

UNIVERSITAT POLITÈCNICA DE CATALUNYA
DEPARTAMENT DE LENGUATGES I SISTEMES INFORMÀTICS
MÀSTER EN COMPUTACIÓ

TESI DE MÀSTER

ENCAIX DE MOLÈCULES
EN UN SISTEMA
DE REALITAT VIRTUAL

ESTUDIANT: CARLES TORTA FERRÉ
DIRECTOR: PERE PAU VÁZQUEZ ALCOCER
ISABEL NAVAZO ALVARO

DATA: 7 DE SETEMBRE DE 2007

Índex

1	Introducció	5
2	L'estat de l'art	7
3	L'algoritme de "Soft Docking"	11
3.1	Càlcul de la graella	12
3.2	Algoritme d'encaix	15
3.3	Millores i detalls de l'algoritme	17
3.4	Resultats	19
4	L'aplicació en detall	21
4.1	Funcionalitats	21
4.1.1	Afegir/Obrir una molècula	21
4.1.2	Eliminar una molècula	22
4.1.3	Seleccionar una molècula	22
4.1.4	Moure la molècula	22
4.1.5	Docking automàtic	22
4.1.6	Unir molècules	23
4.2	Implementació de l'aplicació	23
4.2.1	Mòdul Àtom	23
4.2.2	Mòdul Enllaç	23
4.2.3	Mòdul Graella Intersecció	24
4.2.4	Mòdul Molècula	24
4.2.5	Mòdul Model	25
4.2.6	Interfície gràfica	25
5	Sistema de Realitat Virtual	27
5.1	Sistema de Realitat Virtual	27
5.2	Implementació de l'aplicació	28

Capítol 1

Introducció

Les aplicacions en el camp de la bioinformàtica cada dia són més utilitzades pels científics, sobretot les aplicacions que tenen un caràcter més visual i gràfic. Aquestes eines els ajuden en la seva investigació i els faciliten el seu treball, ja que amb eines visuals poden representar i manipular allò que no és visible. Dintre d'aquestes aplicacions o eines podem trobar les que s'anomenen *Soft Docking*. Aquestes eines tenen com a objectiu estudiar i determinar com interaccionen dos molècules, de les quals s'ha obtingut la seva estructura 3D.

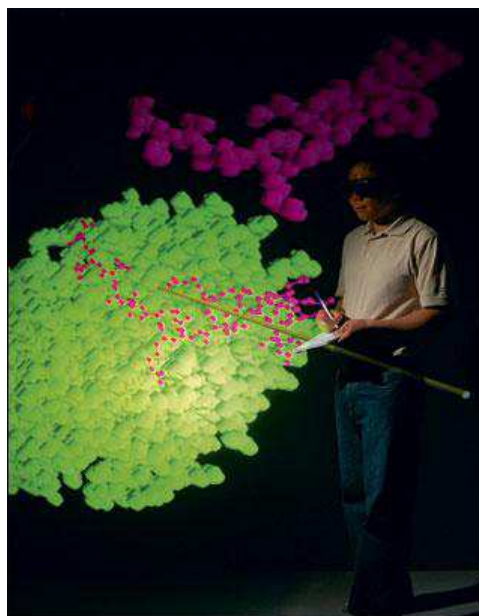


Figura 1.1: Científic de la Universitat de Calgary (Canadà) utilitzant una eina bioinformàtica en un sistema CAVE

Aquesta tesi es centra en aquest tipus d'eines. En les següents pàgines, es descriu un mètode de *Soft Docking* que permet determinar si dues molècules es troben en una localització correcta per tal que puguin interaccionar entre elles. El mètode està basat en les característiques geomètriques de les molècules prenent aquestes com cossos rígids. L'encaix molecular geomètric es considera una condició fonamental per la interacció molecular. No obstant, aquest encaix és una aproximació ja que s'han de tenir en compte altres anàlisis com els energètics.

L'objectiu d'aquesta tesi és la implementació d'una aplicació semiautomàtica que utilitzi aquest mètode com eina de manipulació molecular. A més d'utilitzar-se sobre un PC qualsevol, aquesta aplicació es podrà executar en un entorn virtual per tal d'ajudar a la comprensió del resultat.

Aquest document està estructurat de la forma següent: en el capítol 2 s'introdueix en el camp dels mètodes de *Soft Docking* i es descriuran algunes exemples, el capítol 3 detalla el mètode nou i utilitzat en l'aplicació creada; el capítol 4 descriu les funcionalitats de l'aplicació i la implementació realitzada per executar-se sobre un PC; finalment el capítol 5 mostra com a partir de l'aplicació del capítol anterior s'integra en un sistema de realitat virtual.

Capítol 2

L'estat de l'art

Com ja s'ha dit en la introducció, són molts els mètodes de *Soft Docking* investigats fins al moment i implementats en eines molt utilitzades pels científics [2]. En l'estudi previ per aquesta tesi s'ha prestat més atenció a aquells mètodes que tenen un caràcter més geomètric en la seva resolució. En aquest capítol es descriuran 3 mètodes amb aquestes característiques els quals utilitzen estratègies diferents.

El primer mètode [8] està implementat en l'eina GRAMM mostrant possibles solucions de docking. Realitza el càlcul mitjançant la discretització de les estructures tridimensionals de les macromolècules en cel·les cúbiques. En cada cel·la s'assigna el valor corresponent per distingir les cel·les interiors de les situades en la superfície molecular.

Després, el programa explora tot l'espai tridimensional de solucions possibles entre dos molècules, utilitzant passos rotacionals d'una estructura respecte una altra. Mitjançant transformades de Fourier es calcula el grau de solapament i complementarietat de les molècules per cada solució anterior (figura 2.1).

El següent mètode [6] realitza un procediment també només basat en criteris geomètrics. L'algoritme desenvolupat projecta la superfície molecular en una matriu bidimensional. El càlcul de regions de complementarietat es realitza detectant submatrius respecte la primera. Es realitza un exhaustiu recorregut en l'espai de rotacions en ordre per analitzar totes les possibles orientacions relatives entre les dues molècules. Finalment es seleccionen les orientacions que presenten un resultat més satisfactori.

L'últim mètode es basa en la complementarietat de polígons ESCHER [1]. Aquesta eina realitza el càlcul en tres mètodes diferents, cadascun encarregat d'avaluar una característica diferent. Ens centrarem en la primera part, la qual analitza la forma de les molècules. Avalua la complementarietat en totes les possibles orientacions, i proposa un nombre de solucions identificats

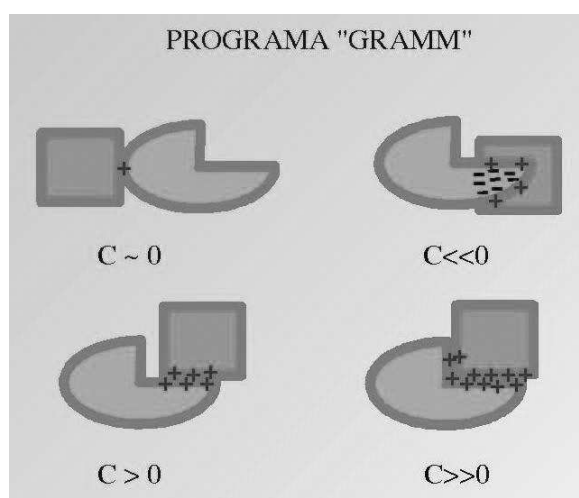


Figura 2.1: Càlcul del grau de complementarietat (C)

per una translació i una rotació. Aquestes transformacions s'apliquen a una molècula anomenada *probe* respecte una altra anomenada *target*.

L'estructura de cada molècula es talla en llesques paral·leles, on en cada llesca es construeix una poligonal amb cares d'igual longitud la qual representa el contorn de la molècula (veure figura 2.2). Les poligonals de cadascuna de les molècules són comparades entre elles i s'avaluen les regions de complementarietat.

Una cerca completa en l'espai de rotacions es realitza aplicant rotacions a la molècula *probe* en totes les possibles orientacions respecte la molècula *target*.

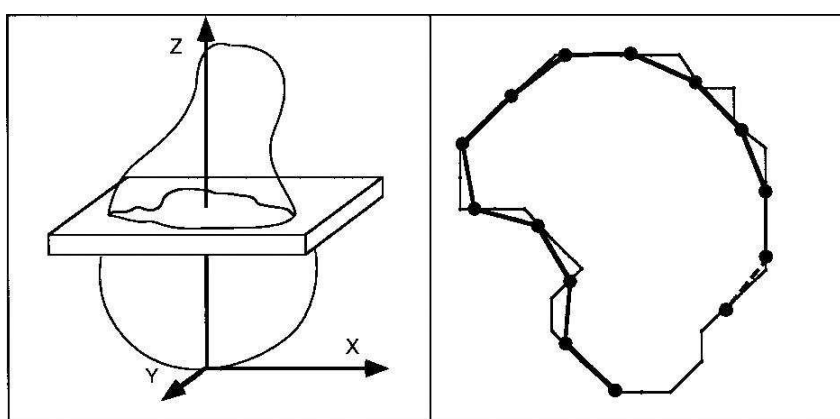


Figura 2.2: Construcció de la poligonal (dreta) a partir d'una llesca (esquerra)

En la bibliografia actual, hi ha molts altres mètodes com per exemple [7], [9], [5], [3], [4] entre d'altres que realitzen mètodes semblants als anteriors, cadascun amb els seus avantatges i inconvenients.

Una vegada estudiats alguns mètodes, podem extraure'n idees com per exemple la discretització de la molècula o l'espai de rotacions per trobar la posició final.

Un aspecte que podem variar és la introducció de la participació de l'usuari. Ens els mètodes anteriors es situa la molècula en qualsevol lloc de l'altra molècula. Potser no interessa cercar en tot l'espai i ens podem centrar en una zona específica. Per tant, en el mètode del capítol següent, l'usuari tindrà un caràcter col·laboratiu i decidirà en quina zona vol situar la molècula.

Capítol 3

L'algoritme de "Soft Docking"

El mètode usat està inspirat amb l'algoritme de l'article [8]. S'han extret idees, però a diferència d'aquest, en l'algoritme següent la participació de l'usuari és més evident, ja que decidirà la zona on es vol realitzar l'encaix molecular.

Per tal que aquest mètode i el resultat obtingut siguin correctes, ha de complir principalment dos requisits:

- La distància existent entre els àtoms d'una molècula i una altra no potser inferior a 3 unitats d'ångström.
- La posició que ocupa una molècula respecte una altra serà millor si la distància existent entre les molècules és inferior, és a dir, si es troben més pròximes, sempre tenint en compte la restricció anterior.

Aquest capítol s'estructura de la següent forma: primer es detallarà la primera part del mètode per tal d'obtenir la descripció geomètrica de la molècula segons les seves coordenades atòmiques; amb aquesta informació, passarem a la segona part del mètode, l'algoritme de *Soft Docking*, tenint en compte els requisits d'aquest enumerats anteriorment. Per últim, exposarem algunes millores i detalls que es poden aplicar, així com els resultats obtinguts.

En aquest algoritme, es tractarà una molècula com una estructura tridimensional, on els àtoms que formen aquesta es representaran per esferes situades a les coordenades de l'àtom i de radi corresponent a aquest.

Prèviament, definim les següents variables i estructures:

- Definim $a_{x,y,z}$ com un àtom de coordenades (x, y, z) en un espai tridimensional.
- Designarem amb m la molècula a tractar, on m és una llista d'àtoms a .

- A la distància mínima entre àtoms entre dues molècules, la definim com D (per defecte valdrà 3).

3.1 Càlcul de la graella

En aquesta primera part, obtenim la informació geomètrica de la superfície i l'emmagatzemem en una graella. Aquesta informació serà utilitzada en la segona part de l'algoritme, per tal de facilitar el càlcul d'encaix molecular.

La graella a calcular té com a objectiu saber si un punt qualsevol es troba a l'exterior o a l'interior de la molècula. Ens referim a l'exterior d'una molècula com l'espai que es troba situat a la zona on és permès situar una molècula per realitzar l'encaix; i a l'interior com l'espai restant. A més, al parlar de superfície molecular ens referirem als punts que delimiten la zona interna de l'externa, és a dir, la superfície molecular real, però *expandida* una distància D (si D pren valor zero, la superfície molecular seria la real i la zona interior la que es troba dins la representació esfèrica dels àtoms). Aquesta geometria té com a propietat el següent fet: per passar d'un punt intern a un altre d'extern, en algun moment passarem per un punt que es troba en la superfície molecular. És a dir, no hi ha forats en la superfície molecular. En la figura 3.1 es representa la zona interna, externa i la superfície d'una molècula.

Per obtenir les característiques geomètriques, podem discretitzar el volum que ocupa la molècula m , projectant-la dins una graella g tridimensional de costats variables. Aquesta graella és una estructura on cada punt $g_{i,j,k}$ conté la distància mínima a la superfície molecular, on i , j i k són els índexs a la graella 3D. A més, conté la posició mínima de la graella p_{min} i les resolucions de cada costat. Amb aquests dos atributs, podem obtenir els índexs i , j i k d'un punt qualsevol i viceversa. Em de tenir en compte que si un punt no està situat a l'interior de la graella els índexs corresponents no seran vàlids.

A partir de la caixa contenidora de la molècula podem calcular les dimensions de la graella, on prèviament s'haurà definit un paràmetre R que ens proporcionarà la resolució d'aquesta. Per tant, obtindrem una graella de dimensions $N \times M \times O$. Com més gran sigui el valor de R cometrem un error més gran en l'extracció de la superfície molecular. D'altra banda, el temps de càlcul de la graella es veurà reduït.

El càlcul de cadascuna de les distàncies dels punts, comporta executar un algoritme d'ordre cúbic. És a dir, per cada punt cal calcular la distància mínima i, per tant, recórrer cada vegada les posicions dels àtoms de la molècula. Per reduir aquest cost, intentem *visitar* cada punt de la graella un sol cop. D'aquesta manera reduïm el cost d'execució final.

Per tal de calcular la distància a la superfície, primer definim quins punts de la graella es troben sobre la superfície de m . Per a cada àtom $a_{x,y,z}$ de m , cal rastrejar el subespai que es troba al voltant de (x, y, z) en una distància D . Per cada punt de la graella que es troba en aquest subespai, es determina si es troba a l'interior, a l'exterior o sobre la superfície de la molècula. Una vegada finalitzat aquest procés, s'haurà extret la informació corresponent representada com la funció discreta següent:

$$g_{i,j,k} = \begin{cases} 1 & \text{superfície de la molècula} \\ 0 & \text{exterior de la molècula} \\ -1 & \text{interior de la molècula} \end{cases}$$

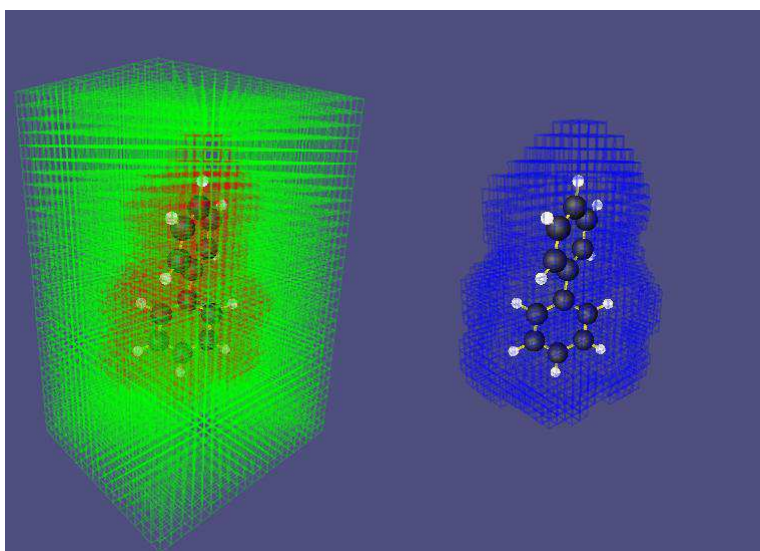


Figura 3.1: A la dreta, en color verd es representa l'exterior i en vermell l'interior de la molècula 2-bifenil. A l'esquerra es mostra en blau la superfície molecular.

El següent pas, és calcular la distància dels punts a la superfície molecular. A partir dels punts seleccionats com superfície molecular, ens expandim caps als punts veïns. Cada vegada que accedim a punt $g_{i,j,k}$ a partir d'un altre punt $g_{i',j',k'}$ veí, calculem la seva distància segons la funció següent:

$$g_{i,j,k} = \begin{cases} (g_{i',j',k'} + 1) & \text{si } g_{i,j,k} \text{ és extern a la molècula i } g_{i,j,k} < (g_{i',j',k'} + 1) \\ (g_{i',j',k'} - 1) & \text{si } g_{i,j,k} \text{ és intern a la molècula} \end{cases}$$

L'expansió dels punts es realitza de forma constant en totes les zones, és a dir, una zona no pot expandir-se més ràpidament que una altra. Per

tal de garantir-ho, s'utilitza una estructura auxiliar amb les propietats d'una cua. Aquesta cua s'inicialitza amb els punts que es troben en la superfície molecular. Al tractar un punt l'obtenim de l'inici de la cua i encuem al final els seus veïns.

Si al calcular la distància d'un punt, assignem una distància inferior a la que ja hi havia, tornem a visitar el seu veïnatge, per tal que tots els punts tinguin una distància mínima. Aquest fet només es realitza en els punts externs a la molècula, ja que si es tracten els punts interns de la mateixa manera podem caure en recorreguts infinits.

Una vegada finalitzada aquesta part, obtenim la graella on cada punt conté la distància mínima a la superfície molecular. A més, els punts interns a la molècula tindran valor negatiu per diferenciar-los dels punts externs a la molècula. Per tant, amb la graella construïda obtenim dos tipus d'informació:

1. Podem decidir si un punt qualsevol es troba dins o fora de la zona on es pot realitzar l'encaix.
2. Podem avaluar com de lluny es troba un punt qualsevol respecte la superfície molecular.

Amb la graella construïda, podem observar visualment la distància dels punts, colorejant-los segons la seva distància respecte la distància mínima i màxima. En la següent figura es mostra aquest fet .

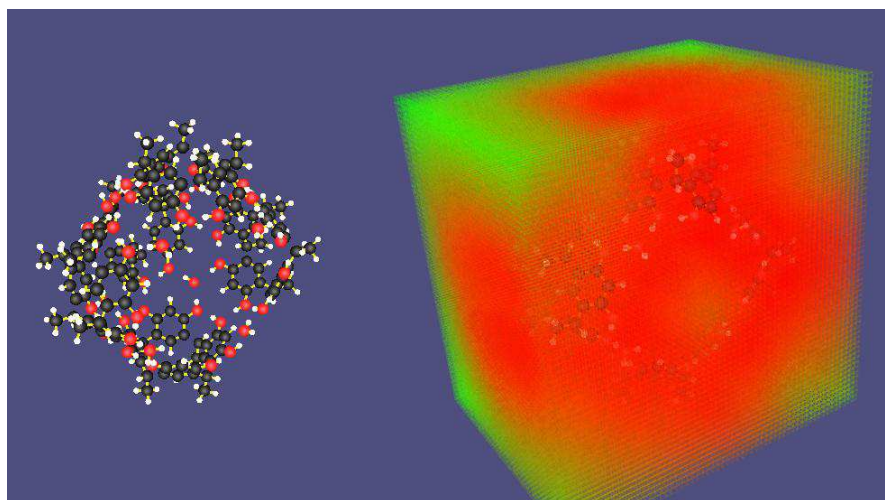


Figura 3.2: Les cel·les de la graella (dreta) es pinten amb la distància que contenen. Les cel·les en vermell es troben més a prop de la superfície, les verdes més lluny. A l'esquerra la mateixa molècula.

3.2 Algoritme d'encaix

Una vegada detallat el procés anterior, podem passar a explicar l'algoritme d'encaix molecular. Aquest algoritme permet situar de forma automàtica una molècula sobre una zona concreta d'una altra molècula, per tal que es pugui realitzar un encaix molecular. Aquesta nova situació de la molècula ha de garantir els requisits imposats detallats en l'inici d'aquest capítol. Per tal de garantir aquestes restriccions i que el procés sigui el més ràpid possible, utilitzarem la informació extreta en la secció anterior. Amb aquesta informació, podem decidir si una molècula està situada en una zona favorable per realitzar l'encaix. A més, podem avaluar la posició que ocupa segons la distància entre els àtoms de les dues molècules. Tot seguit, descriurem l'entrada de l'algoritme per després detallar el procés que es segueix.

L'entrada d'aquest algoritme necessària per realitzar el procés és la següent:

- Una molècula, que l'anomenarem A , on es situarà una altra molècula.
- Una altra molècula, que l'anomenarem B , la qual s'aproximarà a l'anterior.
- Un punt P qualsevol de la superfície de la molècula A on es vol situar la molècula B .

Amb l'entrada descrita podem detallar el procés de l'algoritme. Trobar una nova situació de la molècula B , significa escollir una solució, *la millor*, d'entre el conjunt possible de solucions. Per decidir quina solució és millor podem calcular les distàncies que hi ha entre els àtoms de B a la superfície de la molècula A . Com més petita és la suma de les distàncies, millor posició ocupa la molècula B . A més, no es vol situar en qualsevol posició, sinó en les proximitats del punt P . Per tant, com més petita la distància del conjunt d'àtoms de B respecte el punt P , millor és la solució trobada.

La solució adoptada per resoldre el problema, és utilitzar un algoritme de cerca avançada (hill-climbing) amb un heurístic que ens escollirà d'entre el conjunt de solucions, aquella que s'aproximi més a la solució final. Aquest heurístic intentarà minimitzar, per una banda, la distància dels àtoms de B respecte la superfície de la molècula A i, per una altra banda, la distància dels àtoms de B respecte el punt P . Per tant, obtenim el següent heurístic:

$$H() = \sum_{n=1}^{\|B\|} \text{dist}(a_{x,y,z}^n, A) + \sum_{n=1}^{\|B\|} \text{dist}(a_{x,y,z}^n, P)$$

Calcular l'heurístic per cada solució trobada no té un cost gaire elevat ja que és d'ordre lineal respecte el nombre d'àtoms de B . Calcular cadascuna de les parts de l'heurístic és senzill:

- En la primera part, utilitzem la informació de la graella associada a A (g^A). Per cadascun dels àtoms de B trobem la posició (i, j, k) en g^A . Amb els índexs accedim a g^A i obtenim la distància mínima a la superfície de A . Sumant les distàncies dels àtoms obtenim el resultat d'aquesta part.
- En la segona part de l'heurístic, tan sols cal calcular la distància entre cadascun dels àtoms de B i el punt P . Sumem les distàncies i obtenim el segon resultat.

Si al calcular $dist(a_{x,y,z}^n, A)$ accedim en una posició de g^A no vàlida (fora de rang), assignarem el valor d'infinít a la distància que estem calculant. No obstant, ens podem assegurar que aquest fet no es doni creant la graella (g^A) més gran, per tal que es pugui situar completament B en el seu interior.

A més de l'heurístic, necessitem una solució inicial d'on partirà l'algoritme de cerca, els moviments que realitzarem a la molècula B per obtenir noves solucions i definir quan una solució és vàlida.

Com a solució inicial, podem situar el centre de la molècula B a una distància de P en una direcció concreta. La distància ha de ser la necessària per no haver-hi una col·lisió en un principi. La podem calcular a partir de les caixes contenidores de les molècules A i B . És a dir, la molècula B es situarà fora de la caixa de A . Pel que fa a la direcció, l'obtenim a partir de la participació de l'usuari al seleccionar el punt P . La direcció de visió ens definirà la direcció anterior, és a dir, la direcció que s'estableix entre la posició de l'observador i el punt seleccionat. Amb aquests dos paràmetres obtenim la posició inicial de la molècula B .

Per modificar la solució obtinguda fins al moment per obtenir-ne d'altres que possiblement són millors que l'anterior, s'aplicaran transformacions geomètriques de translació a la solució en curs. Per definir una translació ens fa falta una direcció de desplaçament. Les direccions de desplaçament a aplicar s'obtingran a partir de la direcció de visió explicada en el paràgraf anterior. Del conjunt de direccions de desplaçament original (la direcció dels eixos de coordenades principals i la combinació d'aquests) s'obtenen aquelles que l'angle que forma entre una d'aquestes i la direcció de visió es trobi entre 90° i -90° . És a dir, s'escullen les direccions per tal que la molècula pugui avançar en la direcció de visió i mai pugui retrocedir. A més de la direcció, hem de definir la longitud del desplaçament. En aquest cas s'ha escollit la resolució de la graella, és a dir, el valor del paràmetre R com a aquesta longitud.

Per últim, hem de definir quan una solució és vàlida. La solució ha de complir la restricció imposada, és a dir, que la distància entre els àtoms d'una molècula i l'altra no pot ser inferior al valor del paràmetre D . Per

tant, cadascun dels àtoms de B han d'estar situats en punts de la graella g^A situats en la zona exterior, és a dir, amb valors positius.

Però aquest procés ens trobarà una solució per a una orientació predefinida de la molècula B . Per tant, prèviament hem de definir les orientacions que pot prendre la molècula B . Al ser el nombre d'orientacions molt elevat (per no dir infinit) hem de discretitzar aquest nombre. Podem definir un paràmetre Δ que ens proporcionarà el nombre de graus que hem d'aplicar en la rotació de la molècula B en cadascun dels eixos. En l'article [8, pp. 2197] li donen a aquest paràmetre un valor de 20° . Amb aquest valor obtenim 2916 orientacions possibles.

Amb les orientacions definides, executem per cadascuna d'elles l'algoritme de cerca. En cada execució, obtindrem la millor posició per aquella orientació. De totes les posicions obtingudes, escollim la solució que s'ajusti millor al problema. En la figura 3.3 podem veure els resultats d'aplicar aquest algoritme.

En les següents línies podem resumir l'algoritme:

1. Calculem la graella g^A de la molècula A .
2. Definim el paràmetre Δ per obtenir les orientacions de la molècula B
3. Per cada orientació anterior:
 - (a) Executem l'algoritme de cerca avançada.
 - (b) Obtenim el resultat de l'execució i el comparem amb el millor resultat obtingut fins ara.
4. Situem la molècula B en la millor posició obtinguda.

3.3 Millores i detalls de l'algoritme

L'algoritme descrit en la secció anterior pot donar en algunes situacions problemes en l'execució. Tot seguit descriurem problemes que poden ocórrer i les solucions que podem realitzar. A més, s'explicaran alguns detalls que poden millorar el temps d'execució de l'algoritme.

Un d'aquests problemes es dona quan l'algoritme de cerca escull un camí que porta la molècula B lluny del punt P desitjat. Aquesta elecció comporta un temps d'execució molt elevat, a més de una solució errònia. Per solucionar aquest problema podem definir una zona de cerca al voltant del punt P . Aquest paràmetre el pot definir l'usuari i informarà de la distància màxima respecte el punt P on es pot situar la molècula B ; prenent com a punt de

referència d'aquesta última el seu centre geomètric.

Un altre problema que es pot donar es troba en la posició final de la molècula B escollida. Potser aquesta posició no li interessa a l'usuari. Per solucionar aquest fet tans sols es pot realitzar un ajustament manual de B , veure secció 4.1.4. D'altra banda si l'orientació li és incorrecta, es pot realitzar el mètode a l'inrevés (seleccionar el punt P de la molècula B). Si el resultat tampoc és correcte per a l'usuari, aquest problema no té solució en el mètode actual. Una millora possible del mètode per solucionar aquest problema seria definir una zona d'interès en la molècula B per tal que aquesta zona sigui la *cara* que s'ajusti a la molècula A .

El problema que pot semblar més greu per part de l'usuari és el temps d'execució, si el nombre d'àtoms de la molècula B és molt elevat. Per solucionar-ho, es pot rebaixar el nombre d'orientacions (augmentar el paràmetre Δ) per tal que no es realitzi l'algoritme de cerca tans cops. Una vegada realitzat el procés, si el resultat no és del tot satisfactori es pot refinar aquest resultat per obtenir una solució que s'ajusti més al resultat desitjat.

D'altra banda, podem reduir el cost d'execució de l'algoritme de cerca reduint el conjunt de solucions possibles. Aquesta reducció del conjunt la podem realitzar de dues maneres:

1. Descartan aquelles solucions que s'allunyin massa de la solució millor obtinguda fins al moment.
2. Descartan aquelles solucions que després de realitzar N passos de cerca no troben un solució millor.

No obstant, podria ser que a l'usuari li interessés trobar la millor posició en qualsevol lloc de la molècula A , com els mètodes del capítol 2. Per realitzar-ho, tan sols cal executar l'algoritme anterior amb diferents punts P i escollir la millor posició d'entre totes. Aquests punts es poden obtenir mitjançant mostres sobre l'esfera contenidora de la molècula A i projectant-les sobre la mateixa molècula en direcció al centre d'aquesta.

3.4 Resultats

Les següents taules mostren els temps d'execució de cadascuna de les parts de l'algoritme anterior. L'algoritme s'ha executat sobre un PC amb un processador Core 2 Duo a 1.83 GHz i amb 1 Gb de RAM.

Molècula	Num Àtoms	Resolució Graella	Temps sec.
2-bifenil	22	18x31x21	0.062
Tetrafenil	42	33x33x29	0.109
A-opencube	384	46x58x58	0.859

Taula 3.1: Temps de construcció de la graella de diferents molècules

Molècula	Num Àtoms	Temps Mín	Temps Màx
1-benze	12	10.32	42.34
2-bifenil	22	23.12	88.74
6-1,3,5-tetra-fenil	42	23.73	125.37
A-opencube	384	247.18	673.28

Taula 3.2: Temps mínim i màxim en el procés de docking

Les molècules de la taula 3.2 es refereixen a la molècula *B* de l'algoritme. No cal tenir en compte quina molècula *A* estem utilitzant ja que el càlcul és independent d'aquesta última.

Podem observar que el temps del procés de docking augmenta amb el nombre d'àtoms que té la molècula. A més, la diferència existent entre els temps d'execució es deu a la situació del punt *P*. Com més a prop de la caixa contenidora, és a dir, situat més a l'exterior, el temps del procés disminueix. Això es degut a la distància que s'estableix entre el punt *P* i el punt d'on parteix l'algoritme de cerca, com més petita aquesta distància menys passos de translació haurà d'aplicar.

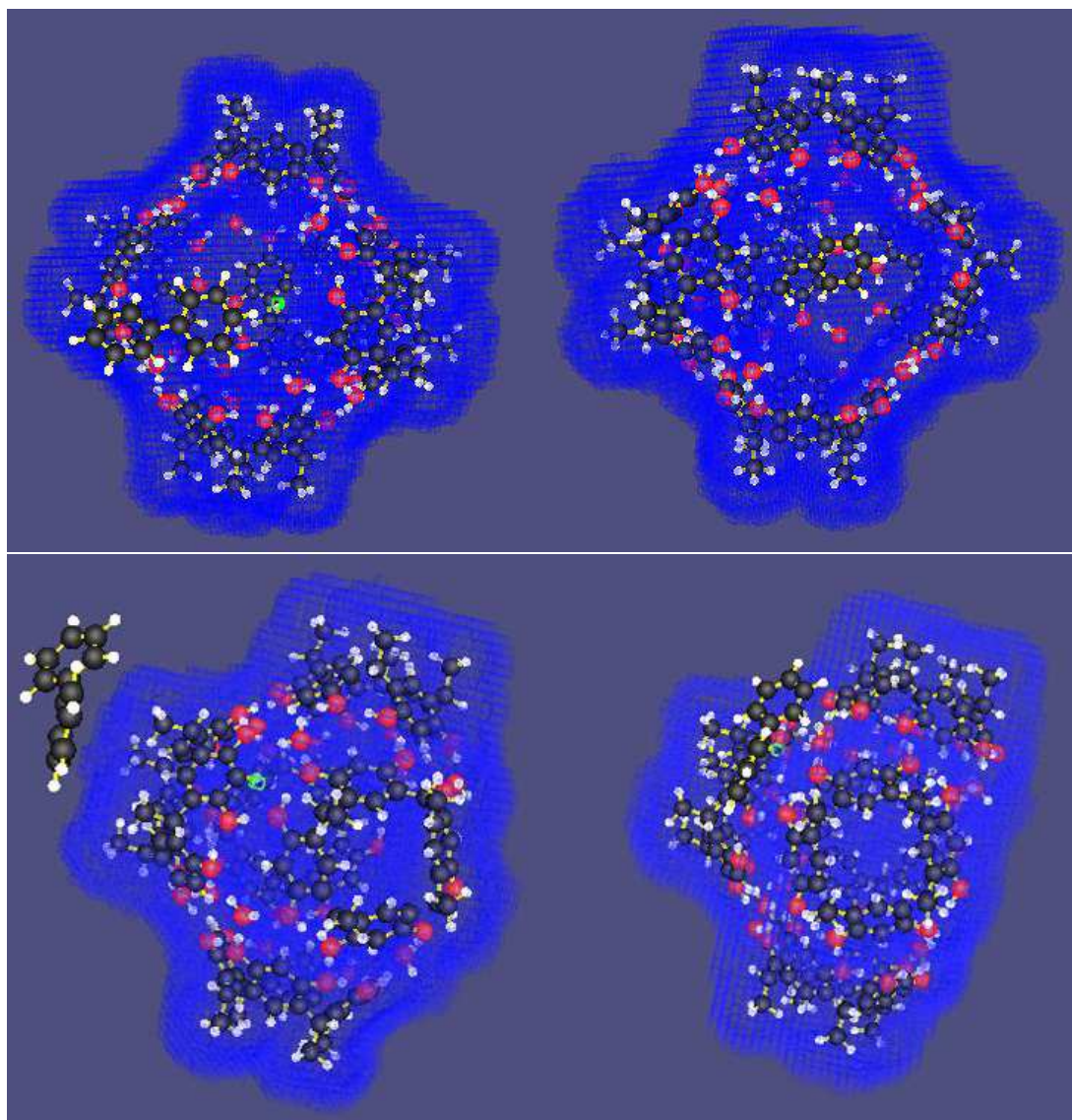


Figura 3.3: Resultats de l'algoritme de cerca (dreta) en dos situacions diferents; a dalt el punt P (de color verd) està situat en el buit de la molècula; a baix el punt P es troba a la cara exterior de la molècula. A l'esquerra es mostren les molècules abans d'aplicar l'algoritme.

Capítol 4

L'aplicació en detall

Amb l'especificació de l'algoritme realitzada, passem a la implementació d'aquest mitjançant una aplicació que funcionarà sobre un PC qualsevol. Aquesta ens permetrà realitzar la manipulació visual necessària de les molècules per poder aplicar el mètode d'encaix.

L'aplicació està desenvolupada en llenguatge C++ i OpenGL en l'apartat gràfic. A més, l'aplicació compta amb una interfície gràfica desenvolupada amb Qt.

A continuació, es detallaran les funcionalitats que permet realitzar l'aplicació, per després donar una pinzellada a la implementació realitzada (descripció dels mòduls creats i els mètodes que conté).

4.1 Funcionalitats

En aquesta secció es presenten les funcionalitats disponibles en l'aplicació desenvolupada. En cadascuna de les funcionalitats, es detalla el seu comportament. Els events necessaris per realitzar les funcionalitats estan descrits de forma genèrica, enlloc de definir els botons corresponents dels dispositius d'entrada, ja que aquests poden ser definits per l'usuari.

4.1.1 Afegir/Obrir una molècula

Permet afegir una molècula a les ja existents en l'escena. Si poden afegir aquelles que tinguin el format PDB o XYZ (aquests dos formats són estàndards i utilitzats en moltes altres aplicacions). L'aplicació reconeix el format d'entrada i llegeix la molècula amb el mètode corresponent segons el format. Durant la lectura, es guarda la informació necessària per a les operacions posteriors. S'emmagatzema el tipus de cadascun dels àtoms, així com la seva

posició en la molècula. Un cop afegida a l'escena, durant el primer moviment que es realitza no es calcula la intersecció amb altres molècules, per tal d'evitar solapaments entre molècules i situar-la en un lloc buit de l'escena.

4.1.2 Eliminar una molècula

Permet eliminar una molècula que estava en l'escena, prèviament seleccionada.

4.1.3 Seleccionar una molècula

Amb l'event corresponent, permet seleccionar una molècula present en l'escena. Per seleccionar una molècula, situem el punter sobre qualsevol dels àtoms d'aquesta i realitzem l'event corresponent de selecció. Un cop seleccionada, es mostra la capça contenidora d'aquesta. Si seleccionem una molècula que prèviament ho estava, es deselecciona.

4.1.4 Moure la molècula

Qualsevol de les molècules es pot moure lliurement per l'escena. Per tal de moure'n una, cal que prèviament estigui seleccionada. Amb el moviment corresponent del dispositiu podem traslladar la molècula per l'escena. Amb el mateix moviment afegint l'event adequat, podem realitzar una rotació sobre la molècula. Durant la realització de qualsevol dels dos moviments es determina si es produeix una intersecció entre molècules. Si en el moviment a realitzar es determina una col·lisió entre molècules, aquest moviment no s'executarà. El càlcul de la col·lisió es detalla en l'apartat 4.2.3 i 4.2.4.

4.1.5 Docking automàtic

Aquest mètode és la part central de l'aplicació i s'encarrega del càlcul del "soft docking" d'una molècula respecte una altra. Per tal de realitzar-ho, cal seleccionar una molècula i després seleccionar qualsevol posició (en direm posició destí) d'una altra molècula present a l'escena. Al calcular la posició destí, es té en compte la posició de l'observador segons el següent mètode: calculem la recta que uneix l'observador i el punter de selecció i, a partir d'aquesta, aconseguim la posició destí com el primer punt que es troba sobre la superfície de la molècula (segons el sentit observador \rightarrow punter de selecció).

Un cop seleccionats la molècula i la posició destí, es realitza el càlcul de la posició final de la molècula. Aquest càlcul es detalla en l'apartat 3.2. Després

del càlcul, es mostra una simulació del moviment de la molècula on es veu com s'aproxima a la posició final calculada.

4.1.6 Unir molècules

Aquest mètode permet unir molècules que es trobin a una distància inferior definida per l'usuari. Per poder unir dues molècules, cal seleccionar primer una i després seleccionar l'altra, tot realitzant l'event d'unió corresponent. Una vegada unides les molècules, totes dues formaran part del mateix conjunt i les accions que es realitzin es tractaran com si fos una única molècula.

4.2 Implementació de l'aplicació

En aquesta secció detallarem com ha estat implementada l'aplicació, així com els mòduls que la conformen i els mètodes i funcions principals que conté cadascun dels anteriors.

L'aplicació està dividida en 5 mòduls amb les seves funcionalitats específiques: Àtom, Enllaç, Molècula, Graella Intersecció i Model. A continuació, entrarem a explicar cadascun d'aquests.

4.2.1 Mòdul Àtom

Aquest mòdul conté la informació d'un àtom, és a dir, la posició que ocupa a l'espai i el tipus d'àtom (el número atòmic amb el qual es sabran les seves propietats).

Per tant, aquesta classe contindrà els mètodes necessaris per crear un àtom nou especificant el seu nombre atòmic i la seva posició, així com d'altres per obtenir la informació específica i per modificar la posició d'aquest. A més, durant la visualització, aquesta classe s'encarrega de representar l'àtom en pantalla.

4.2.2 Mòdul Enllaç

En aquest mòdul es representa la unió existent entre dos àtoms. Per tant, conté la informació necessària d'aquest àtom. En aquest cas, com utilitzem C++, conté els punters dels àtoms. Aquest mòdul tan sols guarda aquesta informació per utilitzar-la en el procés de visualització, per tal de representar gràficament l'enllaç.

Per tant, com a mètodes, tans sols està implementat el mètode per crear un nou enllaç i un altre per representar gràficament l'enllaç durant la visualització.

4.2.3 Mòdul Graella Intersecció

Aquest mòdul conté la graella que s'extrau d'una molècula i es realitza el mètode descrit en la secció 3.1. Per tant, contindrà una matriu tridimensional de longituds corresponents, on cada posició es guarda un byte (amb 256 valors diferents és suficient per representar les distàncies).

Els mètodes que conté permeten:

- El càlcul de les distàncies de la graella descrits en la secció 3.1 indicant la posició dels àtoms d'una molècula.
- La comprovació de la possible intersecció amb una altra molècula M . Aquest mètode es realitza comprovant si la posició de tots els àtoms de la molècula M es situen en la zona exterior de la graella.
- L'encaix automàtic descrit en la secció 3.2, indicant, a més de la graella actual, l'altra molècula on s'haurà aplicat una de les rotacions del conjunt. Una vegada finalitzat el mètode retornarà la millor posició que l'algoritme a escollit.

4.2.4 Mòdul Molècula

Aquest mòdul s'encarrega d'emmagatzemar la informació d'una molècula, és a dir, els àtoms que la formen i els enllaços que uneixen aquests àtoms. Els àtoms estan representats com un array d'aquesta mateixa classe i els enllaços de la mateixa forma. A partir d'aquesta informació podem construir la seva caixa contenidora i la graella associada a la seva geometria.

Els mètodes implementats en aquesta classe permeten la manipulació de la molècula. Els mètodes més importants són els següents:

- Llegir una molècula: indicant el fitxer que conté la representació de la molècula, permet llegir aquesta informació. Segons el format en que es representa, s'escull el mètode de lectura.
- Moure una molècula: permet desplaçar o realitzar un moviment de rotació indicant-li la longitud del desplaçament o la direcció de rotació. A més, es realitzarà el moviment si es compleix el primer requisit dels enumerats en el capítol 3. Per comprovar-ho, s'utilitza la graella associada intersectant-la amb la posició dels àtoms de les altres molècules.

Al realitzar la comprovació entre dues molècules, s'utilitza la graella de la molècula amb major nombre d'àtoms.

- Encaix automàtic: en aquest mètode es realitzen les operacions prèvies per realitzar l'encaix, com el càlcul de direccions de desplaçament o el conjunt de rotacions a aplicar (a la molècula *B*, segons la secció 3.2). Després es realitza la crida pertinent sobre la graella associada (de la molècula *A*, segons la secció 3.2). Després, entre les posicions possibles, s'escull la que millor compleix les restriccions imposades.
- Unió de dues molècules: indicant les dues molècules a unir, es construeix la nova graella associada a la unió realitzada. A partir d'aquest moment, les accions que es realitzen afectaran a les dues molècules.

4.2.5 Mòdul Model

Aquest mòdul conté el conjunt de molècules amb les quals es pot interactuar. És l'entrada a les funcionalitats de la secció anterior; els seus mètodes són els mateixos que els de la classe Molècula, però es realitzen algunes operacions prèvies per adequar l'entrada als mètodes posteriors.

4.2.6 Interfície gràfica

A més dels mòduls descrits anteriorment, l'aplicació que s'executa sobre el PC es complementa amb una interfície gràfica desenvolupada amb Qt.

Aquesta interfície s'encarrega de realitzar les crides pertinents al mòdul Model, segons els events que rep dels dispositius d'entrada (teclat i ratolí).

Capítol 5

Sistema de Realitat Virtual

Uns dels punts d'aquest projecte és l'execució de l'aplicació en un sistema de realitat virtual. En aquest capítol es tractarà el sistema de realitat virtual utilitzat, la plataforma per executar l'aplicació i, finalment, la migració de l'aplicació descrita en el capítol anterior a aquest sistema.

5.1 Sistema de Realitat Virtual

El sistema triat ha estat la CAVE, un dels sistemes que ofereix a l'usuari un major grau d'immersió. Una CAVE (acrònim de Computer-Animated Virtual Environment) consisteix en una habitació amb unes dimensions al voltant dels 3x3x3 m, en la qual de 2 a 4 parets laterals, el terra i/o el sostre, són pantalles on es projecten imatges estereoscòpiques.

Com a dispositiu d'entrada podem utilitzar el mouse o el Wanda; un punter 3D amb el qual, gràcies al sensor magnètic que duu incorporat, coneixem la seva posició i orientació (6DoF), a més d'incorporar tres botons i un mecanisme similar al joystick, de 2 graus de llibertat (2DoF x,y).

Amb el sistema de realitat virtual i els dispositius utilitzats definits, falta per descriure la plataforma d'execució escollida. S'ha escollit XVR (tecnologia desenvolupada per la companyia VRMedia) per la seva facilitat d'ús, tant d'implementació com d'integració en el sistema de realitat virtual. XVR utilitza un llenguatge propi semblant al C++, però amb característiques del llenguatge script. En l'apartat gràfic utilitza OpenGL per a la visualització de l'escena

Generalment, aquesta tecnologia és utilitzada en entorns web per afegir contingut 3D. Però es pot aplicar fàcilment en el sistema CAVE. Per tant, amb l'aplicació finalitzada, podrà ser utilitzada també en un entorn web.



Figura 5.1: A l'esquerra una visualització d'un model en un entorn CAVE; a la dreta el dispositiu Wanda

5.2 Implementació de l'aplicació

Amb tot el sistema utilitzat definit podem passar a explicar la implementació de l'aplicació sobre aquest nou sistema. XVR permet importar llibreries externes i utilitzar les funcions que contenen. Per tant, la implementació descrita en el capítol anterior es pot aprofitar directament amb totes les seves funcionalitats, tan sols s'ha d'encapsular tots els mòduls en una llibreria. A més, s'ha de crear una nova llibreria que realitzarà el paper d'interfície entre XVR i la llibreria anterior, ja que XVR no permet la definició de classes externes. S'ha de tenir en compte que el procés de visualització ha de tenir un cost d'execució molt petit, ja que en el sistema CAVE la geometria a visualitzar s'envia tans cops com pantalles té el sistema. Per tant, forçosament, hem d'escollir mètodes com *Display Lists* en l'etapa de visualització; un mètode que en l'aplicació per PC no era del tot necessari.

Amb les llibreries creades ens falta la part d'interacció amb l'usuari i ens substitueix a l'interfície gràfica de la secció 4.2.6. Aquesta part està realitzada en llenguatge XVR. Es pot dividir en les següents parts:

- Inicialització: aquí es realitzen les inicialitzacions pertinents, com importació de llibreries, definició de paràmetres, etc.
- Tractament d'events: inclou els mètodes per al tractament dels events dels dispositius d'entrada utilitzats. Com aquests mètodes no s'executen quan es produeix un event, s'ha de realitzar cada cert temps (per exemple cada cop que es pinta l'escena) per capturar els possibles events generats pels dispositius.
- Pintat de l'escena: aquesta rutina s'ha d'executar cada cert temps, el qual es pot definir amb un paràmetre. Quan es vol pintar l'escena es

crida el mètode corresponent de la llibreria creada.

Es pot comprovar que la migració de l'aplicació a la nova plataforma no és gaire difícil, tans sols s'han de realitzar uns petits ajustaments per que funcioni correctament. A més, com tots dos sistemes utilitzen OpenGL en la part de visualització es pot realitzar el pintat de l'escena des de la llibreria creada. D'aquesta manera la llibreria pot utilitzar-se tan en la versió per al sistema virtual, com en la versió per PC.

Bibliografia

- [1] Cesareni-G. Ausiello, G. and Helmer-Citterich M. Escher: A new docking procedure applied to the reconstruction of protein tertiary structure. *PROTEINS: Structure, Function, and Genetics*, 28:556–567, 1997.
- [2] J. Cherfils and J. Janin. Protein docking algorithms: Simulating molecular recognition. *Curr. Opin. Struct. Biol.*, 3:265–269, 1993.
- [3] Katchalski-Katzir E. Eisenstein M. Geometric recognition as a tool for predicting structures of molecular complexes. *Letters in Peptide Science*, 5:365–369, 1998.
- [4] Katchalski-Katzir E. Eisenstein M. On proteins, grids, correlations, and docking. *C. R. Biologies*, 327:409–420, 2004.
- [5] Lin-S.L. Wolfson H.L. Nussinov R. Fischer, D. A geometry-based suite of molecular docking processes. 248:459–477, 1995.
- [6] M. Helmer-Citterich and Tramontano A. Puzzle: A new method for automated protein docking based on surface shape complementarity, 1994.
- [7] Fan Jianga and Sung-Hou Kimb. Soft docking: Matching of molecular surface cubes. *Journal of Molecular Biology*.
- [8] Shariv I. Eisenstein M. Friesem A. Aflalo C. Vakser I. Katchalski-Katzir, E. Molecular surface recognition: Determination of geometric fit between proteins and their ligands by correlation techniques. *Proc. Natl. Acad. Sci. USA*, 89:2195–2199, March 1992.
- [9] Kuntz I.D. Shoichet, B. Protein docking and complementarity. *J. Mol. Biol.*, 221:327–346, 1991.

Índex de figures

1.1	Científic de la Universitat de Calgary (Canadà) utilitzant una eina bioinformàtica en un sistema CAVE	5
2.1	Càlcul del grau de complementarietat (C)	8
2.2	Construcció de la poligonal (dreta) a partir d'una llesca (esquerra)	8
3.1	A la dreta, en color verd es representa l'exterior i en vermell l'interior de la molècula 2-bifenil. A l'esquerra es mostra en blau la superfície molecular.	13
3.2	Les cel·les de la graella (dreta) es pinten amb la distància que contenen. Les cel·les en vermell es troben més a prop de la superfície, les verdes més lluny. A l'esquerra la mateixa molècula.	14
3.3	Resultats de l'algoritme de cerca (dreta) en dos situacions diferents; a dalt el punt P (de color verd) està situat en el buit de la molècula; a baix el punt P es troba a la cara exterior de la molècula. A l'esquerra es mostren les molècules abans d'aplicar l'algoritme.	20
5.1	A l'esquerra una visualització d'un model en un entorn CAVE; a la dreta el dispositiu Wanda	28

Índex de taules

3.1	Temps de construcció de la graella de diferents molècules . . .	19
3.2	Temps mínim i màxim en el procés de docking	19