

Università degli Studi di Verona

FACOLTÀ DI SCIENZE
Corso di Laurea in Ingegneria Informatica

TESI DI LAUREA MAGISTRALE

**SIMULAZIONE FISICA IN TEMPO REALE DI CORPI
DEFORMABILI PER LA NAVIGAZIONE CHIRURGICA
INTRA-OPERATIVA**

Candidato:
Enrico Magnabosco
Matricola VR403851

Relatore:
Dott. Diego Dall'Alba

Anno Accademico 2017-2018

Sommario

La moderna pratica chirurgica si sta evolvendo verso procedure minimamente invasive che forniscono significativi vantaggi per il paziente ma che richiedono strumenti e competenze cliniche sempre più avanzate. Una delle procedure minimamente invasive più utilizzata è biopsia percutanea in cui uno strumento simile a un grosso ago viene inserito, attraverso la pelle del paziente, in corrispondenza di un'area patologica che si vuole analizzare (per verificarne l'effettiva malignità) oppure curare mediante tecniche di ablazione chirurgica. Questa tipologia di procedure è solitamente guidata da immagini mediche (l'operatore non può vedere attraverso la pelle del paziente) e questo rende ulteriormente complessa questa procedura. Nel corso della chirurgia minimamente invasiva le strutture anatomiche di interesse subiscono deformazioni e movimenti, per esempio dovuti all'interazione con gli strumenti chirurgici o con le sonde utilizzate per acquisire le immagini mediche. In questa tesi considereremo come caso clinico rappresentativo la biopsia percutanea eseguita sotto guida ecografica.

Riuscire a prevedere le deformazioni e movimenti può essere di grande importanza per migliorare l'esecuzione e l'esito di questa tipologia di interventi. In questo contesto la simulazione fisica in tempo reale è un argomento di ricerca molto attivo e numerosi metodi di simulazione sono stati applicati al caso clinico, come ad esempio modelli FEM, MSM e meshless. Anche se questi modelli forniscono simulazioni fisiche realistiche spesso non ottengono il realismo richiesto a causa dei parametri fisici, che solitamente, sono difficili da stimare o dall'onerosa richiesta computazionale (spesso non compatibile con le tempistiche imposte dalla pratica chirurgica).

Lo scopo di questo lavoro di tesi è lo sviluppo di un simulatore in tempo reale basato su un approccio position based dynamics (PDB) originalmente introdotto per il mercato videoludico. In particolare in questa tesi abbiamo

utilizzato l'implementazione PDB fornita delle librerie NVIDIA Flex abbinata al motore grafico Unreal Engine per la visualizzazione. Anche se questi strumenti non sono pensati specificatamente per simulazioni di strutture deformabili a scopo medico, con l'integrazione di specifiche librerie e opportuni adattamenti, sono capaci di simulare in maniera credibile le deformazioni che subiscono le strutture anatomiche durante una procedura percutanea. Il software sviluppato permette la comunicazione con dispositivi medici attraverso interfaccia di rete utilizzando il protocollo standard OpenIGTLink. In questo modo è possibile ricevere le informazioni da dispositivi clinici biomedicali (sistemi di rilevamento della posizione tridimensionale e dispositivi di imaging ecografico) in tempo reale e quindi poter aggiornare la scena simulata in modo interattivo, valutando le interazioni tra oggetti rigidi e strutture deformabili.

La validazione sperimentale del sistema sviluppato si basa sul confronto tra immagini ecografiche estratte dall'ambiente simulato con quelle ottenute dal sistema ecografico nella scansione di un modello sintetico. Questa analisi sperimentale permette di valutare il realismo della simulazione e come i parametri principali influenzino il risultato finale. Sono stati anche eseguiti esperimenti con modelli anatomici realistici per avere una valutazione preliminare delle prestazioni del metodo di simulazione in condizioni più realistiche.

I risultati dimostrano di ridurre l'errore di localizzazione del target sotto i 2mm. Il lavoro di tesi presentato può contribuire a migliorare il risultato finale della biopsia in termini di precisione e accuratezza, grazie a una simulazione delle deformazioni e dei movimenti dell'area patologia di interesse.

Indice

1	Implementazione	5
1.1	Introduzione	5
1.2	Scena simulata e comunicazione	6
1.3	Set-up sperimentale	8
2	Risultati e discussione	13
2.1	Considerazioni	13
2.2	Parametro particle spacing	14
2.3	Parametro cluster stiffness	16
2.4	Parametro cluster spacing	17
2.5	Discussione	18
	Bibliografia	21

Capitolo 1

Implementazione

1.1 Introduzione

Il sistema dovrà essere in grado di simulare le deformazioni che subiscono le strutture anatomiche durante una procedura di biopsia percutanea sotto guida ecografica. Il software implementato si basa sul motore fisico Unreal Engine 4 (UE4) integrato di librerie NVIDIA Flex per la simulazione di strutture anatomiche deformabili. L'idea di Flex è quella di poter rappresentare qualsiasi tipo di oggetto sotto forma di particelle interconnesse da vincoli. Per la comunicazione con i dispositivi biomedicali (utilizzati nel corso della procedura) si è utilizzato un protocollo di rete standard OpenIGTLink, integrandolo nell'ambiente di sviluppo scelto mediante un opportuna classe che sfrutta i socket TCP messi a disposizione da UE4.

Inizialmente la scena di simulazione è composta da un 'phantom di calibrazione' esso ha lo scopo di identificare i parametri ottimali sui quali agire per poter rendere la simulazione realistica. Una volta che i valori ottimali saranno stati impostati la scena potrà essere estesa con un phantom anatomico che replica il seno (come si può vedere in figura 1.1) con al suo interno dei noduli tumorali.

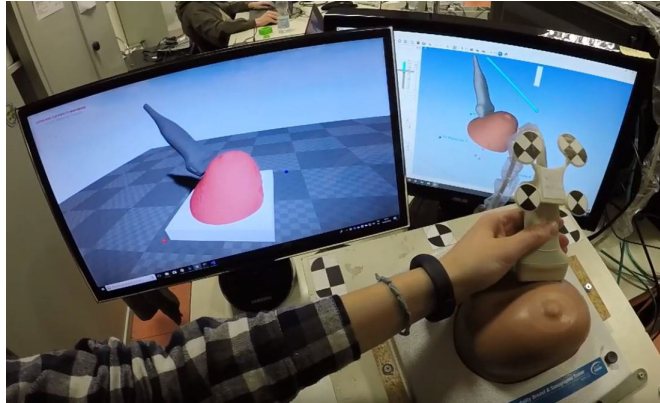


Figura 1.1: Nel monitor principale a sinistra è possibile vedere l'ambiente simulato in Flex. In primo piano possiamo vedere la sonda ecografica con il marker utilizzato dal sistema di misurazione ottico per il rilevamento della posizione tridimensionale e il modello anatomico sintetico del seno.

1.2 Scena simulata e comunicazione

La scena di simulazione replica in modo fedele l'allestimento sperimentale reale, è composta da un 'phantom di calibrazione' con al suo interno dei noduli che presentano una consistenza diversa rispetto al materiale circostante (come in figura 1.2). e da una sonda che porterà delle deformazioni di varie profondità. La scena includerà anche il modello 3d di una sonda ecografica, che verrà utilizzata per l'acquisizione di immagini del phantom e che introdurrà delle deformazioni dovute al protocollo di acquisizione.

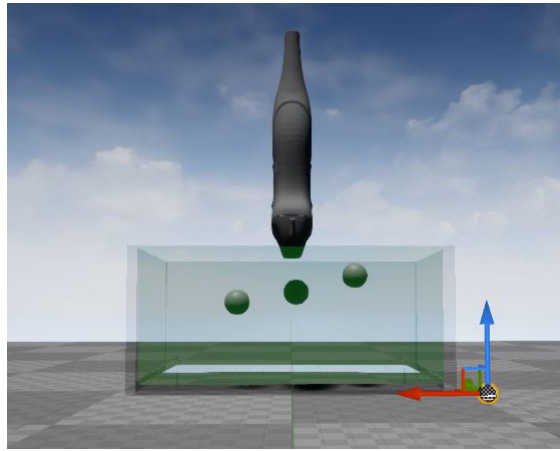


Figura 1.2: Scena simulata in Unreal engine 4

In seguito alla creazione della simulazione si è dovuto creare un metodo di comunicazione tra la scena reale e quella simulata, questa comunicazione viene utilizzata per mappare i movimenti della sonda reale all'interno della simulazione. La sonda presenta dei marker che verranno identificati in modo univoco da un sistema ottico di misurazione della posa tridimensionale (con riferimento alla figura 1.3) e serviranno ad estrarre esattamente la posizione e rotazione assunta dalla sonda in ogni momento. In seguito all'acquisizione della posa (rappresentata in coordinate omogenee con la composizione di una matrice di roto-traslazione composta da una matrice quadrata 3×3 e da un vettore 3×1) essa dovrà essere inviata alla scena simulata

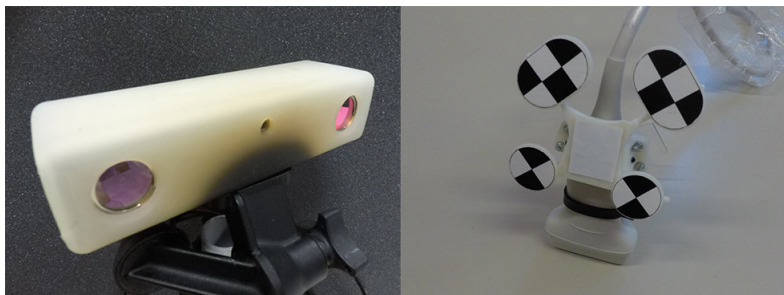


Figura 1.3: A sinistra è presenta la telecamera stereoscopica alla base del sistema di tracking, a destra i marker fissati alla sonda ecografica ed utilizzati per il rilevamento della sua posa.

Dopo l'acquisizione dei questi dati relativi alla posa, essi devono essere trasmessi alla scena simulata in tempo reale sotto forma di pacchetti che integrano sia la matrice di rotazione che il vettore che identifica la traslazione. Oltre queste due informazioni sono presenti ulteriori dettagli che fanno riferimento al tipo di oggetto mappato (questo potrà tornare utile in una possibile estensione sul lavoro in quanto il simulatore può mappare oggetti multipli e non la singola sonda ecografica). Per la comunicazione di queste informazioni spaziali si è deciso di adottare un protocollo di rete standard ossia OpenIGTLink [62].

1.3 Set-up sperimentale

Il set-up che abbiamo utilizzato si compone di un sistema di acquisizione delle immagini ad ultrasuoni (US) e la corrispettiva parte simulata creata in Unreal Engine 4. Per acquisire le immagini US è stata utilizzato una sonda ecografica USB prodotta da Telemed, essa offre un'immagine di alta qualità fino ad una profondità massima di 70mm.

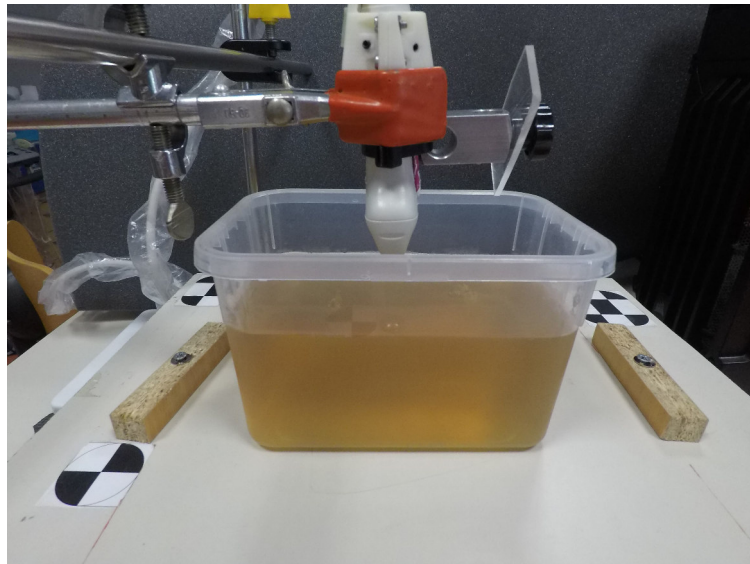


Figura 1.4: Setup sperimentale con il 'phantom di calibrazione' e il sistema di posizionamento della sonda utilizzato

Il sistema utilizzato per l'acquisizione, come si può notare dall'immagine 1.4, è composto dalla sonda montata su di un barra millimetrata e da un phantom realizzato con un gel che simula la consistenza del tessuto (in modo simile a [6]).



Figura 1.5: Sfere di simulazione

All'interno del phantom, come da immagine 1.5, sono state inglobate tre sfere con un diametro di 18mm che rappresentano la consistenza più dura dei noduli rispetto a quella del gel (questo replica il contesto clinico di riferimento del lavoro, infatti i tumori al seno si presentano spesso sotto forma di noduli con consistenza più dura del resto del tessuto). Esse sono state disposte a tre altezze diverse (47mm , 53mm , 63mm) per poter controllare la deformazione che subiscono sotto una specifica pressione. Con riferimento alla figura 1.4, la sonda viene posta al centro di ogni sfera e vengono acquisite cinque diverse immagini: la prima con sonda a riposo, ovvero appoggiata sulla superficie del gel per ottenere un'immagine senza introdurre deformazioni nel tessuto. Successivamente viene spostato il riferimento sulla barra e quindi la sonda scende di 5mm , 10mm , 15mm e 20mm . Una volta acquisita ecografica reale deve essere segmentata (come in figura 1.6) per identificare con precisione il nodulo e quindi poterla confrontare con l'immagine simulata.

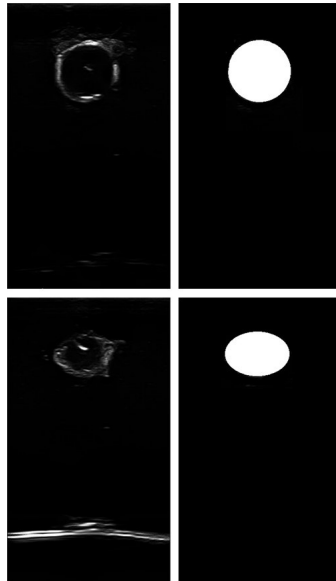


Figura 1.6: Nella colonna di sinistra le immagini acquisite dalla sonda e a destra le stessa immagini segmentate

Per poter creare la scena di simulazione è stato realizzato, attraverso il programma 3D Builder, il modello 3D del phantom composto sia dal gel che dalle sfere che simulano i noduli rispettando le misure reali. Il modello della sonda era direttamente disponibile dal sito del produttore e quindi non si è reso necessario ricostruirlo, è bastato convertirlo in un formato supportato e scolarlo in modo adeguato.

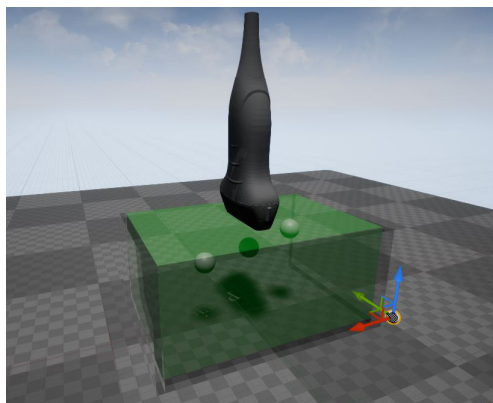


Figura 1.7: Scena di simulazione in Unreal Engine 4

Nella figura 1.7 si può vedere il set-up della scena, esso rispetta le misure esatte per ogni componente. Le acquisizioni fatte in UE4 sono le stesse di quelle sul phantom reale rispettando i cinque spostamenti della sonda (riposo, $5mm$, $10mm$, $15mm$ e $20mm$). Per ogni profondità vengono acquisite le immagini derivanti dall'esperimento reale e le immagini che arrivano dalla simulazione così da poterle confrontare (dopo averle segmentate). Per il confronto si utilizza la distanza euclidea tra il centro dell'ellissi che identifica il nodulo simulato e quello reale, tale distanza sarà espressa in pixel. Oltre a questo confronto, è stata usata la funzione Structural Similarity Index (SSIM) [64], che permette di calcolare la similarità tra immagini e quindi poter avere una misura accurata di quanta differenza ci sia tra immagine reale e simulata.

Una volta trovati i parametri ottimali per la simulazione si è proceduto ad applicarli al modello del seno come quello in figura 1.8 (per maggiori dettagli fare riferimento a [5]). Il modello tridimensionale con i noduli tumorali è stato ottenuto mediante la segmentazione dei dati pre-operativi (TAC) del phantom, in modo da replicare in modo accurato un processo simile a quello che avviene nella normale pratica clinica.

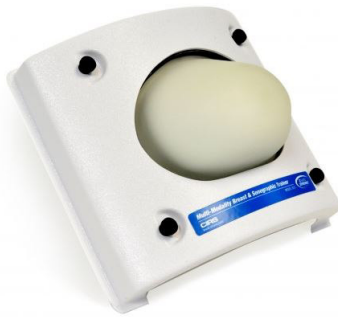


Figura 1.8: Modello del phantom utilizzato, Cirs 073

Una volta convertito nel formato adatto alla scena simulata (dal formato stl al formato fbx) esso è stato reso deformabile attraverso l'applicazione della libreria Flex ed integrato nella scena come in figura 1.9, nei capitoli successivi verranno illustrati i parametri che più hanno influenzato il modello. Allo

scopo di non rendere troppo complesso il modello proposto si è assunto un modello isotropo e omogeneo per il seno, ovvero il parametro di rigidità ottimale che è stato identificato per il phantom di calibrazione è poi stato applicato al seno (e anche alle sue inclusioni).

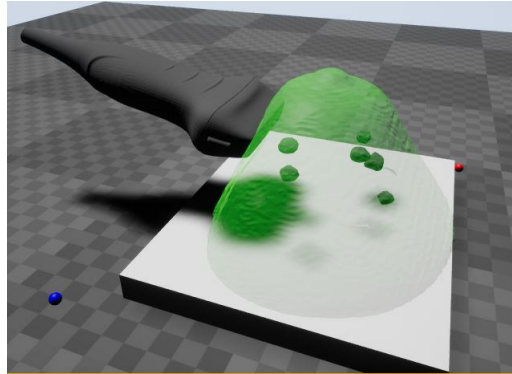


Figura 1.9: Sistema a particella applicato al phnatom del seno

Capitolo 2

Risultati e discussione

Per estrarre i risultati che seguono sono state adottati due approcci differenti, uno basato sulle immagini ed un'altra basandosi sulle coordinate (espresse in pixel) nel centro del nodulo. Le immagini segmentate provenienti dalla simulazione vengono confrontate con la controparte reale della sonda ad ultrasuoni, anch'essa segmentata. Tutte le tabelle che seguono esprimono l'errore in pixel.

2.1 Considerazioni

Il grafico in figura 2.1 vuole rappresentare l'andamento che il nodulo ha in base alla deformazione che subisce. Per il calcolo si è inizialmente segmentato il nodulo e successivamente trovato il centro in coordinate pixel. Una volta calcolato il centro per ogni deformazione ($0mm$, $5mm$, $10mm$, $15mm$, $20mm$) è stata calcolata la distanza euclidea rispetto alle coordinate del nodulo a riposo, ossia della deformazione a $0mm$.

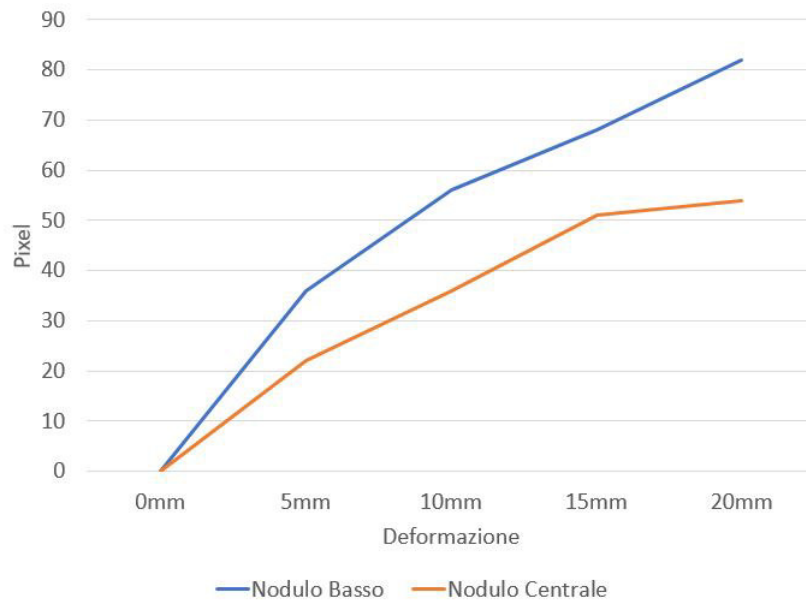


Figura 2.1: Distanza del nodulo rispetto alla posizione a riposo per le diverse deformazioni. Nell'asse delle ordinate è presente l'errore espresso in pixel mentre nelle ascisse viene specificata la deformazione

2.2 Parametro particle spacing

Nella tabella 2.1 vengono rappresentati gli errori in pixel sulla profondità della deformazione in relazione al valore del parametro 'Particle spacing'. Questo parametro identifica la distanza tra particelle vicine, viene preso in considerazione come parametro fondamentale di simulazione in quanto fa variare il numero di particelle con cui l'oggetto sarà rappresentato. È stato fatto variare in un range che va da 5 a 12, questi due valori sono stati scelti in base alle prove effettuate per identificare l'intervallo di valori che influenzasse in modo significativo le prestazioni della simulazione.

Come si può notare per ogni deformazione viene evidenziato il valore del parametro che minimizza l'errore. Per ogni parametro poi, viene calcolata la media degli errori e dunque evidenziato quel parametro che in media ha l'errore più basso indipendentemente dalla deformazione. Il parametro con

errore minore è risultato essere 10.

Particle Spacing	20mm	15mm	10mm	5mm	0mm	Media
Parametro 5.0	13	17	23	19	1	14,6
Parametro 6.0	15	17	24	20	0	15,2
Parametro 7.0	16	18	25	23	1	16,6
Parametro 8.0	2	9	18	15	3	9,4
Parametro 9.0	7	1	11	14	4	7,4
Parametro 10.0	9	2	9	9	5	6,8
Parametro 11.0	19	7	2	8	5	8,2
Parametro 12.0	32	18	4	2	6	12,4

Tabella 2.1: Tabella sugli errori in pixel del parametro Particle Spacing, in giallo sono evidenziati gli errori minori per ciascuna deformazione mentre in rosso il minor errore complessivo

Il grafico in figura 2.2 aiuta a identificare il parametro migliore basandosi anche sulla deviazione standard, come si può notare il parametro 10.0 oltre ad avere l'errore media minore ha anche la deviazione standard minore.

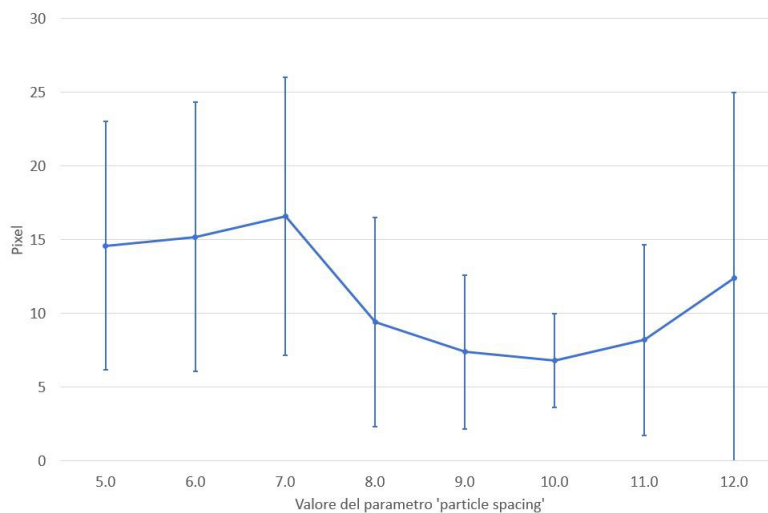


Figura 2.2: Grafico con riferimento alla tabella 2.1

2.3 Parametro cluster stiffness

Il parametro 'Cluster stiffness' rappresenta il valore di rigidità che vincola le particelle tra di loro all'interno dello stesso cluster. Nella nostra simulazione tale parametro non modifica in modo significativo la simulazione in quanto le particelle istanziate in scena non sono molte. Questo si nota anche dagli errori in tabella 2.2 che risultano bassi se comparati con quelli dei parametri precedentemente analizzati. Il parametro va visto come una sorta di ottimizzazione per la simulazione che, anche se non di molto, modifica la scena e porta a ridurre l'errore generale.

Cluster stiffness	20mm	15mm	10mm	5mm	0mm	Media
Parametro 0,1	8	0	11	14	5	7,6
Parametro 0,3	10	2	8	12	4	7,2
Parametro 0,5	9	1	9	11	3	6,6
Parametro 0,7	8	1	9	10	4	6,4
Parametro 0,9	11	3	11	10	5	8

Tabella 2.2: Tabella sugli errori in pixel del parametro Volume sampling, in giallo sono evidenziati gli errori minori per ciascuna deformazione mentre in rosso il minor errore complessivo

Il grafico 2.3 mostra meglio come questo parametro modifica poco la simulazione, la deviazione standard è molto simile per ogni parametro e per questo motivo si è scelto il valore 0,7 basandosi sulla media dell'errore.

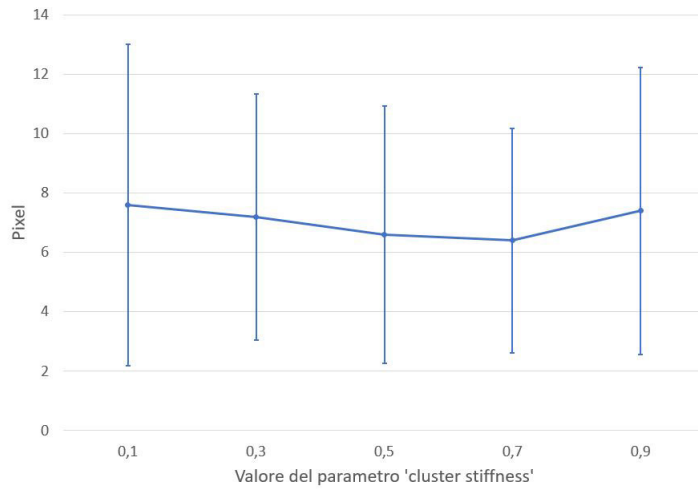


Figura 2.3: Grafico con riferimento alla tabella 2.2

2.4 Parametro cluster spacing

Questo parametro identifica lo spazio che divide un cluster dai vicini. Come per 'Particle spacing' più basso sarà questo parametro e più cluster saranno presenti nell'oggetto, cluster sovrapposti si influenzeranno a vicenda come spiegato nei capitoli precedenti. Dalla tabella 2.3 si estrae, come per i parametri precedentemente analizzati, il parametro che minimizza l'errore medio.

Cluster spacing	20mm	15mm	10mm	5mm	0mm	Media
Parametro 8	13	12	7	6	5	8,6
Parametro 9	12	4	5	4	4	5,8
Parametro 10	9	2	6	3	3	4,6
Parametro 11	23	14	6	2	4	9,8
Parametro 12	30	15	9	4	5	12,6

Tabella 2.3: Tabella sugli errori in pixel del parametro Cluster spacing, in giallo sono evidenziati gli errori minori per ciascuna deformazione mentre in rosso il minor errore complessivo

Con riferimento al grafico in tabella 2.4 il parametro 10 minimizza sia l'errore medio e presenta anche la minor deviazione standard, esso sicuramente rappresenta il miglior valore per il parametro 'Cluster spacing' per la nostra simulazione.

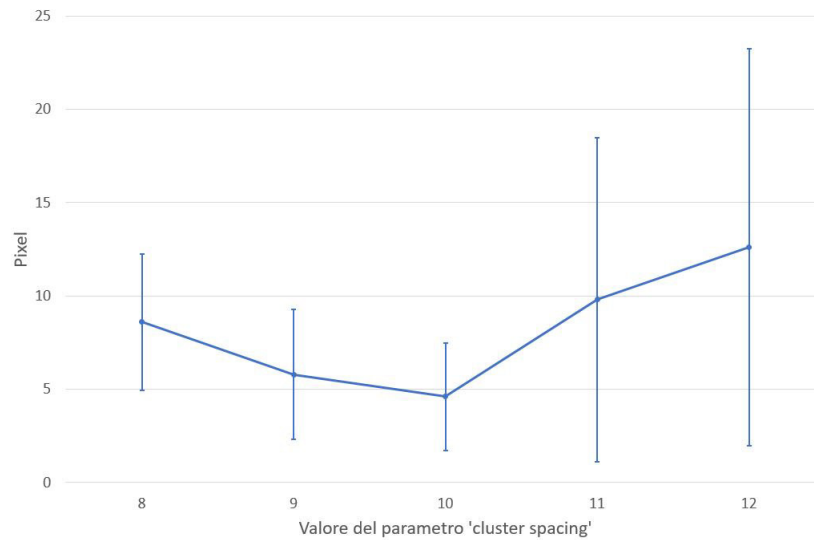


Figura 2.4: Grafico con riferimento alla tabella 2.3

2.5 Discussione

Una volta trovati i parametri ottimali si è fatta una simulazione finale, impostando i valori ottimali per ogni parametro, e sono stati estratti i seguenti errori:

- deformazione $0mm$: errore 3px ;
- deformazione $5mm$: errore 4px ;
- deformazione $10mm$: errore 6px ;
- deformazione $15mm$: errore 11px ;
- deformazione $20mm$: errore 16px .

È stata fatta anche la prova sulla similarità che la funzione SSIM restituisce confrontando le immagini provenienti dalla simulazione con le immagini provenienti dall'acquisizione reale definendo i parametri ottimali, tale funzione restituisce un valore del 93%, inteso come valore medio sulle 5 deformazioni. Essendo basata sulle immagini, questa funzione, è molto più soggetta a errori di segmentazioni rispetto al semplice calcolo della distanza euclidea sul centro dei noduli. I parametri ottimali ottenuti sono paragonabili a quelli che possiamo trovare in letteratura, con riferimento a [14], soprattutto il parametro che identifica la rigidità con la quale simulare il tessuto è risultato lo stesso. Come si nota la correlazione tra le immagini simulate e quelle reali riporta un errore che aumenta all'aumentare della deformazione, questo è un dato atteso in quanto la simulazione risulta più critica quanto più l'oggetto sarà deformato. Gli errori ottenuti si potrebbero abbassare ulteriormente agendo maggiorante sui parametri di simulazione ma questo renderebbe la scena meno fluida. L'obiettivo è quello di avere una simulazione in tempo reale quindi mantenere la scena ad un frame rate elevato risulta un vincolo estremamente importante da rispettare.

Una volta ottenuti i parametri ideali di simulazioni essi sono stati applicati al phantom anatomico del seno. La simulazione reagisce in modo verosimile e i risultati attesi sono stati rispettati anche se risulta molto difficile quantificare in modo preciso la bontà sull'effettiva deformazione (figura 2.5). Un problema è dovuto principalmente a piccoli errori di calibrazione; è stato necessaria infatti stimare una prima calibrazione per poter posizionare il phantom correttamente all'interno della scena e una seconda per poter mappare correttamente i movimenti della sonda rispetto alla posizione del phantom stesso. Gli errori introdotti nella stima di queste calibrazioni (per i dettagli relativi a questa stima si rimanda a [42]) anche se di piccola entità (in media nell'ordine di 1mm e sempre inferiori a 2mm) possono in alcuni casi rendere imprecisa e poco realistica la simulazione. Il phantom del seno presenta noduli di forma sferica/elittica aventi raggio di pochi millimetri e quindi l'accumulo di questi errori può rendere meno accurata l'analisi

dell'influenza dei parametri di calibrazione adottati.



Figura 2.5: Nello schermo a sinistra simulazione del phantom anatomico del seno

Rimangono inoltre dei dubbi relativi a dettagli implementativi nell'integrazione tra UE4 e la libreria Flex. Nonostante la scelta di lavorare con UE4 e Flex abbia permesso una rapida creazione dell'ambiente di simulazione virtuale, ha anche parzialmente ridotto la possibilità di agire sui parametri dei modelli PBD sottostanti. Per esempio, alcuni parametri che in lavori simili ([14] [15]) modificano notevolmente la simulazione, nella nostra scena non risultano così incisivi. I parametri 'shape friction' e 'particle friction' a livello teorico dovrebbero avere una grande influenza, in quanto agiscono sull'attrito tra le particelle e con gli oggetti adiacenti, ma nella nostra simulazione non introducono variazioni significative. Un'ulteriore osservazione va fatta sul parametro 'particle radius' che identifica la grandezza delle particelle, come detto in precedenza, nei lavori citati tale parametro influenza molto la simulazione mentre nel nostro caso agisce in modo trascurabile. Questo è un grosso limite che rende molto difficile creare una simulazione fedele e che ha portato a dover agire su altri parametri per ovviare a tali problemi.

Bibliografia

- [1] Libreria flex. <https://github.com/NvPhysX/UnrealEngine/tree/Flex>, 2008.
- [2] Nvidia compute unified device architecture(cuda)framework. www.nvidia.com/object/cudahome.html, 2008.
- [3] Nvidia physx library. <http://www.nvidia.com/object/nvidiaphysx.html>, 2008.
- [4] Documento flex. <https://docs.nvidia.com/gameworks/content/artisttools/Flex/index.html>, 2015.
- [5] Cirs073. [http://www.cirsinc.com/file/Products/073/073%20DS%20110617\(3\).pdf](http://www.cirsinc.com/file/Products/073/073%20DS%20110617(3).pdf), 2017.
- [6] Richard Amini, Jeffrey Z Kartchner, Lori A Stolz, David Biffar, Allan J Hamilton, and Srikar Adhikari. A novel and inexpensive ballistic gel phantom for ultrasound training. *World journal of emergency medicine*, 6(3):225, 2015.
- [7] David Baraff and Andrew Witkin. Large steps in cloth simulation. pages 43–54, 1998.
- [8] Jan Bender, Dan Koschier, Patrick Charrier, and Daniel Weber. Position-based simulation of continuous materials. *Computers & Graphics*, 44:1–10, 2014.

- [9] Jan Bender, Matthias Müller, and Miles Macklin. Position-based simulation methods in computer graphics. 2015.
- [10] Jan Bender, Matthias Müller, Miguel A Otaduy, and Matthias Teschner. Position-based methods for the simulation of solid objects in computer graphics. pages 1–22, 2013.
- [11] Iago Berndt, Rafael Torchelsen, and Anderson Maciel. Efficient surgical cutting with position-based dynamics. volume 38, pages 24–31. IEEE, 2017.
- [12] Gérald Bianchi, Matthias Harders, and Gábor Székely. Mesh topology identification for mass-spring models. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 50–58. Springer, 2003.
- [13] David E Breen, Donald H House, and Michael J Wozny. Predicting the drape of woven cloth using interacting particles. pages 365–372, 1994.
- [14] Mafalda Camara, Erik Mayer, Ara Darzi, and Philip Pratt. Soft tissue deformation for surgical simulation: a position-based dynamics approach. *International journal of computer assisted radiology and surgery*, 11(6):919–928, 2016.
- [15] Mafalda Camara, Erik Mayer, Ara Darzi, and Philip Pratt. Simulation of patient-specific deformable ultrasound imaging in real time. pages 11–18, 2017.
- [16] John E Chadwick, David R Haumann, and Richard E Parent. Layered construction for deformable animated characters. 23(3):243–252, 1989.
- [17] Yan Chen, Qing-Hong Zhu, Arie Kaufman, and Shigeru Muraki. Physically-based animation of volumetric objects. pages 154–160, 1998.

- [18] Kup-Sze Choi. Interactive cutting of deformable objects using force propagation approach and digital design analogy. volume 30, pages 233–243. Elsevier, 2006.
- [19] Stéphane Cotin, Hervé Delingette, and Nicholas Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *IEEE transactions on Visualization and Computer Graphics*, 5(1):62–73, 1999.
- [20] Stéphane Cotin, Hervé Delingette, and Nicholas Ayache. A hybrid elastic model for real-time cutting, deformations, and force feedback for surgery training and simulation. volume 16, pages 437–452. Springer, 2000.
- [21] Stéphane Cotin, Hervé Delingette, M Bro-Nielsen, Nicholas Ayache, Jean-Marie Clément, V Tasseti, and Jacques Marescaux. Geometric and physical representations for a simulator of hepatic surgery. *Studies in health technology and informatics*, 29:139–151, 1996.
- [22] Steven A Cover, Norberto F Ezquerra, James F O’Brien, Richard Rowe, Thomas Gadacz, and Ellen Palm. Interactively deformable models for surgery simulation. *IEEE Computer Graphics and Applications*, 13(6):68–75, 1993.
- [23] Jessica R Crouch, John C Merriam, and Earl R Crouch. Finite element model of cornea deformation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 591–598. Springer, 2005.
- [24] Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani, and Alan H Barr. Dynamic real-time deformations using space & time adaptive sampling. pages 31–36, 2001.
- [25] Mathieu Desbrun, Peter Schröder, and Alan Barr. Interactive animation of structured deformable objects. 99(5):10, 1999.

- [26] Crispin Deul, Patrick Charrier, and Jan Bender. Position-based rigid-body dynamics. *Computer Animation and Virtual Worlds*, 27(2):103–112, 2016.
- [27] Sarah Gibson, Joe Samosky, Andrew Mor, Christina Fyock, Eric Grimson, Takeo Kanade, Ron Kikinis, Hugh Lauer, Neil McKenzie, Shin Nakajima, et al. Simulating arthroscopic knee surgery using volumetric object representations, real-time volume rendering and haptic feedback. pages 367–378, 1997.
- [28] J-P Gourret, Nadia Magnenat Thalmann, and Daniel Thalmann. Simulation of object and human skin formations in a grasping task. *ACM Siggraph Computer Graphics*, 23(3):21–30, 1989.
- [29] Heinrich M Jaeger, Sidney R Nagel, and Robert P Behringer. Granular solids, liquids, and gases. *Reviews of modern physics*, 68(4):1259, 1996.
- [30] Thomas Jakobsen. Advanced character physics. 3, 2001.
- [31] Doug L James and Dinesh K Pai. Artdefo: accurate real time deformable objects. pages 65–72, 1999.
- [32] Erwin Keeve, Sabine Girod, Ron Kikinis, and Bernd Girod. Deformable modeling of facial tissue for craniofacial surgery simulation. *Computer aided surgery*, 3(5):228–238, 1998.
- [33] Blazej Kubiak, Nico Pietroni, Fabio Ganovelli, and Marco Fratarcangeli. A robust method for real-time thread simulation. In *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, pages 85–88. ACM, 2007.
- [34] UG Kuehnappel and B Neisius. Cad-based graphical computer simulation in endoscopic surgery. *Endoscopic surgery and allied technologies*, 1(3):181–184, 1993.

- [35] Uwe Kühnapfel, Hüseyin Kemâl Cakmak, and Heiko Maaß. Endoscopic surgery training using virtual reality and deformable tissue simulation. volume 24, pages 671–682. Elsevier, 2000.
- [36] LL Lian and YH Chen. Haptic surgical simulation: An application to virtual suture. volume 3, pages 203–210. Taylor & Francis, 2006.
- [37] Tiantian Liu, Adam W Bargteil, James F O’Brien, and Ladislav Kavan. Fast simulation of mass-spring systems. *ACM Transactions on Graphics (TOG)*, 32(6):214, 2013.
- [38] Annie Luciani, Stéphane Jimenez, Jean-Loup Florens, Claude Cadoz, and Olivier Raoult. Computational physics: A modeler-simulator for animated physical objects. pages 425–436, 1991.
- [39] Anderson Maciel, Tansel Halic, Zhonghua Lu, Luciana P Nedel, and Suvranu De. Using the physx engine for physics-based virtual surgery with force feedback. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 5(3):341–353, 2009.
- [40] Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. Unified particle physics for real-time applications. *ACM Transactions on Graphics (TOG)*, 33(4):153, 2014.
- [41] Mahadevappa Mahesh. Fundamentals of medical imaging. *Medical Physics*, 38(3):1735–1735, 2011.
- [42] Laurence Mercier, Thomas Langø, Frank Lindseth, and Louis D Collins. A review of calibration techniques for freehand 3-d ultrasound systems. *Ultrasound in medicine & biology*, 31(2):143–165, 2005.
- [43] P Meseure and C Chaillou. Deformable body simulation with adaptive subdivision and cuttings. 97:361–370, 1997.
- [44] Gavin SP Miller. The motion dynamics of snakes and worms. *ACM Siggraph Computer Graphics*, 22(4):169–173, 1988.

- [45] Jesper Mosegaard. Lr-spring mass model for cardiac surgical simulation. volume 98, pages 256–258, 2004.
- [46] Matthias Müller. Hierarchical position based dynamics. 2008.
- [47] Matthias Müller, Julie Dorsey, Leonard McMillan, Robert Jagnow, and Barbara Cutler. Stable real-time deformations. pages 49–54, 2002.
- [48] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.
- [49] Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross. Meshless deformations based on shape matching. 24(3):471–478, 2005.
- [50] Andrew Nealen, Matthias Müller, Richard Keiser, Eddy Boxerman, and Mark Carlson. Physically based deformable models in computer graphics. In *Computer graphics forum*, volume 25, pages 809–836. Wiley Online Library, 2006.
- [51] Han-Wen Nienhuys and A Frank van der Stappen. A surgery simulation supporting cuts and finite element deformation. In *International conference on medical image computing and computer-assisted intervention*, pages 145–152. Springer, 2001.
- [52] Han-Wen Nienhuys and A Frank van der Stappen. A delaunay approach to interactive cutting in triangulated surfaces. In *Algorithmic Foundations of Robotics V*, pages 113–129. Springer, 2004.
- [53] James F O’Brien and Jessica K Hodgins. Graphical modeling and animation of brittle fracture. pages 137–146, 1999.
- [54] Stephen M Platt and Norman I Badler. Animating facial expressions. 15(3):245–252, 1981.

- [55] David Richens, Mark Field, Shahrul Hashim, Michael Neale, and Charles Oakley. A finite element model of blunt traumatic aortic rupture. volume 25, pages 1039–1047. Elsevier Science BV, 2004.
- [56] Jos Stam. Nucleus: Towards a unified dynamics solver for computer graphics. pages 1–11, 2009.
- [57] Gilbert Strang and George J Fix. An analysis of the finite element method. 212, 1973.
- [58] Joseph Teran, Sylvia Blemker, V Hing, and Ronald Fedkiw. Finite volume methods for the simulation of skeletal muscle. pages 68–74, 2003.
- [59] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. *ACM Siggraph Computer Graphics*, 21(4):205–214, 1987.
- [60] Demetri Terzopoulos and Keith Waters. Physically-based facial modeling, analysis, and animation. *Computer Animation and Virtual Worlds*, 1(2):73–80, 1990.
- [61] Matthias Teschner, Bruno Heidelberger, Matthias Muller, and Markus Gross. A versatile and robust model for geometrically complex deformable solids. pages 312–319, 2004.
- [62] Junichi Tokuda, Gregory S Fischer, Xenophon Papademetris, Ziv Yaniv, Luis Ibanez, Patrick Cheng, Haiying Liu, Jack Blevins, Jumpei Arata, Alexandra J Golby, et al. Openigtlink: an open network protocol for image-guided therapy environment. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 5(4):423–434, 2009.
- [63] Xiaoyuan Tu and Demetri Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. pages 43–50, 1994.

- [64] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [65] Keith Waters. A muscle model for animation three-dimensional facial expression. *Acm siggraph computer graphics*, 21(4):17–24, 1987.
- [66] Keith Waters and Demetri Terzopoulos. Modelling and animating faces using scanned data. *Computer Animation and Virtual Worlds*, 2(4):123–128, 1991.
- [67] James D Westwood et al. A gpu accelerated spring mass system for surgical simulation. volume 111, page 342. IOS Press, 2005.
- [68] Andrew Witkin, Michael Gleicher, and William Welch. Interactive dynamics. 24(2), 1990.
- [69] Jason R Wollmuth, Douglas R Bree, Brian P Cupps, Marc D Krock, Benjamin J Pomerantz, Robert P Pasque, Analyn Howells, Nader Moazami, Nicholas T Kouchoukos, and Michael K Pasque. Left ventricular wall stress in patients with severe aortic insufficiency with finite element analysis. volume 82, pages 840–846. Elsevier, 2006.
- [70] Xunlei Wu, Michael S Downes, Tolga Goktekin, and Frank Tendick. Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. In *Computer Graphics Forum*, volume 20, pages 349–358. Wiley Online Library, 2001.