



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FACOLTA' DI INGEGNERIA

Corso di Laurea Magistrale in Ingegneria Meccanica

**Sviluppo di un metodo innovativo per
l'ottenimento rapido di modelli virtuali di
prodotto per applicazioni di Realtà Estesa**

**Development of an innovative method for
the rapid obtainment of virtual product
models for Extended Reality applications**

Relatore:

Prof. Marco Mandolini

Tesi di laurea di:

Paolo Senesi

Correlatore:

Prof. Alessandra Papetti

Anno Accademico 2021-2022

Ai miei genitori e alle mie sorelle, per il sostegno, il conforto, lo spazio, il tempo.

*“Se non puoi essere un pino sul monte, sii una saggina nella valle,
ma sii la migliore, piccola saggina sulla sponda del ruscello.*

Se non puoi essere un albero, sii un cespuglio.

Se non puoi essere un'autostrada, sii un sentiero.

Se non puoi essere il sole, sii una stella.

Sii sempre il meglio di ciò che sei.”

Martin Luther King

I. Ringraziamenti

Il lavoro svolto per questa tesi è il risultato di un forte slancio di innovazione che sta investendo la Facoltà di Ingegneria dell'Università Politecnica delle Marche: esso parte da una ricerca dell'incontro tra tecnica ed estetica, tra progettazione e design, tra ingegneria ed arte.

Voglio ringraziare tutti gli insegnanti che ho incontrato nel mio percorso di studi, specialmente quelli che si sono spesi per migliorare la qualità della didattica sfruttando creatività e curiosità, ed instaurando relazioni costruttive con gli studenti.

Un ringraziamento particolare è dovuto al Prof. Michele Germani in qualità di insegnante e direttore del DIISM e del gruppo di ricerca di Disegno e Metodi dell'Ingegneria Industriale: la collaborazione da cui è nata questa tesi si deve ad una delle attività proposte nei suoi corsi improntati all'innovazione e alla multidisciplinarietà.

Un ringraziamento a Ing. Barbara Lonzi, Ing. Loredana Banci e Ing. Matteo Bizzaro, referenti di Benelli Armi Spa, per la grande disponibilità ed attenzione messi a disposizione durante lo svolgimento del progetto.

Ringrazio i colleghi del corso per la condivisione dell'esperienza formativa: in particolare per il supporto reciproco che ha contribuito a semplificare i lavori didattici e per i momenti conviviali che hanno arricchito di umanità le attività di apprendimento.

Ringrazio di cuore il Prof. Marco Mandolini per la passione, la competenza, la cura verso gli studenti prima come insegnante e poi come relatore di questa tesi.

II. Sommario

I. Ringraziamenti	i
II. Sommario	ii
III. Elenco delle figure	iii
IV. Elenco delle tabelle	vi
Abstract	1
1 Introduzione	2
2 Stato dell'arte	7
2.1 Modellazione 3D: formati e rappresentazione standard di punti	7
2.2 Rappresentazioni di linee, curve, superfici nella computer grafica	8
2.3 Cenni su Blender	11
2.4 Modellazione in Blender	14
2.4.1 Superfici e shading	14
2.4.2 Editing proporzionale	15
2.5 Modificatori	15
2.5.1 Menù dei Modificatori	15
2.5.2 Subdivision Surface Modifier	17
2.5.3 Boolean Modifier	17
2.5.4 Shrinkwrap Modifier	18
2.5.5 Solidify Modifier	19
2.6 Rendering	19
2.6.1 UV Mapping e Texturing	19
2.6.2 Textures e materiali nella computer grafica 3D	21
3 Metodo	24
3.1 Il problema della topologia	24
3.2 Modificatori come soluzione di automazione	26
3.3 Utilizzo della gabbia di modellazione	28
3.4 Workflow per ottenere la geometria aggiornata con metodo Shrinkwrap	30
3.5 Integrazione metodo Shrinkwrap con Normal Baking	33
4 Casi studio	40
4.1 Preparazione e accorgimenti	40
4.1.1 Cutters	41

4.2	Analisi tempi per gli aggiornamenti di modello del componente calcio	43
4.3	Analisi di dettaglio di tempi per modelli con zigrino di calcio e astina	44
4.4	Modellazione da zero del componente carcassa	45
5	Risultati e discussione	48
5.1	Premessa	48
5.2	Tempi di ottenimento	48
5.3	Qualità dei modelli	50
5.4	Galleria	55
6	Conclusioni	58
7	Bibliografia	60

III. Elenco delle figure

Figura 1-1: Relazione tra Realtà Estesa (XR), Aumentata (AR), Virtuale (VR), Mista (MR)	2
Figura 1-2: Concetti di Metaverso e Realtà Estesa a confronto	3
Figura 2-1: Spline fisiche usate nelle costruzioni di navi	8
Figura 2-2: Differenti strategie di ottenimento di superfici NURBS da curve	10
Figura 2-3: Tipologie di superfici 3D: A: mesh quadrangolare; B: mesh quadrangolare, smoothed; C: superficie NURBS.....	10
Figura 2-4: Subdivision Surface di tipo Catmull-Clark: A: mesh non suddivisa; B: prima suddivisione; C: seconda suddivisione.....	10
Figura 2-5: Interfaccia grafica di Blender e relative aree: 1: Topbar; 2: Outliner; 3: Properties; 4: Timeline; 5: Status Bar; 6: 3D Viewport.....	12
Figura 2-6: Tre oggetti poligonali in Blender, rotazione e scaling in Object Mode	12
Figura 2-7: Edit mode, le tre modalità di selezione disponibili: Vertex, Edge, Face	12
Figura 2-8: Rotazione in Edit mode: Vertici in Vertex Mode (in alto), Bordi in Edge Mode (al centro), Facce in Face Mode (in basso)	13
Figura 2-9: Effetto dello smooth shading in Blender	14
Figura 2-10: Spostamento di un vertice tramite editing proporzionale a fall-off differenti.....	15
Figura 2-11: Menù dei modificatori in Blender	16
Figura 2-12: Effetto del Subdivision Surface Modifier flat shaded (sopra) e smooth shaded (sotto)	17
Figura 2-13: Unione, Intersezione e differenza tra oggetti, con il modificatore Boolean applicato ad una sfera e usando un cubo come target	18
Figura 2-14: Effetto del modificatore Shrinkwrap su una mesh poligonale planare	18
Figura 2-15: Solidify modifier applicato ad una mesh (a sinistra); differenza tra Boolean semplice e complesso ed effetto dell'Angle Clamp.....	19
Figura 2-16: Marcatura Seams (cuciture) per l'UV mapping.....	20
Figura 2-17: Unwrapping automatico e visualizzazione del mapping ottenuto nell'editor UV (sinistra)	20
Figura 2-18: Modifica UV mapping in modo da ottimizzare e regolarizzare lo spazio della texture	20
Figura 2-19: Applicazione di una texture di prova per verificare la correttezza dell'UV mapping	21
Figura 2-20: Combinazione delle varie textures collegandole ai diversi canali del Base Material Node	23
Figura 3-1: Confronto tra topologie di un High Poly importato da CAD e di un modello texturizzabile	24

Figura 3-2: Avvolgimento della gabbia di modellazione esterna (visualizzata in wireframe) sul modello high poly tramite modificatore Shrinkwrap	27
Figura 3-3: Avvolgimento della gabbia texturizzata con proiezione delle seam	28
Figura 3-4: Difetti nel funzionamento dello Shrinkwrap in corrispondenza di feature dettagliate	29
Figura 3-5: Cutter per preparare la gabbia all'avvolgimento corretto in corrispondenza dell'occhiello	29
Figura 3-6: Confronto tra i risultati dell'avvolgimento con o senza cutter.....	30
Figura 3-7: Geometria texturizzata del componente calcio, render in Blender	32
Figura 3-8: Geometria texturizzata del componente calcio, render in Blender	32
Figura 3-9: Geometria texturizzata del componente astina, render in Blender	32
Figura 3-10: Applicazione della normal map per riprodurre dettagli su modelli low poly	33
Figura 3-11: Dall'esterno verso l'interno: Baking Cage (arancio chiaro); Low Poly (arancio scuro), High Poly (wireframe nero).....	34
Figura 3-12: Confronto tra UV mapping (sinistra) e normal map ottenute tramite baking: calcio sopra ed astina sotto. Modelli non zigrinati.....	35
Figura 3-13: Scanalatura leggermente deformata in Unreal a causa della triangolazione del modello..	36
Figura 3-14: Confronto UV-mapping del modello in Unreal (sinistra, triangolarizzato) e quello in Blender (destra, quadrangolare)	36
Figura 3-15: Flowchart del processo Shrinkwrap + Normal Baking.....	38
Figura 3-16: Risultato modelli di calcio e astina senza zigrino con il metodo descritto in Figura 3-15..	39
Figura 4-1: Diverse varianti del componente calcio importate come High Poly in Blender.....	40
Figura 4-2: Deformazione del texturing dovuta allo Shrinkwrap	41
Figura 4-3: Cutters utilizzati per il metodo Shrinkwrap sul componente calcio.....	42
Figura 4-4: Confronto tra modelli di calcio high poly con zigrino e senza	44
Figura 4-5: Avvolgimento della gabbia di modellazione sull'High Poly della carcassa	47
Figura 4-6: Rifinitura feature geometrica all'estremità della carcassa tramite Boolean.....	47
Figura 4-7: Render del modello carcassa ottenuto in Blender.....	47
Figura 5-1: Difetti locali nelle aree sottoposte a cutting tramite modificatore Boolean.....	50
Figura 5-2: Porzione della normal map in cui vi è l'incisione del logo (sinistra); sample di sostituzione ottenuto in GIMP (a destra)	51
Figura 5-3: Confronto tra qualità della scritta incisa se ottenuta con baking semplice (sinistra), o sostituita (destra).....	52

Figura 5-4: Confronto visibilità della discontinuità in corrispondenza della seam sull'impugnatura del calcio con materiali differenti: A sinistra: materiale disomogeneo, seam molto visibile; Al centro: discontinuità poco visibile; A destra: materiale omogeneo, seam non visibile	52
Figura 5-5: Renders di assieme calcio, carcassa, astina in Unreal Engine 5: Legni A, B.....	55
Figura 5-6: Renders di assieme calcio, carcassa, astina in Unreal Engine 5: Legni C, D, E	56
Figura 5-7: Renders di assieme di calcio, carcassa, astina in Unreal Engine 5: Polimero	57

IV. Elenco delle tabelle

Tabella 5.1: Risultati relativi al primo caso studio: tempi indicativi di esecuzione delle macrofasi di processo tra le varie versioni di calcio fornite	48
Tabella 5.2: Risultati relativi al secondo caso studio: tempi di esecuzione fasi di processo da base a ziggrinato per calcio e astina	49
Tabella 5.3: Dati di vertici e facce relativi ai modelli iniziali e finali.....	53

Abstract

Il lavoro svolto in questa attività sperimentale di tesi è una ricerca metodologica che si propone di accelerare la generazione di modelli virtuali, sui quali è possibile applicare textures per una resa fotorealistica, partendo da files CAD progettuali di componenti industriali.

Si sta riconoscendo sempre di più il valore della Realtà Estesa nella manifattura: non solo come mezzo di comunicazione per la prototipazione e la progettazione industriale, ma anche a scopo di marketing per offrire al cliente un'esperienza dei prodotti completa, coinvolgente, rapida e comoda.

È pertanto utile cercare di abbattere i costi di generazione dei modelli virtuali da impiegare in queste tipologie di applicazioni in modo da incentivarne la diffusione.

L'obiettivo dell'attività di ricerca è quindi di comprendere ed analizzare il workflow per l'ottenimento di modelli virtuali allo scopo di automatizzare il più possibile tutti i processi che richiedono l'intervento artigianale del designer, in modo da standardizzare tali procedure e renderle condivise e ripetibili a livello aziendale.

Il metodo messo a punto è stato battezzato con il nome di *Shrinkwrap Modeling* o *Modellazione per Avvolgimento*: esso consente di riprodurre rapidamente un modello virtuale con mappatura UV partendo da un file CAD.

1 Introduzione

La necessità di ottenere dei modelli virtuali fotorealistici è un'esigenza che, storicamente, apparteneva esclusivamente al mondo della Game Industry e che solo recentemente, grazie al forte sviluppo delle tecnologie digitali immersive, è diventata propria anche delle realtà aziendali manifatturiere.

L'evoluzione della Realtà Estesa (termine comprensivo per indicare Realtà Virtuale, Aumentata, Mista; Figura 1-1) sta apportando una vera e propria rivoluzione ai modelli di business: essa viene considerata, in un articolo del 2022 sulla XR per applicazioni mediche [1], un pilastro fondamentale del Metaverso insieme ad altri aspetti come AI, Blockchain, Computer vision, IoT e Robotica.

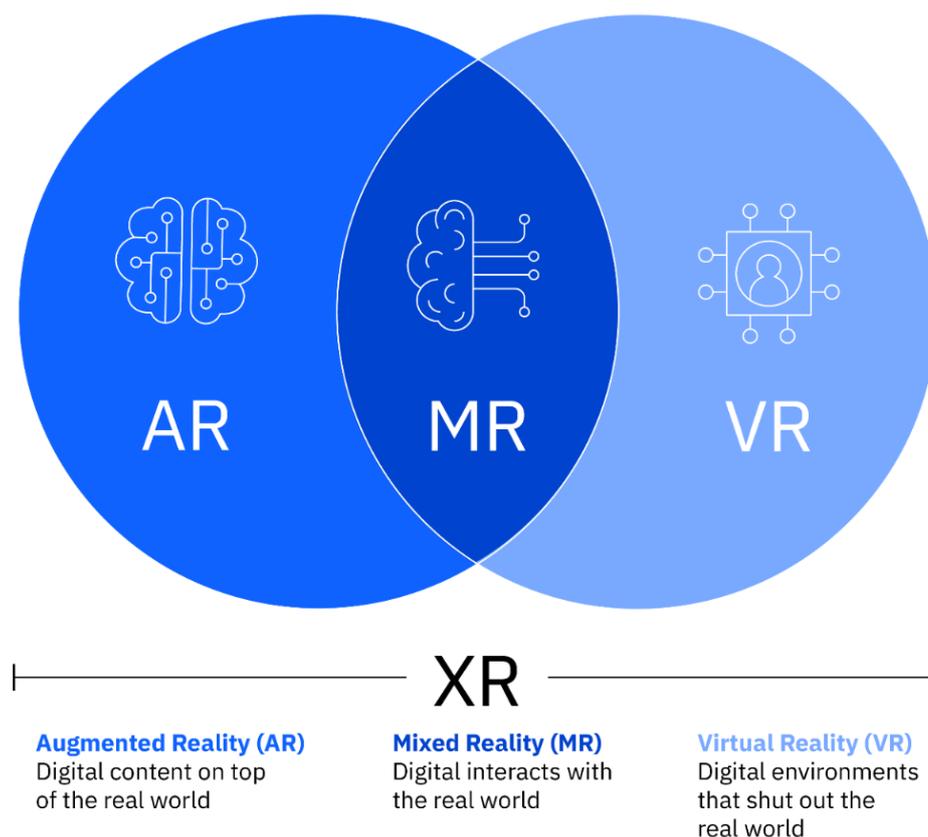


Figura 1-1: Relazione tra Realtà Estesa (XR), Aumentata (AR), Virtuale (VR), Mista (MR) [2]

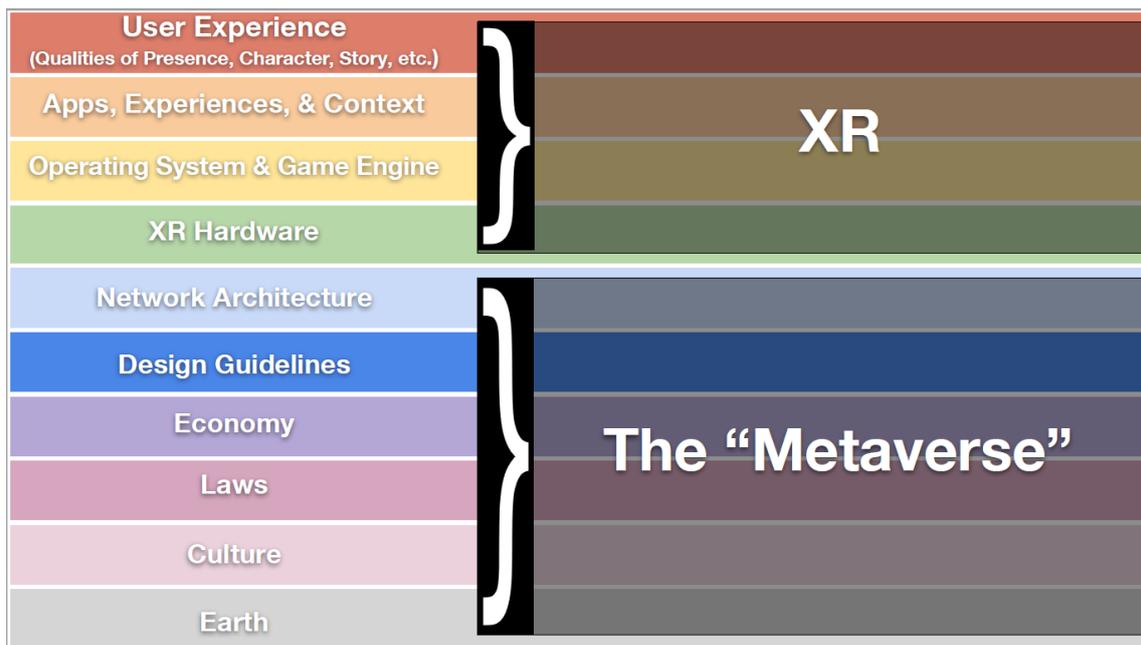


Figura 1-2: Concetti di Metaverso e Realtà Estesa a confronto, IEEE [3]

“Sono i nuovi ruoli professionali a costituire la realtà più interessante del Metaverso, un po' sulla scorta di quanto accaduto con Internet agli inizi del millennio. Quante professioni sono nate con il web? Basta pensare al digital marketing, ai social creator e social media manager, web content... una quantità di figure prima del tutto assenti dalle realtà lavorative. Adesso, allo stesso modo, la nuova frontiera del Metaverso porterà a profili innovativi. Come nel caso degli architetti del Metaverso, figure creative a cui spetta il compito di immaginare, progettare e dar vita a mondi nuovi interattivi, ambienti virtuali ed esperienze 3D immersive, coinvolgenti e fortemente engaging. Alle competenze in animazione, illustrazione e modellazione 3D, dovranno affiancarsi skill di programmazione per essere in grado di padroneggiare software come Unity 3D, Blender, Maya e Unreal Engine. Di pari passo, i costruttori del Metaverso dovranno sviluppare l'infrastruttura alla base dei vari mondi virtuali e che possono dar vita ai progetti ideati e realizzati degli architetti del Metaverso. Queste figure comprendono principalmente programmatori, sviluppatori, ingegneri del software ed esperti di Blockchain.”

da Cliclavoro - Ministero del lavoro e delle politiche sociali [17]

È in questo contesto di sviluppo dell'immersività digitale che viene ideato il progetto 4USER, fondato da Benelli Armi in filiera con Cadland SPA, Dago Elettronica e AM Microsystems e cofinanziato dal POR MARCHE FESR.

Il progetto è volto ad incentivare la ricerca e l'innovazione per la specializzazione intelligente a livello industriale, a proposito si riporta di seguito una descrizione su un articolo della Regione Marche del 11/02/2021 [4].

“Il progetto 4USER si fonda sul pieno coinvolgimento dell'utente nella progettazione e realizzazione del prodotto. Infatti, adottando un approccio user-centered è possibile raccogliere una serie di informazioni dettagliate che permetteranno di sviluppare un manufatto finale in linea con le esigenze dell'utente e che sia sicuro, confortevole ed efficiente. Il metodo del progetto 4USER prevede che la ricerca preventiva di funzionalità, design e operabilità nella fase di sviluppo del prodotto, venga eseguita utilizzando strumenti digitali e virtuali in modo da recepire anticipatamente le esigenze soggettive dell'utente e trasformarle in specifiche tecniche. Il risultato principale di 4USER sarà lo sviluppo di un prototipo virtuale interattivo al fine di sostituire i prototipi fisici tradizionali e le loro limitazioni.”

Capofila del progetto è l'azienda Benelli Armi Spa di Urbino, che sta incentrando il proprio lavoro sulla personalizzazione avanzata dei prodotti dal punto di vista dell'ergonomia, dell'estetica e della funzionalità attraverso l'utilizzo della Mixed Reality. A questo proposito, al momento della stesura di questa tesi, è in fase di sviluppo l'allestimento di uno stand per una User Experience, che permette all'utente di testare un prototipo interattivo realizzato tramite Additive Manufacturing. Attraverso un visore il cliente ha la possibilità di visualizzare l'arma nei colori e materiali desiderati e, attraverso l'implementazione di uno scatto ad azionamento elettronico sul prototipo interattivo, si raccolgono dati soggettivi e oggettivi dell'interazione utente-arma al fine di definire un comportamento dello scatto ottimale, e cercare la configurazione perfetta per il cliente.

La proiezione del modello virtuale sul prototipo reale è realizzata tramite una localizzazione dell'arma all'interno di una zona di tracking e quindi la riproduzione

visiva dei modelli, salvati all'interno dell'applicazione, alle coordinate corrispondenti a quelle fisiche del prototipo.

L'applicazione è stata sviluppata da Cadland Spa su uno dei motori grafici più avanzati e potenti del momento: Unreal Engine.

Unreal Engine è un game engine di computer grafica 3D sviluppato da Epic Games, e lanciato nel 1998. È basato sul linguaggio di programmazione C++ e supporta un range molto ampio di piattaforme desktop, mobile, console ed Extended Reality: sia stand-alone che PC-wired.

È un software completamente open source, che ha acquisito sempre più importanza anche al di fuori del mondo della Game Industry al punto da essere adottato da colossi come Audi o Tesla nel settore dell'automotive [5], o recentemente da Baxter nell'interior design per la realizzazione di configuratori, showrooms o simulatori.

Il software ha riscosso un'enorme risposta dalla community dopo il rilascio dell'ultima versione Unreal Engine 5, grazie a una serie di innovazioni che hanno apportato un'incredibile spinta in avanti al fotorealismo in *real time* il quale è stato reso comparabile, in termini di qualità, alla grafica realizzata tramite *ray tracing*, ma decisamente meno oneroso dal punto di vista computazionale.

Come engine grafico, è perfetto per creare scenari interattivi con oggetti tridimensionali ottenuti ad esempio tramite modellazione in programmi appositi come Blender: un software libero e multipiattaforma di modellazione, rigging, animazione, montaggio video, composizione, rendering e texturing.

Esso, insieme a 3D Studio Max e Maya, rappresenta al momento uno dei migliori software per l'elaborazione ed il texturing di modelli 3D poligonali, ed è per questo che è stato scelto per la creazione dei modelli virtuali dei componenti dei prodotti Benelli da inserire all'interno dell'applicazione MR per la user experience sviluppata da Cadland Spa.

I modelli dei componenti sono, al momento dello sviluppo di questa tesi, ottenuti con tecniche di modellazione standard, prendendo come riferimento i prototipi CAD progettuali sviluppati dal gruppo di disegno Benelli, con costi di produzione relativamente alti.

I prodotti, infatti, sono sottoposti a continue modifiche funzionali e di design: sia a causa del ciclo di sviluppo prodotto flessibile adottato dalla Benelli, sia per l'esigenza di personalizzare la produzione sulla base della domanda di mercato.

Pertanto, è stata riscontrata l'esigenza di sviluppare un metodo che consentisse di accelerare questo processo di "twinning prototipale", dal dominio CAD all'ambito di modellazione poligonale, in modo da rendere rapida la generazione di modelli virtuali: ciò allo scopo sia di offrire al cliente un ampio ventaglio di possibilità di configurazione, sia di mantenere una buona sincronizzazione delle modifiche apportate nello sviluppo del prodotto.

2 Stato dell'arte

2.1 Modellazione 3D: formati e rappresentazione standard di punti

Secondo Conlan C. (2017) [6], la modellazione 3D è l'arte di manipolare dati per creare rappresentazioni tridimensionali di oggetti ed ambientazioni. I modelli 3D sono delle strutture dati costituite dai seguenti elementi:

- a) Il **formato** consente ai modelli di essere ottimizzati in base all'utilizzo; tra le principali estensioni troviamo:
 - Formati di interscambio minimalistici come .stl, .obj oppure .fbx
 - Formati di interscambio CAD come .stp oppure .igs
 - Formati specifici come .blend per Blender, .3ds per 3ds Max, oppure .ma per Maya
 - Formati per il render come .babylon oppure .glsl per gli shaders di OpenGL
- b) I **vertici** e le **facce** definiscono i punti geometrici che compongono il modello e le superfici che connettono tali punti:
 - I vertici sono generalmente costituiti da terne indicizzate di numeri reali che contengono le informazioni sulle coordinate (x,y,z) di ciascun punto dell'oggetto
 - Le facce, come verrà mostrato nelle prossime pagine della trattazione, possono essere un dato complesso ma, nel caso dei formati standard di interscambio, sono delle terne di numeri interi dove (i,j,k) rappresenta il triangolo nello spazio tridimensionale formato rispettivamente dai vertici di indice i, j e k .

2.2 Rappresentazioni di linee, curve, superfici nella computer grafica

Dati due punti o vertici nello spazio tridimensionale (x,y,z) , si considera una linea il segmento che unisce la coppia di punti. Connettendo in serie diversi punti si ottiene una polilinea, che può essere una strategia per rappresentare discretamente una curva. La polilinea, infatti, è una struttura geometrica non continua e come tale avrà una certa risoluzione che dipende dalla densità di vertici che la compongono.

Essa, tuttavia, non costituisce l'unico modo per indicare matematicamente una curva: infatti nella computer grafica, come riportato in un documento relativo alla modellazione nel software Maya [7], troviamo le curve chiamate *splines*.

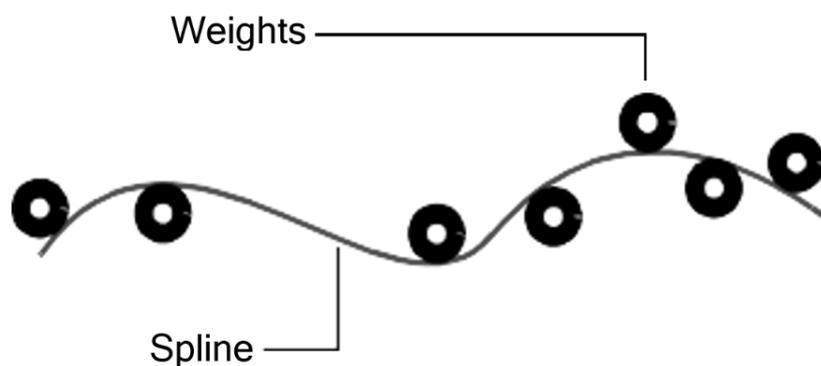


Figura 2-1: Spline fisiche usate nelle costruzioni di navi

Le *splines* sono tipologie di curve che furono storicamente sviluppate per la costruzione di navi prima dell'avvento della modellazione al computer: gli architetti navali avevano bisogno di un modo per disegnare curve morbide che passassero attraverso una serie di punti.

La soluzione fu di piazzare dei pesi metallici (chiamati *nodi*) in corrispondenza di alcuni punti di controllo, per poi deformare un flessibile di metallo sottile o di legno (chiamato appunto *spline*) in modo da farlo passare per i *nodi* (Figura 2-1).

La fisica del flessibile era tale che la sollecitazione di ciascun peso fosse massima al punto di contatto, decrescendo gradualmente allontanandosi da esso: così, per aver un maggior controllo su una certa regione dello spline, si aggiungevano semplicemente altri pesi. Questo schema presentava delle ovvie problematiche per lo scambio di dati:

si aveva bisogno di un metodo matematico per descrivere la forma delle curve. Fu introdotta la curva *spline* polinomiale cubica, perché rappresentativa del comportamento fisico del flessibile di legno: furono poi estese alle *B-spline* (che sta per *Basis-splines*), che sono somme di *splines* polinomiali di grado più basso.

Più il grado della polinomiale sarà basso, più la curva descritta sarà semplice: gradi più alti delle curve possono rappresentare geometrie più complesse ma aumenta decisamente anche l'onere computazionale. Un grado del polinomio eccessivamente alto, inoltre, determina una difficoltà a parametrizzare superfici a tratti piani a causa dell'aumento di sensibilità del modello matematico.

Per questi motivi solitamente non si utilizzano B-splines di grado superiore al settimo. Tipi particolari di *B-splines* sono le *NURBS*, acronimo che sta per *Non-Uniform Rational B-Splines*:

- *Non-Uniform* si riferisce alla parametrizzazione della curva, ossia al fatto che i nodi possono essere distribuiti in maniera non uniforme
- *Rational* si riferisce alla rappresentazione matematica, più in particolare alla possibilità di rappresentare coniche

Con la denominazione *superficie NURBS* (Figura 2-2) ci si riferisce a delle superfici ottenute da una serie di curve *NURBS*, per le quali si individuano quindi due direzioni ortogonali delle isocurve che si indicano con U e V. In Figura 2-2 sono riportate alcune strategie per ottenere superfici da rete di curve in *Maya*.

Relativamente alle direzioni isoparametriche U e V, sono individuate le *normals* come vettori perpendicolari in ciascun punto alla curva o superficie, che quindi possono essere uscenti o entranti dalla stessa consentendo perciò di indicare un'orientazione della superficie di tipo "dentro" o "fuori".

Le *normals* possono essere anche degli indicatori indiretti della geometria della curva o superficie; infatti, dato che sono perpendicolari in ciascun punto della superficie, l'addensarsi o il diradarsi di queste possono rivelare anche dei cambi di curvatura minimi. Dalla metà del secolo scorso fino ad oggi, il metodo di rappresentazione di superfici *NURBS* è stato quello più utilizzato (le rappresentazioni *Bézier* o *B-spline* possono essere considerati dei casi particolari del *NURBS*) tuttavia, grazie agli enormi

progressi nelle performance di memoria ed operative dei calcolatori, si è fatto strada l'utilizzo della *mesh poligonale* che, a differenza del *NURBS*, è un sistema di rappresentazione approssimata e non continua, basato sulla discretizzazione delle superfici in poligoni prevalentemente piani (Figura 2-3).

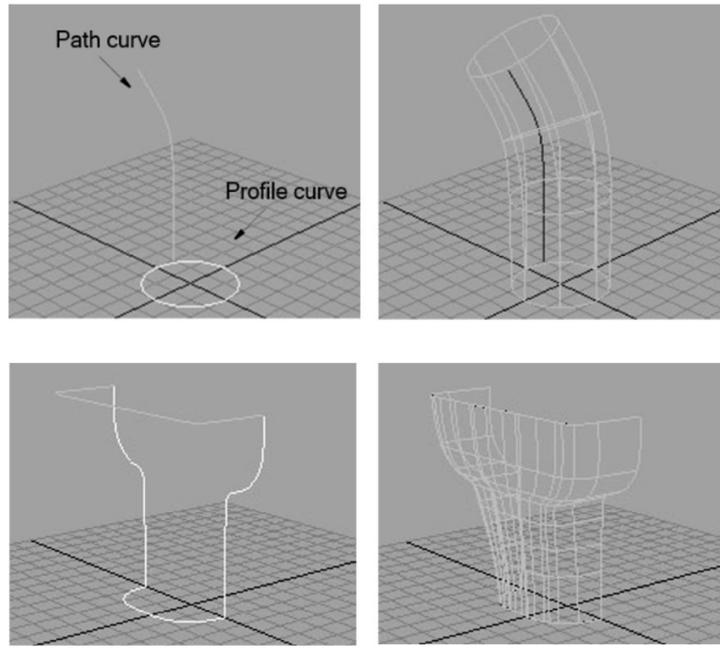


Figura 2-2: Differenti strategie di ottenimento di superfici NURBS da curve

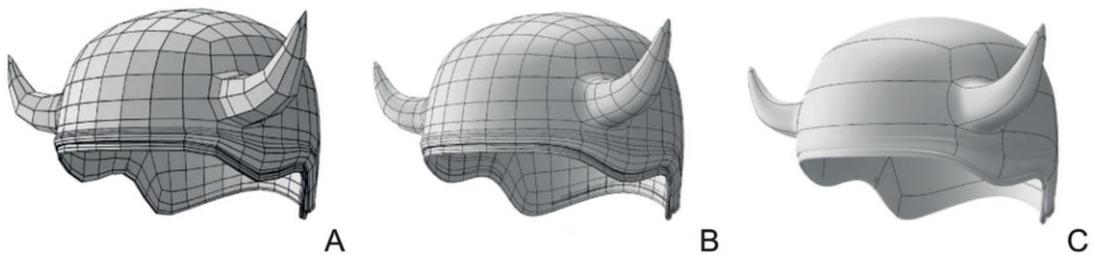


Figura 2-3: Tipologie di superfici 3D: A: mesh quadrangolare; B: mesh quadrangolare, smoothed; C: superficie NURBS [8]

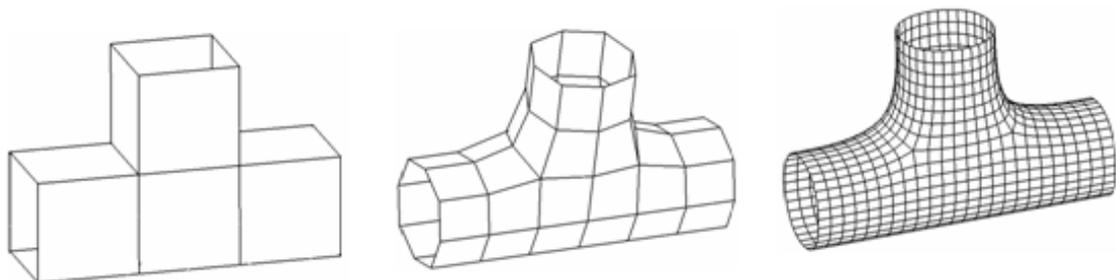


Figura 2-4: Subdivision Surface di tipo Catmull-Clark: A: mesh non suddivisa; B: prima suddivisione; C: seconda suddivisione

Una soluzione intermedia tra rappresentazioni di superfici di tipo discreto e continuo, che consente di superare le problematiche relative allo *shading* e alla computazione della geometria NURBS, è la *subdivision surface*: affrontata accuratamente nello studio di Liao W. et al. (2017), [9]. Si tratta di una mesh poligonale i cui vertici costituiscono dei punti di controllo per una interpolazione o approssimazione determinando un *refinement*, ossia un aumento di risoluzione e quindi un effetto di *smoothing* geometrico.

L'algorithmo di suddivisione della superficie prende il nome di *schema di suddivisione*, uno dei più conosciuti è il Catmull-Clark, mostrato in Figura 2-4.

2.3 Cenni su Blender

I softwares di modellazione 3D, in genere, consentono di lavorare flessibilmente con tutte le rappresentazioni geometriche affrontate in questo capitolo: tra questi spicca Blender, un programma grafico open source multipiattaforma che è diventato uno dei punti di riferimento della modellazione 3D insieme a Maya o 3ds Max.

Il software viene sfruttato, oltre che per la modellazione geometrica, per rigging, animazione, montaggio video, composizione, rendering e texturing. Dispone di funzionalità per mappature UV, simulazioni di fluidi, particle systems e altre simulazioni non lineari.

L'interfaccia è costituita da aree, ciascuna delle quali contiene un *editor*: essa è altamente personalizzabile, nel layout di default la parte centrale è una vista 3D sulla scena, ed è dotata di una serie di tools e opzioni come mostrato in Figura 2-5.

La modellazione in Blender avviene agendo sugli *objects* nella scena in due possibili modalità: in "*Object Mode*" o in "*Edit Mode*". Quando si è in *Object Mode*, tutte le trasformazioni di rotazione e scaling avvengono attorno all'origine dell'oggetto *attivo* (evidenziato in un colore più chiaro degli oggetti *selezionati*).

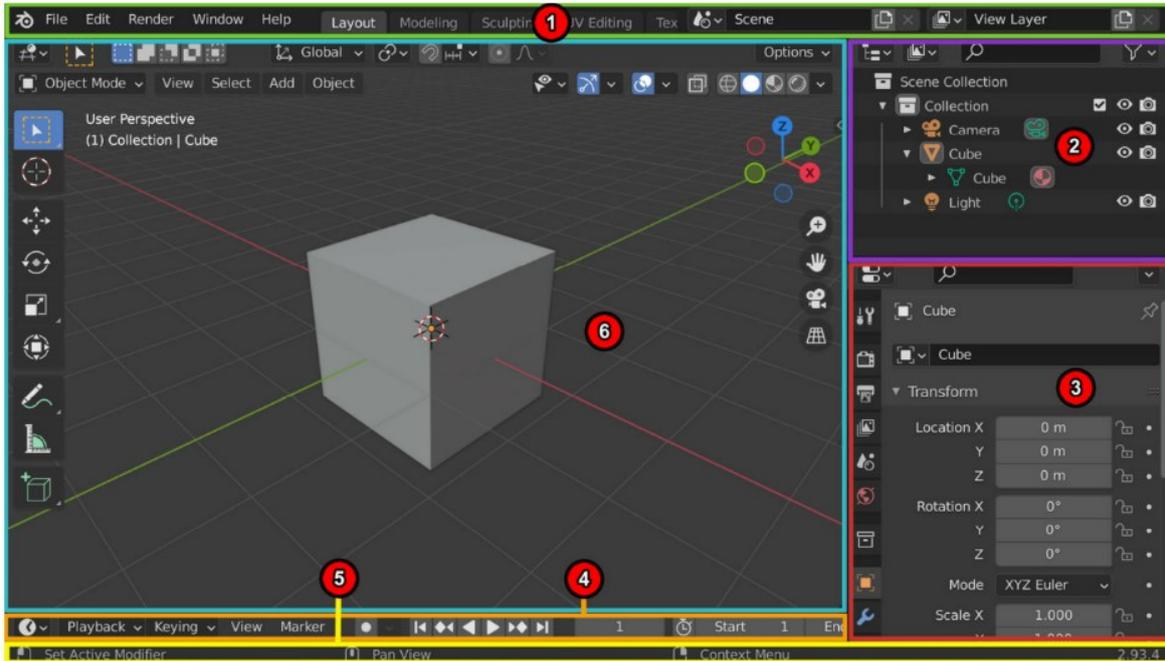


Figura 2-5: Interfaccia grafica di Blender e relative aree: 1: Topbar; 2: Outliner; 3: Properties; 4: Timeline; 5: Status Bar; 6: 3D Viewport [15]

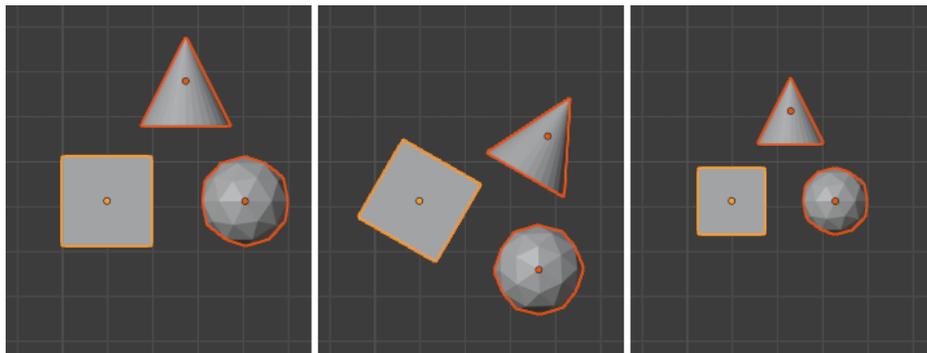


Figura 2-6: Tre oggetti poligonali in Blender, rotazione e scaling in Object Mode [10]

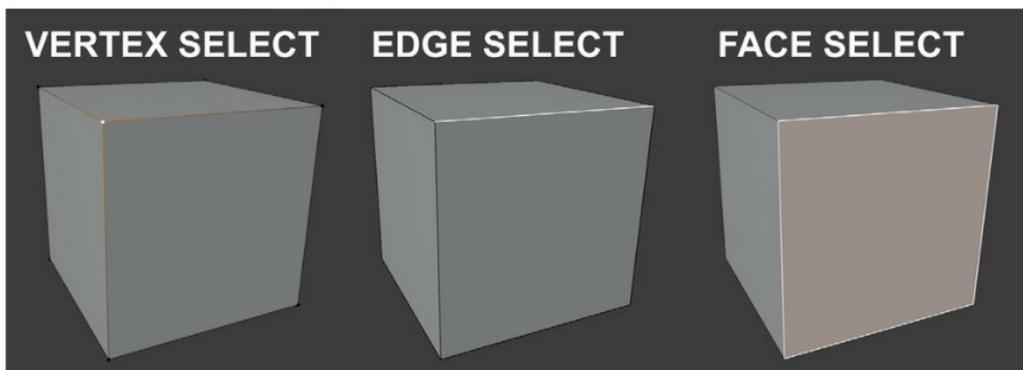


Figura 2-7: Edit mode, le tre modalità di selezione disponibili: Vertex, Edge, Face

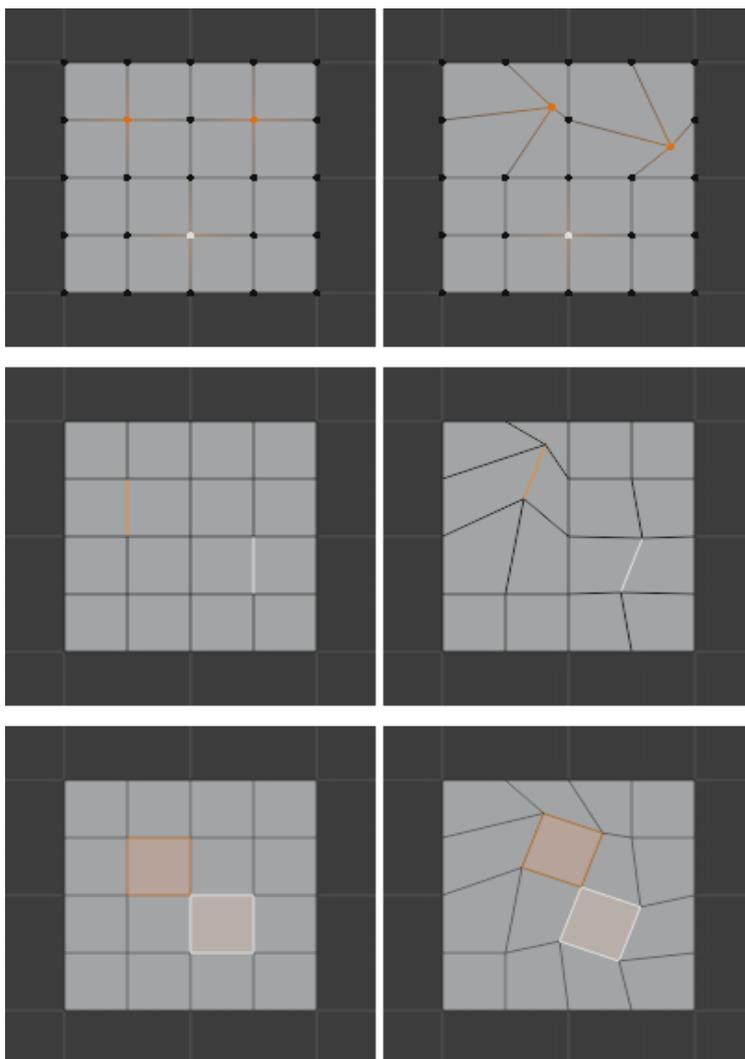


Figura 2-8: Rotazione in Edit mode: Vertici in Vertex Mode (in alto), Bordi in Edge Mode (al centro), Facce in Face Mode (in basso) [10]

Quando invece si è in *Edit Mode*, invece, è possibile agire direttamente su vertici, bordi o facce dell'oggetto attivo a seconda che si sia in *Vertex Mode*, *Edge Mode* o *Face Mode* (Figura 2-7).

Mentre si è in *Edit Mode*, le rotazioni e gli scaling non avvengono più rispetto all'origine dell'oggetto ma relativamente alla posizione reciproca degli elementi selezionati a seconda dell'impostazione del *Pivot Point*, come mostrato in Figura 2-8.

In basso a destra si trova il "*Properties Editor*", attraverso cui è possibile settare un'infinita varietà di impostazioni relativamente all'oggetto e alla scena.

2.4 Modellazione in Blender

2.4.1 Superfici e shading

Blender è principalmente un software di modellazione poligonale: cioè, come già illustrato, ha una modalità di rappresentazione di superfici *faceted* o a facce piane; quindi il modello di rappresentazione più vicino può essere considerato quello della mesh poligonale.

Tuttavia, è possibile costruire tali superfici poligonali in vari modi, anche attraverso la modellazione per curve e superfici di tipo *NURBS*.

Inoltre, è possibile trasformare una mesh poligonale in una *subdivision surface* attraverso l'utilizzo di *modifiers*, di cui si parlerà nelle pagine seguenti.

Particolare rilevanza nella modellazione in Blender è rappresentata dalla tipologia di *shading*, che può essere di tipo *flat* o *smooth*. È possibile applicare un threshold allo smoothing in base all'angolo di transizione tra facce adiacenti: questo fa in modo che gli spigoli vivi vengano visualizzati come tali e si eviti che l'oggetto risulti innaturale nell'aspetto come nel caso della rappresentazione in Figura 2-9.

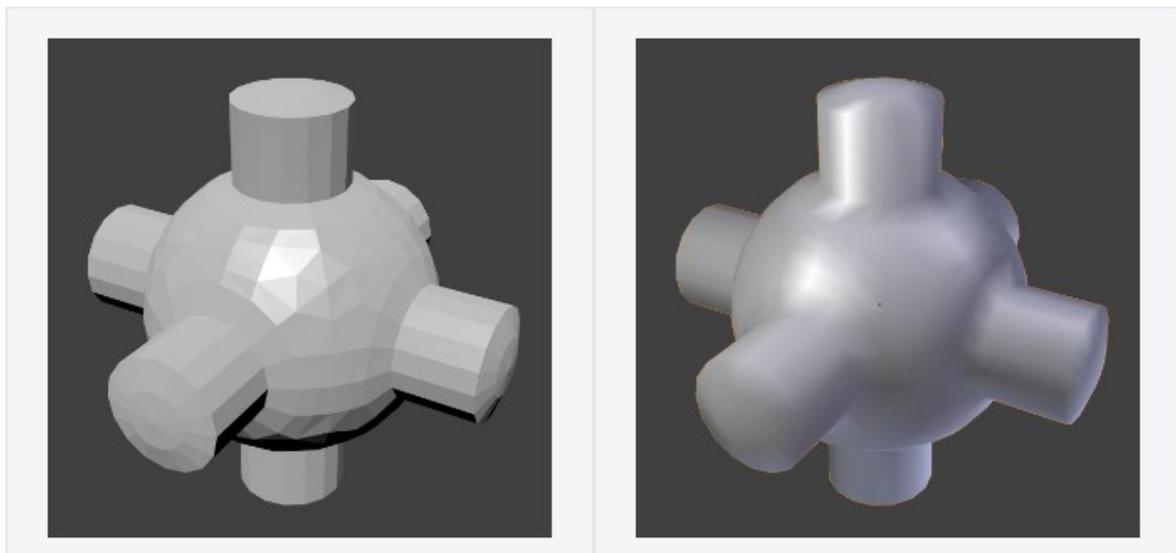


Figura 2-9: Effetto dello smooth shading in Blender [11]

2.4.2 Editing proporzionale

Blender dispone di un'ottima funzione che consente di modellare eseguendo spostamenti, scaling, rotazioni, con una sorta di *fall-off*, il cui raggio di influenza è regolabile dall'utente. Questa è una funzionalità che offre la possibilità di apportare deformazioni alle mesh in maniera estremamente flessibile.

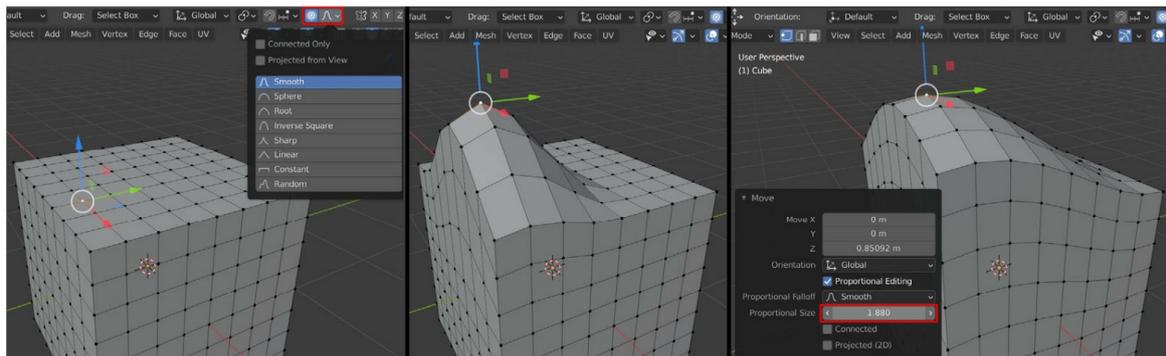


Figura 2-10: Spostamento di un vertice tramite editing proporzionale a fall-off differenti [15]

2.5 Modificatori

2.5.1 Menù dei Modificatori

I modificatori sono algoritmi automatici, applicabili ad una mesh poligonale, che modificano la geometria in modo non distruttivo. Attraverso il loro utilizzo è possibile applicare delle variazioni istantanee di forma e topologia attraverso operazioni che richiederebbero molto tempo per essere svolte manualmente: il tutto senza modificare la struttura della mesh.

Attribuendo ad un oggetto un modificatore, infatti, viene mostrato l'effetto di questo consentendo però di agire, sequenzialmente a priori, sulla mesh originale. Diversi modificatori possono essere applicati in combinazione ad uno stesso elemento poligonale inserendoli nel *Modifier Stack* relativo all'oggetto e, in caso di necessità, il loro effetto può essere applicato alla mesh in maniera permanente (distruttiva)

eseguendone l'*Apply*. La sequenzialità dei modificatori in serie sullo *Stack* è un dato significativo e va gestita correttamente.

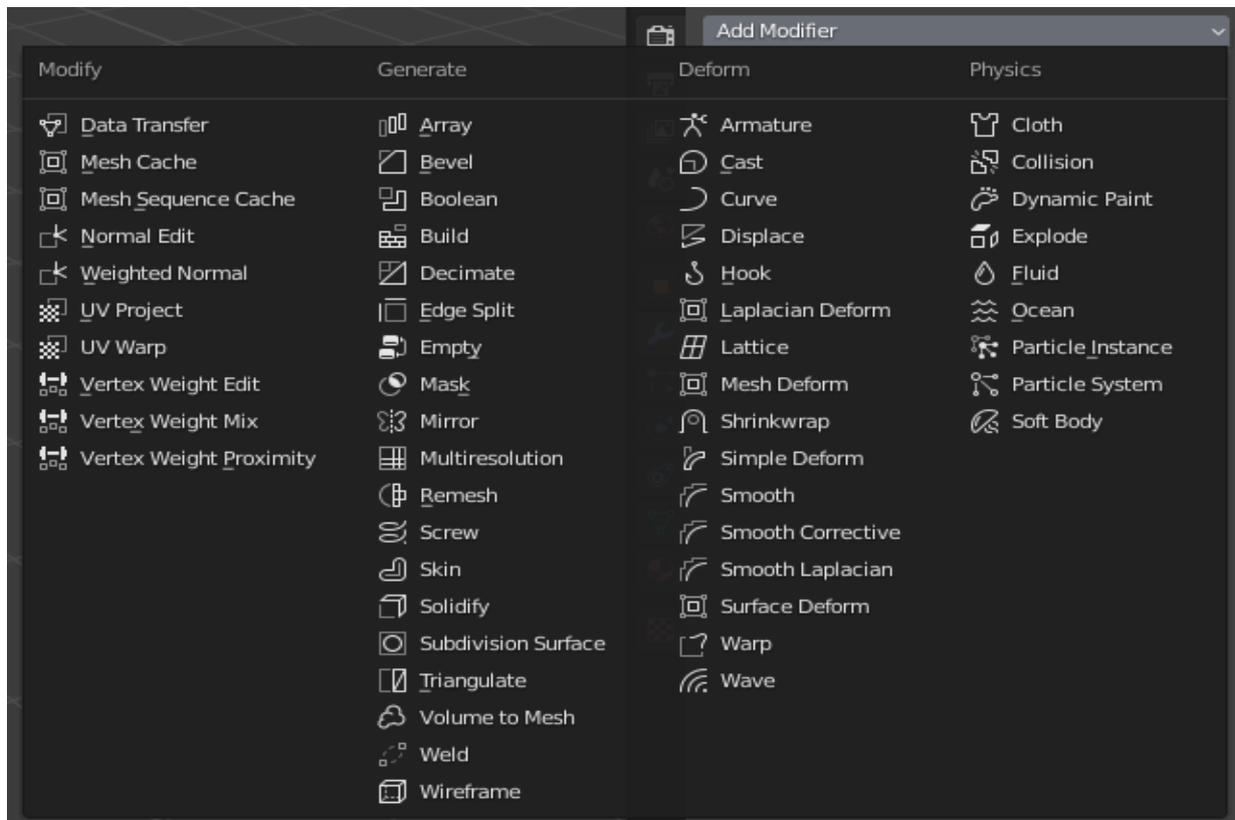


Figura 2-11: Menù dei modificatori in Blender

Come mostrato in Figura 2-11, raffigurante il menù dei modificatori implementati in Blender, ci sono quattro categorie di modificatori:

1. **Modify Modifiers:** Sono simili a quelli *Deform*, ma di solito hanno effetto su dati dell'oggetto differenti dalla geometria.
2. **Generate Modifiers:** Sono quelli che agiscono direttamente su forma e topologia della mesh, possono cambiare la struttura dell'oggetto o aggiungere nuove geometrie ad esso.
3. **Deform Modifiers:** Cambiano solo la forma dell'oggetto senza variarne la topologia.
4. **Simulate Modifiers:** Servono per modellare l'oggetto attraverso delle simulazioni fisiche, utilizzato ad esempio per vestiti, fluidi o particelle.

Tra la grande varietà di modificatori disponibili in Blender, in questa trattazione ci si limiterà a descrivere sommariamente il funzionamento di quelli sfruttati per l'implementazione della metodologia sviluppata.

2.5.2 *Subdivision Surface Modifier*

Attraverso l'applicazione di due possibili schemi di suddivisione (*Simple* o *Catmull-Clark*), applica alla mesh un algoritmo di suddivisione trasformandola in una *subdivision surface*: consentendo le strategie di modellazione proprie di questa categoria di superfici. In particolare, applicando il *Catmull-Clark*, oltre alla moltiplicazione di vertici viene anche eseguito un *refinement* della superficie come presentato al Capitolo 2.2.

Come affermato in [11] ed illustrato in Figura 2-12, aumentare il numero di suddivisioni della mesh è un'operazione diversa rispetto a visualizzare la superficie come *smooth shaded*: l'utilizzo sapiente e combinato di queste due operazioni consente di ottenere delle superfici ben levigate senza aumentare eccessivamente il numero di suddivisioni, e quindi di vertici e facce che appesantiscono il modello.



Figura 2-12: Effetto del *Subdivision Surface Modifier* flat shaded (sopra) e smooth shaded (sotto) [10]

2.5.3 *Boolean Modifier*

Il modificatore *Boolean* combina le geometrie di coppie di oggetti diversi in un unico oggetto in tre possibili modi: Unione, Intersezione, Differenza. In quest'ultima

modalità, in particolare, è possibile modellare per sottrazione di geometrie attraverso l'utilizzo di "cutters" o oggetti di taglio.

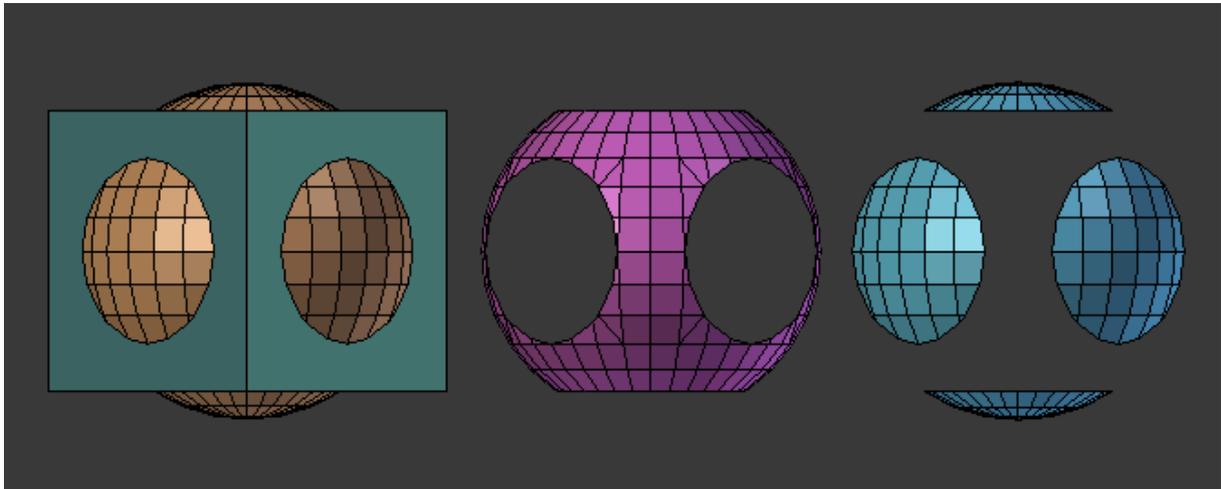


Figura 2-13: Unione, Intersezione e differenza tra oggetti, con il modificatore Boolean applicato ad una sfera e usando un cubo come target [10]

2.5.4 Shrinkwrap Modifier

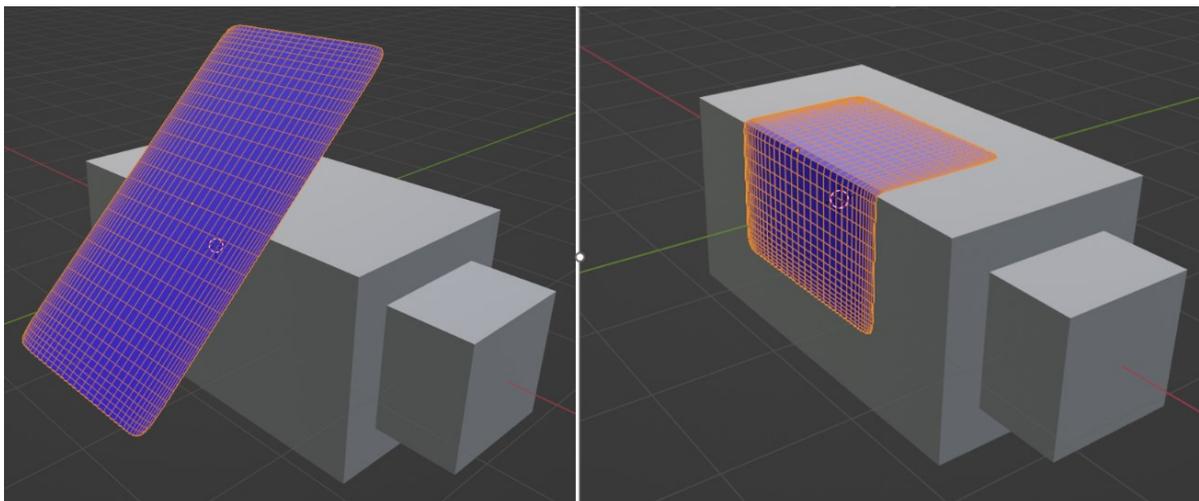


Figura 2-14: Effetto del modificatore Shrinkwrap su una mesh poligonale planare

Il modificatore *Shrinkwrap* può essere applicato a superfici, meshes, lattices ed elementi testo: esso consente di proiettare l'oggetto su una superficie target.

I parametri che regolano l'effetto del modificatore sono:

- *Target*: l'oggetto su cui proiettare la mesh
- *Offset*: la distanza a cui mantenere la mesh avvolta rispetto al target
- *Vertex Group*: il modificatore può agire selettivamente sui vertici della mesh

2.5.5 Solidify Modifier

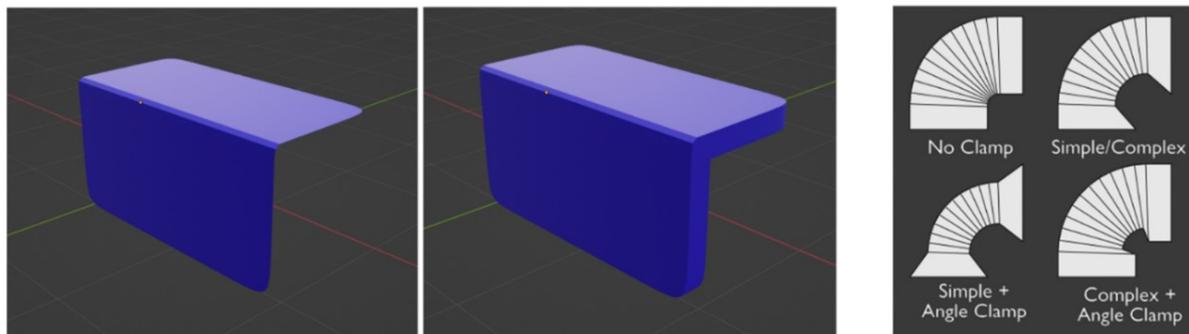


Figura 2-15: Solidify modifier applicato ad una mesh (a sinistra); differenza tra Boolean semplice e complesso ed effetto dell'Angle Clamp [10]

Il modificatore *Solidify* può essere riferito sia a solidi che a superfici non chiuse, con l'effetto di conferire uno spessore alle mesh aperte, e trasformare il solido chiuso in una struttura a guscio. Risulta abbastanza complicato da gestire, ma la distinzione più importante si effettua tra algoritmo semplice e complesso, e un parametro rilevante è la *thickness clamp*: entrambi questi aspetti determinano il comportamento del modificatore sugli spigoli dell'oggetto e il loro effetto è mostrato in Figura 2-15.

Lo spessore può essere applicato considerando come punto medio la superficie originale, oppure può essere spostato verso l'interno o verso l'esterno regolando un opportuno indicatore: questo aspetto condiziona ovviamente le dimensioni interne ed esterne e può comportare delle problematiche di natura geometrica come la sovrapposizione di volumi in corrispondenza degli spigoli.

2.6 Rendering

2.6.1 UV Mapping e Texturing

Le coordinate UV sono essenzialmente una rappresentazione bidimensionale di un oggetto 3D, e la loro corrispondenza sulla superficie tridimensionale del modello prende il nome di *UV-mapping*. L'*unwrapping* è, invece, l'operazione automatica di mappatura secondo alcuni input: principalmente la marcatura delle *seam* (cuciture). Secondo Caudron & Nicq, (2015) [12], l'obiettivo di un buon *unwrapping* sta nello

scegliere la configurazione ottimale per consentire al software di appiattare la geometria senza generare eccessive deformazioni.

Nella sua guida *Blender 2.5 Lighting and Rendering* [13], Powell mostra come eseguire il texturing di una bottiglia, di cui si riportano le immagini da Figura 2-16 a Figura 2-19.

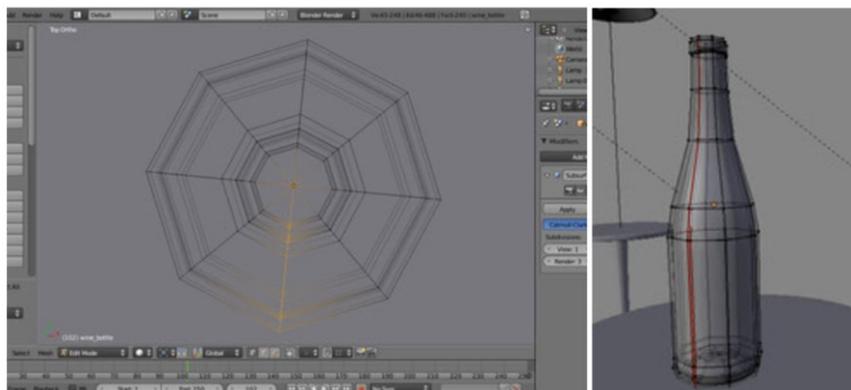


Figura 2-16: Marcatura seams (cuciture) per l'UV mapping

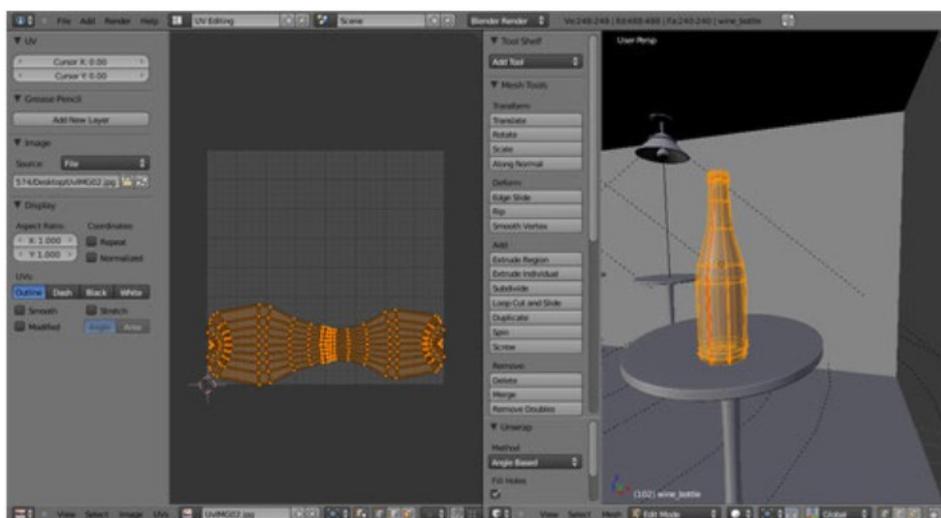


Figura 2-17: Unwrapping automatico e visualizzazione del mapping ottenuto nell'editor UV (sinistra)

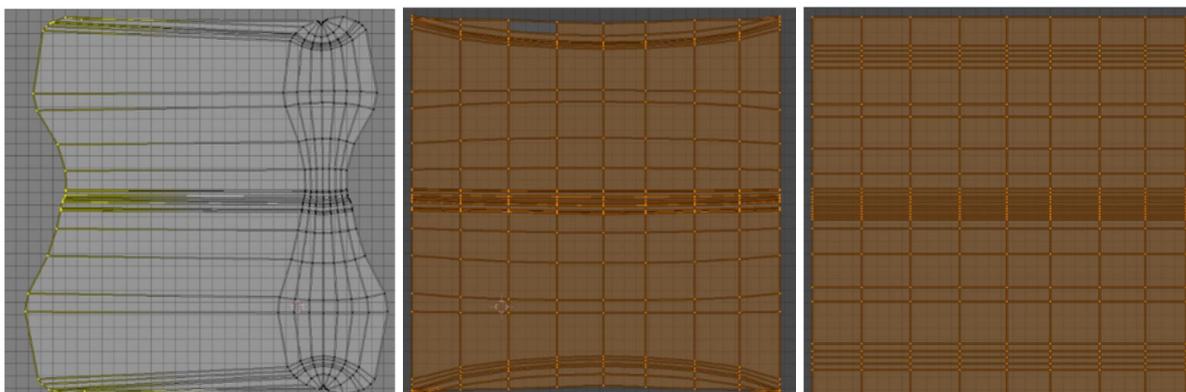


Figura 2-18: Modifica UV mapping in modo da ottimizzare e regolarizzare lo spazio della texture

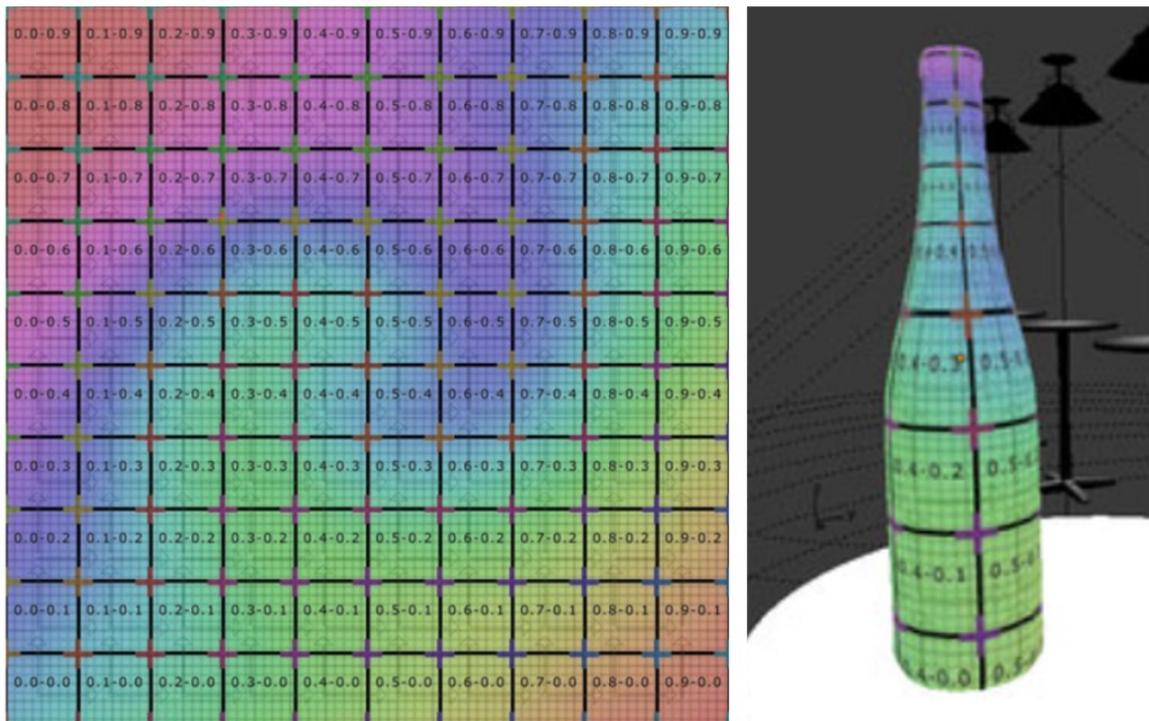


Figura 2-19: Applicazione di una texture di prova per verificare la correttezza dell'UV mapping

Osservando in sequenza le immagini, si nota come il primo step sia quello di individuare le *seams*, o cuciture di bordo per la mappatura UV, dopodiché si procede con il comando di *unwrapping* seguito da una ottimizzazione della disposizione delle facce poligonali sul piano delle textures.

Infine, viene applicata una texture sample per rilevare eventuali anomalie o deformazioni del texturing applicato al modello, per poter eventualmente correggere manualmente l'UV mapping.

2.6.2 Textures e materiali nella computer grafica 3D

Nella computer grafica 3D, per descrivere l'aspetto di una superficie mesh, vengono utilizzati dei file chiamati *textures* o *texture maps*: esse sono immagini raster, e possono essere ottenute anche in maniera procedurale: ossia non mediante acquisizioni fotografiche ma tramite un algoritmo software che elabora un template sample come illustrato nell'esperimento di H. Y. Pai, (2019), [14].

Ecco un elenco delle principali tipologie di textures riportato nella guida di Venter, H. (2022), [15]: esse contengono informazioni differenti e complementari relative al

materiale. Alcune di esse vengono solitamente utilizzate in maniera combinata a seconda delle caratteristiche estetiche della superficie.

- **Albedo Map:** è il colore della superficie priva di riflessi luminosi e di zone d'ombra. In alcuni software come Blender o Unreal Engine è chiamata anche *base color map*.
- **Metalness Map:** indica quali parti della superficie sono di metallo e quali no. Ciascun pixel della texture può comunque assumere valori da 0 a 255, quindi è un'informazione di tipo non binario. Può essere molto utile ad esempio per riprodurre rapidamente la presenza di chiodi metallici su una scatola di legno.
- **Roughness Map:** Contiene l'informazione della rugosità superficiale anch'essa come scala di grigi; determina se il comportamento della luce sulla superficie è diffusivo o riflettente. In sostituzione ad essa si utilizza talvolta la **Specular Map**, che rappresenta il complemento dell'informazione di rugosità.
- **Ambient Occlusion Map:** spesso indicata come *AO map*, è una texture a 8 bit che serve per creare gli effetti di *soft shadowing* dell'illuminazione indiretta in base alla geometria dell'oggetto (ad esempio per scanalature, indentazioni o crepe sulla superficie).
- **Emissive Color Map:** nota anche con il nome di *self-illumination map*, è utilizzata per riprodurre aree di fluorescenza del materiale: combinata con altre textures consente di riprodurre effetti di luce molto particolari.
- **Displacement Map:** è utilizzata per deformare la geometria della mesh in base al valore in scala di grigi dei pixel che compongono la texture.
- **Curvature Map:** è una texture che dà un valore in scala di grigi a spigoli e cavità nella mesh. Questa informazione è principalmente utilizzata per il texturing procedurale.
- **Material ID Map:** Esegue un labeling dei vari materiali assegnando un colore per ciascuno di essi, in modo che si possano poi creare proceduralmente delle textures in cui vengono mixati insieme le proprietà relative a diversi materiali e creare un oggetto multimateriale.

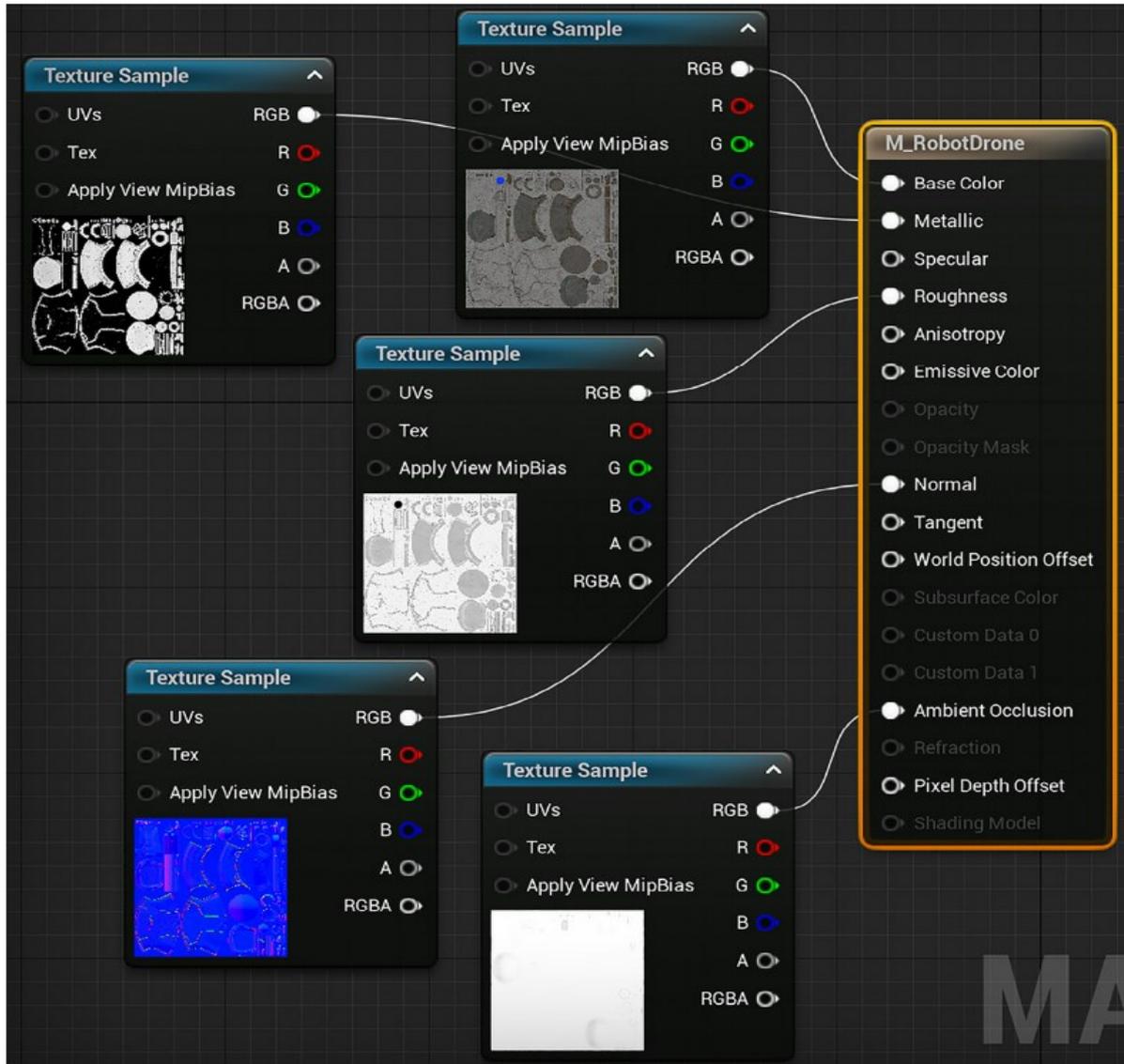


Figura 2-20: Combinazione delle varie textures collegandole ai diversi canali del Base Material Node [15]

All'interno dei software grafici vi è un sistema per poter combinare le varie textures disponibili riferendole allo stesso materiale, che viene poi assegnato all'oggetto.

Nel caso di Unreal Engine 5, che ha un editor di materiali con sistema a nodi logici, le textures vengono importate nel *Material Editor* sotto forma di nodi texture, per poi essere collegate al *Base Material Node*, che rappresenta l'output del materiale (Figura 2-20).

3 Metodo

3.1 Il problema della topologia

Per la collaborazione di questo progetto di tesi è stato richiesto di analizzare tutti gli step necessari per la conversione di un modello CAD in un oggetto virtuale texturizzabile, compatibile con l'applicazione del configuratore di prodotto in Realtà Mista sviluppata da Cadland Spa. L'analisi aveva lo scopo di comprendere le criticità e cercare eventuali soluzioni che snellissero tutta la procedura.

Nello specifico, un prototipo realizzato in un software CAD deve essere riprodotto in ambito di modellazione poligonale affinché, sulla sua superficie, possano essere applicati diversi materiali tramite la combinazione di textures.

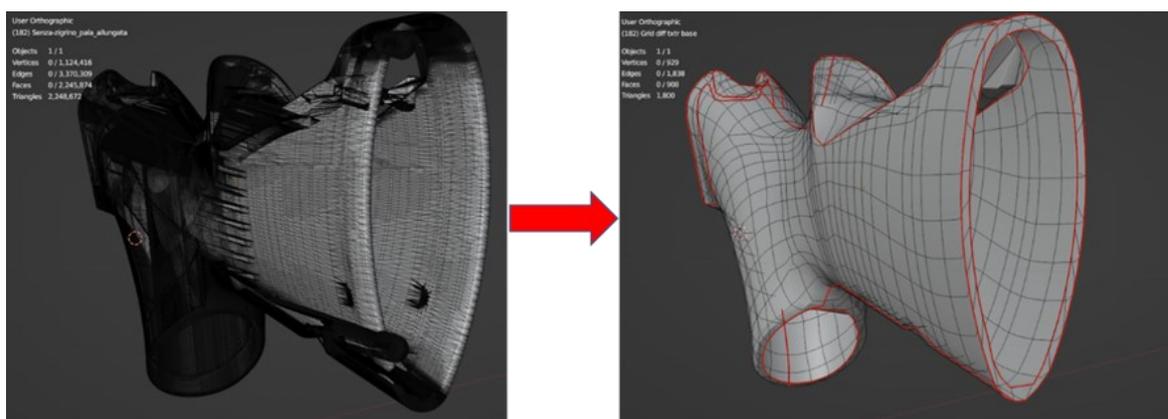


Figura 3-1: Confronto tra topologie di un High Poly importato da CAD e di un modello texturizzabile

Dopo l'esportazione dal dominio CAD in uno dei formati di interscambio come STL o OBJ, si importa un modello triangolarizzato con geometria poligonale approssimata da modellazione solida.

Il modello importato è un *High Poly*: esso risulta inadatto all'operazione di UV Mapping in quanto avente una topologia (configurazione di punti, bordi, facce) estremamente densa e caotica. Deve essere pertanto ottenuto un modello analogo con topologia corretta sia per poter individuare facilmente le *seams* per l'UV Mapping, sia perché il file esportato deve essere leggero per poter essere renderizzato in real time nell'applicazione MR. Inoltre, tutti i dettagli strutturali dell'architettura interna dei

componenti non sono necessari nel modello virtuale poiché è sufficiente che il componente sia riprodotto in maniera fedele soltanto esternamente, ossia nelle parti in vista (Figura 3-1).

Dopo la marcatura dei bordi di cucitura per le textures, si esegue l'*unwrapping* per visualizzare la disposizione della geometria che costituisce il modello nel piano della texture corrispondente. Possono quindi essere applicate delle modifiche manuali e, una volta che si è soddisfatti con l' UV Mapping, si può procedere all'esportazione come file FBX o OBJ.

Il modello può essere importato in Blender come STL, OBJ, FBX o PLY dopodiché deve essere ottenuto un suo analogo con topologia corretta: questa operazione può essere svolta apportando le modifiche direttamente sul vecchio file BLEND precedentemente già UV-mappato, oppure attraverso un processo di rimodellazione da zero del nuovo modello importato se la prima opzione risultasse eccessivamente complessa.

Qualora fosse possibile agire con modifiche manuali sul vecchio file BLEND, alcune delle strategie applicabili potrebbero essere lo spostamento proporzionale features in caso di deviazioni di forma o piccole variazioni dimensionali, aggiunta di smussi o raccordi, di *loop cuts*, etc.

Se non fosse possibile o conveniente apportare modifiche manuali al vecchio modello, è invece necessario crearne uno nuovo da zero, prendendo come riferimento l'*High Poly* importato dal CAD.

Allo stato attuale delle cose, per fare ciò, si procede con una modellazione manuale sfruttando la *subdivision modeling*: tecnica che si basa sulla deformazione di una forma base a cui è applicato un *Subdivision Surface Modifier*, plasmando la geometria finale attraverso lo spostamento di alcuni punti di controllo. Un'altra possibilità potrebbe essere ad esempio ricalcare il profilo con una superficie esterna applicando uno *snapping* superficiale sul modello CAD importato.

Lo scopo di questa attività di tesi è stato quindi di cercare una metodologia snella, rapida e il più possibile automatica per riprodurre da zero un analogo poligonale di modelli CAD, in modo che si possa aggiornare rapidamente in caso di modifiche frequenti.

A tale fine si è pensato che non fosse conveniente sfruttare fin da subito una programmazione con codice *Python* (per cui vi è un ambiente di sviluppo implementato all'interno di Blender). Data la difficoltà di fare fronte alla richiesta flessibilità di utilizzo, infatti, si è preferito in un primo momento sperimentare come poter sfruttare le risorse già sviluppate e ottimizzate all'interno del software per snellire quanto più possibile la procedura. Così, sperimentando l'utilizzo dei vari modificatori di Blender, si è trovato un metodo che vuole ricalcare la superficie esterna di un modello *high poly*, ottenendo un analogo geometrico con topologia corretta automatizzandone l'applicazione delle texture.

Il metodo in questione è stato battezzato con il nome di *Shrinkwrap Modeling*, o *Modellazione per Avvolgimento*, perché sfrutta il modificatore *Shrinkwrap* come elemento principale del processo.

3.2 Modificatori come soluzione di automazione

Un modificatore in Blender è un algoritmo, riferito alla mesh, che svolge in maniera automatica delle operazioni in maniera non distruttiva: cioè non modifica la mesh ma rimane in memoria mostrandone l'effetto, consentendo di continuare a lavorare sulla mesh originale e permettendo una modifica dinamica dei parametri. Diversi modificatori possono essere attribuiti ad uno stesso oggetto eseguendovi, automaticamente ed in serie, operazioni differenti combinabili a piacimento. La sequenzialità dei modificatori deve essere opportunamente gestita perché l'ordine di applicazione dei modificatori è un elemento significativo.

Si è quindi considerato come si potesse adottare un metodo che si basasse su un "grezzo", cioè un modello simile al risultato voluto ma semplificato, dal quale fosse possibile ottenere la geometria finale applicando soltanto modificatori in sequenza. Sfruttando l'azione di questi sarebbe quindi possibile modellare una sola volta il grezzo, adatto ciascuno a un singolo componente, ed esso può essere riutilizzato per una rimodellazione rapida dello stesso componente in versione aggiornata.

Il principio di funzionamento del metodo messo a punto è la proiezione (avvolgimento) di un modello esterno semplice e topologicamente corretto (gabbia, grezzo) sul modello importato da CAD attraverso l'azione del modificatore *Shrinkwrap*, descritto nel Capitolo 2.5.4.

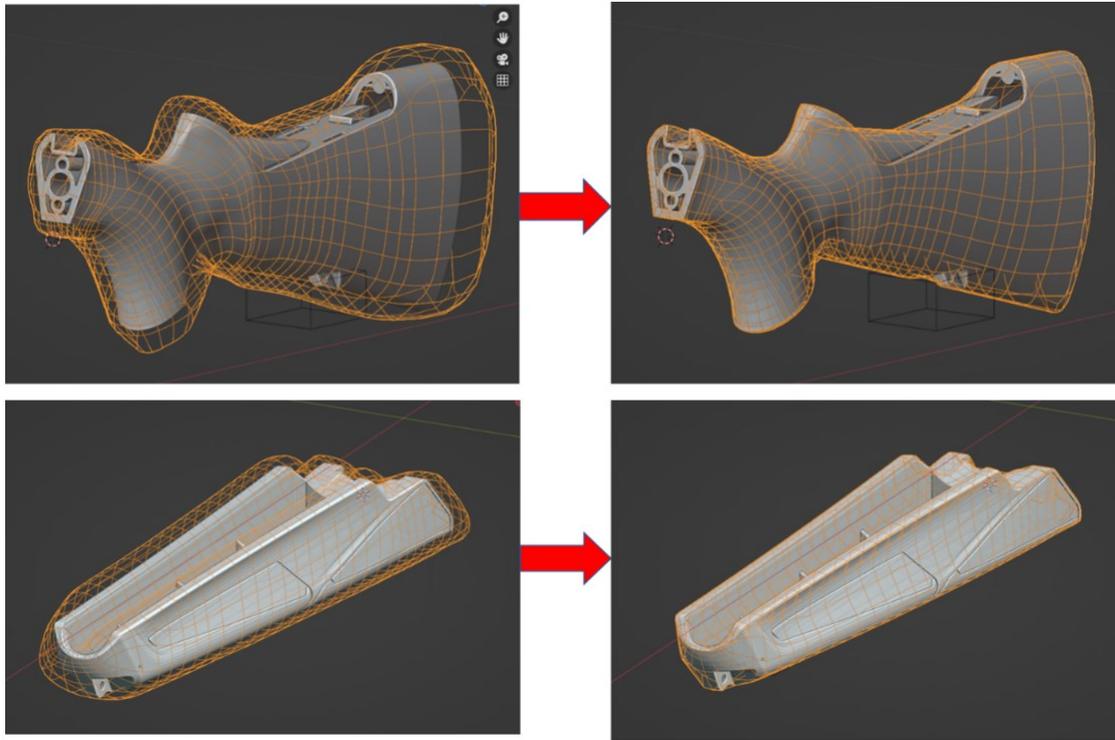


Figura 3-2: Avvolgimento della gabbia di modellazione esterna (visualizzata in wireframe) sul modello high poly tramite modificatore *Shrinkwrap*

La gabbia esterna, adoperata come grezzo, deve essere ottenuta tramite una modellazione manuale una sola volta per ciascun componente per il quale si voglia adottare questo metodo.

La marcatura di *seams* e l'*unwrapping* sono operazioni effettuate sulla gabbia esterna, o gabbia di modellazione, a priori dell'azione dei modificatori: per aggiornare il modello bisogna quindi cambiare il target dei modificatori e assestare con piccole modifiche manuali di spostamenti o deformazioni locali o globali.

L'intervento manuale che viene svolto nel metodo proposto agisce anch'esso a monte dell'azione dei modificatori: qualora si volesse agire a posteriori, ad esempio per correggere manualmente l'effetto di un modificatore, è necessario eseguire l'*Apply* dello stesso per applicarlo in maniera distruttiva, per poi mettervi mano successivamente.

3.3 Utilizzo della gabbia di modellazione

La gabbia di modellazione è ottenuta tramite processo manuale seguendo i profili del componente da avvolgere e testandone periodicamente la correttezza di proiezione.

La marcatura di *seams* è un'operazione svolta sulla gabbia, è necessario solamente verificare che tali bordi siano proiettati correttamente sul modello *high poly*.

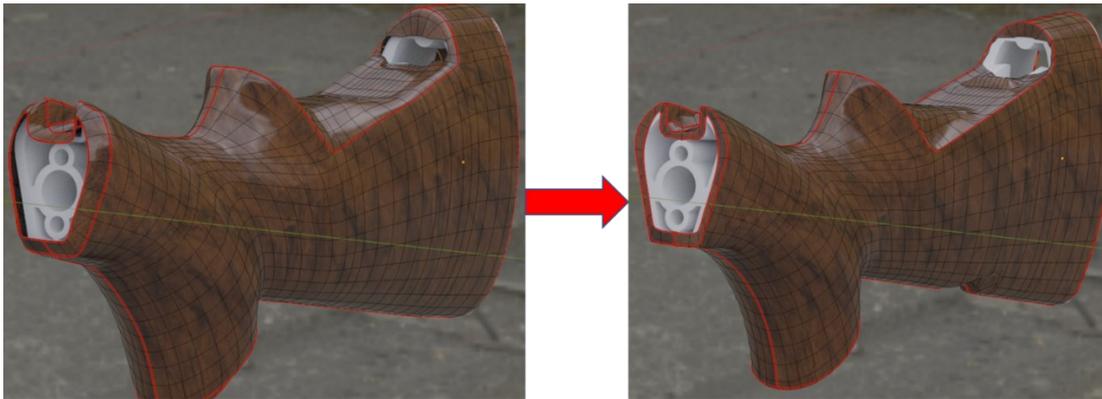


Figura 3-3: Avvolgimento della gabbia texturizzata con proiezione delle seam

Il modificatore *Shrinkwrap* ha, ovviamente, dei limiti di applicabilità: analogamente a ciò che accade ad una pellicola quando essa viene avvolta su un oggetto con superfici scanalate, forate o comunque con dettagli ad elevata curvatura superficiale, l'avvolgimento non avviene correttamente in corrispondenza delle features a medio e alto livello di dettaglio (scanalature, scavi, fori, ecc) ma si creano delle superfici ponte (*bridge-meshes*), come mostrato in Figura 3-4.

È pertanto necessario combinare il modificatore *Shrinkwrap* con altri che agiscono a priori o a posteriori rispetto ad esso, in particolare:

- Modificatore **Subdivision Surface**: a priori dell'avvolgimento, perché la mesh di avvolgimento abbia una definizione più elevata.
- Modificatori **Boolean**: a priori o a posteriori dello *Shrinkwrap* a seconda dei casi, è lo strumento di taglio per rifinire le features di dettaglio medio.
- Modificatore **Solidify**: a posteriori dell'avvolgimento, necessario in alcuni casi per avere un solido in esportazione.

Particolarmente decisivo per le features di medio dettaglio è il modificatore *Boolean*, che consente di modificare localmente la gabbia in modo che abbia la stessa geometria superficiale dell'oggetto di taglio: in questo modo può aderire, con l'azione del modificatore *Shrinkwrap*, alla geometria *high poly* anche nelle zone in questione. È particolarmente interessante il fatto che, come mostrato in Figura 3-5 e in Figura 3-6, l'UV mapping del *cutter* viene trasferito su quello del componente nei punti di contatto tra la geometria da sottrarre e il modello: per questo motivo l'UV Mapping degli oggetti di taglio deve essere effettuato accuratamente su tutti i *cutters*, altrimenti il texturing del modello non sarà corretto nella parte sottoposta a taglio *Boolean*.

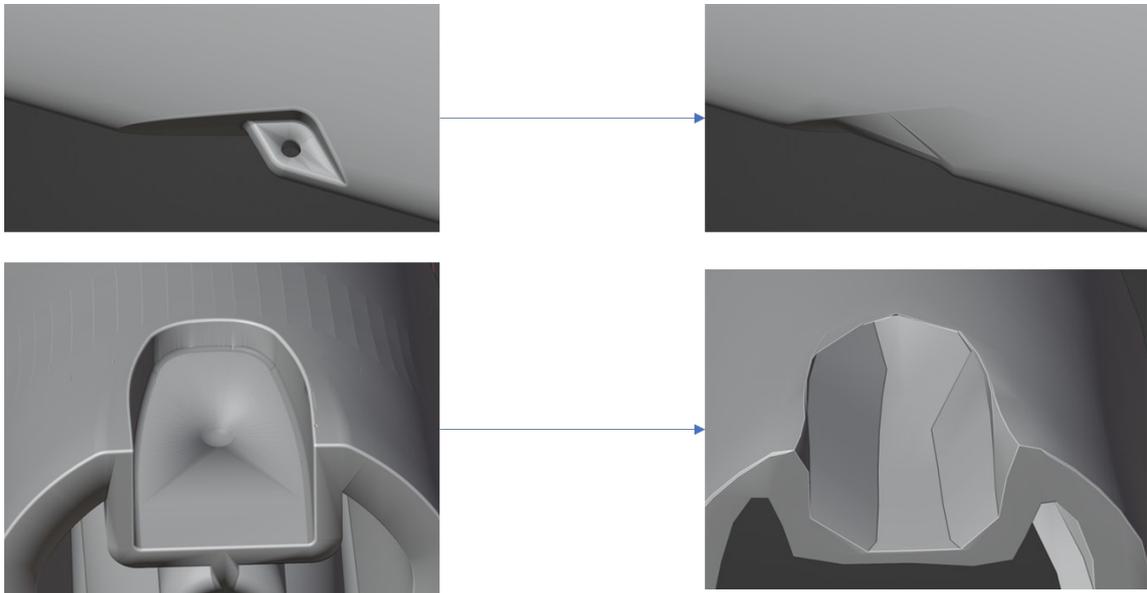


Figura 3-4: Difetti nel funzionamento dello *Shrinkwrap* in corrispondenza di feature dettagliate



Figura 3-5: *Cutter* per preparare la gabbia all'avvolgimento corretto in corrispondenza dell'occhiello

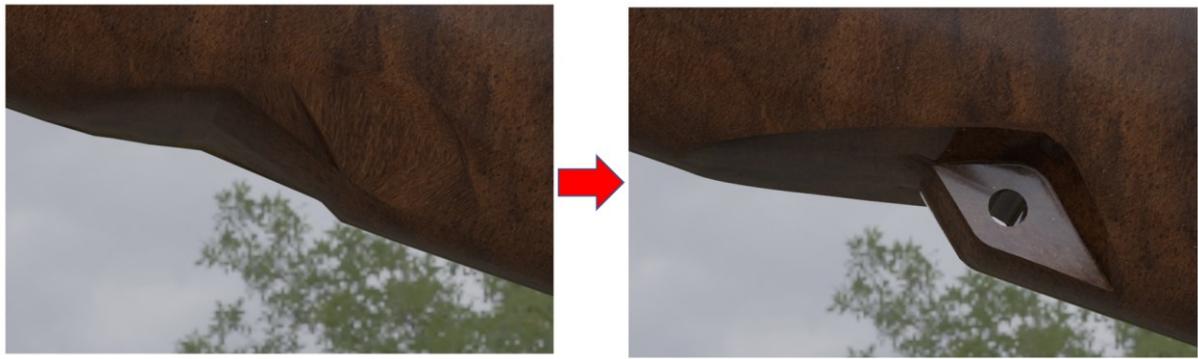


Figura 3-6: Confronto tra i risultati dell'avvolgimento con o senza cutter

3.4 Workflow per ottenere la geometria aggiornata con metodo Shrinkwrap

Immaginare di trovarsi nella situazione descritta nei punti seguenti:

- Sono state apportate delle modifiche a un componente, e si possiede il modello CAD aggiornato
- Il modello virtuale del componente in questione era già stato ottenuto nella versione precedente con il metodo *Shrinkwrap*: per questo motivo si possiede il file BLEND contenente la gabbia di modellazione insieme a tutti i cutters necessari per il procedimento descritto finora. Questi elementi sono già in posizione reciproca corretta, hanno ciascuno tutti i propri modificatori già settati in sequenza, e ciascuno la propria mappatura UV.

Il workflow per l'ottenimento della geometria aggiornata del nuovo modello virtuale, allora, consisterebbe nelle seguenti operazioni:

1. Esportazione del file dal software CAD in formato STL (o OBJ) e importazione nel file BLEND utilizzato per la precedente modellazione.
2. Fare in modo che il modello importato sia contenuto nella gabbia:
 - Se il modello importato risultasse spostato rispetto alla gabbia, posizionare la gabbia in modo che lo contenga

- Nel caso di variazioni di forma significative, il componente potrebbe deviare rispetto al volume della gabbia: a questo punto è opportuno modificare quest'ultima tramite editing proporzionale, localmente o globalmente, fino a quando non risulta un involucro per la nuova versione
3. Impostare come target del modificatore *Shrinkwrap* il modello *high poly* importato.
 4. Se la modifica riguarda una feature di dettaglio (ottenuta con il modificatore *Boolean*), agire manualmente sul *cutter* per riprodurre la modifica, in caso contrario saltare al punto 5.
 5. Verificare che tutti i modificatori stiano agendo in maniera corretta, che non ci siano problemi di *shading*, che la resa estetica delle texture sia buona; altrimenti effettuare qualche assestamento manuale sulla gabbia (azione sequenzialmente a priori rispetto ai modificatori).

Il metodo descritto consente di ricavare molto rapidamente la geometria semplificata dei componenti riproducendola in maniera ripetibile e fedele nelle parti in vista, come si può vedere dai renders da Figura 3-7 a Figura 3-9.

L'UV Mapping rimane già effettuato sulla gabbia di modellazione e sui *cutters*: può essere modificato in caso di problematiche ma, se non ci sono variazioni geometriche importanti, può essere mantenuto durante l'aggiornamento.

Tuttavia, attraverso questo metodo, non si riesce a riprodurre dettagli come scanalature, incisioni, zigrini, rilievi superficiali, ma si limita generare un modello del componente a un livello di dettaglio medio.

Per questo motivo è stato necessario ampliare il metodo, in modo che riuscisse a riprodurre anche il dettaglio di livello avanzato, integrandolo con l'operazione di *Normal Baking*, di cui si parla nel capitolo seguente.



Figura 3-7: Geometria texturizzata del componente calcio, render in Blender



Figura 3-8: Geometria texturizzata del componente calcio, render in Blender



Figura 3-9: Geometria texturizzata del componente astina, render in Blender

3.5 Integrazione metodo Shrinkwrap con Normal Baking

Per questioni legate alla performance, nel mondo della modellazione per Game Development non si utilizzano *High Poly*: motivo per cui è difficile riprodurre dettagli ad alta risoluzione ed estesi sulla superficie. Per riprodurre quindi le deviazioni di altezza rispetto alla superficie del modello si utilizzano delle *normal maps*, impiegate regolarmente per i materiali insieme ad *albedo map*, *roughness map*, *ambient occlusion map* e le altre tipologie mostrate al Capitolo 2.6.2.

Questa tipologia di texture può essere utilizzata anche per riprodurre dettagli superficiali poco profondi sulle superfici dei modelli virtuali: per cui viene utilizzata per ottenere dei modelli dettagliati senza incrementarne eccessivamente l'estensione. L'operazione di creare una *normal map* che riproduca i dettagli superficiali di un modello *high poly* su un *Low Poly* prende il nome di *Normal Baking* (Figura 3-10).

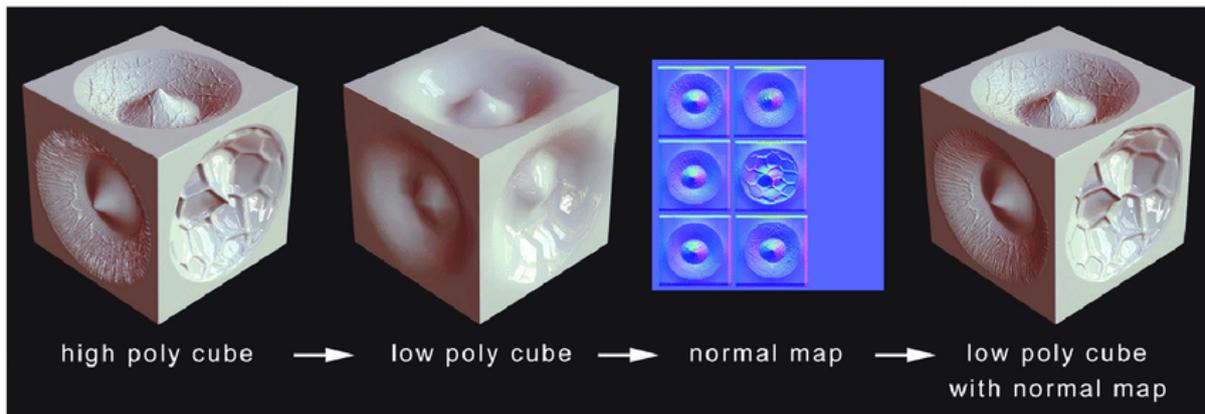


Figura 3-10: Applicazione della normal map per riprodurre dettagli su modelli low poly [16]

In Blender il *Normal Baking* avviene attraverso tre *objects*:

1. **High Poly**: i dettagli superficiali di questo modello vengono trasformati in una *normal map*
2. **Low Poly**: la texture ottenuta nel processo è generata sull'UV mapping di questo modello, riproducendo su di esso i dettagli superficiali dell'*High Poly*
3. **Baking Cage**: Dalla superficie di questo modello vengono proiettati i *normal rays* che, dalla distanza superficiale locale con la mesh dell'*High Poly*, consentono di ottenere una mappatura UV dei dettagli superficiali.

La *Baking Cage* viene ottenuta dal modello *Low Poly* effettuando uno *scaling along the normals*, cioè spostando tutte le facce lungo la loro direzione normale esterna in modo che ci sia un offset uniforme tra tutti i punti delle rispettive meshes. (Figura 3-11).

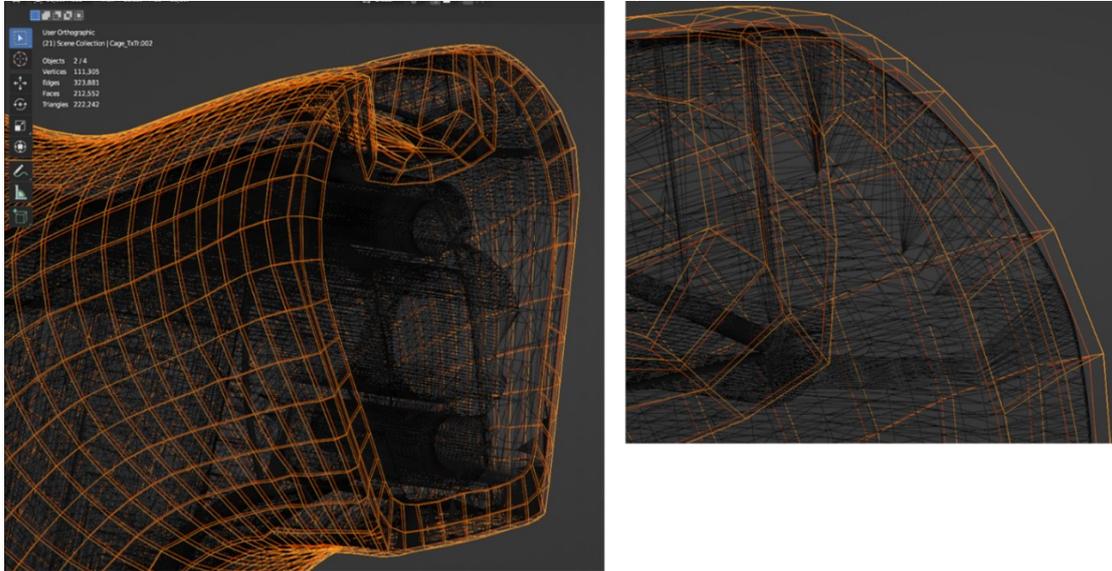


Figura 3-11: Dall'esterno verso l'interno: *Baking Cage* (arancio chiaro); *Low Poly* (arancio scuro), *High Poly* (wireframe nero)

Viene quindi prodotta una *normal map* sulla base dell'UV Mapping del *Low Poly* che, se opportunamente combinata con il materiale dell'oggetto, riproduce in maniera fedele i dettagli superficiali desiderati.

A questo proposito, dopo che la *normal map* con i dettagli superficiali è stata ottenuta, essa deve essere talvolta ripulita in un software di image editing come *GIMP* per evitare di inserire delle informazioni di superficie rumorose o non desiderate. Possono essere anche apportate modifiche di *warping* o deformazione locale, variazione di contrasto e quindi di intensità, ecc. Nel caso del calcio, questa tecnica è stata utilizzata per ottenere il dettaglio della scanalatura sull'impugnatura e del rilievo sulla punta del nasello; per l'astina gli elementi da riprodurre erano rappresentati dalle varie scanalature sugli zigri e, nella parte inferiore, l'incisione con il logo aziendale.

Il tempo di completamento dell'operazione, totalmente automatica, ovviamente dipende dalla potenza di calcolo a disposizione e dalla risoluzione impostata per la *normal map*. Per dare un ordine di grandezza dell'onere computazionale, con lo stesso dispositivo hardware una *normal map* di risoluzione di 4096 x 4096 pixel viene ottenuta nel giro di pochi minuti, mentre una di 8192 x 8192 richiede diverse ore.

In Figura 3-12 sono riportate le *normal maps* ottenute tramite *Baking* relativamente ai modelli di calcio e astina.

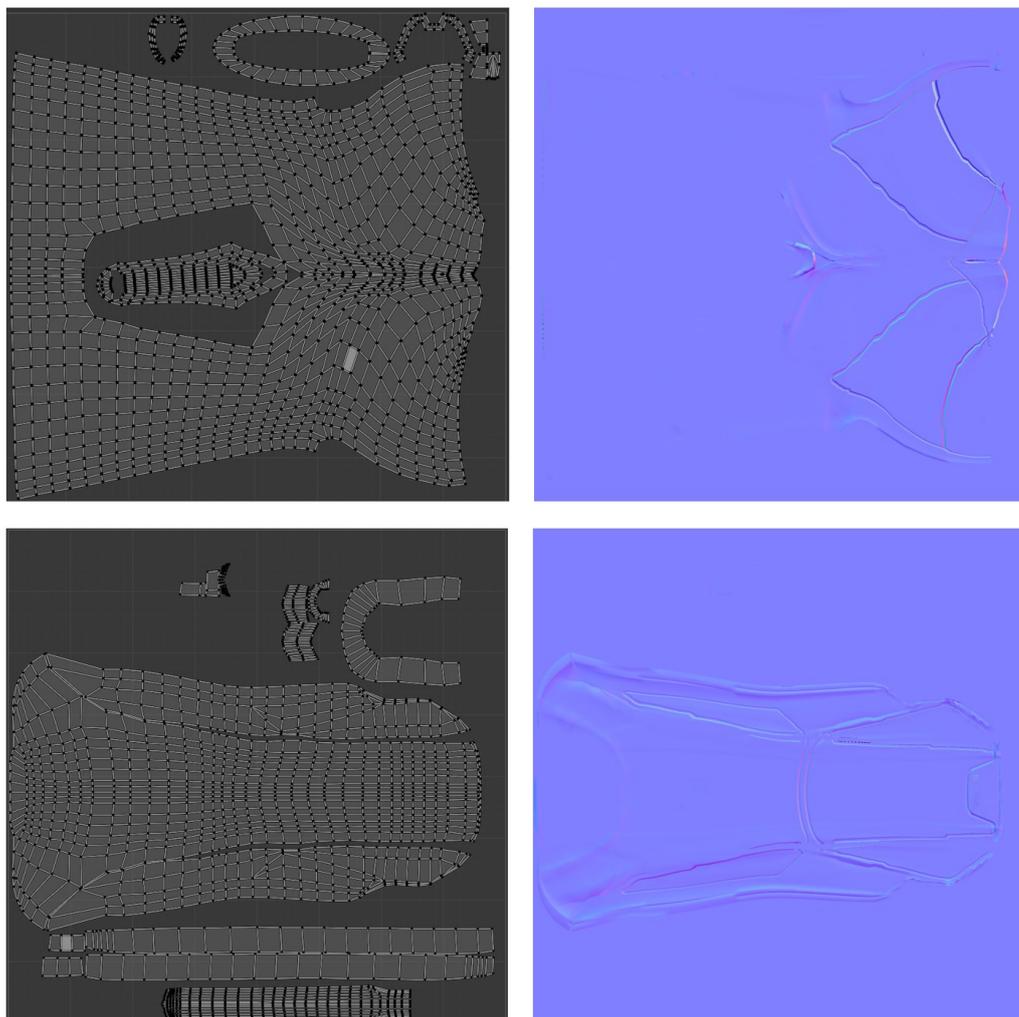


Figura 3-12: Confronto tra UV mapping (sinistra) e normal map ottenute tramite baking: calcio sopra ed astina sotto. Modelli non zigrinati.

Sebbene i dettagli in Blender fossero resi in maniera perfetta applicando la *normal map*, si è notato che invece, una volta esportato il file e importato in Unreal, i dettagli dello stesso modello con la medesima *normal map* applicata risultavano leggermente deformati.

È possibile notare la lieve deformazione della scanalatura, che si generava dopo l'esportazione, in Figura 3-13.



Figura 3-13: Scanalatura leggermente deformata in Unreal a causa della triangolazione del modello

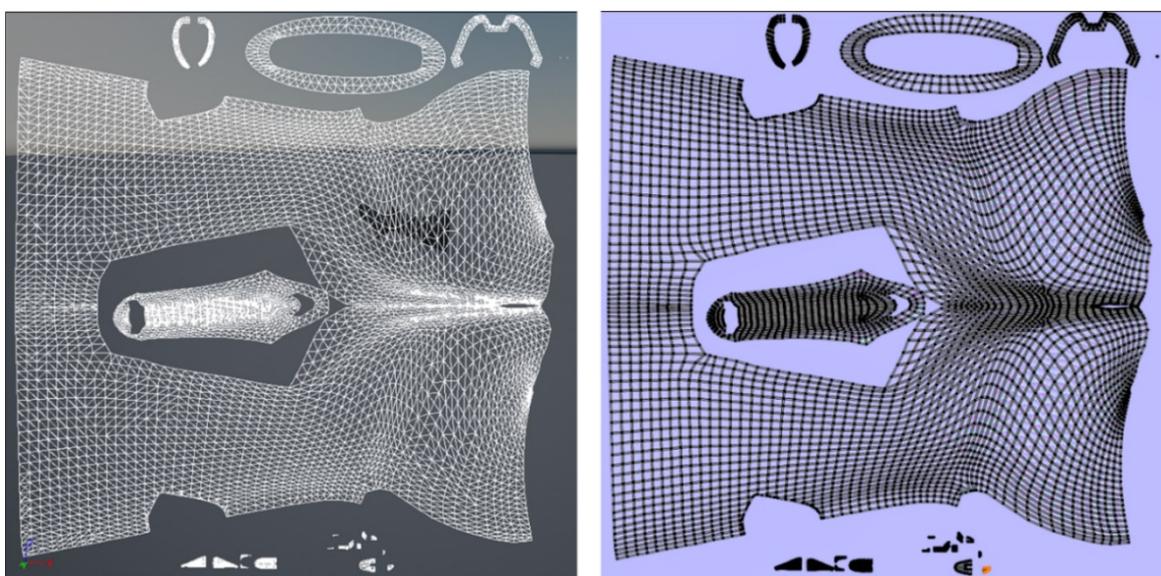


Figura 3-14: Confronto UV-mapping del modello in Unreal (sinistra, triangolarizzato) e quello in Blender (destra, quadrangolare)

Mettendo a confronto le mappature UV dei rispettivi modelli in Blender e Unreal, si è notato che la struttura quadrangolare della mesh del file BLEND si perdeva nel momento in cui si convertiva il file in FBX: dopo tale conversione, infatti, le facce quadrangolari della mesh venivano convertite tutte in triangoli (Figura 3-14).

Per ovviare a tale problema si è pensato di fare in modo che il *Normal Baking* fosse eseguito su un modello con topologia già triangolarizzata, in maniera che la struttura del file FBX fosse analoga a quella del file BLEND.

Il suddetto rimedio ha reso necessario una variazione del workflow: infatti in questo caso non è più possibile eseguire l'esportazione sul modello iniziale, ma è necessario

esportare una copia su cui si sono applicati tutti i modificatori, e quindi triangolarizzare la mesh prima di creare le copie per il *Baking*.

La metodologia definitiva, seguita per il caso studio del prossimo capitolo, è descritta in dettaglio nel flowchart di Figura 3-15.

Osservando il flowchart, è possibile notare come si parta da due files principali: l'*High Poly* importato da CAD e la gabbia precedentemente ottenuta per il componente che si sta rimodellando, più gli oggetti di taglio, anch'essi già presenti nel file di Blender, che fungono da *cutters* per l'applicazione dei modificatori *Boolean*. La gabbia ha già attribuiti a sé tutti i modificatori, inoltre gli oggetti sono probabilmente già in posizione reciproca corretta per cui, se non ci sono problemi di disposizione, è sufficiente cambiare il target dello *Shrinkwrap Modifier* per ottenere il modello a medio livello di dettaglio (primo terminatore rosso nel flowchart).

A questo punto devono essere effettuate, in fasi successive, tre copie del modello ottenuto. La prima rappresenta il file di esportazione: dopo avervi applicato in maniera definitiva e distruttiva il primo modificatore della sequenza, il *Subdivision Surface Modifier*, si può procedere alla triangolarizzazione della mesh (è necessario applicare il modificatore perché l'operazione di triangolarizzazione è logicamente a posteriori dell'azione del primo modificatore, ma a priori rispetto a tutti gli altri).

A questo punto si ha un modello triangolarizzato da cui si può procedere per eseguire il *Normal Baking*: sia per il *Low Poly* che per la *Baking Cage* non è necessario che la geometria sia esatta in corrispondenza delle features di dettaglio medio (cosa invece necessaria per il modello esportabile), perciò è possibile rimuovere tutti i *Boolean Modifiers* per la copia 2. Inoltre, è conveniente rimuovere lo spessore per l'operazione di *Normal Baking* in maniera da avere una superficie semplice. A questo punto si procede creando l'ultima copia che svolgerà la funzione di *Baking Cage*: si ricorda che questa deve essere leggermente più grande del modello *Low Poly* perché deve contenere entrambi gli altri modelli in gioco per l'operazione, per questo motivo si

esegue uno *scaling along the normals* sulla terza copia fino a quando questa non ricopre completamente sia il *Low Poly* che l'*High Poly*.

Si procede quindi con il *Normal Baking* impostando i tre rispettivi attori: una risoluzione di 4196 x 4196 è stata considerata accettabile per ottenere una buona definizione del dettaglio.

Una volta ottenuta la *normal map*, è possibile esportare questa insieme alla prima copia del modello prodotta e, importandole in Unreal Engine in coppia, si procede andando a modificare il materiale applicato all'oggetto-mesh inserendovi, come texture, la *normal map* ottenuta tramite *Baking*.

In Figura 3-16 è possibile visualizzare la resa estetica di modelli di calcio e astina importati in un progetto di Unreal Engine 5.

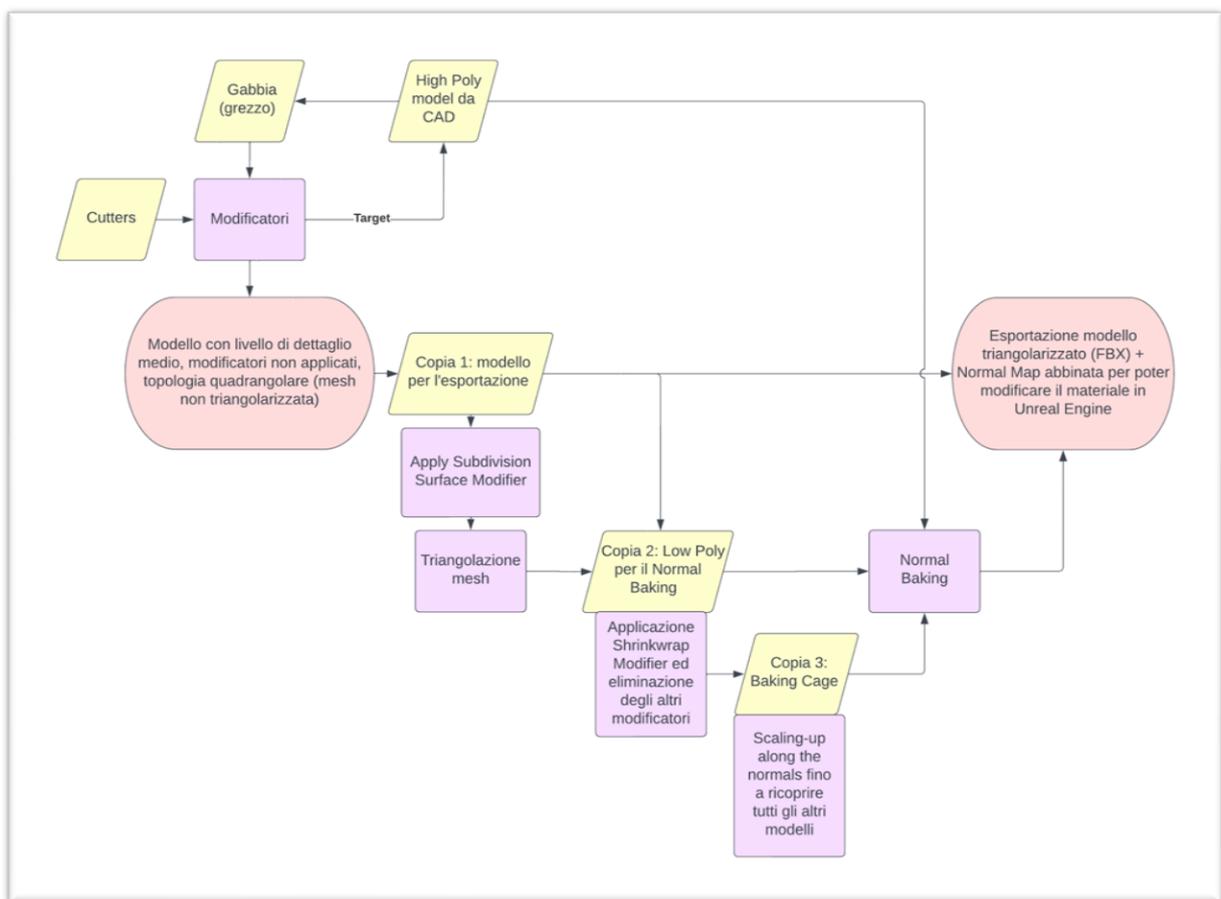


Figura 3-15: Flowchart del processo Shrinkwrap + Normal Baking



Figura 3-16: Risultato modelli di calcio e astina senza zigrino con il metodo descritto in Figura 2-15

4 Casi studio

4.1 Preparazione e accorgimenti

In questo capitolo si affronteranno alcuni casi studio per mettere in pratica il metodo messo a punto e verificarne la rapidità di utilizzo, la flessibilità e l'affidabilità di applicazione, la stabilità del funzionamento, la qualità dei modelli ottenuti.

Sono state elaborate alcune versioni differenti del componente calcio della carabina Lupo, fornite dall'azienda Benelli e mostrate in Figura 4-1, a partire dalla stessa gabbia grezza (o di modellazione). È stato seguito il metodo *Shrinkwrap* per l'ottenimento della geometria, combinato con il *Normal Baking* per il dettaglio di livello avanzato.

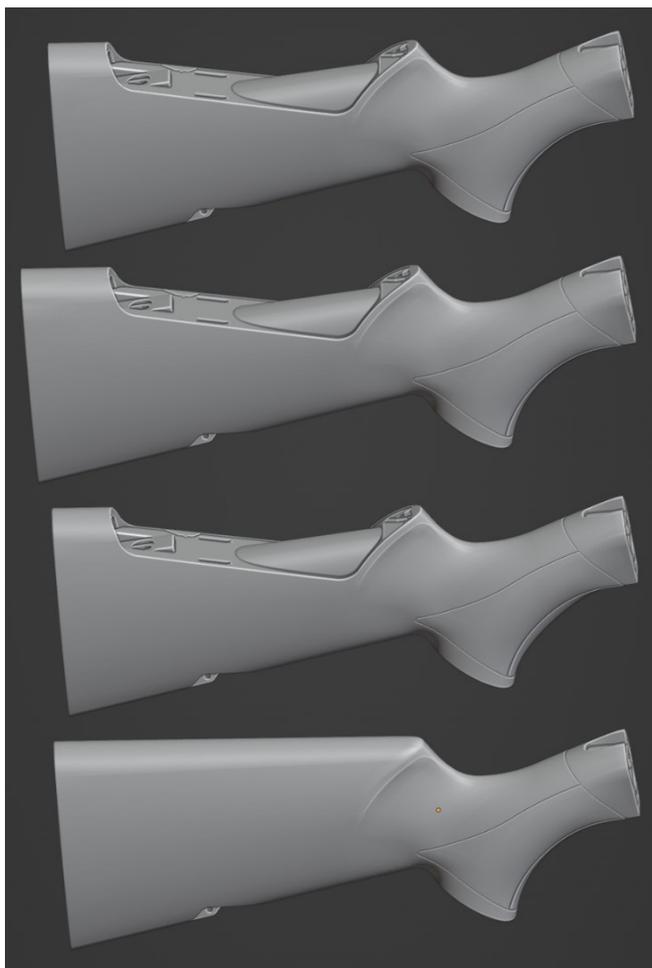


Figura 4-1: Diverse varianti del componente calcio importate come High Poly in Blender

Poiché la gabbia di modellazione deve avvolgere interamente il modello *high poly*, si è pensato di modellare la gabbia sul calcio più lungo per poi adattarla agli altri modelli.

Si è ricostruita la gabbia grezza diverse volte, cercando di capire come gestire al meglio la topologia e la geometria: al fine di ottimizzare la qualità dell'avvolgimento *Shrinkwrap* e minimizzare la deformazione del texturing.

La prima versione di gabbia creata risultava decisamente approssimativa e poco precisa rispetto al modello da avvolgere: come è possibile notare da Figura 4-2. Sebbene l'offset tra le rispettive superfici della gabbia e

dell'*High Poly* fosse decisamente irregolare, la deformazione delle texture sul modello avvolto risultava comunque poco percepibile.

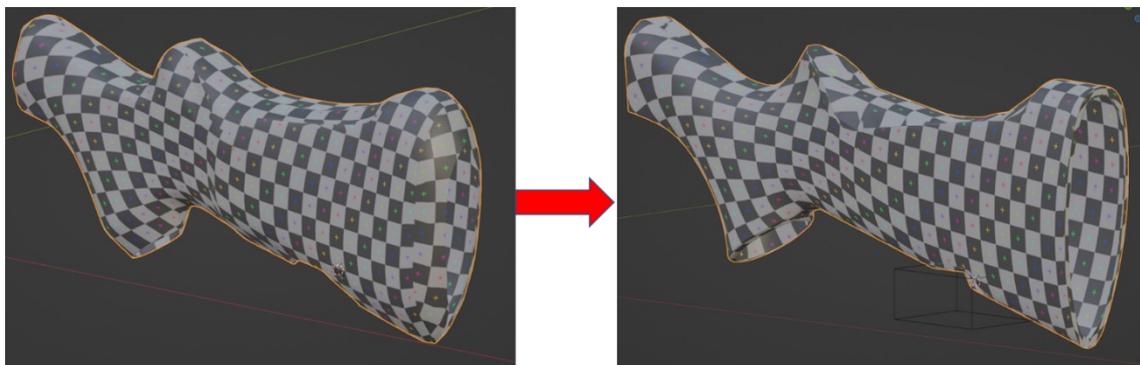


Figura 4-2: Deformazione del texturing dovuta allo *Shrinkwrap*

Tuttavia, vi erano problematiche di instabilità dovute alla eccessiva differenza di dimensione tra i modelli: infatti, quando la distanza tra le superfici è eccessiva, l'algoritmo di proiezione ne risente in precisione e si rischia che vertici distinti della mesh vengano proiettati sullo stesso punto. Per lo stesso motivo si fa anche difficoltà a far coincidere le *seams*, marcate sul modello grezzo, sui bordi corrispondenti nel modello *high poly* da avvolgere.

Si è pertanto ritenuto opportuno fare in modo che la gabbia ricalcasse in maniera fedele, sebbene fosse di dimensione maggiore, la superficie esterna del modello: in modo da stabilizzare l'algoritmo di proiezione del modificatore *Shrinkwrap*. Era pertanto necessario creare una versione *low poly* del calcio che fosse corretta e il più possibile simile alla geometria esterna dell'*High Poly*, eseguendo poi uno *scaling along the normals* per aggiungere un offset costante lungo le superfici dei rispettivi modelli. Dopo diversi tentativi si è arrivati a una versione della gabbia che consentisse di passare rapidamente da un modello all'altro in modo corretto e riducendo in maniera considerevole i problemi di instabilità di proiezione.

4.1.1 *Cutters*

Per il componente calcio, inizialmente, sono stati modellati tre diversi *cutters* da sfruttare per la sottrazione geometrica tramite modificatore *Boolean*, rispettivamente:

- Uno destinato a ritagliare la feature dell'occhiello inferiore per l'allaccio della cinghia

- Uno per lo scavo anteriore per la sede della sicura
- Uno per rifinire l'apertura per l'aggancio del nasello nella parte superiore

Quest'ultimo è stato poi eliminato nell'ultima versione della gabbia perché, gestendo opportunamente la topologia, il risultato era migliore semplicemente sfruttando l'avvolgimento (Figura 4-3).

I *cutters* agiscono applicando dei tagli alla gabbia grezza e andandone ad aumentare localmente la definizione prima dell'avvolgimento, in modo che questa aderisca alla perfezione sul modello anche in caso di geometrie poco morbide.

L'algoritmo è poco stabile e richiede, ogni volta che si cambia minimamente le geometrie in gioco, di riassetare la posizione degli oggetti di taglio rispetto alla gabbia esterna, in modo che non si verifichino problemi di *shading* durante l'avvolgimento.

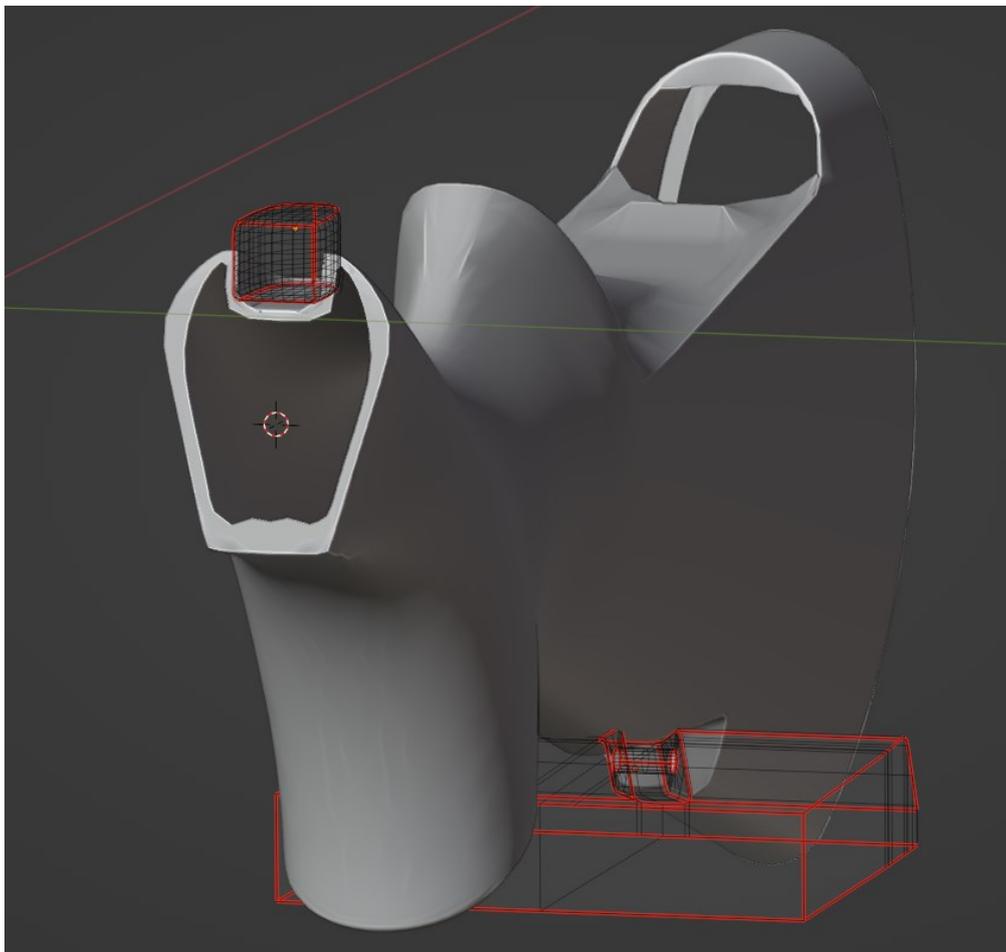


Figura 4-3: Cutters utilizzati per il metodo Shrinkwrap sul componente calcio

Talvolta è anche necessario modificare leggermente la geometria del *cutter* quando si cambia l'assetto: prendendo come esempio quello per l'occhiello, a volte può essere

utile allargare o stringere i fianchi paralleli, o applicare delle deformazioni per alzare o abbassare la parte posteriore.

4.2 Analisi tempi per gli aggiornamenti di modello del componente calcio

Il primo caso studio prevede lo scopo di testare l'applicabilità e la flessibilità di utilizzo del procedimento.

Misurando opportunamente il tempo necessario per ogni singola operazione, sono stati simulati degli aggiornamenti di modello provando le combinazioni vecchia-nuova versione sulle tre varianti di calcio senza nasello (Figura 4-1); la quarta versione è stata esclusa dalla simulazione in quanto avente una geometria parecchio differente, e quindi richiederebbe una gabbia apposita.

Le tre versioni sfruttate per la simulazione saranno chiamate rispettivamente:

- a) *“Calcio Base”*: versione utilizzata come riferimento.
- b) *“Calcio Lungo”*: variante con pala allungata.
- c) *“Calcio Imp”*: versione con impugnatura di geometria leggermente differente.

Sono stati misurati i tempi di processo per aggiornare la versione del modello virtuale da una ad un'altra delle varianti di modelli *high poly* sopra elencate, provenienti dall'ambiente CAD, per un totale di sei combinazioni differenti vecchio-nuovo.

A tale scopo sono stati predisposti tre files BLEND con i rispettivi *High Poly* di partenza e, nel seguire la procedura descritta dal flowchart in Figura 3-15, sono stati calcolati i tempi di tutte le operazioni.

La prima parte del workflow riguarda il processo di ottenimento della geometria UV-mappata del nuovo modello da cui si ottiene l'esportabile: essa comprende le fasi di modifica e importazione del modello *high poly*, posizionamento, cambio target dei modificatori, assestamenti delle estremità e dei *cutters*, generazione e la modifica delle tre copie funzionali al *Normal Baking* successivo. La seconda macrofase, invece, riguarda l'ottenimento automatico della *normal map* dei dettagli superficiali da

applicare al modello da esportare. Infine, la terza macrofase riguarda l'esportazione di modello e *normal map* di dettaglio abbinata, e la pulizia di quest'ultima nel software di image editing.

Si è visto che, per rendere meglio visibili i dettagli a fianchi paralleli stretti, come le scritte incise, è utile applicare un lieve smusso alla scanalatura in modo che il dettaglio sia rilevato in maniera più regolare: tali operazioni sono state eseguite prima dell'importazione (quindi in ambiente CAD) o subito dopo.

Nel Capitolo 5 si analizzeranno i risultati della prova riportando i tempi delle tre macrofasi del processo di ogni test effettuato. Sono stati eseguiti tre test per ognuna delle sei combinazioni, per un totale di 18 test.

4.3 Analisi di dettaglio di tempi per modelli con zigrino di calcio e astina

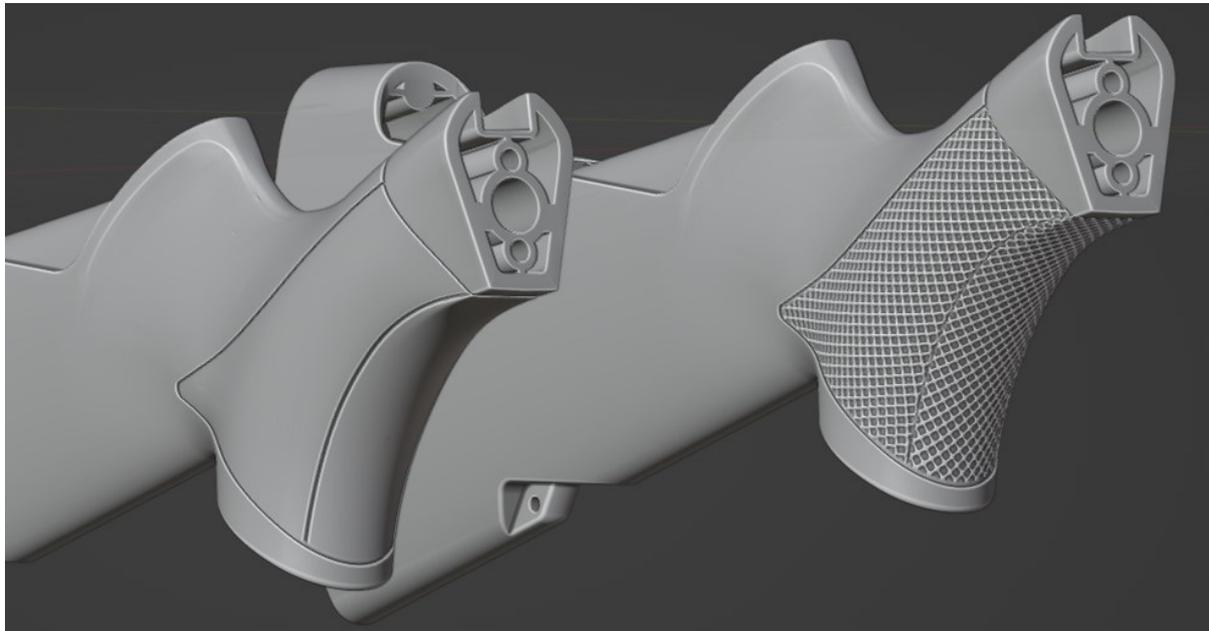


Figura 4-4: Confronto tra modelli di calcio *high poly* con zigrino e senza

I modelli di astina e calcio su cui è stato sviluppato il metodo di modellazione rapida non sono gli originali di serie per la carabina Lupo. Questi, infatti, sono stati semplificati con l'obiettivo che non si ottenessero degli *High Poly* troppo pesanti da

dover elaborare all'interno del software di modellazione poligonale: in particolare, i modelli autentici presentano degli zigrini nelle aree superficiali comprese tra le scanalature (Figura 4-4).

Il processo di modellazione manuale che viene seguito al momento rende conveniente modellare il calcio prendendo come riferimento il file senza zigrino, per poi aggiungere questa feature in un secondo momento. Tuttavia, dato che il metodo messo a punto richiede un'interazione alquanto ridotta con il modello *high poly*, si è ritenuto possibile applicare il procedimento direttamente alle versioni con zigrino: ciò non avrebbe comportato alcuna variazione del workflow nella prima parte, mentre questi dettagli superficiali sarebbero stati rilevati tramite il *Baking* e incorporati nella *normal map* senza un significativo incremento del tempo di processamento.

Sono stati perciò forniti dall'azienda Benelli dei files STEP di calcio e astina con zigrino, allo scopo di testare l'efficacia del metodo direttamente sui modelli zigrinati.

Tra questi, per questo secondo test, sono stati impiegati come *High Poly* due files contenenti i rispettivi modelli base di astina e calcio: questi sono stati sottoposti allo stesso processo seguito nel test precedente per confrontare eventuali significative variazioni nelle tempistiche.

Si è partiti quindi dai modelli non zigrinati per passare a quelli con lo zigrino; anche in questo caso sono stati acquisiti tre test per ognuno dei due processi, stavolta per un totale di sei; inoltre sono stati acquisiti i tempi precisi delle singole sottofasi attraverso delle riprese video durante lo svolgimento delle operazioni.

4.4 Modellazione da zero del componente carcassa

Come ultimo test di collaudo per il metodo sviluppato, è stato chiesto alla Benelli di condividere un altro componente che avesse delle caratteristiche simili agli altri due affrontati precedentemente, in particolare:

- Forma geometrica complessa
- Ricchezza di dettagli superficiali

- Necessità di aggiornamento frequente

Il componente appartenente alla carabina che aveva queste caratteristiche, e che quindi è stato selezionato per la prova di modellazione da zero, è la carcassa: parte di collegamento tra calcio ed astina.

Il test di modellazione da zero è stato eseguito con lo scopo di avere un'idea del tempo necessario, alla luce dell'esperienza acquisita con la modellazione delle gabbie precedenti, per ottenere il "grezzo" di un nuovo componente di geometria complessa il quale, sottoposto a modifiche frequenti, potesse essere utilizzato per ottenere rapidamente il modello virtuale: come è stato provato per i test di aggiornamento su calcio e astina.

Si è pensato che, per creare un BLEND che fosse flessibile a modifiche più o meno radicali, fosse opportuno creare una prima versione grossolana della gabbia, per poi applicarne da una copia un primo *Shrinkwrap* ed ottenere così una versione "fine" che ricalcasse la geometria in maniera più fedele.

Quest'ultima sarebbe adeguata ad un aggiornamento del modello con variazioni di dettaglio, mentre in caso di modifiche geometriche importanti sarebbe meglio ripartire dalla gabbia grossolana e ricavarne nuovamente la versione fine, con cui poi si andranno a costituire le varie copie per il *Baking* ed esportazione.

Il tempo impiegato per la creazione della gabbia, con tutte le correzioni iterative per una resa estetica soddisfacente, è stato di 7 ore e 40 minuti circa; questo tempo comprende:

- Progettazione della forma della gabbia da avvolgere
- Modellazione di una gabbia grossolana
- Prove ricorsive sulla resa e modifiche
- Generazione di copia fine da quella grossolana
- Identificazione delle zone di dettaglio medio da rifinire con *cutters*
- Design dell'unico *cutter* impiegato
- Prova di *Baking*: con controllo della resa estetica del materiale e della *normal map* di dettaglio
- Prove esportazioni e successive modifiche

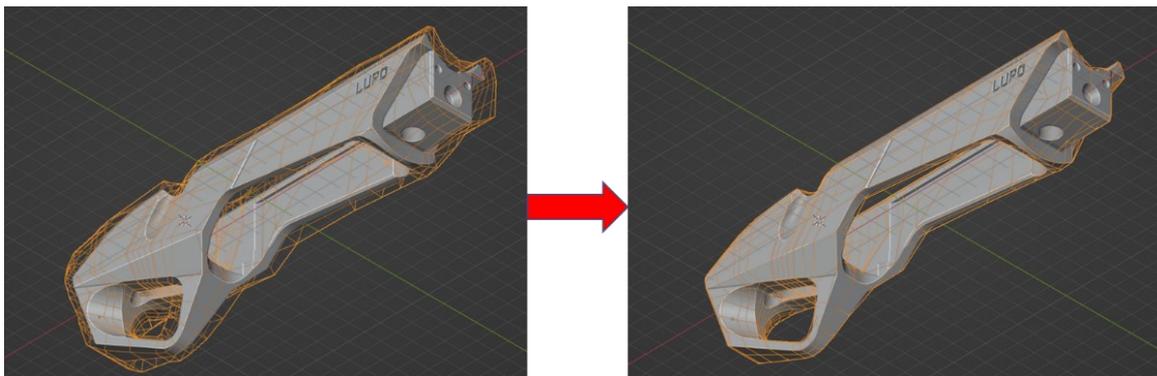


Figura 4-5: Avvolgimento della gabbia di modellazione sull'High Poly della carcassa

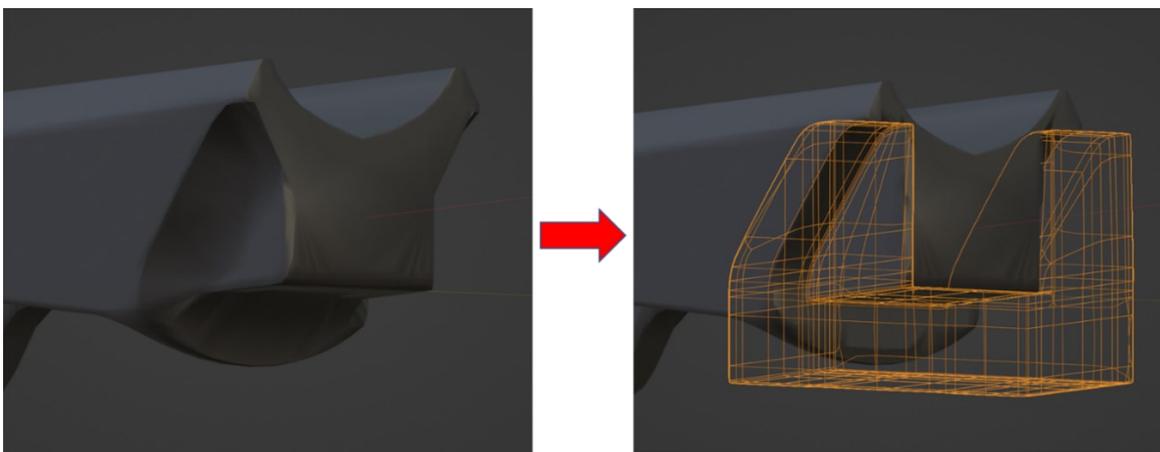


Figura 4-6: Rifinitura feature geometrica all'estremità della carcassa tramite Boolean



Figura 4-7: Render del modello carcassa ottenuto in Blender

5 Risultati e discussione

5.1 Premessa

In questo capitolo sono riportati i risultati degli studi descritti nella sezione precedente, comprendendo nella trattazione delle considerazioni circa la rapidità nell'ottenimento dei modelli finali e la loro qualità: sia in termini estetici, che di leggerezza computazionale per le applicazioni a cui questi sarebbero volti.

I test effettuati sono stati svolti da una persona singola e rappresentano un campione poco significativo dal punto di vista statistico: sia perché il numero dei sample di processo risulta abbastanza limitato, sia perché si reputa che il campione non sia rappresentativo della varietà di componenti a cui, in condizioni operative, potrebbe essere applicato. Pertanto, si evita di effettuare delle considerazioni statistiche sui tempi di processo acquisiti e ci si limita a stime qualitative.

5.2 Tempi di ottenimento

Tabella 5.1: Risultati relativi al primo caso studio: tempi indicativi di esecuzione delle macrofasi di processo tra le varie versioni di calcio fornite

	Numero prova	Base ↓ Lungo (min)	Base ↓ Imp (min)	Lungo ↓ Base (min)	Lungo ↓ Imp (min)	Imp ↓ Base (min)	Imp ↓ Lungo (min)
Preparazione modelli e copie per il baking	Test 1	10	9	10	11	9	11
	Test 2	8	9	9	9	9	9
	Test 3	8	8	8	9	8	8
Normal baking	Test 1	10	9	9	9	9	10
	Test 2	10	9	9	9	9	10
	Test 3	10	9	9	9	9	10
Esportazione, image editing, check	Test 1	6	5	5	5	5	5
	Test 2	5	5	5	5	5	5
	Test 3	5	5	5	5	5	5
Tmedio TOT		24.0	22.7	23.0	23.7	22.7	24.3

Tabella 5.2: Risultati relativi al secondo caso studio: tempi di esecuzione fasi di processo da base a zigrinato per calcio e astina

	Calcio con zigrino: durata operazioni (min)			Astina con zigrino: durata operazioni (min)		
	Test 1	Test 2	Test 3	Test 1	Test 2	Test 3
1.Importazione high poly e modifiche	5.2	4.9	4.8	8.3	8.1	8.1
2.Posizionamento	2.2	1.9	1.9	2.5	2.1	2.0
3.Cambio target Modif. Shrinkwrap	0.2	0.2	0.2	0.2	0.2	0.2
4.Aggiustaggi estremità	0.7	0.6	0.6	X	X	X
5.Aggiustaggi cutters	2.2	1.8	1.8	2.6	2.1	1.9
6.Generaz. copia 1 (esportabile) e operaz.	0.3	0.3	0.3	0.3	0.3	0.3
7.Generaz. copia 2 (low-poly) e operaz.	0.2	0.3	0.2	0.2	0.2	0.2
8.Generaz. copia 3 (baking cage) e operaz.	0.8	0.7	0.7	0.7	0.7	0.7
9.Normal Baking	8.7	8.7	8.8	7.2	7.3	7.2
10.Esportazione	0.9	1.3	1.2	1.2	1.1	0.9
11.Modifica normal map	2.8	2.9	2.6	3.4	3.7	3.3
12.Controllo qualità	0.8	0.7	0.8	1.1	1.0	1.1
TOT	25.0	24.3	23.9	27.7	26.8	25.9

Analizzando la Tabella 5.1, si nota come non ci siano radicali variazioni in termini di tempo di esecuzione delle singole macrofasi da un modello all'altro, e ancor meno per i test relativi alla medesima combinazione vecchio-nuovo modello.

Confrontando i tempi relativi al calcio della seconda prova effettuata (visibili nella parte sinistra di Tabella 5.2) con quelli della prima prova, è possibile notare che le medie dei tempi di ottenimento totali, per i modelli con e senza zigrino, sono confrontabili.

Come era stato ipotizzato non ci sono aumenti significativi di tempo di processamento per i modelli con zigrino rispetto agli analoghi alleggeriti. Il processo di *Normal Baking* non sembra infatti risentire della maggiore pesantezza del modello *high poly*, ma si rileva un leggero incremento del tempo di modellazione (cioè fino al *Baking* escluso), che però risulta mediamente solo intorno al 10% sulle prove effettuate.

È anche opportuno puntualizzare che i modelli con zigrino, in confronto ai rispettivi equivalenti semplificati, sono stati esportati con sistema di riferimento e punto di origine geometrica differenti: è stato pertanto necessario spostarli e ruotarli

manualmente fino a farli combaciare con la posizione dell'altro *High Poly* per l'avvolgimento della gabbia. Facendo attenzione a standardizzare almeno l'orientazione del sistema di riferimento tra le varie versioni CAD esportate, si abbatterebbe quindi il tempo di posizionamento (step 2 nella Tabella 5.2).

Un'altra fase critica del processo è rappresentata dagli "aggiustaggi" dei *cutters* per l'applicazione dei modificatori *Boolean* (step 5). La ragione di ciò è nel fatto che il risultato dell'effetto dei modificatori di *Boolean* e *Shrinkwrap* in serie è decisamente poco prevedibile: motivo per cui, in caso di cambio della topologia o della posizione reciproca degli elementi, vanno applicati degli assestamenti di posizionamento e di geometria sui *cutters*. Nello svolgere queste operazioni, il software ricalcola ciclicamente ad ogni frame tutti gli effetti della catena dei modificatori in base alle modifiche applicate perciò, con l'Hardware utilizzato per lo studio, ogni singolo frame nel processo di modifica può durare anche alcuni secondi.

Essendo l'operazione in questione vincolata all'esperienza di chi esegue l'aggiustaggio, è per ora un'operazione scarsamente automatizzabile e con bassa ripetibilità.

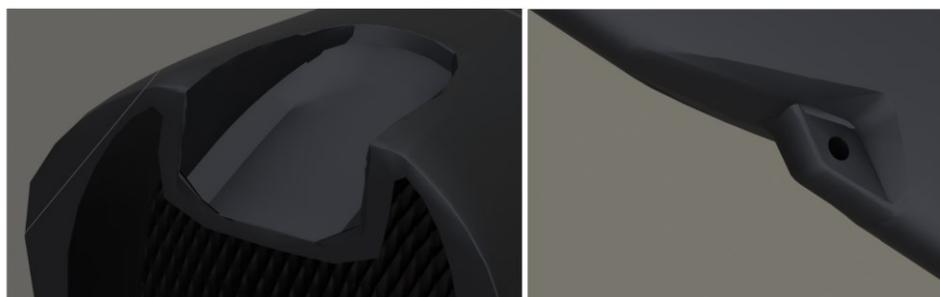


Figura 5-1: Difetti locali nelle aree sottoposte a cutting tramite modificatore Boolean

5.3 Qualità dei modelli

Per quanto riguarda la qualità dei modelli, è importante sottolineare che le porzioni superficiali rifinite tramite modificatore *Boolean*, come mostrato dalle viste di dettaglio di Figura 5-1, sono talvolta tutt'altro che perfette: tali difetti superficiali potrebbero essere considerati tollerabili soltanto perché sono relativi a una minima parte

dell'estensione geometrica del modello, oltre a trovarsi in corrispondenza di zone non molto in vista. Se si volesse fare in modo di rimuoverli per ottenere un modello perfetto, bisognerebbe prendere in considerazione l'idea di svolgere un lavoro di rifinitura manuale in corrispondenza delle aree della mesh in questione prima dell'esportazione.

Confrontando invece i dati del calcio con quelli dell'astina, si nota immediatamente come i tempi di elaborazione dell'*High Poly* di quest'ultima siano decisamente maggiori: ciò è imputabile al fatto che su questo componente è necessario apportare una modifica all'incisione del logo aziendale, non presente nel calcio, all'interno di Blender. Al contrario di quanto si fa per la scanalatura, infatti, non è possibile aggiungere lo smusso direttamente su NX prima dell'importazione in Blender, ma è necessario elaborarla dopo la conversione in STL apportando due modifiche:

- Aumento della profondità dell'incisione
- Applicazione di una rastrematura alle scritte incise in modo che i fianchi risultino leggermente inclinati

Tali modifiche migliorano la qualità della rilevazione del dettaglio sulla *normal map* ottenuta tramite *Baking*, ma comportano due problematiche:

1. Richiedono un intervento manuale su un modello molto pesante: perciò può essere richiesta molta potenza di calcolo
2. La scritta incisa, acquisita tramite *Normal Baking*, risulta comunque imprecisa e poco definita, come visibile nella parte sinistra di Figura 5-3

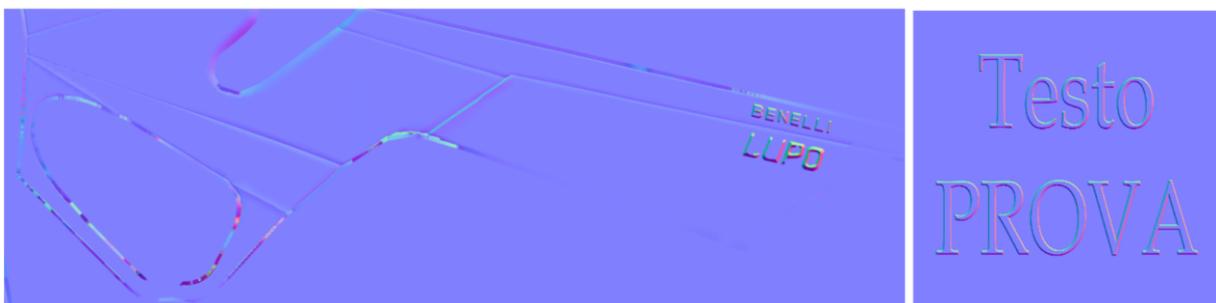


Figura 5-2: Porzione della *normal map* in cui vi è l'incisione del logo (sinistra); sample di sostituzione ottenuto in GIMP (a destra)

Dato che il dettaglio relativo alla scritta incisa sulla *normal map* acquisita tramite *Baking* è circoscritto, facilmente individuabile e sostanzialmente indeformato, potrebbe essere

sensato pensare di aggiungere il dettaglio direttamente come operazione di post processing sulla texture dopo il *Baking*, preparando adeguatamente le ROI delle *normal maps* da sovrapporre (è possibile generarle di qualsiasi profondità attraverso una procedura di *Baking* in Blender o di *image processing* in GIMP come mostrato in Figura 5-2).



Figura 5-3: Confronto tra qualità della scritta incisa se ottenuta con *baking* semplice (sinistra), o sostituita (destra)



Figura 5-4: Confronto visibilità della discontinuità in corrispondenza della *seam* sull'impugnatura del calcio con materiali differenti: A sinistra: materiale disomogeneo, *seam* molto visibile; Al centro: discontinuità poco visibile; A destra: materiale omogeneo, *seam* non visibile

Un altro aspetto rilevante per l'ottenimento di modelli fotorealistici è rappresentato dalla necessaria presenza delle *seams* per l'UV Mapping le quali, in genere,

determinano una discontinuità nella texture applicata. Non sempre è possibile, infatti, fare in modo che queste coincidano con aree del modello non in vista. Nel caso in cui non sia possibile, la resa estetica dipende molto dal materiale: un'eventuale disomogeneità nel pattern della texture renderebbe questa meno adeguata all'utilizzo se il modello presentasse delle *seam* esposte (Figura 5-4).

Tabella 5.3: Dati di vertici e facce relativi ai modelli iniziali e finali

		High Poly iniziale (senza zigrino)	High Poly iniziale (con zigrino)	Modello finale
CALCIO	Vertici	105 535	1 729 179	16 580
	Facce	210 986	3 458 273	20 888
CARCASSA	Vertici	27 869	X	27 001
	Facce	55 762	X	37 331
ASTINA	Vertici	54 384	694 165	8 859
	Facce	108 780	1 388 341	11 372

Per un agile funzionamento dell'applicazione grafica, come illustrato da Ming Tang Y. et al. (2020), [11], è opportuno che l'ammontare di vertici e facce del modello poligonale sia ridotto al minimo. Come è visibile dalla Tabella 5.3, la riduzione della complessità del modello poligonale finale è notevole nei casi di calcio e astina, di entità minore per il componente carcassa.

Questo dato è imputabile alla geometria dei modelli. Sia calcio che astina sono componenti la cui superficie esterna è prevalentemente a curvatura dolce: questi due componenti presentano infatti pochi elementi come spigoli, fianchi stretti paralleli, o comunque cambi bruschi di curvatura. Inoltre, la loro geometria è stata semplificata in maniera da eliminare la geometria interna.

Nel caso della carcassa, si è scelto invece di modellare il componente a geometria non semplificata per testare l'efficacia del metodo su strutture complicate e difficilmente "avvolgibili": il risultato mostra come, per applicare efficacemente il metodo, in questo

caso sia stato necessario aumentare la risoluzione della gabbia a tal punto da rendere confrontabile il numero di facce e vertici del modello finale con quello dell'*High Poly* generato da CAD.

Ciononostante, si reputa che la modellazione per avvolgimento, se i difetti superficiali locali mostrati in precedenza fossero considerati accettabili, possa essere considerata anche in questo caso preferibile all'UV Mapping diretto dell'*High Poly* importato: infatti la marcatura delle *seams*, quando svolta direttamente su una geometria complessa come quella importata da CAD, può risultare un'operazione estremamente lunga e laboriosa.

Per concludere il capitolo si mostrano alcuni renders, ottenuti in un template project di *product configuration* in Unreal Engine 5 opportunamente modificato, raffiguranti l'assieme dei modelli finali comprensivi di zigrino di calcio, carcassa e astina.

I materiali applicati all'assieme sono stati in parte forniti da Benelli (base color dei legni) e in parte ottenuti da librerie online open source, nel caso specifico *Quixel Megascans*.

5.4 Galleria



Figura 5-5: Renders di assieme calcio, carcassa, astina in Unreal Engine 5: Legni A, B



Figura 5-6: Renders di assieme calcio, carcassa, astina in Unreal Engine 5: Legni C, D, E

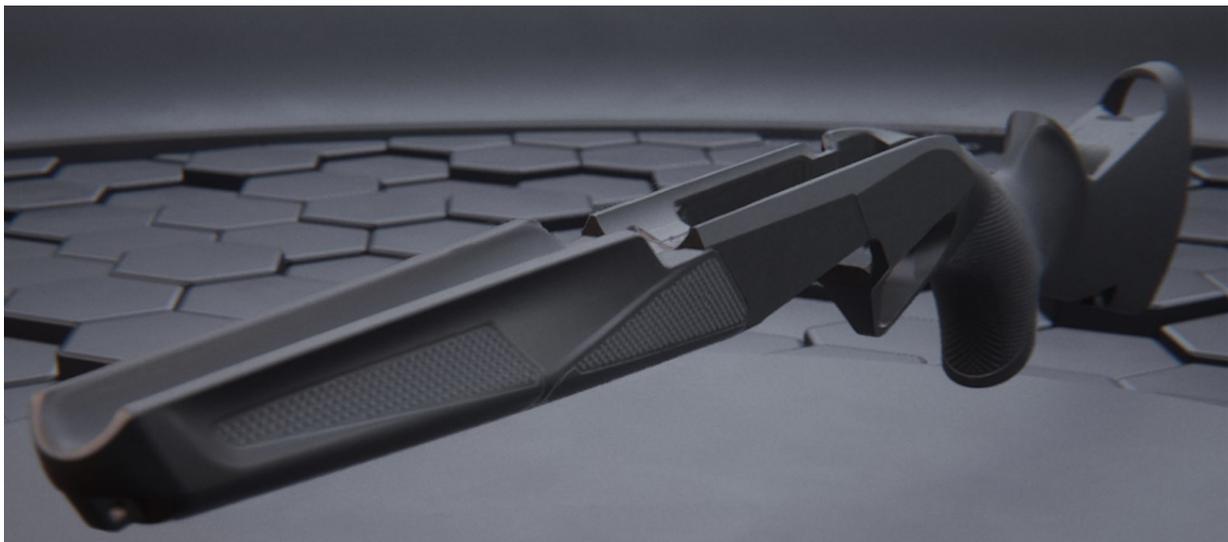
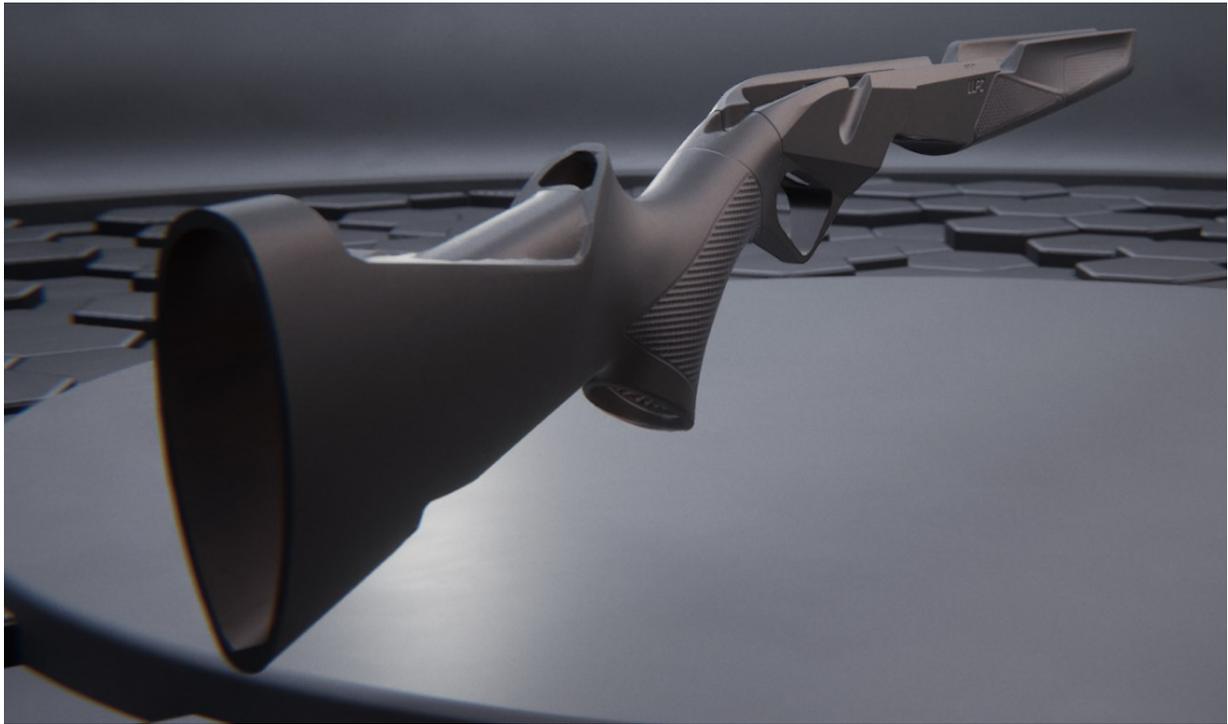


Figura 5-7: Renders di assieme di calcio, carcassa, astina in Unreal Engine 5: Polimero

6 Conclusioni

I risultati ottenuti, sia in termini di velocità d'esecuzione sia in termini di qualità, sono stati reputati più che soddisfacenti grazie alla riproduzione fedele della geometria esterna delle tre tipologie di componenti affrontati: cioè calcio, carcassa ed astina della carabina *Lupo*.

I tempi di aggiornamento modello, nel caso dei casi studio effettuati, sono stati considerati decisamente inferiori rispetto alla procedura alternativa di rimodellazione manuale semplice: per cui la scelta se adottare o no il metodo proposto risiederebbe nei seguenti aspetti:

- **La complessità del modello da ottenere:** per componenti molto semplici la modellazione manuale potrebbe essere la strada migliore
- **L'entità della modifica rispetto al vecchio modello ottenuto:** per piccole modifiche potrebbe essere conveniente mettere mano al vecchio modello precedentemente esportato piuttosto che rigenerarlo da zero per avvolgimento (se l'aggiornamento riguardasse un dato di dettaglio superficiale, si potrebbe procedere direttamente con la fase di *Normal Baking* per riprodurre le *normals* di dettaglio)
- I modelli prodotti tramite modellazione *Shrinkwrap* dovrebbero avere una **qualità** reputata sufficiente rispetto agli analoghi ottenuti tramite modellazione manuale
- **Questioni tecniche** relative alle strutture dati **dell'applicazione** di Mixed Reality sviluppata

A proposito di questi aspetti, sarebbe certamente utile un lavoro di squadra con esperti di modellazione 3D e di programmatori informatici per confrontarsi sulla possibilità di migliorare alcune soluzioni implementate nel metodo allo scopo sia di incrementare la qualità dei modelli, sia di aumentare il grado di automazione del processo.

Sarebbe possibile anche sviluppare un codice Python che lavori con i files BLEND relativi ai componenti per svolgere il processo da riga di comando: possibilmente addirittura eliminando totalmente, o quasi, il lavoro manuale interno al software Blender.

Un altro aspetto critico è quello dei materiali utilizzati nell'applicazione di Mixed Reality: ai modelli creati per questa tesi sono state applicate textures standard quadrate. Sfruttando invece delle textures ottenute proceduralmente sarebbe possibile, ipoteticamente, generarle su misura per l'UV mapping del modello creato. Questo consentirebbe di evitare di avere discontinuità in corrispondenza delle *seams* le quali, con textures standard, risultano più o meno visibili a seconda del materiale applicato e dell'ubicazione.

Inoltre, per un maggior fotorealismo delle textures adoperate, sarebbe interessante sfruttare algoritmi di intelligenza artificiale o textures a strutture dati tridimensionali per correlare il pattern delle textures, ad esempio delle venature nel caso dei legni, alla curvatura superficiale del modello.

7 Bibliografia

- [1] Bhugaonkar K, Bhugaonkar R, Masne N (September 12, 2022) *The Trend of Metaverse and Augmented & Virtual Reality Extending to the Healthcare System*. Cureus.
- [2] Fillmore, H., & Storr, T. (2020). *AR and VR in the workplace: Extended reality reimagines how work is done*. IBM.
- [3] Bye, K. (2022, July 6). *IEEE Metaverse Congress: Benefits & Risks of the Metaverse from the ESG Perspective*, Institute of Electrical and Electronics Engineers Standards Association.
- [4] *Comunicati Stampa*. Regione Marche. Articolo del 11/02/2021. Disponibile su <https://www.regione.marche.it/In-Primo-Piano/ComunicatiStampa?id=29908>
- [5] Wenz, S., 2021. *Audi Business Innovation drives industrial-scale visualization at VW Group*. Unreal Engine. Disponibile su <https://www.unrealengine.com/en-US/spotlights/audi-business-innovation-drives-industrial-scale-visualization-at-vw-group> [Visitato il 26/09/2022].
- [6] Conlan, C. (2017). *The Blender Python API*. Apress.
- [7] *Maya 6: NURBS Modeling (Version 6)*. (2006). Silicon Graphics.
- [8] Zachar, J., Horňák, M., & Novaković, P. (2017). *3D digital recording of archaeological, architectural and artistic heritage*. CONPRA Series.
- [9] Liao, W., Liu, H., & Li, T. (2017). *Subdivision Surface Modeling Technology*. Springer Singapore.
- [10] Docs.blender.org. 2022. *Introduction – Blender Manual*.
- [11] Ming Tang, Y., & Ho, H. L. (2020). *3D Modeling and Computer Graphics in Virtual Reality*. In *Mixed Reality and Three-Dimensional Computer Graphics*. IntechOpen.
- [12] Caudron, R., & Nicq, P. (2015, September 25). *Blender 3D By Example: Design a complete workflow with Blender to create stunning 3D scenes and films step-by-step*. Packt Publishing.
- [13] Powell, A. W. (2010). *Blender 2.5 lighting and rendering: Bring your 3D world to life with lighting, compositing, and rendering*. Packt Publishing.

- [14]H. Y. Pai (2019) "*Texture designs and workflows for physically based rendering using procedural texture generation*". 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)
- [15]Venter, H., & Ogterop, W. (2022, June). *Unreal Engine 5 Character Creation, Animation and Cinematics: Create Custom 3D Assets and Bring Them to Life in Unreal Engine 5 Using MetaHuman, Lumen, and Nanite*. Van Haren Publishing.
- [16]Nicky L Webster (2017) *High poly to low poly workflows for real-time rendering*, Journal of Visual Communication in Medicine
- [17]*Le nuove professioni del metaverso*. ClicLavoro. (2022). Visitato il 27/09/2022 <https://www.cliclavoro.gov.it/page/le-nuove-professioni-del-metaverso?contentId=BLG17356>