POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria Informatica
Dipartimento di Elettronica e Informazione

Progetto e sviluppo di un'interfaccia multi-sensoriale per un simulatore di meccanica molecolare

Relatore: Prof.ssa Sara COMAI
Correlatore: Ing. Davide MAZZA

Tesi di Laurea di:
Lorenzo MUREDDU, matricola 725402

Anno Accademico 2009-2010

A mia moglie Ornella.

Sommario

Le simulazioni in realtà virtuale rappresentano un potente strumento per l'indagine scientifica, tanto più se l'interazione con l'ambiente ricreato avviene attraverso l'utilizzo di molteplici sensi. Un alto grado di immersività può essere raggiunto grazie alla conpresenza di interfacce aptiche, visuali e sonore. La possibilità di interagire attraverso il tatto con le tecnologie e soprattutto di ricevere feedback aptici è ancora sconosciuta al grande pubblico, se non in minima parte attraverso l'avvento dei touch screen, e rappresenta un ambito di ricerca molto attivo.
In questo contesto si inserisce il presente lavoro di tesi che consiste nella realizzazione di un sistema di simulazione molecolare con interfacciamento aptico, grafico e sonoro. Attraverso un particolare device è possibile percepire sia la geometria di una molecola che il campo di forza generato dal suo potenziale, il tutto accompagnato da varie modalità di rappresentazione grafiche e dall'utilizzo di particolari effetti sonori. Inoltre nel sistema è stata inserita una simulazione dinamica della geometria molecolare che permette di agire sull'ambiente virtuale direttamente, attraverso lo strumento aptico, spingendo letteralmente l'oggetto rappresentato.
L'applicazione è stata realizzata al fine di fornire uno strumento sia per la didattica che per la ricerca scientifica nell'ambito chimico. A tal proposito la fase di progetto è stata contraddistinta da continui risconti di esperti nei vari settori interessati quali chimica, design e tecnologie per la didattica.

Ringraziamenti

Il primo e più grande ringraziamento va a mia moglie Ornella che in tutti questi anni di università mi ha costantemente sorretto e incoraggiato. Nella stesura di questa stessa tesi è stata di fondamentale importanza correggendo la costruzione delle frasi ivi contenute, spesso inutilmente contorte. Sposarti è stata la decisione più saggia che potessi mai prendere. Ti amo davvero tanto.
Un sentito grazie alla professoressa Sara Comai e a Davide Mazza, che mi hanno seguito e assistito nell'intero arco del progetto e soprattutto che mi hanno permesso di lavorare su di uno strumento tanto stimolante da un punto di vista creativo. Ringrazio inoltre Marzio Ghezzi per le numerose idee che ha suggerito durante la realizzazione dell'applicazione.
Un grazie particolare va a mio padre che mi ha permesso di completare il percorso accademico nonostante le difficoltà che ci sono state all'interno della nostra famiglia. Ricordati sempre che ti voglio bene papà.
Un saluto e un grazie a tutti i miei amici, che mi hanno accompagnato nella lunga via che porta alla agognata laurea e non solo. I tre testimoni in primis: Sciac, Manuel e Mirons, ma anche Luca, Ogni e Vege, speriamo di poter passare ancora tanto tempo insieme e di restare uniti nonostante la vita ci porti a scegliere strade diverse.
Infine un pensiero particolare va alle persone che ho conosciuto durante tutta la mia vita universitaria. Sono grato a professori come Colombetti, Cagliotti, Barbieri, Bonarini e a professoresse come Garzotto e Comai per essere riusciti a trasmettere la loro passione nelle materie insegnate. Ringrazio inoltre gli assistenti che ho incontrato durante la mia carriera, in particolare Giusti, Febretti e Mazza che mi hanno seguito e assistito nella realizzazione di numerosi progetti. L'università non è fatta solo di esami, prima di tutto è composta dalle persone che ci lavorano e che mettono il cuore nel cercare di trasmettere il loro sapere, con questa accezione sono grato all'intero Politecnico di Milano, sicuro di conservare per tutta la vita bei ricordi legati a questi particolari anni della mia vita.

Indice

Elenco delle figure

Capitolo 1

Introduzione

FIN DAGLI ALBORI della storia l'uomo ha sognato di poter creare e modificare, attraverso la sua immaginazione, la realtà. Prima la mitologia, poi le arti e infine la scienza hanno cercato di dare risposte alla fisicità del mondo attraverso la creatività umana. L'uomo ha sempre vissuto immerso in costruzioni sociali che ne plasmano la percezione della realtà, mentre spesso fugge con la mente in altre realtà fittizie dettate dall'immaginazione altrui. Il fabbisogno creativo dell'uomo è pari solo alla sua necessità di collocarsi nel mondo, nell'universo che lo circonda, alla necessità di dare delle risposte alle infinite domande che sa porsi. Alcuni dispensano idee ed altri le accolgono, per ottenere una nuova visione del loro mondo.

La costruzione di realtà fittizie e controllate non è quindi una novità del mondo contemporaneo in quanto fonda le sue radici nei più intimi fabbisogni dell'uomo. Da tanto lontano deriva l'idea di realtà virtuale, tentativo odierno di ricreare una realtà in cui l'uomo possa fuggire e ripararsi, che possa sviluppare la sua sete di immaginazione. Tramite ambienti simulati l'uomo è oggi in grado di esplorare e vivere esperienze nuove in mondi interamente fittizi con l'ausilio dei suoi stessi sensi. Si pensi anche solo agli interi mondi virtuali rappresentati nei videogiochi.

La realtà virtuale rappresenta quindi un modo per esperire in un contesto protetto elementi altrimenti inaccessibili o pericolosi. Proprio questa caratteristica permette l'uso di simulazioni in realtà virtuale nell'esplorazione di aspetti inaccessibili della realtà stessa.
É ora possibile, programmando opportunamente una simulazione variare le leggi fisiche, visualizzare atomi e molecole, esplorare le caratteristiche più

lontane dell'universo .

L'esperienza di realtà simulate, virtuali, trova una misura della sua efficacia nel grado di immersività che è in grado di offrire. Il coinvolgimento di molteplici sensi nell'interazione con l'utente permette di rendere particolarmente efficace la simulazione, garantendo un alto coinvolgimento emotivo e mentale di quest'ultimo. Lo scopo della realtà virtuale è rapire la mente dell'utente che ne fa uso, in modo da illuderlo, seppur parzialmente, della fisicità dell'oggetto rappresentato. A tal proposito alla riproduzione visuale di oggetti virtuali è stata in primis accostata una rappresentazione sonora, poi un'interazione aptica[1] con l'ambiente simulato ed infine, solo recentemente, caratteristiche olfattive del sistema.

In questo contesto si inserisce il presente lavoro di tesi che consiste nella realizzazione di un simulatore di meccanica molecolare interfacciato visualmente, acusticamente ed apticamente con l'utente. Nello specifico vengono renderizzate molecole che è possibile vedere sotto forma di rappresentazione geometrica, la quale può essere 'toccata' e 'spinta' attraverso l'interfacciamento con un device aptico, il tutto accompagnato da elementi sonori che ne descrivono ulteriori caratteristiche.
Va a sommarsi a questa simulazione geometrica la possibilità di percepire apticamente il campo di forze generato dal potenziale elettrostatico della molecola (MEP[2]) dove il device simula il comportamento di una carica di prova definendo la direzione e la forza con cui la mano dell'utente debba essere spinta. Inoltre, sono state inserite varie modalità di rappresentazione grafica del potenziale in completamento all'interazione aptica, nonché un accompagnamento sonoro capace di fornire ulteriori informazioni.

Il presente lavoro trae le sue origini da un precedente progetto di tesi [1] nel quale veniva rappresentata graficamente la molecola ed in cui era possibile percepire apticamente il potenziale attorno alla molecola. Sebbene abbastanza funzionale, tale progetto presentava gravi carenze sia dal punto di vista grafico che aptico poichè, presentando scarse informazioni portava all'ottenimento di una percezione aptica dettata da instabilità nei punti critici.
In luce degli aspetti appena evidenziati si è lavorato dunque in maniera incrementale correggendo errori nel sistema originale ed aggiungendo fun-

[1] che coinvolge il senso del tatto
[2] Dall'inglese Molecular Electrostatic Potential

zionalità ex novo.

Lo scopo di quanto realizzato è da ricercarsi nell'ambito della didattica della chimica e della ricerca scientifica. Con il presente progetto si intende offrire un valido strumento di facilitazione nello studio della chimica abile altresì nell'indurre i ricercatori a formulare nuove idee sulla base di un'innovativa possibilità di esperire il potenziale molecolare.
La fase progettuale del presente elaborato è stata infatti realizzata in collaborazione con esperti di diverse aree disciplinari, quali chimica e design, nonché con il contruibuto del centro METID (Metodi E Tecnologie Innovative per la Didattica) del Politecnico di Milano per quanto riguarda la definizione degli elementi didattici del sistema.

Per la realizzazione dell'applicazione è stato utilizzato un device aptico conformato come un braccio meccanico capace di restituire forze sul suo end-effector, chiamato PHANToM. Tale strumento possiede 6 gradi di libertà di cui solo 3 capaci di generare una forza[3].
Dal punto di vista software, invece, sono state utilizzate varie librerie C++ ognuna delle quali preposta alla comunicazione con una diversa tipologia di dispositivo[4]. L'intero progetto realizzato è infatti definito da codice e classi C++.

Strutturalmente l'applicazione realizzata si sviluppa su vari engine che definiscono caratteristiche separate dell'applicazione:

- **Engine Grafico**: controlla le rappresentazioni grafiche della moleocola nella scena 3d;
- **Engine per l'Interfaccia Utente**: si occupa di visualizzare e gestire le opzioni del sistema;
- **Engine Sonoro**: gestisce l'interazione sonora del sistema;
- **Engine Aptico**: interfaccia di gestione del device aptico;
- **Engine Dinamico**: si occupa di realizzare la simulazione dinamica dell'oggetto geometrico definito dalla molecola.

Questa struttura ad engine permette di avere un software piuttosto scalabile, capace di essere modificato agevolmente cambiando una sola delle caratteristiche del sistema. Ognuno di questi engine definisce, infatti, una diversa

[3] i tre traslazionali
[4] quali device aptico, scheda video, scheda sonora,...

classe del sistema.

La suddivisione in capitoli definisce:

- **Capitolo 2, Stato dell'arte**: in primis verrà visionato il panorama tecnologico e chimico in cui l'applicazione sviluppata si colloca. Si spiegherà la natura dei concetti chimici introdotti nel lavoro e si porterà vari esempi di interfacce e applicazioni aptiche, sia da un punto di vista tecnologico inerente le tipologie di device presenti in letteratura che da un punto di vista di sistemi didattici per la chimica. Inoltre verranno illustrati esempi di applicazioni multisensoriali facenti uso sia di interfacciamento aptico che sonoro.
- **Capitolo 3, Impostazione del problema di ricerca**: in questo capitolo verranno prima illustrate le motivazioni alla base dello sviluppo del progetto, poi le tecnologie utilizzate sia dal lato hardware che software. Sarà quindi prima di tutto descritto il device aptico presentato, dopodiché verranno illustrate le numerose librerie utilizzate.
- **Capitolo 4, Progetto logico della soluzione del problema**: nel quarto capitolo sarà descritta profusamente la struttura logica del sistema e si andranno a valutare tutte le funzionalità introdotte nei vari engine da un punto di vista logico/matematico. Si motiverà inoltre ogni scelta progettuale affrontata.
- **Capitolo 5, Aspetti Implementativi**: questo capitolo approfondisce la struttura delle classi e dei metodi implementati, fornendo una guida al codice per chi volesse espandere quanto sviluppato in questa sede.
- **Capitolo 6, Realizzazioni sperimentali e valutazione**: a seguire, nel sesto capitolo, verranno descritte tutte le molecole utilizzate nel corso dello sviluppo del sistema e saranno elencati e definiti i metodi di valutazione del sistema, sotto vari punti di vista. Tale valutazione per la natura del sistema stesso presenta diverse sfaccettature che vanno analizzate una ad una.
- **Capitolo 7, Direzioni future di ricerca e conclusioni**: per concludere si analizzano possibili direzioni di sviluppo future e si traggono le fila di quanto effettivamente realizzato.
- **Appendice A, Progetto collaterale al framework realizzato**: in aggiunta al presente lavoro, in appendice, viene presentato un sistema realizzato dal medesimo autore di questa tesi che si propone come

generalizzazione di una parte del lavoro svolto. Si tratta infatti di un framework capace di caricare al suo interno mesh generiche e di permettere la percezione di tali mesh attraverso l'interfaccia aptica. Il lavoro svolto si pone come banco di prova per testare proprietà aptiche delle superfici, attrito e durezza ad esempio, su una generica mesh in formato obj.

Capitolo 2

Stato dell'arte

All'interno del secondo capitolo si introducono gli argomenti affrontati e si illustrano progetti simili a quello qui sviluppato. Si predispongono, quindi, le basi teoriche per la comprensione della successiva trattazione.
Il tool sviluppato in questa sede rappresenta un visualizzatore di molecole con interfaccia grafica, sonora e aptica.
Dal punto di vista chimico il lavoro qui svolto fa riferimento alla modellistica molecolare, branca della chimica che comprende tutti i metodi teorici e le tecniche computazionali utilizzate per rappresentare o simulare il comportamento delle molecole. La caratteristica comune delle tecniche di modellistica molecolare è il livello atomistico di descrizione dei sistemi molecolari; il più piccolo livello di informazione è rappresentato dagli atomi individuali. Questa disciplina fa largo uso del computer ed è utilizzata sia per la rappresentazione grafica delle molecole in campo didattico sia per la realizzazione di simulazioni nell'ambito della ricerca scientifica, dove sono impiegate, ad esempio, per la scoperta di nuovi farmaci o per le indagini di complesse interazioni molecolari che sono alla base dei processi metabolici.
Nello specifico il simulatore realizzato in questa sede tratta gli aspetti della meccanica molecolare, definita come branca della modellistica molecolare. La natura della meccanica molecolare verrà descritta in seguito nel presente capitolo.
A seguire verrà introdotto il concetto di aptica e verranno illustrati esempi di applicazioni e tecnologie facenti uso di interfaccce aptiche eventualmente accompagnate da strumenti visivi e sonori.

2.1 Concetti chimici: Meccanica Molecolare

La meccanica molecolare è quella branca delle chimica computazionale che si prefigge lo scopo di descrivere le molecole tramite le leggi della fisica classica, nonostante si stia parlando di oggetti quantomeccanici. Il vantaggio di tale tecnica consiste nell'evitare un formalismo complesso e dispendioso come quello della meccanica quantistica potendo così trattare con facilità e senza costi computazionali eccessivi molecole grandi come polimeri e proteine.
La meccanica molecolare trova la propria espressività nel Force field o campo di forze, ossia un potenziale che descrive le caratteristiche energetiche che la molecola genera attorno a sé: in questo modo è possibile valutare le proprietà reattive della molecola in esame rispetto ad un particolare ambiente, in una certa conformazione, o rispetto ad un'altra molecola.
Applicando il campo di forze è possibile calcolare l'energia potenziale di una molecola considerando la somma di singoli contributi energetici dati dal legame chimico covalente e da interazioni non di legame. L'energia E risulta data dalla somma

$$E = E_{covalente} + E_{noncovalente} \tag{2.1}$$

dove i singoli contributi energetici sono dati rispettivamente dalle somme

$$E_{covalente} = E_{str} + E_{bend} + E_{tors} \tag{2.2}$$

e

$$E_{noncovalente} = E_{el} + E_{vdW} \tag{2.3}$$

I termini E_{str} e E_{bend}, ovvero le energie di stretching e bending del legame chimico, sono generalmente ricavati utilizzando l'espansione in serie di Taylor del potenziale armonico sulla base di lunghezze di legame di equilibrio ricavate generalmente da calcoli basati su dati sperimentali. Il termine E_{tors}, ovvero l'energia di torsione attorno un legame a formare un nuovo angolo diedro, possiede tipicamente minimi multipli e quindi non può essere determinato utilizzando il modello dell'oscillatore armonico. Questo termine viene normalmente calcolato utilizzando l'espansione in serie di Fourier della funzione periodica $\sum_{n=1}^{n} V_n cos(n\omega)$, dove ω è l'angolo diedro formato e n la periodicità ($n = 1$ 360°, $n = 2$ 180°, e così via).
I termini energetici non di legame richiedono un maggiore costo computazionale per il loro calcolo, coinvolgendo contemporaneamente più atomi rispetto al legame chimico. Il termine legato all'energia elettrostatica, E_{el}, viene calcolato utilizzando il potenziale di Coulomb. E_{vdW}, ovvero il termine legato alle Forze di Van der Waals, viene invece calcolato utilizzando comunemente

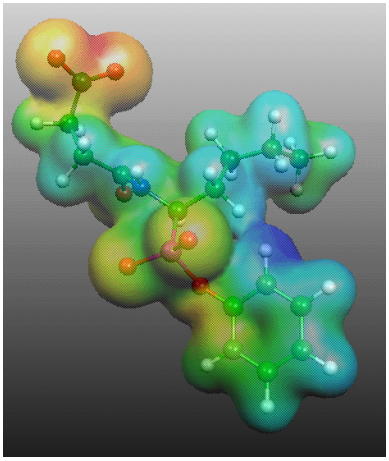

Figura 2.1: Visualizzazione del potenziale molecolare

il potenziale di Lennard-Jones.
In figura 2.1 è mostrato un esempio di visualizzazione del potenziale di una molecola.

2.1.1 Potenziale di Coulomb

La forza di Coulomb, descritta dalla legge di Coulomb, è l'interazione presente tra due corpi elettricamente carichi.

$$F = k\frac{|q_1||q_2|}{d^2} \tag{2.4}$$

La legge esprime quantitativamente l'interazione tra due cariche elettriche puntiformi, q_1 e q_2, a distanza d e ferme nel vuoto.
La costante di Coulomb k è pari a:

$$k = \frac{1}{4\pi\varepsilon_0} = 8,99 \cdot 10^9 \text{ Nm}^2\text{C}^{-2} \tag{2.5}$$

con ε_0 la costante dielettrica del vuoto.
La legge di Coulomb si è dimostrata non valida per distanze inferiori a 10-11 pm, come quelle che separano i nucleoni di un nucleo atomico, che sono uniti dalla forza di interazione forte.
Per trovare la forza che agisce su una piccola carica di prova q dovuta a un sistema di N cariche ferme si può usare il principio di sovrapposizione:

$$\mathbf{F}(\mathbf{r}) = kq\sum_{i=1}^{N}\frac{q_i(\mathbf{r}-\mathbf{r}_i)}{|\mathbf{r}-\mathbf{r}_i|^3}, \tag{2.6}$$

dove q_i e r_i sono il valore e le posizioni della i-esima carica.

Esistono alcuni fenomeni nell'atomo che sembrano contrastanti con il principio della repulsione fra cariche di segno uguale e l'attrazione fra cariche di segno opposto: fra elettroni e protoni dovrebbe agire una forza attrattiva, mentre l'elettrone ruota ad una certa distanza dal nucleo; tra i protoni agisce una forza repulsiva, che dovrebbe indurre una disintegrazione dello stesso nucleo.
È prevedibile che la forza di Coulomb sia vinta da due forze di segno opposto, superiori di uno o più ordini di grandezza.
La forza nucleare debole agisce fra nucleo e elettroni, e spiega la rotazione degli elettroni e il loro mancato 'collassare' sul nucleo.
La forza nucleare forte opera all'interno del nucleo, ed è più intensa della repulsione elettrostatica fra protoni.
Il campo di potenziale $V(r)$ dovuto al contributo coulombiano si ottiene come derivazione nello spazio del force field generato da $\mathbf{F}(\mathbf{r})$. Può essere scritto matematicamente:

$$V(r) = kq \sum_{i=1}^{N} \frac{q_i}{|\mathbf{r} - \mathbf{r}_i|} \tag{2.7}$$

Esistono vari metodi empirici per il calcolo delle cariche associate agli atomi nell'interazione molecolare, esempi sono quelli ideati da Mulliken e da Löwdin. Questi metodi empirici sono necessari pr evitare di calcolare i contributi dovuti alla nube elettronica attorno ai nuclei, che darebbero vita ad integrali tripli di ardua risoluzione.

2.1.2 Potenziale di lennard Jones

Il potenziale di Lennard-Jones è il più noto e il più usato dei potenziali empirici per descrivere l'interazione interatomica ed intermolecolare.
A distanze interatomiche o intermolecolari molto piccole le densità elettroniche si sovrappongono generando forze repulsive molto intense, caratterizzate da un raggio d'azione molto corto e dal fatto che crescono rapidamente all'avvicinarsi delle molecole. Per esse non esiste un'equazione ricavata teoricamente che le descrive e per tale motivo si rende necessario il fare affidamento su alcune funzioni potenziali empiriche.

La più famosa di esse, che comprende anche la parte attrattiva dovuta all'interazione di van der Waals, è il potenziale proposto nel 1931 da John Lennard-Jones all'Università di Bristol:

$$V(r) = 4\epsilon \cdot \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right] \tag{2.8}$$

dove ϵ è la profondità della buca di potenziale e σ è il diametro della sfera che approssima l'atomo o la molecola in un modello a sfera rigida. Questa espressione del potenziale tiene in considerazione anche l'azione delle forze attrattive di van der Waals.

La forza viene ricavata a partire dall'espressione precedente per il potenziale:

$$\mathbf{F}(r) = -\nabla V(r) = -\frac{d}{dr}V(r)\hat{\mathbf{r}} = 4\epsilon\left(12\,\frac{\sigma^{12}}{r^{13}} - 6\,\frac{\sigma^{6}}{r^{7}}\right)\hat{\mathbf{r}}. \tag{2.9}$$

Il potenziale di Lennard-Jones è il risultato di due termini:

1. la parte che va con la sesta potenza è il contributo attrattivo delle forze di Van der Waals (forze dipolo-dipolo e forze dipolo-dipolo indotto) e prevale a distanze grandi.

2. la parte che va con la potenza di dodici descrive le forze repulsive che si instaurano a corto range fra i nuclei che, a distanze piccole non sono più ben schermati dagli elettroni, e fra gli elettroni stessi, soggetti a una forza repulsiva che si genera quando due o più di essi tendono ad occupare gli stessi numeri quantici, in contrasto al principio di Pauli.

Le forze di Van der Waals hanno un range compreso fra qualche Angstrom e un centinaio di Angstrom, mentre le forze repulsive sopra citate entrano in gioco a distanze minori di qualche Angstrom. L'entità delle forze a lungo raggio è conoscibile a partire dalla teoria di Van der Waals, mentre le forze a corto raggio sono determinate per via empirica.
Il potenziale di Lennard-Jones può essere espresso come:

$$V(r) = \epsilon\left[\left(\frac{r_{min}}{r}\right)^{12} - 2\left(\frac{r_{min}}{r}\right)^{6}\right] \tag{2.10}$$

dove $r_{min} = 2^{1/6}\sigma$ è la posizione del minimo del potenziale, ovvero la distanza alla quale si ha la buca di potenziale.

2.2 Tecnologie aptiche

L'aptica, nome derivato dal greco 'apto' (tocco), è la scienza che studia, nella sua accezione tecnologica, le possibili applicazioni di interazione tattile o kinestetica tra utente umano e computer in ambienti reali, virtuali o telecontrollati.
La percezione aptica è il processo di riconoscimento degli oggetti attraverso

il Tatto; deriva dalla combinazione tra la percezione tattile data dagli oggetti sulla superficie della pelle, grazie alla quale viene letta la conformazione e la rugosità degli oggetti, e la Propriocezione o kinestetica che deriva dalla posizione della mano rispetto all'oggetto e al proprio corpo.
Gibson, in [2], definisce il sistema aptico come la 'sensibilità dell'individuo verso il mondo adiacente al suo corpo'.
Gli studi aptici possono essere divisi in tre categorie:

1. **Aptica umana** : lo studio delle componenti meccaniche, sensoriali, motrici e cognitive che l'essere umano possiede e utilizza per sentire e manipolare l'ambiente circostante attraverso il tatto.

2. **Aptica delle macchine** : lo studio e la produzione di device hardware che permettono l'interazione tattile tra utente e computer.

3. **Aptica computerizzata** : lo studio dei metodi computazionali e degli algoritmi di rendering aptico per creare effetti desiderati con un interfaccia aptica.

Dai fondamenti teorici stabiliti dall'aptica umana emerge la necessità di sviluppare due tipologie di device studiati per trasmettere sensazioni sia tattili che kinestetiche. Per ogni tipologia di device e per ogni sensazione trasmissibile, inoltre, è necessario progettare l'adeguato metodo computazionale per la simulazione realistica dello stesso. Le tre branche dell'aptica sono quindi strettamente legate l'una all'altra.

2.2.1 Esempi di device aptici

Dopo i primi anni 90 molti nuovi device sono stati progettati e sviluppati all'interno di università e laboratori di ricerca, nonché, successivamente all'interno di aziende private. Laboratorio molto attivo in questo campo è il laboratorio di ricerca del Dr. Hiroo Iwata, della University of Tsukuba, da cui provengono molti sistemi e device aptici. Uno di questi è il sistema FEELEX[6], figura 2.2, composto da uno schermo deformabile controllato da un array di attuatori. Il sistema permette la riproduzione della forma di una superficie di un oggetto virtuale e la simulazione del comportamento del suo materiale al contatto, garantendo all'utente la possibilità di modificare la superficie deformandola direttamente sullo schermo aptico.
Altro sistema proveniente dal medesimo laboratorio è HapticMaster[7], in figura 2.3, un sistema desktop aptico in grado di trasmettere all'utente informazioni come rigidità o peso di oggetti virtuali tramite un manipolatore a 6 gradi di libertà.

Figura 2.2: Feelex

Figura 2.3: HapticMaster

Figura 2.4: Device aptico a levitazione magnetica

Innovazioni tecnologiche per quanto riguarda gli attuatori utilizzati nei device aptici arrivano dalla Carnegie Mellon University [14], in cui sono stati sviluppati sistemi con attuatori basati su levitazione per forza magnetica di Lorentz, raggiungendo elevati livelli di larghezza di banda e sensitività e rendendo i meccanismi più leggeri e di più facile utilizzo per la mancanza di cavi, connettori e pesanti meccanismi, in figura 2.4 è presentato un esempio di device aptico magnetico. La ricerca aptica si è sviluppata anche in campo militare. Un'altra tipologia di device aptici, infatti, è stata studiata e costruita nell'Armstrong Laboratory dell'aviazione degli Stati Uniti, utilizzando come attuatore uno Shape Memory Alloy (SMA), materiale che può cambiare la sua forma e riprenderla successivamente. I sistemi studiati che comprendono l'utilizzo di SMA sono diversi, tra cui di rilevante importanza è il sistema HAPTAC[9], che fornisce ad utenti non vedenti informazioni in due dimensioni generate da calcolatore.
Dal punto di vista privato la SensAble, in collaborazione con il Touch Laboratory, uno dei centri di ricerca del MIT, ha progettato il primo dei device della serie PHANToM. I device PHANToM sono device aptici desk-based che supportano fino a 6 gradi di libertà in grado di fornire force feedback all'utente, figura 2.5. I PHANToM oggi rappresentano i device aptici commerciali più popolari, e grazie alla loro versatilità, funzionalità e affidabilità vengono utilizzati in una grande varietà di campi, dal design 3D ad applicazioni mediche e odontoiatriche.

Figura 2.5: Phantom a 6 DOF prodotto da Sensable

La Immersion Corporation in partenza si è invece dedicata principalmente a sistemi esoscheletrici per la ricerca aptica, sviluppando molti sistemi commerciali come CyberGrasp, un guanto che permette all'utente di sentire un force feedback realistico da oggetti simulati a computer, CyberGlove, un sistema di motion-capture in grado di digitalizzare in tempo reale i movimenti della mano e delle dita, e CyberForce, sistema che permette la gestione di force feedback su tutta la mano e il braccio, fornendo anche un sistema di tracking posizionale a 6 gradi di libertà, figura 2.6.

Per applicazioni mediche telecontrollate leader nel settore è la Intuitive Surgical Inc., produttrice del 'DaVinci Surgical System',figura 2.7 uno dei più diffusi sistemi di chirugia assitita da calcolatore.

2.2.2 Strumenti software per applicazioni aptiche

Ogni tipologia di software che fa uso di un interfaccia aptica è solitamente formato da diverse fasi:

1. rilevamento della posizione, e possibilmente rotazione, dello strumento di interazione;
2. collision detection con gli oggetti virtuali presenti nella scena 3d;
3. calcolo della forza di risposta da esercitare sull'end effector del device aptico;
4. invio della forza calcolata sullo strumento stesso.

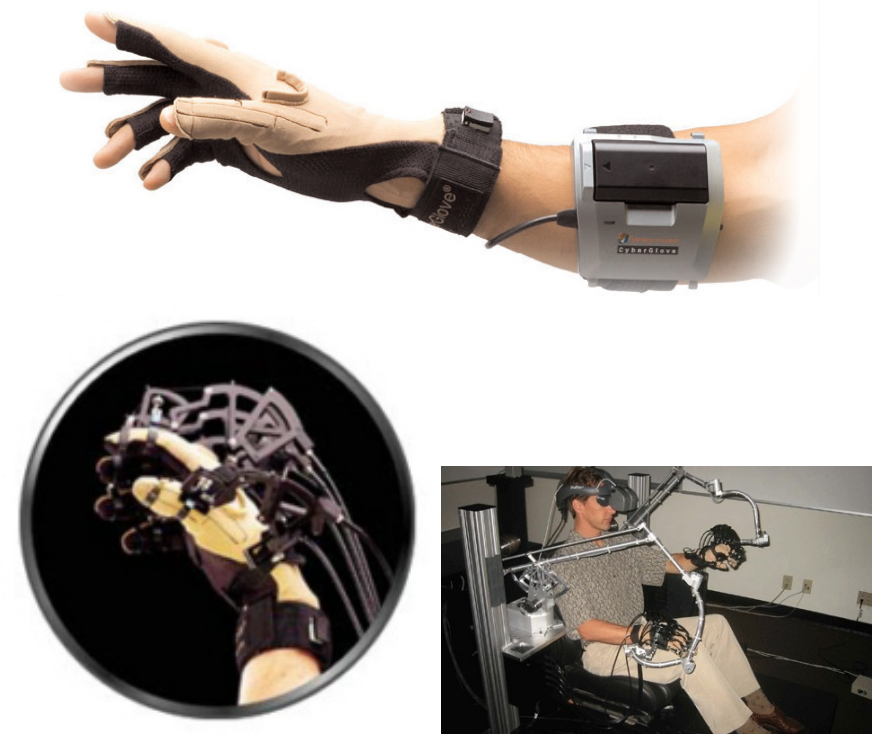

Figura 2.6: Device aptici prodotti da Immersion

Figura 2.7: DaVinci Surgical System

Negli anni le tecniche per effettuare e calcolare queste diverse fasi si sono molto affinate. Ad esempio per la fase di collision detection sono stati presentati diversi algoritmi:

- algoritmo **Vector Field**: prevede la mappatura di ogni punto interno del volume di un oggetto renderizzato ad una forza di risposta, che viene ripresa ed inviata al device in base alla sua posizione durante la collisione con l'oggetto. L'algoritmo si adatta bene solo per la renderizzazione di forme semplici, in quanto la mappatura punto per punto della forza di risposta è piuttosto onerosa.

- algoritmo **God-Object**: utilizza un punto virtuale di contatto (proxy) impossibilitato a penetrare la superfice dell'oggetto renderizzato per il calcolo della forza da inviare allo strumento, proporzionale alla distanza tra proxy e posizione reale de device aptico.

- algoritmo **Virtual Proxy**: estensione del precedente, risolve il problema della determinazione della parte di superficie dell'oggetto rappresentato effettivamente a contatto con il device aptico.

Una raccolta di algoritmi per la determinazione della collision detection è stata proposta in [10] all'interno del framework H-Collide, che comprende gli algoritmi di Spatial Decomposition, Bounding Volume Hierarchy basato su OB-B-Trees e Frame-to-Frame Coherence.
Oltre a singoli algoritmi di renderizzazione per le fasi di un ciclo aptico, ad oggi sono state sviluppate diverse librerie software, come ad esempio Haptik Library [11], Computer Haptics & Active Interfaces (CHAI 3D)[12] o H3D API[8]. Tutte le librerie elencate sono offerte con licenza GNU GPL e forniscono un livello di astrazione hardware per essere utilizzate con più device aptici.
Un altro importante tentativo di standardizzazione per l'aptica computerizzata è l'Haptic Application Meta Language [13]; basato su XML, permette l'utilizzo e la comunicazione di diverse tipologie di device aptici nello stesso ambiente, evitando problemi di compatibilità.

2.3 Simulatori aptici per la didattica della chimica

In letteratura esistono parecchi lavori che trattano l'associazione di strumenti aptici a simulazioni: la maggior parte è orientata alla ricerca; solamente pochi casi sono stati progettati con finalità didattiche.
[3] fornisce una rassegna sulle tecniche di rendering aptico: la maggior parte

Figura 2.8: Chemical Force Feedback

degli algoritmi e le applicazioni si concentrano sulla rappresentazione di feedback tattili dovuti a contatti con superfici, piuttosto che sulla rappresentazione di forze distribuite in tutto lo spazio. Nel campo della visualizzazione di dati applicata al settore chimico e della ricerca scientifica, l'ambiente proposto si situa al confine tra la ormai classica chimica computazionale, che privilegia la gestione e la visualizzazione di grosse quantità di dati sottoforma di valori numerici a discapito dell'usabilità o dell'immediatezza nella comprensione, ed il recente settore della bioinformatica, che introduce una migliore gestione delle interfacce utente per la visualizzazione dei risultati. La maggior parte degli strumenti sviluppati in ambito chimico, presentano l'interazione aptica come funzione collaterale, dovuta solitamente ad aggiunte successive di tale funzionalità a strumenti già esistenti in precedenza. Esempi di tali strumenti includono VMD, PyMOL, e altri ([3]), che offrono una vasta gamma di rappresentazioni grafiche, ma una non sempre semplice interpretazione dei dati, soprattutto nel caso di studenti. Uno strumento più simile a quello qui proposto è [4], figura 2.8: tuttavia, esso è stato progettato basandosi su modelli teorici e pertanto non permette di utilizzare o di esplorare dati ottenuti da un'esperienza più diretta, di ricerca.

Dal punto di vista didattico, esistono alcune opere che coinvolgono tecnologie aptiche e la rappresentazione di molecole. In [16], figura 2.9, viene

Figura 2.9: Sistema a realtà aumentata per la didattica della chimica

presentata un'interfaccia utente fisica per la didattica della chimica. Tale sistema utilizza un dispositivo specifico, sviluppato in proprio dagli stessi autori del lavoro basato sul concetto di realtà aumentata.

[5] propone un lavoro simile al presente, ma che comunque soffre di mancanza di generalità, modellando solo l'interazione tra due molecole d'acqua. Interfacce multi-modali sono state analogamente proposte in letteratura. [15] presenta un sistema multi-modale su una piattaforma di realtà virtuale per la simulazione e la sperimentazione dell'interazione molecolare che prevede l'interazione tattile con un'interfaccia olfattiva e feedback uditivo. In ogni caso tali soluzioni multi-modali sono piuttosto complesse per essere replicate in ambienti educativi. Il lavoro qui presentato, invece, è scalabile e può essere riprodotto facilmente per essere messo a disposizione nei laboratori delle scuole.

2.4 Aptica e suoni

Nell'ultimo decennio vi è stato grande interesse, nel campo della ricerca, per l'introduzione di un interfacciamento Aptico in applicazioni che fino ad allora erano state puramente visuali e talvolta uditive.
L'inserimento di un nuovo senso nell'interazione uomo macchina permette di garantire all'utente una migliore immersività nell'ambiente virtuale, che coincide, spesso, con un incremento delle prestazioni nel compiere il compito che viene richiesto dall'applicazione. Numerosi articoli mostrano infatti come un accostamento di interfaccia aptica, visuale e uditiva migliorino i tempi di reazione dell'utente e la comprensione dell'ambiente virtuale con il quale si vuole interagire. Ad esempio in [30] sono stati realizzati due esperimenti atti ad investigare l'utilità nell'introduzione di un interfaccia audio in un sistema precedentemente solo visuale, nello specifico:

1. Nel primo sono stati inseriti degli indizi audio di eventi nascosti che avvenivano in un ambiente virtuale. Tale esperimento ha mostrato un notevole incremento di prestazioni rispetto al solo interfacciamento visuale.

2. Nel secondo invece sono stati introdotti suoni atti a dare informazioni all'utente sul matching dei dati mostrati a video rispetto ai dati ricercati nell'ambito della ricerca in un database. Al contrario del precedente tale esperimento non ha mostrato alcun miglioramento.

Quest'ultimo esempio fa emergere come il problema dell'integrazione di varie informazioni sensoriali vada studiato ed approfondito in ogni singolo caso in modo da ottenere un'interfaccia efficace.
Spesso l'utilizzo di interfacce aptico uditive viene associato a situazioni in cui non vi è un diretto e continuativo contatto visivo con il sistema virtuale a causa di disabilità visive o talvolta, più semplicemente, per il modo di utilizzo della tecnologia su cui l'applicazione è in esecuzione. Un esempio di tale tecnologia può essere il telefono cellulare, il cui contesto di utilizzo spesso non ci consente di sfruttare la vista, impegnata in altri compiti. Per ovviare a tali problemi in [31] viene proposto un sistema per sviluppare, dal punto di vista del design, interfacce Aptico-Uditive per smart-phones mutuando gli stessi principi di coordinazione tra interfaccia visuale ed uditiva:

1. **Sincronizzazione**: se l'evento audio compare in contemporanea dell'evento aptico.

2. **Linearizzazione temporale**: sfruttare l'audio per creare il senso del tempo rispetto ad un evento aptico.

3. **Mascheramento**: usare l'audio per nascondere o togliere attenzione a particolari informazioni aptiche.

4. **Sincresia**: l'uso di audio e stimoli aptici per suggerire o creare un senso di illusione, garantito da un certo grado di immersività, in modo da migliorare l'esperienza d'uso del sistema.

In [32] viene studiata l'interazione uomo macchina da parte di persone non vedenti o affette da deficit visivi tramite il riconoscimento di forme virtuali e rispettivi punti salienti. Tale serie di esperimenti ha mostrato come usando un singolo senso un interfacciamento aptico risulti più efficace; se però vi è la contemporanea presenza di indizi audio e aptici, le prestazioni aumentano notevolmente rispetto all'utilizzo del solo tatto.

Anche nell'ambito dell'insegnamento, nel quale può essere inserito il nostro lavoro, sono state svolte valutazioni sul miglioramento delle prestazioni grazie ad un integrazione multimodale di più sensi nell'interfacciamento con il software usato per l'apprendimento. In [33] viene presentato un sistema virtuale per migliorare le capacità motorie necessarie all'esperienza di operazioni chirurgiche; tale sistema trae notevole vantaggio dall'inserimento di uno stimolo sonoro in grado di accompagnare l'utente nello svolgimento del compito. Il grande interesse per le interfacce multisensoriali ha portato via via all'introduzione di molteplici campi di ricerca tra i quali spicca in modo significativo l'aptica multimediale, definita come [34]:
'*the acquisition of spatial, temporal, and physical knowledge of the environment through the human touch sensory system and the integration and/or coordination of this knowledge with other sensory displays such as audio, video, and text in a multimedia system.*'
Nello stesso articolo viene fatto riferimento ad una carrellata di applicazioni che sfruttano un interfacciamento aptico; di notevole interesse per la nostra trattazione è l'esempio portato da [35] in cui vengono accostati stimoli audio e aptici per simulare la sensazione di tatto e percepire le proprietà di un tessuto, quali ruvidità, morbidezza, attrito.
In definitiva tutti gli studi riportati mostrano quanto valore aggiunto possa ricevere un sistema virtuale in grado di garantire un corretto interfacciamento multisensoriale.

2.5 Conclusioni del capitolo

In questo capitolo sono stati introdotti i concetti chimici alla base di quanto descritto nel seguito della tesi ed è stata fornita una panoramica delle tecnologie aptiche presenti in letteratura.
Sono stati introdotti inoltre vari lavori in cui veniva utilizzata un'interfaccia sonora in concomitanza con l'interfacciamento aptico.
Nel prossimo capitolo saranno introdotte le tecnologie effettivamente utlizzate per questo progetto di tesi.

Capitolo 3

Impostazione del problema di ricerca

In questo capitolo vengono esplicitate le motivazioni alla base del progetto, nonché la sua struttura sia dal punto di vista hardware che software, focalizzando l'attezione sugli strumenti utilizzati.
Il progetto nasce dall'esigenza di offrire uno strumento per analizzare e percepire il campo di potenziale attorno ad una molecola sia a livello didattico che di ricerca.
Nello specifico, il sistema progettato consiste nell' espandere con nuove funzionalità e rendere maggiormente efficace uno strumento già esistente in una sua fase embrionale, [1].
Oltre a correggere alcuni errori presenti si è provveduto, infatti, ad inserire come detto nuove funzionalità sia grafiche che aptiche nonché ad introdurre una nuova dimensione sensoriale sonora. Inoltre è stato aggiunto un aspetto dinamico del sistema, in grado di far muovere le molecole in seguito a forze erogate attraverso lo strumento aptico.
Procedendo con ordine esaminiamo in dettaglio nelle prossime sezioni: prima i motivi che hanno portato alla creazione del progetto, poi una breve descrizione del comportamento del sistema nel suo complesso, infine gli strumenti utilizzati.

3.1 Motivazioni e scopo del progetto

Come già accennato il progetto sorge in primis dalla necessità di facilitare la comprensione della natura delle molecole nell'ambito della meccanica molecolare agli studenti universitari di chimica. Infatti discutere di atomi

e molecole, e quindi insegnare, diventa particolarmente ostico per l'intangibilità della materia stessa: non si può vedere una molecola singola o toccarla; rimane sempre tutto su un piano astratto quando si parla di legami e di forze molecolari e ciò non facilita di certo la comprensione.
Il sistema realizzato permette di osservare e toccare una molecola virtuale sotto varie forme e convenzioni, il cui campo potenziale è calcolato sia sperimentalmente che tramite modelli teorici. Ciò permette allo studente innanzitutto di creare delle metafore mentali che lo aiutino a comprendere e assimilare i concetti basilari dell'argomento, dopodiché di discriminare sulla bontà di una certa approssimazione teorica del potenziale in virtù dei dati sperimentali.

Quanto detto finora si basa sulla considerazione che la fisica, e quindi la chimica, divengano di più semplice comprensione se alle formule e alle definizioni vengono associate delle metafore d'uso comune che il sistema realizzato non impone all'utente, lasciando quest'ultimo libero di definirle sulla base delle sensazioni date dallo strumento stesso.
L'uso di interfacce aptiche nello studio della chimica non è nuovo: già vari strumenti di ricerca ne fanno uso, ciò poichè la percezione tattile di qualcosa di intangibile come le forze molecolari può permettere di sviluppare nuove idee nella mente dei ricercatori, altrimenti sopite nell'infinità di numeri coinvolti. Nel nostro caso il software si pone altresì come strumento di ricerca in grado di mostrare e far percepire il potenziale elettrico molecolare attorno a qualsiasi molecola, visualizzandolo non come matrice tridimensionale di numeri (forma in cui è immagazzinato dai dati sperimentali) ma come colori della rappresentazione grafica della molecola o come forze a cui la mano dell'utente è sottoposta.

3.2 Funzionamento globale del sistema

Il sistema propone in primis un'interaccia grafica caratterizzata da una serie di opzioni, da cui è possibile caricare una molecola sotto forma di file pdb, mep e inp che verrà poi visualizzata nella finestra principale. A questo punto selezionando le rispettive opzioni si può visualizzare o sentire attraverso l'interfaccia aptica la geometria del sistema o campo di forza attorno alla molecola; a questi feedback visivi e aptici si associa anche un ulteriore ritorno sonoro dall'applicazione.
L'interazione aptica avviene muovendo un cursore virtuale nello spazio tridimensionale associato al movimento dell'end-effector del braccio meccanico

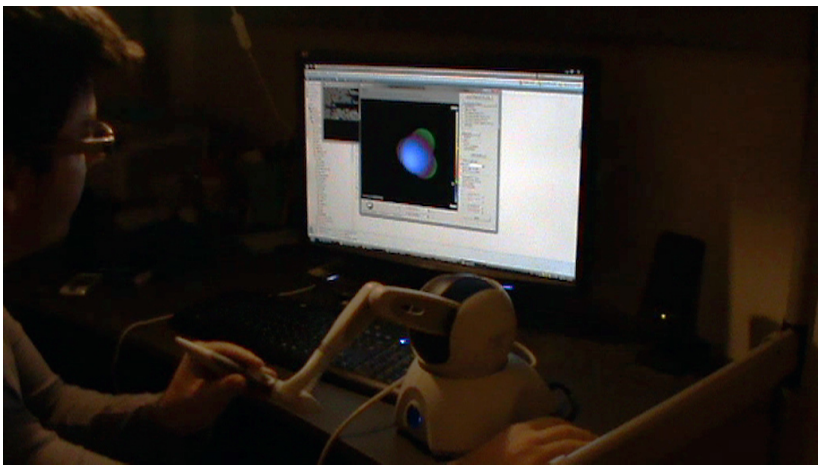

Figura 3.1: Il sistema all'opera

che contraddistingue lo strumento aptico. Il braccio meccanico oppone una certa resistenza o guida la mano in un certo punto dello spazio in base alla modalità di interazione selezionata.
L'uso dello strumento tattile è coadiuvato da numerose opzioni grafiche che permettono di visualizzare la molecola nel modo più appropriato al compito che si vuole svolgere, si può ad esempio modificare il raggio dei singoli atomi a piacimento mostrando differenti porzioni del potenziale, oppure visualizzare le superfici isopotenziali o ancora le linee di forza.
Anche l'interazione sonora permette di accompagnare l'esplorazione aptica producendo suoni proporzionali o alla posizione o al potenziale del punto spaziale in cui si trova il cursore; inoltre sno presenti suoni più semplici di interfaccia o eventi di lettura automatica che riferiscono il nome dell'atomo selezionato.
Infine è stata introdotto un sistema di fisica dinamica in modo da dare l'impressione all'utente, nell'ambito dell'interazione aptica con la geometria della molecola, di spingere la stessa in una direzione voluta, dando un senso di realismo maggiore all'oggetto virtuale rappresentato.
Nel complesso lo strumento propone un approccio multisensoriale alla simulazione meccanica di una molecola, fornendo numerose opzioni per regolare al meglio l'interazione con l'utente.

3.3 Strumenti hardware utilizzati

L'applicazione fa uso, oltre che di un normale Pc, anche di uno strumento chiamato Phantom OMNI capace di fornire un'interfaccia aptica al sistema.

Per informazioni dettagliate si veda [21].
Il PHANToM viene prodotto dall'azienda Sensable e si presenta come un braccio meccanico a 6 gradi di libertà, 3 per la traslazione e 3 per la rotazione del pennino agganciato all'end effector, figura 3.2. I gradi di libertà translazionali sono in grado di generare autonomamente una forza sull'end effector, in modo da guidare la mano dell'utente che tiene il pennino nella direzione desiderata dall'applicazione. Al contrario i 3 DOF rotazionali non sono in grado di generare torsioni nel modello per questa tesi utilizzato benchè il sistema sia in grado di rilevare la rotazione dell'end effector.

Figura 3.2: PHANToM Omni : strumento aptico

Tramite lo strumento PHANToM è quindi possibile individuare la posizione e la rotazione dell'end effector nello spazio, nonché generare una forza verso una direzione. Tale sistema permette tramite l'opportuna programmazione del comportamento di generare vari tipi di sensazioni aptiche. Per gestire lo strumento, Sensable mette a disposizione due diverse librerie, una di basso livello HDAPI e una di più alto livello HLAPI; di tali librerie parleremo più avanti nella trattazione.
Per comprendere cosa è possibile fare con lo strumento andiamo a fare qualche esempio di sensazioni aptiche generabili:

- una prima possibilità è renderizzare apticamente una superficie geo-

metrica: lo strumento non permetterà al cursore virtuale di penetrare in forme solide presenti nello spazio, generando una forza di repulsione, una volta avvenuto il contatto, proporzionale alla profondità raggiunta dalla reale posizione del cursore rispetto alla superficie dell'oggetto e alla 'durezza' desiderata dell'oggetto considerato. Nello specifico, non è opportuno generare una forza di repulsione massima al contatto con l'oggetto, poiché in tal modo si produrrebbero degli sbalzi di forza sull'end effector; al contrario la resistenza fornita dall'oggetto deve essere graduale man mano che si 'penetra' nella superficie dello stesso.

- altra possibilità è quella di vincolare lo strumento a seguire delle linee o superfici nello spazio, in modo che, una volta raggiunta una distanza limite dall'oggetto geometrico considerato, il cursore, e quindi l'end effector, sia vincolato a permanere sulla geometria. In tal modo è possibile ad esempio vincolare l'utente a muovere la mano in un percorso prestrutturato stabilito da una curva nello spazio, che permetta di simulare un cammino in un ambiente virtuale.

- ulteriore utilizzo è la generazione di campi di forza secondo leggi stabilite dal programmatore. Si può far si che l'end-effector sia respinto da talune entità e astratto da altre; si possono così simulare dei sistemi di cariche elettriche, oppure di masse gravitazionali e via dicendo.

- infine, nell'ultimo caso che portiamo ad esempio, vi è la possibilità di utilizzare lo strumento per controllare un oggetto virtuale, in modo da gestirne la posizione e l'orientamento muovendo il pennino nello spazio; in tal modo si può simulare l'azione tra l'oggetto 'avatar' e il resto del mondo, con annesse forze necessarie.

Nel complesso all'interno del presente progetto si vedranno vari utilizzi dell'interfaccia aptica quali illustrati poc'anzi.

3.4 Librerie utilizzate

L'applicazione risulta, per la natura dei vari aspetti sensoriali, suddivisa concettualmente in alcuni settori ben distinti detti engine. Si possono individuare cinque diversi engine ed in aggiunta una struttura dati per la gestione dell'entità chimica molecola:

1. **Engine Grafico**: consiste nella rappresentazione della finestra 3d del software in cui vengono renderizzate tutte le forme geometriche necessarie;

2. **Engine per l'Interfaccia Utente**: si occupa di renderizzare e gestire le opzioni presenti a lato della finestra di renderizzazione 3d;
3. **Engine Sonoro**: gestisce l'intera interazione sonora del sistema, dagli effetti nell'interfaccia utente all'acusmetria;
4. **Engine Aptico**: interfaccia di gestione del PHANToM con il quale renderizzare tutti gli effetti aptici;
5. **Engine Dinamico**:questo engine si occupa di mantenere in memoria la fisicità degli oggetti 3d rappresentati e di fargli assumere un comportamento naturale nel tempo in base alle forze assegnate a tali oggetti;
6. **Sistema Chimico**: permette di gestire le entità chimiche presenti all'interno del sistema con rispettive proprietà.

Tali engine fanno uso di diverse librerie nel loro funzionamento, afferenti alle diverse aree tematiche:

- Engine Grafico
  OpenGL
  GLUT
- Engine GUI
  GLUI
- Engine Sonoro
  OpenAL
  Alut
  SpeechAPI
- Engine Aptico
  OpenHaptics(suddiviso in:)
  HapticDeviceAPI
  HapticLibraryAPI
- Engine Dinamico
  OpenDynamicsEngine
- Engine Chimico
  OpenBabel

Nelle prossime sezioni verrà descritto il principale funzionamento delle librerie utilizzate.

3.4.1 Engine Grafico

Per lo sviluppo della grafica del sistema è stata utilizzata la libreria OpenGL (Open Graphics Library) [17], API di primaria importanza per lo sviluppo di applicazioni 3d realtime. Tale libreria consiste in un insieme di direttive in grado di impartire ordini diretti alla scheda video presente nei sistemi desktop; in tal modo è possibile, attraverso la renderizzazione di semplici strutture geometriche, renderizzare qualsivoglia struttura virtuale in grafica 3d.
In figura 3.3 è visualizzato il logo del progetto OpenGL.

Figura 3.3: Logo OpenGL

OpenGL permette di definire:

- un osservatore, rappresentato da una camera con una determinata trasformazione prospettica(o ortogonale);
- un sistema di illuminazione definito da un massimo di 8 fonti di luce;
- un insieme di vertici nello spazio collegati secondo vari schemi: triangoli,quadrati, poligoni, linee etc.;
- un sistema di gestione di matrici di trasformazioni nello spazio, quali rotazioni e traslazioni;
- un sistema di shading consistente nell'assegnare ad ogni vertice un colore o materiale e una direzione normale alla superficie;
- varie ed eventuali come ad esempio trasparenze, effetti di nebbia etc.

In aggiunta a tale potente strumento è stata utilizzata anche la libreria GLUT (Graphics Library Utility Toolkit) [18], in grado di gestire sia il sistema di finestre di Windows che i possibili input nel sistema attraverso

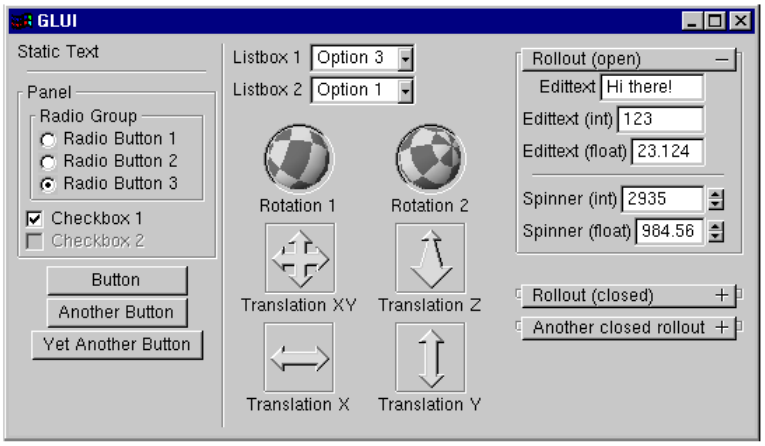

Figura 3.4: Panoramica strumenti GLUI

mouse e tastiera. Infatti, attraverso la registrazione di particolari funzioni, chiamate di callback, è possibile associare un comportamento in risposta a eventi generati dalle periferiche standard del sistema.

3.4.2 Engine per l'interfaccia utente

Nello sviluppo dell'applicazione si è reso necessario l'utilizzo di un sistema in grado di mostrare all'utente le opzioni possibili, nonché la presenza di strumenti per gestire le trasformazioni dell'oggetto 3d visualizzato dall'engine grafico. A tale scopo è stata utilizzata la libreria GLUI(Graphics Library User Interface)[19] che permette di definire pulsanti, slider, checkbox e quant'altro di utile ci possa essere nella definizione di un sistema di interfaccia utente per un sistema. Tale API si appoggia su GLUT per la definizione e istanziazione di una finestra ed agisce disegnando degli oggetti nell'interfaccia e associando ad essi una funzione di callback, la quale verrà poi richiamata ogni qualvolta l'utente agirà sull'oggetto.
In figura 3.4 sono illutrati tutti i possibili pulsanti realizzabili tramite GLUI.

Di seguito forniamo una lista dei controlli utili alla nostra trattazione e ne descriviamo brevemente il funzionamento:

- **Static Text**: Stringa di testo inserita staticamente nell'interfaccia;
- **Separator**: linea in rilievo di separazione;
- **Panel**: elemento contenitore di altri oggetti pulsante;
- **Radio Button**: pulsante di scelta univoca tra varie opzioni contenute in un **Radio Group**;

- **Checkbox**: pulsante che definisce un'opzione booleana, quindi o attivata o disattivata;
- **Button**: pulsante da premere per generare un evento;
- **ListBox**: menu a scomparsa indicante varie opzioni in una lista;
- **Rotation**: elemento di rotazione, permette di gestire una matrice di rotazione ruotando una sfera;
- **Translation**: elemento di traslazione contraddistinto da frecce che ne indicano le direzioni di spostamento;
- **Rollout**: pannello che può essere richiuso in modo da rimanere nascosto;
- **Edittext**: porzione di testo editabile dall'utente;
- **Spinner**: elemento utile a trattare valori numerici, float o int, tramite l'inserimento manuale o lo scorrimento graduale dei valori assumibili dall'opzione corrispondente.

Tramite l'uso di questa libreria è possibile perciò ricevere input numerici o booleani dall'utente, in modo da restituire all'utente un effettivo controllo sui parametri del sistema mostrato a video e definire poi delle opzioni che si ripercuotano sullo stesso. In tal modo l'esperienza-utente diventa dparticolarmente personalizzabile a favore di un uso maggiormente proficuo del software.

3.4.3 Engine Sonoro

Scopo dell'engine sonoro all'interno del sistema è quello di fornire stimoli di supporto sonoro all'interazione; in tal modo si evidenziano gli eventi salienti e si permette all'utente di veicolare le informazioni ricevute dall'applicazione. Per far ciò sono state utilizzate OpenAL(Open Audio Library) [24] ed il suo utility toolkit Alut [25], librerie per lo sviluppo di effetti sonori nello spazio 3d. Entrambe le librerie sono al momento mantenute dalla Creative.

OpenAL ricalca molto OpenGL come stile di programmazione e ben si interfaccia con tale API. Attraverso OpenAL è possibile caricare in memoria dei suoni in formato wave, o sintentizzarne di nuovi definendone la forma d'onda. Dopo tale passaggio si definiscono delle sorgenti a cui associare tali suoni, poste nello spazio 3d. A queste sorgenti è possibile definire varie proprietà che ne individuano timbro, ampiezza, posizione (sfruttando l'effetto

Figura 3.5: Logo OpenAL

stereo delle casse) e, compatibilmente con il movimento della sorgente stessa, effetto doppler. A questo punto basta semplicemente azionare una sorgente, al momento opportunamente definito dal programmatore, per ascoltare il suono selezionato con le proprietà associate.
Da notare che in OpenAL, come in OpenGL è possibile definire la posizione dell'ascoltatore, in modo da permettere al sistema di calcolare i parametri del suono da riprodurre secondo la posizione di quest'ultimo rispetto alla sorgente; ciò permette di introdurre un concetto di spazialità agli effetti sonori.
In aggiunta alla libreria illutrata è stata utilizzata la SAPI (Speech Application Programming Interface) di Windows [26] per quanto riguarda la sintesi vocale di informazioni testuali. Tale Api permette, oltre ad altre funzionalità quali il riconoscimento vocale, la possibilità di far leggere ad una voce automatica delle stringhe, funzionalità usata all'interno dell'applicativo. Per utilizzare tale libreria basta configurare i parametri della voce che si intende utilizzare e passare al sintetizzatore vocale una stringa che si vuole venga letta.

3.4.4 Engine Aptico

Per l'interfacciamento aptico del sistema con lo strumento Phantom Omni è stata utilizzata la relativa API, OpenHaptics [22] sviluppata sempre dalla SensAble.
Nello specifico la libreria consiste in due pardi distinte:

- **HDAPI** (Haptic Device API): libreria di basso livello che permette al programmatore di interfacciarsi direttamente col phantom impostando il valore e la direzione del vettore forza voluto.

- **HLAPI** (Haptic Library API): libreria di alto livello, studiata per essere integrata con le OpenGL, fornisce numerosi strumenti per programmare interazioni aptiche da un punto di vista geometrico.

Per il presente progetto sono state utilizzate entrambe le API: HDAPI per le grandi possibilità di personalizzazione delle sensazioni aptiche che offre, HLAPI invece per la semplicità con cui si riescono a realizzare interazioni più semplici e già collaudate.

HDAPI

Per comprendere il funzionamento della HDAPI bisogna innanzitutto comprendere le differenze di comportamento in merito alla frequenza di aggiornamento del thread tra grafica e aptica. Tale grande differenza deriva dalla diversità con cui i nostri organi di senso riescono a percepire le informazioni. Nonostante i nostri occhi vedano in maniera continua, nel caso di immagini a schermo sono in grado di percepire animazioni fluide a partire da una frequenza di 30 frame al secondo; invece nel caso di esperienza tattile o kinestetica le frequenze in gioco necessarie alla percezione della fluidità del movimento sono molto più alte e partono da un minimo di 1000 Hz. A partire da tale considerazione per ottenere una corretta esperienza sensoriale del sistema è necessario separare i thread che gestiscono grafica e aptica, in modo che vengano aggiornati con frequenze diverse secondo il senso sollecitato.
Con questa premessa andiamo con ordine ad analizzare lo schema di funzionamento della HDAPI, schematizzato nel diagramma in figura 3.6:

1. si inizializza l'ambiente aptico ed il device, in modo che venga attivato dall'applicazione;

2. si attiva la renderizzazione di forze nello strumento;

3. si inizializza lo scheduler che si occuperà di creare e gestire il thread ad alta freqenza (maggiore di 1KHz) e si registra la funzione di callback da richiamare ad ogni ciclo di tale thread;

4. a questo punto siamo all'interno della funzione di callback e inizializziamo il singolo frame aptico;

5. per prima cosa individuiamo la posizione del cursore aptico;

6. si calcola, in base all'effetto voluto, la forza da assegnare al cursore considerando tutti gli elementi in gioco nel mondo virtuale rappresentato;

7. si chiude il frame, inviando la forza da erogare al device e si torna al punto 4;

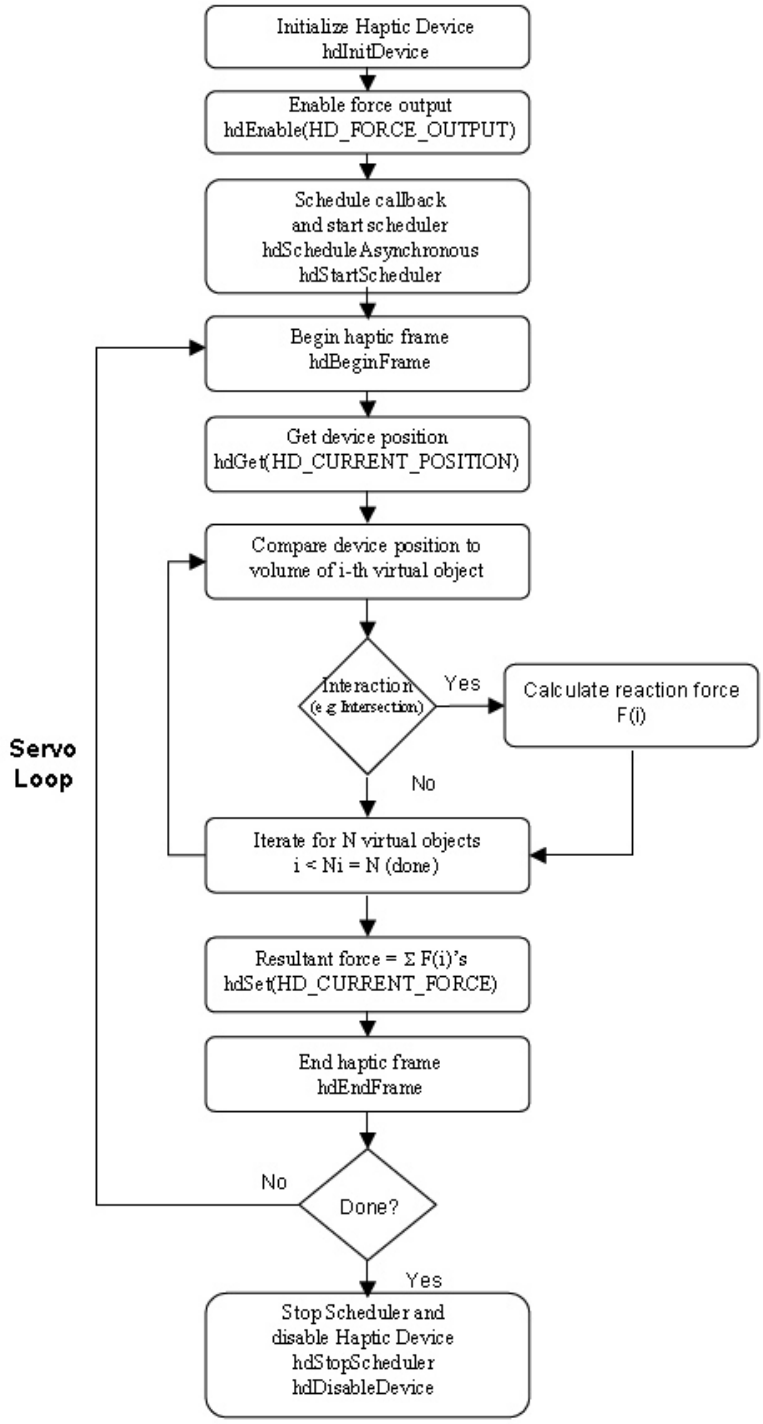

Figura 3.6: funzionamento tipico di HDAPI

8. se si ha finito di usare lo strumento aptico si ferma lo scheduler e si elimina il contesto aptico.

Grossomodo tutte le applicazioni aptiche generate tramite HDAPI funzionano in tale modo. Le grandi differenze si trovano all'interno della funzione di callback che stabilisce la tipologia e l'entità dell'interazione aptica con lo strumento.

HLAPI

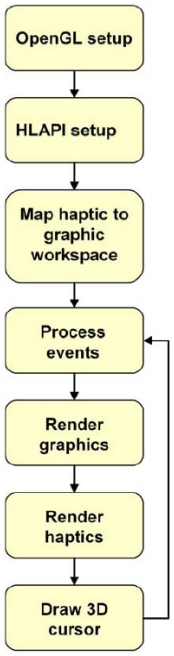

Figura 3.7: funzionamento tipico di HLAPI

La HLAPI lavora in perfetta simbiosi con il sistema grafico generato dalle OpenGL. Se correttamente istanziata la libreria permette di sfruttare la programmazione geometrica gestita dalle OpenGL per renderizzare apticamentee le stesse geometrie. Inoltre a tali geometrie, esattamente come si fa per le proprietà grafiche dei materiali, è possibile associare delle proprietà aptiche, quali morbidezza, attrito e smorzamento. Inoltre possono essere renderizzati anche effetti globali in modo da ottenere una navigazione aptica dello spazio più vincolata.
Infine vi è una complessa interazione con le OpenGL anche per quanto riguarda il calcolo e la renderizzazione del puntatore aptico all'interno della scena 3d. La rappresentazione di tale puntatore infatti risulta necessaria, in

quanto simboleggia l'avatar dell'utente.
Ultima caratteristica della HLAPI utile alla nostra trattazione è la possibilità di gestire specifici eventi all'interno del sistema aptico, quali ad esempio il tocco di un oggetto o la pressione dei pulsanti presenti sul phantom, in tal modo è possibile programmare complesse interazioni con l'ambiente virtuale. In figura 3.7 è rappresentato lo schema di funzionamento di un'applicazione che fa uso di OpenGL e HLAPI; di seguito una breve descrizione dei passi fondamentali:

1. si inizializza l'ambiente grafico;
2. viene inizializzato il contesto aptico;
3. si associa lo spazio di lavoro aptico a quello grafico, mappando correttamente i due spazi 3d;
4. per ogni frame si calcolano i cambiamenti del mondo virtuale;
5. si renderizza graficamente;
6. si renderizza la scena apticamente;
7. si disegna il cursore aptico e si ritorna al punto 4.

Da notare che usando la HLAPI non vi è bisogno di preoccuparsi per il thread ad alta frequenza, mascherato all'interno dell'inizializzazione dell'HLAPI. Infatti le applicazioni che fanno uso di HLAPI fanno uso di tre distinti thread:

- **Client Thread**: thread in cui vengono disegnati gli oggetti, sia da un punto di vista aptico che grafico; in tale thread si inseriscono tutte le funzioni OpenGL e HLAPI ed è l'unico thread effettivamente visibile al programmatore.
- **Servo Thread**: thread che controlla direttamente il device aptico ad alta frequenza (sopra 1 KHz); si occupa continuamente di leggere la posizione e l'orientamento dell'end effector e du aggiornare la forza da erogare sul sistema. Tale thread, simile a quello introdotto tramite le HDAPI è nascosto all'utente.
- **Collision Thread**: thread con una frequenza di circa 100Hz che si occupa di verificare le collisioni tra il cursore aptico e i vari oggetti presenti nella scena. Non è direttamente controllabile dal programmatore, ma è in grado di lanciare eventi rispetto al contatto con un

oggetto e di utilizzare funzioni di callback per il trattamento di tali eventi. Si occupa comunque lui di inviare al **Servo Thread** le informazioni necessarie a renderizzare la forza utile a percepire la geometria degli oggetti disegnati nel **Client Thread**.

Risulta così chiaro come mai per alcuni compiti sia meglio usare una API piuttosto che un'altra: teoricamente con HDAPI si potrebbe fare tutto, ma lo sviluppo di ogni tipologia di effetto aptico è lasciata al programmatore; attraverso HLAPI invece il programmatore è aiutato nella renderizzazione di sensazioni aptiche più banali.

3.4.5 Engine Dinamico

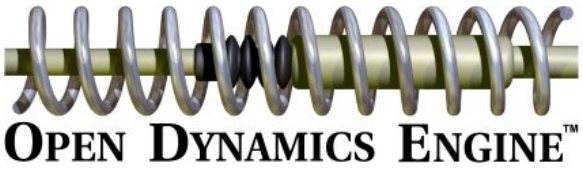

Figura 3.8: Logo di Open Dynamics Engine

Ulteriore parte del sistema è rappresentata dall'interazione dinamica con gli oggetti rappresentati nello spazio 3d. Con dinamica si intende la gestione delle forze a cui gli oggetti all'interno del sistema possono essere sottoposti, in modo da muoversi fisicamente nell'ambiente renderizzato.
Per questa rappresentazione dinamica degli oggetti è stata utilizzata la libreria ODE (Open Dynamics Engine) [23]. Tramite tale strumento è possibile simulare complesse dinamiche del sistema senza doversi occupare dei calcoli integrali sottostanti. Infatti tale libreria permette di definire degli oggetti dal punto di vista fisico, associandoci una massa ed un momento angolare, di localizzarne la posizione e la rotazione all'interno di uno spazio 3d e di permettere l'evoluzione del sistema attraverso l'applicazione di forze che modificano la velocità, e quindi la posizione del corpo.
Per sfruttare ODE basta definire un nuovo mondo e inserire degli oggetti con proprietà fisiche all'interno di esso, dopodichè a ogni iterazione si calcolano gli sviluppi del sistema in base alle forze applicate e alle velocità e si modifica posizione e rotazione dell'oggetto di conseguenza.
ODE permette anche la gestione di giunti e di un sistema di collision detection definendo le geometrie del sistema; tuttavua tali funzionalità non sono state utilizzate nel presente progetto a causa della presenza di un solo oggetto all'interno della scena, al quale vengono applicate delle forze direttamente attraverso il PHANToM.

3.4.6 Sistema Chimico

Per il controllo delle entità chimiche presenti all'interno del sistema è stata utilizzata la libreria OpenBabel che contiene al suo interno tutti gli oggetti, i metodi e le funzioni necessari a recuperare le informazioni chimiche sulle molecole ricercate.

Figura 3.9: logo di OpenBabel

Nello specifico del sistema qui realizzato le librerie OpenBabel vengono impiegate principalmente per due caratteristiche:

- Innanzitutto per una serie di oggetti utili a descrivere le molecole quali OBMol, OBAtom, OBBond e OBElementTable che permettono di gestire rispettivamente l'intera molecola, un singolo atomo, un legame chimico e le proprietà della tavola periodica degli elementi;
- In secondo luogo per la possibilità di costruire gli oggetti di cui sopra in base a dei formati standard per la memorizzazione di molecole; nel nostro caso file in formato pdb.

Il formato PDB (Protein Data Bank) [27] è una rappresentazione creata negli anni '70, nello specifico il Protein Data Bank (PDB) è un archivio per dati strutturali 3D di molecole, specialmente proteine e acidi nucleici. Un file PDB racchiude quindi tutte le informazioni necessarie a collocare nello spazio 3d gli atomi di una molecola e i rispettivi legami. In figura 3.10 è visualizzato un esempio di file in formato pdb rappresentante una molecola di benzene.

Oltre ai file pdb, usati attraverso OpenBabel, sono stati utilizzati altri tre formati di file per la rappresentazione di informazioni chimiche: inp (input), out (output) e mep(molecular electrostatic potential).

I file mep contenengono le informazioni della griglia di potenziale elettrostatico 3D e possono essere generati con MacMolPlt [28], programma

```
REMARK   MSI WebLab Viewer PDB file
REMARK   Created:  Mon Feb 26 16:53:53 ora solare Europa occ. 2001
ATOM      1  C    (NU A   1       1.212  -0.700   0.002  1.00  0.00
ATOM      2  C    (NU A   1       1.212   0.700   0.002  1.00  0.00
ATOM      3  C    (NU A   1       0.000   1.400   0.000  1.00  0.00
ATOM      4  C    (NU A   1      -1.212   0.700  -0.002  1.00  0.00
ATOM      5  C    (NU A   1      -1.212  -0.700  -0.002  1.00  0.00
ATOM      6  C    (NU A   1       0.000  -1.400   0.000  1.00  0.00
ATOM      7  H    (NU A   1       2.078  -1.200   0.003  1.00  0.00
ATOM      8  H    (NU A   1       2.078   1.200   0.003  1.00  0.00
ATOM      9  H    (NU A   1       0.000   2.400   0.000  1.00  0.00
ATOM     10  H    (NU A   1      -2.078   1.200  -0.003  1.00  0.00
ATOM     11  H    (NU A   1      -2.078  -1.200  -0.003  1.00  0.00
ATOM     12  H    (NU A   1       0.000  -2.400   0.000  1.00  0.00
TER
```

Figura 3.10: esempio di file pdb: benzene

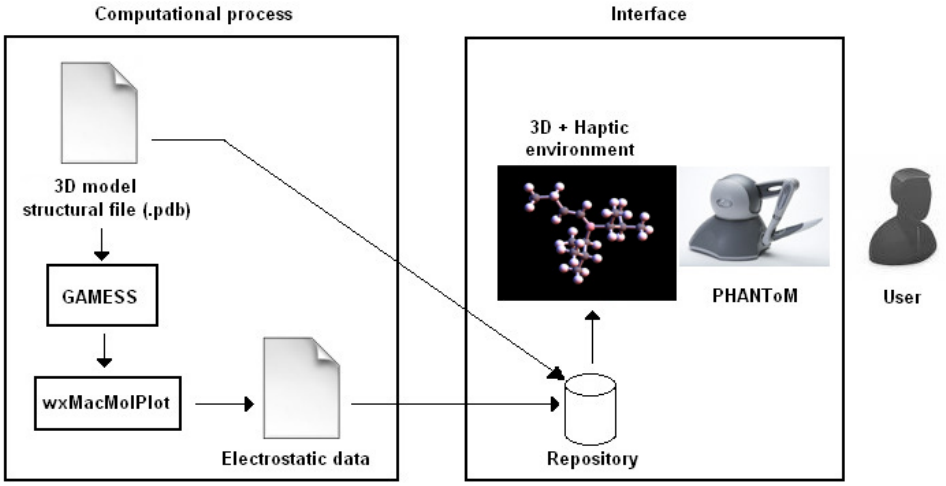

Figura 3.11: schema dei formati esterni utilizzati

freeware che consente la generazione di file dati con informazioni 3D per la renderizzazione di superfici e orbitali molecolari, potenziali elettrostatici, e varie altre informazioni chimiche. Per il calcolo delle mappe elettrostatiche 3D, MacMolPlt necessita di file di ingresso, in formato out, contenenti le informazioni energetiche della molecola analizzata. Questi file possono essere prodotti utilizzando GAMESS [29], programma di calcolo chimico sviluppato dal Gordon research group della Iowa State University rilasciato sotto licenza proprietaria. I file in formato inp rappresentano, infine, l'input con la configurazione per GAMESS.
La struttura di dipendenze tra i vari file che compongono il repository è illutrata nello schema in figura 3.11

3.5 Conclusioni capitolo

In questo capitolo è stata proposta l'intera rosa degli strumenti utilizzati nello sviluppo dell'applicativo qui descritto, le motivazioni che hanno portato alla sua creazione nonché una breve descrizione del funzionamento globale del sistema.
Nel prossimo capitolo analizzeremo tutte le scelte progettuali interne al software e la logica sottostante a quanto implementato. Soffermandoci sulla descrizione dell'interfaccia e sul funzionamento dei vari engine già introdotti in questo capitolo.

Capitolo 4

Progetto logico della soluzione del problema

Questo capitolo tratta della logica alla base di quanto sviluppato nell'ambito del presente progetto, analizzando passo passo le componenti in cui è composto.
Si inizierà dalla descrizione del punto di partenza di questa tesi, individuando le caratteristiche già presenti nel software, nonchè i vari difetti presenti. Subito dopo si analizzeranno le nuove funzionalità introdotte e le correzioni al vecchio sistema.
Al momento è opportuno ricordare le componenti del software realizzato, sebbene nella descrizione successiva spesso i limiti non siano così netti, in quanto i vari sistemi lavorano in stretta collaborazione:

- **Engine Grafico**: si occupa del disegno della scena in grafica 3d, nonchè della gestione degli eventi legati a mouse e tastiera.
- **Engine Interfaccia Utente**: disegna e gestisce l'interfaccia utente, modificando le opzioni del sistema.
- **Engine Audio**: gestisce gli effetti sonori presenti nel sistema.
- **Engine Aptico**: si occupa dell'interazione aptica con il sistema.
- **Engine Dinamico**: mantiene la simulazione fisica delle molecole, viste come oggetti solidi, all'interno dell'applicazione.
- **Interazione chimica**: non è un vero e proprio engine, ma racchiude tutte le interazioni con i file contenenti le informazioni molecolari e si interfaccia con le informazioni chimiche fornite dalle librerie utilizzate.

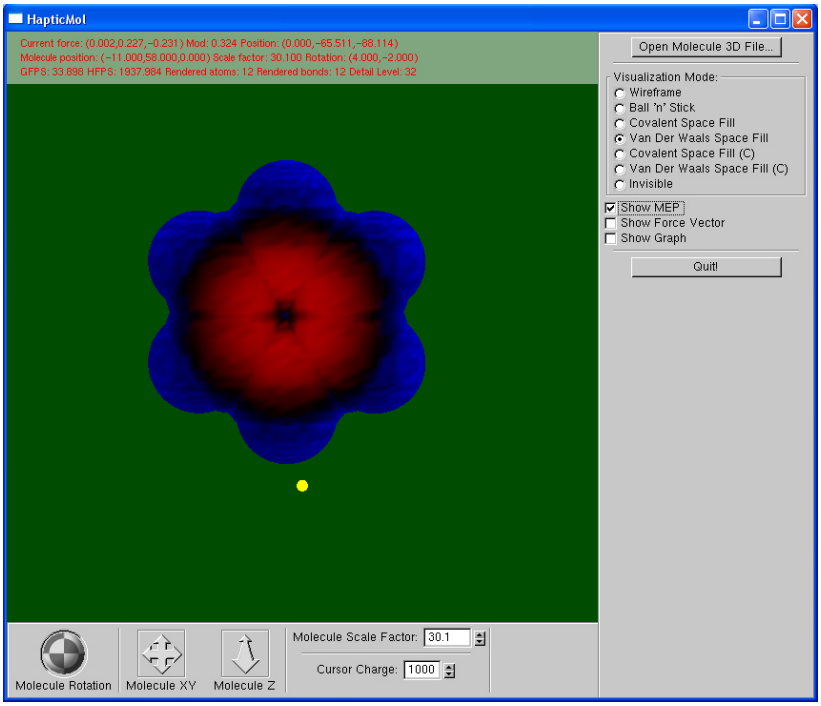

Figura 4.1: screenshot del prototipo di partenza

Nello sviluppo del progetto si è proceduto incrementalmente aggiungendo nuove funzionalità e nuovi engine al sistema presesistente e cercando di mantenerne la stessa struttura a engine. Completamente ex novo sono stati introdotti nell'applicativo gli engine audio e dinamico, e sono stati notevolmente modificati tutti gli altri.
Procedendo con ordine in primo luogo analizziamo il punto di partenza, descritto in [1].

4.1 Prototipo di base

Il sistema inizialmente prevedeva due modalità di visualizzazione e una unica di interazione aptica, usava già un'interfaccia utente della stessa forma di quella attuale, anche se mancante di tante opzioni introdotte e permetteva la rototraslazione della molecola tramite l'uso del mouse. Inoltre era stato inserito un sistema per il calcolo e la gestione del frame rate sia da un punto di vista grafico che aptico.

L'interfaccia utente, mostrata in figura 4.1, presentava la stessa struttura di quella attuale composta da tre componenti:

- una finestra principale contenente la scena 3d OpenGL rappresentata, che consiste in una molecola, se caricata, nel cursore aptico e in una sezione contenente informazioni relative al frame rate grafico, frame rate aptico, alla posizione del cursore aptico e alla posizione della molecola. Tale finestra aveva come colore di sfondo il verde.

- una sottofinestra laterale destra contenente:

  - un pulsante utile a caricare in memoria la molecola. Tale pulsante apriva una finestra di selezione file da cui era possibile individuare un file in formato pdb contenente le informazioni utili ad OpenBabel per creare e gestire la molecola.

  - un radio group da cui era possibile selezionare la tipologia di visualizzazione in merito al raggio atomico. Erano disponibili quattro principali visualizzazioni:

    wireframe, rappresentante la struttura molecolare sotto forma dei suoi legami.

    ball 'n' stick, che rappresenta la struttura della molecola evidenziando gli atomi come sfere di medesimo raggio collegate dai legami.

    Covalent Space fill, in cui gli atomi componenti la molecola vengono rappresentati con il loro raggio covalente. In questa rappresentazione gli atomi risultano tangenti gli uni agli altri.

    Van Der Waals Space fill in quest'ultima rappresentazione vengono usati i raggi di Van Der Waals per la rappresentazione della molecola.

  - una checkbox per variare tra una modalità di visualizzazione e l'altra, nello specifico si tratta di attivare o meno la colorazione della molecola secondo il potenziale elettrostatico molecolare.

  - una checkbox per attivare la visualizzazione del vettore forza sul cursore aptico.

  - una checkbox per visualizzare il grafico di potenziale a partire dal cursore aptico verso il nucleo dell'atomo pià vicino.

  - infine un pulsante quit per terminare il sistema.

- una ulteriore sottofinestra sottostante contenente:

  - un controllo di rotazione della molecola, in realtà non implementato nella versione iniziale.

  - un pulsante di traslazione sugli assi x e y capace di permettere lo spostamento della molecola sul piano dello schermo.
  - un ulteriore pulsante per la traslazione sull'asse z.
  - uno slider che permetteva di gestire lo scaling della molecola (quanto grande dovesse essere rappresentata).
  - uno slider per assegnare un valore alla carica di prova al cursore.

Le due modalità di visualizzazione consistevano in:

- una prima visualizzazione della struttura molecolare sotto forma di sfere illuminate da un'unica luce ambientale e quindi prive di shading. Il colore degli atomi rifletteva la colorazione standard della tipologia atomica individuata tramite OpenBabel.

- una seconda rappresentazione consisteva nella colorazione dell'atomo seguendo la mappa di potenziale: in ogni punto visualizzato l'atomo assumeva una colorazione proporzionale al valore di potenziale associato a quel volume spaziale. I colori passavano dal blu per le aree a potenziale positivo a rosso per il potenziale negativo. Tali valori di colore erano ottenuti moltiplicando per una costante il valore di potenziale individuato, cosichà si era solo in grado di discernere tra aree a potenziale positivo e negativo.

L'unica interazione aptica possibile invece era rappresentata dalla percezione del force field, cioà del gradiente del MEP (Molecular Electrostastic Potential). Questa percezione era ottenuta inviando direttamente al PHANToM la forza presente nella griglia del gradiente, precalcolata, nella posizione individuata dal cursore aptico 3d. Tale gradiente veniva moltiplicato per una costante prima di essere inviato al device, in modo da permettere la percezione delle forze attrattive, lievi rispetto alle forze repulsive. Quest'ultima moltiplicazione effettuava però un clamping, un taglio, delle forze rappresentate in quanto l'utente era in grado di discernere solo tra attrazione e repulsione, ma non tra la reale entità delle forze in gioco.
Il sistema aptico cosà proposto presentava però un grande difetto, al raggiungimento di un minimo di potenziale lo strumento si portava in condizione di instabilità oscillando e vibrando vistosamente. Per questo motivo si era tentato di ammorbidire il cambio di forza sul device mantenendo memoria dello stato di forza precedente. Nello specifico la forza inviata al PHANToM non era pià direttamente la forza individuata sulla griglia dal cursore aptico, bensì una media pesata tra questa e l'ultimo vettore forza erogato. Questa tecnica pultroppo però non permetteva di risolvere il problema, bensà lo

peggiorava, aumentando le oscillazioni dello strumento.
Infine, come già accennato nell'analisi dell'interfaccia, già nella versione iniziale era presente uno strumento di controllo del frame rate che consisteva nel ridurre o aumentare il numero di punti disegnati per ogni atomo a seconda delle prestazioni del computer: se il framerate scendeva sotto i 25 frame al secondo le suddivisioni delle sfere che individuavano gli atomi diminuivano, se salivano oltre i 35 la qualità aumentava aumentando il numero di suddivisioni. L'aumento di suddivisioni permetteva di ottenere sfere più liscie e meno segmentate, ma ovviamente richiedeva maggiori prestazioni da parte del sistema grafico. Il sistema aptico invece non aveva alcun controllo sulle prestazioni, riuscendo a renderizzare ottimamente le forze senza alcun controllo.
Questo era dunque il punto di partenza di questo lavoro di tesi; tale prototipo è stato quasi completamente rivisitato proponendo non solo nuove funzionalità, ma anche migliorie ai vari aspetti gïï¿$\frac{1}{2}$ presenti. I cambiamenti apportati saranno discussi ampiamente nelle prossime sezioni.

4.2 Engine Interfaccia Utente

L'interfaccia utente, mostrata in figura 4.2, è suddivisa in tre parti: una finestra 3D controllata dall'engine grafico, una sottofinestra laterale destra che contiene le opzioni di visualizzazione e una sottofinestra sottostante che consiste nei controlli di rototraslazione e scaling della molecola.

4.2.1 Sottofinestra opzioni

Iniziando dalla sottofinestra a destra si possono trovare:

- un pulsante recante la stringa *Open Molecule 3D File* che permette di mostrare la finestra di selezione file, figura 4.3, dalla quale è possibile localizzare e aprire un file di tipo pdb, contenente tutte le informazioni sulla geometria della molecola che si vuole mostrare a video.

- un pannello a scomparsa da cui è possibile selezionare la tipologia di visualizzazione che si intende mostrare a video in merito alla lunghezza del raggio degli atomi coinvolti. Olte alle sei visualizzazioni già presenti nel prototipo originario è stato introdotta una ulteriore opzione **Custom** che permette di far decidere all'utente la grandezza dei singoli atomi coinvolti a seguito di un processo di selezione degli stessi. Da notare che è presente anche l'opzione **Invisible** che assegna raggio 0 a tutti gli atomi.

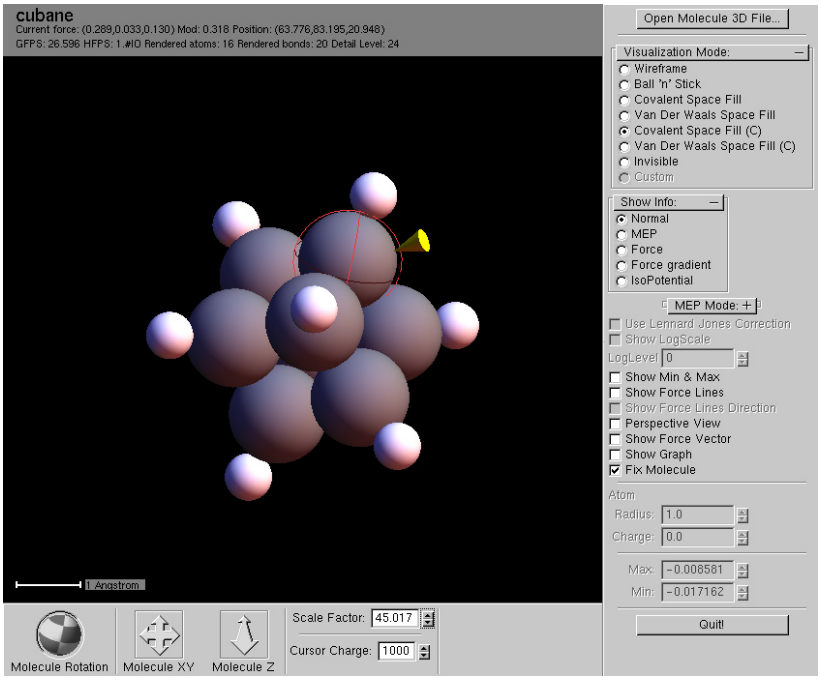

Figura 4.2: interfaccia del sistema

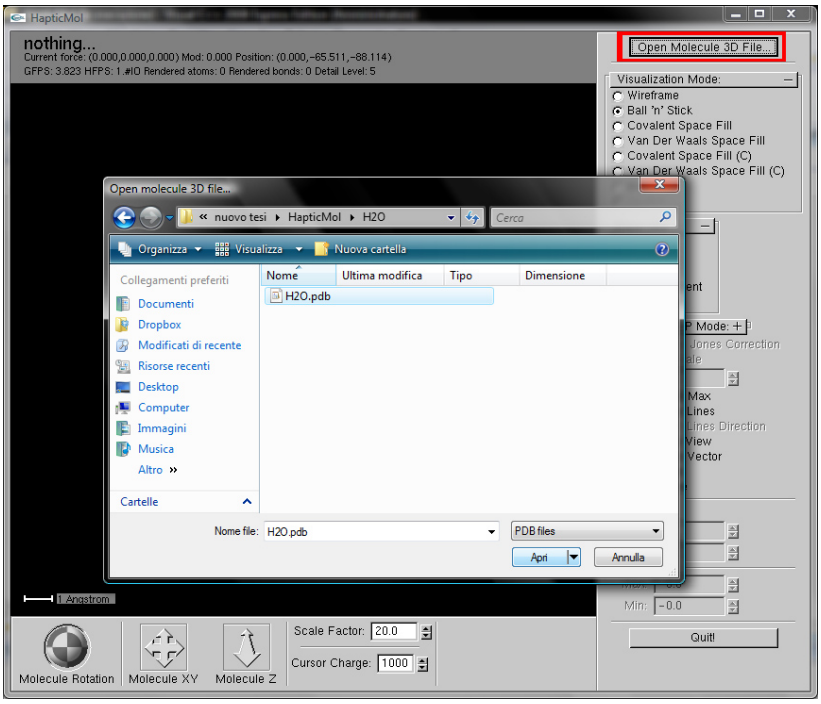

Figura 4.3: finestra di selezione

- un secondo pannello a scomparsa che individua la tipologia di informazione che si intende visualizzare. Può essere:
  - **Normal**: visualizza solo la geometria della molecola sotto forma di sfere colorate uniformemente in base alla tipologia atomica. Questo tipo di visualizzazione è mostrato in figura 4.2.
  - **MEP**: mostra il potenziale sulla superficie della molecola, figura 4.4, sotto forma di una colorazione proporzionale al MEP in ogni punto che definisce le sfere che rappresentano gli atomi.
  - **Force**: mostra la forza sulla superficie della molecola, figura 4.5, sotto forma di tanti piccoli vettori colorati che individuano direzione e entità della forza presente in quel dato punto.
  - **Force Gradient**: visualizza il gradiente della forza, figura 4.6, con le stesse modalità di visualizzazione della forza.
  - **IsoPotential**: mostra i punti di potenziale compresi in un dato intervallo, figura 4.7.
- un terzo pannello a scomparsa, in figura chiuso (visualizzato aperto in figura 4.8), che permette di selezionare le modalità con cui viene calcolato il MEP, puï¿½ essere:
  - **Sperimental grid**: la griglia di potenziale non è calcolata direttamente, ma caricata dalla memoria sotto forma di file mep. Tale griglia consiste in una matrice tridimensionale contenente i valori del potenziale in un determinato volume.
  - **Lowdin charge**: metodo di calcolo del potenziale basato su un sistema coulombiano a cariche; a ogni atomo viene assegnata una carica a partire dalle quali si calcola sia il potenziale che la forza elettrica. Il metodo di Lowdin permette di assegnare opportune cariche agli atomi.
  - **Mulliken charge**: simile al precedente solo che le cariche sono calcolate con il metodo di Mulliken
  - **Lennard Jones**: in realtà si tratta di un metodo per correggere l'errore generato dai metodi a cariche precedenti, però viene anche mostrato da solo in modo da individuarne l'effetto.
  - **Custom charge**: metodo di calcolo a cariche columbiane dove l'utente definisce la carica per ogni singolo atomo.

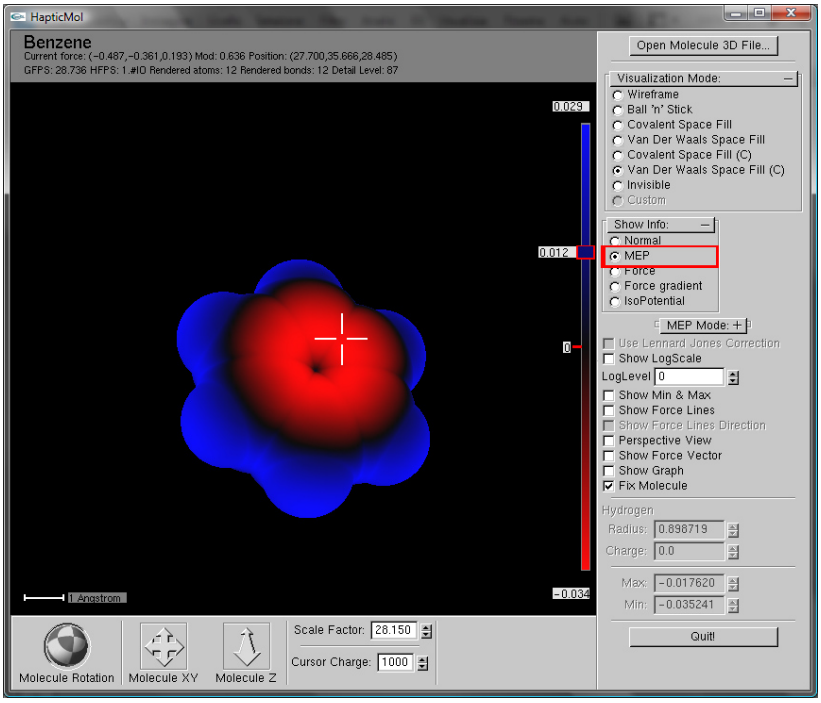

Figura 4.4: visualizzazione potenziale

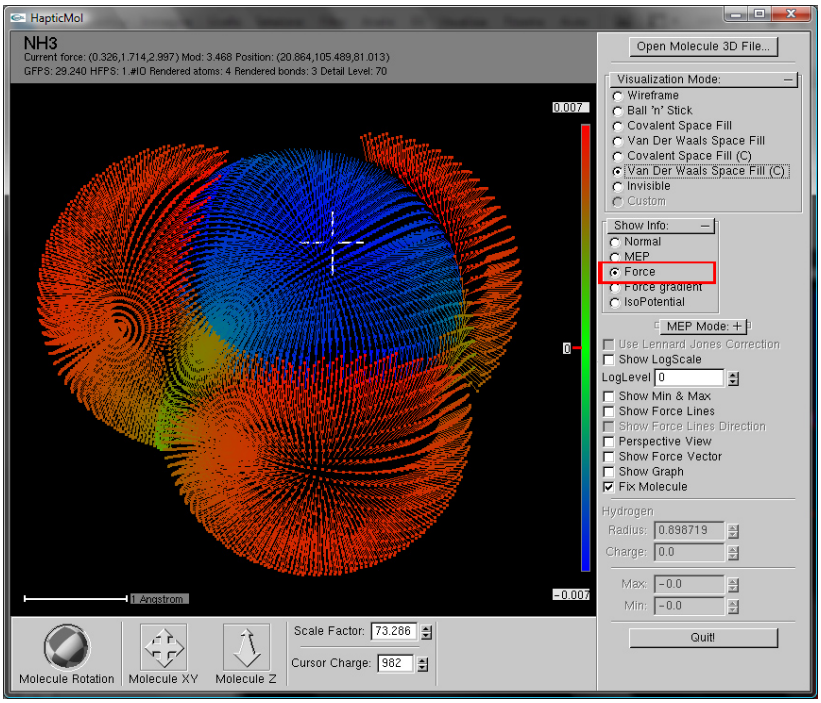

Figura 4.5: visaulizzazione forza

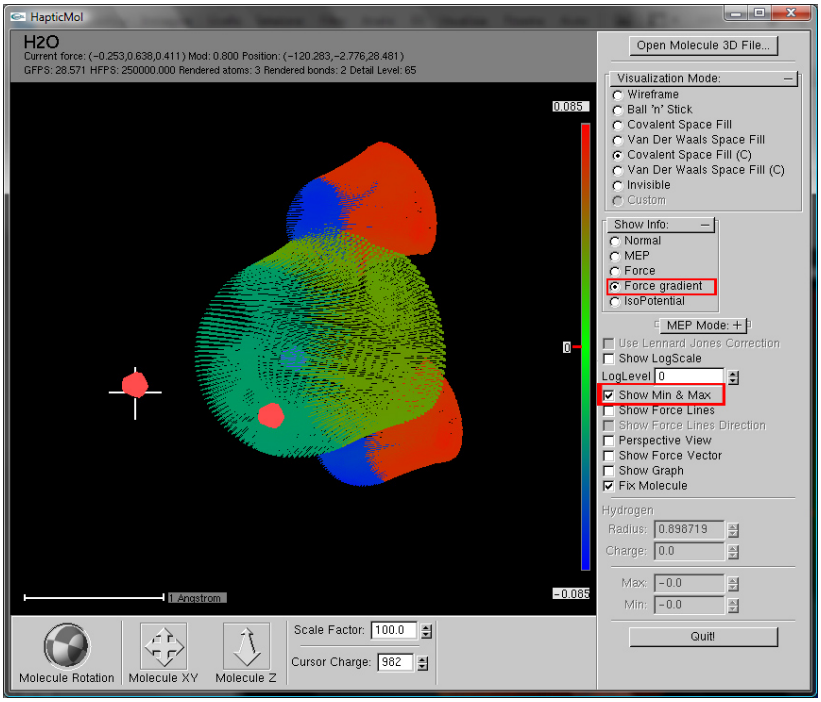

Figura 4.6: visualizzazione gradiente della forza

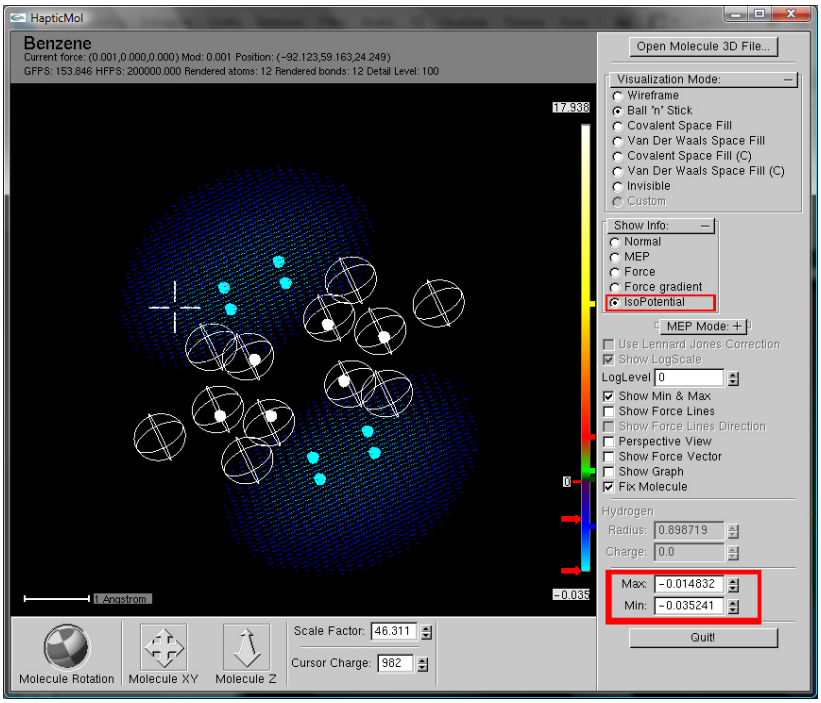

Figura 4.7: volumi isopotenziali

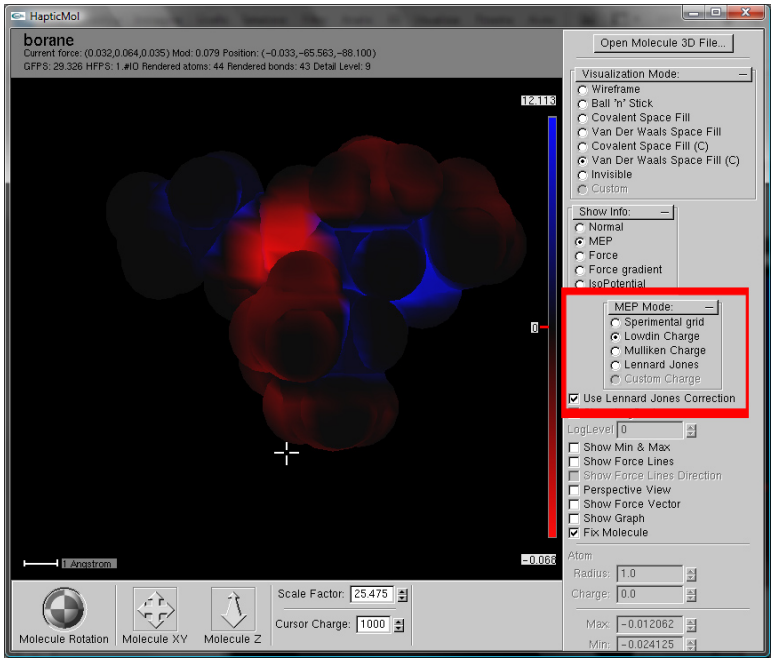

Figura 4.8: altre modalità di calcolo di MEP e forza

- checkbox **Use Lennard Jones Correction** che consente, se attivato, di correggere i sistemi coulombiani precedentemente illustrati con il contributo del potenziale di Lennard Jones.

- checkbox **Show LogScale**: attiva la scala logaritmica che consise di ottenere una colorazione globale del potenziale, figura 4.12.

- slider **LogLevel**: permette di selezionare il livello della scala iper-logaritmica (cioè logaritmo di logaritmo di logaritmo in caso di grado 3). Varia tra 0(lineare) e 200.

- checkbox **Show Min & Max**: permette di visualizzare i punti di minimo e massimo di potenziale come sfererette colorate, vedi figura 4.6.

- checkbox **Show Force Lines**: illustra a schermo le linee di forza del gradiente del potenziale, figura 4.9.

- checkbox **Show Force Lines Direction**: visualizza un vettore di direzione, di colore rosso, sulle linee di forza. Tale indicazione definisce la direzione della linea di forza.

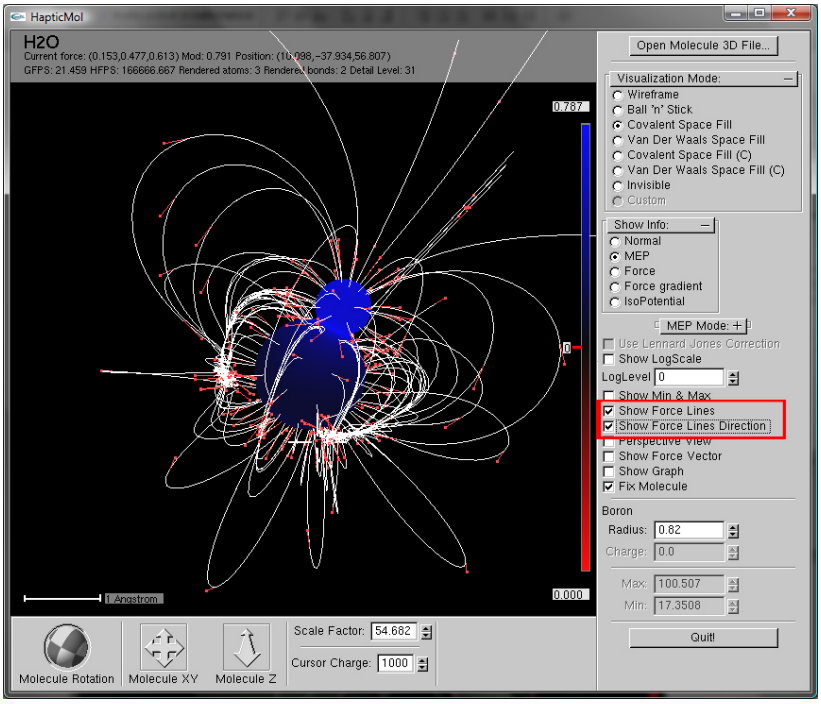

Figura 4.9: visualizzazione linee di forza

- checkbox **Perspective View**: Passa nella modalità di visualizzazione prospettica, figura 4.10
- checkbox **Show Force Vector**: mostra la direzione del vettore forza a cui è sottoposto il cursore aptico in alto a sinistra della scena 3d, figura 4.11.
- checkbox **Show Graph**: mostra il grafico di potenziale tra il cursore e il nucleo dell'atomo più vicino.
- checkbox **Fix Molecule**: permette di bloccare, e quindi di sbloccare, la dinamicità della molecola, cioè la possibilità di muoversi attraverso le forze erogate tramite il PHANToM. Ad esempio permette di far muovere la molecola con la spinta delo strumento aptico.
- tra due linee di separazione vi sono poi tre elementi che vengono attivati solo quando un atomo è stato selezionato, figura 4.15. Tali elementi consistono in:
  - un label contenente il nome dell'atomo selezionato;
  - uno slider **radius** che permette di assegnare a tale atomo un raggio definito dall'utente, varia da 0.0 a 10.0;

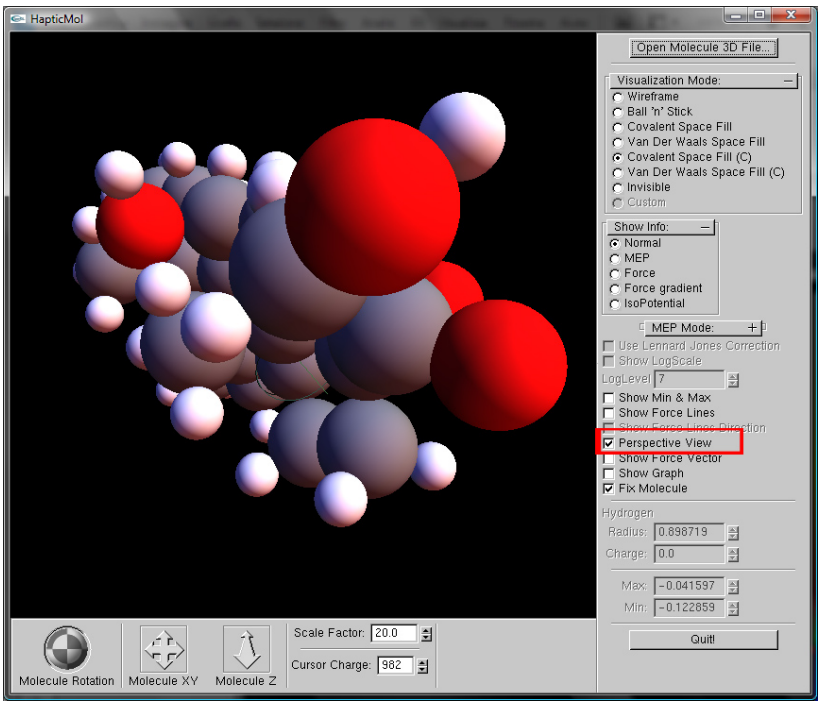

Figura 4.10: visualizzazione prospettica

  - un ulteriore slider **charge** per la selezione della carica da associare all'atomo nel sistema coulombiano, varia da -10.0 a 10.0;

- dopodichè, racchiusi tra due linee di separazione, vi sono due ulteriori slider che servono per controllare l'intervallo di valori per l'individuazione delle superfici isopotenziali:

  - slider **Max** che definisce il valore massimo dell'intervallo, varia tra minimo e massimo assoluti.
  - slider **Min** che individua il minimo dello stesso, anche questo controllo varia tra minimo e massimo assoluti.

- si trova, infine, il pulsante **Quit** per terminare il programma.

Come si evince da tutti i controlli illutrati questa parte di interfaccia utente permette di avere accesso a tutte le modalità di interazione possibili con lo strumento, sia da un punto di vista grafico che aptico. Nella successiva trattazione dell'engine grafico verranno discusse abbondantemente le modalità di visualizzazione fin qui introdotte.

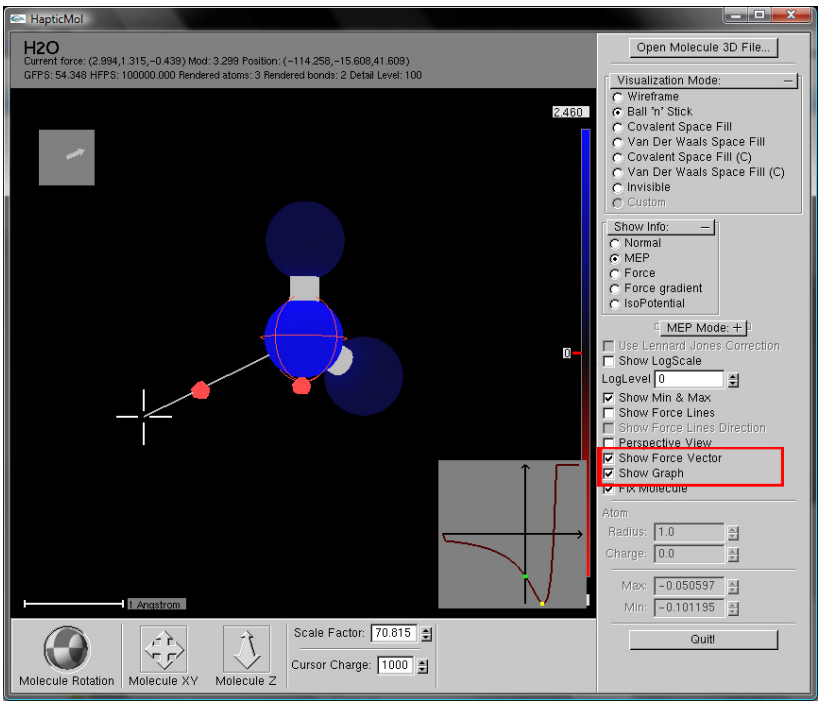

Figura 4.11: visualizzazione grafico e vettore forza

4.2.2 Sottofinestra controlli geometrici molecola

La sottofinestra sottostante la scena 3D presenta i controlli spaziali della molecola, ripresi completamente dall'interfaccia già presente nella versione originale:

- un controllo di rotazione per ruotare la molecola, da notare che la sfera che contraddistingue il pulsante ruota in concordanza con la molecola.
- un controllo di traslazione sugli assi x e y.
- un controllo di traslazione sull'asse z.
- uno slider **Scale factor** che individua il fattore di scala della molecola. Varia da 0.0 a 100.0
- uno slider **Cursor Charge** a cui è associato il valore della carica di prova. Varia da -10000 a 10000.

Rispetto alla versione iniziale del software è stato collegato il controllo in rotazione e sono stati posti dei limiti al controllo del fattore di scaling.

4.3 Engine Grafico

L'engine grafico si manifesta per intero nella finestra centrale del sistema. Tale finestra si compone di 5 elementi, vedere figura 4.12:

- una parte centrale contenente la visualizzazione della molecola secondo modalità che vedremo in seguito.
- un cursore aptico la cui forma è variabile in base alle modalità di visualizzazione.
- un indicatore di scala, posto in basso a sinistra, che identifica la lunchezza di un Angstrom rispetto alla visualizzazione della molecola.
- una scala cromatica a destra che indica il valore associato ai colori con cui il potenziale della molecola è rappresentato.
- un riquadro semitrasparente in alto che contiene informazioni di servizio, quali:

  Nome della molecola;

  Informazioni sulla forza:

  vettore;

  modulo;

  posizione del cursore;

  Graphic frame rate;

  Haptic frame rate;

  Numero di atomi e di legami renderizzati;

  livello di dettaglio della molecola.

L'unica informazione valida in assoluto per tutte le tipologie di visualizzazione è l'ortogonalità della camera, a meno, ovviamente, della modalità prospettica. Delle caratteristiche di tale ortogonalità parleremo nel seguito. Nel seguito provvederemo a definire tutte le visualizzazioni possibili e come esse sono state disegnate. Si rimandano però alla sezione sull'interazione chimica per spiegare come vengono valutati alcuni fattori di potenziale.

4.3.1 Visualizzazione geometrica della molecola

Se la molecola viene vista come ente geometrico e così rappresentata la visualizzazione consiste in sfere di diversa colorazione a seconda della tipologia

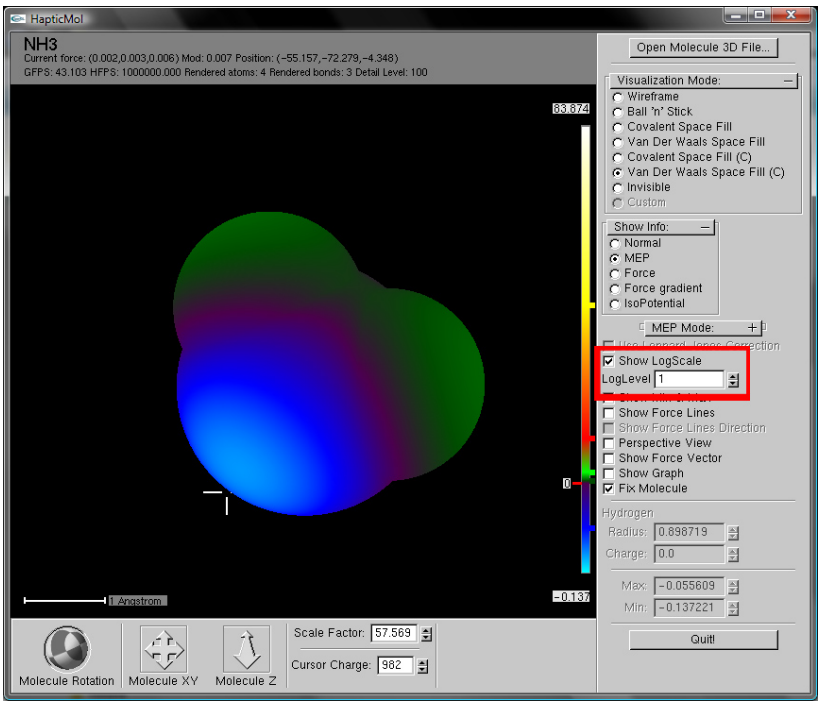

Figura 4.12: visualizzazione logaritmica mep

atomica.
Queste sfere, a differenza del prototipo iniziale possiedono un'ombreggiatura, definita da tre luci definite da altrettante direzioni:

- una prima luce, bianca, posta in direzione $(1, 1, 1, 0)$ angolo alto a destra dello schermo.
- una seconda luce, blu scura opaca, posta in direzione $(-1, 1, 1, 0)$ angolo alto a sinistra dello schermo.
- una terza luce, rossa scura opaca, posta in direzione $(1, -1, 1, 0)$ angolo basso a destra dello schermo.

Gli atomi invece possiedono come caratteristica del materiale la sola componente diffusiva del colore; ï¿½ infatti inutile introdurre anche la componente speculare, dispendiosa da un punto di vista computazionale e inutile allo scopo preposto. Infatti le luci e le ombre sulla molecola sono state introdotte per rendere la tridimensionalità della molecola in ambiente ortogonale.
La geometria della molecola è influenzata, oltre ai controlli di rototraslazione e scaling, dal raggio dei suoi atomi.

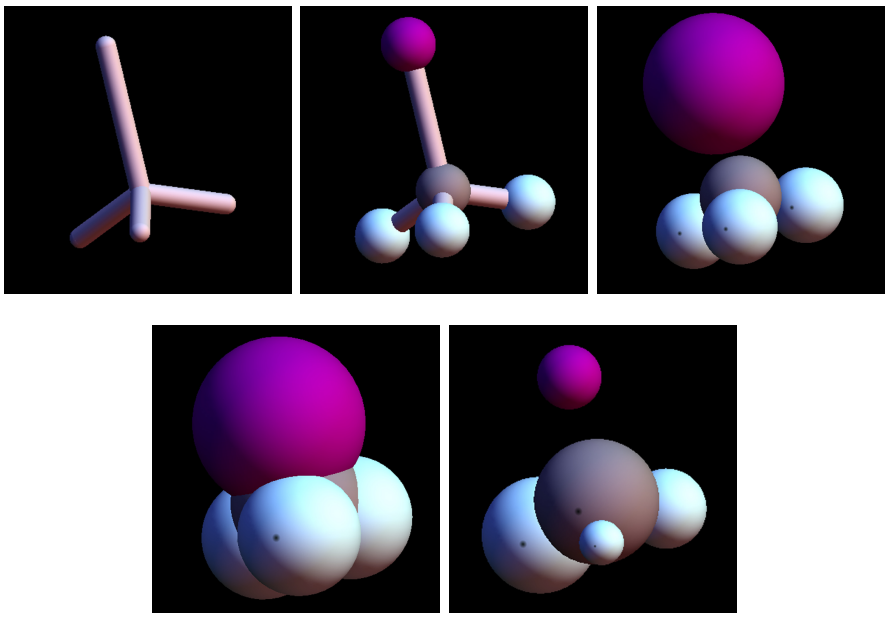

Figura 4.13: CF3I modalità visualizzazione

Nelle figure 4.13 e 4.14 vengono mostrate le cinque modalità di rappresentazione, di cui già si è discusso: wireframe, ball 'n' stick, Covalent Space fill, Van der Waals Space fill e custom radius. La modalità più interessante, e mancante nella precedente edizione del software, è rappresentata dalla possibilità di customizzare il raggio degli atomi coinvolti, attraverso un processo chiamato selezione. Tale strumento permette di interagire direttamente con la molecola e verrà discusso in una sezione a seguito.

Risulta opportuno definire qui come viene gestita la rototraslazione della molecola. Nel prototipo iniziale esistevano tre elementi per gestire la rototraslazione: un vettore spaziale che identificava la traslazione, due rotazione identificate da un numero in gradi, una sull'asse x e l'altra sull'asse y. Tale rappresentazione non permetteva di collegare il pulsante di rotazione collocato nella sottofinestra, che lavorava esclusivamente tramite una matrice di rotazione. Si è cosi deciso di introdurre una ulteriore matrice di rotazione nel controllo della molecola, per inglobare anche la rotazione generata attraverso il pulsante. Infatti nella versione precedente la molecola poteva essere ruotata solo utilizzando il pulsante destro del mouse, che attivava un controllo che ruotava la molecola rispetto gli assi x e y in base allo spostamento del cursore in una direzione piuttosto che nell'altra.

Tale sistema però faceva uso di 3 distinte rotazioni, scomode all'atto dell'introduzione della dinamicità del sistema. Così stato deciso di inglobare le

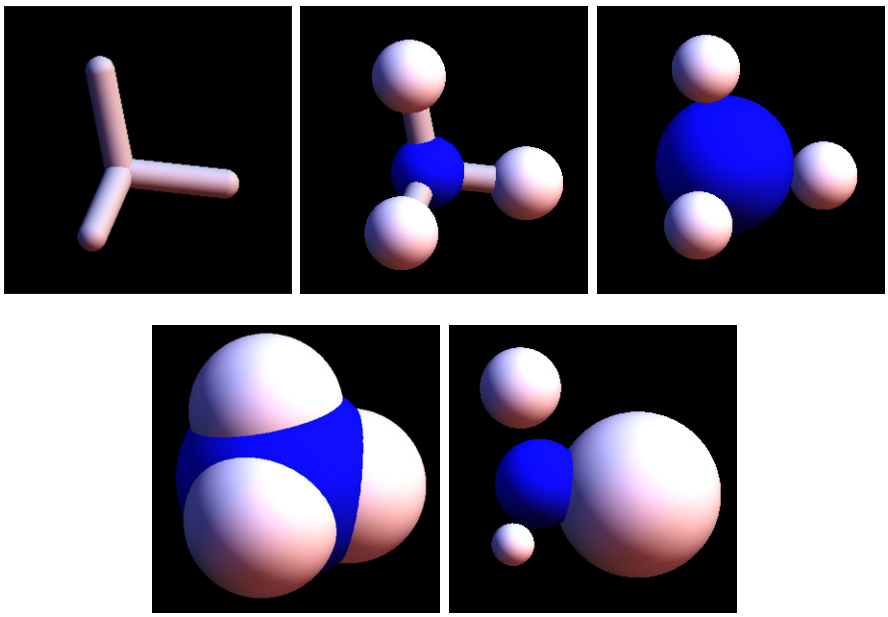

Figura 4.14: NH3 modalità visualizzazione

due rotazioni rispetto all'asse x e y nella matrice di rotazione generata dal pulsante. Per fare ciò è bastato moltiplicare tale matrice per:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\vartheta & -\sin\vartheta & 0 \\ 0 & \sin\vartheta & \cos\vartheta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

per la rotazione rispetto all'asse x, dove ϑ è la rotazione espressa in radianti, e:

$$\begin{pmatrix} \cos\varphi & 0 & \sin\varphi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\varphi & 0 & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

per la rotazione rispetto all'asse y.
In tal modo ogni qualvolta si ruotasse la molecola attraverso il controllo del tasto destro del mouse sulla scena, venivano accumulate le due rotazioni fino al frame grafico successivo in cui venivano effettuatte le moltiplicazioni di matrici per ottenere l'unica rotazione.
Inoltre, a posteriori della rotazione, vi é una ulteriore traslazione atta a portare il centro del sistema di riferimento della molecola nel baricentro della stessa, calcolato come la media dei centri degli atomi componenti nel

sistema di riferimento originario (derivato dal file in input). Per fare ció semplicemente si trasla la molecola, con il sistema di riferimento già rototraslato, di una quantità pari al vettore che identifica le coordinate del baericentro ma in direzione opposta a tale vettore; in tal modo il baricentro si trova esattamente nell'origine del sistema di riferimento della molecola.
Per riassumere le trasformazioni per gestire la matrice che indentifica il sistema di riferimento molecolare si articolano, nella fase finale, in due entità: vettore di traslazione e matrice di rotazione, a queste si somma una ulteriore traslazione che porta il baricentro della molecola nell'origine di questo nuovo sistema di riferimento.
Risulta inoltre opportuno individuare e riassumere qui le modalità con cui si puó rototraslare la molecola all'interno del programma:

- la traslazione si ottiene:

  attraverso il trascinamento del mouse tenendo premuto il tasto sinistro, puntando la scena 3d, valido solo per le direzioni x e y;

  premendo i pulsanti di interfaccia e mantenendo premuto il tasto sinistro, in tal modo è possibile anche traslare sull'asse z con il secondo controllo di traslazione.

- la rotazione si ottiene:

  tenendo premuto il tasto destro e trascinando il mouse, sempre puntando la scena 3d;

  premendo il controllo di rotazione dell'interfaccia e trascinando il mouse, questa azione ruoterá la sfera che individua il pulsante e la molecola di conseguenza.

Oltre a questi due metodi gestiti i primi dall'engine grafico attraverso i callback sul mouse e i secondi attraverso l'engine interfaccia utente si accosta un ulteriore controllo tramite l'engine aptico di cui si parlerá piú avanti.
Da notare infine che nella visualizzazione geometrica non é necessario inserire alcuna scala.

4.3.2 Selezione degli atomi

La selezione degli atomi é uno strumento molto importante dell'applicazione sviluppata perché permette di agire direttamente sui singoli atomi, modificandone proprietà quali raggio e carica e visualizzandone il nome.
In figura 4.15 è mostrato un esempio di utilizzo dello strumento selezione. Si può notare come l'atomo selezionato sia evidenziato da due linee concentriche verdi e nell'interfaccia appaiano le informazioni relative all'atomo

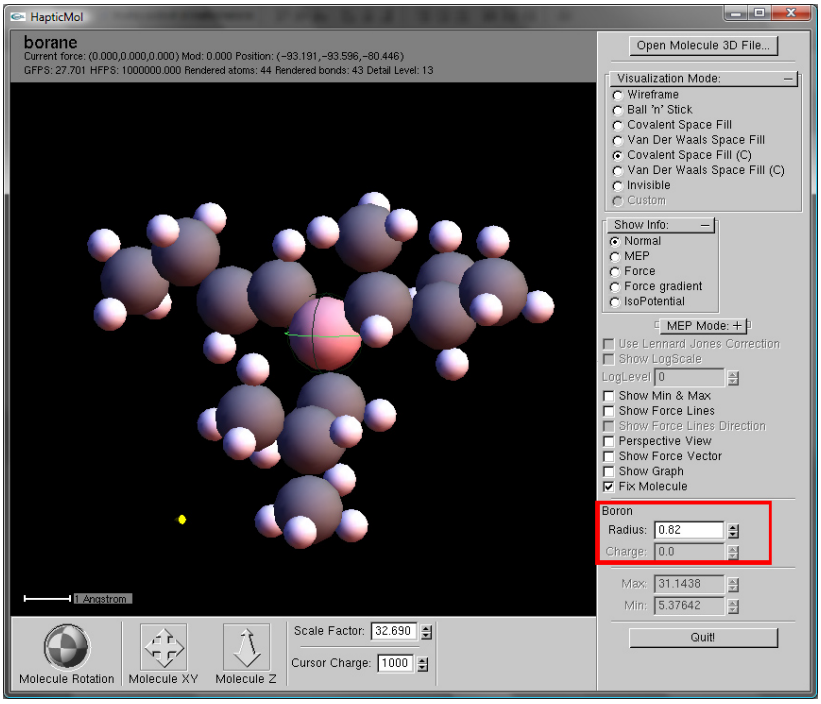

Figura 4.15: selezione di un atomo

quali nome, raggio e carica.
La selezione funziona attraverso 3 fasi:

- in una prima fase l'atomo viene evidenziato con un colore rosso al passaggio del puntatore del mouse sopra lo stesso, prima immagine figura 4.16. Il riconoscimento dell'atomo pre-selezionato avviene mappando la viewport rispetto allo spazio 3d. Tale spazio 3d appare ortogonale e quindi mantiene i viewing ray perpendicolari al piano dello schermo consentendo una semplice mappatura proporzionale tra gli assi x e y e le coordinate dello schermo u e v. In uno spazio ortogonale, infatti, la coordinata z viene mascherata nel sistema e non influisce sulla posizione dell'atomo nello schermo. Ad ogni movimento del cursore del mouse viene quindi controllata la distanza con il nucleo degli atomi presenti, se tale distanza è minore del raggio scalato allora l'atomo è potenzialmente pre-selezionato; a questo punto si discrimina dalle sovrapposizioni di più atomi pre-selezionando quello con un valore maggiore della coordinata z del centro.

- A seguito della prima fase di pre-selezione dell'atomo l'utente può decidere di selezionarlo effettivamente, per far ciò preme semplicemente il pulsante sinistro del mouse e l'atomo viene evidenziato di giallo,

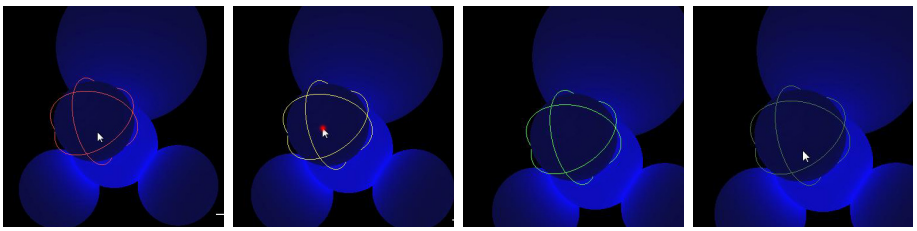

Figura 4.16: varie fasi di selezione

seconda immagine figura 4.16. In questa fase si può anche decidere di spostare l'atomo trascinando il mouse.

- al rilascio del tasto sinistro del mouse, infine, l'atomo apparirà evidenziato di verde e sarà effettivamente selezionato, , terza immagine figura 4.16.

Da notare che a seguito della selezione il centro della molecola non é piú il baricentro, bensí il nucleo dell'atomo selezionato. In tal modo la rotazione avviene mantenendo fisso l'atomo e ruotando tutto il resto. Per cui riprendendo le trasformazioni che coinvolgono la molecola, a posteriori della rototraslazione il sistema non viene più traslato dell'inverso del baricentro, bensí dell'inverso del centro dell'atomo selezionato.
Una volta che un atomo é stato selezionato é possibile deselezionarlo in due modi:

- semplicemente selezionando un altro atomo.
- cliccando ulteriormente sull'atomo selezionato, in tal modo si torna alla condizione iniziale in cui nessun atomo risulta selezionato. Nella fase di de-selezione al momento del passaggio del mouse sopra l'atomo selezionato, e quindi colorato di verde, la colorazione dello stesso diventa verde scuro, in tal modo si evidenzia all'utente la possibilità di de-selezionare l'atomo, ultima immagine figura 4.16

Oltre alla selezione tramite mouse é presente un altro sistema di selezione di cui si parlerá piú avanti basato sull'interfaccia aptica.

4.3.3 Visualizzazione proprietà chimiche della molecola

Per gestire le informazioni chimiche da convogliare all'utente sono state introdotte tre modalità di rappresentazione (una delle quali già effettivamente presente nel sistema originario) che individuano potenziale, forza e gradiente della forza colorando in vario modo la geometria della molecola. A questa

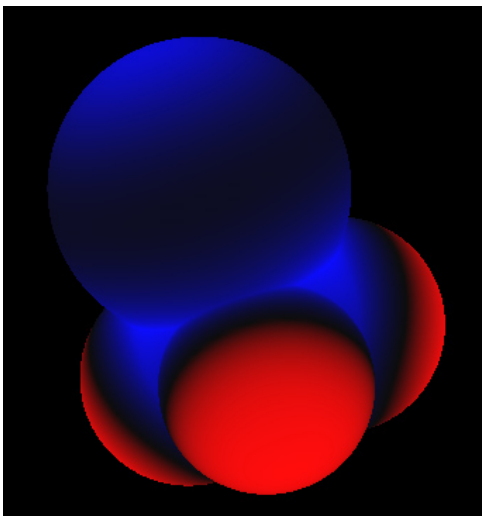

Figura 4.17: potenziale molecolare colorazione locale

variabilità di informazione si va a sommare la possibilità di visualizzare il sistema in scala logaritmica, per ognuna delle tre visualizzazioni. Infine potenziale e forza possono essere calcolati in maniera diversa rispetto all'uso di mappe di potenziale o di gradiente, ma di ciò si parlerà nella apposita sezione di analisi degli aspetti chimici del sistema.
Procedendo con ordine esaminiamo ora tutti i tipi di visualizzazione.

Visualizzazione potenziale

In una prima analisi su scala locale il MEP viene mostrato come colore sulla superficie della molecola, con una gradazione che va dal blu per aree a potenziale positivo al rosso per aree a potenziale negativo, figura 4.17.

Tale colorazione viene ottenuta su scala locale riscalando i colori corrispondenti ai valori di potenziale presenti sulla superficie della molecola. Nello specifico, a differenza di quanto avveniva nel prototipo iniziale, le gradazioni di colore assumono un significato nuovo in quanto: il colore blu di massima intensità viene assegnato al valore più alto di potenziale sulla superficie molecolare attualmente mostrato, mentre il colore di massima intensità rosso corrisponde al minimo superficiale. La decisione di mappare i colori in questo modo è stata introdotta a partire dalla considerazione che i valori di potenziale all'interno della molecola sono estremamente variabili, ad esempio vi è differenza di parecchi ordini di grandezza tra i valori sulla superficie di Van Der Waals e i valori sulla superficie ball 'n' stick, perciò diventa poco significativo effettuare una colorazione della molecola su base

assoluta e lineare (da qui l'introduzione della scala logaritmica). Al contrario nella visualizzazione iniziale il potenziale mostrato era classificato attraverso valori di default che non permettevano di mostrare l'informazione relativa alla quantità di potenziale associato, ma solo l'informazione di segno.
Per attuare questa colorazione bisogna ad ogni cambio di visualizzazione della molecola, rispetto alla variabile raggio atomico, ricalcolare il minimo e il massimo di potenziale sulla superficie, questi valori individueranno la scala di colori da associare al potenziale.
Esiste una visualizzazione su scala assoluta dei colori riferiti al potenziale che sfrutta la concezione di scala logaritmica, di tali scale parleremo più avanti.
Differentemente dal caso geometrico, in cui si è utilizzato una funzione predefinita per disegnare le sfere, nel caso della visualizzazione del potenziale si è reso necessario disegnare manualmente, punto punto, la superficie della sfera. Ciò perché non era possibile altrimenti assegnare ad ogni vertice il colore opportuno. Pertanto i punti della sfera sono stati scorsi nel seguente modo: dati n e m il numero di stacks e slices in cui la sfera deve essere divisa (per meridiani e paralleli), r il raggio dell'atomo, si calcolavano le coordinate (x, y, z) del punto i-esimo come mostrato nell'equazione 4.1.

$$\begin{aligned} \forall i \quad &| \quad 0 <= i <= nm \\ x \quad &= \quad r \sin(\frac{\pi(i\%n)}{n}) \cos(\frac{2\pi \left\lfloor \frac{i}{n} \right\rfloor}{m}) \\ y \quad &= \quad r \cos(\frac{\pi(i\%n)}{n}) \\ z \quad &= \quad r \sin(\frac{\pi(i\%n)}{n}) \sin(\frac{2\pi \left\lfloor \frac{i}{n} \right\rfloor}{m}) \end{aligned} \tag{4.1}$$

Per il disegno della superficie della sfera si è utilizzato la primitiva **QUAD_STRIPS** per cui vengono disegnati dei quadrilateri connessi uno all'altro a formare una superficie. Nello specifico per ogni quadrilatero è necessaria la definizione di 4 punti di cui 2 comuni a quello successivo; riprendendo la posizione definita in 4.1 chiamata A, si trova B sostituendo a i la variabile j definita come $j = i+n$, mentre C e D si ottengono semplicemente con $i+i$ e $j+1$ rispettivamente. A questo punto basta iterare per ogni i da 0 a mn e si ottengono la superficie della sfera.
Per aumentare le prestazioni viene effettuato un pre-clipping della scena, vengono cioè individuati gli atomi che non sono presenti nella scena perché oltre i limiti del viewing volume e viene evitato il calcolo dei punti di tali atomi. Per fare ciò nella visualizzazione ortogonale basta verificare se il centro dell'atomo è più distante del raggio al clipping plane da cui esce.

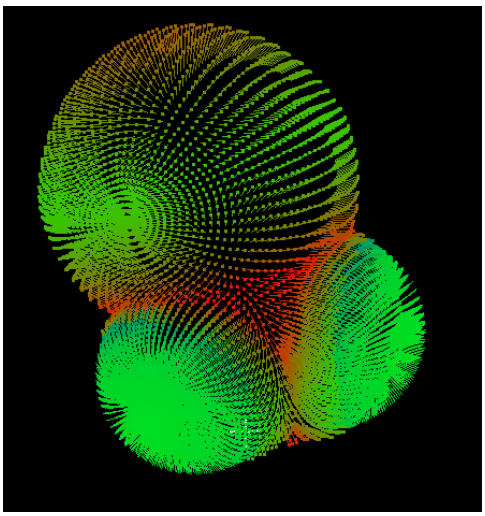

Figura 4.18: forza colorazione locale

Visualizzazione forza

La forza sulla superficie molecolare viene rappresentata come un insieme di vettori la cui origine giace sulla superficie, figura 4.18. Tali vettori sono lunghi un terzo di Angstrom e sono diretti come il campo vettoriale del gradiente, il loro colore indica invece il modulo. Tale colorazione, come nel caso del mep, può essere sia locale che assoluta, questo secondo caso sempre utilizzando la scala logaritmica.
A differenza del potenziale la colorazione del modulo della forza si basa solo sulla concezione di vettore a lunghezza massima, che sarà il valore da associare agli estremi della scala. A questo si aggiunge l'introduzione di un segno nella colorazione dei vettori forza, definiremo un vettore positivo se uscente dalla superficie dell'atomo e negativo se entrante; tale distinzione permette di individuare a colpo d'occhio i punti nello spazio dove la carica di prova è attratta dall'atomo o respinta. Avremo quindi una scala così formata:

- se il vettore forza è diretto verso l'interno dell'atomo e il suo modulo è alto (vicino al massimo modulo locale) la sua colorazione sarà blu;
- se la forza è debole, ma diretta verso l'interno dell'atomo il vettore assumerà una colorazione proporzionale tra il verde e il blu;
- se la forza è pari a zero il suo colore sarà verde;
- se il vettore forza è uscente dall'atomo e abbastanza debole allora avrà un colore proporzionale tra il verde e il rosso;

- infine se il vettore forza ha modulo alto e direzione esterna all'atomo verrà colorato di rosso.

I vettori rappresentati sono semplicemente composti da un trattino al cui estremo, indicante la direzione, è posto un punto. Si è scelto di rappresentare i vettori in maniera così semplice in modo da renderli poco dispendiosi dal punto di vista computazionale e poter permettere al sistema di gestirne una maggiore quantità.
Per derivare la posizione origine dei vettori è bastato scorrere la superficie sferica in punti equispaziati come si è fatto per il disegno dei punti superficiali del potenziale.
Per aumentare le prestazioni, e per garantire una visualizzazione senza interferenze dei mini vettori, è stato introdotto un sistema di culling dei vettori che dovrebbero rimanere nascosti nella rappresentazione. Il culling agiscce su tre livelli:

- in un primo sistema si evitano di disegnare i vettori la cui origine è nella metà non visibile di ogni singolo atomo. Per far ciò semplicemente basta controllare la z del centro dell'atomo e non disegnare i vettori con un valore z inferiore.

- nel secondo livello si individuano i vettori interni alla molecola, a seguito della parziale sovrapposizione di atomi, e se ne evita il disegno. Semplicemente per ogni vettore si controlla quindi che la distanza dal centro degli altri atomi sia maggiore del raggio di quest'ultimi, cioè che non sia interno al volume di un altro atomo, in caso contrario si evita di disegnare il vettore.

- il terzo livello infine evita le sovrapposizioni, cioè se due atomi sono sovrapposti uno all'altro nella visualizzazione a schermo. Nello specifico controlla che la distanza tra le coordinate x e y del vettore da disegnare e le coordinate x e y del centro degli altri atomi sia maggiore del raggio di questi ultimi; altrimenti se la z dell'atomo a cui appartiene il vettore è superiore a quella degli altri disegno il vettore, in caso contrario lo evito. Si noti che questo caso nella sua applicazione sottointende il caso precedente.

Questo algoritmo permette non solo di aumentare notevolmente le prestazioni, riducendo le figure geometriche da disegnare, ma di ottenere delle immagini migliori, senza sovrapposizioni. In figura 4.19 viene mostrato l'effetto di tutte queste operazioni di culling, infatti la prima immagine, confusionaria, ne è priva, mentre la seconda ne sfrutta appieno le potenzialità.

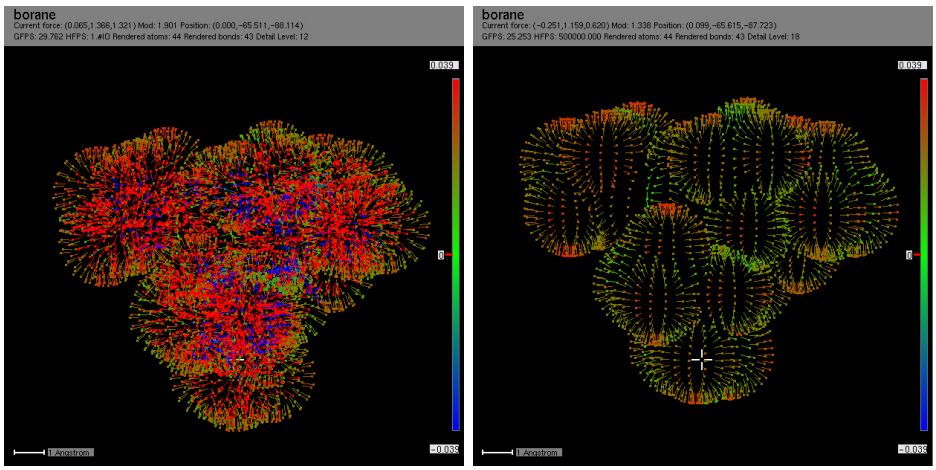

Figura 4.19: esempio di miglioramento prestazioni con culling dei vettori

Oltre al culling viene effettuato anche il clipping illustrato per la visualizzazione del MEP.

Visualizzazione gradiente della forza

La visualizzazione del gradiente della forza è stata introdotta per mostrare all'utente la direzione e l'entità delle variazioni nel campo di forza. Tale rappresentazione risulta molto simile a quella del campo di forze e fa uso delle stesse entità grafiche quali i vettori e lo stesso schema di colorazione. Il gradiente della forza è mostrato in figura 4.20.
A differenza del potenziale e della forza esiste un solo metodo di calcolo del gradiente della forza e avviene a partire dalla mappa del gradiente del potenziale, ma le modalità con cui avviene questo calcolo saranno illustrate più avanti nella trattazione.
Anche nel caso del gradiente della forza sono stati introdotti gli algoritmi di culling e clipping delle modalità precedenti.

4.3.4 Scale di colorazione

Come si è visto precedetemente sono state introdotte diverse tipologie di colorazione nel sistema. A tali colorazioni sono state associate diverse scale che ne indicano visivamente il valore.
In figura 4.21 sono rappresentate diverse scale di colorazione che identificano diversamente le informazioni introdotte. Le scale sono enti grafici che presentano diverse caratteristiche comuni:

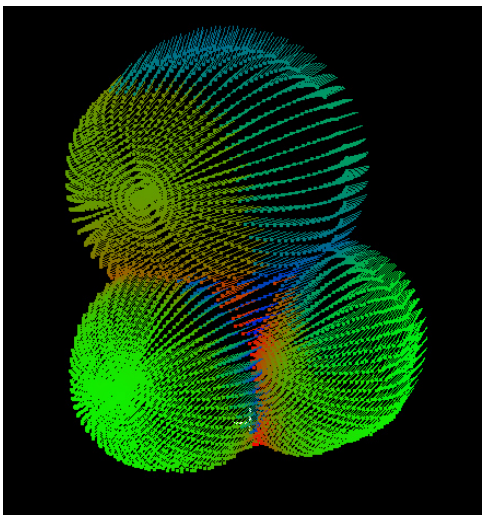

Figura 4.20: gradiente della forza colorazione locale

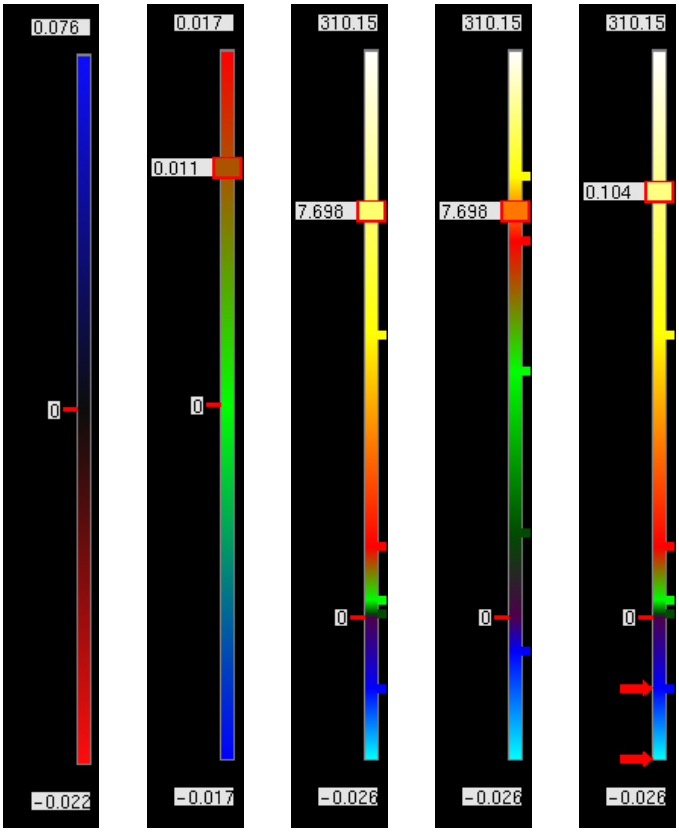

Figura 4.21: diverse scale di colorazione

- presenta una scala colorimetrica continua che crea delle sfumature tra i vari punti salienti della stessa;
- in alto riporta il valore numerico del massimo valore di scala;
- in basso il minimo valore di scala;
- in mezzo alla scala è presente inoltre ben evidenziato lo 0 in modo da discriminare tra valori positivi e negativi;
- al passaggio del mouse sopra la scala viene segnalato il valore numerico che la scala assume in quel punto e il colore al quale si riferisce viene evidenziato.

L'interazione con il mouse, in maniera simile alla selezione degli atomi, è gestita rimappando lo spazio dello schermo con la scena 3d in uno spazio ortogonale.
A queste caratteristiche comuni vengono accostate caratteristiche locali ad ognuna delle scale. Riprendendo la figura 4.21 troviamo:

1. una semplice scala lineare indicante i colori del potenziale tra il minimo e il massimo locali sulla superficie.
2. una scala lineare e locale indicante i colori del modulo dei vettori forza di cui si è discusso precedentemente. Da notare l'evidenziazione di un particolare punto della scala, su cui è puntato il cursore del mouse, e la simmetria della scala: non ha senso avere valori diversi per il valore massimo e minimo dato che tale valore è rappresentato dal modulo dei vettori, come già indicato la positività e la negatività sono state introdotte convenzionalmente per evidenziare l'informazione utile.
3. una scala logaritmica globale per la colorazione di mep forza e gradiente della forza. Vediamo qui che il massimo è molto più elevato dei casi precedenti, addirittura diversi ordini di grandezza superiore, ciò avviene perché i massimi e i minimi di scala sono assoluti rispetto all'intera mappatura di potenziale. Anche qui è evidenziato un valore e si può notare che la scala non procede linearmente, ma evidenzia maggiormente i valori bassi della scala: questo è l'effetto di una scala logaritmica o iperlogaritmica. Lo 0 d'altro canto è posizionato in una parte inferiore della scala rispetto ai casi precedenti, ciò avviee per l'ampia differenza di valori tra elementi positivi e negativi del potenziale, di diversi ordini di grandezza; non avrebbe avuto senso concedere altrettanto spazio e variazione colorimetrica ai valori negativi del potenziale. Si può notare che a differenza dei casi precedenti

in cui 3 soli colori identificavano la scala, qui sono presenti addirittura 8 colori, dal basso azzurro, blu, viola, verde scuro,verde,rosso,giallo e bianco, di cui 5 contraddistinti da una maniglia a lato destro della stessa.

4. le maniglie che definiscono i 5 colori intermedi possono essere spostati, perciò in figura troviamo la stessa scala logaritmica i cui colori sono stati spostati. Ciò permette di personalizzare la scala in modo da evidenziare l'informazione più utile e interessante al momento. Da notare che nonostante i colori siano cambiati i valori di scala posinalmente rimangono gli stessi, infatti il cursore è posto nella stessa posizione del caso precedente e indica lo stesso valore, sebben con un colore differente.

5. Infine vi è un ulteriore strumento per la scala logaritmica che viene attivato per selezionare un intervallo per i volumi isopotenziali. Tale intervallo viene delimitato dalle due frecce a sinistra della scala che possono essere mosse a piacimento sempre utilizzando il mouse. Da notare che in questa scala logaritmica rispetto ai casi precedenti è stata modificato il grado logaritmico per cui ad uno stesso livello posizionale corrisponde un valore numerico corrispondente, nello specifico il grado della scala iperlogaritmica è aumentato.

L'introduzione delle varie scale permette di dare un senso non solo qualitativo, ma anche quantitativo del potenziale e della forza attorno alla molecola e risulta un utile strumento per la navigazione del sistema.

Scala logaritmica

Risulta ora opportuno valutare le motivazioni che hanno portato alla neessita di utilizzare una scala logaritmica per la colorazione della molecola.
Si è già visto come l'ordine di grandezza vari molto in base alla superficie selezionata per cui ad esempio per una molecola di NH3 il potenziale sulla superficie di Van Der Waals arriva ad un massimo valore di 0.048 mentre il massimo assoluto vale 83.874; si può così facilmente capire perchè per valutare un metodo assoluto per attribuire un colore ad un valore di potenziale una scala lineare non sarebbe stata sufficiente. A tal fine è stato introdotta in primo luogo la scala logaritmica, basata sull'idea di sostituire alla scala lineare che assegna a c compreso nel range $[0, 1]$ l'equazione $c = p/m$ dove p

è il valore di potenziale e m è il massimo l'equazione in 4.2.

$$c = \frac{\log(p+1)}{\log(m+1)} \tag{4.2}$$

Attraverso questo sistema si è in grado di accentuare le differenze nei valori bassi a scapito di quelli alti nella scala. Infatti la scala non procede più a intervalli costanti come nel caso lineare, bensì a rapporti costanti, per esempio se x è il valore presente ad un certo punto u della scala e mi chiedo che valore ci sarà a $2u$ nel caso lineare troverò $2x$ mentre nel caso logaritmico ci sarà $(x+1)^2$. L'aggiunta del $+1$ nell'equazione 4.2 è dovuta al fatto che la funzione logaritmo vale 0 in 1, mentre tende a infinito per numeri inferiori all'unità, ma il potenziale possiede anche i valori tra 0 e 1 per cui risulta necessario introdurre un fattore $+1$ che trasformi il dominio del potenziale da $(-\infty, +\infty)$ a $[1, \infty)$. Per il potenziale negativo è bastato calcolare la posizione utilizzando il modulo del potenziale e associargli un'altra scala di colore per i valori negativi.
Pultroppo però un singolo grado logaritmico, sebbene bastasse per la valutazione del potenziale, non era sufficientemente fine per visualizzare le differenze nella forza e nel gradiente della forza, così si è passati al caso iperlogaritmico.

In figura 4.22 sono mostrati gli effetti dell'aumento di grado della scala iperlogaritmica:

1. nella prima figura infatti, nonostante la scala sia già logaritmica di primo grado, tutti i valori assunti dalla molecola di NH3 sulla superficie di Van Der Walls per il gradiente secondo sono estremamente vicini a 0 e presentano un colore viola indicante la colorazione più prossima a zero.

2. nel secondo caso il grado del logaritmo passa a 10, il che significa l'innestazione di 10 logaritmi uno dentro l'altro, e già si incominciano a intravedere le differenze tra le varie parti della molecola; comunque la colorazione occupa ancora i più bassi punti della scala.

3. nella terza immagine passiamo al grado 100 e comunque si rimane a bassi livelli della scala.

4. infine nell'ultima immagine con 200 logaritmi si arriva a dei livelli più soddisfacenti.

La formula utilizzata per gestire le scale iperlogaritmiche è illustrata nell'equazione 4.3 dove si innestano per n volte dei logaritmi (n ne definisce

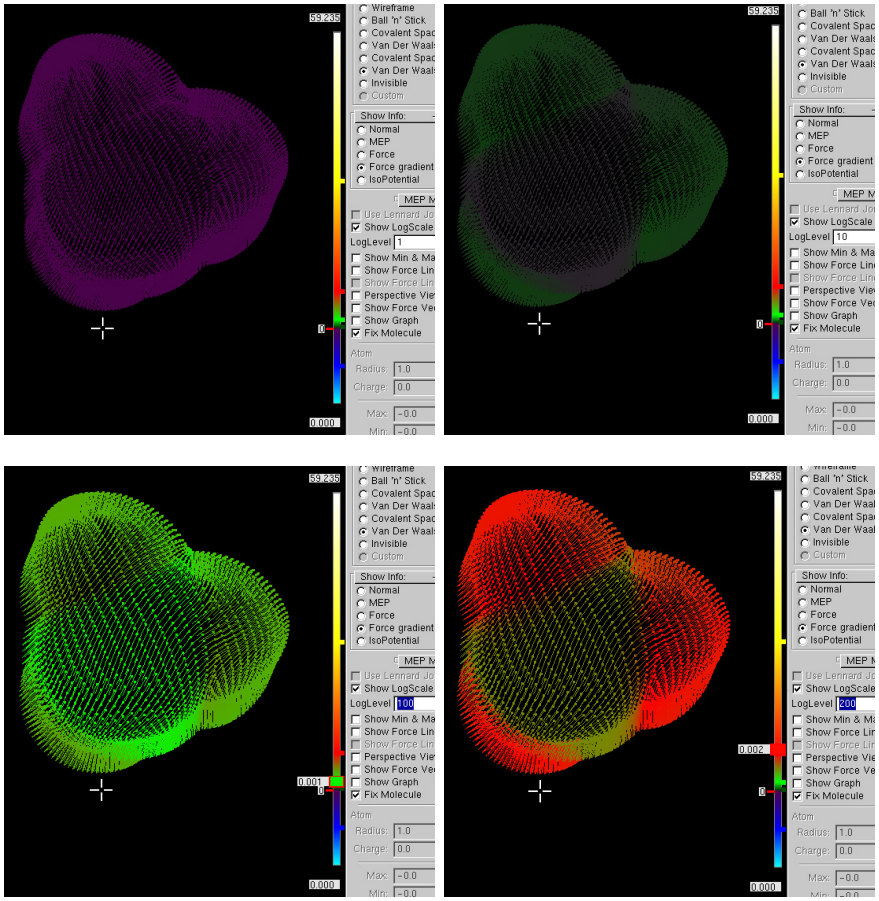

Figura 4.22: esempio scala iperlogaritmica

il grado). Da notare la presenza del fattore correttivo +1 per ogni logaritmo introdotto.

$$c = \frac{\log(\log(\log(\ldots \log(p+1)\ldots)+1)+1)}{\log(\log(\log(\ldots \log(m+1)\ldots)+1)+1)} \tag{4.3}$$

Nel calcolo del valore associato ad un certo colore nella scala si è reso necessario effettuare il procedimento inverso, illustrato in 4.5 per il caso logaritmico semplice e in 4.5 per la scala iperlogaritmica, dove p è il valore di potenziale, m il massimo e c il punto nella scala.

$$p = e^{(c\log(m+1))} - 1 \tag{4.4}$$

$$p = e^{e^{\cdots^{e^{c\log(\log(\log(\ldots\log(m+1)\ldots)+1)+1)}-1}\cdots-1}-1} - 1 \tag{4.5}$$

L'introduzione della scala iperlogaritmica non ha l'unico scopo di evidenziare a video le differenze nelle parti basse della scala, bensì viene utilizzata anche per la renderizzazione della forza a livello aptico. Anche le forze generate, infatti, risentono dell'effetto di differenza di ordine di grandezza tra forze attrattive e repulsive: le forze attrative, in caso di carica di prova positiva, risultano estremamente deboli rispetto a quelle repulsive, così se si mappassero linearmente le forze percepibili attraverso il PHANToM, non si sarebbe in grado di percepire alcunchè di attrattivo. L'introduzione delle scale iperlogaritmiche permette invece di percepire sia le forze attrattive che repulsive, ma di questo tratteremo nello specifico nella sezione riguardante l'engine aptico.

4.3.5 Visualizzazione volumi isopotenziali

La rappresentazione di volumi isopotenziali consiste nel visualizzare i punti appartenenti alla griglia di potenzile compresi in un intervallo definito dall'utente. Tali punti disegnati assieme rendono l'idea dell'area di spazio che racchiude tutti questi punti. Questa modalità permette di individuare non solo i minimi assoluti, ma anche i minimi locali e comunque tutte le aree nello spazio che presentano una certa caratteristica, ad esempio di avere potenziale negativo.
Tale modalità grafica aiuta decisamente l'utente nella navigazione del potenziale, permettendo di individuare direttamente dove lo strumento aptico dovrebbe essere attratto o respinto. Anche per questa visualizzazione sono state usate le scale logaritmiche, in quanto era necessario individuare i colori in maniera assoluta per definire gli intervalli utili.
In figura 4.23 sono mostrati esempi di volumi isopotenziali per una molecola di CF3I, in tale esempio sono mostrate varie informazioni ottenibili con lo strumento:

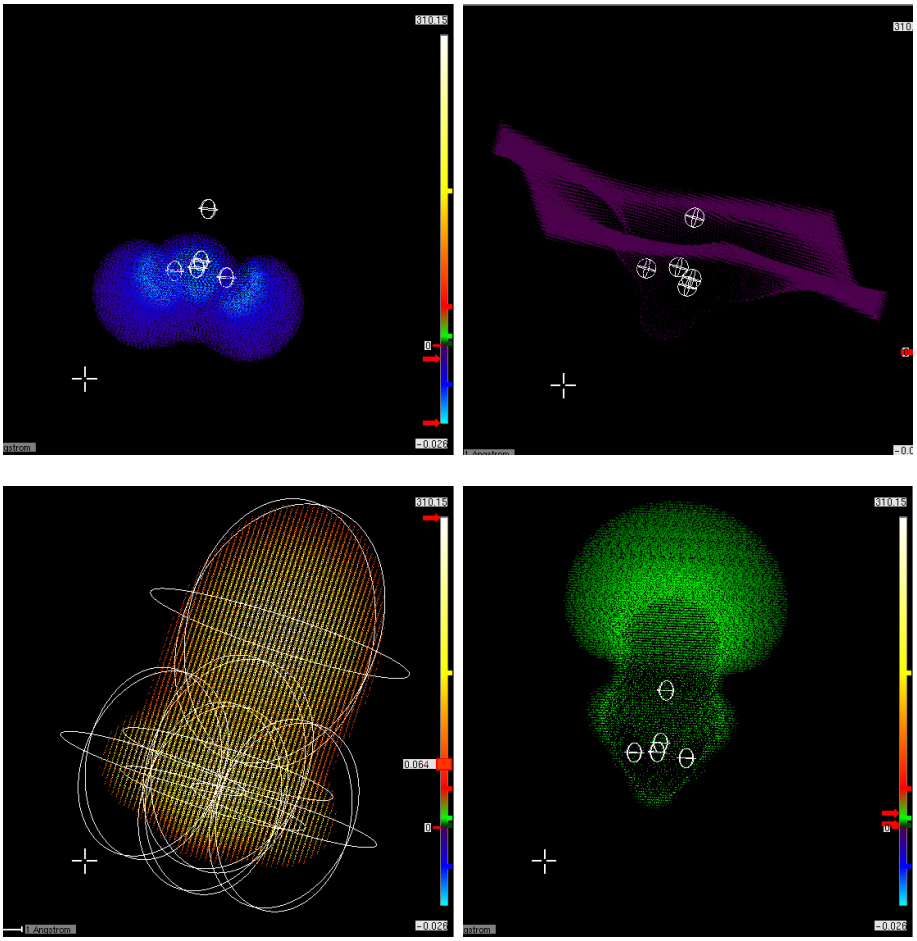

Figura 4.23: esempi volumi isopotenziali

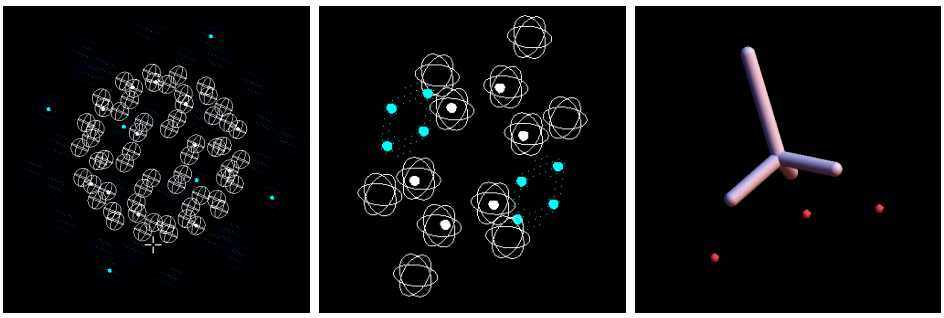

Figura 4.24: varie visualizzazioni minimi e massimi

1. nella prima immagine sono messe in evidenza delle aree a potenziale negativo, da cui, cioè, il cursore aptico dovrebbe essere attratto.

2. nella seconda è mostrata la superficie di taglio tra aree a potenziale positivo e negativo. Tale rappresentazione è ottenuta prendendo come intervallo un intorno dello 0.

3. nella terza immagine sono presentati valori appartenenti ad un intervallo positivo che permette di individuare una forma della molecola probabilmente più vicina a quella reale. Da notare che lo scheletro della molecola, sfere bianche, individua i raggi di Van Der Waals.

4. infine nell'ultima immagine viene presentata una fascia di valori positivi, ma molto bassi, in tal modo si mette in evidenza, nel caso di CF3I, la forma a 'fungo' della molecola

Come già è stato accennato, oltre alla presenza dei punti che individuano il potenziale è presente un altro elemento nella rappresentazione: gli scheletri della molecola, che permettono di ottenere un riferimento rispetto alla posizione degli atomi della molecola; il raggio degli atomi di tale scheletro è gestibile esattamente come nelle altre modalità di visualizzazione.
Da notare che per motivi prestazionali è stato introdotto un controllo sul frame rate anche per quanto riguarda i volumi isopotenziali: in base al frame rate corrente viene preso un punto ogni tot dalla lista dei punti compatibili con l'intervallo, in tale modo è possibile non rallentare eccessivamente il sistema nel trattamento di intervalli contenenti un alto numero di punti.

4.3.6 Rappresentazione massimi e minimi

I minimi e i massimi vengono rappresentati nel sistema come delle piccole sfere colorate diversamente, figura 4.24. Dalla figura si può notare che i

minimi e massimi assoluti spesso non sono unici, ci sono cioè vari punti nello spazio in cui il potenziale raggiunge valori molto prossimi ai minimi e ai massimi. Analizzando le singole immagini notiamo:

1. Il primo caso presentato è il fullerene composto da 60 atomi di carbonio legati in una struttura a 'pallone da calcio'. Si può qui notare che vi sono vari minimi assoluti, evidenziati dalle palline azzure, e vari massimi assoluti, palline bianche all'interno dello scheletro della molecola. Nonostante tutto vi sono comunque tante altre aree evidenziate da un potenziale particolarmente basso che non ricadono nella categoria minimi, questo perché i valori assunti in tali minimi non sono abbastanza prossimi ai minimi assoluti, fatto che li rende minimi locali. Allo stesso modo solo alcuni dei nuclei degli atomi di carbonio sono rappresentati da massimi assoluti.

2. Nel secondo caso, molecola di benzene, notiamo che i nuclei degli atomi che contraddistinguono i massimi assoluti sono i 6 carbonii mentre non è facilmente distinguibile un unico punto di minimo. Questo avviene perché in questo caso il minimo non corrisponde ad un punto, bensi ad un volume che forma una 'ciambella', i punti minimo selezionati diventano quindi dei punti equispaziati su tale ciambella.

3. Nell'ultimo caso, infine, esaminiamo una molecola di CF3I la cui visualizzazione dei minimi e massimi è legata ad altri colori, rosso per i minimi e blu per i massimi. In questa immagine si può notare come i minimi possano essere visualizzati anche nella visualizzazione geometrica. Qui i minimi sono netti in corrispondenza dei 3 atomi di fluoro, mentre il massimo, non visibile, è sito nel nucleo dell'atomo di iodio.

Dagli esempi riportati è facile notare che i massimi corrispondono sempre ai nuclei degli atomi più energetici, mentre le aree di minimo possono assumere varie forme che vengono sintetizzate in una scelta di punti salienti. Ma come vengono scelti tali punti salienti? Semplicemente si scelgono dei punti il cui valore di minimo sia non meno del 99% del minimo assoluto, cioè viene tollerato un errore del 1% rispetto al minimo assoluto. Inoltre vengono scartati i punti la cui distanza da un altro punto di minimo sia inferiore ad un certo limite di soglia, in tale modo si evita di selezionare più minimi per una singolo volume di minimo (o di massimo). Allo stesso modo vengono presi tutti i massimi il cui errore rispetto a quello assoluto non superi il 5%. Non sta agli autori di questo applicativo individuare le ragioni per cui alcuni minimi e massimi siano rispettivamente più minimi e più massimi degli altri,

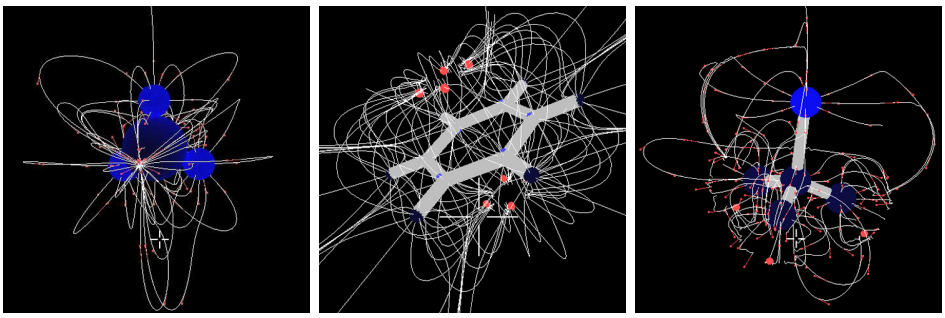

Figura 4.25: visualizzazione linee di forza

ad ogni modo si potrebbe ricercare il problema nei dati sperimentali da cui la ricostruzione del potenziale parte, la cui variabilità causa queste oscillazioni.

4.3.7 Linee di forza

Un ulteriore modalità di visualizzazione permette di rendere graficamente le linee di forza, figura 4.25. Nelle tre immagini rappresentate si possono notare le due modalità di visualizzazione delle linee di forza, una con dei vettori che ne indichino la direzione, mentre l'altra priva di essi. Nello specifico in figura 4.25 troviamo:

1. una molecola di NH3 le cui linee di forza sono rappresentate comprensive di direzione, i piccoli vettori rossi. Si può notare che tutte le linee di direzione sono dirette esternamente agli atomi e che si accumulano in direzione del minimo di potenziale.

2. una molecola di benzene, stavolta senza vettori di direzione, le cui linee di forza finiscono comunque nei punti di minimo, evidenziati in figura.

3. una molecola di CF3I in cui sono rappresentati sia minimi che linee di forza che direzioni.

Innanzitutto, cosa sono le linee di forza? Rappresentano le traiettore che una carica di prova positiva percorrerebbe se il suo punto di partenza fosse uno dei punti equispaziati sulla superficie della molecola. Nello specifico seguire le linee di forza significa seguire il gradiente da un determinato punto di partenza fino all'equilibrio o all'allantonamento dalla griglia di potenziale. Nel sistema sviluppato si sono individuati come punti di inizio dei punti equispaziati sulla superficie della sfera: in particolare se m e n sono gli stacks e gli slices della sfera che definisce un atomo, il cui numero è regolato dal controllo sul frame rate, i punti di partenza delle linee di forza saranno

definiti dai punti in corrispondenza di $\sqrt{m}+1$ stacks e $\sqrt{n}+1$ slices; in atl modo si ottiene anche un controllo sul frame rate per quanto riguarda le linee di forza.
Si è deciso di separare la rappresentazione delle linee di forza dalle rispettive direzioni per cercare di mantenere un frame rate e un dettaglio grafico più elevato all'occorenza. Infatti l'introduzione dei vettori direzionali riduce sensibilmente il frame rate e riduce quindi il numero di linee effettivamente disegnate. Si è ritenuto quindi opportuno rinviare la scelta di visualizzare o meno la direzione delle linee di forze all'utente.
Vediamo ora nella nostra trattazione le modalità con cui si sono calcolate e disegnate tali linee di forza. Per ogni linea di forza:

1. si individua il punto di inizio della linea in base ai punti equispaziati sulla molecola disegnata di cui si è parlato in precedenza;

2. si controlla il gradiente del potenziale in tale punto;

3. si segue tale gradiente per una distanza definita pari a un decimo di Angstrom, se il punto di arrivo è dentro la molecola fermati, altrimenti ricomincia da tale nuovo punto;

4. si continua a seguire il gradiente finché:

   (a) si finisce fuori dalla griglia di potenziale;

   (b) il modulo del gradiente è inferiore a 0.00001;

   (c) il gradiente oscilla attorno ad un punto, il che significa che a due passi successivi si ritorna ad un punto che dista meno di un decimo di Angstrom dal pre-predecessore dello stesso;

   (d) si è ragginta l'iterazione numero 500.

5. si torna ad 2 con un nuovo punto iniziale.

Inoltre in caso di visualizzazione della direzione ogni 30 punti si disegna un vettore con la direzione del gradiente, lungo un terzo di Angstrom, che risulterà quindi tangente alla curva.
Questa visualizzazione permette di prevedere efficacemente le traiettore che il PHANToM dovrebbe naturalmente seguire in caso il cursore venga posto in corrispondenza della superficie della molecola e rappresenta uno strumento potente per aiutare l'utente nei compiti che potrà svolgere con lo strumento realizzato.

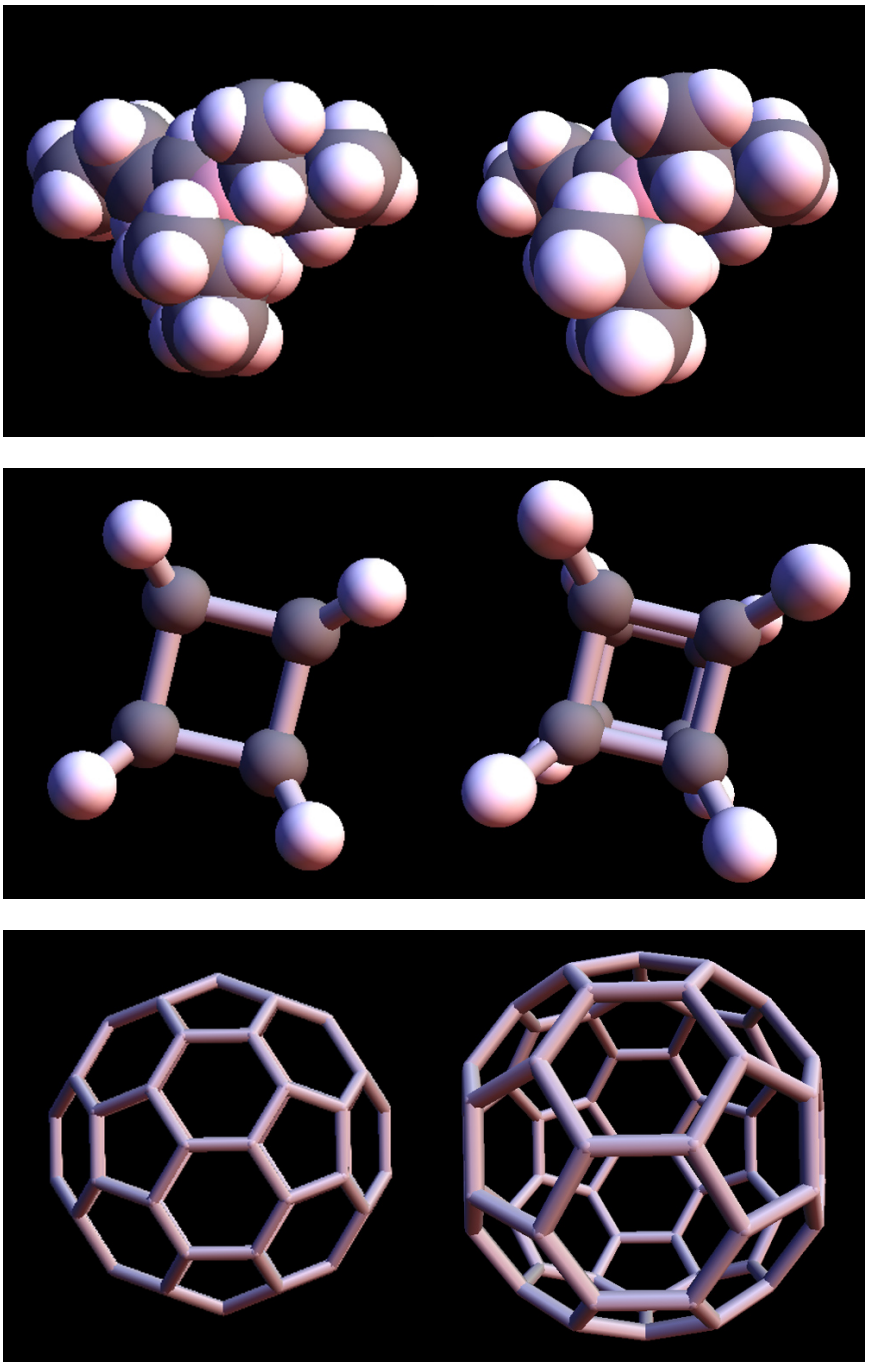

Figura 4.26: differenze tra camera ortogonale e prospettica

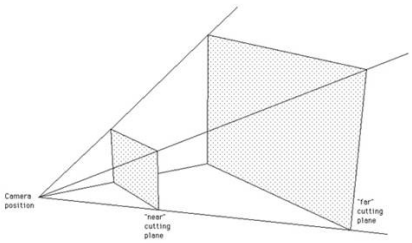

Figura 4.27: rappresentazione viewing frustum

4.3.8 Proprietà della camera

Esistono due tipologie di viasualizzazione, ortogonale e prospettica. Tra le due modalità di rappresentazione esistono notevoli differenze dal punto di vista della resa grafica, come si può ben vedere in figura 4.26 dove della stessa molecola vengono mostrati a sinistra la visualizzazione ortogonale, a destra quella prospettica.
Procediamo con ordine nella analisi delle caratteristiche delle due rappresentazioni:

- **Ortogonale**: la visualizzazione ortogonale è caratterizzata da un viewing volume a forma di parallelepipedo e il cui centro di proiezione (COP dall'inglese) si trova all'infinito generando solo una direzione di proezione (DOP). Questa visualizzazione non modifica l'immagine in base alla posizione dell'oggetto sull'asse z in quanto l'immagine viene schiacciatasul piano xy che contraddistingue il piano di proiezione. Tale caratteristica permette però alla scena 3d di mantenere uguale la dimesione di atomi dello stesso tipo indipendentemente dal valore z dei loro centri, in questo modo ha senso assegnare una misura di scala che ne indichi le dimensioni relative, poiché le lunghezze apparenti sono proporzionali alle lunghezze reali. Nel sistema sviluppato il viewing volume è un cubo di lato 420 caratterizzato dai punti $(-210, -160, 210)$ e $(210, 260, -210)$, la disparità dell'asse y rispetto agli altri due è dovuto alla preseza nella scena 3d del riquadro di servizio in alto.

- **Prospettica**: tale visualizzazione, invece, presenta un viewing volume a forma di frustum, tronco di piramide, figura 4.27. Questo frustum è contraddistinto da un COP (Center of projection) e da due piani di clipping (taglio) che delimitano l'area di visione, nonché da un apertura focale, che indica quanto deve essere ampio il tronco di cono. In questa modalità di rappresentazione si ha il cosìdetto effetto prospet-

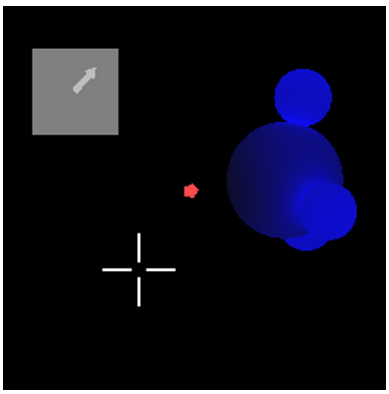

Figura 4.28: vettore indicante la forza sul cursore aptico

tico, in quanto oggetti più vicini alla camera, localizzata nel COP, appaiono più grandi di quelli distanti; ciò, sebbene distorga le proporzioni tra i vari atomi, permette di avere una maggior percezione della profondità e permette di avere un aspetto più realistico (i nostri occhi vedono prospetticamente). Nel nostro caso specifico il COP è posto in posizione $(0.0, 0.0, 250)$ il clipping plane più vicino dista 20, mentre quello lontano 500 e l'apertura focale è definita dal rettangolo di taglio del frustum al clipping plane vicino i cui estremi sono, in coordinate (x, y), $(-10.0, -10.0)$ e $10.0, 10.0$.

Sebbene inizialmente fosse presente nel sistema solo la versione ortogonale di visualizzazione, che permetteva la presenza di una metrica, è stato deciso di introdurre la visione prospettica per ovviare al problema di localizzazione del cursore aptico nello spazio rispetto alla molecola, nella visione ortogonale, infatti, è praticamente impossibile rendersi conto della posizione del cursore e della molecola sull'asse z questo crea lo sconveniente di non rendere immediatamente percepibile la posizione relativa del cursore aptico rispetto alla molecola rappresentata. Nella visualizzazione prospettica, invece, si ha modo di riconoscere l'asse z in base alla grandezza degli oggetti rappresentati e ciò favorisce l'interazione dell'utente con il sistema.

4.3.9 Vettore forza e grafico potenziale

Esistono due ulteriori strumenti grafici legati alla posizione del cursore aptico, il disegno della direzione cui il vettore forza del cursore è sottoposto e il disegno del grafico di potenziale tra il cursore stesso e il nucleo dell'atomo più vicino.

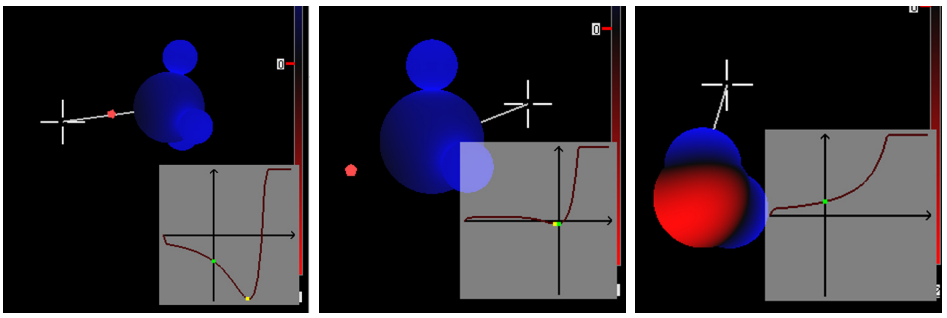

Figura 4.29: grafo di potenziale

In figura 4.28 è mostrato, in alto a sinistra il vettore indicante la direzione della forza cui il cursore è sottoposto. Inizialmente tale vettore era collegato direttamente al cursore aptico e indicava anche il modulo con la lunghezza di tale vettore, poi, sotto consiglio di designer a cui ci siamo rivolti per una consulenza sull'interfaccia è stato deciso di decentrare tale vettore e collocarlo esternamente al cursore e di individuare con tale vettore la sola direzione.
La scelta relativa alla rappresentazione del vettore solo come direzione è derivata dalla ampia variabilità che contraddistingue il sistema quando si tratta di valori di potenziale o di forza, si è gia visto, infatti, come le forze attrattive siano estremamente piccole rispetto alle forze repulsive, fatto che ha portato all'introduzione delle scale logaritmiche; risulta quindi di scarso impatto grafico l'assenza del vettore, troppo corto per essere visualizzato, nel caso di forza attrattiva, e l'eccessiva lunghezza dello stesso, tale da uscire dal viewing volume, nel caso di forze repulsive.
Invece la scelta relativa al decentramento del vettore deriva dal cambiamento che è stato approntato al cursore, inizialmente una piccola sfera, tramutatosi in un mirino. Risulta avere poco senso associare un vettore graficamente ad un mirino, per cui si è deciso di spostarlo e renderlo fisso, in modo da essere sempre individuabile e da agire come 'bussola' all'interno del campo di potenziale.

In figura 4.29 è mosrato il secondo strumento legato al cursore aptico: il grafo di potenziale. In primo luogo analizziamo le immagini presentate, rappresentanti una molecola di NH3:

1. in questa prima immagine è visualizzabile il tratto di unione che congiunge il cursore all'atomo più vicino, l'azoto centrale. Su tale linea viene calcolato il potenziale e visualizzato sotto forma di grafico. Si può facilmente notare che la linea pasas dal minimo, per cui il pallino giallo sul grafo di potenziale indica un punto di minimo nella buca di

potenziale. L'incrocio degli assi invece rappresenta la posizione del cursore rispetto al centro della molecola, posto in fondo al grafico, mentre il pallino verde è la proiezione delcursore sulla linea di potenziale di colore rosso scuro.

2. in questa seconda immagine è mostrato sul grafo un minimo locale, sempre mostrato in giallo. Si è in grado di distinguere il fatto che si tratti di minimo locale perché il grafo è scalato in modo che l'asse y è delimitato dai valori $[-m, +m]$ dove m indica il minimo di poteziale assoluto, in tal modo nel grafico un punto è di minimo assoluto solo se giunge ai limiti inferiori del grafico.

3. nell'ultima immagine, infine, notiamo un grafico di potenziale in cui non sono presenti buche di potenziale, per cui il cursore verrà respinto dall'atomo.

Da notare che il grafico rappresenta bene anche il comportamento del gradiente del potenziale, risulta infatti immediato desumere dal grafico la direzione in cui il cursore sarà portato ad essere spinto; infatti il cursore aptico tende a seguire la discesa del gradiente in cerca dei minimi, ecco perché le buche di potenziale rappresentano dei punti in cui il cursore trova la sua stabilità.
Altro elemento da notare è il fatto che i valori positivi tendono sempre a saturare il grafico, andando facilmente fuori scala, ciò avviene sempre per lo stesso motivo differenze di ordine di grandezza enormi tra potenziale positivo e negativo. Inoltre, a differenza del progetto originario, il grafico si ferma in prossimità del nucleo dell'atomo, che rappresenta anche un punto di massimo; nella versione da cui si è partiti il grafico procedeva anche oltre, si è deciso di troncare il grafico in quel punto perché il nucleo dell'atomo rappresenta un punto limite del potenziale, per cui una carica di prova positiva non è possibilitata a raggiungerne le immediate prossimità e rappresenterebbe un controsenso farle raggiungere la parte opposta dell'atomo nella medesima direzione.
Il grafico di potenziale descrive, quindi, tutti i punti di potenziale sulla congiungente tra cursore e nucleo dell'atomo più vicino, nonché i punti sulla stessa direzione prima del cursore stesso finché non si trova un valore di potenziale inferiore ad una certa soglia; ciò permette di individuare non solo cosa succede al cursore avvicinandosi all'atomo, ma anche allontanandosi dallo stesso.

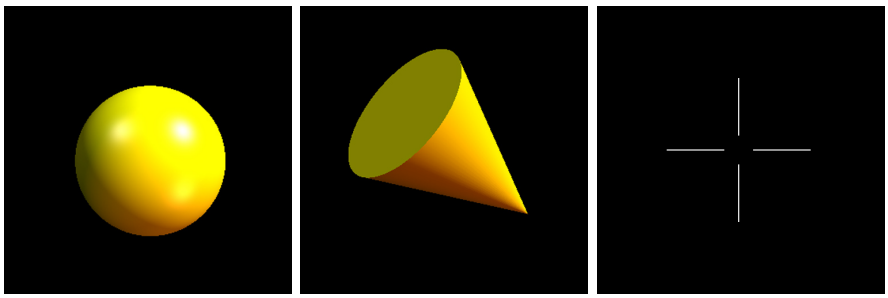

Figura 4.30: Vari cursori aptici introdotti

4.3.10 Cursori aptici

Durante lo sviluppo del software sono stati introdotte tre diverse tipologie di cursore aptico, di cui solo due sono rimaste nella versione definitiva, figura 4.30. Ricordiamo che il cursore aptico è il corrispettivo del puntatore del mouse rispetto allo strumento aptico, rappresenta quindi la posizione del end-effector del PHANToM in uno spazio 3d.
Nello specifico i 3 cursori aptici hanno le seguenti caratteristiche:

1. **Sferico**: è il primo cursore introdotto, ereditato direttamente dalla prima versione del software, di semplice realizzazione, una piccola sfera. Tale cursore è stato rimpiazzato nelle fasi finali dai due successivi.

2. **Freccia 3d**: è la diretta evoluzione del cursore precedente, permette di mostrare a schermo 6 DOF rispetto ai 3 della sfera: la posizione della freccia indica i tre traslazionali, mentre la direzione della stessa i tre rotazionali. Rispetto al cursore precedente permette di rispecchiare meglio la punta dell'end effector del PHANToM e quindi di facilitare la navigazione da parte dell'utente che fa meno fatica a colmare il gap tra strumento e scena 3d. Questo cursore viene utilizzato per l'esplorazione geometrica della molecola, non essendo adatto a rispecchiare una carica di prova. Per essere disegnato necessità di un'informazione in più rispetto alla sola posizione dell'end effector nello spazio 3d, tale informazione viene ottenuta applicando la trasformazione, sotto forma di matrice 4×4, che porta dall'origine del sistema aptico alla rototraslazione del l'end effector.

3. **Mirino**: questo cursore aptico è stato introdotto per sostituire il pallino nella navigazione del MEP, sotto consiglio dei designer. Il pallino forniva infatti un'idea fuorviante di quello che doveva essere rappresentato, cioè una carica di prova, rendendola graficamente come se fosse

effettivamente una particella reale con una dimensione che avesse significato rispetto agli atomi mostrati. Tale malinteso forniva all'utente un'idea sbagliata di quanto può succedere nella realtà avvicinando un protone reale, invece di una carica di prova. Pertanto il mirino risulta più intuitivo come esploratore del potenziale attorno alla molecola.

I cursori presentavano un ulteriore problema in merito alla visualizzazione ortogonale nella scena. Senza accorgimenti la dimensione del cursore non variava muovendo lo strumento rispetto all'asse z, in tal modo non era possibile stabilire, visualizzando la scena 3d, la posizione precisa dello stesso. Per ovviare a tale problema è stato introdotto un mezzo artificioso che, proporzionalmente alla coordinata z del puntatore aptico, ne variasse le dimensione, in modo da simulare un effetto prospettico in grado di permettere all'utente di orientarsi maggiormente nello spazio 3d.

4.4 Engine Aptico

Nel sistema sviluppato esistono due principali interazioni aptiche: con la geometria della molecola e con il campo di forze derivante dal MEP. Queste due modalità di interazioni fanno capo ognuna ad una differente libreria del pacchetto OpenHaptics, HLAPI per la geometria e HDAPI per il campo di forza, questo per le caratteristiche delle due API che agevolano un compito piuttosto che l'altro.
Inoltre nel sistema è stato introdotto un metodo di selezione degli atomi all'interno della molecola attraverso lo strumento aptico e la successiva possibilità di rototraslare la stessa direttamente con il PHANToM.
Infine è sata introdotta la possibilità di 'spingere' la molecola attraverso l'uso del PHANToM, di questo argomento si tratterà però nella sezione relativa all'engine dinamico.
Ma procedendo con ordine andiamo a spiegare suddetti concetti nelle sotto sezioni dedicate.

4.4.1 Percezione geometrica della molecola

Come già accennato la renderizzazione aptica della geometria della molecola fa uso quasi esclusivo della libreria HLAPI. Attraverso tale strumento è possibile renderizzare le forze da inviare al phantom per dare l'illusione di contatto con la superficie della molecola direttamente dalla dichiarazione delle primitive grafiche OpenGL, come già ampiamente discusso nel capitolo 3. Quindi per rendere apticamente la forma di un oggetto 3d basta impostare correttamente il sistema e disegnare le forme attraverso le OpenGL, avendo

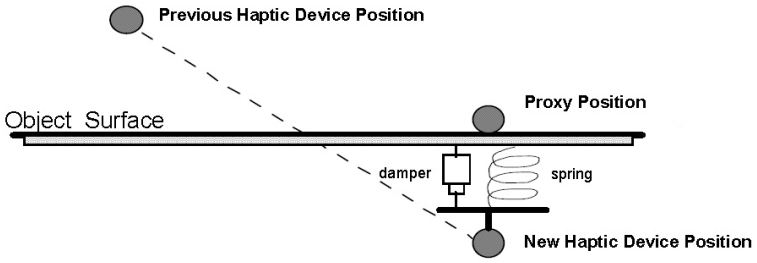

Figura 4.31: schema di funzionamento del proxy

però alcune accortezze che andremo a discuere.
Ma cosa significa rendere apticamente la superficie di un oggetto? L'illusione del tocco viene creata dalla resistenza che lo strumento impone alla penetrazione di una superficie. Per visualizzare tale comportamento basta bensare ad una molla e ad un un ammortizzatore attaccati alla superficie, una volta superato il limite con il cursore aptico la molla e l'ammortizzatore generano una forza proporzionale allo spostamento e alla velocità del cursore in direzione opposta a tale spostamento, quindi in direzione normale e uscente dalla superficie. Questo comportamento è illusrato in figura 4.31. In pratica non si renderizza la superficie come qualcosa di rigido completamente impenetrabile, bensì si genera una forza opposta alla penetrazione. Nella figura 4.31 viene introdotto un altro concetto fondamentale per la percezione della forma di un oggeto, il concetto di proxy. Per proxy si intende un'immagine del cursore aptico falsata rispetto alla posizione reale, in modo da permanere sulla superficie dell'oggetto da percepire nonostante il device lo abbia penetrato. Il proxy rappresenta più un'entità grafica che aptica, ma viene gestita direttamente dalla HLAPI. Lo scopo di tale oggetto è creare l'illusione che il device non possa entrare all'interno dell'oggetto (nel nostro caso una molecola), rappresentandolo sempre al di fuori della superficie, questo rappresenta un'illusione che combinata alla sensazione di forza generata dallo strumento aptico permette all'utente di percepire la forma della nostra molecola. Il proxy va quindi a sostituire il device reale ogni qualvolta vi sia un contatto con l'oggetto rappresentato.
La HLAPI dunziona intercettando le chiamate OpenGL e mantenendo in memoria la struttura di ciò che viene rappresentato, in modo che il Collision thread sia in grado di identificare se vi sia stato un contatto o meno con tali superfici. Esistono due modi in ci è possibile intercettare le chiamate OpenGl:

- contestualmente alle chiamate grafiche: sebbene più veloce dell'altro

metodo presenta un grande svantaggio, non permette di rappresentare le superfici non visibili, cio poiché i vertici selezionati dal sistema sono già stati ridotti dalle varie tipologie di culling.

- a posteriori tramite la duplicazione dei comandi grafici: intercettando direttamente tutte le chiamate non permette di disegnare a video contemporaneamente alla renderizzazione aptica, però consente di renderizzare tutte le superfici nascoste all'osservatore in modo che possano essere 'toccate'.

Per ovvi motivi nel sistema qui presentato è stata scelta la seconda soluzione. Riassumendo, l'interazione aptica con la geometia del sistema si ottiene duplicando i comandi grafici all'interno di un contesto aptico. Ciò che viene rappresentato graficamente viene rappresentato apticamente.
Esistono però delle variazioni nel modo in cui questi elementi grafici vengono renderizzati, se la geometria della molecola viene renderizzata sotto forma di contatto, per cui viene percepita una forza solo quando il contatto è effettivamente avvenuto, metodologia della figura 4.31, altri parametri come i minimi di potenziale vengono renderizzati diversamente. Nella sottosezione dedicata 4.4.1 verrà illustrata la diversa metodologia.
Oltre alla percezione della geometria della molecola venbgono introdotte anche alcune proprietà della sua superficie, legati a parametri chimici degli atomi che compongono la molecola stessa. In 4.4.1 sono illustrate tali caratteristiche.

Renderizzazione aptica minimi nel contesto geometrico

I minimi, nel contesto geometrico, vengono renderizzti come punti attrattivi del cursore aptico: una volta che il cursore oltrepassa una distanza soglia dal minimo assoluto di potenziale, lo strumento viene attratto nel punto di minimo, e ne rimane agganciato finché non si riporta il cursore oltre la distanza minima vincendo la forza di attrazione. I punti di minimo si comportano quindi come dei punti di aggancio di una molla a cui il cursore rimane attaccato se oltrepassa la distanza minima, tale molla genererà una forza attrattiva verso il punto di minimo che diventa la posizione di riposo della stessa, in tal modo se l'utente vuole uscire dal minino deve vincere la forza elastica che lega il cursore al punto in questione. Questo genere di forze viene denominato Coulombiane in quanto agiscono come due cariche di segno opposto che tendono ad attrarsi.
La forza generata da questo tipo di sistema è $F = -kd$ dove k è la costante elastica della molla di cui si è parlato in precedenza, parametro definito per

il punto di minimo, e d è la distanza tra il punto di minimo e il cursore aptico. La forza viene attivata se $d < s$ dove s è una soglia definita come parametro del sistema.

Proprietà delle superfici aptiche

Nel sistema vengono definite due proprietà dipendenti dalle caratteristiche degli atomi che compongono la molecola: durezza e attrito dinamico. Analizzandole una ad una:

- **durezza**: questo parametro definisce la morbidezza apparente della superficie aptica, la costante k della molla che lega il cursore alla superficie. Chimicamente è legato alla massa atomica dell'atomo rispetto alla massa atomica del più pesante all'interno della molecola. Tale valore varia da 0.15 a 0.95 per evitare valori limite che potrebberp annullare o rendere eccessivo l'effetto. Inizialmente si era pensato di legare questa proprietà alla densità atomica, pultroppo la libreria OpenBabel non possiede questo tipo di informazioni per cui si è ripiegato sulla massa.
- **attrito dinamico**: questo secondo parametro invece definisce la ruvidità della superficie, cioé quanto la superficie si oppone allo scorrimento del cursore sopra di essa stessa, rappresenta quindi una forza superficiale che si oppone al moto del cursore di forma $F_a = k_d F_n$ dove k_d è il coefficiente d'attrito e F_n e la forza perpendicolare alla superficie che il cursore applica. Questo parametro è legato all'elettronegatività, che assume valori tra $(0 - 4]$ ed indica la facilità con cui cede(valori bassi) o acquista (valori alti) elettroni dagli altri atomi. Si è pensato di legarlo all'idea di attrito dinamico per richiamare l'idea di strappare elettroni, una superficie ruvida tende a trattenere a se gli elettroni e a strapparli al cursore, mentre una superficie liscia dona facilmente elettroni. Si è cercato quindi di introdurre una metafora che permettesse di comprendere al meglio il significato di elettronegatività, legando il concetto ad una proprietà aptica della molecola.

Riteniamo che l'introduzione di tali caratteristiche ulteriori possano agevolare lo studente nella comprensione della chimica molecolare in quanto l'elettronegatività degli atomi incide spesso sulla formazione di legami e sulla natura della molecola stessa.

4.4.2 Selezione aptica di un atomo

La selezione dell'atomo dal punto di vista aptico è stata introdotta per dare modo all'utente di controllare la rototraslazione della molecola direttamente

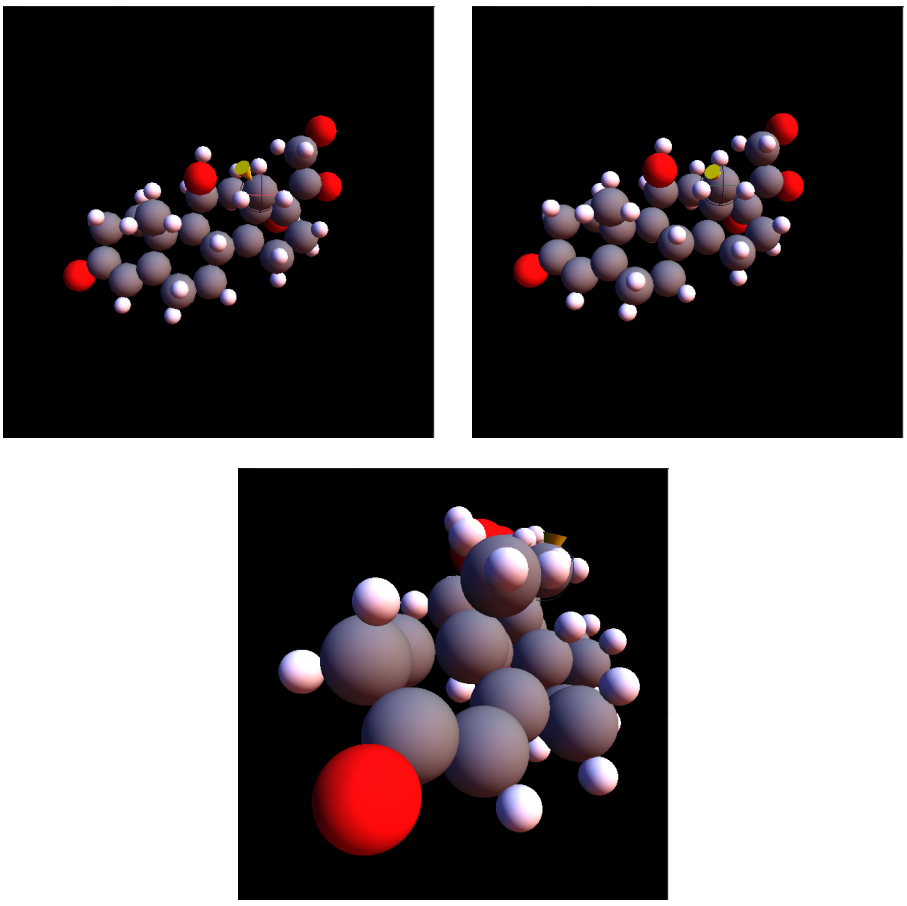

Figura 4.32: selezione aptica e rototraslazione

con l'interfaccia aptica. Si tratta nello specifico di un metodo grazie al quale l'utente seleziona l'atomo da agganciare che diventerà il centro della rototraslazione, dopodiché ruota e trasla la molecola spostando il cursore nello spazio3d.

In figura 4.32 è mostrato l'intero procedimento:

1. in primo luogo si arriva a contatto con la molecola, evidenziandola con un colore rosso. Il processo di preselezione avviene verificando per ogni frame se la distanza tra il cursore aptico e l'atomo più vicino sia inferiore del raggio di quest'ultimo maggiorato del 5%, in caso affermativo l'atomo è preselezionato.

2. secondariamente si preme il pulsante sul PHANToM e viene attivato il processo di selezione e agganciamento vero e proprio:

   (a) Il nucleo dell'atomo preselezionato diventa attrattivo, per cui il cursore aptico è attratto dal nucleo e l'atomo viene evidenziato di giallo;

   (b) una volta che la distanza tra nucleo e cursore diventa inferiore a un decimo Angstrom l'atomo viene agganciato e viene evidenziato di verde;

   (c) viene salvata la matrice di rotazione A della molecola.

3. a questo punto ad ogni rotazione e traslazione dello strumento aptico viene fatta corrispondere una rotazione della molecola. Al passo precedente viene salvata in memoria A in modo da gestire la rotazione ulteriore con lo strumento a partire da quella condizione, viene salvata anche la matrice B che identifica la rotazione dello strumento aptico nel momento dell'agganciamento. A questo punto B viene invertita e moltiplicata per A, al fine di ottenere $C = AB^t$ [1]. C rappresenta la matrice di partenza di tutte le successive rotazioni dello strumento aptico, quindi ad ogni frame si prende D, la matrice di rotazione dello strumento aptico, e la si moltiplica a C ottenendo la rotazione finale da assegnare alla molecola $E = CD = AB^tD$. A questo viene aggiunta la traslazione che più semplicemente passa direttamente la posizione del cursore aptico al punto di riferimento della molecola.

Al rilascio del pulsante la molecola viene sganciata e si ritorna alla modalità normale, avendo cura di traslare l'ultima volta la molecola in modo da correggere l'errore dovuto al cambiamento di centro di rotazione, che passa dal nucleo dell'atomo selezionato al baricentro. Per effettuare questa correzione semplicemente si trasla la molecola di una quantita $\vec{A}$ pari alla distanza tra $\vec{B}$ che indica il baricentro e $\vec{C}$ che indica il centro dell'atomo selezionato, cioè $\vec{A} = \vec{B} - \vec{C}$.
Per realizzare la selezione aptica si è fatto uso di tre funzioni di callback:

- una funzione di callback per gestire la pressione del pulsante, se un atomo è preselezionato dice al sistema aptico di rendere attrattivo il nucleo dell'atomo e attiva la seconda funzione di callback;

- una seconda per gestire la rototraslazione della molecola rispetto alla rototraslazione dello strumento aptico, viene richiamata ogni qualvolta il cursore si muove e nonostante vengaavviata dalla precedente si attiva solo al raggiungimento del nucleo dell'atomo da parte del cursore;

[1] Si noti che essendo B una matrice di rotazione ortogonale $B^{-1} = B^t$ cioè la matrice inversa coincide con la sua trasposta

- una terza funzione per gestire il rilascio del pulsante, cancella la funzione di callback precedente e sistema la traslazione finale rilasciando la molecola.

La selezione aptica rappresenta uno strumento potente per l'utente per controllare direttamente la posizione della molecola on modo da visualizzarne correttamente la struttura.

4.4.3 Percezione del force field attorno alla molecola

Seconda modalità di interazione aptica, e probabilmente più importante, è rappresentata dalla percezione del campo di forze generato dal MEP attorno alla molecola. Tale percezione avviene tramite varie modalità che verranno illustrate a breve e fa uso esclusivo della libreria HDAPI.
In origine era presente un unico modo per percepire il potenziale attorno alla molecola, che consisteva nell'individuare a quale cella della mappa del gradiente del MEP corrispondesse la posizione del cursore aptico e a inviare direttamente tale gradiente sul device aptico, dopo averlo moltiplicato per una costante che individuava la carica del cursore. Questa modalità presentava grosse lacune però, come presentato nella sezione 4.1, in quanto permetteva all'utente di distinguere solo tra forze attrattive e repulsive, tagliando tutta la variabilità dei potenziali positivi, inoltre presentava instabilità nei punti di minimo. Nonostante questa modalità diretta sia stata mantenuta nel sistema, è stata apportata una grande modifica che permette di risolvere almeno uno dei due problemi presentati, al posto di assegnare direttamente il valore del gradiente presente nella cella della griglia si è provveduto a interpolare il gradiente con le celle contigue, in modo da ottenere una certa continuità nel campo di forza che permettesse di attenuare i contrasti esistenti nei dintorni del minimo tra celle contigue. Il processo di interpolazione verra spiegato profusamente nella sottosezione 4.4.3.
Oltre a tale modalità di percezione aptica è stata accostata, come si è già visto in altre parti del sistema, la variante logaritmica. Nel caso aptico si tratta di riscalare le forze presenti nel sistema sulle forze erogabili dal device aptico. Usando le varie scale nel sistema ci si accorge che è necessario un grado elevato di scala iperlogaritmica per permettere all'utente di percepire forze attrattive verso i minimi di potenziale. In figura 4.33 sono mostrati vari livelli di forza attorno al minimo in base al grado della scala iperlogaritmica scelta:

1. grado **0**, scala lineare: si può vedere facilmente come tutte le forze sono prossime a 0, anche quelle più repulsive dell'immagine, rappresentate

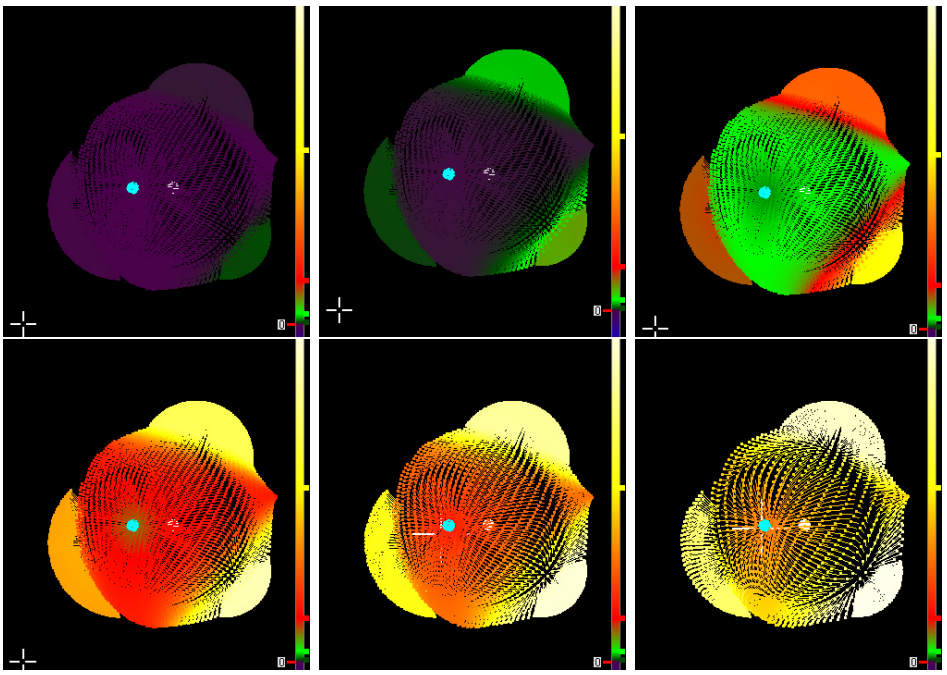

Figura 4.33: scale logaritmiche per la forza

dall'atomo in basso a destra; con la scala lineare è infatti possibile appena sentire le forze repulsive nelle immediate prossimità dei nuclei.

2. grado **1**, scala logaritmica: lievi miglioramenti per quanto riguarda le forze repulsive, che allargano la loro sfera di influenza, rimanendo sempre però piuttosto deboli se non nelle vicinanze dei nuclei degli atomi; le forze attrattive sono invece ancora praticamente nulle.

3. grado **10**, scala iperlogaritmica: le forze repulsive diventano percepibili nella loro interezza, il punto di massimo nell'immagine raggiunge la metà superiore della scala, le forze attrattive invece, diventano appena percepibili.

4. grado **50**: le forze attattive diventano percepibili, sebbene piuttosto deboli, mentre le forze repulsive sono molto vicine ai massimi.

5. grado **100**: le forze attrattive sono a questo punto consistenti e permettono di mantenere il cursore nella zona di minimo, anche in assenza dell'utente, comunque raggiungono appena un terzo della scala.

6. grado **200**: infine, al grado più alto della scala, le forze attrattive raggiungono circa la metà della stessa diventando percepibili nella loro interezza.

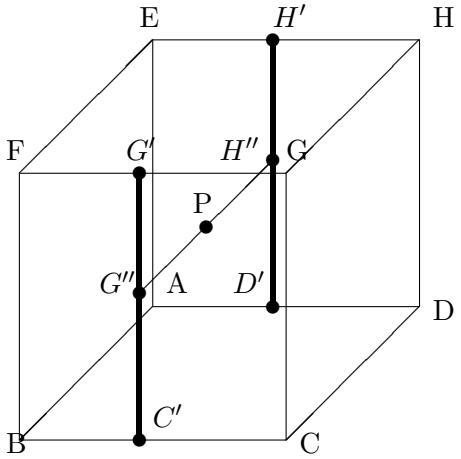

Figura 4.34: punti per il calcolo dell'interpolazione 3d

L'introduzione delle scale logaritmiche applicate all'aptica del sistema permette all'utente di percepire e di assimilare non solo la presenza di forze attrattive e repulsive, ma anche la loro entità, riuscendo così a discriminare tra le varie forze in gioco.
Ulteriore modifica alla percezione aptica del potenziale è stata introdotta in merito alla possibilità di usare altre modalità per il calcolo della forza. All'interno del sistema è stata infatti introdotta la modalità coulombiana, per cui la forza da assegnare al PHANToM viene calcolata attraverso una formula, invece che attraverso la mappa di potenziale, di queste formule si parlerà più avanti nella sezione dedicata. Pultroppo a queste nuove forme di calcolo della forza non possono essere associate le scale logaritmiche, essendo impossibile determinare dei valori minimi e massimi per il campo di forze, che in certi punti raggiunge ∞, tuttavia è possibile controllare la carica di prova con il pulsante apposito nell'interfaccia per navigare il campo di forze generato dalle varie formule introdotte.
Le innovazioni prodotte nel contesto della navigazione del force field rappresentano un notevole miglioramento della percezione aptica del sistema capace ora di renderizzare diverse tipologie ed entità di forze.

Interpolazione 3d

Il processo di interpolazione 3d, i cui punti salienti sono riassunti in figura 4.34, consiste nell'interpolare linearmente i valori di potenziale, o i vettori del grdiente primo e secondo, sulle 3 dimensioni spaziali in modo da rendere continuo lo spazio. I punti A, B, C, D, E, F, G e H rappresentano 8 valori contigui presenti sulla griglia di potenziale, mentre P rappresenta il

punto di cui vogliamo valutare l'informazione utile. il procedimento ha luogo seguendo queste fasi, di cui apporteremo anche un semplice esempio:

- Innanzitutto si selezionano gli 8 punti del potenziale da cui vogliamo interpolare i valori, e valutiamo il potenziale in quei punti, ($A_p = 1.0, B_p = 2.0, C_p = 3.0, D_p = 4.0, E_p = 5.0, F_p = 6.0, G_p = 7.0, H_p = 8.0$).

- in secondo luogo valutiamo la distanza di P rispetto ad A in ognuna delle 3 direzioni (x, y, z), ipotizziamo che sia esattamente nel centro del cubo di lato 1 per cui ($d_x = 0.5, d_y = 0.5, d_z = 0.5$).

- a questo punto calcoliamo i quattro punti C' ,G' ,H' e D' interpolando linearmente sull'asse x i quattro segmenti $\overline{AD}$, $\overline{BC}$, $\overline{FG}$ e $\overline{EH}$ e i loro valori, quindi le coordinate di C' ad esempio saranno ($B_x + \frac{d_x}{C_x - B_x} = P_x, B_y, B_z$). Da notare che $D_x - A_x = C_x - B_x = G_x - F_x = H_x - E_x$ per cui chiameremo questa distanza semplicemente δ_x. Risulta quindi semplice definire i valori dei 4 punti come $C'_p = B_p + \frac{(C_p - B_p)d_x}{\delta_x}$[2] e così via per gli altri 3 punti. Nel nostro esempio verranno quindi ($C'_v = 2.5, G' = 6.5, H' = 6.5, D' = 2.5$).

- al passaggio successivo si calcolano G'' e H'' interpolando la y sui due segmenti $\overline{C'G'}$ e $\overline{D'H'}$ e rispettivi valori di potenziale. Notando che $G'_y - C'_y = H'_y - D'_y = E_y - A_y = \ldots$ chiameremo questa quantità δ_y. Le coordinate del punto G'' e il rispettivo valore saranno rispettivamente ($P_x, C'_y + \frac{d_y}{\delta_y} = P_y, G'_z$) mentre $G''_p = C'_p + \frac{(G'_p - C'_p)d_y}{\delta_y}$, mentre per H'' si agirà di conseguenza. Nell'esempio ($G''_v = 4.5, H''_v = 4.5$).

- infine si calcola il punto P e il suo valore interpolando la z sul segmento $\overline{G''H''}$. Ricordiamo $G''_z - H''_z = B_z - A_z = \ldots$ per cui chiameremo questa quantità δ_z. Il valore del punto P sarà quindi $P_p = H''_p + \frac{(H''_p - G''_p)d_z}{\delta_z}$. Per finire il nostro esempio $P_v = 4.5$.

Per l'interpolazione del gradiente si opera con lo stesso metodo, ma al posto dei valori X_p si useranno dei vettori e delle somme pesate di vettori.
Sebbene piuttosto complessa da calcolare l'interpolazione permette di rendere sia la grafica del potenziale che la forza erogata dal PHANToM più fluide e naturali, in tal modo si risolve il problema dell'instabilità nei minimi di potenziale e la colorazione delle molecole appare continua [3].

[2] $d_x = P_x - A_x = P_x - B_x = \ldots$

[3] a differenza della versione del prototipo in cui era possibile vedere delle 'chiazze' di colorazione del MEP

4.5 Engine Sonoro

Il sistema da noi progettato e sviluppato fa uso, in aggiunta ad un sistema visuale ed aptico, di stimoli audio per convogliare le informazioni all'utente e per permettere una migliore immersività.
L'interfacciamento sonoro si articola in diversi interventi distinti:

- Il primo, e più semplice, definisce degli effetti sonori per l'interfaccia delle opzioni del sistema. Tali effetti sono stati inseriti per permettere all'utente di avere un indizio audio ogni qualvolta venisse cambiato qualche parametro dell'applicazione, in modo da evitare cambiamenti non voluti e non appropriatamente segnalati della realtà simulata. Questo approccio si basa sulla considerazione che l'utente è maggiormente in grado di accorgersi di cambiamenti in un ambiente virtuale se essi sono accompagnati da un segnale sonoro che ne indichi l'avvenimento.

- Secondo metodo di interfacciamento sonoro è rappresentato dalla definizione di uno spazio tridimensionale riconoscibile con il solo stimolo audio. Per tale approccio ci si è ispirati all'acusmetria definita in [36]. Questo sistema consiste nel mappare le tre dimensioni spaziali a tre diverse proprietà del suono; nel nostro caso abbiamo associato l'effetto stereo del suono alla dimensione spaziale X, la frequenza del suono alla Y, mentre il guadagno e la potenza del suono all'asse Z. In tale modo un utente opportunamente addestrato può essere in grado di percepire la posizione di un punto nello spazio grazie al solo stimolo audio. Nello specifico questo sistema è stato usato in due distinte fasi del programma:

  - In primis nella selezione di un atomo nella molecola attraverso l'uso del mouse. Il suono relativo al passaggio del mouse, alla selezione e alla de selezione di un atomo viene prodotto in base ala posizione del nucleo dello stesso.
  - Il secondo caso individua invece lo sfregamento, attraverso l'interfaccia aptica, della superficie della molecola, in modo da mantenere un costante sistema di riferimento uditivo della posizione assoluta della molecola.

- Come terzo sistema viene proposto un particolare effetto audio al tocco della geometria della molecola tramite l'interfaccia aptica. Tale suono dipende dalle caratteristiche dell'atomo toccato e nello specifico dalla sua elettronegatività, permettendo di avere un feedback sonoro su una

qualità del sistema. In un certo senso tale utilizzo dell'audio richiama quanto visto in [35].

- quarto effetto generato è stato uno stimolo sonoro da accostare alla percezione del campo di forze attraverso l'interfacciamento aptico. Di questa metodologià di interfacciamento sonoro ne sono state fatte due versioni distine:

  - il primo sistema consisteva nel produrre un fischio la cui frequenza aumentava con l'aumentare della forza, in tal modo l'utente aveva un doppio feedback rispetto alla forza percepita: all'aumentare della forza erogata dal PHANTOM, aumentava anche l'altezza in frequenza del suono generato. In questo modo l'utente era maggiormente conscio delle forze in gioco, potendole percepire attraverso tre distinti sensi.
  - nella fase finale del progetto invece la frequenza del suono percepito è stata assegnato al potenziale invece che alla forza, in tal modo mentre attraverso lo strumento aptico si è in grado di percepire la forza attraverso il suono si sente l'entità di potenziale nel punto in cui il cursore aptico è situato nello spazio 3d. Nello specifico si è pensato di associare al potenziale positivo e negativo due suoni dal timbro diverso e di aumentare la frequenza di questi suoni in relazione al modulo del potenziale, fino a diventare 0 in ogni punto in cui il potenziale è particolarmente basso. Per poter essere percepito correttamente anche questo sistema sonoro fa uso delle scale logaritmiche per determinare l'entità del suono prodotto, se infatti si utilizzasse la sola scala lineare risultano estremamente poche le aree in cui il potenziale positivo emetterebbe un suono percepibile. Si è deciso di sostituire con questo sistema il precedente per non duplicare l'informazione ricevuta dall'utente, attraverso la novità introdotta è, infatti, in grado di percepire in un colpo solo tutte le informazioni utili, forza e potenziale.

- infine come ultimo effetto sonoro è stata introdotta la lettura vocale di alcune proprietà del sistema. Tale lettura vocale avviene in due fasi distinte:

  - all'apertura del file viene letto ad alta voce il nome della molecola, in modo che l'utente possa essere subito informato della natura di quanto visualizato a schermo;

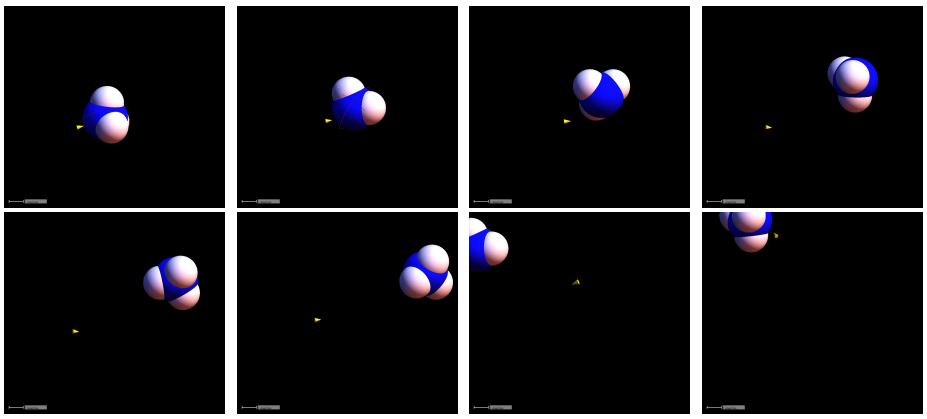

Figura 4.35: Esempio di simulazione dinamica

– durante la selezione di un atomo, aptica o normale che sia, viene pronunciato dal sistema il nome dell'atomo, questo fornisce uno stimolo ulteriore oltre alla scritta sull'interfaccia per capire che tipologia di atomo si abbia selezionato.

Per individuare gli eventi aptici che accadono nel sistema vengono usate delle funzioni di callback del sistema aptico che individuano, ad esempio il tocco di un oggetto con lo strumento aptico.
Per concludere riteniamo che l'introduzione di effetti sonori e lo studio di un interfacciamento audio fornisca un grande valore aggiunto al simulatore, poichè così si riesce a convogliare maggiormente e ad accentuare le informazioni che si vogliono far passare e assimilare all'utente.

4.6 Engine Dinamico

L'introduzione dell'engine dinamico ha avuto lo scopo di dare all'utente la possibilità di interagire con la molecola usando il PHANToM da un punto di vista geometrico, fino ad allora infatti lo strumento aptico era stato usato solo come output del sistema senza la possibilita di interagire direttamente con esso [4]. Non era ancora stata introdotta infatti la selezione della molecola attraverso lo strumento aptico.

In figura 4.35 è rappresentato un esempio di simulazione dinamica in cui a seguito di una spinta da parte del cursore aptico è stato possibile muovere la molecola nello spazio.
Il sistema dinamico concettualmente funziona in maniera piuttosto semplice:

[4] se si esclude la rappresentazione del cursore aptico

- si inizializza il 'mondo' dinamico;
- si mantiene in memoria una descrizione degli oggetti presenti nella scena da un punto di vista fisico, nel caso specifico una molecola con determinate geometria ed massa composta da sfere, gli atomi;
- si associa a tali oggetti una posizione ed una velocità nello spazio;
- si aggiornano tali valori in base alle forze che vengono introdotte nel sistema.

Nel caso specifico dello strumento realizzato si genera l'ente dinamico ponendo delle sfere dotate di massa nello spazio attorno al baricentro della molecola, dopodiché lo si posiziona concordemete a quanto visualizzato dall'engine grafico. A questo punto il controllo passa all'engine aptico che tramite una funzione di callback è in grado di definire quando il cursore tocca effettivamente un oggetto. All'atto del tocco viene attivata una nuova funzione che si attiva ogni qualvolta il cursore aptico si muove, tale sistema permette di associare alla molecola dinamica una forza uguale ed opposta rispetto a quella percepita con il PHANToM, moltiplicata per una costante che individua la potenza fisica del cursore aptico [5], identificato dallo stesso elemento grafico che definisce la carica del cursore. Quest'ultima funzione di callback viene disattivata attraverso l'uso di un'altra funzione che definisce quando avviene il distacco da un oggetto. Da notare che al momento del distacco la forza da associare alla molecola deve essere portata a 0, altrimenti si ha una continua accelerazione dell'oggetto.
A questo punto ad ogni iterazione del thread che si occupa di gestire la grafica viene calcolata la nuova rotazione e la nuova posizione dell'oggetto secondo le forze associate e viene aggiornata la molecola grafica di conseguenza. Nel caso la visualizzazione sia ortogonale la molecola non deve spostarsi sull'asse z, pena la possibilità di 'perderla' nello spazio 3D; per cui la variabile z viene costantemente posta a 0 nonostante le forze associate. Questo non vale, ovviamente, se si passa alla visualizzazione prospettica.
Inoltre è stato introdotto un sistema per evitare che la molecola si perda uscendo dal viewing volume: una volta che la molecola oltrepassa un viewing plane riappare dall'altra parte dello schermo e del viewing volume, ciò viene fatto attraverso la modulazione delle coordinate della molecola.
Una volta che la molecola è messa in moto, si pone il problema di fermarla.

[5] infatti può capitare che una molecola sia molto più pesante di un'altra, in tale modo si può associare al cursore aptico una forza in grado di spostare più facilmente le molecole più pesanti

A tal proposito si è creata una nuova funzionalità del pulsante del PHANToM che una volta premuto, se non a contatto con un'atomo[6], provvede ad azzerare la velocità lineare ed angolare dell'oggetto dinamico, e quindi a fermarlo nello spazio.
L'introduzione dell'engine dinamico ha rappresentato un modo per illudere l'utente della fisicità degli oggetti all'interno del sistema, permettendo quindi maggior immersività ed interazione.

4.7 Aspetti chimici

In questa sezione vengono illustrati gli aspetti progettuali del presente sistema legati alla chimica. In primo luogo viene definito come è stato calcolato il gradiente del campo di forze. Dopodiché vengono mostrate le modalità con cui viene calcolato il potenziale senza l'ausilio delle mappe.

4.7.1 Calcolo del gradiente secondo

Per il calcolo del gradiente secondo si è ulterioremente differenziato la griglia contenente i vettori del gradiente.
Per ognuna delle tre direzioni X,Y e Z si differenziano i vettori del gradiente e si crea un nuovo vettore le cui 3 componenti sono la norma di tali vettori differenziati, questo vettore rappresenta il gradiente secondo.
Traducendo il tutto in matrice 3d si procede così:

1. Si seleziona una cella della matrice di coordinate (i, j, k);

2. si calcola il vettore $\vec{V_x}$ sottraendo i vettori consecutivi della matrice del gradiente primo $\vec{V_i}$ $\vec{V_{i+1}}$ e $\vec{V_{i-i}}$ posti rispettivamente nelle celle (i, j, k), $(i + 1, j, k)$ e $(i - 1, j, k)$ e dividendoli poi per due:

$$\vec{V_x} = \frac{(\vec{V_i} - \vec{V_{i-1}}) + (\vec{V_{i+i}} - \vec{V_i})}{2}$$

3. allo stesso modo si calcolano $\vec{V_y}$ e $\vec{V_z}$ a partire dalle celle $(i, j - 1, k)$ e $(i, j + 1, k)$ per $\vec{V_y}$ e $(i, j, k - 1)$ e $(i, j, k + 1)$ per $\vec{V_z}$:

$$\begin{aligned} \vec{V_y} &= \frac{(\vec{V_j} - \vec{V_{j-1}}) + (\vec{V_{j+i}} - \vec{V_j})}{2} \\ \vec{V_z} &= \frac{(\vec{V_k} - \vec{V_{k-1}}) + (\vec{V_{k+i}} - \vec{V_k})}{2} \end{aligned}$$

[6] cosa che attiverebbe la procedura di selezione aptica

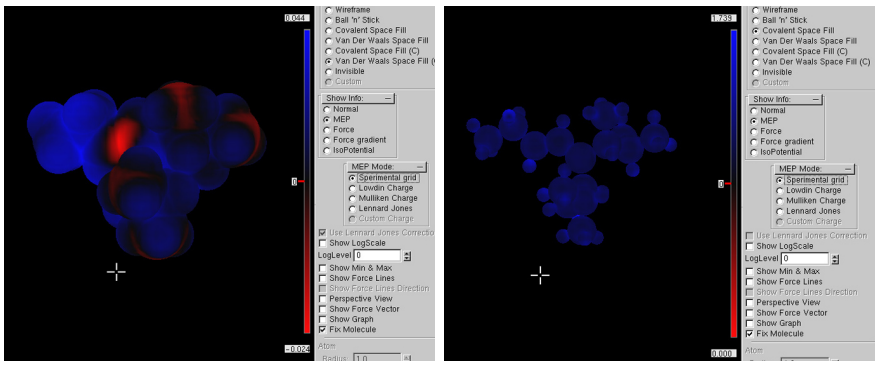

Figura 4.36: potenziale calcolato con la griglia sperimentale

4. A questo punto si crea un nuovo vettore $\vec{V_g}$ le cui componenti sono le norme dei tre vettori: $\vec{V_g} = (\left\|\vec{V_x}\right\|, \left\|\vec{V_y}\right\|, \left\|\vec{V_z}\right\|)$, quest'ultimo vettore rappresenta il gradiente secondo nella cella (i, j, k);

5. si torna ad uno selezionando una nuova cella per la computazione.

Attraverso questo sistema si ottiene una mappa 3D di vettori che formano un campo vettoriale indicante il gradiente secondo del potenziale. Questa nuova informazione dice come varia la forza nello spazio e può essere usata dai ricercatori per ottenere nuove informazioni sulle molecole.

4.7.2 Formule per il calcolo del MEP e del Force Field

Oltre alla possibilità di definire il MEP e la forza attraverso le griglie sperimentali precalcolate sono state introdotte nuove modalità di calcolo delle due caratteristiche. Queste nuove modalità si basano su diverse formule che approssimano il contenuto del potenziale o della forza. L'obbiettivo è quello di simulare il più correttamente possibile il comportamento delle griglie sperimentaili di potenziale, figura 4.36.

In primo luogo sono state introdotte due modalità di calcolo del potenziale basate su cariche fittizie assegnate ai vari atomi che compongono la molecola. I due metodi derivano da Lowdin e Mulliken e rappresentano due metodi alternativi di assegnare tali cariche, rispettivamente figura 4.37. Dall'immagine si possono notare le lievi differenze tra i due metodi.

Questi metodi calcolano il potenziale P in questo modo, dove k è la costante di coulomb pari a $8.99 \cdot 10^9 \frac{Nm^2}{C^2}$, mentre q_i e d_i sono rispettivamente la carica e la distanza dal punto in questione dell'i-esimo atomo , n infine è

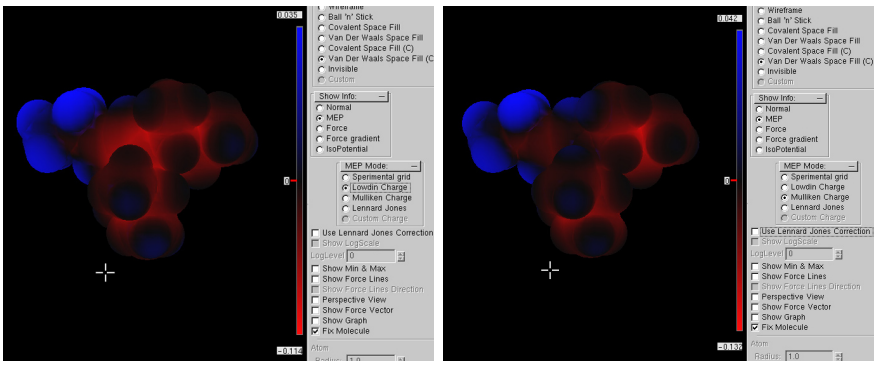

Figura 4.37: MEP: cariche di Lowdin e di Mulliken

il numero degli atomi:

$$P = k \sum_{i=0}^{n} \frac{q_i}{d_i} \tag{4.6}$$

Come si evince dall'immagine 4.37 rispetto alla figura 4.36, le due approssimazioni sono piuttosto grossolane, ma presentano una visualizzazione coerente con i risultati sperimentali alla distanza di Van Der Waals. Se si va però a visualizzare il potenziale in un area più prossima ai nuclei, ad esempio raggio covalente, ci si accorge dell'enorme errore derivante dall'uso del metodo coulombiano per calcolare il potenziale in tutto lo spazio, figura 4.38[7] rispetto alla seconda immagine in 4.36. Mentre col l'ausilio delle griglie di potenziale il MEP attorno agli atomi centrali è positivo, usando le cariche è negativo.

Questo effetto è dovuto al fatto che i minimi di potenziale non sono collocati nei punti dove dovrebbero, bensì nel nucleo degli atomi a carica negativa, e questo rappresenta un controsenso, anche perché nei nuclei di tali atomi il potenziale raggiunge valori pari a $-\infty$, mentre dovrebbe essere ∞. Questa riflessione indica che il metodo delle cariche funziona solo ad una certa distanza dai nuclei degli atomi, infatti per i raggi di Van der Waals rappresenta una buona approsimazione, sia per quanto riguarda il potenziale che la forza.

Le cariche di Mulliken e di Lowdin vengono ottenute effettuando il parsing dei file out in output a GAMESS. In tali file sono infatti contenute le cariche di Lowdin e Mulliken per ognuno degli atomi presenti nella molecola, derivate da calcoli sperimentali.

Allo stesso modo con cui viene calcolato il potenziale, è possibile calcolare la forza $\vec{F}$ attraverso le cariche, ipotizzando una carica di prova unitaria,

[7] per Mulliken abbiamo lo stesso risultato

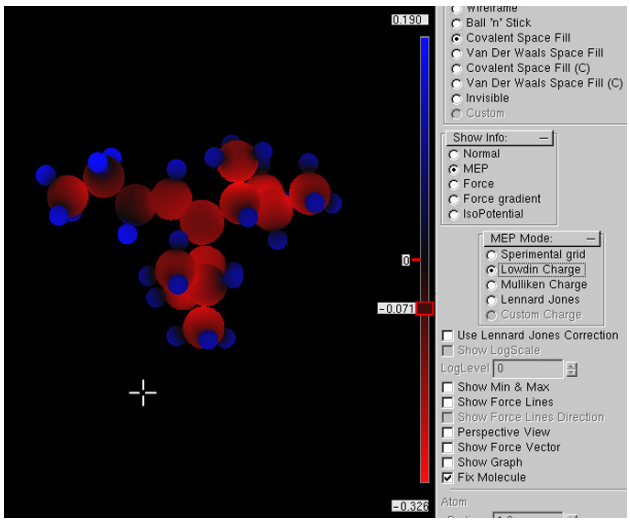

Figura 4.38: MEP: cariche di Lowdin, raggio covalente

espressione:

$$\vec{F} = k \sum_{i=0}^{n} \frac{q_i}{d_i^2} \vec{u_i} \tag{4.7}$$

dove $\vec{u_i}$ è il versore indicante la direzione tra la carica di prova e la carica i-esima.
In figura 4.39 è rappresentato il campo di forze con la griglia sperimentale a sinistra e con Lowdin a destra. Si può notare che, alla distanza di Van der Waals, rappresentano più o meno lo stesso comportamento. Se però ci si avvicina al nucleo dell'atomo, con i raggi covalenti ad esempio, si nota lo stesso errore che caratterizza il MEP, figura 4.40. In questo caso il problema è ancora più grave perché, portando il discorso su un livello aptico, il cursore a carica positiva verrebbe attratto dal nucleo di un atomo, invece che venirne respinto, inoltre con una forza che tende a ∞.

Per quanto mostrato finora parrebbe impossibile tradurre le formule precedenti in una corretta percezione aptica del force field attorno alla molecola, a tal proposito è stata inserita una nuova componente di potenziale e quindi di forza chiamata potenziale di Lennard Jones.
Il potenziale di Lennard Jones agisce sulle distanze inferiori alle cariche coulombiane, e permette di sintetizzare l'effetto repulsivo dei nuclei degli atomi. In figura 4.41 è mostrato l'effetto della correzione di Lennard jones su un sistema a cariche di Mulliken e di Lowdin. Si può notare che grazie al potenziale di Lennard Jones viene annullato l'effetto attrattivo dei nuclei a carica negativa.
Il contributo di Lennard Jones P viene calcolato in questo modo, dati n

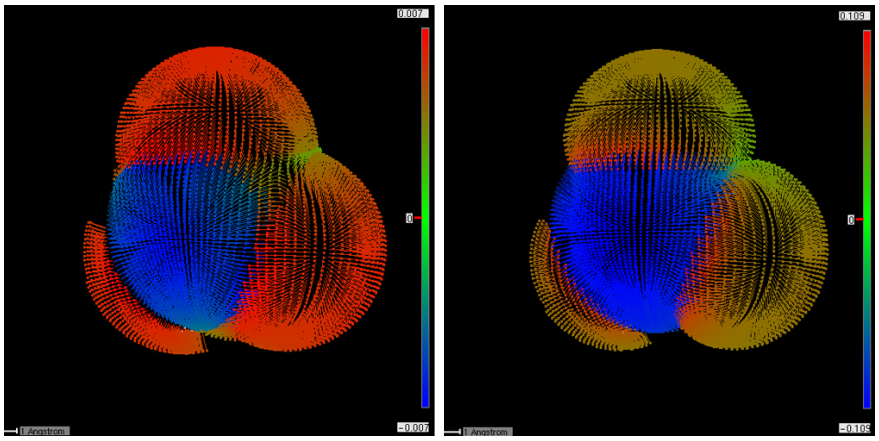

Figura 4.39: Forza: griglia sperimentale e Lowdin, Van der Waals

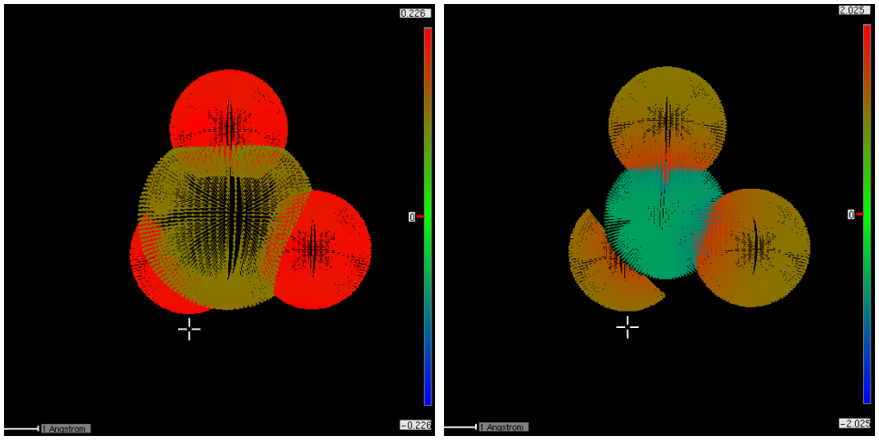

Figura 4.40: Forza: griglia sperimentale e Lowdin, raggi Covalenti

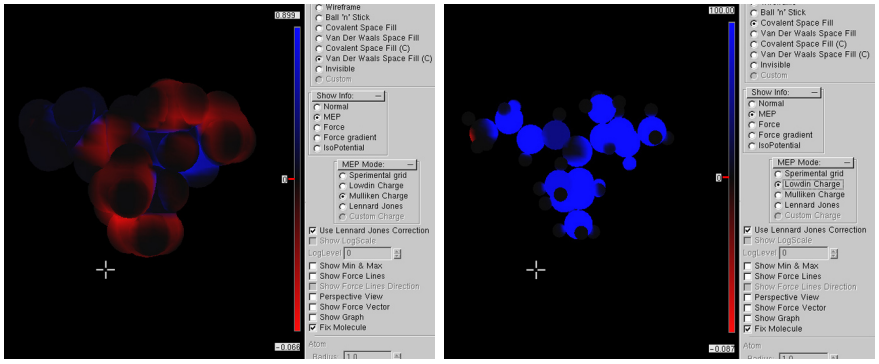

Figura 4.41: Correzione lennard jones

atomi che compongono la molecola, d_i le distanze tra il cursore e il nucleo dell'atomo, d_{m_i} la distanza dal minimo di potenziale più vicino all'atomo i-esimo, e v_i il valore di questa buca di potenziale:

$$P = \sum_{i=0}^{n} v_i[(\frac{d_{m_i}}{d_i})^{12} - 2(\frac{d_{m_i}}{d_i})^{6}] \tag{4.8}$$

Al che sorge il problema di come calcolare le distanze dei minimi più vicini ai vari atomi, dato indispensabile per il calcolo di Lennard Jones. Questo problema è stato affrontato sfruttando le informazioni di cui si era a conoscenza tramite la griglia del MEP. Similmente a quanto fatto per il calcolo e il disegno delle linee di forza, si procede in questo modo:

1. si pone come punto di partenza della ricerca il nucleo dell'atomo di cui cercare il minimo e si fanno partire 6 raggi in direzione delle 3 direzioni spaziali (x, y, z) per entrambi i versi;

2. questi vettori, di lunghezza pari a un decimo di Angstrom, identificano dei punti da cui seguire il gradiente del potenziale calcolato tramite la griglia per una distanza pari a un decimo di Angstrom;

3. si continua così finché la distanza in due passi successivi è inferiore al decimo di Angstrom, momento in cui si dimezza il passo;

4. si continua questo procedimento finché non si arriva a un passo pari a 3 millesimi di Angstrom, o fino a che non si fanno 1000 iterazioni;

5. a questo punto se si è arrivati in un punto a potenziale negativo si considera questo come il possibile punto di minimo più vicino;

6. si calcola la distanza dei minimi raggiunti con i 6 differenti raggi e si sceglie quello a distanza minore.

Con questa metodologia, particolarmente empirica[8], è possibile individuare tuti i dati necessari al calcolo del potenziale di Lennard Jones. In questa sede è stata sfruttata l'idea alla base del potenziale di Lennard Jones, rielaborandola all'esigenza, d'altronde la metodologia introdotta da Lennard Jones non si basa su considerazioni fisiche accurate, ma rappresenta un metodo approssimato per simulare comportamenti molecolari. A questo proposito è stato introdotto un comportamento sommatorio derivante dalle componenti di tutti gli atomi coinvolti, non presente nella formulazione originale, inoltre

[8] da notare che anche il calcolo stesso teorico del potenziale di Lennard Jones è una formula empirica

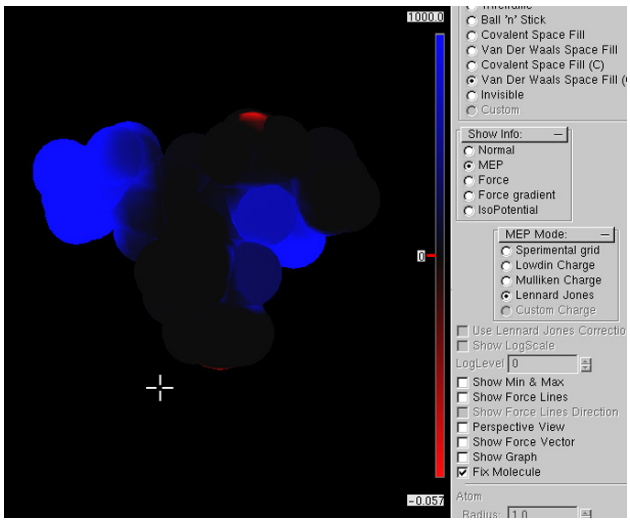

Figura 4.42: Visualizzazione potenziale di Lennard Jones

nell'equazione 4.8 se la distanza tra il minimo locale di potenziale e il nucleo dell'atomo i-esimo è maggiore di 3 volte il raggio di Van Der Waals, la quantità viene omessa dalla sommatoria. Questo per evitare che un atomo sia legato ad un minimo troppo distante, fatto che influenzerebbe troppo il potenziale.
La componente di Lennard Jones può essere introdotta anche per il calcolo della forza:

$$\vec{F} = \sum_{i=0}^{n} 12 v_i \left(\frac{d_{m_i}^{12}}{d_i^{13}} - \frac{d_{m_i}^{6}}{d_i^{7}}\right)\vec{u_i} \tag{4.9}$$

dove $\vec{u_i}$ è la direzione congiungente il cursore aptico al nucleo dell'atomo.
In totale potenziale e forza si calcolano nel modo seguente:

$$P = k \sum_{i=0}^{n} \frac{q_i}{d_i} + \frac{1}{10n} \sum_{i=0}^{n} v_i \left[\left(\frac{d_{m_i}}{d_i}\right)^{12} - 2\left(\frac{d_{m_i}}{d_i}\right)^{6}\right] \tag{4.10}$$

$$\vec{F} = k \sum_{i=0}^{n} \frac{q_i}{d_i^2}\vec{u_i} + \frac{1}{10n} \sum_{i=0}^{n} 12 v_i \left(\frac{d_{m_i}^{12}}{d_i^{13}} - \frac{d_{m_i}^{6}}{d_i^{7}}\right)\vec{u_i} \tag{4.11}$$

$$\tag{4.12}$$

In cui il fattore $1/10n$ è stato introdotto per pareggiare le unità di misura.
Nel sistema è stata introdotta anche la possibilità di utilizzare il solo contributo di Lennard Jones, figura 4.42. Sebbene il potenziale si discosti molto da quello effettivo proiettato attraverso la griglia di dati sperimentale[9], la sen-

[9] c'è da contare che in questo caso non viene fatta distinzione rispetto alla posizione del minimi più vicini

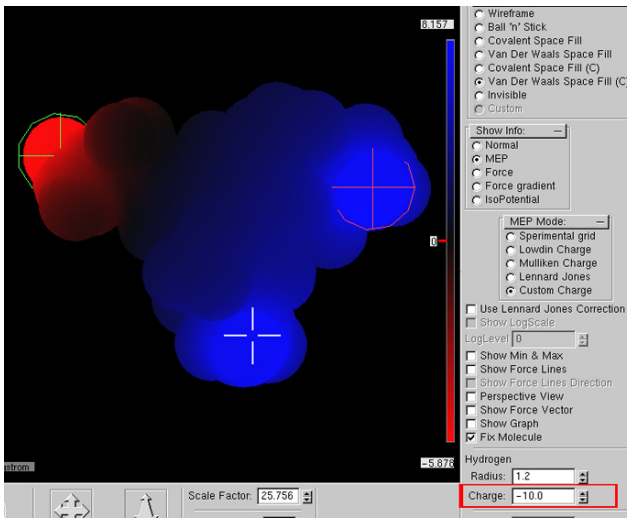

Figura 4.43: Potenziale con cariche definite dall'utente

sazione aptica percepita risulta piuttosto corretta, al punto da individuare correttamente i minimi. La soluzione approntata è comunque meno precisa rispetto al caso Coulombiano con l'aggiunta della correzione di Lennard Jones.
Inoltre è stata data la possibilità all'utente di personalizzare il potenziale attraverso l'uso di cariche definite dallo stesso. In figura 4.43 è mostrato un esempio in cui all'atomo selezionato è stato assegnato un valore di carica pari a $-10.0C$ mentre a quello in fase di preselezione un valore di $10.0C$, questo modifica completamente il grafico di potenziale e le forze percepite.

Riassumendo, rispetto al prototipo iniziale, sono state introdotte nuove possibilità di calcolo del potenziale e della forza attraverso varie approssimazioni. Lo strumento realizzato permette di misurare la bontà di tali approsimazioni e di valutare lo scostamento rispetto al caso sperimentale.

4.8 Conclusioni del capitolo

In questo capitolo è stata illustrata la logica alla base di tutto quanto sviluppato, nonché la metodologia con cui sono state introdotte le nuove funzionalità. Durante la trattazione sono state spiegate dettagliatamente le differenze con il prototipo inziale, cosa è stato corretto e cosa è stato sviluppato ex novo.
La maggior parte delle scelte progettuali e delle idee inerenti le nuove fun-

zionalità introdotte deriva da accorgimenti originali ideati dal sottoscritto tesista, quali:

- l'ideazione delle scale logaritmiche;
- la visualizzazione del campo di forze come vettori;
- la possibilità di interagire geometricamente con la molecola;
- la rappresentazione dei volumi isopotenziali;
- la correzione al potenziale utilizzando Lennard Jones;
- l'introduzione dello stimolo sonoro per indicare il MEP percepito dal cursore aptico;
- l'interpolazione 3D del potenziale, gradiente e gradiente II;
- la selezione aptica di un atomo;
- la simulazione dinamica del comportamento geometrico della molecola;
- la selezione degli atomi e la conseguente possibilità di modificare raggio [10] e carica;
- la rappresentazione di più massimi e minimi assoluti;
- l'introduzione di percezioni aptiche legate a proprietà atomiche;
- ...

Quanto presentato in questo capitolo rappresenta quindi un insieme di idee sviluppate attorno al nucleo di funzionalità presente dal principio, ovviamente riveduto e corretto come illustrato in questa sede. Tali idee rappresentano una spinta all'innovazione capace di portare nuovi utilizzi del software, anche nel campo della ricerca.
Nel prossimo capitolo verrà illustrata la struttura delle classi e del codice sviluppato, senza scendere eccessivamente nel dettaglio. Si vedrà come i vari engine interagiscono fisicamente tra di loro e quali metodi sono stati sviluppati all'interno delle varie classi.

[10] al fine di esplorare il potenziale attorno alla molecola

Capitolo 5

Aspetti Implementativi

In questo capitolo vengono analizzati gli aspetti implementativi di quanto trattato nel capitolo 4. Si riprendono ad uno ad uno tutte le parti del programma e si scende nel dettaglio delle librerie e degli strumenti utilizzati. Andando a esaminare il codice C++ è necessario innanzitutto familiarizzare con le classi introdotte:

- **AudioEngine**: classe che permette di controllare l'engine audio;
- **DynamicEngine**: classe che definisce la struttura del simulatore dinamico;
- **GraphicEngine**: classe per la gestione della grafica 3d;
- **HapticApplication**: classe contenitore di tutti gli engine;
- **HapticEngine**: classe che permette di interagire con il sistema aptico;
- **HapticMol**: contiene il main e registra le funzioni di callback di GLUT;
- **MEP**: gestisce la mappa di potenziale e le griglie del gradiente primo e secondo;
- **Molecule**: permette di gestire la molecola;
- **UserInterfaceEngine**: classe per la gestione dell'interfaccia utente;
- **Utilities**: contenitore di funzioni utli per il sistema;
- **Vector**: classe per la gestione dei vettori.

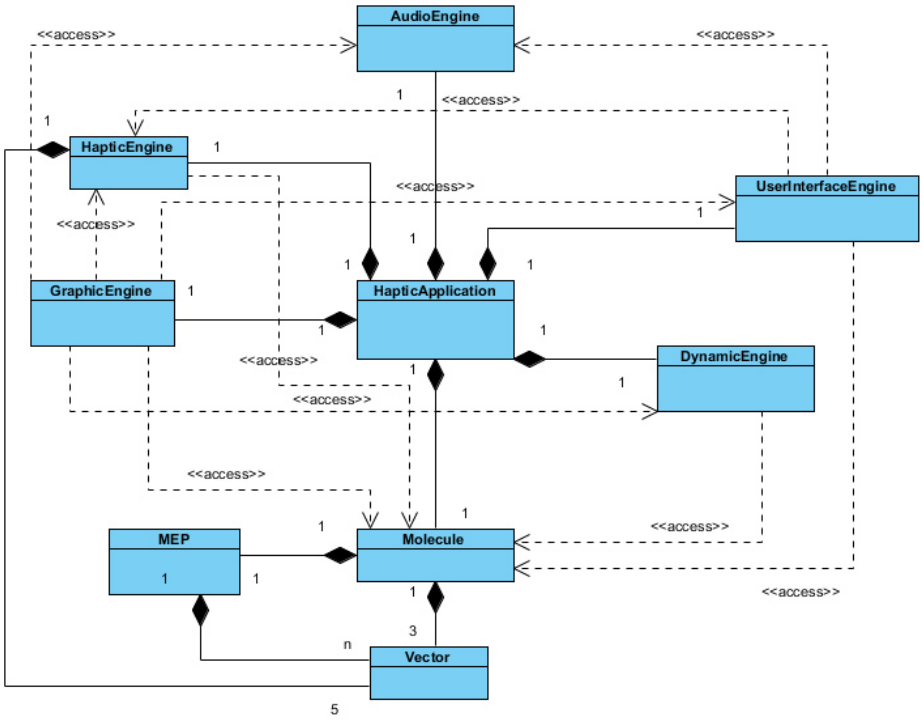

Figura 5.1: Diagramma uml delle classi

In figura 5.1 è rappresentato il diagramma uml delle classi.

Nel seguito verrano descritti i principali metodi e attributi presenti all'interno di suddette classi.

5.1 Classe AudioEngine

La classe AudioEngine definisce tutte le funzioni relative all'interfacciamento sonoro. Dopo aver inizializzato i suoni principali all'interno del costruttore è possibile richiamarli attraverso il metodo **playsound**, inoltre è possibile assegnare varie proprietà a tali suoni attraverso i vari setter. Per finire vi è un metodo **Read** che permette di leggere la stringa passata in input.

Attributi privati

Gli attributi della classe AudioEngine rappresentano quasi tutti dei suoni presenti all'interno del sistema attraverso due elementi **Buffer** e **Source**. I **Buffer** contengono la forma d'onda che rappresenta il suono e la mantengono in memoria, mentre le **Source** rappresentano un oggetto sonoro nello spazio, con una posizione e delle proprietà sonore. Ad ogni *Source* è assegnato almeno un $Buffer$.

- **ALuint clickBuffer**: buffer contenente il suono assegnato alla selezione degli atomi con il mouse o all'attivazione delle checkbox nell'interfaccia, tale suono è caricato da un file wave in memoria;
- **ALuint clickSource**: sorgente a cui è assegnato il buffer precedente, assume le proprietà derivanti dall'acusmetria [1] nel caso venga usato per il click nella selezione, mentre possiede proprietà statiche[2] per la pressione delle checkbox nell'interfaccia;
- **ALuint unclickBuffer**: contiene il suono wave associato alla deselezione attraverso il mouse, inoltre identifica la disattivazione di una checkbox;
- **ALuint unclickSource**: come nel caso del click, in caso di deselezione fa uso delle proprietà acusmetriche, mentre nell'interfaccia possiede proprietà statiche;

[1] cioè mappatura dell'asse x con l'effetto stereo del suono, asse y frequenza e asse z potenza dello stesso

[2] con proprietà statiche si intende che ogni volta che la sorgente viene richiamata presenta sempre le stesse caratteristiche, nel caso degli elementi nell'interfaccia la posizione della sorgente a $(0, 0, 1)$ cioè centrale e leggermente avanzata verso l'ascoltatore.

- **ALuint selectBuffer**: suono wave emesso al passaggio del mouse sopra un atomo o nella selezione da un radiogroup;
- **ALuint selectSource**: gestisce il suono precedente, proprietà dell'acusmetria per la pre-selezione, proprietà statiche per la selezione di un radio button;
- **ALuint wheelBuffer**: rappresenta un suono wave usato per indicare l'utilizzo degli slider all'interno dell'interfaccia;
- **ALuint wheelSource**: tale sorgente presenta sempre proprietà statiche;
- **ALuint popBuffer**: suono wave emesso al passaggio del mouse sopra la scala colorimetrica;
- **ALuint popSource**: questa sorgente presenta sempre proprietà statiche, nello specifico una posizione pari a $(10, 0, 0)$ in modo da rappresentare il fatto che la scala sia rappresentata a destra della scena 3d;
- **ALuint buttonBuffer**: suono wave che identifca la pressione di un pulsante nell'interfaccia, quali quello per caricare la molecola e il pulsante quit;
- **ALuint buttonSource**: anche questa sorgente viene rappresentata staticamente;
- **ALuint ForcePosBuffer**: suono sintetizzato dal sistema, rappresenta la base per i punti a potenziale positivo, viene usato per renderizzare il suono del potenziale nel punto in cui si trova il cursore aptico;
- **ALuint ForcePosSource**: questa sorgente assume parzialmente proprietà acusmetriche, sono presenti ancora infatti la mappatura della posizione sugli assi x e z, la frequenza viene però usata per identificare il potenziale positivo, nello specifico varia sulla scala logaritmica in un intervallo $[0, 4]$ dove 4 viene associato al massimo valore di scala;
- **ALuint ForceNegBuffer**: suono sintetizzato diverso dal precedente per identificare le aree a potenziale negativo;
- **ALuint ForceNegSource**: possiede le stesse caratteristiche del caso positivo solo che l'intervallo di valori della frequenza sulla scala varia da $[0, 1]$ con 1 il minimo di potenziale;
- **ALuint TouchBuffer**: suono sintetizzato dal sistema, identifica il tocco di un atomo da parte dell'interfaccia aptica nel caso geometrico;

- **ALuint TouchSource**: anche in questo caso la sorgente possiede parzialmente proprietà acusmetriche, assi x e z, mentre la frequenza è associata al valore di elettronegatività dell'atomo associato in un intervallo $[0, 2]$ dove 2 identifica un valore di elettronegatività pari a 4, il massimo della scala;
- **ALuint ContactBuffer**: suono sintetizzato, viene emesso quando il cursore aptico è a contatto con la superficie geometrica della molecola;
- **ALuint ContactSource**: la sorgente possiede sempre proprietà acusmetriche, anche per l'asse y;
- **ISpVoice * pVoice**: puntatore che serve per definire una voce del sintetizzatore vocale tramite la SAPI.

Metodi Pubblici

A parte ovvi usi dei costruttori e distruttori, presenta metodi setter per impostare le caratteristiche dei suoni le cui proprietà variano nell'utilizzo del sistema.

- **AudioEngine()**: costruttore, inizializza la libreria sonora, alloca i vari buffer e li associa alle sorgenti, inoltre definisce anche la posizione dell'ascoltatore a $(0, 0, 21)$[3], infine inizializza il sintetizzatore vocale assegnando la voce standard del sistema operativo all'applicazione;
- **∼AudioEngine(void)**: distruttore, rilascia la voce utilizzata per il sintetizzatore vocale e chiude la libreria sonora;
- **void playsound(int soundtype)**: esegue il suono associato ad una sorgente modificato attraverso le relative proprietà;

  INPUT

  - **int soundtype**: costante che identifica un suono nel sistema, ad es. click corrisponde a 1.
- **void setForcePos(Vector position,ALfloat value)**: imposta le caratteristiche della sorgente legata al potenziale positivo;

  INPUT

  - **Vector position**: vettore che indentifica la posizione del cursore aptico al momento;

[3]lo spazio sonoro definito dall'acusmetria rappresenta lo spazio grafico diviso per 10, per non eccedere nell'attenuazione dei suoni lontani

  - **ALfloat value**: identifica il valore del potenziale positivo, già trasformato e riscalato attraverso il logaritmo;

- **void setForceNeg(Vector position,ALfloat value)**: come il caso precedente solamente legato al potenziale negativo

  INPUT

  - **Vector position**: vettore contenente la posizione del cursore aptico;
  - **ALfloat value**: valore del potenziale negativo, opportunamente riscalato;

- **void setTouch(Vector position,ALfloat value)**: imposta le proprietà della sorgente associata al tocco aptico di un atomo;

  INPUT

  - **Vector position**: indentifica la posizione del cursore aptico al momento del tocco della molecola;
  - **ALfloat value**: identifica il valore di elettronegatività dell'atomo toccato;

- **void setClick(Vector position)**: imposta le caratteristiche del suono click di selezione;

  INPUT

  - **Vector position**: vettore che indentifica la posizione del nucleo dell'atomo selezionato, oppure la costante $(0, 0, 1)$ per l'utilizzo del click nell'interfaccia;

- **void setUnclick(Vector position)**: identifica le proprietà del suono deselezione.

  INPUT

  - **Vector position**:indentifica la posizione del nucleo dell'atomo deselezionato, oppure la costante $(0, 0, 1)$;

- **void setSelect(Vector position)**: definisce le caratteristiche del suono di preselezione;

  INPUT

  - **Vector position**: posizione del nucleo dell'atomo in preselezione, oppure la costante $(0, 0, 1)$;

- **void setContact(Vector position)**: definisce la posizione della sorgente del suono che identifica il contatto, nell'istante prima di essere eseguito;

  INPUT

  – **Vector position**: posizione cursore aptico nel momento del contatto con la molecola;

- **void Read(LPCWSTR lettura)**: legge una stringa tramite il sintetizzatore vocale, utilizzando la voce standard del sistema;

  INPUT

  – **LPCWSTR lettura**: indica la stringa da far leggere al sintetizzatore.

5.2 Classe DynamicEngine

La classe DynamicEngine definisce i metodi e gli elementi necessari al corretto funzionamento del sistema dinamico. Dopo aver inizializzato l'engine, utilizza vari metodi per modificare lo stato del mondo dinamico mantenuto in memoria.

Attributi privati

- **dWorldID wID**: identificativo del mondo dinamico definito e creato nel contesto dell'engine;
- **dBodyID BodyID**: identificativo dell'oggetto dinamico che fa riferimento alla molecola;
- **Molecule *mol**: puntatore alla molecola attualmente rappresentata nella scena 3d.

Metodi pubblici

- **DynamicEngine()**: costruttore, inizializza l'engine e la variabile **wID** contenente il mondo dinamico;
- **~DynamicEngine()**: distruttore, distrugge il mondo creato identificato da **wID** e chiude l'engine;

- **void Init(Molecule *mol)**: inizializza l'oggetto fisico **BodyID** e, dopo aver collegato l'engine alla molecola visualizzata a schermo, richiama la funzione **Refresh**;

  INPUT

  - **Molecule *mol**: puntatore all'oggetto molecola attualmente visualizzato nella scena 3d.

- **void Refresh()**: funzione che aggiorna l'oggetto dinamico **BodyID** sulla base delle caratteristiche della molecola associata **mol**, quali posizione, rotazione, massa e dimensione dei singoli atomi, in modo da permettere all'engine di identificare la massa inerziale dell'oggetto;

- **void Reset()**: azzera forze e velocità dell'oggetto identificato da **BodyID**;

- **void Apply()**: applica a **mol** la rototraslazione dell'oggetto dinamico definito da **BodyID**;

- **void SetForceAtPos(Vector pos,Vector force)**: imposta una forza sull'oggetto dinamico che porterà al cambiamento di stato dell'oggetto;

  INPUT

  - **Vector pos**: vettore che indica il punto della molecola in cui la forza viene applicata;
  - **Vector force**: indica l'ammontare e la direzione della forza applicata.

- **void ResetForce()**: annulla la forza e la torsione assegnata all'oggetto dinamico;

- **dWorldID getWorldId()**: restituisce l'identificativo del mondo, utile per richiamare la funzione **dWorldFastStep** dal GraphicEngine; tale funzione permette di far progredire la dinamica del sistema.

  OUTPUT

  - **dWorldID**: identificativo del mondo dinamico corrente.

5.3 Classe GraphicEngine

La classe GraphicEngine definisce tutti i metodi utili a disegnare gli elementi nella scena 3D, inoltre definisce le funzioni di callback per mouse e tastiera,

infine propone gli attributi per contenere le opzioni relative alla grafica del sistema.

Attributi e metodi privati

Attributi

- **Molecule* HapticMolecule**: puntatore alla molecola rappresentata nella scena 3d;
- **UserInterfaceEngine* UIEngine**: puntatore alla UserInterfaceEngine corrente, utile per aggiornare l'interfaccia utente con i cambiamenti di proprietà grafiche del sistema uali rototraslazioni della molecola o limiti delle superfici isopotenzieli;
- **HapticEngine* HEngine**: puntatore all'engine aptico, utile per disegnare il cursore aptico e per la gestione della HLAPI;
- **AudioEngine* AEngine**: puntatore all'engine audio, utilizzato nella gestione degli eventi sonori interni alla scena 3d;
- **DynamicEngine* DEngine**: puntatore all'engine dinamico, per ogni ciclo grafico viene infatti richiamato l'engine dinamico;
- **string HapticInfo**: stringa che rappresenta le informazioni aptiche descritte nel rettangolo in alto, quali posizione, forza erogata sotto forma di modulo e direzione;
- **string MoleculeName**: nome della molecola rappresentata;
- **string MoleculeInfo**: informazioni utili della molecola, quali posizione, rotazione e scaling;
- **string PerformanceInfo**: contiene le informazioni sui frame al secondo, sia aptici che grafici e indica il numero e la qualità degli atomi rappresentati;
- **bool normalLight**: indica se è impostata l'illuminazione 'normale', quella cioè definita dalle tre luci colorate posizionate agli angoli dello schermo;
- **Color* lockedColor**: puntatore al colore agganciato tramite il mouse sulla scala logaritmica[4];

[4]le maniglie subito a destra della scala logaritmica che indicano la posizione dei colori primari utilizzati

- **double downLimitColor**: indica il vincolo inferiore in cui può essere spostato il colore agganciato descritto dal puntatore **lockedColor**;
- **double upLimitColor**: indica il vincolo superiore, entrambi gli attributi vengono definiti in base alla posizione del colore primario al di sopra o al di sotto di quello selezionato;
- **bool changingColor**: indica se si sta o meno cambiando il colore sulla scala logaritmica;
- **bool changingIsoLimitsmin**: indica se si sta cambiando o meno il limite inferiore sulla scala logaritmica in caso di volumi isopotenziali;
- **bool changingIsoLimitsmax**: come sopra indicando il limite superiore;
- **bool DrawLS**: indica se devono essere rappresentato il quadratino sulla scala colorimetrica indicante il valore effettivo di potenziale o forza;
- **double yLS**: definisce la posizione in cui il quadratino di cui sopra deve essere disegnato;
- **bool DrawCircle**: definisce se deve essere disegnato o meno l'insieme di cerchi che indica la selezione di un atomo nella molecola;
- **bool DrawCircleHaptic**: stessa cosa di sopra ma si riferisce alla selezione aptica;
- **bool selecting**: indica se si sta selezionando un atomo, cioè se si sta cliccando su di un atomo preselezionato;
- **unsigned int passingOverAtom**: indice dell'atomo sopra cui il cursore e posto;
- **unsigned int DetailLevel**: livello di dettaglio della rappresentazione degli atomi come sfere, indica inoltre il numero di punti che vanno rappresentati sulle superfici isopotenziali e il numero di linee di forza da rappresentare;
- **unsigned int FramesCount,DWORD TimeElapsed,DWORD ActualTimeElapsed**: variabili utili al calcolo dei frame al secondo grafici, rappresentanti le prestazioni del motore grafico.

Metodi

- **void GLInitialization(void)**:inizializza il sistema grafico definendo diverse proprietà della visualizzazione, nonché la posizione e la tipologia delle luci inserite.
- **void Draw(void)**:funzione principale di disegno dell'applicazione, è richiamata ogni singolo frame per visualizzare costantemente l'evolversi del sistema. All'interno della funzione vengono chiamate tutte le funzioni di disegno degli oggetti che compongono la scena 3D e la misurazione del frame rate del motore grafico. Inolter gestisce anche i frame dinamici e aptici per quanto riguarda la renderizzazione aptica della geometria molecolare.
- **void DrawMoleculeAtoms(void)**:funzione di disegno degli atomi della molecola, nonché di tutte le proprietà grafiche che si riferiscono direttamente a tali atomi, quali linee di forza superfici isopotenziali e minimi e massimi. Tale metodo deve tenere sempre in considerazioni le proprietà della molecola nell'istante in cui viene rappresentata, quali raggio dei suoi atomi, tipo di scala e tipo di informazione.
- **void DrawMoleculeBonds(void)**:metodo che disegna i legami della molecola attuale.
- **void feelMoleculeAtoms()**: funzione che renderizza la molecola da un punto di vista aptico, associando ad ogni singolo atomo le proprietà aptiche opportune, duplica i comandi geometrici di **DrawMoleculeAtoms**.
- **void DrawHapticDeviceCursor(void)**:disegna il cursore aptiche dipendentemente dal contesto, una freccia conica nell'esplorazione geometrica della molecola ed un mirino nella percezione del MEP. Costantemente comunica con l'engine aptico per ottenere tutte le informazioni relativi alla posizione e all'orientamento del proxy o dell'end-effector del device aptico.
- **void DrawGraph(void)**: disegna il grafico di potenziale durante la percezione del MEP con il cursore aptico. Anche questo metodo necessita di comunicare con il device aptico per conoscerne la posizione.
- **void DrawLogLittleSquare(double y)**: Disegna il quadratino[5] sulla scala logaritmica.

[5]quello contenente l'informazione puntuale sul valore del potenziale associato ad una particolare tonalità di colore nella scala

INPUT

- **double y**: identifica l'altezza della scala in cui disegnare il quadratino.

- **void DrawLinearLittleSquare(double y)**: come sopra, ma riferito ad una scala lineare tra minimo e massimo relativo alla superficie rappresentata.

  INPUT

  - **double y**: esattamente come sopra identifica l'altezza della scala in cui disegnare il quadratino.

- **void DrawIsoPotential(unsigned int quality)**: disegna i volumi isopotenziali sotto forma di puntini. Scorre tutto l'array contenente gli indici della mappa di potenziale che soddisfano la condizione di essere interni all'intervallo selezionato.

  INPUT

  - **unsigned int quality**: indica la qualità della visualizzazione dei volumi isopotenziali, cioè la frequenza con cui vengono selezionati i punti da disegnare presenti nell'array[6].

- **void MEPColoredSphere(Vector position, double radius, unsigned int slices, unsigned int stacks)**: disegna delle sfere colorate secondo la mappa di potenziale o il calcolo del potenziale tramite formule. In questo caso con una colorazione locale dal rosso al blu.

  INPUT

  - **Vector position**: posizione del nucleo dell'atomo secondo la geometria originale caricata dal file pdb;
  - **double radius**: raggio di tale sfera;
  - **unsigned int slices**: numero di fette in cui è suddivisa la sfera, spesso coincide con il detail level;
  - **unsigned int stacks**: numero di spicchi in cui è suddivisa la sfera, spesso coincide con il detail level.

- **void GradientColoredSphere(Vector position, double radius, unsigned int slices, unsigned int stacks)**: disegna una sfera formata da tanti mini-vettori colorati e direzionati secondo il gradiente calcolato nei modi possibili. Colorazione locale.

[6] se devono essere presi tutti, uno ogni due, uno ogni tre etc

INPUT

- **Vector position**: posizione del nucleo dell'atomo;
- **double radius**: raggio della sfera;
- **unsigned int slices**: numero di vettori rappresentati nella direzione longitudinale;
- **unsigned int stacks**: numero di vettori nella direzione latitudinale della sfera.

- **void GradientIIColoredSphere(Vector position, double radius, unsigned int slices, unsigned int stacks)**: disegna una sfera con colorazione locale del gradiente secondo calcolato con l'unica modalità possibile cioè la griglia precalcolata di vettori.

  INPUT

  - **Vector position**: come sopra, posizione del nucleo dell'atomo;
  - **double radius**: raggio della sfera;
  - **unsigned int slices**: numero di vettori rappresentati nella direzione longitudinale;
  - **unsigned int stacks**: numero di vettori nella direzione latitudinale della sfera.

- **void MEPLogColoredSphere(Vector position, double radius, unsigned int slices, unsigned int stacks)**: versione assoluta e logaritmica del metodo precedente **MEPColoredSphere**. Cambia il calcolo del potenziale che viene filtrato attraverso la scala logaritmica. I paramteri di input sono gli stessi.

- **void GradientLogColoredSphere(Vector position, double radius, unsigned int slices, unsigned int stacks)**: versione assoluta e logaritmica del metodo **GradientColoredSphere**. I paramteri di input sono gli stessi.

- **void GradientIILogColoredSphere(Vector position, double radius, unsigned int slices, unsigned int stacks)**: versione assoluta e logaritmica del metodo **GradientIIColoredSphere**. I paramteri di input sono gli stessi.

- **void DrawForceLines(Vector position, double radius, unsigned int slices, unsigned int stacks)**: disegna le linee di forza partendo dalla superficie di una sfera.

  INPUT

  - **Vector position**: posizione del nucleo dell'atomo;
  - **double radius**: raggio della sfera da cui partiranno le linee di forza;
  - **unsigned int slices**: indice del numero di punti di partenza delle linee di forza longitudinalmente, di solito uguale all'indicatore di dettaglio;
  - **unsigned int stacks**: indice del numero di punti di partenza delle linee di forza latitudinalmente, di solito uguale all'indicatore di dettaglio.

- **void DrawLogColorScale()**: disegna la scala colorimetrica logaritmica.

- **void DrawLinearColorScale()**: disegna la scala colorimetrica lineare.

- **void Sphere(Vector position, double radius, unsigned int slices, unsigned int stacks)**: disegna una semplice sfera, utilizzato per disegnare gli atomi colorati uniformemente nella rappresentazione geometrica.

  INPUT

  - **Vector position**: posizione del centro della sfera;
  - **double radius**: raggio della sfera;
  - **unsigned int slices**: numero di fette in cui è suddivisa la sfera, spesso coincide con il detail level;
  - **unsigned int stacks**: numero di spicchi in cui è suddivisa la sfera, spesso coincide con il detail level.

- **void Cone(float rotation[16], double radius, unsigned int slices)**: disegna un cono, usato per la rappresentazione del cursore aptico.

  INPUT

  - **float rotation[16]**: matrice che indica posizione e rotazione del cursore aptico;
  - **double radius**: raggio delle circonferenza alla base del cono;
  - **unsigned int slices**: numero di suddivisioni del cono, coincide con il detail level.

- **void Sight(Vector position, double radius)**: disegna un mirino, utilizzato per la visualzzazione del cursore aptico nel caso di renderizzazine aptica del potenziale.

  INPUT

  - **Vector position**: posizione del centro del mirino;
  - **double radius**: grandezza del mirino rappresentato.

- **void Circle(Vector position, double radius, unsigned int slices)**: disegna 3 cerchi concentrici, usato per la selezione degli atomi e per mostrare lo scheletro della mpolecola nella visualizzazione dei volumi isopotenziali.

  INPUT

  - **Vector position**: posizione del centro dei 3 cerchi;
  - **double radius**: raggio delle circonferenze;
  - **unsigned int slices**: numero di suddivisioni di ciascuna circonferenza, coincide con il detail level.

- **void DrawVector(Vector position, Vector vector)**: disegna un vettore come un cilindro più un cono, usato per la visualizzazione del vettore forza associato al cursore aptico.

  INPUT

  - **Vector position**: posizione del punto di partenza del vettore;
  - **Vector vector**: vettore da rappresentare.

- **void DrawVectorLight(Vector position, Vector vector)**: disegna un vettore 'leggero' composto da un trattino ed un punto che ne indica la freccia. Usato per la rappresentazione del gradiente e del gradiente secondo.

  INPUT

  - **Vector position**: posizione del punto di partenza del vettore;
  - **Vector vector**: vettore da rappresentare.

- **void hapticSelection(Vector position)**:verifica se un atomo è al momento pre-selezionato apticamente, inoltre si occupa di controllare se un atomo è stato agganciato o meno nel momento dell'effettiva selezione.

  INPUT

  - **Vector position**: posizione da controllare, indica la posizione del cursore aptico.

- **void DrawMeter()**: disegna la scala metrica in basso a sinistra, che identifica la lunghezza di un Angostrom.

- **void DrawInfo()**: disegna le informazioni salienti nel rettangolo in alto.

- **void ResetNormalLight()**: riporta le luci alla condizione vuota in cui era presente la sola luce ambientale, utilizzato quando si vuole rappresentare il MEP, in cui non è utile rappresentare anche le ombre.

- **void SetNormalLight()**: imposta le 3 luci presenti nel caso di visualizzazione geometrica della molecola[7].

Metodi Pubblici

- **GraphicEngine(void)**: costruttore, inizializza tutte le variabili della classe.

- **∼GraphicEngine(void)**: distruttore di default, non fa nulla.

- **void Init(Molecule* molecule, UserInterfaceEngine* userInterfaceEngine, HapticEngine* hapticEngine,AudioEngine* audengine,DynamicEngine* Deng)**: metodo collegare l'engine grafico agli altri engine utilizzati e alla molecola.

  INPUT

  - **Molecule* molecule**: riferimento alla molecola presente nel sistema;
  - **UserInterfaceEngine* userInterfaceEngine**: riferimento all'engine interfaccia grafica;
  - **HapticEngine* hapticEngine**: riferimento all'engine aptico;
  - **AudioEngine* audengine**: riferimento all'engine sonoro;
  - **DynamicEngine* Deng**: riferimento all'engine dinamico.

- **void MainLoop()**: funzione che fornisce l'ingresso al main loop delle librerie OpenGL.

[7] ricordiamo una bianca in alto a destra, una rossa in basso a destra e una blu in alto a sinistra

- **void DisplayGraphicCallback(void)**: funzione di callback idle che svuota i buffer di colore e profondità, disegna tramite **draw** e effettua lo swap dei frame buffers.
- **void KeyboardGraphicCallback(unsigned char key, int x, int y)**: funzione di callback per gli input da tastiera.

  INPUT

  - **unsigned char key**: indicatore del tasto premuto nella tastiera;
  - **int x**: posizione del cursore del mouse nello schermo al momento della pressione del tasto da tastiera, asse x;
  - **int y**: posizione del cursore del mouse, asse y;
- **void MouseGraphicCallback(int button, int state, int x, int y)**: funzione di callback per la pressione o il rilascio di un pulsante del mouse.

  INPUT

  - **int button**: indice del tasto premuto;
  - **int state**: stato del tasto premuto, può essere up o down dipendentemente dal fatto che sia stato premuto o rilasciato;
  - **int x**: posizione del cursore del mouse, asse x;
  - **int y**: posizione del cursore del mouse, asse y;
- **void PassiveMotionCallback(int x, int y)**:funzione di callback che viene richiamata ogni qualvolta viene mosso il mouse sullo schermo, anche senza che nessun pulsante venga premuto. Viene utilizzata per la preselezione e per la visualizzazione del quadratino sulla scala colorimetrica.

  INPUT

  - **int x**: posizione del cursore del mouse, asse x;
  - **int y**: posizione del cursore del mouse, asse y;
- **void MotionGraphicCallback(int x, int y)**: funzione di callback per i movimenti del mouse quando è premuto un pulsante. Usato per ruotare e traslare la molecola, per cambiare i colori della scala oppure per cambiare gli estremi dell'intervallo dei volumi ispotenziali.

  INPUT

  - **int x**: posizione del cursore del mouse, asse x;
  - **int y**: posizione del cursore del mouse, asse y;

5.4 Classe HapticApplication

La classe HapticApplication fa da contenitore a tutti gli engine. Inizializza gli engine e li associa gli uni agli altri secondo le necessità.

Attributi privati

- **Molecule* HapticMolecule**: puntatore all'oggetto molecola.
- **GraphicEngine* GEngine**: puntatore all'engine grafico.
- **UserInterfaceEngine* UIEngine**: puntatore all'engine interfaccia utente.
- **HapticEngine* HEngine**: puntatore all'engine aptico.
- **DynamicEngine* DEngine**: puntatore all'engine dinamico.
- **AudioEngine* AEngine**: puntatore all'engine sonoro.
- **bool HapticEngineEnabled**: indica se lo strumento aptico è abilitato, se cioè deve essere predisposto l'interfacciamento aptico o meno.

Metodi pubblici

- **HapticApplication(void)**: costruttore di default della classe, contiene l'allocazione dei motori utilizzati dall'applicazione e della classe di gestione della molecola.
- **∼HapticApplication(void)**: distruttore di default della classe, dealloca gli engine e la classe di gestione della molecola.
- **bool Init(int* argcPointer, char** argv, bool hapticEngineEnabled)**: inizializza i vari engine associandoli gli uni agli altri secondo le necessità.

  INPUT

  – **int* argcPointer**: non usato, viene ripreso direttamente dai parametri in ingresso al main dell'applicazione;
  – **char** argv**: non usato, parametro passato dal main;
  – **bool hapticEngineEnabled**: indica se devono essere inizializzati anche l'engine dinamico e aptico.

  OUTPUT

  - **bool**: true se tutto è andato a buon fine, false se ci sono stati degli errori.

- **bool MainLoop(void)**: richiama il loop di disegno della finestra interno all'engine grafico.

  OUTPUT

  - **bool**: false se ci sono stati errori, true altrimenti

- **GraphicEngine* GetGraphicEngine()**: restituisce il puntatore all'engine grafico.

- **HapticEngine* GetHapticEngine()**: restituisce il puntatore all'engine aptico.

- **UserInterfaceEngine* GetUserInterfaceEngine()**: restituisce il puntatore all'engine interfaccia utente.

- **AudioEngine* GetAudioEngine()**: restituisce il puntatore all'engine audio.

- **DynamicEngine* GetDynamicEngine()**: restituisce il puntatore all'engine dinamico.

Altre funzioni

Infine vi è una funzione separato dalla classe :

- **void UIControlsUpdateCallback()**: definisce la funzione di callback per l'aggiornamento dei controlli dell'interfaccia utente per eventi esterni all'interfaccia stessa.

5.5 Classe HapticEngine

La classe **HapticEngine** racchiude tutti i metodi per gestire l'interfacciamento aptico con il sistema. Tale classe fa uso di due strutture usate per la memorizzazione dello stato dello strumento aptico.

5.5.1 Struttura HapticDeviceState

Memorizza lo stato del cursore aptico, viene utilizzata in vari frangenti, quali disegno del cursore aptico, calcolo e impostazione di una nuova forza sullo strumento e selezione aptica della molecola.

- **Vector Position**: vettore che indica la posizine del proxy o del cursore aptico secondo la modalità di interazione aptica del sistema.
- **Vector RealPosition**: in caso di interazione geometrica con il sistema indica la posizione reale del cursore, differisce dal precedente in quanto **Position** contiene la posizione del proxy nell'interfacciamento geometrico con la molecola.
- **Vector Force**: indica l'ultimo vettore forza erogato dal PHANToM.
- **float Rotation[16]**: definisce la rototraslazione dell'end effector dello strumento aptico.

5.5.2 Struttura HapticDeviceWorkspace

Quest'altra struttura definisce lo spazio di lavoro del sistema aptico.

- **Vector TopRightFront**: posizione del punto più in alto, più vicino e più a destra dello spazio 3d raggiungibile dal puntatore del device.
- **Vector LowLeftBack**: posizione del punto più in basso, più lontano e più a sinistra raggiungibile dal puntatore del device.

Metodi e attributi privati

Attributi

- **unsigned int counter**: variabile contatore utilizzato per contare il numero di cicli aptici trascorsi dall'ultima riproduzione dell'effetto sonoro indicante il potenziale sul cursore aptico. Infatti ogni 10 cicli aptici viene riprodotto tale suono.
- **double FramesPerSecond; unsigned int FramesCount; DWORD TimeElapsed;**: variabili utilizzate per il conteggio dei frame aptici.
- **void (*UIControlsUpdateCallback)(void)**: punatore alla funzione di update dell'intefaccia utente.
- **HHD HapticDeviceHandle**: variabile da usare come identificativo del supporto aptico con la libreria HDAPI.
- **HHLRC HapticDeviceContext**: contesto aptico in cui lavora l'H-LAPI.

- **HDSchedulerHandle ForceGenerationHapticCallbackHandle**: identificativo della funzione di callback schedulata per la generazione della forza relativa al campo di potenziale.
- **HapticDeviceState LastRetrievedHapticDeviceState**: variabile contenente l'ultimo stato calcolaro del sistema aptico.

Metodi

- **bool HDInitialization(void)**: metodo di inizializzazione dell'engine aptico, inizializza lo scheduler e associa le funzioni di callback della HLAPI, quali tocco e distacco da un oggetto e pressione dei pulsanti.

  OUTPUT

  - **bool**: false se ci sono stati errori, true altrimenti.
- **void DeviceButtonsStateCheck(void)**: metodo per controllare lo stato dei pulsanti presenti sul device aptico.

Metodi e attributi pubblici

Attributi

- **HLuint* MoleculeShapeId**: identificativo della molecola dal punto di vista aptico, rappresenta un array contenente l'identificativo di forma aptica di ognuno degli atomi che lo compongono.
- **HLuint BondsShapeId**: identificativo di forma dei legami della molecola.
- **HLuint MinimaxShapeId**: identificativo di forma delle palline indicanti i minimi e i massimi di potenziale.
- **unsigned int counter2**:contatore utilizzato per il conteggio dei frame grafici avvenuti dal tocco della molecola, viene utilizzato per la gestione dell'effetto sonoro riferito allo strisciamento sulla superficie molecolare. Infatti calcola 10 frame prima di riprodurre l'effetto sonoro, si ottiene così un distacco temporale tra il suono indicante il tocco di un oggetto e il suono relativo allo strisciamento.
- **float rotmat[16]**: matrice da usare come base per il calcolo della rotazione finale quando l'oggetto viene agganciato, serve per mantenere continuità rispetto allo stato precedente della rototraslazione della molecola nel sistema.

- **bool rotmatok**: indica se la matrice originaria è stata correttamente salvata, inoltre serve a tenere conto di successive rotazioni in seguito alla rotazione della molecola tramite altri strumenti. Si può vedere come un dirty bit sulla matrice **rotmat**.
- **bool selecting**: indica se si è in modalità selezione a seguito della pressione del tasto del PHANToM in prossimità di un'atomo della molecola.
- **bool catched**: indica se l'oggetto è stato agganciato correttamente raggiungendo il nucleo dell'atomo in modalità selezione.
- **int selectingIndex**: definisce l'indice dell'atomo attualmente in fase di selezione aptica.
- **int ForceGain**: indica il moltiplicatore di forza o carica del sistema, usato sia per la simulazione dinamica, che per la definizione del valore della carica di prova nella simulazione aptica del force field molecolare.
- **Molecule* HapticMolecule**: riferimento alla molecola attuale.
- **AudioEngine* AEngine**: riferimento all'engine audio, viene utilizzato per riprodurre i suoni relativi all'interazione aptica.
- **DynamicEngine* DEngine**: riferimento all'engine dinamico, utilizzato dal sistema aptico per l'associazione delle forze di spinta alla molecola.
- **HapticEngine(void (*uiControlsUpdateCallback)(void))**: costruttore, inizializza tutte gli attributi della classe ad un valore di default.

  INPUT

  - **void (*uiControlsUpdateCallback)(void)**: funzione di callback per l'aggiornamento dell'interfaccia utente.
- **~HapticEngine(void)**: distruttore di default.
- **bool Init(Molecule* molecule,AudioEngine* Aeng,DynamicEngine * DEng)**: inizializza l'engine aptico con gli altri engine utilizzati dallo stesso.

  INPUT

  - **Molecule* molecule**: riferimento alla molecola.
  - **AudioEngine* Aeng**: riferimento all'engine audio.

  - **DynamicEngine * DEng**: riferimento all'engine dinamico.

    OUTPUT

  - **bool**: indica se ci sono stati o meno errori nella fase di inizializzazione.

- **void IncrementForceGain()**: incrementa il guadagno in forza o in carica, dipendentemente dal contesto.
- **void DecrementForceGain()**: decrementa il guadagno in forza.
- **int GetForceGain()**: restituisce il valore della carica di prova o il guadagno in forza del cursore aptico.
- **void SetForceGain(int forceGain)**: imposta il valore del guadagno in forza direttamente.
- **bool StartForceGeneration(void)**: schedula la funzione di callback per la percezione della forza derivante dal potenziale molecolare, inoltre genera le forme necessarie alla percezione aptica delle geometrie molecolari.

  OUTPUT

  - **bool**: indica se ci sono stati o meno errori nella fase di schedulazione.

- **bool StopForceGeneration(void)**: unschedula la funzione per la gestione del force field.

  OUTPUT

  - **bool**: false se è avvenuto un errore, true altrimenti.

- **bool WaitForForceGenerationHapticCallbackCompletion(void)**: attende il completamento della funzione di callback associata alla generazione del force field.

  OUTPUT

  - **bool**: false se la funzione **ForceGenerationHapticCallback** è già terminata, true altrimenti.

- **HapticDeviceState GetSynchronousHapticDeviceState(void)**: ottiene lo stato del device aptico in maniera sincrona, schedulando la

funzione di callback in caso di HDAPI, richiamando la funzione opportuna tramite HLAPI, in base cioè al contesto di renderizzazione aptica[8].

OUTPUT

- **HapticDeviceState**: contiene tutte le informazioni utili su posizione forza e trasformazione del cursore aptico.

- **HapticDeviceWorkspace GetHapticDeviceWorkspace(void)**: calcola e restituisce lo spazio di lavoro del sistema aptico.

  OUTPUT

  - **HapticDeviceWorkspace**: definisce lo spazio di lavoro del contesto aptico.

- **double GetFramesPerSecond()**: restituisce i frame al secondo aptici del sistema.

- **bool ForceGenerationHapticCallback(void)**: funzione di callback per la generazione della forza dovuta al campo di potenziale. All'interno di essa avviene tutta la computazione relativa all'aggiornamento della forza sul device aptico, sia che la forza derivi dalla mappa del gradiente, sia che venga calcolata tramite formule. Questa funzione inoltre calcola i frame rate aptici e genera i suoni relativi al potenziale.

  OUTPUT

  - **bool**: restituisce false se ci sono stati errori nella computazione, true altrimenti.

- **void DeviceStateHapticCallback(void)**: comunica con il device aptico per aggiornare la variabile **LastRetrievedHapticDeviceState** contenente le ultime informazioni relative a posizione e forza del device aptico. Utilizzato nell'ambito dell'HDAPI.

- **void ComputePosition(void)**: comunica con il PHANToM per ottenere la posizione del proxy, la posizione dell'end-effector, la forza erogata e la matrice di rototraslazione che individua il cursore aptico. Metodo usato nell'ambito dell'HLAPI.

- **void ExitHapticCallback(void)**: funzione richiamata all'uscita dal programma, disattiva il PHANToM e chiude lo scheduler.

[8] infatti ci sono differenze tr le due modalità, ad esempio nel caso di percezione aptica della geometria la posizione e definita dal proxy e non dalla posizione fisica del device

5.5.3 Funzioni esterne di callback

All'esterno della classe vi sono le funzioni di callback aptico usate nel contesto del sistema.

- **HDCallbackCode HDCALLBACK ForceGenerationHapticCallbackHandler(void* data)**: richiama la funzione **ForceGenerationHapticCallback** interna alla classe.

  INPUT

  – **void* data**: parametro delle funzioni di callback che permette di passare un oggetto alla funzione, in questo caso lo stesso Engine Aptico.

  OUTPUT

  – **HDCallbackCode**: codice di ritorno delle funzioni di callback, può essere **HD_CALLBACK_DONE** se si vuole che la funzione non venga più richiamata o **HD_CALLBACK_CONTINUE** altrimenti. In questo caso vale **HD_CALLBACK_CONTINUE** a meno di errori o terminazione dello sheduler.

- **HDCallbackCode HDCALLBACK DeviceStateHapticCallbackHandler(void* data)**: richiama la funzione **DeviceStateHapticCallback** interna alla classe.

  INPUT

  – **void* data**: l'oggetto che identifica l'engine aptico.

  OUTPUT

  – **HDCallbackCode**: in questo caso vale sempre **HD_CALLBACK_DONE** in quanto richiesta singola e sincrona.

- **void HLCALLBACK touchShapeCallback(HLenum event,HLuint object,HLenum thread,HLcache *cache,void *userdata)**: funzione di callback per l'evento tocco nella percezione aptica della geometria molecolare. Oltre a gestire l'effetto sonoro corrispondente attiva la funzione di callback **contactMotionShapeCallback**.

  INPUT

  – **HLenum event**: l'evento che genera la chiamata alla funzione di callback, **HL_EVENT_TOUCH** in questo caso.

  - **HLuint object**: l'oggetto a cui l'evento si riferisce, identificato dal suo id di forma, ad esempio un atomo.
  - **HLenum thread**: il thread che gestisce la funzione di callback, può essere **HL_CLIENT_THREAD** o **HL_COLLISION_THREAD** in base a quando si vuole l'evento venga gestito. In questo caso si è scelto **HL_CLIENT_THREAD** in quanto l'evento deve essere gestito ad ogni frame grafico.
  - **HLcache *cache**: un oggetto contenitore dello stato del device aptico al momento dell'evento richiamante.
  - **void *userdata**: un puntatore ad un oggetto che si vuole passare come parametro, in questo caso lo stesso Engine Aptico.

- **void HLCALLBACK untouchShapeCallback(HLenum event,HLuint Object3d,HLenum thread,HLcache *cache,void *userdata)**: funzione di callback richiamata al momento del distacco da un oggetto nella scena 3d. Disattiva la funzione **contactMotionShapeCallback**.

  INPUT

  - **HLenum event**: questa volta l'evento è **HL_EVENT_UNTOUCH**.
  - **HLuint object**: sempre l'oggetto a cui l'evento si riferisce, identificato dal suo id di forma, ad esempio un atomo.
  - **HLenum thread**: si è scelto **HL_CLIENT_THREAD** anche in questo caso.
  - **HLcache *cache**: stato del device aptico al momento del distacco dall'oggetto.
  - **void *userdata**: puntatore all'Engine Aptico.

- **void HLCALLBACK contactMotionShapeCallback(HLenum event,HLuint Object3d,HLenum thread,HLcache *cache,void *userdata)**: funzione di callback che gestisce gli eventi sonori legati al contatto tra il proxy e la superficie della molecola, associa inoltre la forza all'oggetto dinamico in modeo da gestire la simulaizione dinamica.

  INPUT

  - **HLenum event**: l'evento è **HL_EVENT_MOVE** che viene lanciato ogni volta che il cursore aptico si muove più di una quantità predefinita.
  - **HLuint object**: irrilevante.

  - **HLenum thread**: si è scelto comunque **HL_CLIENT_THREAD** dato che il ciclo dinamico avviene contestualmente al ciclo grafico.
  - **HLcache *cache**: ultimo stato del device aptico.
  - **void *userdata**: puntatore all'Engine Aptico.

- **void HLCALLBACK mousePressingCallback(HLenum event,HLuint Object3d,HLenum thread,HLcache *cache,void *userdata)**: funzione di callback al momento della pressione del tasto dello strumento aptico, avvia la fase di selezione se abbastanza vicino ad un atomo, altrimenti ferma la molecola nella simulazione dinamica. Inoltre attiva la funzione seguente **catchedMotionCallback**.

  INPUT

  - **HLenum event**: questa volta l'evento è **HL_EVENT_1BUTTONDOWN**, cioè quando il tasto è premuto effettivamente.
  - **HLuint object**: irrilevante.
  - **HLenum thread**: sempre **HL_CLIENT_THREAD**.
  - **HLcache *cache**: irrilevante, la preselezione non fa uso di questo strumento.
  - **void *userdata**: puntatore all'Engine Aptico.

- **void HLCALLBACK catchedMotionCallback(HLenum event,HLuint Object3d,HLenum thread,HLcache *cache,void *userdata)**: funzione di callback che si occupa di rototraslare la molecola secondo la rototraslazione del puntatore aptico.

  INPUT

  - **HLenum event**: l'evento è **HL_EVENT_MOVE**.
  - **HLuint object**: irrilevante.
  - **HLenum thread**: sempre e comunque **HL_CLIENT_THREAD**, soprattutto per non sovracaricare il sistema con continui aggiornamenti inutili.
  - **HLcache *cache**: ultimo stato del device aptico, utilizzato per ottenere l'ultima rototraslazione.
  - **void *userdata**: puntatore all'Engine Aptico.

- **void HLCALLBACK mouseReleasingCallback(HLenum event,HLuint Object3d,HLenum thread,HLcache *cache,void *userdata)**: funzione di callback per chiudere la fase di selezione aptica al rilascio del

pulsante, inoltre disattiva la funzione **catchedMotionCallback**. Infine si occupa di correggere l'ultima traslazione tra il nucleo dell'atomo selezionato e il baricentro.

INPUT

- **HLenum event**: l'evento è **HL_EVENT_1BUTTONUP**, cioè quando il tasto viene rilasciato.
- **HLuint object**: irrilevante.
- **HLenum thread**: sempre **HL_CLIENT_THREAD**.
- **HLcache *cache**: irrilevante.
- **void *userdata**: puntatore all'Engine Aptico.

5.6 Classe MEP

La classe MEP serve a mantenere in memoria le griglie di potenziale,gradiente e gradiente secondo, nonché ad allocare e deallocare tali strutture dati. La classe si occupa anche di ottenere i dati da un file di testo e di gestire gli accessi indicizzati a tali griglie.

Attributi e metodi privati

Attributi

- **double *** GridValues**: puntatore ad un array 3d, contenente i valori di potenziale. I 3 livelli di indirezione[9] si riferiscono alle 3 direzioni spaziali.
- **Vector *** GridGradients**: puntatore ad un array tridimensionale contenente i vettori che identificano il gradiente di potenziale.
- **Vector *** GridGradientsII**: array 3d contenente i vettori del gradiente secondo.
- **Vector * IsoPotential**: array di vettori che contiene le posizioni di tutti i punti il cui valore di potenziale è compreso nell'intervallo di selezione
- **unsigned int numIsoPot**: numero di elementi dell'array contenente l'informazione sui punti isopotenziali.

[9] si tratta infatti di un array contenente array che contengono ulteriormente array

- **unsigned int numMin**: numero di minimi assoluti o estremamente prossimi individuati.
- **unsigned int numMax**: numero di massimi individuati.
- **unsigned int GridXSize, GridYSize, GridZSize**: indici delle 3 dimensioni delle griglie, indicano il numero di elementi sui 3 assi dell'array.
- **double GridXOrigin, GridYOrigin, GridZOrigin**: 3 coordinate del punto di riferimento della griglia, identificate dall'angolo con le coordinate minori su tutti e tre gli assi del parallelepipedo che identifica la griglia. Permette di posizionare la griglia nello spazio 3d.
- **double GridXIncrement, GridYIncrement, GridZIncrement**: identifica l'incremento tra una cella e l'altra delle griglie di poteziale su ognuno dei tre assi.
- **double MaxPositiveValue; double MinNegativeValue**: indicano rispettivamente il valore del minimo e del massimo di potenziale.
- **Vector* MaxPositivePosition**: array contenente le posizioni spaziali dei massimi.
- **Vector* MinNegativePosition**: array contenente le posizioni spaziali dei minimi.
- **double MaxGradient**: valore massimo, in modulo, del gradiente del potenziale.
- **double MaxGradientII**: valore massimo, in modulo, del gradiente secondo del potenziale.

Metodi

- **void AllocateGridValues(void)**: alloca lo spazio necessario a contenere la griglia di potenziale.
- **void DeallocateGridValues(void)**: dealloca la griglia di potenziale, deallocando i puntatori in ognuno dei tre livelli di array.
- **void AllocateGridGradients(void)**: alloca lo spazio necessario a contenere la griglia di vettori che indica il gradiente di potenziale.
- **void DeallocateGridGradients(void)**: dealloca la griglia del gradiente e rispettivi vettori.

- **void AllocateGridGradientsII(void)**: alloca lo spazio necessario a contenere la griglia di vettori che indica il gradiente secondo del potenziale.
- **void DeallocateGridGradientsII(void)**: dealloca la griglia del gradiente secondo e rispettivi vettori.
- **void AllocateIsoPotential(int num)**: alloca l'array contenente i vettori delle posizioni dei punti isopotenziali.

  INPUT

  - **int num**: dimensione dell'array da allocare.
- **void DeallocateIsoPotential(void)**:dealloca l'array contenente i vettori che identificano le posizioni dei punti isopotenziali.
- **void AllocateMaxMinPosition(int maxs,int mins)**: alloca gli array contenenti i minimi e i massimi di potenziale.

  INPUT

  - **int maxs**: numero di massimi trovati da allocare.
  - **int mins**: numero di minimi da allocare.
- **void DeallocateMaxMinPosition(void)**: dealloca gli array contenenti minimi e massimi di potenziale.

Metodi pubblici

- **MEP()**:costruttore di default,inizializza tutti gli attributi a NULL o ad un valore di default.
- **∼MEP()**:distruttore, dealloca i tre array tridimensionali, gli array dei minimi e massimi e l'array contenenti i punti isopotenziali.
- **unsigned int GetGridXSize()**: restituisce il numero di elementi della griglia sull'asse x.
- **unsigned int GetGridYSize()**: restituisce la dimensione della griglia sull'asse y.
- **unsigned int GetGridZSize()**: restituisce la dimensione sull'asse z.
- **double GetGridXOrigin()**: restituisce la coordinata x del punto di origine della griglia.

- **double GetGridYOrigin()**: restituisce la coordinata y dell'origine.
- **double GetGridZOrigin()**: restituisce la coordinata z dell'origine
- **double GetGridXIncrement()**: restituisce l'incremento della griglia sull'asse x.
- **double GetGridYIncrement()**: restituisce l'incremento sull'asse y.
- **double GetGridZIncrement()**: restituisce l'incremento sull'asse z.
- **double GetMEPValue(Vector position)**: ritorna il valore di potenziale nella posizione identificata dal vettore V, individuando la cella dell'array in questo modo:

$$(i, j, k) = (\left\lfloor \frac{V_x - G_{o_x}}{G_{i_x}} \right\rfloor, \left\lfloor \frac{V_y - G_{o_y}}{G_{i_y}} \right\rfloor, \left\lfloor \frac{V_z - G_{o_z}}{G_{i_z}} \right\rfloor) \qquad (5.1)$$

  dove (i, j, k) rappresentano gli indici dell'array 3d, mentre G_o sono le coordinate dell'origine e G_i gli incrementi. Inoltre viene fatta un'interpolazione con le celle adiacenti

  INPUT

  – **Vector position**: vettore che indica la posizione di cui si vuole conoscere il potenziale.

  OUTPUT

  – **double**: valore del potenziale associato alla posizione definita dal vettore, questo valore viene ottenuto per interpolazione a partire dalla cella (i, j, k).

- **Vector GetGradient(Vector position)**:ritorna il vettore interpolato del gradiente a partire dalla cella (i, j, k) identificato dalla posizione in input, come mostrato nell'equazione 5.1

  INPUT

  – **Vector position**: vettore che indica la posizione in cui si vuole conoscere il gradiente.

  OUTPUT

  – **Vector**: vettore gradiente corrispondente alla posizione V.

- **Vector GetGradientII(Vector position)**: ritorna il vettore interpolato del gradiente secondo come nel caso precedente

  INPUT

  – **Vector position**: vettore che indica la posizione di cui si vuole conoscere il gradiente secondo.

  OUTPUT

  – **Vector**: vettore gradiente secondo corrispondente alla posizione V.

- **Vector* GetIsoPotential()**: restituisce il puntatore all'array contenente le posizioni dei punti isopotenziali.

- **unsigned int GetNumIsoPot()**: restituisce il numero di elementi dell'array delle posizioni dei punti isopotenziali.

- **void CalcIsopotential(double min, double max)**: verifica quali punti di potenziale appartengono all'intervallo voluto e riempie l'array corrispondente.

  INPUT

  – **double min**: valore minimo dell'intervallo di selezione dei punti isopotenziali.

  – **double max**: valore massimo dell'intervallo di selezione dei punti isopotenziali.

- **bool LoadFromFile(string fileName)**:carica la griglia di potenziale da un file e calcola le griglie del gradiente primo e secondo, inoltre calcola i minimi e i massimi di potenziale e il massimo del gradiente primo e del gradiente secondo.

  INPUT

  – **string fileName**: nome del file di estensione mep da cui caricare la griglia di potenziale.

  OUTPUT

  – **bool**: se non ci sono stati problemi nell'interazione con il file restituisce true, altrimenti restituisce false[10].

[10] file mancante, formattazione dei dati sbagliata, etc.

- **Vector* GetMaxPositivePosition()**: ritorna l'array contenente i vettori che identificano le posizioni dei massimi di potenziale.
- **unsigned int GetNumMax()**: ritorna il numero di massimi trovati.
- **Vector* GetMinNegativePosition()**: ritorna l'array contenente i vettori che identificano le posizioni dei minimi di potenziale.
- **unsigned int GetNumMin()**: ritorna il numero di minimi trovati.
- **double GetMaxPositiveValue()**: ritorna il valore del massimo di potenziale
- **double GetMinNegativeValue()**: ritorna il valore del minimo di potenziale
- **double GetMaxGradientValue()**: ritorna il valore del massimo del gradiente, calcolato come modulo del vettore di lunghezza massima.
- **double GetMaxGradientIIValue()**: ritorna il valore del massimo del gradiente secondo.

5.7 Classe Molecule

La classe Molecule definisce tutte le proprietà di interesse della molecola caricata. Fa uso di strutture definite nel contesto della classe, di cui la più importante è la struttura MoleculeState che definisce lo stato della molecola, quale posizione nello spazio, opzioni attivate, rotazione etc.

5.7.1 Struttura Minimum

Questa struttura serve a definire le informazioni sui minimi utili al calcolo del potenziale di lennard jones.

- **double distance**: distanza d del minimo più vicino.
- **double dist6**: valore di distanza elevato alla sesta potenza d^6, viene precalcolato per comodità coomputazionale.
- **double depth**: profondità della buca di potenziale del minimo più vicino.

5.7.2 Struttura Color

Tale struttura serve a definire gli elementi della scala di colore logaritmica.

- **double limit**: indica la posizione nella scala sotto forma di frazione $1/l$ della totalità.
- **double Rvalue**: valore compreso da 0 a 1 che indica la componente rosso del colore.
- **double Gvalue**: componente verde del colore.
- **double Bvalue**: componente blu.
- **bool last**: indica se è l'ultimo colore della scala
- **bool neg**: indica se il colore fa riferimento a potenziale negativo
- **Color* post**: puntatore al colore seguente.

5.7.3 Struttura MoleculeState

La struttura MoleculeState contiene lo stato della molecola con tutte le proprietà attivate e la posizione e rotazione nello spazio.

- **string name**: nome della molecola ottenuto dal nome del file da cui i dati molecolari sono stati estratti.
- **double ScaleFactor**: fattore di scala della molecola.
- **double YRotationDegrees**: valore in gradi della rotazione rispetto all'asse y dovuta all'azione del mouse. Ogni frame viene 'assorbito' dalla matrice di rotazione.
- **double XRotationDegrees**: valore temporaneo della rotazione rispetto all'asse x.
- **float RotationMatrix[16]**:matrice di rotazione della molecola.
- **VisualizationMode VisualMode**: variabile che identifica la modalità di visualizzazione sotto forma del raggio degli atomi della molecola, viene scelta tra una enumerazione di elementi:
  - **VM_WIREFRAME**: visualizzazione wireframe identificata da un raggio degli atomi pari a 0.15;
  - **VM_BALLNSTICK**: visualizzazione ball'n'stick identificata da un raggio degli atomi pari a 0.4;

  - **VM_SPACEFILLCOVALENT**: utilizzo del raggio covalente per gli atomi;
  - **VM_SPACEFILLVDW**: utilizzo del raggio di Van Der Waals;
  - **VM_INVISIBLE**: raggio atomico imposto a 0;
  - **VM_CUSTOM**: raggio atomico customizzato.

- **bool ShowVector**: identifica se deve essere mostrato a video il vettore riferito alla forza sul cursore aptico.
- **bool ShowGraph**: identifica se deve essere visualizzato il grafico di potenziale.
- **bool ShowLog**: identifica se si usa o meno la scala logaritmica.
- **unsigned int LogLevel**: livello della scala iperlogaritmica.
- **bool ShowLines**: indica se devono essere mostrate le linee di forza.
- **bool ShowLinesDirection**: indica se devono essere mostrati i vettori direzione sulle linee di forza.
- **bool PerspectiveView**: definisce se è attiva la visualizzazione prospettica.
- **bool externalChange**: definisce se ci sono stati cambiamenti alla rototraslazione della molecola esterni al sistema dinamico.
- **Vector TranslationVector**: vettore di traslazione, indica la posizione del centro della molecola.
- **int DragStartX, DragStartY**: coordinate del punto di origine sullo schermo del trascinamento del mouse[11].
- **bool addLennardJonesCorr**: indica se deve essere usata la correzione di LennardJones per il calcolo del potenziale tramite sistema a cariche coulombiane.
- **MEPMode Mmode**: definisce la modalità con cui deve essere calcolato il MEP e la forza. Scelto sulla base di una enumerazione:
  - **SPERIMENTAL_GRID**: usa i dati nelle griglie di potenziale.
  - **LOWDIN_CHARGE**: usa le cariche di Lowdin nel modello coulombiano.

[11] utilizzato nelle traslazioni e rotazioni attraverso il mouse

  - **MULLIKEN_CHARGE**: usa le cariche di Mulliken.
  - **ONLY_LENNARD_JONES**: visualizza il potenziale e la forza dovuti al solo potenziale di Lennard Jones
  - **CUSTOM_CHARGE**: usa cariche customizzate dall'utente.

- **InfoTypeMode itype**: definisce il tipo di informazione da visualizzare a schermo, anche in questo caso si fa uso di una enumerazione per la scelta:

  - **NORMAL**: visualizzazione geometrica della molecola.
  - **INFO_MEP**: visualizzazione del potenziale molecolare.
  - **FORCE**: visualizzazione della forza.
  - **FORCE_GRADIENT**: visualizzazione del gradiente della forza.
  - **ISOPOTENTIAL**: visualizzazione dei punti isopotenziali.

- **bool ShowMinMax**: indica se devono essere mostrati i minimi e i massimi.

- **double minIso**: valore limite inferiore dell'intervallo dei punti isopotenziali.

- **double maxIso**:valore limite superiore dell'intervallo dei punti isopotenziali.

- **bool Moving**: definisce se l'oggetto sta traslando attraverso l'intervento del mouse.

- **bool Rotating**: definisce se l'oggetto sta ruotando a causa dell'utilizzo del tasto destro del mouse.

- **bool fixed**: definisce se la simulazione dinamica è attivata o meno.

- **Color* colorScale**: lista che contiene i valori RGB dei punti salienti della scala logaritmica.

Attributi privati

- **OBMol OBMolecule**: indica l'oggetto molecola nell'accezione definita da OpenBabel.

- **OBElementTable OBTable**: identifica una tavola periodica degli elementi nel formato definito da OpenBabel.

- **MoleculeState* State**: puntatore allo stato della molecola.
- **Vector center**: vettore che indica la posizione del centro geometrico della molecola rispetto alle coordinate ottenute dal file di importazione.
- **Vector centerOfMass**: vettore che indica il baricentro, cioè il centro di massa, calcolato come media pesata della posizione dei vari atomi nelle coordinate del sistema importato.
- **double maxAtomMass**: massa dell'atomo più massivo all'interno della molecola.
- **double totalMass**: massa totale della molecola
- **bool selected**: indica se vi è un atomo selezionato tramite il mouse.
- **unsigned int selectedAtom**: indice dell'atomo selezionato tramite il mouse.
- **double* radiuslist**: array contenente il raggio customizzato di ognuno degli atomi componenti la molecola.
- **Minimum* nearestMinList**: array di Minimum contenente il minimo più vicino per ognuno degli atomi della molecola, usato per Lennard Jones.
- **double* chargeM**: array contenente il valore delle cariche di Mulliken per ognuno degli atomi.
- **double* chargeL**: array contenente il valore delle cariche di Lowdin per ognuno degli atomi.
- **double* chargeCustom**: array contenente il valore delle cariche customizzato per ognuno degli atomi.
- **MEP LoadedMEP**: oggetto MEP associato alla molecola contenente il potenziale molecolare, il gradiente e il gradiente secondo.
- **double MinVisualMEP**: valore locale del minimo di potenziale relativo alla superficie attualmente visualizzata a schermo.
- **double MaxVisualMEP**: valore locale del massimo di potenziale relativo alla superficie attualmente visualizzata a schermo.

Metodi Pubblici

- **Molecule(void)**:costruttore, crea e inizializza le variabili assegnando un valore di default.
- ∼**Molecule(void)**: distruttore, cancella la varibile contenente lo stato della molecola, e libera tutti gli array utilizzati.
- **bool LoadOpenBabelFile(string fileName)**: estrae le informazioni chimiche da un file attraverso la libreria OpenBabel. Inoltre sfrutta queste informazioni per assegnare valori a vari attributi della classe, quali ad esempio centro e baricentro.

  INPUT

  – **string fileName**: nome del file di estensione pdb da cui OpenBabel estrae le informazioni sulla geometria molecolare.

  OUTPUT

  – **bool**: restituisce false se ci sono stati problemi nell'interazione con il file.

- **bool LoadMEPFile(string fileName)**: passa la stringa contenente il nome file al caricatore da file dell'oggetto MEP, dopodiché richiama il metodo **searchMinimum** per la ricerca dei minimi locali più vicini ai vari atomi.

  INPUT

  – **string fileName**: nome del file di estensione mep da cui l'oggetto MEP deve calcolare le griglie di informazioni chimiche.

  OUTPUT

  – **bool**: restituisce false se ci sono stati problemi nell'interazione con il file.

- **bool LoadAtomsChargesFile(string fileName)**: estrae da file le informazioni inerenti le cariche di Lowdin e Mulliken degli atomi della molecola.

  INPUT

  – **string fileName**: nome del file di estensione out da cui estrarre le informazioni.

  OUTPUT

  - **bool**: restituisce false se ci sono stati problemi nell'interazione con il file.

- **unsigned int GetAtomsCount(void)**: restituisce il numero di atomi nella molecola.

- **void GetAtomCoordinates(unsigned int atomIndex, Vector &coordinates)**: metodo usato per ottenere la posizione del nucleo di uno specifico atomo.

  INPUT

  - **unsigned int atomIndex**: indice dell'atomo di cui si vuole conoscere la posizione.
  - **Vector &coordinates**: reference della variabile vettore in cui inserire l'informazione relativa alla posizione del nucleo dell'atomo indicato.

- **vector¡double¿ GetAtomColor(unsigned int atomIndex)**: restituisce il colore standard associato ad un particolare atomo

  INPUT

  - **unsigned int atomIndex**: indice dell'atomo di cui si vuole conoscere il colore

  OUTPUT

  - **vector¡double¿**: vettore contenente le informazioni RGB del colore dell'atomo indicato.

- **double GetAtomCovalentRadius(unsigned int atomIndex)**: restituisce il raggio covalente dell'atomo i-esimo[12].

- **double GetAtomVdwRadius(unsigned int atomIndex)**: restituisce il raggio di Van Der Waals dell'atomo i-esimo.

- **double GetCustomRadius(unsigned int atomIndex)**: restituisce il raggio customizzato dell'atomo i-esimo[13].

- **void setCustomRadius(unsigned int atomIndex,double radius)**:imposta il raggio customizzato all'atomo i-esimo.

[12] per brevità non si visualizzano in questi casi banali le informazioni in merito a input e output

[13] sebbene appaia come i casi precedenti, in questo caso l'informazione non è ottenuta da OpenBabel, bensì dalla lista radiuslist presente negli attributi privati

- **void setCustomCharge(unsigned int atomIndex,double radius)**:imposta la carica customizzata all'atomo i-esimo.
- **string getAtomName(unsigned int atomIndex)**: restituisce il nome dell'atomo i-esimo.
- **double getAtomElectroNeg(unsigned int atomIndex)**:restituisce l'elettronegatività dell'atomo i-esimo.
- **double getAtomMass(unsigned int atomIndex)**:restituisce la massa atomica dell'atomo i-esimo.
- **double getMaxAtomMass()**: restituisce la massa atomica massima tra gli atomi componenti la molecola.
- **unsigned int GetBondsCount(void)**:restituisce il numero dei legami.
- **void GetBondStartCoordinates(unsigned int bondIndex, Vector &coordinates)**:ottiene le coordinate di inizio del legame i-esimo.

  INPUT

  - **unsigned int atomIndex**: indice del legame iesimo di cui si vuole conoscere la posizione iniziale.
  - **Vector &coordinates**: reference al vettore che dovrà contenere le coordinate dell'inizio del legame.
- **void GetBondEndCoordinates(unsigned int bondIndex, Vector &coordinates)**:similmente al precedente, ottiene le coordinate della fine del legame i-esimo.
- **double GetBondLength(unsigned int bondIndex)**: restituisce la lunghezza del legame i-esimo. Indica la distanza di legame tra i due atomi collegati.
- **double GetMEPValue(Vector position)**: chiede all'oggetto MEP il valore del potenziale nella posizione definita dal parametro in ingresso e lo restituisce.
- **double GetLowdinMEPValue(Vector position)**: calcola il potenziale dovuto alle cariche di Lowdin [14] nella posizione definita e lo restituisce.

[14] se addLennardJonesCorr è true allora calcola anche il contributo del potenziale di Lennard Jones

- **double GetMullikenMEPValue(Vector position)**: calcola il potenziale dovuto alle cariche di Mulliken nella posizione definita e lo restituisce.
- **double GetLennardJonesMEPValue(Vector position)**: calcola il potenziale di Lennard Jones nella posizione richiesta e lo restituisce.
- **double GetCustomMEPValue(Vector position)**:calcola il potenziale dovuto alle cariche customizzate nella posizione definita e lo restituisce.
- **double GetMullikenCharge(unsigned int AtomIndex)**: restituisce la carica di Mulliken associata all'atomo i-esimo.
- **double GetLowdinCharge(unsigned int AtomIndex)**: restituisce la carica di Lowdin associata all'atomo i-esimo.
- **double GetCustomCharge(unsigned int AtomIndex)**: restituisce la carica customizzata associata all'atomo i-esimo.
- **Minimum GetNearestMinimum(unsigned int AtomIndex)**: restituisce l'oggetto Minimum che identifica il minimo locale più vicino all'atomo i-esimo. Utilizzato per il calcolo di Lennard Jones.
- **Vector GetGradient(Vector position)**: similmente a **GetMEPValue** richiama il metodo relativo dell'oggetto MEP e restituisce il risultato.
- **Vector GetGradientMullikenCharge(Vector position)**:calcola la forza dovuta alle cariche di Mulliken nella posizione definita e lo restituisce.
- **Vector GetGradientLowdinCharge(Vector position)**:calcola la forza dovuta alle cariche di Lowdin nella posizione definita e lo restituisce.
- **Vector GetGradientLennardJones(Vector position)**:calcola la forza dovuta al potenziele di Lennard Jones nella posizione definita e lo restituisce.
- **Vector GetGradientCustomCharge(Vector position)**:calcola la forza dovuta alle cariche customizzate nella posizione definita e lo restituisce.

- **Vector GetGradientII(Vector position)**:similmente a **GetMEPValue** e **GetGradient** richiama il metodo relativo dell'oggetto MEP e restituisce il risultato.
- **Vector* GetIsoPotential()**: restituisce l'array contenente i vettori che identificano le coordinate dei punti isopotenziali richimando l'opportuna funzione dell'oggetto MEP.
- **unsigned int IsoPotentialSize()**: restituisce la dimensione dell'array di punti isopotenziali.
- **unsigned int numMaxSize()**: restituisce il numero dei massimi assoluti.
- **unsigned int numMinSize()**: restituisce il numero dei minimi assoluti.
- **void CalcIsoPotential()**: forza il ricalcolo dei punti isopotenziali nel nuovo intervallo definito dagli attributi **minIso** e **maxIso** presenti nello stato della molecola. Il calcolo viene effettuato all'interno dell'oggetto MEP.
- **Vector GetGradientHaptic(Vector position)**:restituisce il vettore associato alla posizione assoluta definita in input. A differenza di **GetGradient** questa funzione richiede la trasformazione del vettore posizione prima di richiamare il metodo dell'oggetto MEP. Il vettore **position** rappresenta infatti le coordinate assolute del cursore aptico, al lordo della rototraslazione della molecola nello spazio; va quindi invertita la rototraslazione della molecola per ottenere la posizione corretta con cui richiedere il gradiente all'oggetto MEP.
- **float GetMepHaptic(Vector position)**: come nel caso precedente varia rispetto a **GetMEPValue** per il sistema di riferimento del vettore passato in ingresso. Questo valore viene calcolato per la riproduzione del suono relativo al potenziale molecolare.
- **Vector GetCenter()**:restituisce la posizione del centro geometrico.
- **Vector GetCenterOfMass()**:restituisce la posizione del baricentro.
- **double GetMoleculeMass()**: restituisce la massa della molecola.
- **MoleculeState* GetState()**: restituisce un riferimento allo stato della molecola. Viene restituito il riferimento invece che il semplice valore in modo da rendere lo stato modficabile dall'esterno della classe.

- **double GetMEPMaxPositiveValue()**:restituisce il valore del massimo assoluto di potenziale.
- **double GetMEPMinNegativeValue()**:restituisce il valore del minimo assoluto di potenziale.
- **double GetMinVisualMEP()**:restituisce il valore del minimo locale superficiale del potenziale.
- **double GetMaxVisualMEP()**:restituisce il valore del massimo locale superficiale del potenziale
- **double GetMaxGradient()**:restituisce il valore del massimo del gradiente.
- **double GetMaxGradientII()**:restituisce il valore del massimo del gradiente secondo.
- **Minimum searchMinimum(unsigned int i)**: calcola il valore di Minimum, nel contesto del potenziale di Lennard Jones, dell'atomo i-esimo e lo restituisce.
- **void resetMinMaxVisual()**: ricalcola il minimo e massimo locali alla superficie della molecola, avviene a seguito del cambiamento del raggio degli atomi rappresentati.
- **Vector* GetMEPMaxPositivePosition()**:restituisce l'array contenente la posizione dei massimi assoluti di potenziale.
- **Vector* GetMEPMinNegativePosition()**:restituisce l'array contenente la posizione dei minimi assoluti di potenziale.
- **map¡int,double¿ GetMEPGraphOnDirection(Vector startingPoint, Vector endingPoint, double samplesDistance)**: calcola il grafico di potenziale sulla congiungente del cursore aptico all'atomo più vicino. I punti del grafico vengono calcolati campionando la congiungente ad un passo costante e definito e calcolando in quei punti il valore di potenziale, si ottiene così un insieme ordinato di punti che verranno mostrati a video sotto forma di grafico.

  INPUT

  – **Vector startingPoint**: posizione del punto di partenza del grafico, generalmente posizione del cursore aptico; da notare che tale posizione è al lordo della rototraslazione della molecola che dovrà quindi essere annullata per ottenere il valore di potenziale.

  - **Vector endingPoint**:posizione del punto final del grafico , corrispondente al nucleo dell'atomo più vicino, per coerenza anch'esso è espresso in coordinate assolute.
  - **double samplesDistance**: distanza di campionemento sulla linea dal punto iniziale e finale, valore imposto nel sistema ad un decimo di Angstrom.

  OUTPUT

  - **map¡int,double¿**: variabile map che identifica l'associazione tra un indice intero e un valore di potenziale double, il grafico si compone quindi scorrendo gli indici e segnando il valore di potenziale corrispondente.

- **Vector GetNearestAtomCoordinates(Vector point)**: restituisce le coordinate dell'atomo più vicino al vettore passato come parametro, generalmente la posizione del cursore aptico.
- **bool isSelected()**:restituisce se un atomo della molecola è stato selezionato o meno.
- **void setSelected(bool value)**: imposta la variabile che controlla la selezione al valore identificato dal parametro in ingresso.
- **void setSelectedAtom(unsigned int AtomIndex)**: imposta l'atomo i-esimo come atomo selezionato.
- **unsigned int getSelectedAtom()**: ritorna l'indice dell'atomo selezionato.
- **double calcLog(double x)**: calcola l'iperlogaritmo di un numero. Il grado dell'iperlogaritmo è definito dall'attributo **loglevel** della variabile di stato della molecola.
- **double calcExp(double x)**: calcola l'iperesponenziale di un numero, con il grado definito dallo stato della molecola.

5.8 Classe UserInterfaceEngine

La classe UserInterfaceEngine definisce e gestisce tutti gli elementi presenti nell'interfaccia grafica. Per ogni elemento, inoltre, definisce una funzione di callback.

Attributi e metodi privati

Attributi

- **Molecule* HapticMolecule**: puntatore alla molecola presente nel sistema, dallo stato della molecola si estraggono quasi tutte le informazioni salienti da inserire nell'interfaccia.
- **HapticEngine* HEngine**: puntatore all'engine aptico, utilizzato nella definizione delle opzioni riferite all'interfaccia aptica al sistema.
- **AudioEngine* AEngine**: puntatore all'engine audio, permette di associare effetti sonori all'interfaccia utente.
- **DynamicEngine* DEngine**: puntatore all'engine dinamico, permette di controllare lo stato della simulazione dinamica dall'interfaccia utente, nel caso, ad esempio, si passi dalla visualizzazione del MEP alla visualizzazione della geometria molecolare
- **int MainWindowId**: identificativo della finestra creata da GLUT che contiene l'intera interfaccia visiva dell'applicazione.
- **int RenderingAreaSize**:dimensione dell'area di rendering quadrata, impostata a 620 pixel.
- **bool showCustom**: indica se deve essere visualizzato il radio button custom nella lista delle visualizzazioni possibili.
- **bool showCustomCharge**:indica se deve essere visualizzato il radio button custom charge nella lista delle possibilità di calcolo di potenziale e forza.
- **GLUI_RadioButton* CustomRadioButton**: radio button che definisce l'opzione custom nelle possibili visualizzazioni.
- **GLUI_RadioButton* CustomChargeRadioButton**:radio button che definisce l'opzione custom charge nelle possibili forme di calcolo di MEP.
- **GLUI* GLUIRightSubwindow**: puntatore alla sottofinestra destra contenente le opzioni della molecola.
- **GLUI_Button* OpenMolecule3DFileControl**: puntatore al pulsante per il caricamento del file molecolare.
- **GLUI_Button* QuitControl**: puntatore al pulsante di quit.

- **GLUI_RadioGroup* VisualizationModeControl**: radio group che contiene le possibili visualizzazioni in merito al raggio degli atomi.
- **GLUI_RadioGroup* MepModeControl**: radio group per scegliere le modalità di calcolo del MEP.
- **GLUI_RadioGroup* InfoTypeControl**: radio group con le opzioni riferite alla tipologia di informazione mostrata dal sistema.
- **GLUI_Checkbox* LennardJonesControl**: controller dell'opzione correzione di Lennard Jones.
- **GLUI_Checkbox* ShowVectorControl**: controller dell'opzione mostra vettore.
- **GLUI_Checkbox* ShowGraphControl**: controller dell'opzione mostra grafico di potenziale.
- **GLUI_Checkbox* FixedControl**: controller dell'opzione fissa la molecola nello spazio[15].
- **GLUI_Checkbox* ShowLogControl**: controller dell'opzione utilizza scala logaritmica.
- **GLUI_Spinner* LogLevelControl**: controlla il grado della scala iperlogaritmica.
- **GLUI_Checkbox* ShowLinesControl**: controller dell'opzione mostra linee di forza.
- **GLUI_Checkbox* ShowLinesDirectionControl**:controller dell'opzione mostra la direzione delle linee di forza.
- **GLUI_Checkbox* ShowMinMaxControl**:controller dell'opzione mostra minimi e massimi.
- **GLUI_Checkbox* PerspectiveViewControl**: controller dell'opzione visualizzazione prospettiva.
- **GLUI_StaticText* AtomName**: testo sull'interfaccia contenente il nome dell'atomo.
- **GLUI_Spinner* AtomRadiusControl**: controlla il raggio dell'atomo selezionato.

[15] attiva o meno la simulazione dinamica

- **GLUI_Spinner* AtomChargeControl**: controlla la carica dell'atomo selezionato.
- **GLUI_Spinner* IsoMaxControl**: controlla l'estremo maggiore dell'intervallo dei punti isopotenziali.
- **GLUI_Spinner* IsoMinControl**: controlla l'estremo minore dell'intervallo dei punti isopotenziali.
- **GLUI* GLUIBottomSubwindow**: puntatore alla sottofinestra in basso contenente i controller spaziali della molecola.
- **GLUI_Rotation* MoleculeRotationControl**: controllore della rotazione della molecola.
- **GLUI_Translation* MoleculeXYTranslationControl**:controllore della traslazione della molecola sugli assi x, y.
- **GLUI_Translation* MoleculeZTranslationControl**: controllore della traslazione della molecola sull'asse z.
- **GLUI_Spinner* MoleculeScaleFactorControl**: controlla lo fattore di scala.
- **GLUI_Spinner* CursorChargeControl**: controlla la carica, o la forza, associata al cursore aptico.

Metodi

- **void GLUTInitialization(int* argcPointer, char** argv)**: crea e inizializza la finestra principale con GLUT, inoltre la ridimensiona. I parametri sono gli stessi passati al main.
- **void GLUIInitialization(void)**: crea effettivamente l'interfaccia grafica richiamando le due funzioni che creano le finestre: **CreateRightSubwindow** e **CreateBottomSubwindow**.
- **void CreateRightSubwindow(void)**: crea la sottofinestra laterale destra inserendo e inizializzando tutti gli elementi di interfaccia voluti.
- **void CreateBottomSubwindow(void)**: crea la sottofinestra sottostante la scena 3d inserendo e inizializzando tutti gli elementi di interfaccia voluti.

Metodi pubblici

La maggior parte dei callback della classe si occupa di riaggiornare i valori delle variabili di stato contenute nei vari engine.

- **UserInterfaceEngine(void)**:costruttore, inizializza tutte le variabili a NULL o ad un valore di default.
- **∼UserInterfaceEngine(void)**:distruttore di default.
- **bool Init(int* argcPointer, char** argv, Molecule* molecule, HapticEngine* hapticEngine, AudioEngine* audEngine, DynamicEngine* DynamicEngine)**: inizializza l'engine interfaccia utente associando i vari engine necessari. Inoltre richiama il metodo di creazione della finestra e il metodo di creazione dell'interfaccia grafica.

  INPUT

  - **int* argcPointer**: arriva dal main e viene passato alla funzione creatrice della finestra con GLUT.
  - **char** argv**: arriva dal main e viene passato alla funzione creatrice della finestra con GLUT.
  - **Molecule* molecule**: passa il riferimento alla molecola del sistema.
  - **HapticEngine* hapticEngine**: passa il riferimento all'engine aptico.
  - **AudioEngine* audEngine**: passa il riferimento all'engine sonoro.
  - **DynamicEngine* DynamicEngine**: passa il riferimento all'engine dinamico.

  OUTPUT

  - **bool**: true se tutto è andato a buon fine, false se ci sono stati degli errori.
- **void UpdateUIControls()**: aggiorna i controller sull'interfaccia in base ai valori contenuti negli engine e nello stato della molecola. Viene chiamata a seguito di cambiamenti esterni all'interfaccia.
- **void MoleculeTranslationCallback(void)**: callback del pulsante di traslazione.
- **void MoleculeScaleFactorCallback(void)**: callback del controller del fattore di scala.

- **void CursorChargeCallback(void)**: callback del controller della carica del cursore aptico.
- **void OpenMolecule3DFileCallback(void)**: callback del pulsante apertura file. Apre la finestra di selezione file.
- **void QuitCallback(void)**: callback del pulsante quit, termina il programma.
- **void VisualizationModeCallback(void)**: callback del selettore della modalità di visualizzazione.
- **void MEPModeCallback(void)**: callback del selettore della modalità di calcolo del MEP.
- **void InfoTypeModeCallback(void)**: callback del selettore della tipologia di informazione visualizzata.
- **void LennardJonesCallback(void)**: callback del controller per la correzione di Lennard Jones.
- **void ShowVectorCallback(void)**:callback del controller per visualizzazione del vettore.
- **void ShowGraphCallback(void)**: callback del controller per visualizzazione del grafico di potenziale.
- **void FixedCallback(void)**: callback del controller per fissare la molecola.
- **void ShowLogCallback(void)**: callback del controller per utilizzare le scale logaritmiche.
- **void ShowLinesCallback(void)**: callback del controller per mostrare le linee di forza.
- **void ShowLinesDirectionCallback(void)**: callback del controller per mostrare la direzione delle linee di forza.
- **void LogLevelCallback(void)**: callback del controller per il grado dlele scale iperlogaritmiche.
- **void PerspectiveViewCallback(void)**:callback del controller per la visualizzazione prospettica.

- **void ShowMinMaxCallback(void)**: callback del controller per la visualizzazione dei minimi e dei massimi.
- **void RotationCallback(void)**: callback del pulsante di rotazione.
- **void AtomRadiusCallback(void)**: callback del controller del raggio atomico.
- **void AtomChargeCallback(void)**: callback del controller della carica atomica.
- **void IsoMaxCallback(void)**: callback del controller dell'estremo superiore dell'intervallo isopotenziale.
- **void IsoMinCallback(void)**: callback del controller dell'estremo inferiore dell'intervallo isopotenziale.
- **void IdleUICallback(void)**: funzione di idle, ridisegna continuamente l'interfaccia grafica.
- **void ReshapeUICallback(int width, int height)**: funzione di callback per il ridimensionamento della finestra. Definisce la viewport della finestra grafica e la proiezione, ortogonale o prospettica, della scena 3d attraverso l'uso della libreria OpenGL.

  INPUT

  - **int width**: nuova larghezza della finestra in pixel.
  - **int height**: nuova altezza della finestra.
- **int getRenderingAreaSize()**: restituisce la grandezza dell'area di rendering.
- **void enableAtom()**: attiva la visualizzazione delle opzioni relative all'atomo selezionato.
- **void disableAtom()**: disattiva le opzioni relative alla selezione dell'atomo.

Funzione esterna

- **void UIControlsCallbackHandler(int controlId)**:definisce il callback generale per tutti i controlli presenti nell'interfaccia. Tramite uno switch sceglie la funzione di callback da richiamare a seconda del valore del parametro in ingresso **controlId**.

  INPUT

– **int controlId**: definisce la funzione di callback da richiamare, ad ogni controller è associata una define indicante il proprio id.

5.9 Utilities

Utilities non è una classe, bensì un contenitore di funzioni utili al sistema.

- **string OpenFileDialog(OpenFileDialogMode openFileDialogMode**: crea e gestisce la finestra di selezione file.

  INPUT

  – **OpenFileDialogMode openFileDialogMode**: identifica il tipo di file da selezionare. Può essere MOLECULE_SHAPE_OPEN_MODE, he indica l'apertura di un file pdb o MEP_OPEN_MODE che indica l'apertura di un file mep. Nel sistema attuale viene usata solo la modalità MOLECULE_SHAPE_OPEN_MODE.

  OUTPUT

  – **string**: stringa contenente il path del file selezionato .

- **string ConvertLPWSTRToString(LPWSTR lpwstr**: funzione che converte una stringa dal formato LPWSTR[16],in input, a string,in output.

- **string ChangeFileExt(string fileName, string ext**: cambia l'estensione di un file in una stringa contenente il path del file stesso.

  INPUT

  – **string fileName**: stringa contenente il path di un file, terminata dall'estensione dello stesso.

  – **string ext**: nuova estensione che si vuole sostituire al path.

  OUTPUT

  – **string**: stringa contenente il path con l'estensione del file modificata.

- **wstring s2ws(const string& s)**: converte una stringa dal formato string al formato wstring.

[16]Long Pointer to Wide String

5.10 Classe Vector

L'ultima classe,la classe Vector, permette di manipolare i vettori presenti all'interno del sistema.

Attributi privati

- **double X**:prima componente del vettore, contiene la coordinata x.
- **double Y**:seconda componente del vettore, contiene la coordinata y.
- **double Z**:terza e ultima componente del vettore, contiene la coordinata z.

Metodi pubblici

- **Vector(void)**: costuttore, inizializza il vettore a $(0.0, 0.0, 0.0)$.
- ∼**Vector(void)**:distruttore, non fa nulla.
- **double GetX(void)**: restituisce la coordinata x.
- **double GetY(void)**: restituisce la coordinata y.
- **double GetZ(void)**: restituisce la coordinata z.
- **void Get(double &x, double &y, double &z)**: associa ai parametri reference in ingresso il valore delle coordinate del vettore.
- **void SetX(double x)**: imposta la coordinata x al valore passato come parametro.
- **void SetY(double y)**: imposta la coordinata y.
- **void SetZ(double z)**: imposta la coordinata z.
- **void Set(double x, double y, double z)**: imposta le 3 dimensioni del vettore.
- **double GetLength(void)**: restituisce la lunghezza, o norma, del vettore calcolata come $\sqrt{x^2 + y^2 + z^2}$.
- **Vector GetVersor(void)**: restituisce un versore con la stessa direzione del vettore originario, ma lunghezza, norma, unitaria.

- **Vector WeightedSum(Vector vec2, double ind)**: fa la somma pesata con un altro vettore, il vettore risultante sarà: $\vec{V_3} = \alpha\vec{V_2} + (1 - \alpha)\vec{V_1}$. Viene usato per il calcolo dell'interpolazione 3d di gradiente e gradiente secondo.

  INPUT

  - **Vector vec2**: vettore V_2 da sommare al vettore corrente.
  - **double ind**: valore α peso della somma di vettori

  OUTPUT

  - **Vector**: un nuovo vettore ottenuto dalla somma pesata degli altri due..

- **void Translate(Vector translation)**: trasla un vettore facendo la somma componente per componente rispetto al vettore passato come parametro.

- **void XTranslate(double xIncrement)**: trasla un vettore rispetto ad una componente sull'asse x.

- **void YTranslate(double yIncrement)**: trasla un vettore rispetto ad una componente sull'asse y.

- **void ZTranslate(double zIncrement)**: trasla un vettore rispetto ad una componente sull'asse z.

- **void Scale(double scaleFactor)**: moltiplica il vettore per una quantità pari allo **scaleFactor** passato come parametro, il vettore viene così scalato.

- **Vector GetInverse(void)**: restituisce il vettore inverso, invertendo il segno di ognuna delle componenti

- **void ApplyMatrix(float RotationMatrix[16])**: applica una matrice al vettore, moltiplicando lo stesso per la matrice: $\vec{V} = A\vec{V}$.

- **void ApplyInverseRotationMatrix(float RotationMatrix[16])**: applica al vettore la matrice inversa della matrice passata come parametro. Da notare che la matrice deve essere di rotazione e quindi ortogonale[17].

- **Vector GetDistance(Vector point)**: restituisce un vettore che identifica la distanza tra il vettore attuale ed un punto.

[17] la matrice inversa è qui infatti calcolata solo come matrice trasposta, e questo e valido solo per matrici ortogonali

Funzioni esterne

Infine vi sono all'esterno della classe delle funzioni per manipolare le matrici direttamente.

- **void XRotateMatrix(double degrees,float RotationMatrix[16])**: applica una rotazione rispetto all'asse x ad una matrice.

  INPUT

  - **double degrees**: gradi della rotazione rispetto all'asse x.
  - **float RotationMatrix[16]**: matrice che identifica una roto-traslazione da ruotare.

- **void YRotateMatrix(double degrees,float RotationMatrix[16])**: come la funzione sovrastante solamente che la rotazione è rispetto all'asse y.

- **void ZRotateMatrix(double degrees,float RotationMatrix[16])**: ruota una matrice rispetto all'asse z.

- **void MatrixXMatrix(float a[16],const float b[16])**: moltiplica le due matrici passate come parametro e inserisce il risultato nella prima matrice.

- **void TrasposeMatrix(float a[16])**: trasforma una matrice nella sua trasposta.

- **void fromQtoM(double q[4],float m[16])**: trasforma il quaternione q in una matrice di rotazione, dati i quattro elementi q_1, q_2, q_3 e q_4 la matrice di rotazione si ottiene nel seguente modo.

$$\begin{pmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_3q_4) & 2(q_1q_3+q_2q_4) & 0 \\ 2(q_1q_2+q_3q_4) & 1-2(q_1^2+q_3^2) & 2(q_2q_3-q_1q_4) & 0 \\ 2(q_1q_3-q_2q_4) & 2(q_2q_3+q_1q_4) & 1-2(q_1^2+q_2^2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

5.11 HapticMol

In HapticMol è contenuto il main del programma, che inizializza la classe HapticApplication e registra le funzioni di callback legate agli input da mouse e tastiera. Le funzioni presenti sono:

- **void glDisplay()**: funzione di callback per il disegno della scena 3d.

- **void uiReshape(int width, int height)**:funzione di callback per il ridimensionamento della finestra
- **void uiKeyboard(unsigned char key, int x, int y)**: funzione di callback per gli input da tastiera
- **void uiIdle()**: funzione di callback idle per il disegno dell'interfaccia grafica.
- **void uiMouse(int button, int state, int x, int y)**: funzione di callback per la pressione dei tasti del mouse.
- **void glPMotion(int x, int y)**: funzione di callback per i movimenti passivi[18] del mouse.
- **void glMotion(int x, int y)**: funzione di allback per i movimenti attivi[19] del mouse.
- **void hdExit()**: funzione di callback richiamata all'uscita dal programma.
- **int _tmain(int argc, _TCHAR* argv[])**: funzione main, inizializza la variabile Application di tipo HapticApplication e registra tutte le funzioni di callback, dopodichè si mette in loop infinito sul mainloop dell'oggetto Application.

5.12 Conclusioni del capitolo

In questo capitolo sono state illustrate le classi che compongono il progetto e tutti i metodi e gli attributi contenuti al loro interno.
Si è visto come il codice è stato suddiviso ordinatamente per engine, ognuno dei quali afferente ad una diversa componente del sistema. Ciò garantisce un elevato grado di modularizzazione e la possibilità di modificare uno degli engine senza disturbare gli altri.
Nel prossimo capitolo verranno illustrate le prove sperimentali svolte e i pareri degli esperti che hanno visionato il sistema. Verranno inoltre introdotte idee utili alla valutazione del sistema da un punto di vista didattico e da un punto di vista di usabilità. Inoltre verranno illustrati una serie di esperimenti atti a giudicare l'interazione aptica con il sistema.

[18] cioè senza la pressione di un tasto del mouse
[19] cioè con il tasto del mouse premuto

Capitolo 6

Realizzazioni sperimentali e valutazione

DURANTE IL CORSO DEL PRESENTE CAPITOLO vengono descritte le prove sperimentali svolte, gli utilizzi del software dal punto di vista didattico e possibili prove di valutazione da effettuare sul sistema.
Viene focalizzata, infatti, l'attenzione sulle molecole utilizzate durante lo sviluppo del sistema, valutandone la complessità. A seguito vengono analizzate le possibili esperienze didattiche che facciano uso del sistema da sottoporre agli studenti. Si passerà poi ad esaminare i giudizi degli esperti in merito all'applicativo prodotto. Infine si procederà ad illustrare opportuni metodi di valutazione del sistema.

6.1 Molecole Utilizzate

Nel corso dello sviluppo del sistema sono state utilizzate diverse tipologie di molecole di varia complessità. Nello specifico:

- Acqua(H_20)
- Ammoniaca(NH_3)
- Trifluoroiodometano(CF_3I)
- Benzene(C_6H_6)
- Cubano(C_8H_8)
- Variante del borano($C_{14}H_{29}B$)
- Cortisolo($C_{21}H_{30}O_5$)

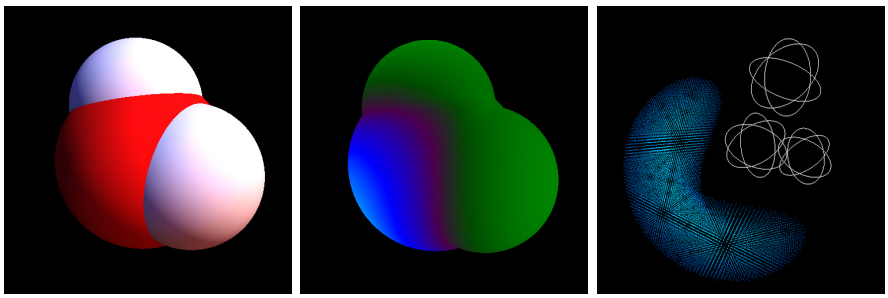

Figura 6.1: Acqua H_2O

- Fullerene(C_{60})
- Filamento di DNA

Di cui del filamento di DNA è stata data solo la rappresentazione geometrica.
Nel seguito sarà esaminata la natura delle varie molecole presenti, la complessità e la visualizzazione del MEP.

6.1.1 Acqua H_2O

L'acqua è la molecola più semplice utilizzata nel sistema, caratterizzata da tre soli atomi, figura 6.1. Risulta un elemento particolarmente interessante per la sua polarità.
Dato che l'ossigeno ha una elettronegatività maggiore, il vertice della molecola ospita una parziale carica elettrica negativa ($\delta-$), mentre le estremità recano una parziale carica elettrica positiva ($\delta+$). Una molecola che presenta questo squilibrio di cariche elettriche è detta essere un dipolo elettrico.
Le cariche fanno sì che molecole d'acqua vengano attratte reciprocamente l'una dall'altra. Questa attrazione nell'acqua è particolarmente intensa (anche se è più debole dei legami covalenti interni alla molecola stessa) e prende il nome di legame idrogeno (o H-bond) e spiega molte delle proprietà fisiche tipiche dell'acqua.
Queste caratteristiche sono facilmente riscontrabili nel sistema, in quanto è possibile percepire sia l'elettronegatività dei vari atomi[1], sia la buca di potenziale nelle prossimità dell'atomo di ossigeno, individuando facilmente il punto esatto dove si forma il legame idrogeno.
Da notare che la forma della buca di potenziale non individua un unico punto, ma un'area allungata, adatta a formare legami a idrogeno potenzialmente con altre due molecole.

[1] sotto forma di attrito sulla geometria molecolare

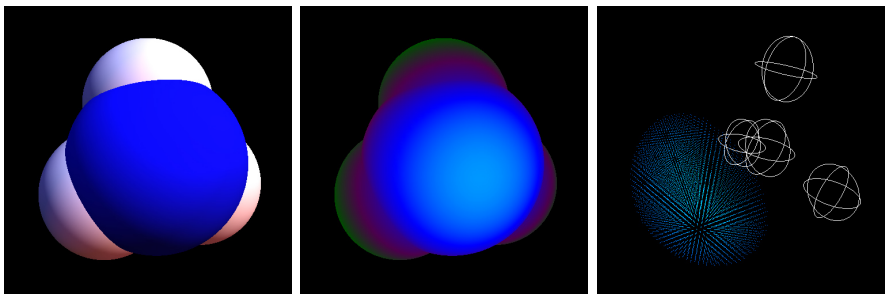

Figura 6.2: Ammoniaca NH_3

6.1.2 Ammoniaca NH_3

L'ammoniaca è un composto dell'azoto di formula chimica NH_3, figura 6.2. Si presenta come un gas incolore, tossico, dall'odore caratteristico. Come l'acqua è estremamente semplice ed è composto da soli 4 atomi.
La forma della molecola dell' ammoniaca anidra è tetraedrica deformata; l'atomo di azoto occupa la posizione centrale e lega i tre atomi di idrogeno. La base è un triangolo equilatero occupato dai tre atomi di idrogeno mentre il quarto vertice del tetraedro è occupato da una coppia elettronica di non legame (lone pair), che è la principale responsabile di tutte le proprietà della molecola (formazione di legami a idrogeno, basicità secondo Lewis e secondo Bronsted-Lowry, elevata permittività elettrica e momento dipolare, elevata solubilità in acqua).
Questa molecola è importante nel sistema proprio per la sua struttura tetraedrica, è facile infatti notare che la buca di potenziale 'lone pair' è posta al quarto vertice del tetraedro e presenta caratteristiche simili all'acqua vista in precedenza. L'ammoniaca è infatti molto solubile in acqua perché è in grado di formare facilmente legami a idrogeno.

6.1.3 Trifluoroiodometano CF_3I

Il trifluoroiodometano CF_3I è una molecola usata spesso negli estintori come soppressore gassoso del fuoco, ha una struttura piuttosto semplice composta da soli 5 atomi.
Presenta tre punti di minimo in corrispondenza dei 3 atomi di fluoro. Risulta particolarmente interessante per le particolari forme che assumono le sue superfici isopotenziali.

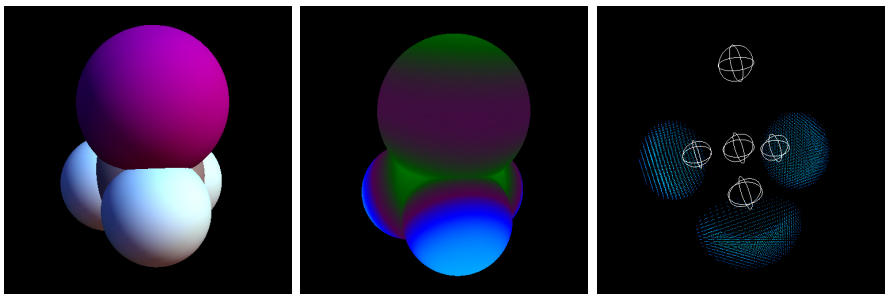

Figura 6.3: Trifluoroiodometano CF_3I

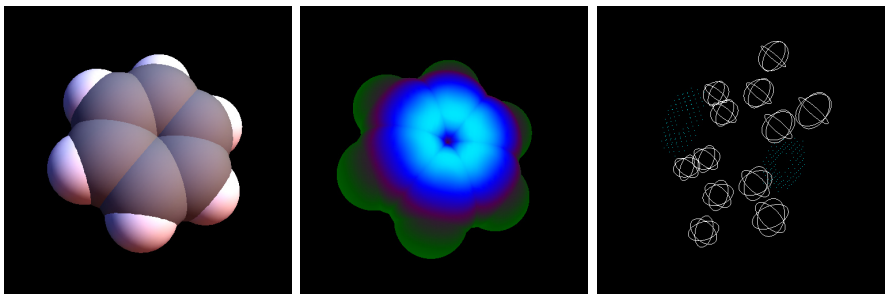

Figura 6.4: Benzene C_6H_6

6.1.4 Benzene C_6H_6

Il benzene è un composto chimico che a temperatura ambiente e pressione atmosferica si presenta sotto forma di liquido volatile incolore altamente infiammabile, dall'odore caratteristico. Dal punto di vista chimico, il benzene è un idrocarburo aromatico monociclico avente formula bruta C_6H_6, figura 6.4. Il benzene è un costituente naturale del petrolio, ma viene sintetizzato a partire da altri composti chimici presenti nel petrolio stesso.
La sua molecola è planare, i sei atomi di carbonio sono disposti ai vertici di una struttura a esagono regolare; ad ognuno di essi è legato un atomo di idrogeno. Ogni atomo di carbonio condivide con gli altri un elettrone spaiato perpendicolare al piano della molecola.
Per poter meglio rappresentare la natura delocalizzata del legame, spesso l'anello benzenico viene rappresentato da un esagono (ogni vertice è un atomo di carbonio, gli idrogeni sono omessi) con all'interno un cerchio.
Questa nuvola elettronica condivisa e ciclica interna alla molecola è facilmente visualizzabile nel sistema e rappresenta la buca di potenziale. Inoltre l'interessante forma geometrica della molecola la rende molto utile a scopi didattici.

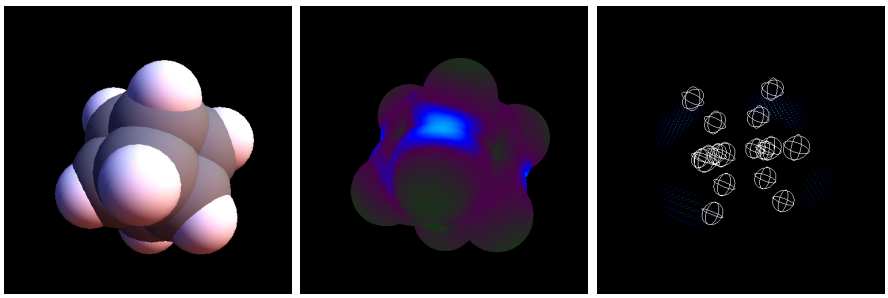

Figura 6.5: Cubano C_8H_8

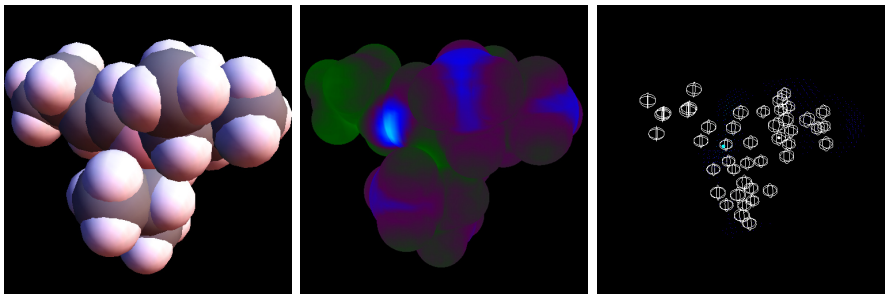

Figura 6.6: dibutyl-(1-methylenepentyl)borane $C_{14}H_{29}B$

6.1.5 Cubano C_8H_8

Il cubano è un idrocarburo policiclico alifatico avente formula C_8H_8, figura 6.5. Deve il suo nome al fatto che gli 8 atomi di carbonio sono disposti ai vertici di un cubo. A temperatura e pressione ambiente è un solido cristallino.
Il cubano ed i suoi derivati, causa dell'elevata energia dei legami in tensione, trovano applicazione come additivo per carburanti ed esplosivi ad elevata densità.
La particolare struttura atomica, composta da 18 atomi, è particolarmente interessante nel sistema qui svviluppato proprio per la sua forma geometrica.

6.1.6 Variante del borano $C_{14}H_{29}B$

Questa molecola, particolarmente complessa, è composta da 44 atomi ed è stata utilizzata nel sistema per testare le capacità del motore grafico con strutture molecolari complesse, figura 6.6.

6.1.7 Cortisolo $C_{21}H_{30}O_5$

Il cortisolo è un ormone prodotto dalle ghiandole surrenali dalla struttura molecolare piuttosto complessa, composto da 56 atomi. Come il precedente

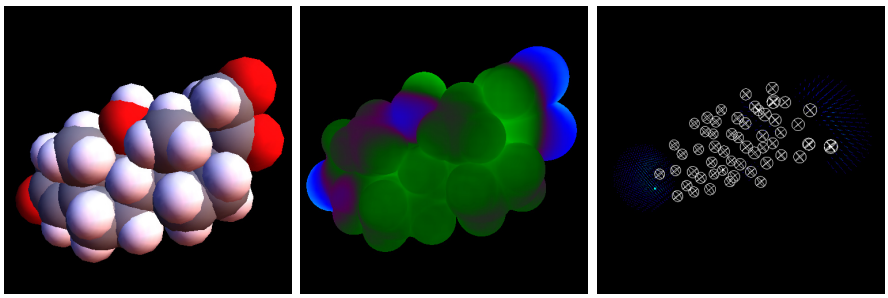

Figura 6.7: cortisolo $C_{21}H_{30}O_5$

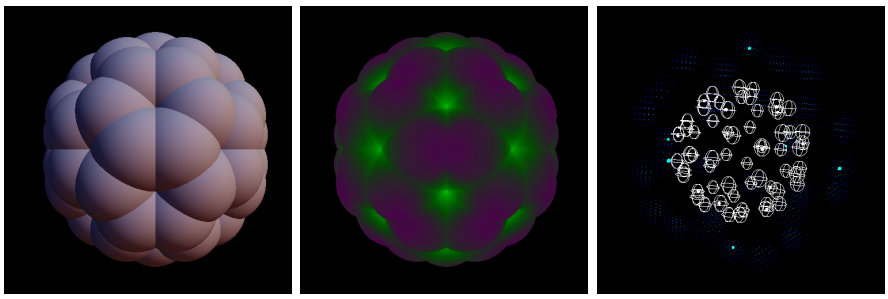

Figura 6.8: fullerene C_{60}

viene utilizzato per testare il sistema con molecole complesse, inoltre, in corrispondenza di un ossigeno all'estremo della molecola presenta una buca di potenziale dalla forma simile a quella trovata nell'acqua.

6.1.8 Fullerene C_{60}

Le molecole di fullerene, composte interamente di carbonio, assumono una forma simile a una sfera cava e sono generalmente stabili a temperatura e pressione ambiente, nonostante siano energeticamente sfavorite rispetto ad altri allotropi del carbonio, quali la grafite e il diamante, figura 6.8.
Il più piccolo fullerene in cui nessuna coppia di pentagoni condivide un bordo è C_60 (fullerite) e in quanto tale risulta essere inoltre il più diffuso. La struttura del C_60 è quella di un icosaedro troncato, che assomiglia a un pallone da calcio, costituito da esagoni e pentagoni, ai cui vertici si posiziona ciascun atomo di carbonio e i cui bordi rappresentano i legami.
L'introduzione della molecola nel sistema deriva dalla sua particolare struttura molecolare.

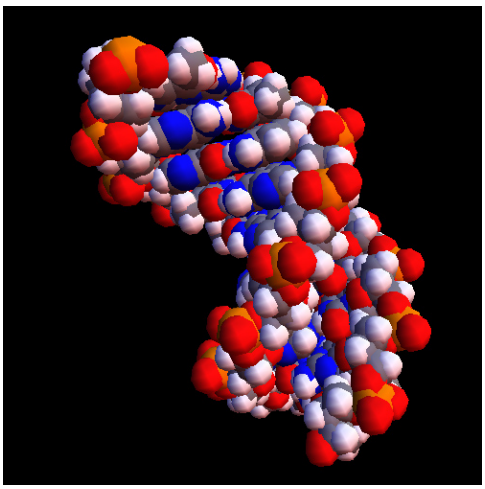

Figura 6.9: frammento filamento di DNA

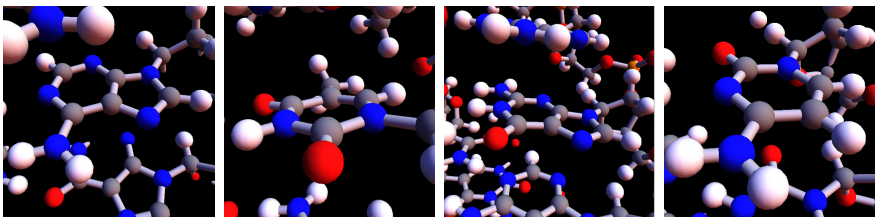

Figura 6.10: ATGC (Adenina, Timina, Guanina, Citosina)

6.1.9 DNA

L'acido desossiribonucleico o deossiribonucleico (DNA) è un acido nucleico che contiene le informazioni genetiche necessarie alla biosintesi di RNA e proteine, molecole indispensabili per lo sviluppo ed il corretto funzionamento della maggior parte degli organismi viventi.
Dal punto di vista chimico, il DNA è un polimero organico costituito da monomeri chiamati nucleotidi. Quattro sono le basi azotate che possono essere utilizzate nella formazione dei nucleotidi da incorporare nella molecola di DNA: adenina, guanina, citosina e timina, figura 6.10.

La disposizione in sequenza di queste quattro basi costituisce l'informazione genetica, leggibile attraverso il codice genetico, che ne permette la traduzione in amminoacidi.
Risulta probabilmente la molecola complessa più famosa tra quelle presenti nel nostro organismo ed è conosciuta da tutti per la sua particolare importanza nella genesi dell'essere umano e vivente in generale. Con i suoi 637 atomi rappresenta la molecola più complessa che il sistema si è trovato ad affrontare. Nonostante la complessità il sistema è in grado di renderiz-

zare l'intera molecola senza scendere eccessivamente di dettaglio, notevoli i dettagli delle sue 4 componenti in figura 6.10.

6.2 Esperienze didattiche

Come già si è accennato nella precedente trattazione, l'applicazione qui descritta nasce come strumento didattico e a tal fine è stato strutturato l'intervento dello studente. Grazie al software sviluppato gli studenti possono, infatti, esplorare la superficie elettrostatica di varie molecole contenute nel repository: percepire apticamente il loro campo elettrico e la posizione in cui si trovano le buche di potenziale, fondamentali per la creazione di legami. Non solo, possono anche familiarizzare con la struttura geometrica e comprendere le relazioni tra potenziale e elettronegatività degli atomi.
L'aver sintetizzato le griglie di potenziale con un software impiegato nella ricerca chimica implica che l'intero lavoro si basi su dati che simulano il comportamento delle interazioni nel modo più realistico possibile, ovvero con la massima precisione nella descrizione dei fenomeni. Questo approccio è ben diverso dal considerare l'interazione tra una carica e una molecola basandosi su modelli prettamente teorici, come fatto ad esempio da moltissimi applicativi che si possono facilmente trovare anche sul Web. L'affidabilità e la coerenza nella riproduzione dei fenomeni simulati hanno avuto perciò fin dall'inizio la massima attenzione.
Dopo aver incontrato gli esperti di diversi settori (chimica, design), si sta progettando la fase di inserimento dello strumento come attività integrativa di laboratorio in associazione ad un corso di chimica curriculare. Tale attività sarà svolta da studenti, iscritti ai primi anni di un corso di ingegneria comprendente l'esame di Chimica generale, che volontariamente decideranno di parteciparvi. Il laboratorio consisterà in sessioni assistite da un docente che spiegherà passo-passo ed in maniera estremamente guidata come utilizzare lo strumento per testare i comportamenti di interazione inter-molecolare spiegati in teoria.
I feedback degli utenti verranno raccolti attraverso questionari somministrati alla fine di ciascuna esperienza. In tali questionari si richiederà agli utenti:

1. la facilità / difficoltà con cui è stato per loro possibile ottenere il risultato richiesto attraverso l'uso dello strumento, per avere in tale modo un'idea della sua usabilità;

2. se ritengono di aver compreso con maggior dettaglio il fenomeno analizzato attraverso l'attività integrativa: in tale caso, si chiederà di fornire

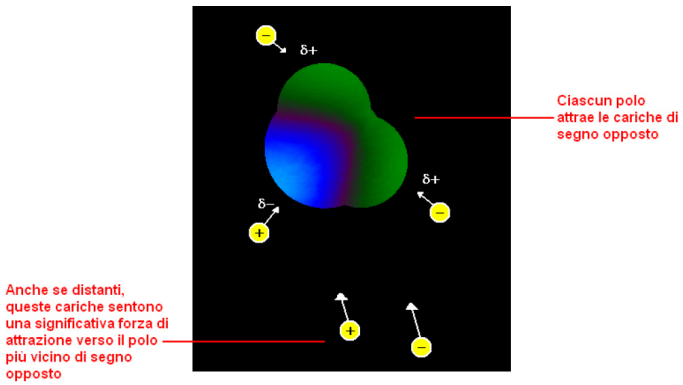

Figura 6.11: attività didattica, studio polarità

uno o più commenti testuali che spieghino come la conoscenza del fenomeno interessato sia mutata rispetto a prima dell'esperienza[2] e quali siano le caratteristiche dello strumento che abbiano permesso di migliorare le nozioni già acquisite.

Gli argomenti affrontati durante tali attività sono già stati individuati e riguarderanno lo studio di caratteristiche specifiche delle molecole, per le quali lo strumento mette bene in evidenza il comportamento dei fenomeni.

La prima attività, figura 6.11, consiste nello studio della polarità delle molecole, ovvero nell'analisi delle differenti modalità di attrazione di aree diverse di una molecola polare (es. acqua) in base al segno della carica elettrica (positiva o negativa). Lo strumento aptico permette di far sentire all'utente le diverse forze (di attrazione o repulsione) in base alla posizione e alla distanza della carica rispetto alla molecola. Alcune molecole infatti diventano dipoli (detti istantanei) solo se la distanza tra molecola e carica è sufficientemente ravvicinata.

Una seconda attività, figura 6.12, consiste nello sperimentare l'anisotropia dell'interazione, ovvero come una stessa carica elettrica possa risentire di forze di diversa intensità in base alla direzione di avvicinamento alla molecola. In relazione a questa attività è possibile determinare un'area di influenza della molecola, in base alla quale si può capire come l'effetto della molecola sulla carica sia dipendente dalla distanza relativa. Lo strumento aptico è qui rilevante per far sentire le diverse intensità di forza (trascurabile, lieve, forte) in base alla distanza e alla direzione di avvicinamento. Un'ulteriore

[2] nuove caratteristiche del fenomeno, nuovi dettagli della sua dinamica e azione, ecc...

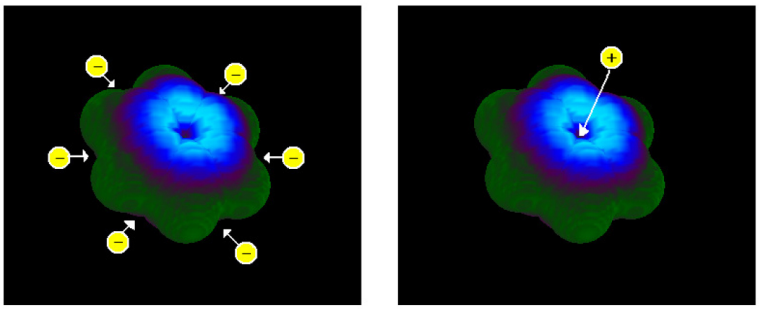

Figura 6.12: attività didattica, anisotropia dell'interazione

esperienza consiste nel determinare i punti critici del campo elettrico attorno alla molecola. La localizzazione di tali punti nello spazio permette di comprendere il concetto di distanza di legame e l'uso dello strumento aptico aiuta a capire come le forze in gioco in tali punti conferiscano stabilità alla posizione della carica. La visualizzazione del grafico del potenziale permette inoltre di verificare come tali punti coincidano con i minimi della funzione rappresentata. La carica positiva della parte destra della figura 6.11, ad esempio, trova un minimo nel centro della molecola di benzene, che funge in tal modo da tasca per le cariche negative.
Dagli esempi forniti precedentemente risultano particolarmente chiare le potenzialità didattiche dello strumento, per cui la creazione di esperienze dipende soltanto dalla fantasia del docente utilizzatore. Attraverso le numerose funzionlità introdotte è possibile analizzare vari aspetti delle molecole introdotte; inoltre la possibilità di generare attraverso tool esterni nuovi file da aggiungere al repository, permette di personalizzare l'esperienza con le molecule ritenute di particolare interesse dal docente stesso.

6.3 Prove di esperti del settore

Lo strumento è stato testato da esperti di diversi settori su molecole differenti e di crescente complessità in termini di numero di atomi e di legami coinvolti. Test sono stati fatti, ad esempio, con acqua, ammoniaca, metano, benzene. Il campione di utenti considerato era formato da docenti, ricercatori ed esperti di diversi settori (chimica, design), il che ci ha permesso di avere da ciascuno dei feedback su diversi aspetti del tool[3]. Essendo lo stesso strumento sviluppato anche con finalità di ricerca, alcuni ricercatori hanno

[3]usabilità dai designer, aderenza ai comportamenti reali delle molecole dai chimici, ecc. . .

provato lo strumento, lo hanno apprezzato e ritenuto un valido aiuto per migliorare la consapevolezza dei fenomeni da esso modellati, altrimenti di difficile comprensione a livello intuitivo. I ricercatori, in particolare, hanno provato a verificare come lo strumento sia in grado di riprodurre caratteristiche specifiche di alcune molecole, ben note agli addetti ai lavori. Questo test è stato prezioso, in quanto ha permesso di apprezzare la bontà e la correttezza della simulazione.
Dai responsi ottenuti si può concludere che il punto di forza di uno strumento come questo è la possibilità di combinare la visualizzazione di dati, solitamente disponibili solo in formato numerico, con la sensazione aptica della forza delle interazioni da essi rappresentate. Studenti e ricercatori/docenti hanno accolto positivamente la novità di un tale approccio in affiancamento alla regolare attività ex cathedra.
Tante caratteristiche del sistema sono state modellate sulla base di riscontri ottenuti dai vari esperti di settore. Dai designer è emerso:

- la necessità di modificare la visualizzazione del cursore aptico da pallina a mirino nell'ambito della percezione del potenziale molecolare;
- l'introduzione di una scala che definisse le grandezze in gioco;
- la rappresentazione delle linee di forza, considerata una modalità di rappresentazione standard del force field molecolare;
- la modalità di visualizzazione prospettica della scena 3d;
- la definizione delle ombreggiature sulla superficie geometrica delle molecole, prima illuminate solo da luce ambientale e quindi di apparenza particolarmente 'piatta';
- lo spostamento del vettore forza associato al cursore aptico ad una zona fissa dell'interfaccia grafica;
- ...

Mentre dai chimici sono emerse, oltre alle esperienze didattiche possibili:

- la necessità di modificare il grafico di potenziale evidenziando il punto di minimo e il punto che identifica la posizione del cursore aptico;
- l'impossibilità della carica positiva ad attraversare il nucleo di un'atomo, e quindi l'inutilità e l'inappropriatezza della rappresentazione nel grafico dei punti seguenti al nucleo dell'atomo;

- la possibilità di introdurre nuove molecole nel repository quali: BeF_2, BC_{l3}, SO_2, O_3, CH_4, PF_5, SF_4, ICl_3, XeF_2, SF_6, IF_5 e Xe_F4, che consentirebbero di studiare gli orbitali ibridi e la geometria molecolare; e etano, etilene, acetilene, etanolo, acetaldeide, acido acetico, acetato d'etile e acetammide per poter anche insegnare un minimo di chimica organica.

Si può dunque comprendere come la fase di sviluppo sia stata seguita e guidata attraverso continui incontri con possibili utilizzatori del sistema, i chimici, esperti nel settore delle tecnologie per la didattica del Metid, che hanno proposto e guidato le modalità di interazione sonora, e esperti di usabilità, per valutare il sistema a priori della rappresentazione chimica.

6.4 Valutazione del sistema

Punto critico che richiede un'accurata analisi sono le mosalità di valutazione del sistema introdotto. Questa valutazione deve essere affrontata sotto vari punti e aspetti paralleli:

- **correttezza dell'informazione proposta**: di pertinenza di esperti in chimica che possano validare o meno il contenuto del sistema e l'informazione veicolata;
- **efficacia didattica dello strumento**: punto particolarmente critico per sua natura, va strutturato con cautela per poter essere identificato oome elemento oggettivo;
- **usabilità dell'interfaccia del sistema**: probabilmente l'aspetto più semplice, in quanto approfonditamente studiato in letteratura;
- **efficacia dell'interazione aptica**: altro punto ostico, si tratta di definire e valutare la qualità delle sensazioni aptiche prodotte e la naturalezza con cui l'utente riesce ad approcciarsi, se, cioé, le metafore introdotte per la percezione dei vari aspetti del sistema siano appropriate a descrivere il fenomeno presentato o meno;
- **efficacia dello stimolo sonoro come accompagnamento dell'esperienza aptica**: consiste nel valutare se la presenza dell'audio sia effettivamente di supporto o meno all'esplorazione aptica della molecola.

Si evince che non è compito facile, valutare il sistema. Del primo aspetto si è già discusso in precedenza e consiste prevalentemente nella valutazione

del sistema da parte di esperti che possano esaminare i vari concetti chimici proposti e la loro fondatezza.

6.4.1 Efficacia didattica attraverso un metodo statistico

Riguardo la valutazione dell'efficacia didattica si è pensato di ottenere dati oggettivi tramite prove di valutazione concepite per l'occasione. Si ritiene, infatti, che questionari proposti direttamente agli studenti non permetterebbero di ottenere dati oggettivi, in quanto si baserebbero su giudizi personali e su impressioni, lo studente utente potrebbe aver frainteso l'informazione veicolata dal sistema e la sua credenza di aver compreso l'argomento potrebbe essere fallace.
Sotto queste basi si è ipotizzato un esperimento per misuare l'effettiva efficacia didattica sulla base di dati statistici: si tratta di valutare i risultati di una prova d'esame sugli argomenti inerenti agli aspetti chimici presentati dal sistema. A questo punto si sottoporrebbe il test ad un campione di due tipologie di studenti distinti:

1. studenti che effettuano il test senza aver utilizzato lo strumento, ma avendo studiato gli argomenti attraverso metodi tradizionali;

2. studenti che effettuano il test a posteriori dell'utilizzo del software.

I voti assegnati a suddette prove di valutazione costituiranno rispettivamente due variabili aleatorie X e Y presumibilmente gaussiane di varianza σ^2 uguale, ma incognita. L'ipotesi sperimentale diventa quindi:

- $H0$ la media dei voti degli studenti X è uguale alla media degli studenti Y, cioè $H0 : Y - X = 0$, contro

- $H1$ la media dei voti degli studenti Y è maggiore della media degli studenti X, $H1 : Y - X > 0$.

Un test di questo tipo prende il nome di *t-test* e si risolve partendo dalla considerazione:

$$\frac{\overline{Y} - \overline{X}}{\sqrt{S_p^2(\frac{1}{m} + \frac{1}{n})}} \sim t_{m+n-2} \tag{6.1}$$

dove $\overline{Y}$ e $\overline{X}$ sono le medie campionarie delle due variabili aleatorie, S_p^2 è la varianza pooled, m e n sono le numerosità dei due campioni e t_{m+n-2} è la distrubuzione t di Student di grado $m + n + 2$. Rifiutiamo quindi $H0$ e

riteniamo valida $H1$ a livello di significatività α se:

$$\overline{Y} - \overline{X} \geq t_{m+n-2}(1-\alpha)\sqrt{S_p^2(\frac{1}{m}+\frac{1}{n})} \tag{6.2}$$

Da un esperimento del genere otterremmo quindi un dato oggettivo importante sulla bontà del sistema da un punto di vista didattico. Il rifiutare l'ipotesi nulla a favore di $H1$ indicherebbe un'evidenza sperimentale a favore dell'ipotesi che lo strumento qui realizzato sia di effettivo ausilio alla didattica, e che il suo utilizzo giovi alla comprensione della materia da parte degli studenti.
Attenzione che particolare attenzione dovrebbe essere posta nella selezione dei campioni X e Y che dovrebbero essere scelti a caso nel campione di studenti. Affidarsi alla volontarietà degli studenti a provare lo strumento falserebbe il campione, in quanto presumibilmente psrteciperebbero all'utilizzo dello strumento gli studenti già di per se più volenterosi e quindi capaci di ottenere, generalmente, voti maggiori dei compagni meno volenterosi.

6.4.2 Usabilità dell'interfaccia

Per quanto riguarda la valutazione dell'usabilità dell'interfaccia, invece, la strada da seguire è più semplice e consiste nella preparazione di questionari. Si tratta quindi di valutare il corretto svolgimento dei compiti assegnati all'utente evidenziando le sue difficolta. Esempi di domande potrebbero essere, dato ad esempio il compito di caricare la molecola di NH_3 e visualizzare i volumi isopotenziali in un certo intervallo:

- Sei riuscito ad aprire il file con la molecola? Se no perché?(test a risposta multipla)
  - si
  - si, ma con difficoltà
  - no, non ho trovato il file
  - no, il file selezionato non funziona
  - altro, spiegare (con annesso spazio per scrivere)
- Hai visualizzato correttamente le superfici isopotenziali?
  - si
  - no, non ho trovato l'opzione
  - ...

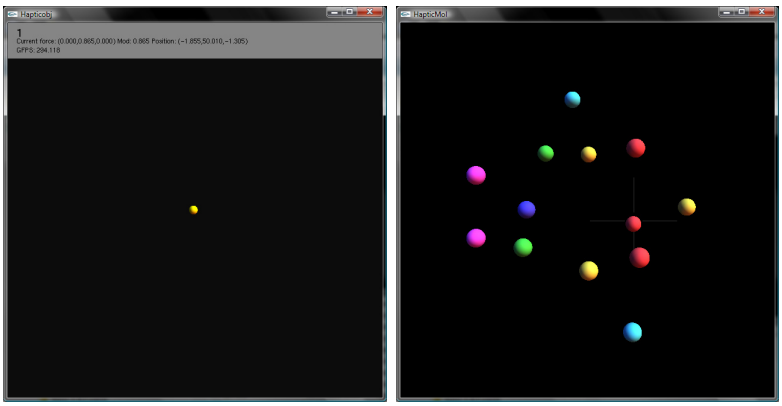

Figura 6.13: prove sperimentali per la valutazione dell'interfaccia aptica

- Sei riuscito a selezionare l'intervallo di valori desiderato?

  ...

- Come valuti in una scala da 1 a 5 la visualizzazione delle superfici isopotenziali?

- ...

Dalle risposte ai questionari si potrebbero evidenziare delle falle nell'interfaccia che porterebbero ad un miglioramento dell'usabilità.
Altro strumento per valutare l'usabilità del sistema consisterebbe nel registrare intere sedute di utilizzo del sistema da parte di utenti, applicando la tecnica del *thinking aloud* per cui l'utente accompagna la sua interazione con lo strumento con l'esposizione dei suoi pensieri a voce alta. Questo aiuterebbe, anche maggiormente dei questionari, a valutare le difficoltà dell'utente, nonché le sue aspettative e l'entusiasmo per il sistema.

6.4.3 Efficacia dell'interazione multisensoriale

Al fine di valutare l'efficacia dell'interazione aptica e sonora si è pensato di strutturare varie prove che proponessero dei semplici quesiti all'utente. La percentuale di superamento di tale prove, che identificano varianti rappresentative di uno stesso fenomeno, consente di valutare una modalità di rappresentazione rispetto ad un'altra, e di migliorare la qualità dell'interazione. Nello specifico queste prove consistono nell'isolare particolari aspetti dell'applicazione, ad esempio la sola interfaccia aptica, in modo da poter valutare l'interazione dell'utente con il particolare sistema.
In figura 6.13 sono presentati due tipologie di prove sperimentali introdotte:

1. la prima consiste in un'interfaccia estremamente semplificata contraddistinta da una schermata graficamente vuota che contiene una semplice forma geometrica nascosta. Tale forma è percepibile con lo strumento aptico secondo diverse modalità, inoltre allo stimolo aptico viene associato, in una variante, uno stimolo sonoro. Scopo della prova è, ovviamente, il riconoscimento della forma. Questo esperimento permette di indagare la bontà di varie modalità di interazione aptica, se, infatti, più utenti riescono a riconoscere la forma in una certa modalità rispetto ad un'altra, significa che tale modalità rappresenta meglio l'entità geometrica. Nello specifico le dimensioni indagate con questo esperimento sono:
   - la rappresentazione aptica di una forma geometrica reagendo a seguito del contatto con la forma stessa o rendendo tale forma un vincolo da cui lo strumento aptico non è possibilitato a distaccarsi;
   - se l'aggiunta dell'acusmetria a seguito del contatto con la molecola aiuta o meno l'utente nel suo compito di riconoscimento della forma.
2. la seconda consiste nel riconoscimento del colore dei punti di attrazione e di repulsione all'interno di un campo di forze in cui sono renderizzate graficamente diverse palline colorate. L'utente attraverso l'esplorazione del campo di forze con lo strumento aptico deve riconoscere il colore delle palline che formano questo campo di potenziale. L'interazione e fornita con e senza supporto sonoro in modo da appurare l'apporto che viene fornito all'esplorazione aptica del potenziale dalla presenza dei suoni proporzionali al potenziale.

Di queste due esperienze sono stati effettivamente realizzati dei prototipi utilizzabili, di altre si è solo sviluppata l'idea.
Altri esperimenti che sono stati pensati sono:

1. La valutazione delle proprietà aptiche degli oggetti quali ruvidezza e morbidità. A schermo vengono mostrate 3 sfere di diverso colore a cui vengono associati valori diversi rispetto ad una proprietà, diciamo la stiffness; l'utente dovrà individuare il colore della sfera che percepisce più morbida, ad esempio nel caso della stifness, o più ruvida nel caso di attrito.
2. La localizzazione di una sorgente audio tramite acusmetria variando le mappatura delle dimensioni spaziali rispetto proprietà sonore. L'utente deve individuare rispetto a diverse palline colorate quella che

secondo lui sta emettendo il suono. Questo serve a valutare quali caratteristiche del suono si prestano meglio ad una interpretazione acusmetrica.

3. La selezione aptica di un oggetto tra tanti, ad esempio sfere, celando o meno il cursore aptico e attivando o meno gli effetti sonori. In questo modo si capirebbe se l'audio aiuta ad orientare l'utente in uno spazio 3d e se l'utente è in grado di orientarsi senza altro stimolo che quello aptico.

4. Prove di interazione in cui all'utente viene chiesto di spostare un oggetto in una certa posizione e con una certa rotazione utilizzando il mouse e utilizzando lo strumento aptico. Misurando i tempi di svolgimento del compito si avrebbe un'indicazione sulla bontà e sulla praticità di una modalità di interazione rispetto all'altra.

5. ...

Risulta chiaro come questi esperimenti siano in grado di fornire dati concreti rispetto ad un semplice questionario. Dalla ripresa degli utenti durante le prove e da un'intervista successiva è inoltre possibile ottenere ulteriori informazioni e spunti.
Non solo l'interazione aptica e sonora possono essere indagate attraverso questi strumenti, anche l'interazione grafica può usufruire di una simile metodologia. Forniamo di seguito un esempio: i colori associati alla scala logaritmica sono stati assegnati in maniera arbitraria, l'analisi di diverse tipologie di colori assegnate a forze e potenziale potrebbe fornire una colorazione più efficace per rappresentare l'informazione associata. Ad alcuni utenti potrebbe essere richiesto di fornire un'imprssione sul valore di potenziale associato ad una scala di colori, questo permetterebbe di associare l'informazione colorimetrica all'idea di potenziale secondo una metafora stabilita dagli utenti piuttosto che convenzionale. Se nella costruzione mentale degli utenti si associa un potenziale negativo a qualcosa di freddo, allora il colore più naturale sarà il blu o l'azzurro, se invece viene associato a qualcosa di caldo diventerà rosso e via dicendo. In figura 6.14 sono mostrati diversi esempi di scale colorimetriche logaritmiche, in cui la scelta di una piuttosto che un'altra potrebbe essere ricondotta al giudizio degli utenti secondo la loro idea di associazione con l'idea di potenziale.

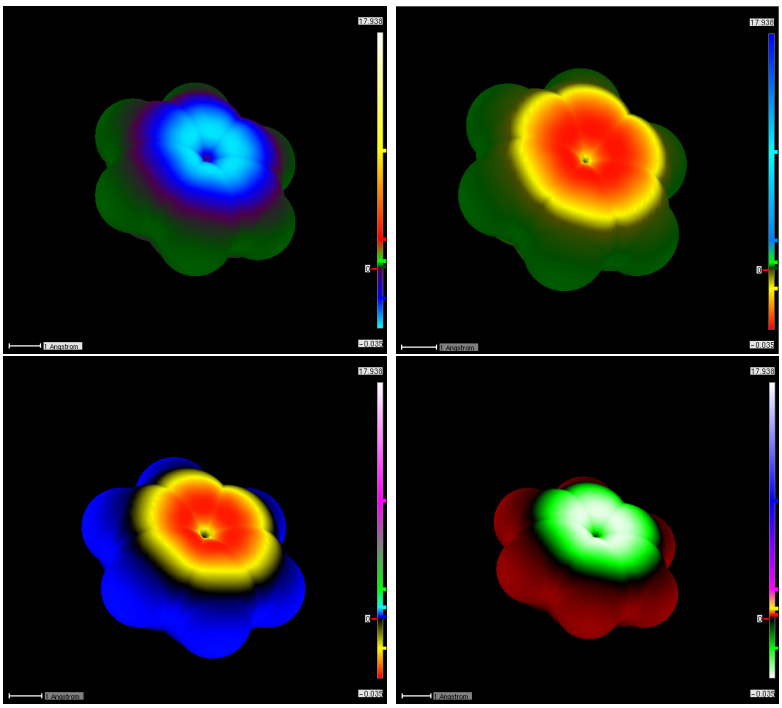

Figura 6.14: diverse scale di colorazione logaritmica

6.5 Conclusioni del capitolo

In questo capitolo sono state elencate le molecole utilizzate nel sistema e sono stati introdotte le modalità di utilizzo principali, quali la didattica e la ricerca, individuando dei casi specifici di impiego nell'ambito didattico.
Inoltre sono stati introdotti vari spunti per la valutazione del sistema. Questi possono essere di grande ausilio ad una successiva valutazione del sistema atta ad individuarne dei miglioramenti e possono rappresentare un punto di inizio per una espansione del simulatore.
Il prossimo capitolo, quello conclusivo, trae le fila di quanto realizzato ed introduce possibili strade per continuare lo sviluppo dello strumento aggiungendo funzionalità o migliorando quelle esistenti.

Capitolo 7

Direzioni future di ricerca e conclusioni

RIASSUMENDO nel presente lavoro di tesi è stato introdotto e spiegato profusamente un sistema di simulazione di meccanica molecolare capace di renderizzare visualmente, apticamente ed acusticamente le informazioni salienti relative alla molecola in esame, quali geometria e derivati del MEP[1] calcolati secondo varie metodologie e dinamiche.
Nel corso della trattazione sono state illustrate le difficoltà riscontrate e la relativa risoluzione creativa come, ad esempio, l'introduzione delle scale iperlogaritmiche per la renderizzazione aptica delle forze.

Nello scorso capitolo sono state proposte idee in merito ad una valutazione approfondita della qualità del sistema da utilizzare come traccia per future analisi sull'usabilità e sull'interazione dell'utente con l'interfaccia aptica e sonora del sistema stesso. Questo punto andrà sicuramente approfondito in futuro, utilizzando come strumento per la valutazione dell'interazione aptica anche il framework spiegato successivamente in appendice.
Una valutazione completa di tali aspetti permetterebbe, infatti, di definire i punti dolenti del sistema e di fornire una traccia per il miglioramento dello stesso, nonché di giudicare obiettivamente il lavoro svolto.

Possibili estensioni del sistema sono inoltre da ricercare nell'introduzione di nuove dinamiche molecola-molecola, trasformando il cursore aptico da carica di prova a molecola di prova, in grado di mostrare non solo il com-

[1] potenziale, campo di forze, minimi e massimi, punti isopotenziali, linee di forza, gradiente della forza *ldots*

portamento di una carica di prova all'interno di un campo di potenziale, ma anche l'interazione tra varie molecole compresa la formazione di nuovi legami. Tale compito risulta più complesso di quanto possa sembrare, in quanto necessita dell'ideazione di un sistema per permettere alle due griglie di potenziale di influenzarsi reciprocamente[2] secondo modalità quanto più precise e vicine alla realtà possibili. Modellare quindi il comportamento di più atomi in maniera precisa richiede approfondite competenze chimiche, poiché si potrebbe essere facilmente tratti in inganno da una soluzione semplice, ma completamente scorretta da un punto di vista teorico.
Caratteristica fondamentale del sistema realizzato è infatti l'aderenza a dati sperimentali reali quali le griglie di potenziale, rappresentanti un punto di riferimento rispetto a tutte le funzionalità introdotte. Realizzare un sistema funzionante ed esteticamente appagante potrebbe risultare un completo fallimento se non venisse rispettata l'aderenza alla realtà rappresentata, inoltre, essendo uno strumento didattico, potrebbe fuorviare la comprensione stessa della materia da parte dello studente utente.

Oltre all'introduzione della dinamica molecolare nel sistema è possibile migliorare altri aspetti di quanto realizzato:

- rappresentazione migliore delle superfici isopotenziali attraverso tecniche di volume rendering. Sebbene la visualizzazione per punti raggiunga lo scopo preposto, diventa confusionaria e poco efficace quando sono presenti troppi o troppo pochi punti nell'intervallo di potenziale selezionato e quando la griglia di potenziale risulta poco 'fitta' rispetto alle dimensioni di un singolo atomo. Tramite tecniche di Volume rendering si potrebbero sfruttare le informazioni ottenute interpolando i dati e disegnare superfici isopotenziali più accurate.

- miglioramento degli effetti sonori relativi alla percezione del potenziale sul cursore aptico. Sebbene funzionale questa caratteristica richiede una analisi approfondita di design del suono; la sonorità del sistema risulta infatti particolarmente molesta quando i suoni vengono generati direttamente dal sistema, senza cioè l'ausilio di un file wave esterno. Un modo per risolvere il problema potrebbe essere associare a tali eventi sonori dei suoni gradevoli appositamente costruiti o in alternativa reingegnerizzare l'interfaccia sonora proponendo soluzioni alternative.

[2] si noti che il MEP chimicamente non è additivo, per cui avvicinando due molecole varia in maniera non lineare

- il calcolo del potenziale tramite formule potrebbe essere reso più preciso aggiungendo contributi di potenziale, come quanto fatto nel presente sistema con l'aggiunta della correzione di Lennard Jones. Questa modifica richiede però una conoscenza specialistica degli argomenti chimici trattati che esula dalle competenze di un ingegnere informatico. La presenza di un calcolo più preciso del potenziale tramite formule permetterebbe l'introduzione di elementi di dinamica molecolare nel sistema quali, ad esempio, creazione di legami e interazione molecola-molecola.

- introduzione di nuovi formati grafici per la rappresentazione della molecola che esulano dalla rappresentazione a sfere, quali ad esempio quella a nastro presente in altri software di simulazione molecolare.

- ...

Inoltre il presente sistema può essere visto come una base per la creazione di piattaforme più complesse che comprendano, ad esempio, modalità di interazione cooperativa per cui un utente speciale, diciamo il professore di chimica, abbia il completo controllo sulle opzioni e sulla rototraslazione della molecola nel sistema, mentre altri utenti, gli studenti, possano soltanto percepire apticamente, visivamente e acusticamente l'oggetto senza interagire direttamente. Un'estensione del genere sarebbe utilissima da un punto di vista didattico, ma richiederebb l'implementazione di una architettura di rete con una architettura client-server capace di gestire varie tipologie di utente.
Una architettura client-server sul web permetterebbe di mantenere un enorme repository di molecole in un server centrale, scaricando il client dalla necessità di possendere in proprio un repository piuttosto dispendioso di memoria[3].

Utilizzando la piattaforma realizzata in questo contesto, opportunamente modificata, si potrebbe pensare di realizzare dei giochi-test da sottoporre all'utente, con dei compiti e degli scopi precisi da raggiungere. Questo trasformerebbe la piattaforma in un sistema didattico sotto forma di gioco utile allo studente per autovalutarsi divertendosi. Esempi di task potrebbero essere:

- individuazione del comportamento sbagliato, in cui vengono proposte varie alternative di potenziale per una molecola, l'utente deve dire quale di questi comportamenti sia corretto o meno.

[3]ogni file mep occupa infatti una trentina di MB

- localizzazione del punto di minimo, l'utente, dopo aver percepito il campo di forze per qualche secondo, deve individuare nello spazio i punti di minimo del potenziale molecolare cliccando il pulsante sul phantom, in base alla distanza dal punto reale di minimo riceve un punteggio;

- individuazione della molecola a partire dalla percezione del potenziale, per cui percependo apticamente il potenziale molecolare l'utente deve discriminare la sorgente tra tante possibili molecole.

- ...

Come si può dunque notare dall'ampia trattazione, le possibilità di ampliamento del progetto dipendono solo dalla creatività di chi vorrà continuare a lavorarci, in quanto il sistema realizzato è indiscutibilmente uno strumento dalle enormi potenzialità con i suoi tratti originali rispetto a quanto già presente sul mercato.

Grazie anche all'aumento esponenziale della potenza di calcolo dei computer,si ipotizza che simulazioni in realtà virtuale saranno sempre più presenti e richieste in futuro.
L'introduzione e l'affinamento di strumenti aptici è il futuro dell'interazione umana con i computer. Negli ultimi anni è infatti cresciuto sempre più il fabbisogno di nuove periferiche con cui interagire con le tecnologie; basti pensare a sistemi ad interfacce tangibili o alle tecnologie touch oggi molto in voga per via della massificazione dei dispositivi elettronici. La pervasione di tecnologie in cui l'uomo è immerso richiede una particolare attenzione all'interazione uomo-macchina che deve essere il più naturale possibile E cosa c'è di più naturale per l'uomo dell'utilizzo del suo stesso corpo come device per interfacciarsi con la tecnologia. Tecnologia che deve essere dal canto suo capace di coinvolgere attivamente tutti i sensi dell'utente sia dal punto di vista di input, accettando e riconoscendo i gesti dello stesso, sia da un punto di vista di output, essendo in grado di restituire feedback in ogni dimensione sensoriale.

Concludendo, l'intera società umana è a una svolta della sua storia, nel pieno di una nuova rivoluzione tecnologica, capace potenzialmente di indurre radicali cambiamenti alla vita sull'intero pianeta.
Simulazioni in realtà virtuale possono concretamente aiutare la ricerca in quasi ogni campo scientifico. Tramite controlli in telepresenza, in grado di offire maggiori possibilità di interazione con l'utente umano, possono essere

prodotti robot capaci di sostituirsi a compiti particolarmente pericolosi. Mondi virtuali possono permettere all'uomo di esperienziare l'impossibile senza rischi, e sempre più flebile sarà il limite tra realtà e finzione. Nonostante quanto detto, l'uomo deve sviluppare un senso critico affinato per discriminare tra ciò che è reale e ciò che non lo è. Se da un lato l'uomo necessita di sognare attraverso l'immaginazione, i sogni hanno bisogno della concretezza del reale per essere raggiunti.

Bibliografia

[1] Marco faverio, *Progettazione e realizzazione di visualizzatore molecolare 3D con supporto aptico*,Tesi di laurea Specialistica 2007/08, Politecnico di Milano, sede di Como

[2] J.J. Gibson, *The senses considered as perceptual systems.*, Boston, Houghton Mifflin, 1966.

[3] Laycock, S. D., Day, A.M., *A Survey of Haptic Rendering Techniques*, COMPUTER GRAPHICS, 26, 1, 2007, 50-65.

[4] Petter Bivall Persson, Matthew Cooper, Anders Ynnerman, Lena Tibell*Chemical Force Feedback*, Web site, http://cff.itn.liu.se/public/ .

[5] Sato, M., Liu, X., Murayama, J., Akahane, K., Isshiki, M., *A Haptic Virtual Environment for Molecular Chemistry Education*, Transactions on Edutainment I, 5080, 2008, 28-39.

[6] H. Iwata et al., *Project FEELEX: Adding Haptic Surface to. Graphics*, Proceedings of SIGGRAPH 2001, 2001.

[7] Tsukuba University, VR Lab, *HapticMaster*, Web site, http://intron.kz.tsukuba.ac.jp/hapticmaster/hapticmaster_e.html

[8] *H3D.org - Open Source Haptics*, Web site, http://www.h3dapi.org/

[9] Hasser, Christopher J., *HAPTAC: A Haptic Tactile Display for the Presentation of Two-Dimensional Virtual or Remote Environments.*,Proc. of the IEEE Int. Conf. on Robotics and Automation, 1992.

[10] Arthur D. Gregory, Ming C. Lin, Stefan Gottschalk, and Russell M. Taylor II., *H-Collide: A Framework for Fast and Accurate Collision Detection for Haptic Interaction*, Proceedings of IEEE Virtual Reality Conference 1999.

[11] M. de Pascale and D. Prattichizzo, *The Haptik Library: A Component Based Architecture for Uniform Access to Haptic Devices*, IEEE Robotics & Automation Magazine, vol. 14, no. 4, pp. 64-75, Dec. 2007

[12] Conti F, Barbagli F, Morris D, Sewell C., *CHAI 3D: An Open-Source Library for the Rapid Development of Haptic Scenes*, IEEE World Haptics, Pisa, Italy, March 2005.

[13] F. R. El-Far, M. Eid, M. Orozco, A. El Saddik, *Haptic Applications Meta-Language*, Proceedings of the 10th IEEE International Symposium on Distributed Simulation and Real Time Applications, 2006.

[14] Peter Berkelman, *Tool-Based Haptic Interaction with Dynamic Physical Simulations using Lorentz Magnetic Levitation*, Robotics Institute, Carnegie Mellon University, 1999.

[15] Richard, E., Tijou, A., Richard, P., *Multi-modal virtual environments for education with haptic and olfactory feedback*, Virtual Reality, 10, 3-4, 2006, 207-225.

[16] Fjeld M., Juchli P., Voegtli B., *Chemistry education: a tangible interaction approach*, Proc. of INTERACT 2003, Zurich, Switzerland, 2003, 287-294.

[17] *OpenGL: The Industry's Foundation for High Performance Graphics*, Web site, http://www.opengl.org/

[18] *GLUT The OpenGL Utility Toolkit*, Web site, http://www.opengl.org/resources/libraries/glut/

[19] Paul Rademacher, *GLUI User Interface Library*, Web site, http://www.cs.unc.edu/řademach/glui/

[20] *OpenBabel: The Open Source Chemistry Toolbox*, Web site, http://openbabel.org/wiki/Main_Page

[21] Sensable, *PHANTOM Omni*, Web site, http://www.sensable.com/haptic-phantom-omni.htm

[22] Sensable, *OpenHaptics Toolkit*, Web site, http://www.sensable.com/products-openhaptics-toolkit.htm

[23] Russell Smith, *Open Dynamics Engine*, Web site, http://www.ode.org/

[24] Creative Labs, *OpenAL: Open Audio Library*, Web site, http://connect.creativelabs.com/openal/default.aspx

[25] Creative Labs, *The OpenAL Utility Toolkit (ALUT)*, Web site, http://connect.creativelabs.com/openal/Documentation/The%20OpenAL%20Utility%20Toolkit.htm

[26] Microsoft,*Microsoft Speech API*, Web site, http://msdn.microsoft.com/en-us/library/ms723627%28v=VS.85%29.aspx

[27] *PDB: Protein data bank. A Resource for Studying Biological Macromolecules* , Web site, http://www.pdb.org/pdb/home/home.do

[28] B.M. Bode and M.S. Gordon, *MacMolPlt: A Graphical User Interface for GAMESS*, J Molec. Graphics., 16, 133(1999).

[29] Gordon Group, *GAMESS: The General Atomic and Molecular Electronic Structure System.*, Web site, http://www.msg.chem.iastate.edu/gamess/

[30] M. Rauterberg, E. Styger, A. Baumgartner, A. Jenny, and M. de Lange, *Additional sound feedback in man-computer interaction: two empirical investigations.*

[31] Angela Chang and Conor O'Sullivan, *An Audio-Haptic Aesthetic Framework Influenced by Visual Theory.*

[32] Caroline Jay, Robert Stevens,, *Roger Hubbold and Mashhuda Glencross , Using Haptic Cues to Aid Non-Visual Structure Recognition.*

[33] Christian Müller-Tomfelde, *Interaction Sound Feedback in a Haptic Virtual Environment to Improve Motor Skill Acquisition.*

[34] Mohamad Eid, Mauricio Orozco and Abdulmotaleb El Saddik, *A guided tour in haptic audio visual environments and applications.*

[35] G. Huang, D. Metaxas and M. Govindaraj, *Feel the .Fabric.: An Audio-Haptic Interface.*

[36] E.Lariani, M.Maiocchi, F.Rampichini, *Acusmetria.*

Appendice A

Progetto collaterale al framework realizzato

In questa appendice è illustrato un progetto collaterale a quello presentato in questo lavoro di tesi sviluppato nel contesto del corso Computer Graphics Systems and Applications, realizzato dal medesimo autore. Viene pertanto inserita una versione ridotta, per evitare ridondanza, di quanto realizzato. Viene omessa inoltre la descrizione delle classi molto simili a quelle introdotte durante la trattazione di questa tesi.

A.1 Introduzione

Un alto grado di immersività in un sistema virtuale è molto difficile da raggiungere a causa della mancanza quasi totale di feedback tattili nelle applicazioni desktop, dovuta alla scarsa diffusione degli strumenti necessari a ottenere un'interazione aptica. Se infatti evolute schede video e audio permettono un elevato grado di interazione visiva e sonora, dal punto di vista aptico bisogna interfacciarsi con strumenti esterni, ancora molto lontani dall'essere uno standard e spesso molto limitati. Nel sistema che si andrà a descrivere nelle prossime pagine è stata utilizzato il PHANTOM Omni prodotto dalla Sensable, uno strumento composto da un braccio meccanico al cui end effector è collegata una penna, che è in grado di restituire un feedback in forza all'utente, guidando la mano che la impugna nell'ambiente 3d.
Il presente progetto rappresenta una generalizzazione di un framework aptico di simulazione molecolare sviluppato nell'ambito di una tesi di laurea di secondo livello. Come tale rielabora la struttura del sistema da cui deriva rendendola più generale, in grado cioè di importare una qualsiasi mesh in

formato object.
Lo scopo di questa applicazione è fornire uno strumento di test rispetto alle potenzialità dello strumento aptico, infatti chiave dello strumento è la possibilità da parte dell'utente di regolare i parametri caratteristici, in modo da poter studiare a fondo le dinamiche di interazione del sistema e il livello delle sensazioni percepite dall'utente che vi si interfaccia.
Dal punto di vista logico il sistema si articola in diversi engine, ognuno corrispondente ad un aspetto dell'applicazione:

- **GraphicEngine**: corrisponde alla parte grafica del sistema; sfrutta la libreria OpenGL per visualizzare a schermo la mesh selezionata.
- **UserInterfaceEngine**: gestisce la finestra in cui il sistema svolge le sue funzionalità e definisce l'interfaccia per le opzioni del sistema; usa le librerie GLUI e GLUT.
- **AudioEngine**: si occupa di gestire gli output sonori del sistema; si interfacia con la scheda audio attraverso le librerie OpenAL e Alut, inoltre fa uso della libreria SAPI di windows per la gestione della sintesi vocale.
- **HapticEngine**: si interfaccia con lo strumento aptico definendone i comportamenti; sfrutta la libreria OpenHaptics.
- **DynamicEngine**: rappresenta e definiscde la dinamica fisica del sistema; utilizza la libreria OpenDynamicsEngine.

Come si può notare l'applicazione si sviluppa su diverse dimensioni, in grado di fornire all'utente un livello di interazione con il sistema vario e complesso. Nei prossimi capitoli si analizzerà nel dettaglio il sistema dal punto di vista logioo, in modo da fornire al lettore una panoramica approfondita del sistema atta ad agevolare eventuali sviluppi futuri.

A.2 Funzionamento logico del sistema

Come già accennato il sistema si articola in cinque differenti engine, grafico, audio, aptico, interfaccia utente e dinamico che comunicano tra di loro attraverso un oggetto, chiamato *Object3d* che rappresenta le proprietà della mesh da renderizzare.
Nello specifico dopo aver inizializzato tutti gli engine, l'interfaccia utente si occupa di ottenere il file desiderato dall'utente attraverso un pulsante che apre una finestra di selezione. A questo punto, parsando correttamente suddetto file viene creato l'oggetto *Object3d*, che mantiene al suo interno tutte

le proprieta dell'oggetto, quali materiale grafico, vertici, normali, proprietà aptiche etc.
L'engine grafico si occupa quindi di rendere la mesh graficamente caricando in memoria la lista di vertici e di normali. Nel frattempo l'engine aptico si occupa di renderizzare la stessa mesh da un punto di vista aptico. Infine l'engine dinamico crea, a partire dalle proprieta spaziali della mesh, un oggetto dotato di massa, nel nostro caso un cubo di peso e grandezza definibili dall'utente, che rappresenta la mesh renderizzata graficamente.
In questo contesto l'interfaccia utente permette di:

- modificare i parametri dell'oggetto grafici, aptici e dinamici;
- modificare il modo con cui viene renderizzato l'oggetto graficamente (camera prospettica o ortogonale).
- modificare come viene renderizzato apticamente l'oggetto cioè considerando la mesh come una superficie di contatto, come vincolo per cui l'utente è costretto a seguire la superficie dell'oggetto, oppure non considerarla affatto disattivando del tutto la percezione aptica, secondo le esigenze.
- infine vi è la possibilità di attivare/ disattivare la dinamica dell'oggetto.

L'interfaccia audio fa da sfondo a tutto questo agendo sia nell'interfaccia utente sia nel sistema grafico. Nello specifico viene attivato un evento sonoro ogni qualvolta vengono premuti pulsanti nell'interfaccia e viene fatta leggere alla sintesi vocale l'opzione selezionata/deselezionata, se possibile. In aggiunta viene attivato un suono ogni qualvolta il phantom tocca l'oggetto 3d, suono che da un'indicazione percettiva del punto nello spazio in cui è avvenuto il contatto. Questo avviene grazie alla mappatura dello spazio 3d in tre dimensioni sonore: la caratteristica stereo del suono indica l'asse x, la frequenza del suono l'asse y, mentre il guadagno l'asse z. Infine all'apertura del file ne viene pronunciato il nome.

A.2.1 Interfaccia utente

L'interfaccia utente, mostrata in figura A.1, si compone di tre parti distinte:

- Una finestra grafica 3d OpenGL in cui vengono renderizzate la mesh e il cursore aptico;

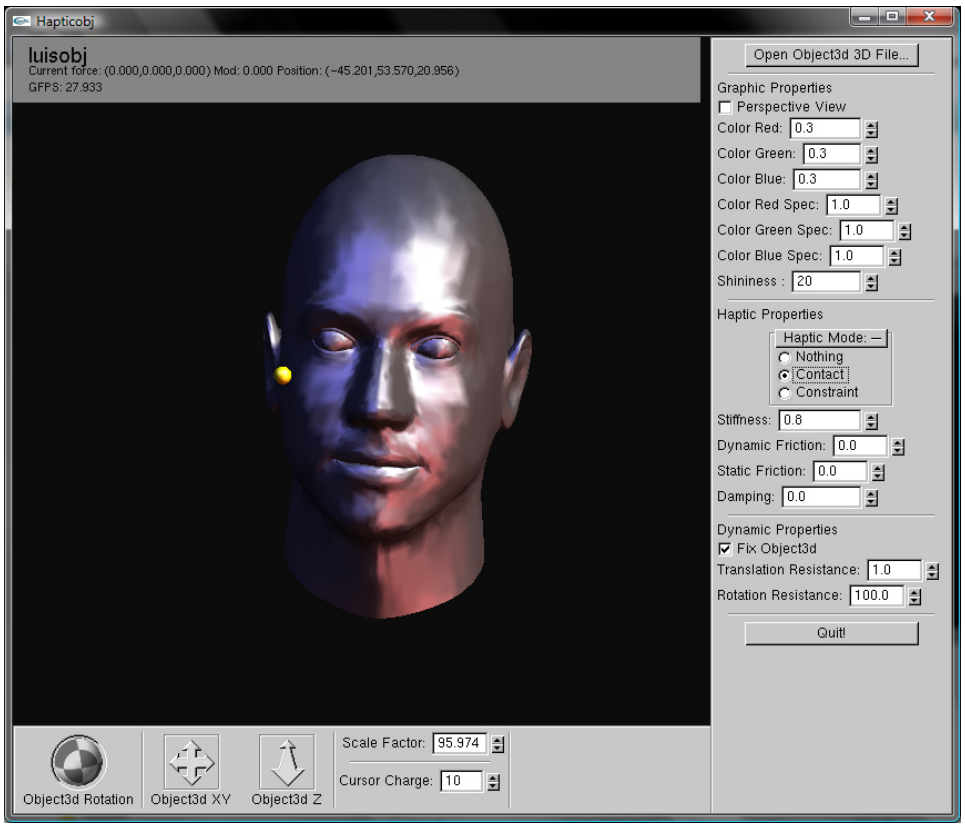

Figura A.1: Interfaccia del sistema

- Un insieme di comandi sottostanti tale finestra, che identificano i controlli diretti sull'oggetto (rotazione, traslazione, scaling).
- Un insieme di opzioni laterali a destra che identificano le proprietà dell'oggetto.

Analizzando con ordine l'interfaccia si può facilmente notare la mesh renderizzata, nell'esempio una testa, e a lato di questa una piccola sfera gialla che rappresenta il cursore aptico nello spazio 3d. Nella stessa finestra in alto a sinistra è visibile il nome dell'oggetto, mentre nello spazio sottostante sono indicate alcune informazioni, quali la forza al momento renderizzata dal sistema aptico e la posizione del cursore, mentre l'ultima informazione riportata sono gli attuali frame al secondo.
Nella finestra di comandi sottostanti troviamo in ordine:

- un controllo per la rotazione, identificato da una sfera rotante: premendo tale pulsante e muovendo il mouse è possibile ruotare l'oggetto renderizzato.

- un controllo per la traslazione sugli assi x, y, cioè sul piano dello schermo, identificato da due frecce incrociate; premendo e spostando il mouse anche l'oggetto renderizzato trasla.

- un controllo per la traslazione sull'asse z, perpendicolare allo schermo.

- uno slider per lo scaling dell'oggetto, assume un valore da 0.0 a 999 e permette di ingrandire/rimpicciolire la mesh.

- un ulteriore slider per definire la forza associata al cursore che rappresenta un fattore moltiplicativo della forza percepita dal sistema nell'interazione dinamica tra oggetto e phantom.

La finestra laterale, invece, può essere suddivisa in 5 parti:

- un pulsante **Open Object3d File** che una volta premuto apre una finestra di selezione file.

- un insieme di controlli sulle proprietà grafiche del sistema:

  - una checkbox che identifica l'attivazione o meno di una camera prospettica. Se non attivato implica l'ortogonalità della proiezione.
  - tre slider che definiscono un valore da 0.0 a 1.0 per i colori RGB della mesh, per quanto riguarda le caratteristiche diffusive e ambientali del materiale.
  - tre ulteriori slider da 0.0 a 1.0 per definire le proprietà RGB della componente speculare del materiale.
  - uno slider che identifica il grado di brillantezza, in rispetto alla componente speculare della luce, della mesh, varia da 0 componente speculare disattivata, a 200.

- un insieme di controlli sulle proprietà aptiche del sistema:

  - un selettore di opzioni in cui è possibile decidere la modalità di interazione aptica con la mesh:
    * Nothing: indica l'assenza di tale interazione, lo strumento aptico non incontra vincoli che lo blocchino nell'ambiente 3d.
    * Contact: la mesh viene renderizzata come superficie, perciò viene percepita dall'utente solo quando il cursore aptico collide con la mesh.

        * Constaint: Indica che la superfice dell'oggetto è renderizzata come vincolo che obbliga il cursore aptico a essere sempre a contatto con la mesh; la forza percepita sul phantom è quindi proporzionale alla distanza dalla superficie stessa.
    - uno slider che indica la stiffness, cioè durezza del materiale. Questo valore, come tutti quelli che seguono, varia da 0.0 a 1.0 e indica la resistenza che si ottiene premendo sulla mesh.
    - uno slider per l'attrito dinamico, indica quanto il materiale resiste rispetto allo scorrimento del cursore sulla superficie una volta che il cursore è in movimento.
    - uno slider per l'attrito statico, indica la difficolta una volta venuti a contatto con la superficie di iniziare a strisciare su di essa.
    - infine uno slider per il damping, cioè lo smorzamento, che indica la resistenza del materiale alla velocità con cui viene toccato.

- un insieme di controlli sulle proprietà dinamiche del sistema:
    - una checkbox per definire se l'oggetto debba rimanere fisso o se sia suscettibile alle forze esercitate con il phantom.
    - uno slider da 0.001 a 100 che definisce la resistenza del corpo alla traslazione [1]. Più è alto tale valore più il corpo tenderà a mantenere il suo centro fermo.
    - uno slider da 0.001 a 1000 che indica la resistenza dell'oggetto a ruotare[2].

- infine un pulsante per terminare l'applicazione.

Vi sono inoltre due metodi di controllo ulteriore dell'interfaccia: tramite il mouse è possibile traslare o ruotare direttamente l'oggetto nella schermata 3d tenendo premuto il tasto sinistro e destro rispettivamente e muovendo il puntatore in una direzione; oppure tramite tastiera, ad esempio i tasti + e − aumentano e diminuiscono lo scale dell'oggetto.

A.2.2 Importazione oggetto 3d

Sì è gia visto tramite l'interfaccia come è possibile aprire la finestra per selezionare un file, ma come viene importato effetivamente nel programma? I file che il programma è in grado di aprire e parsare sono di tipo object

[1] indica la massa dell'oggetto

[2] sono le dimensioni del cubo che identifica dinamicamente l'oggetto

```
# Blender3D v248 OBJ File:
# www.blender3d.org
v 1.000000 -1.000000 -1.000000
v 1.000000 -1.000000 1.000000
v -1.000000 -1.000000 1.000000
v -1.000000 -1.000000 -1.000000
v 1.000000 1.000000 -1.000000
v 0.999999 1.000000 1.000001
v -1.000000 1.000000 1.000000
v -1.000000 1.000000 -1.000000
vn -0.000000 -1.000000 0.000000
vn 0.000000 1.000000 -0.000000
vn 1.000000 0.000000 0.000000
vn -0.000000 -0.000000 1.000000
vn -1.000000 -0.000000 -0.000000
vn 0.000000 0.000000 -1.000000
usemtl (null)
s off
f 1//1 2//1 3//1 4//1
f 5//2 8//2 7//2 6//2
f 1//3 5//3 6//3 2//3
f 2//4 6//4 7//4 3//4
f 3//5 7//5 8//5 4//5
f 5//6 1//6 4//6 8//6
```

Figura A.2: file .obj di un semplice cubo

(.obj). Tali file hanno una struttura particolare al loro interno, vedere figura A.2, e indicano in ordine:

- una lista di vertici, formattati nel seguente modo v 0.114575 0.456612 1.55765 in cui i tre numeri floating point rappresentano le coordinate del punto.
- una lista di coordinate per le texture, che noi non considereremo.
- una lista di vettori normali formattati allo stesso modo dei vertici vn 1.0000 0.00000 0.00000.
- infine una lista di poligoni identificati nel seguente modo f 123/12/15 142/24/17 157/165/12 dove i tre valori separati da / indicano rispettivamente il vertice, le coordinate della texture e il vettore normale corrispondenti ad un punto della mesh triangolare definita dai tre valori. Può avere anche quattro diversi punti e in tal caso identifica un rettangolo.

L'algoritmo sviluppato per il parsing funziona in questo modo:

1. in primo luogo parsa tutto il file per avere le seguenti informazioni: numero di Vertici, numero di Normali, numero di Figure, e numero di vertici per figura(triangoli o rettangoli).
2. alloca tre array uno per i vertici v, uno per le normali n e uno per le figure f moltiplicando per 3 il numero trovato in precedenza (o 4 in

f nel caso che le mesh siano rettangolari). Alloca infine un ulteriore array della dimensione dei vertici per riordinare le normali n_1.

3. scorre nuovamente tutto il file riempiendo v e n.

4. giunto alla parte delle mesh, riempe f con l'indice dei vertici, e nel frattempo riordina i vettori normali associando all'indice corrsipondente al vertice il valore di normale corrispondente. n_1 conterrà ora gli stessi dati di n, riordinati però in modo da essere associati univocamente al vertice corrispondente.

5. nel frattempo calcola il centro della mesh facendo la media di tutti i vertici, e calcola il valore di scale adatto a mostrare la mesh ad una dimensione predefinita.

Finita questa procedura viene deallocato n e si ottengono i tre vettori che servono per renderizzare la mesh.
Nel frattempo le altre proprietà dell'oggetto, quali traslazione, rotazione, materiale grafico e aptico e varie opzioni vengono impostate ad un valore di default che potrà essere modificato attraverso l'interfaccia.

A.2.3 Sistema grafico

Una volta ottenuto l'oggetto si passa alla visualizzazione della mesh che lo rappresenta.
Innanzitutto vengono identificati i parametri della proiezione del sistema: nel caso in cui la proiezione sia ortogonale viene definito un cubo di lato 420 dove il minimo della x e della z corrisponde a -210 mentre il minimo della y a -160, il motivo di questa divergenza è da ricercare nella presenza del rettangolo nella parte alta dello schermo che non rende lo spazio simmetrico.
Nel caso di proiezione prospettica invece il frustum applicato ha un piano di near a 20, un piano far a 500 e i piani left e bottom a -10.0, mentre right e up a 10, inoltre il frustum viene traslato di 250 unità sull'asse z.
Oltre alle varie opzioni OpenGL attivate di rito viene definito un sistema di illuminazione composto da tre luci:

- una prima luce, bianca, posta in direzione $(1, 1, 1, 0)$ angolo alto a destra dello schermo.

- una seconda luce, blu scura opaca, posta in direzione $(-1, 1, 1, 0)$ angolo alto a sinistra dello schermo.

- una terza luce, rossa scura opaca, posta in direzione $(1, -1, 1, 0)$ angolo basso a destra dello schermo.

Tutte e tre le luci hanno lo stesso colore per la componente diffusiva e speculare.
Prima di procedere all'analisi del ciclo di disegno, vengono attivati e caricati gli array che contengono i vertici e le normali della mesh.
A questo punto si può analizzare il ciclo di disegno del sistema. Per prima cosa si applicano le trasformazioni definite nell'oggetto 3d, una traslazione e una rotazione, dopodiché si effettua una ulteriore traslazione per centrare la mesh nell'origine dello spazio 3d, utilizzando il centro precalcolato. A seguire si disegna la mesh, sfruttando l'array f contenente gli indici dei vertici che corrispondono ad una mesh triangolare (o rettangolare). Dopodiché si torna nel sistema di coordinate globali e si disegna il cursore aptico. Infine, cambiando il materiale corrente in modo da risultare emissivo, si disegnano le informazioni contenute nel rettangolo in alto.
Misurando il tempo che il sistema impiega nel fare tutte queste operazioni è possibile ottenere l'informazione relativa ai frame grafici al secondo che viene stampata a video in alto.

A.2.4 Sistema aptico

Come già accennato il sistema aptico lavora in stretto contatto con la controparte grafica. Per cui tutto il lavoro relativo alla mesh viene effettuato nella stessa parte di codice dove viene renderizzata a schermo la mesh.
Ma procediamo con ordine, dopo le varie inizializzazioni del caso (attivazione del dispositivo aptico e creazione di un nuovo contesto) viene inizializzato una variabile intera come iodentificativo dell'oggetto 3d da renderizzare apticamnte. Vengono poi registrate quattro metodi di callback, due per la gestione del tocco di una superficie (touch e untouch) e due per la pressione del pulsante presente sul phantom (pressed e released).
A questo punto per ogni ciclo grafico viene interrogato il sistema aptico per verificare la presenza di eventi, di cui parleremo poco più avanti nella trattazione; dopodichè viene ridisegnata la mesh all'interno di un costrutto definito dalla libreria aptica, costrutto che permette di intercettare tutte le chiamate successive alle routines OpenGL e di tradurle in informazioni aptiche, da registrare così nel collision thread aptico. Quest'ultimo thread permette di renderizzare apticamente la forma dell'oggetto.
Pultroppo questo passaggio va a scapito delle prestazioni, infatti è necessario ridisegnare la mesh per tener conto anche delle superfici non visualizzate dal sistema grafico. Esiste anche un metodo più veloce che permette di renderizzare sia le proprietà grafiche che aptiche del sistema in un solo passaggio, tale sistema però non permette di renderizzare apticamente le facce nascoste

della mesh.
Ovviamente la mesh viene disegnata apticamente secondo le opzioni selezionate dall'utente.
Passiamo ora a definire a cosa servono le funzioni di callback di cui si accennava sopra. L'evento tocco viene utilizzato per attivare un ulteriore funzione di callback chiamata di contatto che viene attivata ogni volta che il cursore si muove sulla superficie, l'evento untouch invece cancella questa funzione di callback dall'ordine di esecuzione. Alla funzione contatto sono delegati due compiti: il primo è attivare un evento sonoro quando il cursore si muove sulla superficie, il secondo di registrare le forze esercitate dall'utente nel sistema dinamico. Nello specifico l'effetto sonoro attivato rispecchia il punto in cui sta avvenendo il contatto con la superficie, mentre la forza esercitata dall'utente è l'opposto della forza che il phantom trasmette allo stesso, in tal modo conoscendo la forza erogata dal phantom conosco la pressione che l'oggetto riceve.
Il pulsante del phantom viene utilizzato per introdurre una nuova funzionalità al sistema: la possibilità di rototraslare l'oggetto tentedo in considerazione la posizione e la rotazione dei 6 gradi di libertà del phantom. Nello specifico alla pressione del pulsante viene modificata la renderizzazione aptica e la mano dell'utente viene attratta al centro della mesh, una volta raggiunto tale punto si aggancia la mesh, evento segnalato dal sintetizzatore vocale, dopodiché a ogni movimento dell'end effector del phantom viene fatto corrispondere un movimento della mesh, sia esso una traslazione o una rotazione. Infine quando si rilascia il pulsante l'oggetto viene sganciato e si ritorna allo stato iniziale (ad eccezione della rototraslazione finale che viene mantenuta).

A.2.5 Sistema sonoro

Come già accennato nel corso della trattazione sono presenti diverse tipologie di stimoli sonori nel sistema, che si caratterizzano in:

- suoni d'interfaccia, che corrispondono alla pressione dei pulsanti presenti nella gui generata attraverso GLUI. Vi è un suono diverso per ogni tipologia di pulsante, quali checkbox, slider o radiobutton
- messaggi vocali, identificano la lettura di stringhe da parte del sintetizzatore vocale. Tali messaggi vengono attivati in diverse occasioni:
  - all'apertura del file viene pronunciato il nome dello stesso;
  - in alcune opzioni viene letto l'effetto della selezione/ deselezione ad esempio in base al fatto che venga o meno selezionata l'opzione

perspectiveView viene letta dal sintetizzatore vocale *Perspective View* o *Orthogonal View*.

- al momento dell'aggancio della mesh attraverso lo strumento aptico, viene pronunciata la parola *caught*.

- infine ultima modalità di interazione sonora è rappresentata dal suono spaziale 3d al momento del contatto con la superficie. Tale suono si basa sulla mappatura delle tre dimensioni spaziali su equivalenti dimensioni sonore:

  - Asse X contraddistinto dall'effetto stereo del suono.
  - Asse Y contraddistinto dalla frequenza del suono.
  - Asse Z contraddistinto dalla guadagno dell'effetto sonoro.

Quest'ultima modalità di utilizzo del suono viene chiamata acusmentria.
Dal punto di vista tecnico prima di tutto viene inizializzato il sistema sonoro, dopodichè si caricano in memoria i vari suoni, o si generano se necessario, e si associano a delle sorgenti nello spazio. Si colloca infine nello spazio l'ascoltatore, nel nostro caso nella stessa posizione della camera. Così il sistema è pronto per eseguire i suoni quando necessario.
I suoni effettivamente renderizzati sono:

- Click: identificato da un wave viene attivato quando si attiva un checkbutton;
- Unclick: identificato da un wave viene attivato quando si disattiva un checkbutton;
- Select: identificato da un wave viene attivato quando si seleziona un'opzione da un radiogroup;
- Wheel: sempre identificato da un wave, attivato quando si agisce su uno slider;
- Button: altro wave che viene attivato quando si preme un button;
- Contact: suono generato lato codice che indica il suono spaziale durante il contatto con la superficie.

Il sistema audio rappresenta quindi uno strumento per enfatizzare avvenimenti che altrimenti rischierebero di passare in secondo piano.

A.2.6 Simulazione dinamica

Il sistema dinamico, come già indicato, permette di avere un'interazione fisica con l'oggetto virtuale. La simulazione permette di usare il phantom come uno strumento per spingere letteralmente l'oggetto renderizzato.
Per avviare la simulazione innanzitutto si crea un nuovo contesto fisico in cui si inserisce il corrispettivo fisico dell'oggetto renderizzato che per semplicità viene assimilato ad un cubo. Al cubo, centrato nell'origine viene innanzitutto assegnata una massa ed un momento, dopodiché si assegna la rototraslazione corrispondente all'oggetto visualizzato. A questo punto quando l'oggetto viene toccato dallo strumento aptico si applica una forza che fa muovere la mesh secondo le modalità descritte precedentemente.
Il calcolo effettivo della dinamica dell'oggetto viene compiuto all'interno del ciclo grafico prima di disegnare la mesh con le OpenGL. Le operazioni da svolgere dal punto di vista dinamico sono tre:

- controllare se ci sono stati cambiamenti esterni della rototraslazione dell'oggetto, se si applicarli.
- calcolare quanti step di calcolo integrale è necessario eseguire (dipende dal frame rate) ed eseguire effettivamente i calcoli che fanno evolvere dinamicamente il sistema.
- applicare le modifiche della simulazione dinamica all'oggetto, in modo da apportare le modifiche all'intero sistema.

Da notare che il metodo di visualizzazione, ortogonale o prospettico, incide sulla simulazione dinamica, in quanto nel caso ortogonale, l'asse z del centro dell'oggetto è vincolato al valore 0, in modo che l'oggetto possa traslare solo sulle direzioni x e y, nella modalità prospettica invece si tiene conto di tutte e tre le dimensioni spaziali. Inoltre per non perdere la mesh, è stato definito un sistema per cui una volta che vengano superati i limiti del viewing volume, l'oggetto rientri in scena dalla parte opposta.
Questi piccoli accorgimenti, uniti all'utilizzo di ODE permettono di dare all'utente l'illusione di spingere effettivamente l'oggetto e di interagire dinamicamente con esso.

A.3 Conclusioni e sviluppi futuri

Il sistema rappresenta un buono spunto per studiare meglio l'interazione tra utente e oggetti aptici. É infatti emerso durante il lavoro di tesi il bisogno di

comprendere chiaramente con che meccanismi percettivi l'utente si approccia ad uno strumento come il phantom. Il tool creato rappresenta quindi un giocattolo con cui poter sperimentare vari livelli di interazione. Ciononostante può essere un primo passo per lo sviluppo di applicazioni specifiche per non vedenti, in grado di permettere loro di interfacciarsi a elementi virtuali altrimenti irraggiungibili o comunque per lo sviluppo di applicazioni in realtà virtuale con un alto grado di immersività.

Riflettendo sugli sviluppi futuri è possibile pensare di estendere il tool in modo che sia in grado di importare anche i materiali delle mesh, e di caricare più mesh contemporanemente, in tal modo si potrebbero creare dei veri e propri ambienti virtuali interattivi apticamente. Inoltre si potrebbe pensare di assegnare a suddette mesh dei materiali aptici predefiniti, in modo da salvarli e reimportarli a piacimento. Tutto questo renderebbe l'esperienza virtuale effettivamente suggestiva permettendo ad esempio di sentire la differenza, sia dal punto di vista grafico che aptico tra una palla di gomma ed una pietra.

Concludo con il dire che sono certo che l'interfacciamento aptico lato desktop sia il futuro, sebbene probabilmente non attraverso uno strumento come il phantom. Perciò è utile e necessario portare la ricerca in quella direzione, basti pensare al successo che stanno avendo ultimamente gli schermi touchscreen. L'esperienza in realtà virtuale è estremamente condizionata dai sensi coinvolti nell'interazione e il tatto unito alla kinestetica e alla propriocezione sono quanto di più potente abbiamo per distinguere realtà e finzione. Una volta che riusciremo a simulare correttamente tali aspetti, gli ambienti virtuali diventeranno davvero sempre più reali.