrització lineal hi ha pedaços que no pot arregar gaire, sobretot aquells en què hi ha fortes variacions de la superfície en diferents direccions.

Queda encara una altra qüestió per respondre: la reparametrització afecta de diferent forma a l'algorisme isòtrop que la direccional? Per respondre aquesta pregunta, hem realitzat una sèrie de tests, consistents en executar els dos algorismes de triangulació amb i sense la reparametrització en el quart objecte del joc de proves, la copa, la qual sabem que es veu bastant afectada per la forma com estiguin parametritzats els seus pedaços.

Hem executat doncs ambdós algorismes, amb diverses toleràncies per tenir un conjunt de dades més ampli. La gràfica 6.27 mostra els resultats d'aquestes proves.



Algorisme isotrop

- reparametritzar
- sense reparametritzar

Algorisme direccional

- reparametritzar
- sense reparametritzar

Figura 6.27: Comparació entre el nombre de punts resultants de l'algorisme isòtrop i el direccional aplicant i sense aplicar el preprocessament de la reparametrització.

Podem observar que el nombre de punts resultant aplicant la reparametrització o sense aplicar-la varia molt més en el cas isòtrop que en el direccional, és a dir, que la reparametrització afecta molt menys a l'algorisme direccional que a l'isòtrop. Aquest efecte resulta fàcil de raonar: les fites de l'algorisme direccional inclouen més informació de la forma de les superfícies originals en espai imatge que l'isòtrop, ja que diferencia les diverses direccions de la curvatura. En altres paraules, l'algorisme isòtrop té molt més en compte la forma dels triangles que el direccional, però no oblidem que les mesures de la forma es realitzen en espai paramètric. Per tant, una parametrització molt diferent tindrà com a efecte que els triangles resultants no s'assemblin gaire quan s'aplica la reparametrització que quan no s'aplica. La conclusió és que si bé en el cas isòtrop sempre val la pena utilitzar el pre-processament de reparametrització, en el cas direccional pot ser que sigui

un pas no gaire útils en alguns casos.

Avaluació de la forma dels triangles

En aquest apartat valorarem els dos postprocessaments que s'ocupen de millorar la forma dels triangles, la relaxació i el mètode de millora de la forma. Recordem que la relaxació és un pas addicional que podem aplicar per tal de millorar la forma dels triangles i que consisteix en un algorisme iteratiu que mou els punts de la triangulació en espai paramètric segons unes forces calculades en funció de l'admissibilitat de les entitats (arestes i triangles). El mètode de millora de la forma en espai imatge, en canvi, mesura la geometria dels triangles en \mathbb{R}^3 i ha estat específicament desenvolupat amb la finalitat d'avaluar si es poden aconseguir triangulacions de millor qualitat, en termes de la forma dels triangles, que les obtingudes amb els nostres mètodes de triangulació.

Presentarem un exemple d'aplicació d'aquest algorisme sobre un objecte del joc de proves. Hem triat la cara superior de la superfície 1, perquè sabem que la seva parametrització s'adiu bastant amb les distàncies en espai imatge i que per tant les triangulacions que obtinguem són de forma prou bona. Hem obtingut una sèrie de triangulacions d'aquest objecte amb diferents toleràncies utilitzant l'algorisme direccional i en cada cas hem mesurat la forma dels triangles en espai paramètric i en espai imatge. Una manera d'avaluar la qualitat de la triangulació en termes de la forma consisteix en mesurar els angles mínims dels triangles; com més s'apropin a 60 deg, més propers a ser equilàters seran els triangles. Les taules 6.3 i 6.4 mostren els resultats de les execucions en aplicar el mètode direccional sense aplicar cap postprocessament (columnes segona i tercera), aplicant la relaxació (columnes quarta i cinquena) i aplicant a més a més el mètode de millora en espai imatge (columnes sisena i setena). La taula 6.4 mostra també els resultats del mètode de la forma dels triangles aplicat sobre la triangulació a la qual no s'ha realitzat la relaxació (dues darreres columnes). En cada cas, hem mesurat tant l'angle mínim de tota la triangulació (a_{min}) com la mitjana dels angles mínims de tots els triangles ($\overline{a_{min}}$).

A l'hora de mesurar els angles mínims dels triangles, cal tenir en compte que n'hi ha que resultaran inalterats tant per la relaxació com per la millora de la forma en \mathbb{R}^3 , els formats per dues arestes del contorn d'una cara, ja que tant sols els vèrtexs interiors són els que es mouen. Per tant, les taules ja no tenen en compte aquests angles.

Els resultats d'aquestes execucions són ben ilustratius. En primer lloc, vegem que el procés de relaxació en tots els casos millora considerablement la mitjana dels angles mínims en espai paramètric, si bé en algunes ocasions pot empitjorar l'angle mínim absolut. És a dir, que aquest procés resulta útil per millorar globalment la forma dels triangles, però que en alguns casos pot afectar negativament algun angle en concret. Això resulta ben

ϵ	Direccional		+ Relaxació		+ Millora \mathbb{R}^3	
	a_{min}	$\overline{a_{min}}$	a_{min}	$\overline{a_{min}}$	a_{min}	$\overline{a_{min}}$
.20	16.13	39.72	10.98	42.49	13.23	42.63
.15	8.83	38.53	16.40	42.38	15.16	42.58
.10	8.83	38.87	12.10	40.86	12.10	40.73
.08	16.23	39.76	10.76	43.28	10.76	42.57
.06	10.32	40.43	4.49	45.47	5.01	45.14
.04	6.07	38.17	7.23	43.00	7.23	42.55
.005	7.93	37.01	4.01	41.45	4.01	41.28

Taula 6.3: Angles mínims en el pla paramètric (\mathbb{R}^2) dels triangles resultants en aplicar l'algorisme direccional amb i sense els passos de la relaxació i de millora de la forma en espai imatge.

ϵ	Direccional		+ Relaxació		+ Millora \mathbb{R}^3		+ Millora \mathbb{R}^3 - Relaxació	
	a_{min}	$\overline{a_{min}}$	a_{min}	$\overline{a_{min}}$	a_{min}	$\overline{a_{min}}$	a_{min}	$\overline{a_{min}}$
.20	12.28	40.34	8.24	51.99	8.72	52.08	12.28	40.42
.15	7.05	40.88	7.66	48.26	7.87	48.29	7.05	41.66
.10	7.05	40.52	9.13	44.11	9.13	44.20	7.05	40.71
.08	11.63	38.82	7.89	48.57	7.89	48.58	11.63	39.52
.06	7.98	45.29	5.06	50.06	5.17	50.09	7.98	45.29
.04	4.54	41.16	4.88	42.34	4.88	42.41	4.54	42.84
.005	6.36	37.40	3.02	38.85	3.02	39.00	6.36	39.16

Taula 6.4: Angles mínims en espai imatge (\mathbb{R}^3) dels triangles resultants en aplicar l'algorisme direccional amb i sense els passos de la relaxació i de millora de la forma en espai imatge.

entenedor si es té en compte que es tracta d'un procés que mou cada punt de la triangulació a partir d'una força ponderada, no pas tenint en compte la pitjor situació d'admissibilitat. A més a més, com que tal com ja hem dit la parametrització de la superfície 1 es correspon bastant amb l'espai imatge, la millora dels angles provocada per la relaxació també és notòria en \mathbb{R}^3 , cosa que podria no ocórrer en un pedaç de parametrització no adequada.

En segon lloc, la taula 6.3 mostra que, com és lògic, no té sentit mesurar en \mathbb{R}^2 el procés de millora de la forma en \mathbb{R}^3 ; comparant els angles en espai paramètric just abans d'aplicar aquest procés (és a dir, els resultants de la relaxació) amb els angles finals, observem que tant poden millorar-se com empitjorar. Cal fixar-se doncs en les dades de la taula 6.4 i comparar els angles abans i després d'aplicar aquest postprocessament, o sigui, les columnes quarta i cinquena amb la sisena i setena, i les columnes segona

i tercera amb les dues últimes. Observem que aquest mètode és capaç de millorar la qualitat dels triangles en \mathbb{R}^3 . En moltes ocasions no millora l'angle mínim absolut, però en tot cas no l'empitjora a conseqüència de com ha estat dissenyat l'algorisme. En canvi, en totes les proves tabulades menys una, la mitjana dels angles mínims creix, encara que sigui lleugerament i de forma no tan notòria com la relaxació; no oblidem però que es tracta d'una mitjana. De fet, l'objectiu d'aquest mètode ja era aquest, comprovar que les triangulacions es poden millorar fins i tot amb un algorisme molt simple. Si la millora no és excessiva és degut tant sols a què es tracta d'un el mètode de força bruta que no altera la configuració topològica de la triangulació de partida i que conté moltes restriccions (referents a l'admissibilitat, sobretot).

Finalment, esmentem que el fet d'utilitzar un algorisme de triangulació de Delaunay i l'algorisme de refinament d'arestes tenen com a conseqüència que els triangles, ja d'entrada, tendeixin a tenir formes no excessivament dolentes. Com és ben sabut, la triangulació de Delaunay té la particularitat que maximitza els angles mínims de la triangulació [Bow81, GS85]. D'altra banda, l'algorisme de refinament d'arestes tendeix a millorar els angles per la forma com insereix els punts auxiliars, ja que en inserir un punt en una triangulació de Delaunay justament en el circumcentre d'un dels triangles, el triangle desapareixerà i se'n crearà un de nou amb un angle doble que el de l'anterior [SH92, Rup92]. En conseqüència, els algorismes que hem desenvolupat, tot i no tenir com a primer objectiu optimitzar la forma dels triangles, sí que tenen en compte aquest criteri, i llevat de casos patològics obtindran triangulacions bastant acceptables des d'un punt de vista de la forma. Aquesta afirmació és molt més vàlida per a l'algorisme isòtrop que no pas pel direccional, el qual utilitza versions modificades dels mètode d'obtenció d'una CDT i de refinament d'arestes.

Altres exemples

En aquesta secció hem inclòs uns quants exemples més que ratifiquen les afirmacions que hem fet a les seccions anteriors. També poden servir per acabar de veure el comportament dels algorismes que hem desenvolupat i comprovar que són vàlids no tant sols per triangular sòlids compostos per un nombre reduït de cares sinó també per aproximar fidelment sòlids més complexos limitats per un conjunt elevat de pedaços retallats.

La figura 6.28 ens serveix per acabar de comparar l'algorisme direccional amb l'isòtrop. Com ja hem dit, l'objectiu de la forma dels triangles és menys important en el cas de l'algorisme direccional. De fet, si bé l'algorisme direccional obté molts menys triangles també és cert que alguns d'ells són de forma pitjor (contenen angles petits). Això no obstant, podem afirmar que, fixada una tolerància, els triangles que construeixi l'algorisme direccional aproximaran millor la superfície original que els que aconseguixi l'isòtrop, ja que en els dos casos seran triangles admissibles i en el cas direccional redu-

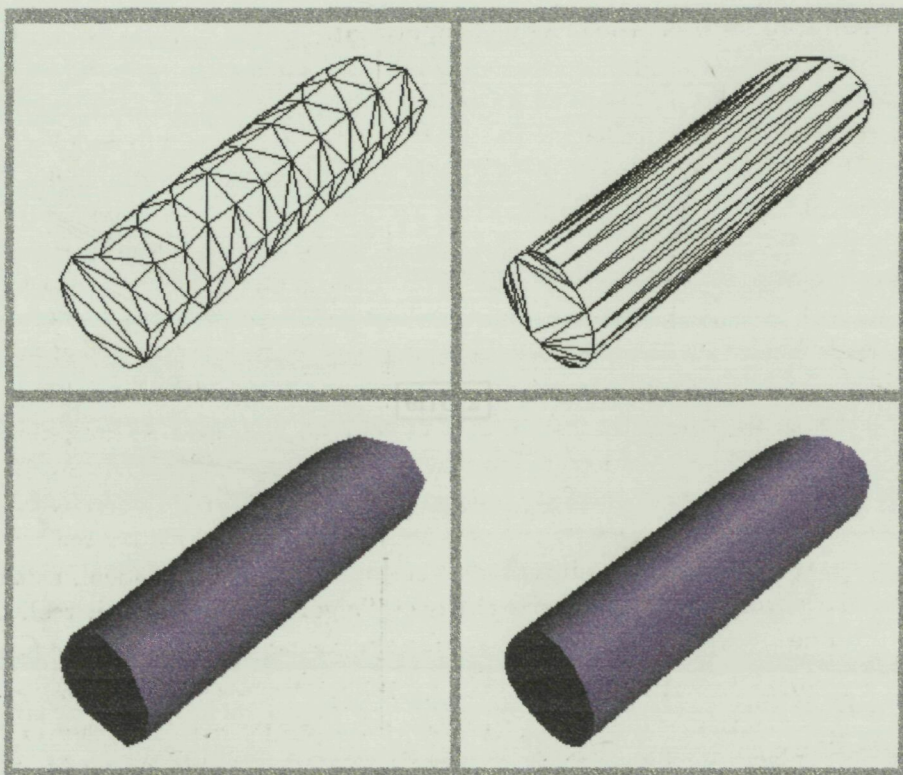


Figura 6.28: Exemple que mostra la diferència entre minimitzar el nombre de triangles i optimitzar l'error de segona espècie. A l'esquerra, triangulació de la superfície 3 obtinguda mitjançant l'algorisme isòtrop (reparametritzant); a la dreta, la mateixa superfície triangulada amb l'algorisme direccional.

eixen l'error de segona espècie (la diferència entre les normals dels triangles i les de la superfície, vegeu la secció 2.2.3). De fet, de vegades l'algorisme de relaxació direccional aconsegueix una millora la forma dels triangles però empitjora l'error de segona espècie. A la figura 6.28 es pot veure que malgrat que la triangulació resultant de l'algorisme isòtrop té més triangles que la que s'obté amb el mètode direccional, la representació amb suavitzat és millor en el segon cas que en el primer.

Les figures 6.29 i 6.30 mostren d'altres exemples d'aplicació dels algorismes de triangulació que hem desenvolupat en aquesta tesi.

6.4 Conclusions

A les seccions anterior hem presentat una sèrie d'exemples que permeten avaluar els algorismes isòtrop i direccional, així com els diversos subprocessos que hem desenvolupat específicament per a aquestes aplicacions. Aquesta

secció recull les conclusions a què hem arribat.

- Conversió dels pedaços degenerats (secció 3.4): es tracta d'un preprocessament en principi independent dels algorismes de triangulació que a la pràctica sempre resulta indispensable, per evitar situacions patològiques. Mentre que el primer algorisme aproxima el pedaça degenerat amb continuïtat d'ordre 0 (C^0), el segon obté continuïtat de primer ordre (G^1), tot i que no exacta, sinó tant sols aproximada. No obstant, hem mostrat com es pot focalitzar l'error de continuïtat aprop del vèrtex degenerat elevant de grau el pedaça o bé subdivint-lo.
- Reparametrització dels pedaços (secció 3.5): és un preprocessament senzill l'únic inconvenient del qual pot ser l'elevació de grau (no massa perillosa) dels pedaços originals. Hem observat que en diversos casos freqüents resulta un procés molt útil per millorar la forma dels triangles. Ara bé, si els objectius de la triangulació no inclouen la forma dels triangles, com pot ser el cas de l'algorisme direccional, realitzar aquest preprocessament pot resultar un pas no massa útil.
- Refinament d'arestes (seccions 4.5.1 i 5.4.3): aquest pas és el nucli dels algorismes que hem desenvolupat, atès que és el que obté la triangulació de l'interior de les cares. Està basat en l'ús de les fites de la curvatura i en l'algorisme incremental que obté una CDT, per tant n'existeixen dues versions, una per al cas isòtrop i l'altre per al direccional. L'inconvenient que es pot objectar és que treballa en espai paramètric i per tant depèn fortament de la parametrització dels pedaços. Aquesta qüestió resulta molt menys important en el cas de l'algorisme direccional, gràcies al fet que les fites direccionals inclouen més informació geomètrica sobre la forma dels pedaços en espai imatge.
- Relaxació (seccions 4.5.4 i 5.4.4): malgrat que aquest mètode ha estat dissenyat per millorar la forma dels triangles, pot valer la pena no només fer-lo servir en el cas isòtrop sinó també en el direccional, ja que es tracta d'un postprocessament independent que no incrementa el nombre de triangles ni vèrtexs resultants.
- Algorisme d'obtenció d'una CDT (apèndix A): es tracta d'un algorisme útil per a triangular un graf qualsevol i que té l'avantatge que treballa incrementalment, és a dir, que permet afegir vèrtexs i arestes a una CDT donada. Com en el cas dels dos algorismes anterior, n'existeixen dues versions, una per al cas direccional i l'altra per a l'isòtrop; ambdues poden fer-se servir en qualsevol altre àmbit de treball que faci ús del concepte de triangulació de Delaunay, ja es tracti de la CDT clàssica o d'una que tingui en compte distàncies diferents de l'habitual.

- Comparació dels dos algorismes de triangulació proposats (secció 6.3.3): tal com ens esperàvem, l'algorisme isòtrop obté triangulacions amb més triangles i vèrtexs que el direccional. Mentre que els triangles creats per l'algorisme isòtrop són millors des del punt de vista de la forma, els produïts per l'algorisme direccional s'adapten millor a la superfície en termes de l'error de segona espècie. D'altra banda, com ja hem dit l'algorisme isòtrop depèn més de la parametrització dels pedaços que el direccional.
- Algorisme de millora de la forma (secció 6.2): tot i que no ha estat pensat per ser un algorisme eficient, ens ha permès comprovar que es pot optimitzar la qualitat de les triangulacions produïdes en espai imatge, fins i tot sense haver de modificar la topologia de la malla.

6.5 Treball futur

Durant el procés de desenvolupament d'un treball tan voluminós i llarg de dur a terme com una tesi, és òbvi que apareixen una sèrie de qüestions que hom creu que poden millorar-se o d'altres que seria interessant investigar, però que requeririen un estudi a part. En aquesta secció incloem les que se'ns han anat acudint.

- La implementació actual dels algorismes de triangulació tant sols permeten treballar amb pedaços de Bézier retallats. Una versió futura podria admetre altres tipus de representacions, d'altra banda cada cop més freqüents en el món del disseny assistit per ordinador, com puguin ésser les splines o les NURBS. Remarquem que aquesta adaptació dels nostres algorismes tant sols requeriria poder llegir i emmagatzemar d'altres formats i saber calcular les fites sobre les segones derivades de les superfícies, ja que les fites de la curvatura es poden obtenir independentment del tipus de representació que s'utilitzi.
- Una altra millora que proposem dels algorismes consistiria en adaptar els programes de manera que admetessin grups de superfícies que no descrivissin obligatòriament un sòlid tancat. Aquesta ampliació no seria gaire costosa, tant sols caldria suprimir les comprovacions topològiques que garanteixen que el sòlid d'entrada descriu un volum tancat i discretitzar les corbes de retallat "obertes" (aquelles corbes que no limitessin amb cap altra veïna) tenint en compte la curvatura de la pròpia corba, no la dels pedaços que hi convergeixen (vegeu la secció 4.4). Això es podria fer, per exemple, tenint en compte els resultats del teorema 2, el qual proporciona fites de la màxima longitud que pot tenir un segment per aproximar una corba en funció de la curvatura, sempblantment a com la solució adoptada a [SH92].

- El nostre esquema algorísmic en primer lloc poligonalitza les arestes del sòlid original i posteriorment triangula l'interior de les cares. D'aquesta forma es garanteix la conformitat de la triangulació resultant i les cares es poden triangular per separat, independentment unes de les altres. No obstant, aquest enfoc té alguns inconvenients:
 - Les cares de l'objecte planes (o gairebé planes) limitades per arestes curvades queden aproximades per un conjunt de triangles de forma qualsevol, sovint per triangles afuats. Aquest problema potser caldria evitar-lo, sobretot si es dóna preferència a la forma dels triangles sobre la quantitat de triangles. No obstant les solucions que se'ns han acudit fins ara impliquen un esquema algorísmic que treballés globalment amb tots els pedaços alhora, la qual cosa sembla difícil de definir i costosíssima de dur a terme.
 - La relaxació treballa localment sobre cada cara del sòlid. Una alternativa podria ser un mètode de relaxació que permetés moure els punts al llarg de les corbes de retallat o fins i tot entre cares veïnes. Nogensmenys, hem pensat en desenvolupar un nou algorisme de triangulació que permetés situar triangles amb vèrtexs en cares veïnes (sempre que els pedaços de les dues cares s'unissin a l'aresta amb continuïtat suficient).
- L'algorisme de millora de la forma en espai imatge mostra que es pot millorar la qualitat de les triangulacions produïdes fins i tot limitant-se a fer servir el mateix nombre de vèrtexs i la mateixa configuració topològica. Una idea interessant podria ser desenvolupar nous algorismes per als casos en què l'objectiu de la forma dels triangles sigui prioritària o complementària a la minimització del nombre de triangles.
- Tal com també hem esmentat a les conclusions, l'algorisme que obté una CDT es pot fer servir en molts àmbits de treball. Atès que hem desenvolupat una generalització d'aquest algorisme incremental que permet manegar informació direccional associada a cada punt del pla, seria molt interessant fer servir aquest algorisme modificat a qualsevol lloc on la noció de direccionalitat tingui alguna rellevància. En concret, estem pensant en un algorisme de simplificació de malles triangulars que tingués en compte la curvatura direccional associada a cada vèrtex de la malla original.

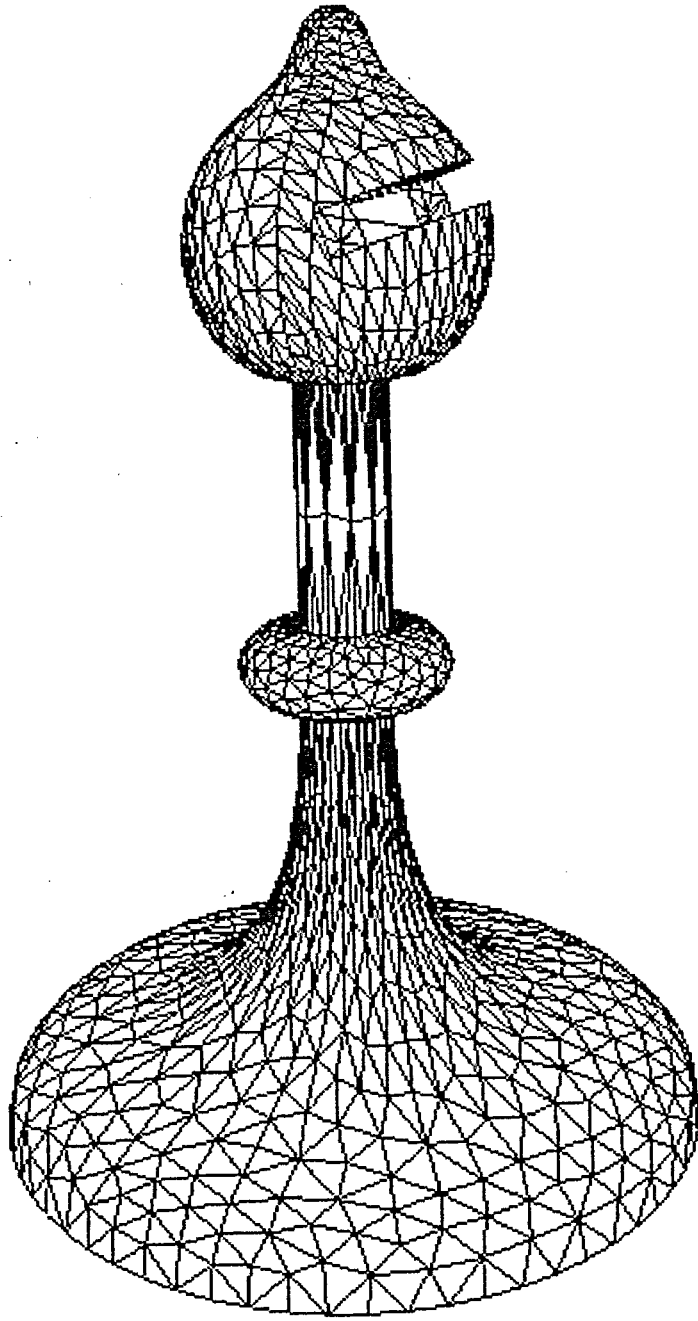


Figura 6.29: Objecte triangulat utilitzant l'algorisme isòtrop.



Figura 6.30: Escena composta per un conjunt d'objectes que han estat triangulats utilitzant els algorismes desenvolupats. A dalt a la dreta podem veure la mateixa escena visualitzada utilitzant l'algorisme de Phong.

Apèndix A

Triangulació de Delaunay restringida

A.1 Introducció

El problema de la obtenció de la triangulació de Delaunay d'un conjunt de punts donat i el seu graf dual de Voronoi ha estat estudiat extensivament per nombrosos autors, i s'han proposat diferents algorismes per solucionar-lo (vegeu [GS78], [Bow81], [GS85], [PS85], [Slo87] i [OBS92], entre molts d'altres). D'altres treballs, no tant nombrosos, s'ocupen de la obtenció d'una triangulació de Delaunay d'un conjunt de punts que inclogui a més a més un conjunt d'arestes donat (per exemple, [LL86], [Che89a], i [dFP92]). Aquest problema és conegut dins la geometria computacional amb el nom de la triangulació de Delaunay restringida pel conjunt d'arestes (CDT) i té aplicacions pràctiques en diverses àrees: triangulació de superfícies, anàlisi d'elements finits, modelatge de terrenys i reconstrucció d'objectes, entre d'altres.

La majoria d'algorismes proposats per obtenir una triangulació de Delaunay restringida (amb algunes excepcions, com [dFP92]) requereixen que tots els punts i arestes que formen l'entrada de l'algorisme siguin coneguts a priori. Aquesta condició no sempre es pot aconseguir, de vegades els punts i arestes es van adquirint de forma incremental. Algunes aplicacions, per exemple, requereixen calcular una triangulació inicial i refinar-la en certes zones, afegint-hi nous punts i arestes.

En aquest apèndix, presentem un algorisme incremental que donat un graf planar $\mathcal{G} = (\mathcal{A}, \mathcal{V})$ obté una triangulació de Delaunay del conjunt de vèrtexs \mathcal{V} restringida per les arestes de \mathcal{A} . Presentarem també dues generalitzacions de l'algorisme incremental que permeten treballar amb distàncies el·líptiques així com amb informació direccional diferent associada a cada punt del pla. El treball inclòs en aquest apèndix ha donat lloc a un article publicat en una revista, [Vig97], i a dos informes de recerca del departament de Llenguatges i Sistemes Informàtics, [Vig95] i [Vig98].

La notació que farem servir en aquest apèndix és la mateixa que la usada al llarg de tota la tesi que s'explica a la secció 2.4. No obstant, com que estem parlant d'entitats geomètriques en dues dimensions, sempre que ens referim a un graf estarem parlant d'un graf planar.

L'algorisme que volem obtenir serà incremental, és a dir, que obtindrà la CDT d'un graf afegint un a un cadascun dels punts i arestes del graf. En conseqüència, desenvoluparem dues accions, una primera que donada la CDT d'un graf $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ i un vèrtex $p \notin \mathcal{V}$ obtingui la CDT del graf augmentat $(\mathcal{V} \cup \{p\}, \mathcal{A})$; i una segona que donada una CDT d'un graf $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ i una aresta $ab \notin \mathcal{A}$, $a, b \in \mathcal{V}$, obtingui la CDT del graf augmentat $(\mathcal{V}, \mathcal{A} \cup \{ab\})$.

Les arestes del conjunt \mathcal{A} , les que restringeixen la triangulació, les anomenarem *arestes fixes*, donat que, pel fet de ser incremental, l'algorisme en cap moment les eliminarà de la CDT en curs.

Aquest plantejament coincideix amb el de [dFP92], no obstant difereix tant en la solució trobada per al mètode de vèrtexs com en el mètode d'inserció d'arestes.

A.2 Algorisme d'inserció de vèrtexs

El mètode d'inserció de punts que utilitzem és un mètode força conegut. Ha estat usat, entre d'altres, per Sloan amb la finalitat d'aconseguir un algorisme incremental que calcula la DT d'un conjunt de punts. Aquest mètode es basa en la següent proposició:

Proposició 9. *Sigui Δ una DT d'un conjunt de punts donat, S . Llavors, la inserció d'un punt $p \notin S$ interior a la regió triangulada, formant la nova DT de $S \cup \{p\}$, modifica únicament els triangles de Δ tals que el seu circumcercle conté el punt p .*

Per a una demostració d'aquesta proposició vegeu [Law77]. L'algorisme que se'n deriva comença generant un triangle que conté tots els punts dels quals es vol aconseguir la DT, per tal de garantir que tots els punts cauran dins la triangulació en curs. Els punts es van inserint un a un dins la triangulació; cada punt inserit implica fer un seguit d'intercanvis d'arestes compartides per dos triangles, fins aconseguir que tots els triangles tals que contenen el punt inserit s'hagin actualitzat. El mateix Lawson demostra que aquest procés iteratiu convergeix després d'un nombre finit de passos a la nova DT.

L'anterior proposició es pot generalitzar en el cas que es tracti de construir una CDT enlloc d'una DT. En aquest cas, sols cal considerar la condició addicional de visibilitat entre punts, obtenint la nova proposició:

Proposició 10. *Sigui Δ una CDT d'un conjunt de punts donat, S . Llavors, la inserció d'un punt $p \notin S$ interior a la regió triangulada, formant la nova CDT de $S \cup \{p\}$, modifica únicament els triangles de Δ tals que:*

- (i) *el seu circumcercle conté el punt p ; i*
- (ii) *els tres vèrtexs del triangle són visibles des de p .*

La demostració d'aquesta altra proposició es pot trobar a [dFP92]. Aquest plantejament ens duu a proposar el mateix mètode per a la inserció d'un punt dins una CDT que el de la proposta [Slo87], però tenint en compte que les arestes que restringeixen la CDT no poden ser mai alterades. En conseqüència, n'hi ha prou amb mantenir un indicador a l'estructura de dades que identifiqui les arestes fixes, és a dir, les que restringeixen la triangulació. Durant el procés d'intercanvi d'arestes, impedim que les arestes fixes es modifiquin. A continuació, presentem la rutina escrita en pseudocodi per inserir un punt a una CDT, amb la precondició que el punt ha de ser interior a la regió del pla recoberta per la CDT.

```

acció AfegirPuntCDT( $\mathcal{T} : CDT, p : Vèrtex$ )
  {Prec:  $p \notin \mathcal{T} \wedge p \in Interior(\mathcal{T})$ }
  pila := PilaTrianglesBuida
  Trobar el triangle  $t \in \mathcal{T}$  que conté  $p$ 
  Dividir  $t$  en tres triangles,  $t_1, t_2$  i  $t_3$ , segons  $p$ 
  Push(pila,  $t_1$ )
  Push(pila,  $t_2$ )
  Push(pila,  $t_3$ )
  mentre  $\neg EsBuida(pila)$  fer
     $t := Pop(pila)$ 
     $t_{opo} := TriangleOposat(t, p)$ 
    si l'aresta compartida per  $t$  i  $t_{opo}$  no és fixa i
       $p \in Circumcercle(t_{opo})$  llavors
        Intercanviar l'aresta compartida per  $t$  i  $t_{opo}$ 
        Push(pila,  $t$ )
        Push(pila,  $t_{opo}$ )
    fsi
  fmentre
facció

```

El pas que divideix un triangle t en tres segons un punt interior p , forma tres nous triangles. Si els vèrtexs del triangle t eren v_1, v_2 i v_3 , els nous triangles seran $t_1 = (v_1, v_2, p)$, $t_2 = (v_2, v_3, p)$ i $t_3 = (v_3, v_1, p)$. Aquesta divisió està representada a la figura A.1. Aquesta acció i totes les que canvien la triangulació \mathcal{T} actualitzen la informació adequadament, de forma que s'introdueixen els nous triangles necessaris, s'esborren els vells i s'actualitzen les relacions de veïnatge.

La rutina `TriangleOposat` donat un triangle t i un dels seus vèrtexs, p , retorna el triangle veí de t tal que és oposat a p , és a dir, el triangle veí de t que no comparteix el vèrtex p .

L'operació d'intercanvi d'arestes, coneguda en anglès com a *edge swap*, donats dos triangles veïns que formen un quadrilàter convex, canvia l'aresta compartida pels triangles per la que uneix els seus vèrtexs no compartits. Aquesta operació també està representada a la figura A.1. Com que estem parlant de triangulacions, és a dir, de grafs planars, aquesta operació solament serà vàlida quan els dos triangles veïns formen un quadrilàter convex.

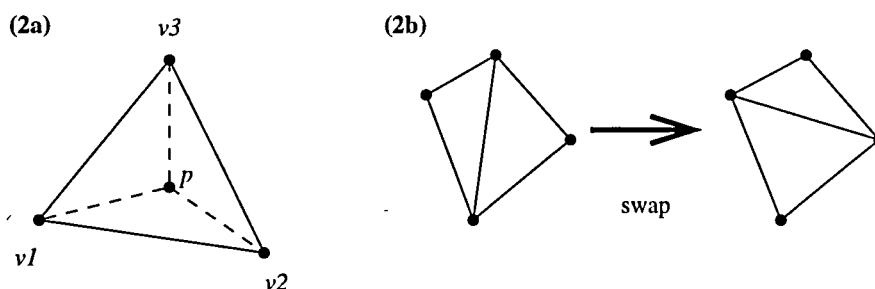


Figura A.1: Operació de divisió d'un triangle segons un punt interior i operació d'intercanvi d'arestes.

Lema 5. *Siguin $t_1 = T(a, b, q)$ i $t_2 = T(q, b, a)$ dos triangles veïns d'una CDT. Llavors, tot punt interior al $CC(t_1)$ oposat a c per l'aresta ab és també interior al $CC(t_2)$.*

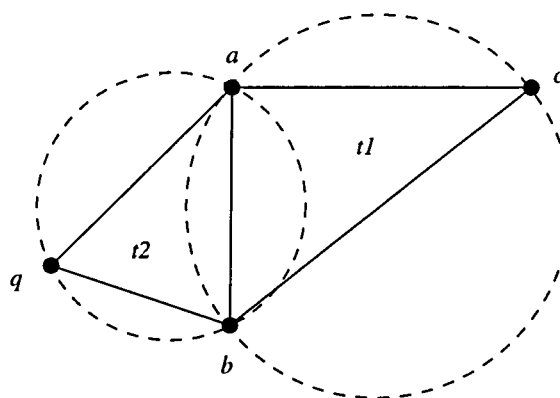


Figura A.2: Representació del lema 5

Demostració. Com que t_1 i t_2 són triangles d'una CDT, compleixen la condició del circumcercler buit. En concret, el punt q serà exterior a $CC(t_1)$. En conseqüència, els dos circumcerclers es tallen en els punts a i b (vegeu la

figura A.2) i la recta que passa que per aquests dos punts divideix el cercle $CC(t_1)$ en dues parts, una de les quals queda inclosa dins $CC(t_2)$ i l'altra, que correspon a la zona on hi ha el vèrtex c . Per tant, tot punt dins de $CC(t_1)$ oposat a c per l'aresta ab serà interior al $CC(t_2)$. \square

Proposició 11. *L'algorisme anterior, basat en l'intercanvi d'arestes, donada una CDT d'un graf $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ i un vèrtex $p \notin \mathcal{A}$ obté la CDT del graf augmentat $\mathcal{G}' = (\mathcal{V} \cup \{p\}, \mathcal{A})$*

Demostració. La proposició 10 ens assegura que els únics triangles $t = T(a, b, c)$ que es modificaran són tals que contenen el vèrtex p i que els seus vèrtexs a, b i c siguin visibles de p . A més, sabem que aquests triangles formen una regió està delimitada per un polígon de forma estrellada Q_p tal que la CDT de \mathcal{G}' s'obté unint tots els seus vèrtexs amb p [dFP92]. Considerem ara un triangle qualsevol $t \in Q_p$ i vegem que, efectivament, el nostre algorisme acabarà unint p amb els seus vèrtexs de manera que es conservin les fronteres externes de Q_p .

En primer lloc, vegem que en el cas que p sigui interior al triangle t , la part inicial de l'algorisme dividirà el triangle en tres i, per tant, els vèrtexs de t quedaran units amb p .

En cas contrari, en què p sigui exterior a t , notem que hi ha un únic vèrtex de t tal que l'aresta del triangle oposada al vèrtex deixa el vèrtex i el punt p en semiplans diferents. Anomenarem c a aquest vèrtex de t i els altres dos vèrtexs del triangle els anomenarem a i b (vegeu la figura A.3).

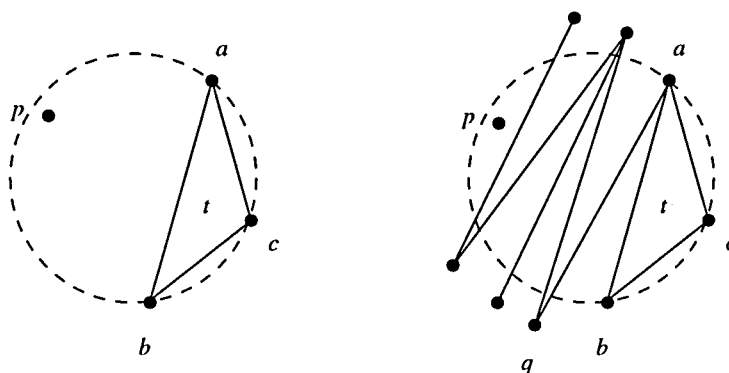


Figura A.3: El punt p està contingut dins el circumcentre de t però no dins del propi triangle t .

Com que t és un triangle de la CDT del graf original, el seu circumcercle, $CC(a, b, c)$ no conté cap punt de \mathcal{V} . Conseqüentment, existeixen una sèrie d'arestes de la CDT original tals que separen el vèrtex p del vèrtex c , amb els seus extrems no interiors a $CC(a, b, c)$, que podem ordenar de més a menys propera al punt p (vegeu la figura A.3). Fixem-nos que cap d'aquestes arestes

no pot ser fixa, donat que altrament el vèrtex c no seria visible des de p . Farem inducció sobre el nombre d'arestes amb aquestes característiques.

Com a mínim, hi haurà una aresta d'aquestes característiques, l'aresta ab del propi triangle t . Per tant, el cas base de la inducció és que sols hi hagi una aresta, la ab . En aquest cas, l'algorisme haurà introduït p dins el triangle veí de t segons l'aresta ab (dividint el triangle en tres) i, per tant, s'haurà format el triangle $T(a, b, p)$, el qual s'haurà afegit a la pila el triangle t . L'algorisme desempilarà aquest triangle en algun moment i considerarà (i efectuarà, en comprovar que és vàlid) l'intercanvi de l'aresta ab per la cp .

Passem al cas en què existeixin n arestes separadores, és a dir, sigui $t = T(a, b, c)$ un triangle de la CDT inicial tal que el seu circumcerclé conté el vèrtex p , amb els seus tres vèrtexs visibles des de p i que està separat de p per n arestes (inclosa l'aresta ab). Sense pèrdua de generalitat, suposem que la següent aresta d'entre les separen el triangle de p és l'aresta aq (vegeu la figura A.3; l'altre cas possible és que la següent aresta sigui del tipus br i es resoluria de forma simètrica). En aquestes condicions, el lema 5 ens assegura que $CC(a, b, q)$ conté p , donat que els triangles t i $T(q, b, a)$ són triangles veïns de de la CDT original, el punt p és interior a $CC(a, b, c)$ i la recta que passa per a i b deixa a costats oposats els punts q i c . D'altra banda, el vèrtex q és visible des de p donat que $T(a, b, q)$ era un triangle de la CDT original i, per tant, $CC(a, b, q)$ no conté cap altre punt del conjunt \mathcal{V} .

Per hipòtesi d'inducció, en ser el triangle $T(a, b, q)$ un triangle separat de p per $n - 1$ arestes, amb tots els seus vèrtexs visibles des de p i tal que el seu circumcerclé conté a p , l'algorisme en algun moment haurà fet els intercanvis d'arestes per arribar a obtenir els triangles $T(p, b, q)$ i $T(a, b, p)$. Per tant, s'haurà empilat el triangle t (veí de $T(a, b, p)$) i l'algorisme considerarà l'intercanvi de l'aresta ab per la pc . \square

A.3 Algorisme d'inserció d'arestes

L'algorisme per inserir una aresta ab en una CDT d'un graf segueix l'esquema següent:

1. Eliminar de la CDT els triangles t_1, \dots, t_k tallats per ab , amb el què quedarà una regió sense triangular.
2. Afegir al resultat l'aresta ab .
3. Retriangular les regions superior i inferior de l'aresta ab que han quedat sense triangular en el primer pas.

La figura A.4 mostra el procés d'inserció d'una aresta dins una CDT. Seguidament, detallarem cadascun d'aquests passos i comprovarem que la nova triangulació obtinguda és la CDT que busquem.

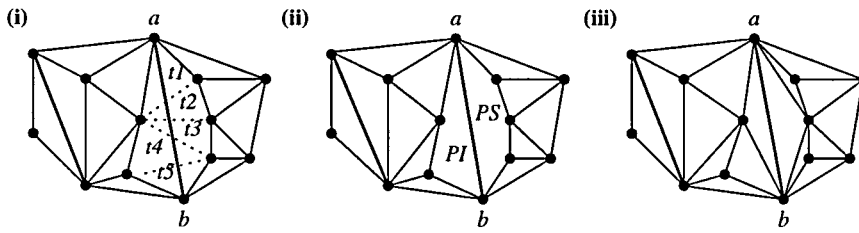


Figura A.4: Inserció d'una aresta dins una CDT: (i) Localització i eliminació dels triangles tallats per l'aresta; (ii) Inserció de l'aresta (iii) Retriangulació de les regions veïnes a l'aresta.

El primer pas implica trobar en primer lloc el triangle de la CDT que conté el vèrtex a i que és tallat pel segment ab . La localització del triangle d'una triangulació tal que conté un punt és una acció ben coneguda; la forma més efectiva de realitzar-la consisteix en partir d'un triangle qualsevol de la triangulació i anar-se desplaçant cap a triangles veïns, segons la classificació del punt en qüestió respecte a les arestes del triangle, fins arribar al triangle que conté el punt. Aquest mateix procediment és vàlid per a la localització d'un triangle qualsevol, t , de la triangulació tal que tingui com a un dels seus vèrtexs el punt a .

A partir del triangle t , podem fer servir la relació de veïnatge entre els triangles per recórrer els triangles que convergeixen a a , girant en sentit horari, fins trobar el triangle tal que és tallat pel segment ab , al qual anomenarem t_1 (vegeu la figura A.5).

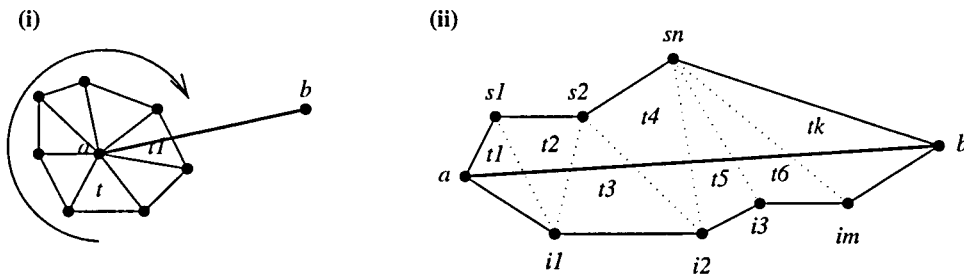


Figura A.5: Inserció d'una aresta: (i) Localització del triangle t_1 que conté a i que és tallat per ab (ii) Eliminació dels triangles t_1, \dots, t_k tallats per ab .

Un cop s'ha obtingut el triangle t_1 , la resta de triangles t_2, \dots, t_k tallats per ab es troben fàcilment usant les mateixes relacions de veïnatge i el test de classificació d'un punt respecte a una recta. Aquest procés ens permet ara classificar els vèrtexs dels triangles $\{t_j\}_{j=1}^k$ en dos conjunts, el dels vèrtexs superiors a l'aresta ab , $\{s_j\}_{j=1}^n$, i el dels inferiors $\{i_j\}_{j=1}^m$. La sèrie

de vèrtexs $\{a, s_1, \dots, s_n, b\}$ tanquen una regió de l'espai situada per sobre l'aresta ab , a la qual anomenarem *pseudopolígon superior* i notarem com P_S ; de la mateixa manera, els vèrtexs $\{b, i_1, \dots, i_m, a\}$ limiten una regió de l'espai que anomenarem *pseudopolígon inferior* i notarem com P_I . Aquestes dues regions P_S i P_I són les que caldrà tornar a triangular.

Fem notar que, estrictament parlant, les regions P_S i P_I poden no ser un polígon tal com convencionalment s'entén, donat que poden contenir vèrtexs repetits dins la sèrie de vèrtexs que formen el seu contorn¹. La figura A.6 mostra un exemple d'un cas en què el pseudopolígon superior conté vèrtexs repetits. Aquest fet no complica el mètode proposat per a la retriangulació dels pseudopolígons més que en la part en què es refà el lligam entre els nous triangles generats per la retriangulació i la triangulació ja existent.

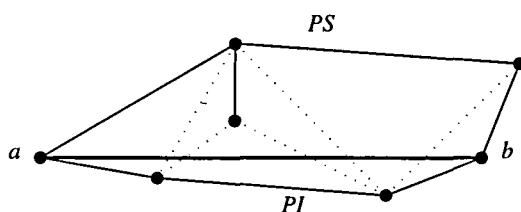


Figura A.6: Exemple en què els pseudopolígon superior conté vèrtexs repetits

La forma com es retriangula l'interior dels pseudopolígons P_I i P_S per obtenir la nova CDT és mitjançant un algorisme recursiu. Cadascun dels dos pseudopolígons es triangula per separat, donat que la nova CDT no pot contenir arestes amb un extrem en P_S i l'altre en P_I tal que tallin ab , ja que aquesta aresta impedeix la seva visibilitat. Descriurem, doncs, l'algorisme per al cas de P_S ; per a P_I els mateixos raonaments són vàlids.

En primer lloc, fixem-nos que tota triangulació del pseudopolígon P_S ha de contenir forçosament un sol triangle amb ab com una de les seves arestes, és a dir, un cert triangle amb vèrtexs a, b i c , on $c = s_l, 1 \leq l \leq n$. Com que la triangulació que es vol obtenir ha de ser de Delaunay, el triangle $T(a, b, c)$ ha de complir la condició del circumcercle buit, és a dir, que, en concret, el seu circumcercle no pot contenir cap vèrtex de P_S (a part dels seus tres extrems, a, b i c). D'aquesta propietat se'n deriva l'algorisme recursiu per triangular l'interior de P_S (figura A.7):

- Buscar el vèrtex $c = s_l$ tal que el circumcercle de $T(a, b, c)$ no contingui cap altre punt de P_S ;
- formar el triangle $T(a, b, c)$, que divideix P_S dues subregions, $P_E = \{a, s_1, \dots, s_l\}$ i $P_D = \{s_l, \dots, s_n, b\}$;

¹Aquesta circumstància no s'especifica explícitament en l'algorisme proposat en [dFP92]

- aplicar l'esquema recursivament a les regions P_E i P_D , respecte a les arestes ac i cb respectivament en cada cas.

El cas base de la recursivitat és aquell en què la regió que cal triangular està limitada per tres o menys punts, casos trivials de tractar.

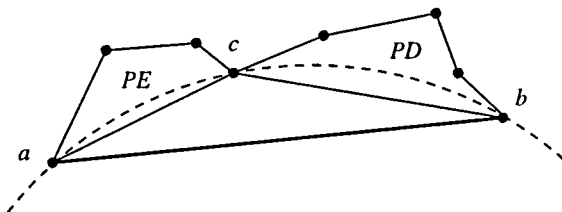


Figura A.7: Representació de l'algorisme recursiu per triangular un pseudopolígon.

A continuació veurem que la triangulació que resulta de l'aplicació de l'algorisme descrit és la CDT del graf original al qual se li ha afegit l'aresta ab .

Proposició 12. *Sigui Δ una CDT triangulació d'un graf donat $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ i sigui ab una aresta tal que $ab \notin \mathcal{A}$, $a, b \in \mathcal{V}$. Llavors, la triangulació obtinguda eliminant les arestes de Δ tallades per ab , inserir l'aresta ab i retriangulant els pseudopolígons P_S i P_I (resultants de l'eliminació d'arestes) és la CDT del graf $\mathcal{G} = (\mathcal{V}, \mathcal{A} \cup \{ab\})$.*

Demostració. El segment ab ha de ser una aresta de la nova CDT; per tant, com que la triangulació és un graf planar, les arestes tallades per ab no poden formar part de la nova CDT i poden ser eliminades. Atès que els punts del graf són els mateixos que els de la CDT original, les arestes que hauran quedat continuen complint la condició del circumcercler buit, i per tant segueixen essent arestes vàlides de la nova CDT. En conseqüència, la nova CDT es pot obtenir fusionant el que resta de la CDT després d'eliminar les arestes amb la triangulació de Delaunay dels pseudopolígons P_S i P_I , restringits per les arestes que els formen el seu contorn.

□

A continuació presentem la rutina que insereix una aresta a una CDT i el refinament de l'acció que calcula la triangulació de Delaunay d'un pseudopolígon.

acció AfegirArestaCDT($\mathcal{T} : \text{CDT}, ab : \text{Aresta}$)
 {Prec: $a, b \in \mathcal{T} \wedge ab \notin \mathcal{T}$ }
 Trobar el triangle $t \in \mathcal{T}$ que té com a vèrtex a
 i que és tallat per ab

```

PS := LlistaVèrtexsBuida()
PI := LlistaVèrtexsBuida()
v := a
mentre t no conté b fer
  tseg := TriangleOposat(t, v)
  vseg := VèrtexOposat(tseg, t)
  si vseg per sobre l'aresta ab llavors
    AfegirLlista(PS, vseg)
    v := Vèrtex comú a t i tseg que està per sobre ab
  si no
    AfegirLlista(PI, vseg)
    v := Vèrtex comú a t i tseg que està per sota ab
  fsi
  Eliminar t de  $\mathcal{T}$ 
  t := tseg
fmentre
  TriangularPseudopolígonCDT(PS, ab,  $\mathcal{T}$ )
  TriangularPseudopolígonCDT(PI, ab,  $\mathcal{T}$ )
  Refer els veïnatges de  $\mathcal{T}$ 
  Afegir l'aresta ab dins  $\mathcal{T}$ 
  Marcar l'aresta ab dins  $\mathcal{T}$  com a fixa
facció

```

La rutina TriangleOposat és la mateixa que la de l'algorisme d'inserció de punts i la funció VèrtexOposat donats dos triangles veïns, t i t_{opo} , retorna el vèrtex de t_{opo} que no pertany a t .

El pas en què es refan els veïnatges interiors és necessari per tal de reconstruir les relacions de veïnatge entre els nous triangles resultants de la triangulació de l'interior dels pseudopolígons i els triangles de la CDT inicial. Cal anar amb compte en el cas que un pseudopolígon contingui punts repetits, ja que en aquest cas és possible que calgui lligar com a veïns dos triangles de l'interior del pseudopolígon a través d'una aresta que ja existia en la CDT inicial.

```

acció TriangularPseudopolígonCDT(P : LlistaVèrtexs,
                                   ab : Aresta,  $\mathcal{T}$  : CDT)
  si P té més d'un element llavors
    c := PrimerVèrtex(P)
    per cada vèrtex v ∈ P fer
      si v ∈ Circumcercl(a, b, c) llavors
        c := v
      fsi
    fper

```

```

    Dividir  $P$  en  $P_E$  i  $P_D$ , on  $P = P_E + \{c\} + P_D$ 
    TriangularPseudopoligonCDT( $P_E, ac, \mathcal{T}$ )
    TriangularPseudopoligonCDT( $P_D, cb, \mathcal{T}$ )
  fsi
  si  $P$  no és buida llavors
    Afegir el triangle de vèrtexs  $a, b, c$  dins  $\mathcal{T}$ 
  fsi
facció

```

A.4 Algorisme de triangulació de Delaunay d'un polígon

Els algorismes anteriors permeten aconseguir una CDT d'un graf genèric qualsevol. En concret, poden servir per aconseguir la triangulació de Delaunay d'un polígon.

La triangulació de Delaunay d'un polígon no és més que la CDT del seu graf, és a dir, la DT dels seus vèrtexs restringida pel conjunt de les seves arestes. En conseqüència, podem utilitzar els algorismes que acabem d'exposar per aconseguir la CDT del polígon, afegint els vèrtexs i arestes del polígon de forma incremental. Cal, però, partir d'una CDT inicial que contingui tots els punts del polígon, donat que l'algorisme d'inserció de punts així ho exigeix. Com en la proposta [Slo87], la CDT inicial que prenem és la formada per un sol triangle (el *supertriangle*) que conté el polígon, el qual es pot calcular fàcilment a partir de la caixa contenidora dels vèrtexs del polígon.

Habitualment, quan es parla de la triangulació d'un polígon s'entén que s'està parlant d'un recobriment de l'interior del polígon per triangles (disjunts). En el nostre cas, la CDT del polígon que haurem obtingut no sols recobreix l'interior, sinó també part de l'exterior (inclou tota la regió del supertriangle). Per aconseguir un recobriment del polígon, n'hi ha prou amb retirar de la CDT resultant els triangles que són exteriors al polígon. Això es pot aconseguir de forma senzilla mitjançant un algorisme de gèrmen que, partint d'un triangle exterior (qualsevol que tingui un dels vèrtexs del supertriangle és vàlid), recorre tots els triangles de la CDT seguint la relació de veïnatge. A cada pas, va marcant cada triangle com a exterior o interior, depenent de la paritat del nombre de cops que hagi avançat a través d'una aresta fixa dins la CDT. Acabat aquest procés de marcatge, n'hi ha prou amb quedar-se amb els triangles identificats com a interiors.

Atès que l'algorisme és incremental, deixa llibertat sobre l'ordre en què s'introdueixen els punts i les arestes del graf. Ja que el cost de la introducció d'una aresta dins una CDT depèn del nombre de triangles que quedin tallats per l'aresta, convé afegir primer les arestes com més aviat millor a la CDT

(d'aquesta manera, tallaran menys triangles). Una altra raó per introduir les arestes el més aviat possible és que d'aquesta manera impedeixen que es realitzin accions de *swap* que altrament podrien arribar-se a realitzar en inserir nous punts a la CDT (recordem que les arestes fixes limiten la visibilitat entre punts i , per tant, tallen el procés d'intercanvi d'arestes). Per la mateixa raó, és recomanable inserir com més aviat millor les arestes de l'interior del graf, donat que és probable que limitin la visibilitat dels punts més exteriors.

Aquestes consideracions ens han dut a proposar un esquema per a la triangulació d'un polígon que segueixi les premisses següents:

- Inserir els punts i les arestes en l'ordre definit pels polígons que limiten el polígon.
- Inserir en primer lloc els vèrtexs i arestes dels forats del polígon i posteriorment els vèrtexs i arestes del polígon exterior.

La primera condició està basada en l'observació que habitualment dos vèrtexs que formen una aresta el contorn d'un polígon són propers entre ells, i, per tant, en inserir-la dins la CDT en curs no tallarà gaire triangles. A continuació presentem la nostra proposta d'un algorisme per triangular un polígon basat en la introducció incremental dels vèrtexs i arestes dels polígons que la formen.

```

funció CDTPol( $C$  : Polígon) retorna Triangulació
  Calcular el supertriangle que conté C
   $\mathcal{T} :=$  triangulació que sols conté el supertriangle
  per cada polígon forat P de C fer
    per cada vèrtex v de P fer
      AfegirPuntCDT( $\mathcal{T}, v$ )
       $x :=$  SegüentVèrtex( $v, P$ )
      AfegirArestaCDT( $\mathcal{T}, vx$ )
    fper
  fper
   $P :=$  Polígon exterior de C
  per cada vèrtex v de P fer
    AfegirPuntCDT( $\mathcal{T}, v$ )
     $x :=$  SegüentVèrtex( $v, P$ )
    AfegirArestaCDT( $\mathcal{T}, vx$ )
  fper
  retorna  $\mathcal{T}$ 
ffunció

```

La figura A.8 mostra cinc exemples d'aplicació d'aquest algorisme de triangulació de polígons. El cas (a) és un polígon simple; el cas (b) és un

polígon amb vèrtexs al seu interior; el cas (c) també és un polígon amb punts al seu interior que ha estat especialment dissenyat per comprovar que l'algorisme sap tractar la inclusió d'arestes de restricció fins i tot en casos complexos; el cas (d) és un polígon amb forats; i el cas (e) és un polígon amb forats i vèrtexs interiors.

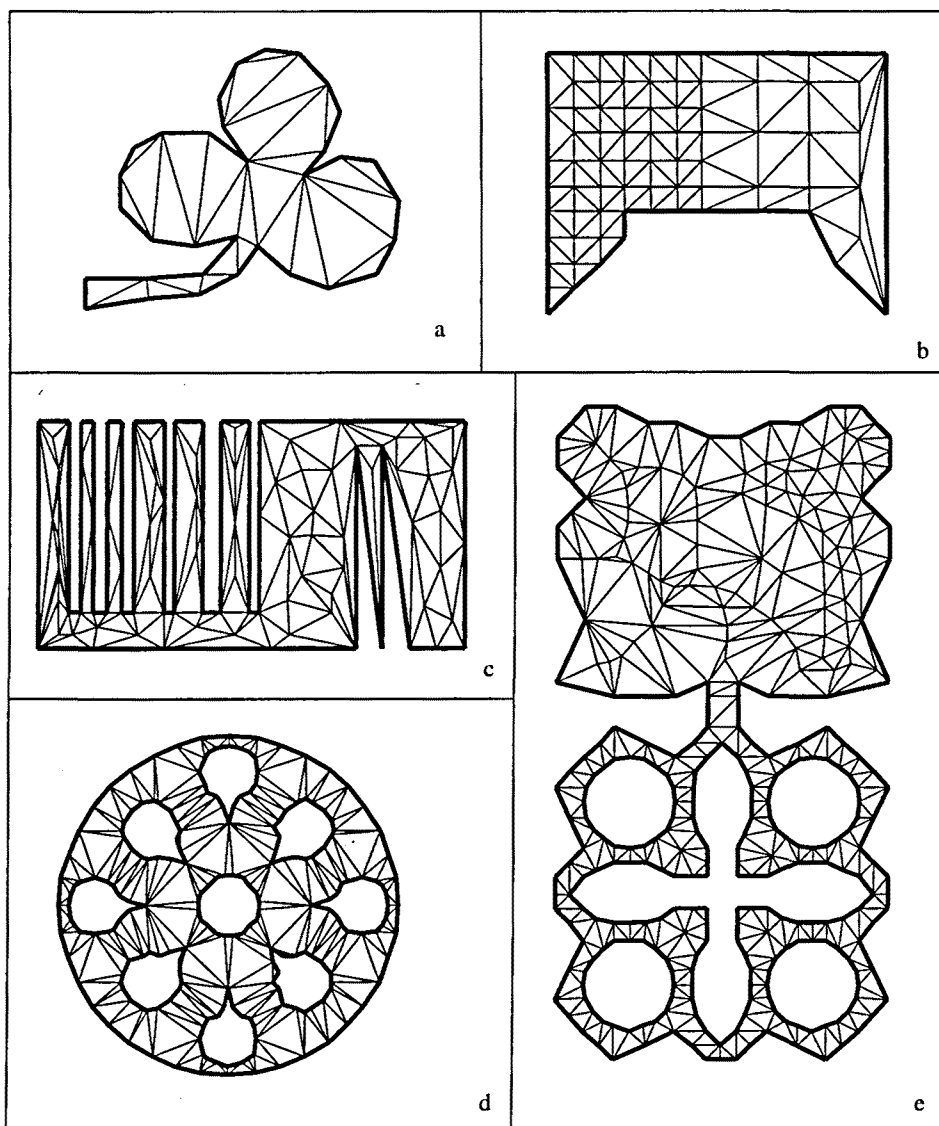


Figura A.8: Diversos exemples d'aplicació d'obtenció de la CDT de diversos polígons, alguns d'ells amb forats i vèrtexs addicional al seu interior.

A.5 Adaptació dels algorismes per a la distància induïda per una el·lipse

L'objectiu d'aquesta secció és generalitzar els algorismes incrementals que acabem de presentar de manera que serveixin per obtenir la CDT d'un graf tenint en compte que la distància entre els punts no és l'habitual (l'euclidiana en el pla), sinó la determinada per una el·lipse deformadora del pla, o fins i tot que tinguem una funció que determini una deformació diferent per a cada punt del pla.

Atès que introduïm la noció de deformació, utilitzarem una notació per diferenciar els dos plans: \mathbb{D} serà el pla original, mentre que el pla deformat l'anomenarem \mathbb{D}_E . Així mateix, com hem dit disposem d'una el·lipse que ens indica com transformar el pla, i determina la distància; aquesta el·lipse, que anomenarem el·lipse deformadora, la notarem per E . Farem dues aproximacions al problema: una primera en la qual l'el·lipse deformadora és única per a tot el pla, i una segona en la qual E serà diferent per a cada punt del pla.

Les triangulacions que volem obtenir són tals que tenen en compte la transformació del pla determinada per l'el·lipse (o el·lipses) deformadora, de tal manera que les arestes de la triangulació resultant s'aliniïn seguint la direcció dels eixos d'aquestes el·lipses (per fer-se una idea a priori dels resultats que desitgem obtenir, vegeu les figures A.9 i A.10).

Altres autors han utilitzat la noció de triangulació de Delaunay generalitzada per a distàncies no habituals, alguns d'ells amb finalitats properes a la nostra. Així, en [CD86] es generalitza un algorisme per obtenir la DT d'un conjunt de punts per a una distància convexa qualsevol, en [NF89] s'inclou una generalització del graf de Voronoi i de la DT per a normes invariants a transformacions afins i [OBS92] descriu diverses generalitzacions del graf de Voronoi per a distàncies no habituals (com distàncies en les quals cada punt del pla té un pes assignat, etc). Potser els mètodes més propers al que presentem en aquesta secció siguin els descrits en [DS89], [Mav90] i [Pos93].

El primer d'aquests tres articles, [DS89], s'ocupa de triangular una funció quadràtica convexa (univaluada) de forma que es minimitzi l'error d'aproximació. Els autors afirmen que l'objectiu de minimitzar aquest error és contrari a optimitzar els angles dels triangles. L'algorisme que obté la triangulació que recobreix l'envolupant convexa d'un conjunt de punts donat de tal manera que l'error d'aproximació és el mínim. L'algorisme és una modificació d'un mètode ben conegut per obtenir una TD d'un conjunt de punts, el qual consisteix en partir d'una triangulació inicial qualsevol i anar efectant intercanvis d'arestes fins obtenir la TD (no restringida). La modificació que introdueixen a aquest algorisme és el criteri per realitzar els intercanvis d'arestes, que es basa en els punts transformats segons una el·lipse calculada a partir de les curvatures principals de la superfície quadràtica. Fem notar

que aquesta transformació és única per a tots els vèrtexs.

El mètode que es presenta a [Mav90] serveix per obtenir una malla adaptable d'un flux viscos de manera que es tingui en compte una funció de deformació direccional (que proporciona per a cada punt del pla un vector de deformació, és a dir, una magnitud i una direcció). La solució que s'hi proposa és, efectivament, similar a la nostra: modificar un algorisme del càlcul d'una DT de manera que tingui en compte la deformació direccional, de manera que el mètode resultant, tal com afirma el mateix autor, es pot veure com un procediment de construcció d'una DT on els circumcercles dels triangles en espai deformat corresponen a el·lipses en l'espai original.

El tercer d'aquests articles, [Pos93], és una proposta d'un mètode per obtenir una DT generalitzat per al càlcul d'una malla triangular, que suposadament també s'utilitzaria per resoldre problemes de dinàmica de fluids mitjançant la tècnica dels FEM. La proposta suposa que una de les dades d'entrada és una funció que per a cada punt del pla proporciona la seva "circum-forma" (generalitzant la idea de circumcercle del cas de les triangulacions de Delaunay), la qual s'imposa que sigui convexa, però no té perquè ser una el·lipse.

En el nostre cas la funció que deforma el pla, determinada per les el·lipses, serà una transformació lineal. Notem que la nostra proposta és una generalització de l'algorisme de càlcul d'una CDT, és a dir, que no només admet un conjunt de punts sinó també arestes restrictives.

Tal com remarquen dos dels autors dels articles que acabem d'esmentar, cal que la funció de deformació sigui "suau", és a dir, que variï poc comparada amb la densitat dels punts que es volen triangular [Mav90, Pos93]. Efectivament, si tenim una ràpida variació de la funció de deformació entre dos punts propers, com que els mètodes proposats treballen prenent el valor de la funció assignat a un punt, com si en un entorn fos constant, la triangulació resultant no podrà reflectir aquests canvis. En el nostre cas, l'algorisme que proposem també té aquesta característica, ja que, com veurem, treballa localment per a cadascun dels punts o arestes inserits, és a dir, suposa que l'el·lipse deformadora del pla és constant per a cada pas incremental que realitza.

Tot seguit, definirem la transformació del pla induïda per una el·lipse qualsevol i presentarem les dues generalitzacions del mètode incremental tenint en compte aquestes deformacions.

A.5.1 Transformació induïda per una el·lipse

Disposem d'una el·lipse E que té com a matriu associada M_E . Anomenarem λ_1 i λ_2 als valors propis de la matriu, i els vectors propis, que indiquen la direcció dels eixos principals de l'el·lipse, els anomenarem $v_1 = (e_1, e_2)$ i $v_2 = (e_2, -e_1)$. Així doncs, tenim que els punts de l'el·lipse $x \in \mathbb{D}$ són aquells

$x \in \mathbb{D}$ que compleixen que

$$x^T M_E x = 1$$

amb

$$M_E = \begin{pmatrix} e_1 & e_2 \\ e_2 & -e_1 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} e_1 & e_2 \\ e_2 & -e_1 \end{pmatrix} = \begin{pmatrix} \lambda_1 e_1^2 + \lambda_2 e_2^2 & (\lambda_1 - \lambda_2) e_1 e_2 \\ (\lambda_1 - \lambda_2) e_1 e_2 & \lambda_1 e_2^2 + \lambda_2 e_1^2 \end{pmatrix}$$

Volem aconseguir una funció de deformació de l'espai, $f_E : \mathbb{D} \rightarrow \mathbb{D}_E$, tal que es compleixi que tres punts estan sobre l'el·lipse E si i sols si les seves imatges segons f_E estan sobre un cercle unitari en \mathbb{D}_E . És a dir, que el que volem aconseguir és que dos punts en \mathbb{D} estiguin a distància el·líptica r si i només si estan a distància r en \mathbb{D}_E (vegeu la definició 37, de distància el·líptica).

Per tant, la funció f_E és una transformació lineal, que podem definir de la següent manera:

$$f_E(x) = Mx$$

on

$$\begin{aligned} M &= \begin{pmatrix} e_1 & e_2 \\ e_2 & -e_1 \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \begin{pmatrix} e_1 & e_2 \\ e_2 & -e_1 \end{pmatrix} = \\ &= \begin{pmatrix} \sqrt{\lambda_1} e_1^2 + \sqrt{\lambda_2} e_2^2 & (\sqrt{\lambda_1} - \sqrt{\lambda_2}) e_1 e_2 \\ (\sqrt{\lambda_1} - \sqrt{\lambda_2}) e_1 e_2 & \sqrt{\lambda_1} e_2^2 + \sqrt{\lambda_2} e_1^2 \end{pmatrix} \end{aligned}$$

Anomenarem f_E **funció de transformació induïda per l'el·lipse E** . Notem que una alternativa hauria estat definir la funció de transformació com que un sol gir i un escalat; és a dir, si tinguéssim

$$M' = \begin{pmatrix} e_1 & e_2 \\ e_2 & -e_1 \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix},$$

la condició que hem imposat referent a les distàncies també es compliria.

A.5.2 El mètode incremental generalitzat

El mètode incremental que obté una CDT d'un graf general es basa en dos algorismes, un que afegeix un vèrtex al graf i un altre per afegir una aresta entre dos vèrtexs del graf. Les comprovacions geomètriques que realitzen ambdós algorismes per garantir que la triangulació resultant sigui de Delaunay queden recollides pel test que comprova si un punt és interior al circumcercle d'un triangle. Per tant, n'hi haurà prou amb adaptar la rutina que realitza aquest test, de manera que tingui en compte la distància el·líptica enlloc de la distància euclídia habitual, per obtenir el nou mètode.

Tinguem en compte que gràcies al fet que l'algorisme que hem desenvolupat per obtenir una CDT d'un graf és incremental, el test de distància el·líptica no afecta a tots els vèrtexs del graf, sinó tant sols als d'un entorn del vèrtex que s'insereix; per tant, és només a aquest subconjunt reduït de vèrtexs a qui caldrà aplicar les deformacions per al càlcul de la distància el·líptica.

La manera més senzilla d'adaptar aquesta rutina possiblement sigui transformar els punts que rep com a paràmetres d'entrada segons la funció de transformació induïda per l'el·lipse, per tot seguit realitzar el test habitual. Cal fer notar que, com que es tracta d'una transformació lineal, els canvis topològics que realitzen els algorismes (intercanvis d'arestes en el cas de l'algorisme d'inserció d'un vèrtex, triangulació de pseudopolígons en el cas de la inserció d'una aresta) seran canvis vàlids en espai deformat: una transformació lineal no afecta a la condició de convexitat ni tampoc a la de visibilitat entre entitats.

Recordem que el primer pas de l'algorisme general que triangula un graf consisteix en escalar els vèrtexs del graf, de forma que les seves coordenades quedin normalitzades, i tot seguit afegir un supertriangle, perquè la rutina d'inserció d'un punt a la CDT requereix que els punts inserits siguin interiors a la triangulació en curs (vegeu la secció A.4). Per tant, cal tenir en compte que la deformació l'hem d'aplicar sobre les coordenades originals dels vèrtexs, abans de ser normalitzades, i també cal anar en compte amb els vèrtexs del supertriangle, atès que pot no tenir sentit calcular-hi la funció de transformació f_E .

Tot seguit, presentem dues generalitzacions del mètode d'obtenció d'una CDT: en el primer cas, suposem que existeix una única el·lipse que deforma tot l'espai segons la mateixa transformació, mentre que en el segon disposem d'una funció que proporciona una el·lipse diferent per a cada punt del pla.

Cas I: Existeix una sola el·lipse que determina la deformació de l'espai

Seguint la notació de l'apartat anterior, ens referirem a l'el·lipse com a E i la matriu de deformació de l'espai serà M . En aquest cas, el mètode incremental d'obtenció d'una CDT pot treballar en espai deformat, \mathbb{D}_E , o sigui, tenint en compte la distància el·líptica induïda per E . Per tant, pel què fa a la implementació, el nou mètode pot transformar les coordenades dels vèrtexs del graf segons la funció de transformació induïda per E , aplicar el mètode incremental per obtenir la CDT, i finalment recuperar les coordenades dels vèrtexs en l'espai original \mathbb{D} .

Si en el cas de la CDT habitual els triangles compleixen el criteri del circumcercle buit, en el aquest altre cas els triangles resultants compliran el que hem anomenat **criteri de la circumel·lipse buida**.

Definició 40 (Circumel·lipse d'un triangle). Donada una el·lipse E amb

matriu associada M_E i un triangle T , anomenarem circumel·lipse de T a l'el·lipse E' que té els mateixos eixos i relació d'aspecte dels eixos que l'el·lipse donada, E , i que passa pels tres vèrtexs del triangle. És a dir, E' és l'el·lipse amb matriu associada $M_{E'}$,

$$M_{E'} = rM_E \quad \text{amb } r \in \mathbb{R}^+$$

tal que passa pels tres vèrtexs x_1, x_2, x_3 de T .

$$x_i^T M_{E'} x_i = 1, \quad i \in \{1, 2, 3\}$$

Proposició 13. *Tot triangle d'una CDT obtinguda aplicant el mètode incremental adaptat per a distàncies el·líptiques és tal que la seva circumel·lipse no conté cap punt de la triangulació visible des d'ell.*

Aquesta proposició és trivial de demostrar si es té en compte que la funció f_E transforma un cercle de \mathbb{D} en l'el·lipse E a l'espai \mathbb{D}_E .

La figura A.9 mostra un exemple d'aplicació de l'algorisme sobre un graf compost per un polígon amb un forat i un conjunt de punts interiors. Podem observar que la triangulació resultant compleix la propietat de la circumel·lipse buida i, en conseqüència, les arestes tendeixen a alinear-se segons la direcció marcada per l'el·lipse E , tal com havíem demanat al principi.

Cas II: Cada punt del pla té assignada una el·lipse deformadora

En aquest cas, les el·lipses varien en funció del punt del pla; o, el què és equivalent, la funció f_E no és constant, sinó que depèn del punt del pla. La generalització que proposem aprofita el fet que l'algorisme treballi incrementalment per variar l'el·lipse a cada crida a les rutines d'inserció d'un punt i d'una aresta a la triangulació (AfegirPuntCDT i AfegirArestaCDT). Durant la inserció d'un nou punt a la triangulació, l'el·lipse que determina la deformació de l'espai serà l'associada al punt del pla afegit. Aquest és un altre punt que diferencia el nostre mètode del proposat en [Mav90], en el qual la deformació que s'aplica és un promig de les deformacions associades als punts que es veurien afectats per la inclusió del nou punt en cas que es tractés d'una CDT amb la distància habitual. En el nostre cas, aquesta elecció ens permetrà enunciar la proposició 14 sobre la triangulació resultant (enunciada més avall). Fem notar que, gràcies al fet que l'algorisme treballa incrementalment, la deformació determinada per una el·lipse cal aplicar-la tant sols a un subconjunt de punts reduït (els punts veïns al que s'ha inserit). L'algorisme modificat per inserir un punt en una CDT és el següent:


```

acció AfegirPuntCDTModif( $\mathcal{T} : CDT, p : Vèrtex$ )
  {Prec:  $p \notin \mathcal{T} \wedge p \in Interior(\mathcal{T})$ }
   $pila := PilaTrianglesBuida$ 
  Trobar el triangle  $t \in \mathcal{T}$  que conté  $p$ 
  Dividir  $t$  en tres triangles,  $t_1, t_2$  i  $t_3$ , segons  $p$ 
   $ell := EllipseDeformadora(p)$ 
  Push( $pila, t_1$ )
  Push( $pila, t_2$ )
  Push( $pila, t_3$ )
  mentre  $\neg EsBuida(pila)$  fer
     $t := Pop(pila)$ 
     $t_{opo} := TriangleOposat(t, p)$ 
    si l'aresta compartida per  $t$  i  $t_{opo}$  no és fixa i
       $p \in CircumEllipse(t_{opo}, ell)$  llavors
        Intercanviar l'aresta compartida per  $t$  i  $t_{opo}$ 
        Push( $pila, t$ )
        Push( $pila, t_{opo}$ )
      fsi
    fmentre
  facció

```

La funció `EllipseDeformadora` és la rutina que retorna l'el·lipse associada a un punt qualsevol del pla \mathbb{D} . Aquesta rutina la proporciona l'usuari de la CDT modificada.

Quan es tracta d'afegir una aresta de restricció a la triangulació, l'el·lipse deformadora varia durant el procés d'inserció, enlloc de ser una única. Fixem-nos que el criteri del circumcercler buit en aquest cas tant sols es fa servir durant la triangulació dels pseudopolígons inferior i superior (vegeu la secció A.3). Per tant, a la rutina `TriangularPseudopoligonCDT` n'hi haurà prou amb triar una el·lipse deformadora en cada pas de la recursió per triar el punt en què queda subdividit el pseudopolígon (punt c de l'algorisme); l'el·lipse escollida ha estat l'associada a un dels extrems del pseudopolígon. Si hem triat aquesta opció enlloc d'utilitzar una sola el·lipse deformadora durant tot el procés d'inserció d'una aresta a la triangulació és principalment per poder garantir la proposició 14: fixem-nos que els triangles formats a cada pas de la recursió són tals que la seva circumel·lipse (triant com a el·lipse l'associada a un dels seus extrems) és buida. Fem notar que el mètode proposat en [Mav90] no s'ocupa de les arestes de restricció (és a dir, treballa com si es tractés d'una DT no restringida) perquè, segons assegura l'autor, la mateixa naturalesa de les dades fa que la distribució dels punts sigui tal que no calgui preocupar-se'n (la triangulació sempre continuarà les arestes del contorn).

Atès que el procediment que insereix una aresta entre dos vèrtexs d'una CDT no realitza cap test geomètric relacionat amb el criteri del circumcercler

buit, roman inalterat (es pot fer servir el mateix que en el cas de la CDT no modificada). És a dir, que tant sols caldrà canviar la subrutina que retriangula l'interior d'un pseudopolígon aresta-visible:

```

acció TriangularPseudopolígonCDTModif( $P : \text{LlistaVèrtexs}$ ,
                                          $ab : \text{Aresta}, \mathcal{T} : \text{CDT}$ )

  si  $P$  té més d'un element llavors
     $c := \text{PrimerVèrtex}(P)$ 
     $ell := \text{EllipseDeformadora}(a)$ 
    per cada vèrtex  $v \in P$  fer
      si  $v \in \text{CircumEllipse}(a, b, c, ell)$  llavors
         $c := v$ 
      fsi
    fper
    Dividir  $P$  en  $P_E$  i  $P_D$ , on  $P = P_E + \{c\} + P_D$ 
    TriangularPseudopolígonCDT( $P_E, ac, \mathcal{T}$ )
    TriangularPseudopolígonCDT( $P_D, cb, \mathcal{T}$ )
  fsi
  si  $P$  no és buida llavors
    Afegir el triangle de vèrtexs  $a, b, c$  dins  $\mathcal{T}$ 
  fsi
facció

```

Remarquem que en aquest cas l'el·lipse deformadora triada hauria pogut ser l'associada a l'altre extrem de l'aresta, és a dir, la del punt b .

Seguint aquest procediment, ens assegurem que l'algorisme general que obté la triangulació segueix essent vàlid, ja que cadascun dels passos incrementals que afegixen una aresta o un punt a la triangulació funcionen correctament i convergeixen en un nombre finit de passos. Demostrar aquestes afirmacions no implica fer altre raonament que en el cas anterior, en què l'el·lipse és única per a cada punt.

Això no obstant, cal tenir en compte que, pel fet d'existir diferents el·lipses, la noció de distància el·líptica deixa de ser vàlida, i per tant no existeixi una sola triangulació que compleixi la condició del circumcercle buit – o, millor dit, de les circumel·lipses buides. Fins i tot, com en el cas de les distàncies no convexes, pot ocórrer que el diagrama de Voronoi contingui teselles limitades només per dues arestes, és a dir, que el seu dual, el graf de Delaunay, no sigui una triangulació [CD86]. Ara bé, fem notar que, malgrat tot, tal com acabem de veure el mètode incremental obté una triangulació vàlida; l'únic que ocorre és que la triangulació que s'obindrà pot depèn de l'ordre de la inserció dels punts.

Per la forma com hem descrit que treballa l'algorisme, podem afirmar que la triangulació obtinguda complirà la següent propietat, que hem anomenat **criteri de la circumel·lipse quasi-buida**:

Proposició 14. *Tot triangle d'una triangulació obtinguda aplicant el mètode incremental adaptat per al cas en què l'el·lipse depèn del punt del pla és tal que la circumel·lipse associada a algun dels seus tres vèrtexs no conté cap altre punt de la triangulació visible des d'ell i que s'hagi afegit abans que ell.*

Aquesta proposició resulta immediata de demostrar si es té en compte que, tal com hem dit, els algorismes que afegeixen nous punts i arestes de restricció a la triangulació treballen deformant l'espai respecte a l'el·lipse associada al punt afegit, o als extrems del pseudopolígon que s'està triangulant en el cas que s'estigui afegint una aresta a la triangulació.

La figura A.10 mostra un exemple en el qual s'ha aplicat l'algorisme al mateix graf que en el de la figura A.9, però tenint en compte l'el·lipse deformadora és diferent per a cada punt del pla $u \times v$. Tal com es pot veure, la relació d'aspecte de les el·lipses en cada punt és funció de la direcció horitzontal (u) del pla, i l'orientació dels eixos principals de les el·lipses és funció de la direcció vertical (v). Tal com era d'esperar, la triangulació que s'obté mostra clarament les propietats enunciades. Notem que la direccionalitat que prenen les arestes resultants concorda perfectament amb les direccions de les el·lipses.

La figura A.11 és un test simple per comprovar la propietat que hem anunciat referent a què la triangulació resultant en aplicar l'algorisme modificat per al cas II (cas en què cada punt del pla té assignada una el·lipse) depèn de l'ordre en què s'hagin inserit els punts. En aquest exemple, el graf està format per quinze punts, catorze d'ells situats en els vèrtexs d'un polígon regular i el quinze en el centre del polígon. En aquesta ocasió, atès que el graf original no té arestes de restricció que limitin un contorn, la triangulació resultant és la que recobreix l'envolupant convexa dels punts. Obtenir un algorisme que calculi la triangulació de Delaunay que recobreix l'envolupant convexa d'un graf basant-se en les dues rutines incrementals resulta trivial de fer: tant sols cal modificar l'esquema que ens ha servit per al càlcul d'una triangulació de Delaunay d'una cara de manera que, al final, enlloc de treure els triangles exteriors de la cara s'eliminin els que són exteriors a l'envolupant convexa, és a dir, aquells que tenen un dels vèrtexs del supertriangle.

Seguint amb l'exemple de la figura A.11, les el·lipses deformadores en aquest cas són iguals que les de l'exemple següent (vegeu la figura A.12 de baix a l'esquerra), és a dir, són el·lipses que apunten cap al centre de la figura i que són més allargassades com més allunyades del centre estan. Els punts en el primer cas (figura A.11a) han estat inserits començant pel punt del centre i acabant pel superior, mentre que en el segon cas (figura A.11b) l'ordre d'inserció és justament l'invers. Atès que el test geomètric de l'algorisme

d'inserció d'un punt en una CDT que comprova si un punt és interior a una circumel·lipse depén de l'el·lipse associada al punt inserit, l'algorisme es comporta de forma diferent en un o altre cas, és a dir, que realitzarà uns o altres intercanvis d'arestes. Tal com es pot comprovar en aquest exemple, la triangulació resultant en algunes ocasions pot ser molt diferent.

Aquest mateix exemple ens serveix també per il·lustrar la relació enuncuada entre la distribució dels punts del graf original i la variació de les el·lipses deformadores. Segons hem dit, i tal com també fan notar d'altres autors [Mav90, Pos93], si el graf conté una zona sense cap punt i en la qual la forma de les el·lipses deformadores canvia molt, serà impossible que les arestes (o, si es vol, els triangles) de la triangulació resultant reflexin aquesta gran variació, i s'orientin seguint la direccionalitat. Aquesta circumstància és justament la que ocorre en el cas de la figura A.11, atés que l'el·lipse en la qual les el·lipses associades a cadascun dels vèrtexs del graf és molt diferent.

La figura A.12 és un altre exemple d'aplicació de l'algorisme direccional sobre un graf amb un nombre considerable d'elements. En aquest cas, el graf consisteix en un polígon amb forma de trèbol de quatre fulles i un conjunt de punts interiors. Les el·lipses deformadores són les mateixes que en l'exemple anterior, i els punts interiors estan distribuïts de forma diferent en cadascuna de les quatre fulles del trèbol: en la fulla superior esquerra els punts formen una retícula quadrada; la superior dreta conté un nombre elevat de punts al seu interior, més o menys emplaçats de forma aleatòria; la inferior esquerra conté pocs punts; i en la inferior dreta hi ha un nombre bastant elevat de punts, agrupats en cercles.

Aquesta distribució heterogènia de punts interiors té com a efecte que l'algorisme de la CDT modificada es comporti de manera diferent en cadascuna de les quatre fulles. En la fulla superior dreta l'algorisme crea el mateix tipus de triangles que en el cas de la CDT no modificada (triangles rectangles), amb l'única diferència que les arestes en aquest cas apunten, com era d'esperar, cap al centre del trèbol. En la fulla superior dreta, com que hi ha un nombre molt elevat de punts l'algorisme pot crear els triangles que més li plagui, i per tant els resultats són els esperats en aquesta zona. Pel contrari, en no haver-hi gaires punts en la fulla inferior dreta, els triangles que crea l'algorisme depenen de l'ordre en que s'han inserit els punts a la triangulació. Finalment, en la fulla inferior esquerra només alguns dels triangles s'adapten a la direccionalitat marcada per les el·lipses, degut a la distribució irregular dels punts en aquesta regió.

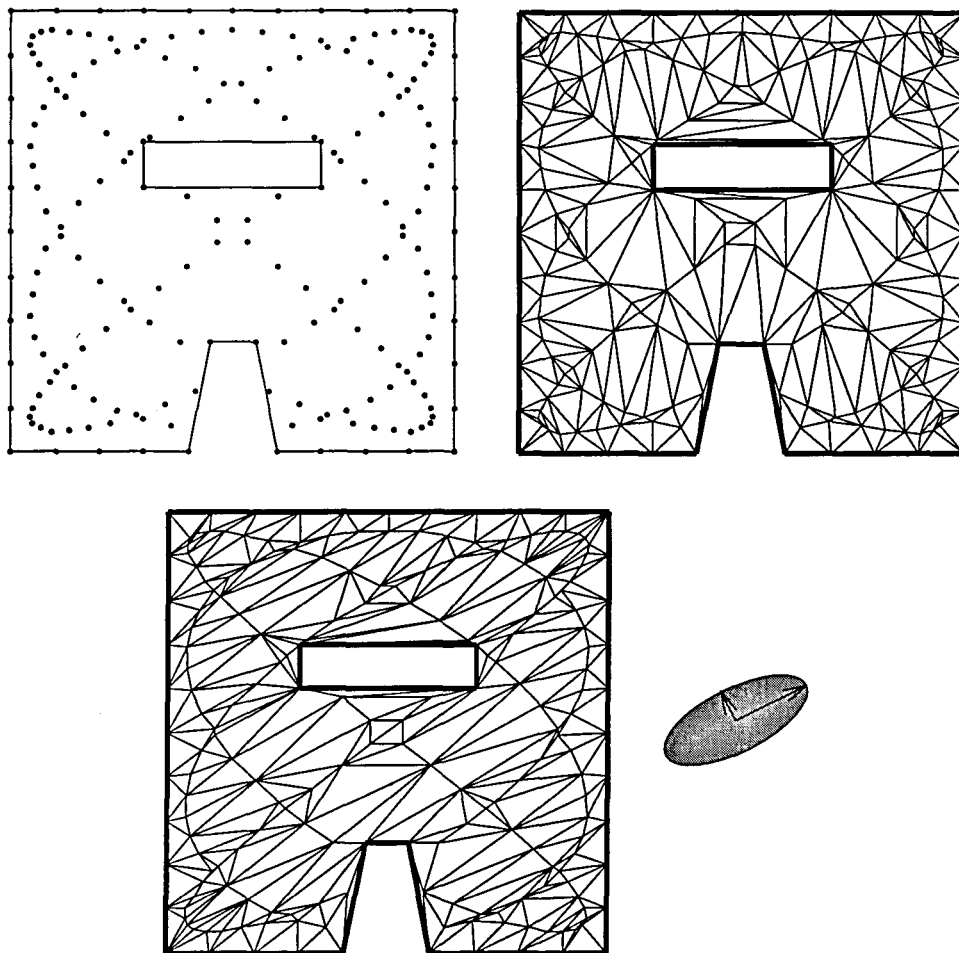


Figura A.9: Exemple d'aplicació de l'algorisme per al cas d'una sola el·lipse. La figura superior esquerra mostra el graf original, mentre que la superior esquerra és la CDT del graf segons la distància habitual. La figura inferior mostra la triangulació que s'aconsegueix tenint en compte la deformació induïda per l'el·lipse representada.

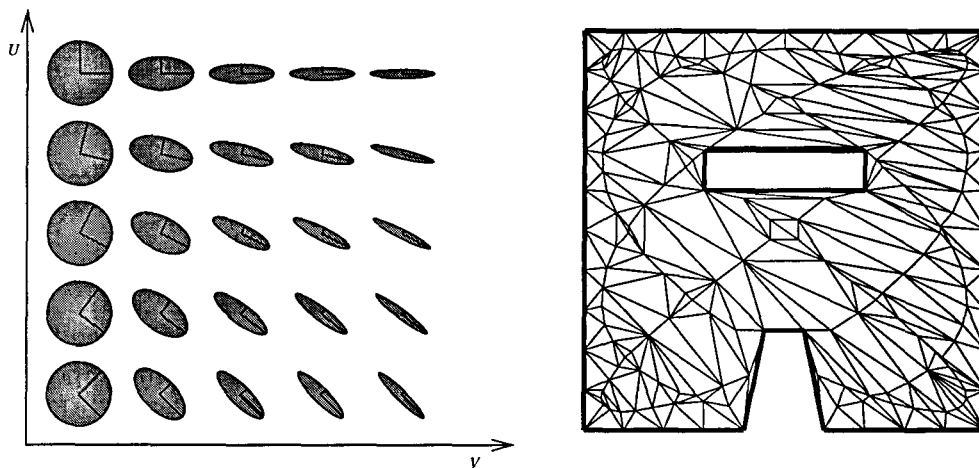


Figura A.10: Exemple d'aplicació de l'algorisme en el cas en què l'el·lipse deformadora depèn del punt del pla. La figura de l'esquerra representa diverses el·lipses en diferents punts del pla, mentre que la figura de l'esquerra mostra la triangulació resultant.

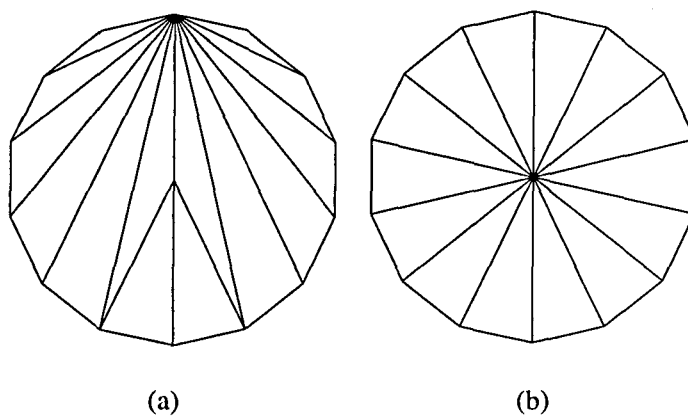


Figura A.11: Exemple senzill que mostra que la triangulació produïda per l'algorisme modificat en el cas II depèn de l'ordre d'inserció dels punts en la triangulació. Els punts en la triangulació de l'esquerra han estat introduïts en l'ordre invers que en la de la dreta, i el resultat és completament diferent malgrat que les el·lipses deformadores són les mateixes en ambdós casos.

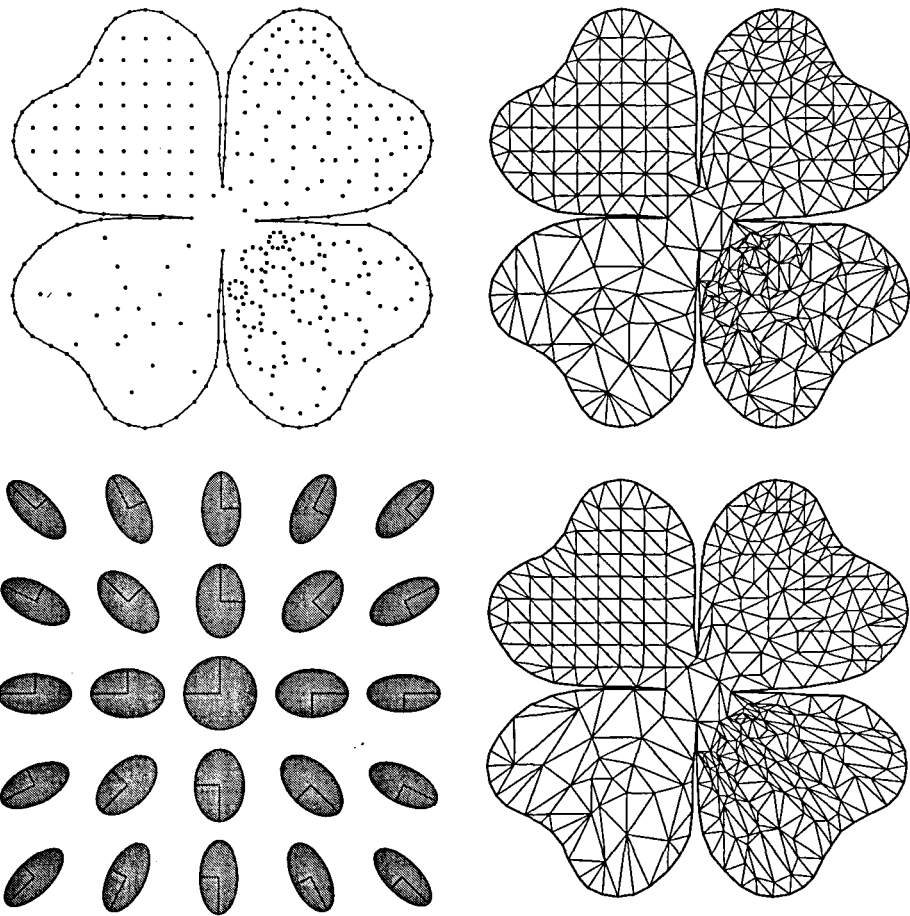


Figura A.12: Exemple d'aplicació de l'algorisme sobre un graf amb un nombre elevat d'entitats. A dalt, el graf original i la seva CDT habitual (no modificada). A baix, les el·lipses deformadores del pla evaluades en diferents punts del pla i la triangulació resultant.

Bibliografia

- [AES91] S.S. Abi-Ezzi and L.A. Shirman. Tessellation of curved surfaces under highly varying transformations. In F.H. Post and W. Barth, editors, *Proceeds. of EUROGRAPHICS'91*, pages 385–397. Elsevier Science, 1991.
- [AG87] E.L. Allgower and S. Gnutzmann. An algorithm for piecewise linear approximation of implicitly defined 2d surfaces. *SIAM J. Numer. Anal.*, 24:452–469, 1987.
- [AS87] E.L. Allgower and P.H. Schmidt. An algorithm for piecewise linear approximation of implicitly defined manifold. *SIAM J. Numer. Anal.*, 22:322–346, 1987.
- [AS96] M. Algorri and F. Schmitt. Mesh simplification. In *Proceeds. of EUROGRAPHICS'96*, pages 78–86, 1996.
- [Bak89] T.J. Baker. Developments and trends in three-dimensional mesh generation. *Applied Numerical Mathematics*, 5:275–304, 1989.
- [BBX95] C.L. Bajaj, F. Bernardini, and G. Xu. Automatic reconstruction of surfaces and scalar fields from 3D scans. In *Computer Graphics Proceeds. (SIGGRAPH)*, pages 109–118, 1995.
- [Blo88] J. Bloomenthal. Polygonalization of implicit surfaces. *Computer Aided Geometric Design*, 5:341–355, 1988.
- [Boi85] J.D. Boissonnat. Surface reconstruction from planar cross section. *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 393–397, 1985.
- [Bow81] A. Bowyer. Computing Dirichlet tessellations. *The Computer Journal*, 24(2):162–166, 1981.
- [BR94] C.L. Bajaj and A. Royappa. Triangulation and display of rational parametric surfaces. In *IEEE Visualization Conference'94*, pages 69–76. IEEE Compu. Soc. Press, 1994.

- [Bro91] J.L. Brown. Vertex based data dependent triangulations. *Computer Aided Geometric Design*, 8:239–251, 1991.
- [Bru80] I. Brueckner. Construction of Bézier points quadrilaterals from those of triangles. *Computer Aided Design*, 12(1):21–24, 1980.
- [BS89] A.D. Brown and E.W. Stockley. Relaxation methods in CAGD. *Computer Aided Design*, 21(5):303–308, 1989.
- [BV92] P. Brunet and A. Vinacua. Modeling closed surfaces: a comparison of existing methods. In Tom Lynche and Larry L. Schumaker, editors, *Mathematical Methods in CAGD II*, pages 29–42. Academic Press, 1992.
- [BV95] P. Brunet and M. Vigo. Piecewise linear approximation of trimmed surfaces. In G. Farin H. Hagen and H. Noltemeier, editors, *Geometric Modelling*, Computing Suppl. 10, pages 341–356. Springer Verlag, 1995.
- [BW76] K. Bathe and E. Wilson. *Numerical methods in finite element analysis*. Prentice-Hall, N.J., 1976.
- [BW92] J. Bonet and R.D. Wood. Mesh enrichment procedures for the finite element analysis of thin sheet forming processes. In *Numerical Methods in Industrial Forming Processes*, pages 221–227, 1992.
- [Cas87] M. Casale. Freeform solid modeling with trimmed patches. *IEEE Comput. Graphics and Applications*, pages 33–43, 1987.
- [CC78] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10(6):350–355, 1978.
- [CD86] L.P. Chew and R.L. Drysdale. Voronoi diagrams based on convex distance functions. Technical Report Tech Report PCS-TR86-132, Dartmouth College, Dept. of Computer Science, nov 1986.
- [Che89a] L.P. Chew. Constrained Delaunay triangulations. *Algorithmica*, 4:97–108, 1989.
- [Che89b] L.P. Chew. Guaranteed-quality triangular meshes. Technical Report Tech. Report TR 89-983, Cornell University, Dept. of Comput. Science, 1989.
- [Che93] L.P. Chew. Guaranteed quality mesh generation for curved surfaces. In *Proceedings of the ACM Symposium on Computational Geometry*, pages 274–280, 1993.

- [Cra85] M. Crampin et al. Linear approximation of curves with bounded curvature and data reduction algorithm. *Computer Aided Design*, 17(6):257–261, 1985.
- [Cui98] J.C. Cuillère. An adaptive method for the automatic triangulation of 3D parametric surfaces. *CAD*, 30(2):139–149, 1998.
- [CW95] M.F. Cohen and J.R. Wallace. *Radiosity and realistic Image Synthesis*, chapter 8 (Meshing). Academic Press Inc., 1995.
- [dF89] L. de Floriani. A pyramidal data structure for triangle-based surface description. *IEEE Computer Graphics and Applications*, pages 67–78, March 1989.
- [DF96] G. Drettakis and E. Fiume. Structured penumbral irradiance computation. *IEEE Trans. on Vis. and Comput. Graphics*, 2(4):299–312, Dec. 1996.
- [dFFP85] L. de Floriani, B. Falcideno, and C. Pienovi. Delaunay-based representations of surfaces defined over arbitrarily shaped domains. *Computer Vision, Graphics and Image Processing*, 32:127–140, 1985.
- [dFP92] L. de Floriani and A. Puppo. An on-line algorithm for constrained Delaunay triangulation. *Computer Vision, Graphics and Image Processing*, 54(3):290–300, July 1992.
- [DM90] Dolenc and Makela. Optimized triangulation of parametric surfaces. *Mathematics of Surfaces IV*, 1990.
- [DM92] T. DeRose and S. Mann. . In T. Lyche and L. L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design II*. Academic Press, 1992.
- [DS89] E.F. D’Azzevedo and R.B. Simpson. On optimal interpolation triangle incidences. *SIAM J. Sci. Stat. Comput.*, 10(6):1063–1076, 1989.
- [DSB92] T.K. Dey, K. Sugihara, and C. Bajaj. Delaunay triangulations in three dimensions with the finite precision arithmetic. *CAGD*, 9:457–470, 1992.
- [DZ91] M.J. DeHaemmer and M.J. Zyda. Simplification of objects rendered by polygonal approximations. *Comput. & Graphics*, 15(2):175–184, 1991.
- [ET85] H. ElGindy and G.T. Toussaint. Efficient algorithms for intersecting and deleting edges from triangulations. In *Proceedings*

of the International Conferences on Foundations of Data Organizations, May 1985.

- [ET89] H. ElGindy and G.T. Toussaint. On geodesic properties of polygons relevant to linear time triangulation. *The Visual Computer*, 5:68–74, 1989.
- [Far90] G. Farin. *Curves and Surfaces for computer aided geometric design*. Academic Press, 1990.
- [Far94] R.T. Farouki. Watch your (parametric) speed! In Adrian Bowyer, editor, *The Mathematics of Surfaces IV*. Clarendon Press. Oxford, 1994.
- [FF91] W. Frey and D. Field. Mesh relaxation: a new technique for improving triangulations. *Int. Journal for Numerical Methods in Engineering*, 31:1121–1133, 1991.
- [Fie89] D.A. Field. Laplacian smoothing and Delaunay triangulations. *Commun. appl. numer. methods*, 4:709–712, 1989.
- [FMM86] D. Filip, R. Magedson, and R. Markot. Surface algorithm using bounds on derivatives. *Computer Aided Geometric Design*, 3:295–311, 1986.
- [FR87] R.T. Farouki and V.T. Rajan. On the numerical condition of polynomials in Berstein form. *Computer Aided Geometric Design*, 4:191–216, 1987.
- [FR88] R.T. Farouki and V.T. Rajan. Algorithms for polynomials in Bernstein form. *Computer Aided Geometric Design*, 5:1–26, 1988.
- [Geo89] P.L. George. Mailleur automatique en tetraèdres. *Bulletin de liaison de la recherche en informatique et en automatique*, 125, 1989.
- [Gou94] R.J. Goult. Standards for curves and surface data exchange. In Adrian Bowyer, editor, *The Mathematics of Surfaces IV*. Clarendon Press. Oxford, 1994.
- [GS78] P.J. Green and R. Sibson. Computing Dirichelet tessellations in the plane. *The Computer Journal*, 21:168–175, 1978.
- [GS85] L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, 4(2):74–123, 1985.

- [Ham94] B. Hamann. A data reduction scheme for triangulated surfaces. *Computer Aided Geometric Design*, 11:197–214, 1994.
- [HB87] B. Von Herzen and A. Barr. Accurate triangulations of deformed, intersecting surfaces. *Computer Graphics*, 21(4):103–110, 1987.
- [HDD⁺93] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. *Computer Graphics*, pages 19–26, August 1993.
- [HL88] K. Ho-Le. Finite element mesh generation methods: a review and classification. *Computer Aided Design*, 20(1):27–38, 1988.
- [HO77] E. Hinton and CD.R.J. Owen. *Finite element programming*. Academic Press, London, 1977.
- [HW90] M. Hall and J. Warren. Adaptative polygonalization of implicitly defined surfaces. *IEEE Computer Graphics and Applications*, 10(6):33–42, November 1990.
- [IGE90] IGES/PDES Organization, U.S. Dept. of Commerce. *The Initial Graphic Exchange Specification (IGES) Version 5.0*, 1990.
- [Joe91] B. Joe. Construction of three-dimensional Delaunay triangulations using local transformations. *Computer Aided Geometric Design*, 8:123–142, 1991.
- [JSSB93] P. Jacobs, L. Sadler, P. Stucki, and J. Bresenham. Real virtuality: Stereolithography - rapid prototyping in 3d computer graphics. In *Computer Graphics Proceedings, Annual Conference Series (SIGGRAPH)*, pages 377–378, 1993.
- [JT80] C.L. Jackins and S.L. Tanimoto. Octrees and their use in representing three-dimensional objects. *Comput. Graphics and Image Proc.*, 14:249–270, 1980.
- [KL96] V. Krishnamurthy and M. Levoy. Fitting smooth surfaces to dense polygonal meshes. *Computer Graphics Proceeds. (SIGGRAPH)*, pages 313–324, 1996.
- [KLN89] M. Karasick, D. Lieber, and L.R. Nackman. Efficient Delaunay triangulation using rational arithmetic. Technical report, IBM, 1989.
- [KM95] S. Kumar and D. Manocha. Efficient rendering of trimmed NURBS surfaces. *Computer Aided Design*, 27(7):509–521, 1995.

- [Kos91] M. Kusters. Curvature dependent parameterization of curves and surfaces. *Computer Aided Design*, 23(8):569–578, October 1991.
- [KS95] R. Klein and W. Straßer. Large mesh generation from boundary models with parametric face representation. In *Proceedings of Solid Modeling'95*, pages 431–440, 1995.
- [KTS85] P. Koistinen, M. Tamminen, and H. Samet. Viewing solid models by bintree conversion. In *Proceeds. of EUROGRAP-HICS'85*, pages 147–157. North-Holland Publ. Co., 1985.
- [Law77] C.L. Lawson. Software for C1 surface interpolation. In J.R. Rice, editor, *Mathematical Software III*, pages 161–194. Academic Press, 1977.
- [LC87] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM Computer Graphics*, 21(4):163–169, July 1987.
- [LL86] D. T. Lee and H. K. Lin. Generalized Delaunay triangulation for planar graphs. *Discrete Computational Geometry*, 1:201–217, 1986.
- [Man88] M. Mantyla. *An introduction to solid modeling*. Computer Science Press, 1988.
- [Mav90] D. Mavriplis. Adaptative mesh generation for viscous flows using Delaunay triangulation. *Journal of Computational Physics*, pages 271–291, 1990.
- [Mea82] D. Meagher. Geometric modeling using octree encoding. *Comput. Graphics and Image Proc.*, 19:129–147, 1982.
- [Mel92] E. Melissaratos. Optimal size finite element meshes without obtuse and small angles. Technical Report RUU-CS-92-39, Utrecht Univ., Dept. of Computer Science, December 1992.
- [NF89] G.M. Nielson and T.A. Foley. A survey of applications of an affine invariant norm. In Tom Lyche and Larry L. Schumaker, editors, *Mathematical Methods in CAGD*, pages 445–467, 1989.
- [Nim92] Uwe M. Nimscheck. Adaptive mesh generation for radiosity methods. Technical report, U of North Carolina, 1992.
- [NTK92] Nakajima, S. Tokumasu, and Y. Kunitomo. Feature-based heuristics for finite-element meshing using quadtrees and octrees. *Computer Aided Design*, 24(2):677–690, December 1992.

- [OBS92] A. Okabe, B. Boots, and K. Sugihara. *Spatial Tessellations. Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons Ltd, 1992.
- [Omo89] S. Omohundro. The Delaunay triangulation and function learning. Technical Report TR-90-001, ICSI, 1989.
- [Per89] A. Perronnet. Quelques methodes actuelles pour generer un maillage non structure d'un objet 3d. *Bulletin de liason de la recherche en informatique et en automatique*, 125, 1989.
- [Pet89] Peters. Smooth interpolation of a mesh of curves. *Constructive Approximation*, 7:221–247, 1989.
- [Pir89] O. Pironneau. Triangulations automatiques. *Bulletin de liason de la recherche en informatique et en automatique*, 125, 1989.
- [Pos93] M.K. Posenau. Approaches to high aspect ratio triangulations. In *Proceeds of 5th canadian Conference on Comp. Geom.*, Waterloo, Ontario, 1993.
- [Pow92a] P.L. Powar. Minimal roughness property of the Delaunay triangulation: a shorter approach. *Computer Aided Geometric Design*, 9:491–494, 1992.
- [Pow92b] P.L. Powar. The neutral case for the min-max angle criterion: a generalized concept. *CAGD*, 9:413–418, 1992.
- [PR95] L.A. Piegl and A.M. Richard. Tessellating trimmed NURBS surfaces. *CAD*, 27(1):16–26, 1995.
- [Pro89] Project 322 ESPRIT research report. *CAD Data Transfer for solid models*, 1989. Springer-Verlag.
- [PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry*. Springer-Verlag, 1985.
- [PSK89] R. Perucchio, M. Saxena, and A. Kela. Automatic mesh generation from solid models based on recursive spatial decomposition. *Int. Journal for Numerical Methods in Engineering*, 28:2469–2501, 1989.
- [PT98] L.A. Piegl and W. Tillert. Geometry-based triangulation of trimmed NURBS surfaces. *CAD*, 30(1):11–18, 1998.
- [QS91] E. Quack and L.L. Schumaker. Least squares fitting by linear splines on data-dependent triangulations. In P.J. Laurent, A. LeMehaute, and L.L. Schumaker, editors, *Curves and surfaces*, pages 387–390. Academic Press, New 1991.

- [Req80] A. Requicha. Representations of rigid solids: Theory, methods and systems. *Computing surveys of the ACM*, 12:437–464, 1980.
- [RHD89] A. Rockwood, K. Heaton, and T. Davis. Real-time rendering of trimmed surfaces. *Computer Graphics*, 23(3):107–116, 1989.
- [Rip90] S. Rippa. Minimal roughness property of the Delaunay triangulation. *Computer Aided Geometric Design*, 7(6):489–497, 1990.
- [Rip93] S. Rippa. Transforming triangulations in polygonal domains. *Computer Aided Geometric Design*, 10:531–536, 1993.
- [Rog85] D. F. Rogers. *Procedural Elements for Computer Graphics*. McGraw-Hill, 1985.
- [Rot82] S.D. Roth. Ray casting for modeling solids. *Comput. Graphics and Image Proc.*, 18(2):109–144, 1982.
- [Rup92] J. Ruppert. A new and simple algorithm for 2-dimensional mesh generation. Technical report, Computer Science Division, Univ. of California at Berkeley, 1992.
- [RV83] A.A.G. Requicha and H.B. Voelcker. Boolean operations in solid modelling: boundary evaluation and merging algorithms. *Proc. IEEE*, 3(7):30–40, Oct. 1983.
- [SA94] J. Samareh-Abolhassanni. Triangulation of NURBS surfaces. In *Proceeds. of the 4th Int. Conf. on Numerical Grid Generation*, pages 377–388, Swansea, Wales (UK), April 1994.
- [Sch93] L.L. Schumaker. Triangulations in CAGD. *IEEE Computer Graphics and Applications*, pages 47–52, January 1993.
- [Sed85] T. W. Sederberg. Piecewise algebraic surface patches. *Computer Aided Geometric Design*, 2:53–59, 1985.
- [Sed90] T. W. Sederberg. Techniques for cubic algebraic surfaces. *IEEE Computer Graphics and Applications*, pages 12–21, 1990.
- [Sei91] R. Seidel. A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Computational Geometry: Theory and Applications*, 1:51–64, 1991.
- [SFBHD86] Schmitt, Francis, Barsky, and Wen Hui-Du. An adaptative subdivision method for surface-fitting from sampled data. *Computer Graphics*, 20(4):179–188, 1986.

- [SG92] K. Shimada and D.C. Gossard. Computational methods for physically-based FE mesh generation. In G.J. Olling and F. Kimura, editors, *Human aspects in computer generated manufacturing*, pages 411–420, 1992.
- [SG95] K. Shimada and D.C. Gossard. Bubble mesh: Automated triangular meshing of non-manifold geometry by sphere packing. In *Proceeds of Solid Modeling'95*, pages 409–419, 1995.
- [SG98] K. Shimada and D.C. Gossard. Automatic triangular mesh generation of trimmed parametric surfaces for finite element analysis. *CAGD*, 15:199–222, 1998.
- [SH92] X. Sheng and B.E. Hirsh. Triangulation of trimmed surfaces in parametric space. *Computer Aided Design*, 24(8):437–444, August 1992.
- [She91] T. Shermer. Computing bushy and thin triangulations. *Computational Geometry: Theory and Applications*, 1:115–125, 1991.
- [Shi81] S.N. Shihari. Representation of three-dimensional images. *ACM Computing Surveys*, 13(4):399–423, 1981.
- [Slo87] S.W. Sloan. A fast algorithm for constructing Delaunay triangulations in the plane. *Adv. Eng. Software*, 9(1):34–55, 1987.
- [SP89] N. Sapidis and R. Peruccio. Advanced techniques for automatic finite element meshing from solid models. *Computer Aided Design*, 21(4):248–253, May 1989.
- [SP91] N. Sapidis and R. Peruccio. Delaunay triangulation of arbitrarily shaped planar domains. *Computer Aided Geometric Design*, 8:421–437, 1991.
- [SR94] F. Schröder and P. Rossbach. Managing the complexity of digital terrain models. *Computer & Graphics*, 18(6):775–783, 1994.
- [ST85] H. Samet and M. Tamminen. Bintrees, CSG trees and time. *Computer Graphics (SIGGRAPH'85)*, 19(3):121–130, 1985.
- [SZL92] W.J. Schroeder, J.A. Zarge, and W.E. Lorensen. Decimation of triangle meshes. *Computer Graphics*, 26(2):65–70, 1992.
- [Til80] R.B. Tilove. Set membership classification: a unified approach to geometric intersection problems. *IEEE Trans. on Computers*, C-29(10):847–883, 1980.

- [TN87] W.C. Thibault and B.F. Naylor. Set operations on polyhedra using binary space partitioning trees. In *SIGGRAPH 87*, pages 153–162, 1987.
- [Tur91] G. Turk. Generating textures on arbitrary surfaces using reaction-diffusion. *Computer Graphics*, 25(4):289–298, July 1991.
- [Tur92] G. Turk. Re-tiling polygonal surfaces. *Computer Graphics*, 26(2):55–64, July 1992.
- [VDA87] VDA Working Group CAD/CAM. *Vda Surface Data Interface (VDAFS) Version 2.0*, 1987.
- [Vel90] L. Velho. Adaptive polygonalization of implicit surfaces using simplicial decomposition and boundary constraints. In *Euro-Graphics'90*, pages 125–136, 1990.
- [Vig92] M. Vigo. Compactació d'octrees estesos. Technical Report LSI-92-2-T, Universitat Politècnica de Catalunya, Dept. de Llenguatges i Sistemes Informàtics, 1992.
- [Vig95] M. Vigo. An incremental algorithm based on edge swapping for constructing restricted Delaunay triangulations. Technical Report LSI-95-43-R, Universitat Politècnica de Catalunya, Dept. Llenguatges i Sistemes Informàtics, 1995.
- [Vig97] M. Vigo. An improved incremental algorithm for constructing restricted Delaunay triangulations. *Comput. & Graphics*, 21(2):215–223, 1997.
- [Vig98] M. Vigo. Directional constrained Delaunay triangulation. Technical Report LSI-98-29-R, Universitat Politècnica de Catalunya, Dept. Llenguatges i Sistemes Informàtics, 1998.
- [VPB95] M. Vigo, N. Pla, and P. Brunet. From degenerate patches to triangular and trimmed patches. Technical Report LSI-95-20-R, Universitat Politècnica de Catalunya, Dept. de Llenguatges i Sistemes Informàtics, 1995.
- [VPB97] M. Vigo, N. Pla, and P. Brunet. From degenerate patches to triangular and trimmed patches. In A. Le Mehaute and A.L. Allgower, editors, *Curves and Surfaces*, 1997.
- [VPB98] M. Vigo, N. Pla, and P. Brunet. Directional adaptive surface triangulation. *Computer Aided Geometric Design*, 1998. To be published.

- [Wil90] R. Wildman. An efficient algorithm for the triangulation of surfaces in R^3 . Preprint, Colorado State University, 1990.
- [WW88] M.A. Watkins and J. Worsey. Degree reduction of Bézier curves. *Computer Aided Design*, 20(3):398–405, September 1988.
- [WW94] W. Welch and A. Witkin. Free-form shape design using triangulated surfaces. *Computer Graphics*, pages 247–256, 1994.

