POLITECNICO DI TORINO

Corso di Laurea
in Ingegneria Informatica

Tesi di Laurea

A machine learning approach for avatar reconstruction from RGB data

Relatori
prof. Federico Manuri
prof. Andrea Sanna

Candidato
Gioele Ramondetti

Anno Accademico 2023-2024

† A mia nonna Giovanna
e a mio nonno Francesco

Sommario

Il lavoro di questa tesi ha come obiettivo la realizzazione di un avatar 3D animabile partendo da una singola immagine RGB. Il dato fornito in input viene utilizzato per la generazione del vestiario e dei capelli. Sfruttando le informazioni presenti nell'immagine, si possono ottenere dettagli aggiuntivi, come la presenza di barba e baffi e il colore della pelle. Per raggiungere questi risultati, si impiegano diverse tecniche di computer vision e modelli di deep learning. L'esecuzione dei modelli genera oggetti tridimensionali utilizzabili sul modello per ottenere un risultato il più verosimile possibile all'immagine di riferimento. Prevedendo la possibilità che il riconoscimento e la conseguente ricostruzione possano fallire o portare a risultati poco soddisfacenti, si permette di sostituire l'oggetto con un altro della stessa tipologia, scelto da una libreria predefinita. Per validare la qualità del lavoro svolto, è stata adottata come riferimento una mesh ricostruita priva e comprensiva di vestiario, sottoposta a un'animazione di camminata. I valori delle metriche SPD, $SWSPD$, $DMPD$ sono stati calcolati su diverse tipologie di modelli e successivamente mediati, al fine di ottenere metriche di riferimento affidabili e oggettive per il confronto. Successivamente, il calcolo è stato ripetuto su diverse mesh comprensive di vestiario, utilizzando la medesima animazione per garantire coerenza nell'analisi. Completato il processo, si ottiene un modello umano 3D completo, animabile e personalizzabile, consentendo eventuali modifiche nel caso in cui il risultato finale non soddisfi le aspettative.

Ringraziamenti

Ringrazio di cuore tutte le persone che, con la loro presenza e il loro sostegno, hanno reso possibile la realizzazione di questo lavoro e mi hanno accompagnato durante questi anni di studi. Ringrazio mia mamma Luisa, mio papà Dionigi e mio fratello Marco, che nella mia vita, così come nello sport, sono stati i miei tifosi numero uno, facendomi sentire sempre a mio agio, soprattutto nei momenti più difficili. I pranzi e le cene insieme, che portano sempre a grosse risate, sono per me una grande fonte di felicità. Un pensiero speciale va ai miei nonni paterni Francesco e Giovanna che sono sempre stati i primi a dirmi "mi raccomando studia!" e sarebbero orgogliosi di questo traguardo forse di più di quanto lo sono io, questa tesi è dedicata a voi. Ringrazio tutti i miei parenti che sono un pilastro fondamentale nella mia vita e non mi hanno fatto mai mancare nulla soprattutto il supporto nei momenti più duri soprattutto nonno Andrea e nonna Piera che mi hanno cresciuto fin da piccolo e continuano a crescermi anche ora. A Giada, che è stata un punto di riferimento e mi ha dimostrato che è possibile combattere e vincere contro nemici invisibili che all'apparenza non è possibile contrastare ma sopprattutto per avermi insegnato il valore dell'amore. Un grazie speciale va ad Alessandro, Dario, Federico, Luana, Kevin, Noemi e Sante, che hanno reso il percorso della magistrale più leggero e piacevole, condividendo successi e traguardi con allegria; auguro loro tutto il successo che meritano. Un ringraziamento va ad Arianna, Christian, Gianluca, Luca, Madalin e Matteo, che hanno contribuito alla mia crescita personale e al mio miglioramento; senza di loro probabilmente non avrei raggiunto questo traguardo. Un pensiero speciale anche per Alessio, Anna, Marianna, Martina, Sara e Vittoria, che sono sempre stati presenti, nonostante la difficoltà di vedersi spesso. Le cene e le gite insieme sono sempre state una fonte incredibile di vita per me. A Paolo, compagno di studi, coinquilino, chef e terzino di fiducia, con cui ho condiviso tutto ciò che era possibile condividere in questi anni. Sei una fonte di ispirazione costante facendo emergere il meglio di me stesso e riesci sempre a farmi sentire a casa in qualsiasi situazione, sei l'amico che tutti vorrebbero. Un ringraziamento ad Alberto, che è come un fratello, quando mi vedeva in difficoltà, mi obbligava a uscire di casa e a prendermi una pausa. La sicurezza di poter contare su di lui è impagabile. Un ringraziamento speciale va al mondo della pallacanestro, che mi ha dato tanto e ha rappresentato e rappresenterà sempre un rifugio importante. Sarei impazzito senza le ore passate in palestra, insieme alle meravigliose persone incontrate in questo sport. In particolare, ringrazio tutti gli amici con cui condivido le ore di allenamento, amo trascorrere anche il mio tempo libero con voi non siete solo dei colleghi ma molto di più. Infine, ringrazio i miei relatori per il supporto e la disponibilità dimostrati durante

il periodo di sviluppo del progetto e di stesura della tesi.

Indice

Elenco delle tabelle

Elenco delle figure

Se non credi in te stesso,
chi ci crederà?

Capitolo 1

Introduzione

La creazione di avatar 3D realistici rappresenta una frontiera significativa nel campo della computer grafica, con applicazioni che spaziano dall'intrattenimento e dai videogiochi, fino a settori avanzati come la realtà virtuale, la realtà aumentata e il metaveso. Grazie all'evoluzione tecnologica e all'espansione del concetto di metaverso, gli avatar digitali stanno acquisendo un'importanza crescente come rappresentazioni virtuali degli individui, capaci di incarnare caratteristiche fisiche e stilistiche personalizzate. La possibilità di costruire e possedere un proprio avatar all'interno di un mondo virtuale apre nuove opportunità di espressione personale, dove gli utenti possono interagire in ambienti differenti da quelli reali, modificando liberamente l'aspetto e le caratteristiche del proprio alter ego digitale per adattarlo ai propri desideri e bisogni. Questi avatar trovano utilizzo in diversi contesti: oltre al gaming, dove rappresentano un mezzo di identificazione e immersione, gli avatar 3D vengono impiegati anche nella formazione e nella simulazione, nella telepresenza per riunioni virtuali, nella comunicazione e persino nella medicina per applicazioni di riabilitazione virtuale. Nel settore dell'animazione, gli avatar 3D consentono di ridurre significativamente il tempo e i costi di produzione, permettendo a registi e designer di creare contenuti di alta qualità con maggiore efficienza. Parallelamente, l'integrazione degli avatar con sistemi di tracking, come tute di motion capture e sensori di movimento, consente di animare i modelli in tempo reale, offrendo una resa più naturale e fedele al movimento umano. sfruttando sistemi di tracking diretto o tute per il controllo dei movimenti. La costruzione di un avatar dettagliato, che rispecchia le caratteristiche di una persona reale o di un personaggio ideato, non si limita infatti alla semplice modellazione e texturizzazione ma anche nella capacità di rendere questi modelli interattivi e animati. Grazie ai sistemi di tracking del corpo (Jiang et al. [2022]), alle tute e gli accessori (Perret and Vander Poorten [2018]) dotati di sensori di movimento, è possibile trasferire in tempo reale i movimenti del corpo umano direttamente all'avatar digitale. Ogni gesto, postura o cambiamento di posizione viene catturato e interpretato dai sensori, che lo trasformano in dati pronti per essere applicati all'avatar. Questo approccio non solo assicura una riproduzione fluida e fedele del movimento umano, ma apre anche possibilità in vari ambiti come l'animazione e la produzione cinematografica, dove i movimenti realistici migliorano la qualità visiva, al mondo dei videogiochi e delle esperienze VR, dove la possibilità di controllare un avatar in tempo reale migliora l'immersione e l'interattività. Inoltre, questa

tecnologia offre potenzialità straordinarie per applicazioni nell'ambito della riabilitazione, del fitness e dello sport, dove monitorare e replicare i movimenti può aiutare ad analizzare, correggere e migliorare la postura o la performance fisica. La creazione di avatar animati tramite sistemi di tracking rappresenta quindi un punto d'incontro tra tecnologia digitale e movimento umano, dando vita a esperienze realistiche che superano la barriera tra virtuale e reale. La ricostruzione di avatar 3D realistici a partire da dati 2D, come fotografie o video, rappresenta una sfida complessa che richiede l'uso di tecniche avanzate in computer vision, machine learning e strutture dati 3D. L'obiettivo principale è ricostruire in modo accurato le caratteristiche geometriche e fisiche di un individuo mantenendo coerenza con l'immagine originale utilizzata come input. Questo processo si avvale di algoritmi di deep learning, in particolare reti neurali convoluzionali, che estraggono informazioni dall'immagine 2D e le traducono in una rappresentazione tridimensionale. Algoritmi di stima della posa, tecniche di ricostruzione volumetrica e modelli parametrici permettono di ottenere rappresentazioni tridimensionali congruenti con la realtà, modellando non solo la struttura del corpo, ma anche dettagli complessi come espressioni facciali, capelli e vestiario. Questa tesi si focalizza sullo sviluppo di una pipeline automatizzata per la creazione di avatar 3D da una singola immagine RGB, con l'obiettivo di integrare diverse tecnologie di computer grafica e machine learning per ottenere una ricostruzione completa e dettagliata del corpo umano. L'approccio proposto sfrutta una combinazione di strumenti e modelli già esistenti, adattandoli e integrandoli in un workflow unico e ottimizzato. Nello specifico, per la ricostruzione della posa e delle espressioni facciali è stato utilizzato SMPLify-X (Pavlakos et al. [2019]), che permette di posizionare accuratamente i giunti corporei, inclusi quelli delle dita e del volto. La generazione del vestiario si basa invece su ClothWild (Moon et al. [2022b]), che identifica il tipo di abbigliamento presente nell'immagine e ne crea una mesh 3D dettagliata. La pipeline include anche HairStep (Zheng et al. [2023b]) per la ricostruzione dei capelli, e uno script per il riconoscimento della barba, garantendo un alto grado di personalizzazione e realismo. L'integrazione di questa pipeline all'interno di un software di modellazione 3D come Blender (Blender Foundation [2024]) rappresenta il passo finale di questo progetto. Blender, grazie alla possibilità di creare add-on personalizzati, consente di offrire agli utenti un'interfaccia intuitiva e accessibile, attraverso la quale è possibile importare e modificare i risultati della pipeline. Nel caso in cui la ricostruzione automatica non soddisfi le aspettative, la pipeline prevede una libreria di modelli predefiniti di abbigliamento e accessori, da cui l'utente può attingere per migliorare o personalizzare ulteriormente l'avatar generato. Questa tesi esplora quindi le opportunità offerte dalla ricostruzione 3D automatizzata e analizza l'impatto della creazione di avatar personalizzati nei contesti sopra menzionati, evidenziando il potenziale e le sfide della fusione tra intelligenza artificiale, computer vision e modellazione tridimensionale. La valutazione della qualità percettiva di mesh animate si basa su metriche oggettive per misurare le distorsioni geometriche e dinamiche in relazione alla percezione umana. La metrica di valutazione integra distorsioni spaziali, come la ruvidità locale, e parametri temporali, come velocità e direzione del movimento per offrire una valutazione complessiva. Le metriche di valutazione oggettive saranno impiegate per analizzare le prestazioni delle mesh animabili. Questi parametri quantitativi permetteranno di determinare l'efficacia e la qualità del risultato ottenuto.

Capitolo 2

Stato dell'arte

In questa sezione saranno presentati gli stati dell'arte per ogni parte della pipeline di realizzazione.

2.1 Ricostruzione della posa

La ricostruzione della posa è un ambito di ricerca molto attivo legato anche all'evoluzione della realtà virtuale o aumentata e del suo utilizzo nell industria del gaming. I lavori presenti nello stato dell'arte sono molteplici e presentano diversi approcci nella ricostruzione della posa di un modello. Prima dell'utilizzo dei Deep Neural Network, molti metodi di ricostruzione della posa 3D utilizzavano modelli geometrici e statistici. Questi metodi si basano su un modello tridimensionale predefinito del corpo umano, come il modello Skinned Multi-Person Linear (SMPL) (Loper et al. [2015]) o il modello SCAPE (Liang et al. [2024]), che rappresentano il corpo umano attraverso una mesh parametrica come mostrato in figura 2.1. Questi modelli sono costruiti utilizzando dataset di scansioni 3D, che vengono successivamente deformati per essere conformi ai dati 2D. L'idea di base di questi metodi è quella di trovare corrispondenze tra i punti chiave (keypoints) rilevati in 2D e le loro controparti nel modello 3D, utilizzando tecniche di ottimizzazione. L'algoritmo cerca di minimizzare l'errore di proiezione tra il modello 3D e i keypoints osservati nell'immagine. La qualità della ricostruzione dipende fortemente dall'accuratezza delle corrispondenze. Inoltre, soffrono di ambiguità, poiché una singola immagine 2D può avere molteplici configurazioni 3D che non corrispondono alla posizione Bidimensionale. Con lo sviluppo del deep learning l'approccio alla ricostruzione della posa 3D ha subito un notevole passo in avanti. I modelli basati su deep learning sono in grado di apprendere in modo diretto la mappatura tra immagini 2D e pose 3D utilizzando grandi dataset etichettati. Un metodo molto utilizzato prevede la stima diretta della posa 3D da immagini 2D tramite l'uso di Convolutional Neural Networks(CNN) per estrarre caratteristiche visive dalle immagini e prevedere la posa 3D direttamente da queste informazioni. Le reti sono addestrate utilizzando dataset etichettati con pose 3D, come Human3.6M (Ionescu et al. [2014]), che contengono immagini 2D mappate direttamente con i dati di posa 3D. Un approccio comune è la regressione diretta delle coordinate 3D a partire dall'immagine.

Figura 2.1: Mesh parametrica di SMPL con le posizioni dei giunti. Fonte: https://images.app.goo.gl/BtN2GGKNsApeikRf8

Algoritmi come Vnec (Mehta et al. [2017]) combinano l'estrazione di caratteristiche da CNN e tecniche di regressione per prevedere direttamente la posizione 3D delle articolazioni. Un altro approccio diffuso prevede la stima della posa 2D dalle immagini RGB e successivamente la ricostruzione della posa 3D partendo da questi keypoints 2D. Algoritmi come OpenPose (Cao et al. [2019]) utilizzano reti neurali per rilevare keypoints in 2D e successivamente proiettare questi punti nello spazio 3D cercando di ottimizzare il risultato. Spesso vengono utilizzate due reti in sqeuenza, dove la prima rete prevede i keypoints 2D e una seconda rete traduce la posa in 3D direttamente dai keypoints bidimensionali. Un esempio di questo approccio è il modello LCR-Net (Rogez et al. [2019]), che utilizza una CNN per stimare la posa 2D e successivamente una Recurrent neural network (RNN) per stimare la posa 3D. Negli ultimi anni, alcuni metodi combinano dati diversi come immagini RGB e dati di profondità (RGB-D), per migliorare la precisione della ricostruzione della posa 3D. L'utilizzo di telecamere di profondità come Kinect permette di avere informazioni di profondità che semplificano la stima della posizione delle articolazioni. La ricostruzione della posa 3D a partire da immagini singole presenta diverse difficoltà dovute alla mancanza di informazioni sulle parti del corpo non visibili da una singola prospettiva. Per superare questo limite, alcuni approcci utilizzano immagini multivista, che catturano il soggetto da diverse angolazioni. Questo metodo consente di ottenere una rappresentazione geometrica più completa, sfruttando la complementarità delle diverse viste per ricostruire con maggiore precisione la struttura tridimensionale del corpo umano. A causa della mancanza di informazioni invece di utilizzare immagini multivista alcuni approcci sperimentano l'uso di sequenze video ciò consente di migliorare la ricostruzione

della posa 3D sfruttando la coerenza temporale. Modelli come Temporal Convolutional Networks (TCN) e Long Short-Term Memory (LSTM) sfruttano le informazioni temporali per ottenere una predizione precisa e stabile della posa nel tempo. Alcuni metodi per la ricostruzione 3D sfruttano le informazioni provenienti da sequenze video, utilizzando i dati temporali e spaziali raccolti da diverse angolazioni per ottenere modelli tridimensionali dettagliati e realistici. Questi approcci offrono significativi vantaggi rispetto alle tecniche basate su singole immagini, poiché i video forniscono molteplici prospettive e dettagli dinamici utili per una ricostruzione più accurata (Alldieck et al. [2018]). Altri metodi di ricostruzione 3D utilizzano l'optical flow per estrarre informazioni sul movimento di pixel tra fotogrammi consecutivi in un video. Questa tecnica consente di comprendere come le superfici e gli oggetti si spostano nello spazio e nel tempo, fornendo dettagli utili per la ricostruzione tridimensionale. Grazie all'analisi delle variazioni di intensità luminosa tra i frame, l'optical flow può rilevare direzioni e velocità di movimento, anche in situazioni complesse con cambi di illuminazione o texture. L'integrazione dell'optical flow nei processi di ricostruzione 3D migliora l'accuratezza dei modelli, poiché permette di acquisire non solo la geometria degli oggetti, ma anche la loro dinamica (Achenbach et al. [2017]).

2.2 Ricostruzione dei vestiti

La ricostruzione dei vestiti è di notevole importanza data l'evoluzione della realtà virtuale o aumentata e soprattutto nell ambito dell'animazione. Lo stato dell'arte nella ricostruzione 3D dei vestiti si concentra su tecniche avanzate per modellare l'abbigliamento digitale con un alto livello di fedeltà visiva e precisione geometrica. Uno degli approcci più comuni è basato sull'uso di modelli parametrici del corpo umano, come il modello SMPL (Loper et al. [2015]). Questi modelli vengono estesi per includere la geometria dell'abbigliamento, rappresentato come una mesh deformabile. I vestiti possono essere trattati come un'estensione della superficie del corpo. Modelli come SMPLIcit (Corona et al. [2021]) combinano la ricostruzione del corpo umano con la geometria dei vestiti, utilizzando un sistema di deformazione che segue i movimenti del corpo su cui sono poggiati. La geometria del vestito viene adattata alle pose del corpo. Un'ambito di ricerca recente si concentra su modelli che stimano la geometria dell'abbigliamento a partire dalla posa del corpo. Questi approcci cercano di apprendere una mappatura tra la forma del corpo e la geometria dei vestiti. Il modello ClothCap (Pons-Moll et al. [2017]) ricostruisce vestiti realistici catturando dettagli come pieghe e ondulazioni, a partire dai movimenti creati dalla posa del corpo. Altre ricerche utilizzano reti neurali per prevedere la mesh del vestito in base alla posa del corpo, creando una corrispondenza tra la struttura sottostante e la superficie dell'abbigliamento. Un altro approccio importante per la ricostruzione dei vestiti si basa su tecniche di simulazione fisica, che mirano a replicare accuratamente il comportamento e le proprietà fisiche dei tessuti che porta ad un risultato molto realistico ma molto gravoso dal punto di vista computazionale. Un ulteriore strategia utilizza Modelli basati su particelle e simulazioni FEM (Finite Element Method) (Tekkaya and Soyarslan [2014]). Le simulazioni basate su particelle e FEM sono utilizzate per replicare il movimento realistico dei vestiti sotto l'effetto di forze come la gravità o il vento. Questi approcci generano una mesh 3D del vestito che si deforma dinamicamente in base alle interazioni tra il tessuto e

l'ambiente in cui è inserito. I metodi che utilizzano i FEM sono utili per simulare le proprietà meccaniche più complesse dei tessuti, come l'elasticità, la resistenza alla trazione e le pieghe. Questo metodo offre un'accuratezza superiore, ma ad un costo computazionale decisamente elevato. Per migliorare la stabilità delle simulazioni e catturare dettagli 3D dei vestiti, si sono sviluppati approcci che rappresentano i vestiti come voxel. Questo permette di simulare collisioni e pieghe con maggiore precisione, evitando compenetrazioni tra il corpo e l'abbigliamento. Il deep learning anche in questo ambito costituisce un area di notevole interesse, le reti generative come le Generative Adversarial Networks (GAN) o le Variational Autoencoders (VAE) sono utilizzate per generare vestiti realistici in 3D a partire da immagini 2D. Questi modelli apprendono a stimare la geometria dell'abbigliamento direttamente dalle immagini RGB, sfruttando dataset di immagini con annotazioni 3D per il training. TailorNet (Patel et al. [2020]) utilizza reti neurali per generare vestiti realistici, apprendendo dalle pose del corpo e dalle proprietà del tessuto come mostrato in figura 2.2. Questo approccio permette di creare vestiti digitali personalizzati. Utilizzando

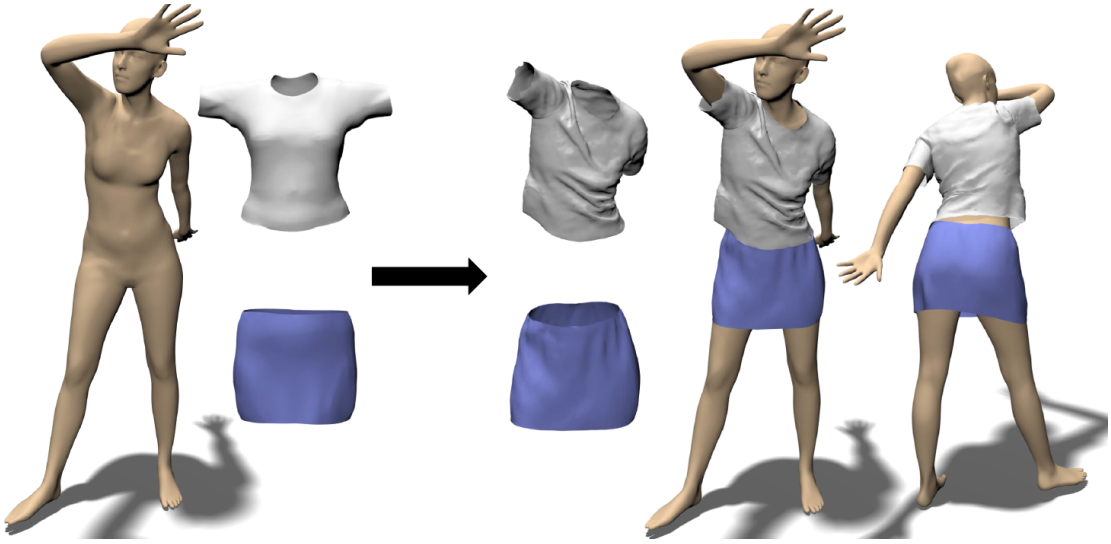


Figura 2.2: Ricostruzione effettuata con TailorNet. Fonte: https://images.app.goo.gl/kq2MswgwvrcoCmU18

immagini catturate da diverse angolazioni, alcuni metodi di deep learning ricostruiscono sia la geometria sia la texture dell'abbigliamento. Le reti neurali convoluzionali sono utilizzate per mappare texture dettagliate sui modelli 3D, migliorando la fedeltà visiva dei vestiti digitali. Le tecniche per la generazione di vestiti virtuali realistici puntano a replicare pieghe e deformazioni del tessuto in modo accurato, utilizzando modelli di deep learning e dati tridimensionali. Questi metodi combinano reti neurali e rappresentazioni di superficie per catturare dettagli naturali del tessuto, come increspature e curvature, in risposta al movimento del corpo. Il processo tipicamente implica l'addestramento di un modello su dati che rappresentano abiti in varie pose, così da apprendere come le pieghe si formano e si modificano. Successivamente, il modello applica queste caratteristiche a una struttura di base, creando pieghe e increspature in modo adattivo, a seconda della

postura e della posizione del corpo (Laehner et al. [2018]).

2.3 Ricostruzione dei capelli

La ricostruzione 3D dei capelli rappresenta una delle sfide più complesse nel campo della computer grafica. La struttura fine, la complessità geometrica e il comportamento dei capelli rendono questo problema particolarmente difficile da affrontare. Negli ultimi anni sono stati sviluppati numerosi metodi che sfruttano tecniche avanzate di modellazione, oltre a algoritmi di deep learning, per ricostruire in modo accurato i capelli di un soggetto a partire da immagini o video. Uno dei primi approcci per la ricostruzione dei capelli si basa su tecniche di modellazione geometrica. Questi metodi mirano a rappresentare i capelli come una serie di curve o spline, che approssimano la direzione e il movimento delle ciocche. Il procedimento parte da un modello di base, spesso uno scalpo approssimato, da cui le curve vengono generate e modificate per seguire la direzione dei capelli presenti nell'immagine in input. Un ulteriore approccio consiste nel creare curve tridimensionali che seguono la direzione dei capelli presenti nelle immagini di riferimento. Le curve sono generate da un insieme di punti guida, definiti in partenza o estratti automaticamente dalle immagini. Questo metodo, pur essendo relativamente semplice, soffre di problemi di accuratezza, soprattutto quando si tratta di catturare la complessità dei movimenti dei capelli riproducendo un comportamento realsitico. Un altro metodo classico è l'uso di modelli basati su volumi per rappresentare la densità e la direzione dei capelli. Questi modelli sono spesso impiegati per simulare il comportamento collettivo delle ciocche, riducendo la complessità della modellazione di ogni singolo capello. Tuttavia, la risoluzione dei volumi può limitare la precisione nei dettagli. Con l'aumento delle capacità di calcolo e l'avanzamento delle tecniche di computer vision, i metodi che sfruttano immagini e video per la ricostruzione dei capelli sono maggiormente utilizzati. Questi approcci cercano di sfruttare le informazioni presenti nelle immagini 2D per ricostruire una rappresentazione 3D dei capelli che con l'aggiunta di altre informazioni permette di avere ottime prestazioni. Uno dei metodi più efficaci è la ricostruzione multiview, in cui il soggetto viene catturato da più viste. Utilizzando la corrispondenza dei punti tra le diverse immagini, è possibile ottenere una stima della posizione 3D dei capelli. Questo approccio, combinato con tecniche di ricostruzione basate su stereo vision, permette di creare modelli piuttosto accurati dei capelli. Il costo computazionale e la necessità di multiple viste limitano l'applicabilità di questo metodo in contesti con una singola immagine che comunque portano a risultati più che soddisfacenti. Un altro approccio utile è Structure from Motion (SfM), che utilizza video o sequenze di immagini per stimare la struttura tridimensionale dei capelli, sfruttando il movimento del soggetto o della fotocamera. Anche se efficace in molti casi, il metodo può soffrire in presenza di movimenti troppo veloci o complessi dei capelli. Con la diffusione del deep learning, sono stati sviluppati nuovi approcci basati su CNN e GAN che hanno migliorato in modo significativo la capacità di ricostruire capelli realistici da immagini singole o video. Questi metodi sono capaci di apprendere caratteristiche sottili dei capelli, come la forma, la densità e la texture, direttamente dai dati in input. Un modello basato su reti neurali progettato per generare una rappresentazione volumetrica dei capelli a partire da una singola immagine è HairNet (Rolland et al. [2022]). HairNet

utilizza un'architettura CNN per apprendere la distribuzione spaziale dei capelli, combinando informazioni sulla forma del viso e della testa per produrre un modello 3D. Sebbene i risultati siano promettenti, questo metodo ha ancora difficoltà a gestire situazioni in cui i capelli sono molto complessi o presentano geometrie intrecciate e particolarente confuse. Sebbene non sia specificamente progettato per i capelli, Pixel2Mesh (Wang et al. [2018]) genera una mesh 3D a partire da un'immagine RGB utilizzando una rete neurale. Alcuni ricercatori hanno adattato questo tipo di architettura per la ricostruzione dei capelli, ottenendo risultati promettenti nella modellazione di forme complesse. Alcuni lavori recenti utilizzano GAN per generare ricostruzioni 3D di capelli molto realistiche. In particolare, le GAN vengono utilizzate per sintetizzare texture di capelli e dettagli fini che sarebbero difficili da modellare come in figura 2.3. Questo approccio risulta particolarmente utile per creare mesh e texture dettagliate che migliorano l'aspetto dei capelli ricostruiti. Una

Figura 2.3: Ricostruzione capelli che utilizza una GAN. Fonte: https://images.app.goo.gl/WMvFoer7RTrTr9S9A

volta che la geometria dei capelli è stata ricostruita, un'altra sfida cruciale è la simulazione del loro comportamento fisico e il rendering realistico. I capelli interagiscono tra loro, con il corpo e con l'ambiente circostante, ma in questo progetto ci si è concentrati sulla ricostruzione in modo che sia il più realistico possibile ignorandone il comportamento fisico.

2.4 Ricostruzione Barba

La ricostruzione della barba, come la ricostruzione di capelli, è un'area di ricerca che si concentra sull'ottenimento di modelli realistici di peli facciali quali barba e baffi. In recenti studi, sono stati sviluppati metodi avanzati basati su reti generative avversarie condizionali (CGAN), modelli di StyleGAN e architetture encoder-decoder per la segmentazione e trasferimento di attributi facciali. Nel lavoto Strand-accurate multi-view facial hair reconstruction (Li and Liu [2024]), la ricostruzione della barba è stata studiata con una pipeline che sfrutta tecniche di machine learning per separare attributi facciali e della barba oltre

alla divisione dai capelli, migliorando l'accuratezza della generazione di modelli 3D. Questo approccio usa un'architettura basata su GAN per la creazione di dettagli fini, come la crescita e la distribuzione della barba, aumentando il livello di realismo del risultato. In un altro studio, Single View Facial Hair 3D Reconstruction (Rotger et al. [2019]), vengono descritti algoritmi che sfruttano reti neurali convoluzionali per segmentare e modificare le caratteristiche della barba. Questa ricerca si basa su un dataset di immagini ad alta risoluzione e sfrutta una combinazione di modelli pre-addestrati per migliorare la coerenza tra la barba e il viso generato mantenendo la separazione con i capelli come mostrato in figura 2.4. In entrambi i casi, le tecniche si concentrano sul miglioramento dell'adatta-

Figura 2.4: Ricostruzione Barba da singola immagine. Fonte: https://images.app.goo.gl/UMV6DQ9qgovLGfSA7

mento tra la barba e le altre caratteristiche facciali, integrando il colore, la texture e la geometria per ottenere una ricostruzione più accurata e realistica. In questo lavoro ci si è concentrati sulla classificazione del soggetto riconoscendo la presenza di barba e baffi in modo da fornire la possibilità all'utente finale di personalizzare il proprio avatar.

2.5 Riconoscimento carnagione

La creazione di texture della pelle da immagini RGB ha visto importanti sviluppi attraverso l'uso di tecniche di computer vision oppure tramite lo sfruttamento di metodi di machine learning. Uno degli approcci più recenti è rappresentato nell'articolo "Skin Segmentation Using Deep Learning Approaches" (Li et al. [2023]). , che esplora l'uso di DNN per la segmentazione e la creazione di texture realistiche. Questo lavoro si focalizza sull'uso di CNN per isolare la pelle in immagini RGB, migliorando l'accuratezza della segmentazione rispetto ai metodi tradizionali. Le tecniche basate su reti GAN sono state ampiamente utilizzate per la generazione di texture della pelle. Questi modelli non solo apprendono le caratteristiche visive della pelle ma generano anche texture che variano realisticamente in risposta a diverse condizioni di illuminazione e angolazione. In generale, gli attuali approcci utilizzano tecniche di deep learning per garantire un risultato realistico nella simulazione della pelle umana. In questo lavoro si è preferito utilizzare tecniche di computer vison in modo da utilizzare il risultato per inserire direttamente il colore sul modello.

Capitolo 3

Design del sistema e tecnologie

In questo capitolo verranno illustrati tutti i passaggi necessari alla progettazione e realizzazione dei componenti che hanno permesso la creazione di una pipeline per generare un avatar 3D animabile.

3.1 Pipeline per la ricostruzione dell'avatar

Il primo passo è stato quello di comprendere i vari elementi necessari al raggiungimento di tale obiettivo, con l'intento di analizzare lo stato dell'arte e individuare le soluzioni più avanzate e consolidate. L'analisi dello stato dell'arte ha permesso di ottenere una visione complessiva delle metodologie utilizzate per affrontare i problemi legati alla ricostruzione 3D a partire da un'immagine RGB. Questo processo è stato di fondamentale importanza per individuare progetti che condividessero una base comune, facilitando l'integrazione dei diversi componenti. In particolare, è stato deciso di focalizzarsi su progetti che utilizzano come base il modello umanoide SMPL (Loper et al. [2015]). L'impiego di progetti basati sullo stesso modello garantisce la coerenza nella gestione delle informazioni relative alla nomenclatura dei giunti e l'uso della stessa mesh, che successivamente viene animata e posizionata in pose diverse. Questo aspetto si è rivelato cruciale, soprattutto per la creazione e il posizionamento dei vestiti sul modello, operazioni che sono risultate semplificate proprio grazie all'utilizzo del modello SMPL. Per quanto riguarda la ricostruzione dei capelli, una volta generata la mesh del corpo, il loro posizionamento è risultato molto più semplice. Pertanto, è stato scelto di adottare un approccio che garantisse un alto livello di accuratezza e realismo. L'obiettivo principale del progetto è quello di creare un'interfaccia grafica in un software di modellazione e animazione, che consenta di eseguire tutte le fasi della ricostruzione 3D e di poter utilizzare il risultato finale per scopi di animazione o rendering. Per implementare la pipeline, si è optato per la creazione di un Add-on in Blender, poiché questo software offre la possibilità di sviluppare componenti aggiuntivi open-source e di eseguire i processi di ricostruzione direttamente all'interno dell'interfaccia (figura 3.12). Per la ricostruzione del corpo, è stato scelto SMPLify-X, non solo per la sua efficienza computazionale, ma anche per la possibilità di sfruttare parte delle risorse già presenti nell'Add-on sviluppato dagli ideatori del progetto, che consente di ottenere

un modello di SMPL all'interno di Blender. La ricostruzione degli abiti è stata affidata a ClothWild (Moon et al. [2022b]), un approccio che utilizza il modello SMPL come base su cui vengono ricostruiti gli indumenti. Il sistema sceglie il vestiario più appropriato classificando l'immagine in input e applicando l'indumento in modo coerente con il corpo ricostruito. Per la ricostruzione dei capelli, si è optato per HairStep (Zheng et al. [2023b]), un metodo che ricostruisce in maniera estremamente realistica i capelli a partire da una singola immagine RGB, offrendo un risultato visivamente accurato. Infine, la ricostruzione della barba e il riconoscimento del colore della pelle sono stati realizzati attraverso tecniche di computer vision e l'utilizzo di classificatori. Questi strumenti permettono all'utente finale di personalizzare l'avatar, scegliendo vari aspetti estetici in modo interattivo. Tutti gli approcci selezionati condividono la caratteristica comune di richiedere come input una singola immagine e di produrre risultati accurati nonostante le poche informazioni fornite. Il risultato finale della pipeline sarà ottenibile attraverso l'Add-on sviluppato per Blender. Questo permetterà di eseguire tutti i modelli di ricostruzione citati, con la possibilità di personalizzare ulteriormente gli indumenti selezionando oggetti da una libreria predefinita. Una volta completato il processo, l'utente avrà a disposizione un avatar 3D pronto per essere utilizzato in scenari di rendering o animazione, offrendo un elevato grado di flessibilità e realismo.

3.2 SMPLify-X

Il progetto Expressive Body Capture: 3D Hands, Face, and Body from a Single Image introduce SMPL-X (Pavlakos et al. [2019]), un modello 3D avanzato che combina corpo, mani e viso in un'unica rappresentazione. Questo modello è stato sviluppato per affrontare le limitazioni dei modelli precedenti, che non erano in grado di catturare adeguatamente le espressioni facciali e i movimenti dettagliati delle mani. L'integrazione di queste componenti è cruciale per comprendere in modo completo le interazioni umane e le dinamiche emotive che emergono dalle immagini. Questa implementazione è un notevole miglioramento rispetto ai lavori precedenti di SMPL (Loper et al. [2015]) integrando MANO (Romero et al. [2017]) e FLAME (Li et al. [2017]). MANO è un modello 3D parametrico sviluppato per la rappresentazione mano umana in diverse pose e forme. MANO è progettato specificamente per catturare le variazioni nella forma della mano e nelle deformazioni articolari, utilizzando dati di scansione 3D di mani in numerose pose mantenendo così un risultato realistico. Il modello MANO è composto da 30 giunti che controllano la posa delle dita, utilizzando uno spazio di posa basato sulla Principal Component Analysis (PCA) per ridurre la complessità delle articolazioni e rappresentare le deformazioni in modo più compatto. A differenza dei modelli precedenti, che spesso consideravano la mano separata dal corpo, MANO è stato progettato per essere integrato con modelli completi del corpo umano fornendo così un ottima integrazione con SMPL consentendo una rappresentazione più realistica delle mani in movimento o in una particolare posa. FLAME è un modello 3D sviluppato per rappresentare il viso in modo da rendere realistiche le espressioni di un volto. A differenza dei modelli facciali tradizionali, FLAME non si limita solo alla parte anteriore del viso, ma include anche la testa e il collo, migliorando la continuità tra viso e corpo. FLAME è basato su scansioni 3D di volti e

cattura sia la forma del volto che le variazioni dovute alle espressioni, utilizzando uno spazio di deformazione parametrico che consente la manipolazione delle espressioni facciali. Il modello è composto da parametri che controllano la rotazione della testa e del collo, la forma individuale del volto e le espressioni basate su blendshape, ispirate al sistema FACS (Facial Action Coding System). FLAME come MANO è pensato per essere integrato con modelli corporei come SMPL, facilitando l'uso in applicazioni grafiche e di animazione, dove la veridicità di una ricostruzione è fondamentale. Un aspetto fondamentale del lavoro è SMPLify-X, un metodo di ottimizzazione che si basa sul concetto di SMPLify, ma estende le sue funzionalità per adattare il modello SMPL-X alle caratteristiche 2D rilevate nelle immagini. Si è scelto di utilizzare SMPLify-X per la sua capacità di mantenere le funzionalità offerte da SMPL-X ma utilizzando come input una singola immagine RGB. SMPLify-X affronta una delle principali sfide nella ricostruzione 3D da singole immagini, ossia senza dati accoppiati tra immagini e ground truth in 3D. Per ovviare a questo problema, il metodo utilizza OpenPose, una libreria che rileva automaticamente i keypoints del corpo, delle mani e del viso nelle immagini 2D. Successivamente, il modello SMPL-X viene adattato a queste caratteristiche tramite un processo di ottimizzazione. SMPLify-X migliora in modo significativo l'approccio originale SMPLify in diversi aspetti, rileva e utilizza le caratteristiche 2D non solo del corpo, ma anche delle mani, dei piedi e del viso, introduce un nuovo pose prior basato su dati provenienti da motion capture, migliorando così la qualità delle pose stimate come visibile in tabella 3.1 . Un ulteriore miglioramento

Modello	**Keypoints**	**errore v2v**	**Joint error**
SMPL	Body	57.6	63.5
SMPL	Body+Hands+Face	64.5	71.7
SMPL+H	Body+Hands	54.2	63.9
SMPL-X	Body+Hands+Face	52.9	62.6

Tabella 3.1: Paragone dei metodi allo stato dell'arte basato sukeypoints, v2v error, and joint error.

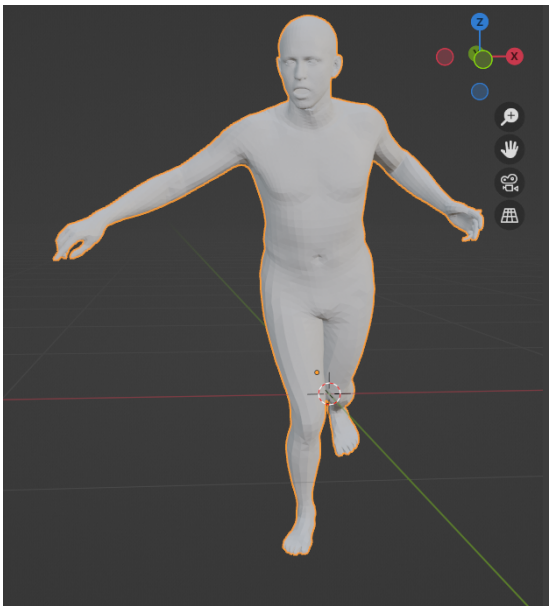
si concretizza in una nuova penalizzazione delle interpenetrazioni nella mesh che è sia veloce sia accurata, consentendo una gestione più efficiente delle collisioni tra le varie parti del corpo. SMPLify-X rileva automaticamente il genere della persona, adattando così il modello corporeo, nel caso la rilevazione del genere sia incerta verrà utilizzato un modello neutro. La pipeline di SMPLify-X per la ricostruzione di una posa 3D a partire da un'immagine RGB segue una serie di passaggi che coinvolgono l'estrazione di caratteristiche 2D, il fitting di un modello 3D di SMPL-X e l'ottimizzazione dei parametri del modello. L'Estrazione delle caratteristiche 2D dall'immagine in input inizia utilizzando OpenPose, che rileva le posizioni dei giunti del corpo, delle mani, del viso e dei piedi. Questi punti chiave 2D servono come input per la successiva fase di fitting del modello. Successivamente viene inizializzato il modello parametrico di SMPL-X con una serie di parametri che controllano la posa, la forma e le espressioni facciali. SMPL-X utilizza una combinazione di modelli espressivi, inclusi FLAME per il viso e MANO per le mani. Fitting del modello alle caratteristiche 2D: Una volta estratte le caratteristiche 2D dal sogetto nell'immagine in input, viene eseguito il fitting del modello SMPL-X tramite un approccio

basato sull'ottimizzazione. L'obiettivo è minimizzare la distanza tra le proiezioni 2D dei giunti del modello 3D e i keypoints rilevati da OpenPose nell'immagine 2D. Questo fitting viene formulato come un problema di ottimizzazione che non migliora solo la distanza tra le proiezioni con i giunti. un ulteriore miglioramento è offerto dall'utilizzo di VPoser (Müftüoğlu et al. [2024]),autoencoder variazionale, addestrato su dati di motion capture per penalizzare le pose non realistiche ed avere risultati migliori. Vposer Riduce le interpenetrazioni fisicamente impossibili tra le diverse parti del corpo, ad esempio mani che attraversano il corpo per mantenere una posa reale. Regolarizzazione della forma e delle espressioni: Penalizza deviazioni dalla forma media del corpo e dalle espressioni facciali standard. Ottimizzazione iterativa: L'ottimizzazione viene eseguita in modo iterativo utilizzando l'algoritmo L-BFGS (Liu and Nocedal [1989]), affinando i parametri del modello in modo che le proiezioni 3D si allineino sempre meglio ai keypoints 2D. Viene utilizzato un approccio in cui inizialmente si ottimizzano la traslazione e l'orientamento globale del corpo, seguiti dalla posa e dalla forma del corpo, delle mani e del viso. Durante il processo, viene progressivamente ridotta la regolarizzazione per migliorare il risultato finale. Al termine dell'ottimizzazione, il risultato è una posa 3D dettagliata e coerente con i dati bidimensionali rilevati nell'immagine. SMPLify-x produce come output un file .obj, e un file .pkl. il formato .obj è utilizzato per rappresentare modelli 3D. È un formato di testo semplice che contiene informazioni relative a vertici, facce, normali e coordinate texture di un modello tridimensionale. il formato .pkl è un file binario usato in Python per serializzare e salvare oggetti Python. La libreria pickle consente di inserire in una struttura dati un oggetto Python in un file e successivamente ricaricarlo per riutilizzarlo in un'altra sessione. Il file .pkl in output contiene la posizione e le informazioni dimensionali rilevate nell'immagine in input ciò permette l'utilizzo di questi dati per posizionare corettamente l'armatura legata ad una mesh. Nel lavoro di SMPLify-X vengono utilizzate diverse reti neurali e modelli di machine learning per risolvere il problema della ricostruzione 3D di corpo, mani e volto da immagini RGB. Viene utilizzata una rete di regressione basata su PyTorchPaszke et al. [2019b] per accelerare l'ottimizzazione e migliorare l'efficienza computazionale, con un notevole incremento di velocità rispetto alle implementazioni precedenti, come quella in ChumpyLoper [2014]. Per migliorare ulteriormente il fitting, viene definita una nuova penalizzazione delle interpenetrazioni tramite un modello basato su BVH (Bounding Volume Hierarchies), che rileva collisioni e penetra sulle superfici per garantire che il modello non presenti artefatti fisicamente irrealistici. Oltre a queste reti, viene addestrato anche un classificatore di genere tramite una ResNet18He et al. [2015], che rileva automaticamente il sesso del soggetto in base ai punti chiave rilevati da OpenPose e seleziona il modello corporeo appropriato, maschile, femminile o neutro. Questo approccio aiuta a migliorare l'accuratezza del fitting, adattando il modello di base alla conformazione fisica del soggetto rappresentato nell'immagine. l'approccio utilizzato nell'Add-on è stato quello di non utilizzare il classidicatore per rilevare automaticamente il genere ma di permettere direttamente all'utente di poter scegliere il genere che preferisce in modo da dare completa liberà di personalizzazione ed evitare che per ragioni etiche si provi fastidio nell'avere assegnato un genere che non rappresenta l'utilizzatore del software.

Limiti

SMPLify-x migliora molto la ricostruzione della posa rispetto ai modelli precedenti ma soffre di alcune limitazioni dovute al fatto che l'input per la ricostruzione è una singola immagine RGB. I problemi principali sono legati alle occlusioni che nonostante il modulo per la penalizzazione delle collisioni va a creare artefatti o effetti poco realistici sul risultato. Le occlusioni possono essere causate sia da parti del corpo che oggetti che nascondono un articolazione come in figura 3.1. La casistica presenta un oggetto che occlude

(a) Immagine in input. Fonte : https://www.quotidianodelsud.it/wp-content/uploads/2024/03/20240302_0149-980x551.jpg



(b) Ricostruzione.

Figura 3.1: Ricostruzione di una posa con occlusioni.

completamente la mano del modello, in questo caso la ricostruzione produce dei problemi legati ad un interpenetrazione delle dita del modello fornendo un risulatato poco realistico mostrato nella figura 3.2. Un ulteriore problema risiede nel fatto che l'armatura dei modelli proposti è personalizzata, come illustrato in figura 3.3, rendendo l'utilizzo finale dell'avatar poco pratico. L'utilizzo di un rig standard per un modello su Blender, mostrato in figura 3.4 rappresenta un vantaggio significativo per l'efficienza e la coerenza del flusso di lavoro. Un rig standardizzato è già dotato di configurazioni predefinite e di un set completo di controlli per le articolazioni, che riducono il tempo necessario per creare e configurare manualmente l'armatura e i pesi della mesh. Questa uniformità permette di applicare rapidamente animazioni, soprattutto nei casi in cui si lavora con librerie di movimenti o su progetti che richiedono molte iterazioni. Inoltre, un rig standard facilita il trasferimento di animazioni tra diversi modelli, mantenendo compatibili i movimenti e consentendo modifiche senza compromettere la qualità dell'animazione. L'uso di rig standardizzati contribuisce anche alla qualità visiva, poiché minimizza le distorsioni indesiderate delle articolazioni grazie alla distribuzione bilanciata dei pesi. In sintesi, adottare

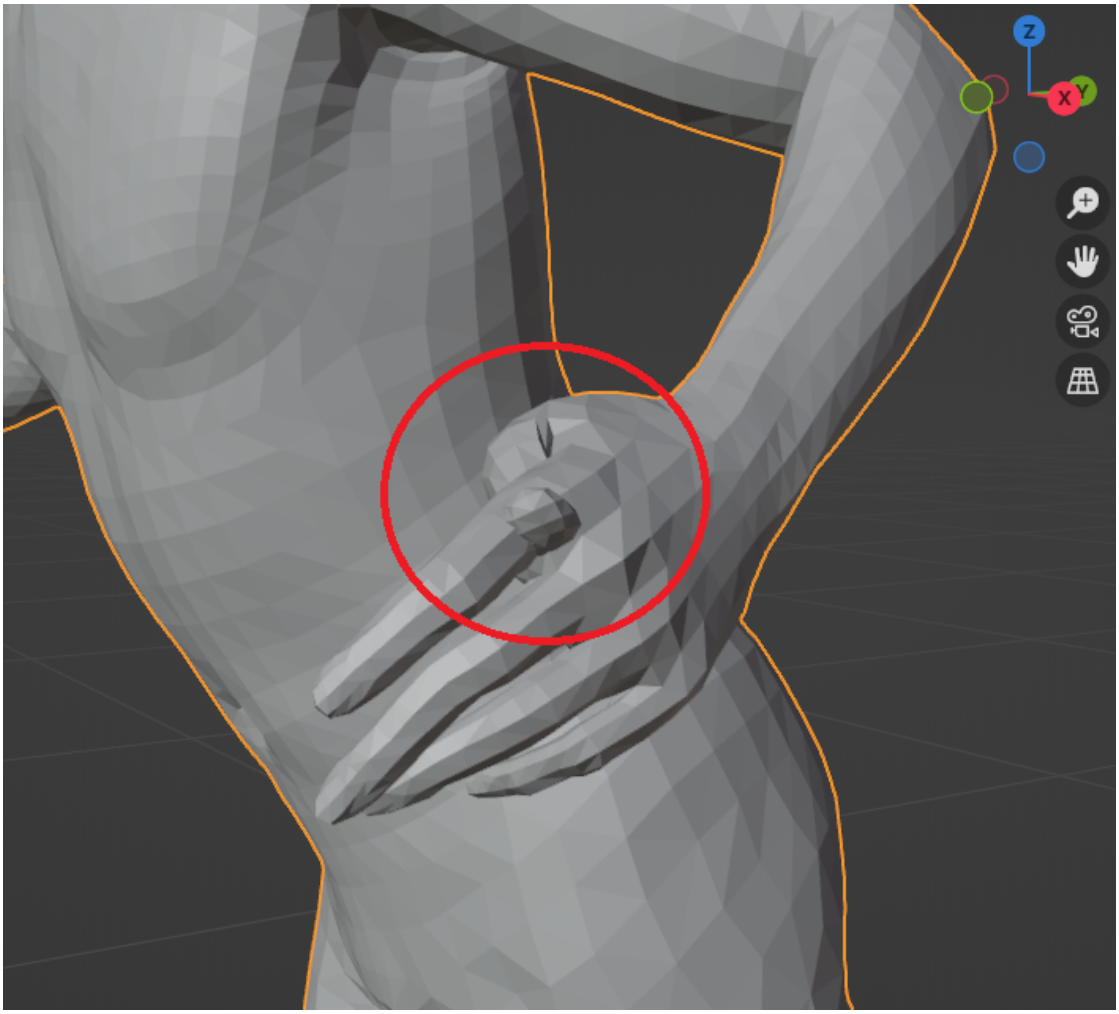


Figura 3.2: Errore causato dall'occlusione di un oggetto.

un rig standard su Blender ottimizza l'intero processo di animazione, riducendo il tempo e garantendo la consistenza sia a livello tecnico sia visivo. L'armatura di SMPL-X risulta fondamentale per il caricamento della posa riconosciuta a partire dall'immagine. Si è quindi deciso di mantenere l'armatura proposta dal modello SMPL-X, offrendo al contempo la possibilità di inserire un modello personalizzato direttamente tramite l'interfaccia grafica dell'addon, come illustrato in figura. 3.5

3.3 ClothWild

La ricostruzione 3D di persone in contesti reali è estremamente complessa per diverse ragioni. Prima di tutto, le persone possono trovarsi in pose complesse e dinamiche, in ambienti con illuminazioni non uniformi e soggette a occlusioni parziali o complete del

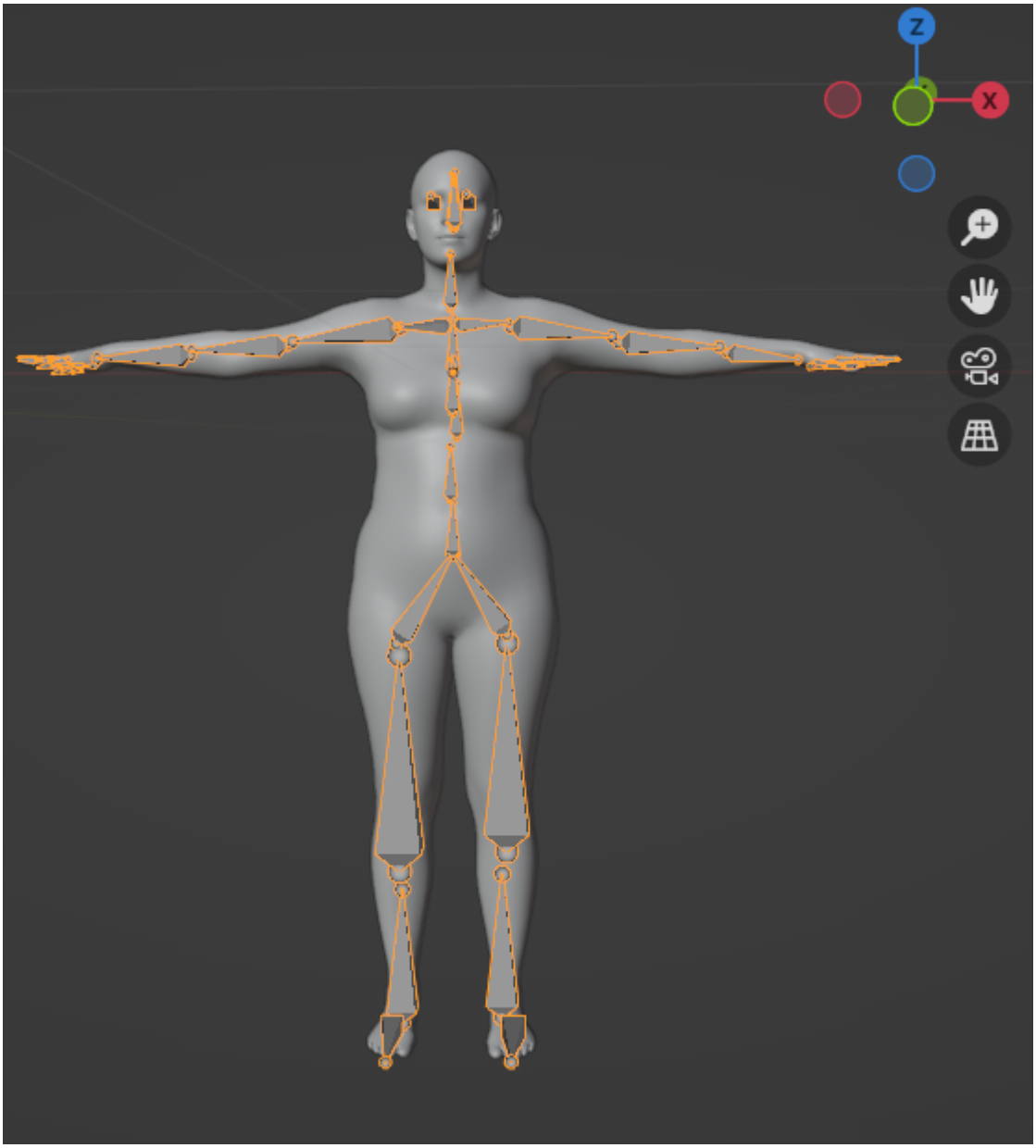


Figura 3.3: Armatura di SMPL-X.

corpo. Inoltre, il tipo e la varietà degli indumenti indossati introducono una notevole variabilità, poiché i vestiti non solo si deformano in modi differenti, ma possono anche nascondere le forme sottostanti del corpo umano. Questo rende difficoltosa l'applicazione diretta di metodi standard basati su dataset sintetici o controllati, in cui le pose e le condizioni di cattura sono più uniformi e prevedibili. L'approccio proposto da ClothWild si distingue proprio per la sua capacità di operare in condizioni non strutturate, utilizzando

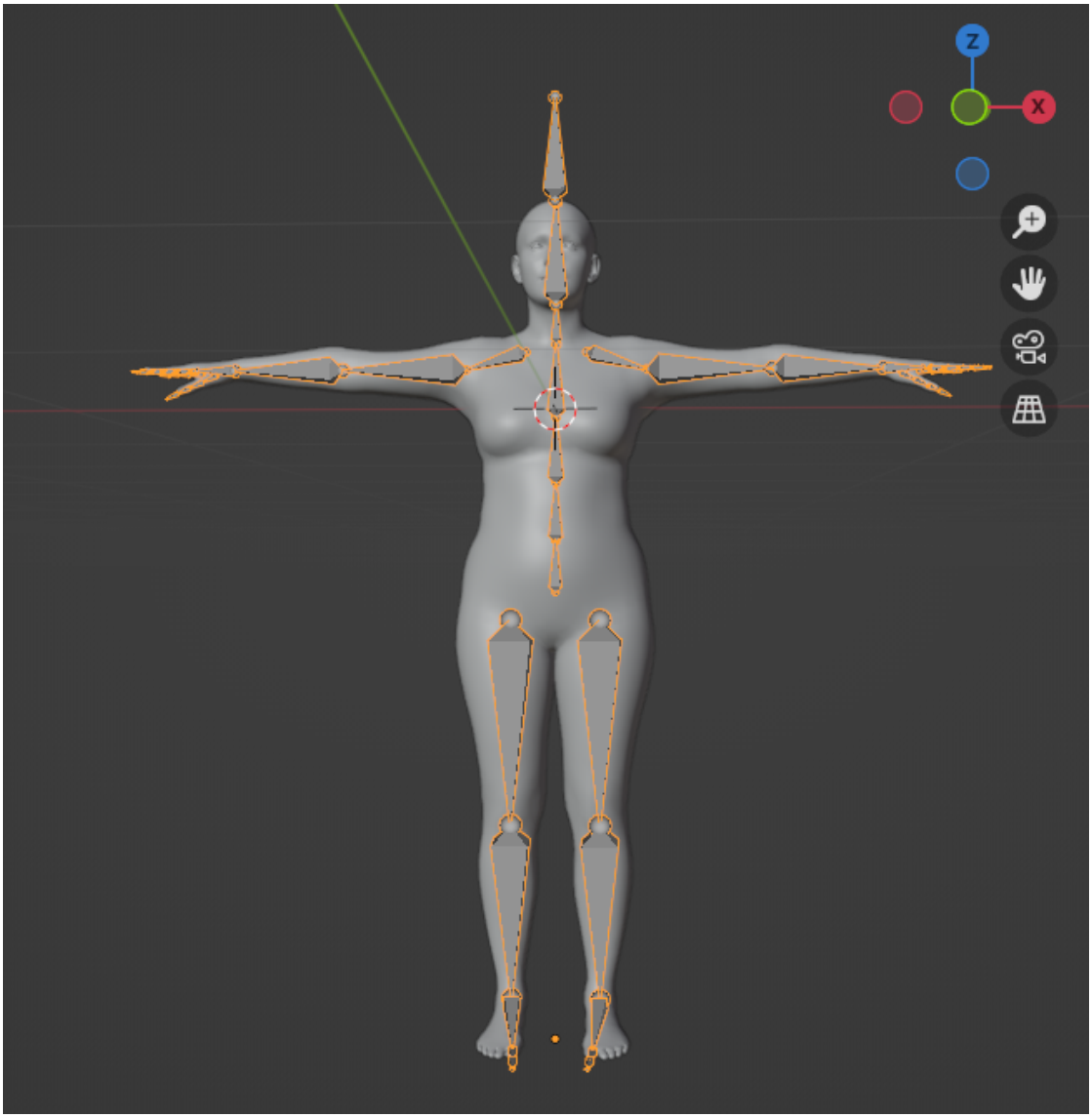


Figura 3.4: Armatura standard.

una pipeline debolmente supervisionata che non richiede dati 3D espliciti per l'addestramento. A differenza di molte soluzioni tradizionali che dipendono da complessi dataset 3D, come scansioni di persone o modelli catturati tramite tecniche di acquisizione avanzate con l'utilizzo di appositi scanner. Il framework di questo lavoro sfrutta supervisione bidimensionale, Ciò significa che l'addestramento del modello è basato su informazioni bidimensionali derivate da immagini RGB e non su dati volumetrici completi. Questo approccio permette di utilizzare un'ampia gamma di immagini "in-the-wild", spesso disponibili in grandi quantità, aumentando la robustezza del modello e riducendo i costi

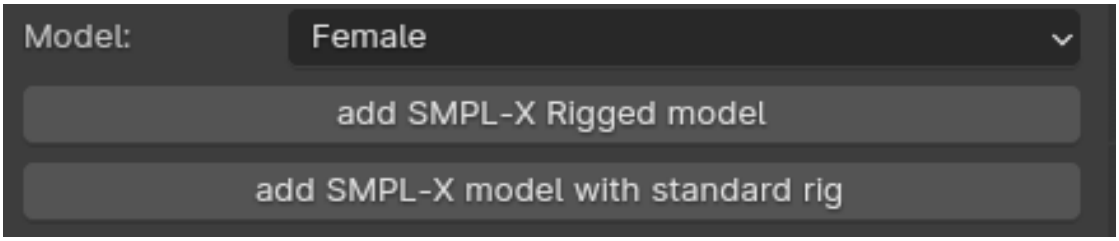


Figura 3.5: interfaccia grafica per l'aggiunta dei modelli.

legati alla raccolta di dati specializzati e contemporaneamente inserendo grande variabilità utile al modello per generalizzare il problema. ClothWild, che si occupa di ricostruire la posa e simulare abbigliamento a partire da immagini 2D, la stima accurata dei giunti 2D è cruciale per generare pose umane realistiche. ClothWild potrebbe trae vantaggio dall'approccio di Hand4Whole (Moon et al. [2022a]) per ottenere posizioni precise dei giunti delle mani e delle articolazioni MCP. . In particolare, l'utilizzo delle articolazioni metacarpo-falangee consente di prevedere movimenti realistici del polso, migliorando la rappresentazione delle pose anche quando le mani sono parzialmente nascoste o non completamente visibili nell'immagine. Hand4Whole si concentra sul miglioramento della stima della posa delle mani nel contesto della ricostruzione della mesh 3D dell'intero corpo umano. La sfida principale è legata alla difficoltà di predire accuratamente la posizione delle mani e delle dita, poiché le articolazioni delle mani sono complesse e di piccole dimensioni rispetto al resto del corpo. La corretta stima della posa delle mani è fondamentale, poiché contribuisce a rendere più credibile l'intero modello 3D, inclusi il volto e il corpo. Uno dei problemi principali nello stato dell'arte è che le tecniche di stima 3D non consideravano in modo adeguato le articolazioni metacarpo-falangee, che sono cruciali per determinare la rotazione del polso. Le articolazioni metacarpo-falangee sono fondamentali nella catena cinetica del corpo umano e forniscono informazioni essenziali per prevedere la rotazione del polso. Hand4Whole utilizza le caratteristiche di queste articolazioni per migliorare la stima delle rotazioni 3D del polso, assicurando che queste siano realistiche e connesse in modo fluido al resto del corpo. Il modello è in grado di ottenere previsioni plausibili anche quando le mani sono parzialmente occluse o fuori dal campo visivo. Il metodo proposto da Hand4whole affronta anche il problema della complessità delle dita, che presentano numerose articolazioni con movimenti indipendenti dal polso. I metodi precedenti utilizzavano caratteristiche generali del corpo per prevedere le rotazioni delle dita, ma questo approccio è limitato, poiché le informazioni globali del corpo non sono sufficientemente dettagliate per rappresentare correttamente la mano. Hand4Whole risolve questa limitazione separando le caratteristiche del corpo da quelle della mano nella fase di stima delle rotazioni delle dita, migliorando così la precisione del modello complessivo. La capacità di Hand4Whole di utilizzare informazioni sia globali, relative al corpo, sia locali, specifiche delle mani, assicura una transizione fluida tra la stima della posa del corpo e quella delle mani. Questo approccio integrato è particolarmente utile per ClothWild, che richiede una stretta coerenza tra le diverse parti del corpo per simulare correttamente l'interazione dell'abbigliamento con il modello umano. Inoltre, la stima precisa delle articolazioni 2D può migliorare l'animazione e la simulazione dei vestiti, poiché permette di calcolare con

maggiore accuratezza i movimenti e le deformazioni del tessuto attorno alle articolazioni mobili, come quelle delle mani. In sintesi, l'approccio descritto in Hand4Whole offre una soluzione avanzata per la stima delle pose delle mani e delle articolazioni del corpo, garantendo risultati più accurati e plausibili rispetto ai metodi precedenti. Integrando tecniche simili, ClothWild migliorara significativamente la qualità delle sue ricostruzioni 2D e 3D, ottenendo stime più precise delle posizioni dei giunti 2D, che sono essenziali per la simulazione e l'animazione degli abiti. Dopo la ricostruzione della posa 2D l'informazione viene utilizzata dai componenti di ClothWild per la ricostruzione della posa e dei vestiti. ClothWild è composto da due componenti principali, il primo componente è GAN che si basa su un modello 3D chiamato SMPLicit(Corona et al. [2021]). Questo modello è un'estensione del noto modello parametric SMPL, adattato per gestire anche l'abbigliamento. La rete generativa è incaricata di creare una rappresentazione tridimensionale del corpo e dei vestiti, partendo da un corpo umano in posa di default.La posa di default è T-pose il che significa avere il modello in piedi con le gambe leggermente divaricate e le braccia parallele alle spalle. La generazione della geometria dei vestiti viene realizzata attraverso l'apprendimento di informazioni da spazi latenti che rappresentano i diversi indumenti. Questo processo di generazione non avviene direttamente dall'immagine 2D, ma attraverso una trasformazione di alto livello in cui i vestiti vengono appresi separatamente dal corpo, migliorando così la qualità e l'indipendenza delle deformazioni degli indumenti rispetto alla posa del soggetto. L'addestramento di SMPLicit avviene tramite l'utilizzo di dataset di scansioni 3D, in cui il corpo e gli indumenti sono acquisiti contemporaneamente. Da queste scansioni, il modello impara una funzione implicita che descrive la geometria dell'abbigliamento in relazione alla posa del corpo e alla sua forma per mantenere la coerenza. Questo approccio permette al modello di generalizzare a nuove pose e forme corporee che non erano presenti nel dataset di addestramento, mantenendo l'abbigliamento congruo alla posa. Un aspetto fondamentale di SMPLicit è la sua capacità di generare indumenti a diverse risoluzioni. Poiché la rappresentazione implicita è continua, il livello di dettaglio può essere facilmente adattato in base alle esigenze dell'applicazione, consentendo di ottenere un compromesso tra qualità visiva e costi computazionali. Inoltre, SMPLicit non richiede una connessione esplicita tra il corpo e l'abbigliamento durante la fase di generazione. Questo significa che gli indumenti possono essere generati e posizionati attorno al corpo senza dover essere vincolati alla mesh del corpo stesso. Tale caratteristica consente una maggiore flessibilità nel modellare vestiti più complessi, come cappotti, gonne o abiti larghi, che non aderiscono direttamente alla superficie del corpo. Il secondo componente dei CLothWild è ClothNetZhang and Ma [2023], una rete di regressione progettata per predire i codici latenti a partire dall'immagine RGB di input. L'obiettivo principale di ClothNet è quello di mappare le caratteristiche visibili di un'immagine 2D di una persona verso una rappresentazione tridimensionale del corpo e degli indumenti, affrontando le sfide associate alla variabilità degli abiti, delle pose e delle condizioni di acquisizione. ClothNet opera come una rete di regressione che ha il compito di predire spazi latenti, i quali rappresentano sia gli indumenti sia il corpo sottostante. Questa rete è fondamentale per tradurre le informazioni bidimensionali di un'immagine in una rappresentazione 3D utilizzabile dalla rete generativa che crea la geometria tridimensionale degli abiti attorno al modello del corpo umano. A differenza di altri approcci basati esclusivamente su immagini sintetiche o dataset 3D completi, ClothNet si avvale

di supervisione debole, utilizzando dati 2D derivati da immagini reali, come maschere di segmentazione degli abiti o coordinate DensePose (Güler et al. [2018]), per addestrare la rete a generalizzare in condizioni non strutturate e complesse. DensePose crea una corrispondenza tra la superficie UV del corpo umano e i pixel dell'immagine bidimensionale, consentendo a ClothNet di gestire meglio l'ambiguità spaziale e la mancanza di informazioni di profondità che caratterizzano le immagini RGB. Questo approccio consente una maggiore precisione nel predire la geometria degli abiti e nel mappare la loro interazione con il corpo umano in movimento. La pipeline di ClothNet è progettata per funzionare con un modello 3D del corpo, in particolare il modello SMPL-X, per questo motivo si è scelto di integrare ClothWild in questo lavoro. La rete riceve in input un'immagine RGB e sfrutta feature extraction per isolare le caratteristiche visive rilevanti degli indumenti e delle pose. L'immagine viene prima analizzata per rilevare la silhouette della persona, le pose delle articolazioni e i segmenti degli abiti, dopodiché la rete regredisce verso una rappresentazione degli spazi latenti che descrivono la geometria degli indumenti e del corpo. ClothNet utilizza anche segmentazioni di abiti come supervisione, sfruttando un'architettura convoluzionale che estrae le caratteristiche rilevanti per ogni tipo di indumento, come giacche, pantaloni o vestiti, e genera una descrizione latente che può essere trasformata in geometria tridimensionale. Questa descrizione latente contiene informazioni sulla forma, la piegatura e il comportamento del tessuto, nonché sulla relazione tra il tessuto e la pelle o il corpo sottostante. Il modello è addestrato per distinguere tra i diversi tipi di indumenti e gestire il modo in cui questi si deformano in base alla posa e alla forma del corpo. Un altro punto cruciale dell'approccio di ClothNet è la generalizzazione delle pose e degli abiti. Durante l'addestramento, la rete impara a mappare non solo le caratteristiche degli abiti, ma anche la loro interazione con pose diverse e forme corporee variabili. Questo consente a ClothNet di adattarsi a una vasta gamma di situazioni, dalle pose statiche alle pose dinamiche complesse, con risultati robusti anche in presenza di occlusioni parziali o angoli di visuale difficili. La capacità di gestire queste variazioni rende ClothNet un sistema altamente flessibile, in grado di operare su immagini acquisite in contesti realistici, senza la necessità di condizioni controllate o dati volumetrici completi. Il processo di inferenza di ClothNet si basa su una combinazione di regressione e predizione condizionale, in cui l'immagine in input viene utilizzata per determinare i parametri di vestizione e gli spazi latenti associati alla geometria degli abiti. A differenza dei metodi tradizionali di ricostruzione 3D, che richiedono la disponibilità di dati 3D espliciti, ClothNet riesce a produrre una rappresentazione tridimensionale precisa utilizzando esclusivamente immagini 2D come supervisione. Questo approccio è particolarmente efficiente dal punto di vista computazionale, poiché elimina la necessità di ricostruzioni complete del corpo in ogni fase del processo, focalizzandosi invece sull'accuratezza degli indumenti e sulla loro interazione con la pelle. ClothNet inoltre viene utilizzato per la classificazione dei vestiti tramite una rete neurale profonda che utilizza un backbone ResNet (He et al. [2015]). Questa rete prende in input delle immagini e, attraverso il backbone, estrae delle caratteristiche rilevanti dall'immagine. Successivamente, il modulo ClothNet usa queste caratteristiche per predire i parametri relativi ai vestiti, come il tipo di abito e il suo genere. Durante l'allenamento, il sistema usa diverse funzioni di perdita per confrontare le previsioni con le etichette corrette, inclusa una perdita specifica per la classificazione dei vestiti (ClothClsLoss) e una per la classificazione del genere (GenderClsLoss). Le etichette utilizzate

comprendono una vasta gamma di capi di abbigliamento comuni, coprendo la maggior parte degli indumenti che possono essere indossati da una persona. Tra questi rientrano giacche, pantaloni lunghi e corti, gonne, scarpe, canotte e t-shirt. Questo insieme di etichette consente al sistema di classificare in modo efficace la maggior parte dei vestiti visibili in una fotografia, garantendo una suddivisione accurata degli indumenti indossati dagli individui nelle immagini analizzate. Il sistema prevede inoltre i parametri SMPL necessari per modellare la geometria 3D del corpo umano, a cui si aggiungono le mesh 3D dei vestiti presenti nelle classi del classificatore attraverso SMPLicit. I risultati ottenuti con ClothWild dimostrano che è possibile ottenere ricostruzioni di alta qualità degli indumenti anche in contesti non strutturati dimostrando la sua potenzialità in termini di robustezza su immagini "in-the-wild". Test empirici condotti su dataset pubblici, come MSCOCO (Lin et al. [2015]) e 3DPW (von Marcard et al. [2018]), dimostrano che la rete è in grado di generalizzare efficacemente su una vasta gamma di abiti e pose, superando le prestazioni di metodi precedenti, come PIFuHD (Saito et al. [2020]) e BCNet (Jiang et al. [2020]). L'architettura di ClothNet non solo migliora la qualità della ricostruzione, ma è anche più efficiente in termini di tempo di elaborazione superando le prestazioni di molti progetti sviluppati, inoltre migliorando il tempo di esecuzione e l'efficenza computazionale è migliore rispetto agli approcci standard come visibile in tabella 3.2.

Methods	**CD** $\downarrow$
PIFuHDSaito et al. [2020]	137.50
BCNet Jiang et al. [2020]	118.75
ICON Xiu et al. [2022]	75.00
PIFu Saito et al. [2020]	67.25
SMPLicit fits	45.66
ClothWild	**40.34**

Tabella 3.2: Paragone con i metodi presenti nello stato dell'arte in base allla distanza di Chamfer (CD), valori più bassi indicano prestazioni migliori.

L'output di CLothWild è un file .obj che contiene in un unica mesh il modello e i vestiti ricostruiti che rappresenta un risultato ridondante dato che la ricostruzione della posa viene affidata a SMPLify-x che porta ad un risultato più accurato. si è scelto quindi di modificare lo script in modo da separare la mesh dei vestiti dalla mesh del corpo, in modo da poter utilizzare il solo risultato di interesse.

3.3.1 Limiti

Uno dei principali problemi risiede nella difficoltà di gestire il divario significativo tra i dataset utilizzati per l'addestramento e le immagini reali catturate in ambienti non controllati. Questo "domain gap" causa spesso risultati inaccurati, poiché i modelli addestrati su immagini sintetiche con pose semplici non riescono a generalizzare correttamente su pose e condizioni di illuminazione complesse tipiche delle immagini "in-the-wild". Di conseguenza, la ricostruzione può risultare fragile di fronte a occlusioni, sovrapposizioni o pose umane particolarmente elaborate. Un'ulteriore limitazione riguarda l'ambiguità

nella profondità e la scarsa supervisione offerta dai dati bidimensionali. Essendo questi basati su viste di una singola telecamera, essi non garantiscono una rappresentazione accurata della geometria tridimensionale dei capi d'abbigliamento da diverse prospettive. La supervisione limitata alle sole segmentazioni 2D può condurre a ricostruzioni errate e incongruenze tra la proiezione 3D e la segmentazione bidimensionale originale. Le performance di ClothWild sono inoltre condizionate dalla capacità del modello generativo di rappresentare una vasta gamma di tessuti e stili di abbigliamento. Sebbene il modello sia in grado di gestire una serie di abiti standard, resta comunque limitato nella rappresentazione di dettagli più complessi come pieghe elaborate o materiali specifici, rendendo difficile la riproduzione fedele di tessuti con proprietà geometriche particolari. Un problema complesso affrontato da ClothWild riguarda la generazione delle gonne, la cui complessità è principalmente dovuta alla loro struttura geometrica e alle dinamiche fisiche che le caratterizzano. Le gonne, a differenza di pantaloni o magliette, presentano una superficie più ampia e libera che interagisce in modo più complesso con il corpo e con l'ambiente circostante. Uno dei principali ostacoli nella ricostruzione accurata delle gonne consiste nella corretta rappresentazione delle pieghe. Le gonne, specialmente quelle ampie o realizzate con tessuti leggeri, mostrano un movimento fluttuante che varia in base alla posa del corpo e ai movimenti della persona. Questo comportamento dinamico non può essere catturato direttamente nelle immagini statiche, rendendo difficile per i modelli di ricostruzione 3D prevedere con precisione la forma della gonna in tutte le pose. Le gonne, inoltre, introducono complessità aggiuntive nella ricostruzione a causa delle occlusioni. Quando un soggetto indossa una gonna, gran parte del tessuto può risultare parzialmente o completamente nascosto, sia a causa del corpo che copre parte del capo, sia per via di angoli di visuale ostruiti. Questa parziale visibilità rende la ricostruzione del capo imprecisa, in quanto il modello deve interpolare informazioni mancanti. Un altro aspetto problematico è la difficoltà nel distinguere il tessuto della gonna dal corpo o da altri indumenti, soprattutto quando i materiali e i colori sono simili. La segmentazione bidimensionale delle immagini può non fornire informazioni sufficienti per separare nettamente la gonna dal resto dell'abbigliamento o dal corpo, generando così errori nelle zone di contatto tra gli indumenti e il corpo stesso. ClothWild presenta problematiche quando viene utilizzata un'immagine di input contenente una gonna. Sebbene la classificazione dell'indumento avvenga correttamente, la generazione della gonna può risultare inefficace o produrre mesh errate, come evidenziato in figura 3.6 in cui la gonna viene ricostruita erroneamente nonostante la sua corretta rilevazione secondo il classificatore. Un'ulteriore problematica riscontrata riguarda la ricostruzione dei capi, che in condizioni particolari di illuminazione o nel caso di indumenti particolarmente aderenti può generare mesh con strutture discontinue, come mostrato in figura 3.7, in cui la maglia ricostruita presenta una serie di buchi nella mesh. Tali problematiche sono state affrontate attraverso l'implementazione del dataset precedentemente citato, che consente all'utente di utilizzare una serie di capi predefiniti anche nei casi in cui la ricostruzione presenti difetti. Il limite citato in precedenza si aggiunge alla difficoltà di garantire sempre la corretta dimensione dei vestiti generati; in particolare, la lunghezza delle maniche o dei pantaloni risulta talvolta inaccurata, portando alla produzione di capi dimensionalmente più corti rispetto alle proporzioni previste come si osserva in figura 3.8. Dopo alcuni test effettuati sull'utilizzo dei vertex group per vincolare il vestiario al modello, è emersa una problematica significativa,

(a) Immagine in input. Fonte : https://cdn.laredoute.com/cdn-cgi/image/width=1200,height=1200,fit=pad,dpr=1/products/3/a/c/3ac0c916c993c6c938c94c2481c61ce1.jpg

(b) Ricostruzione.

Figura 3.6: Ricostruzione fallimentare di una gonna.

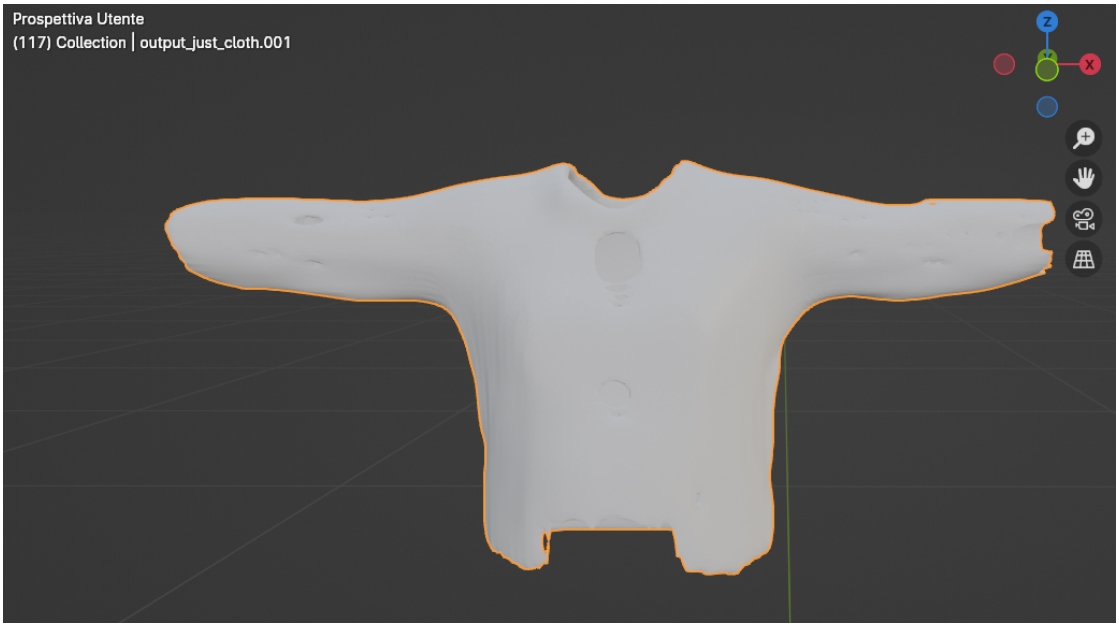
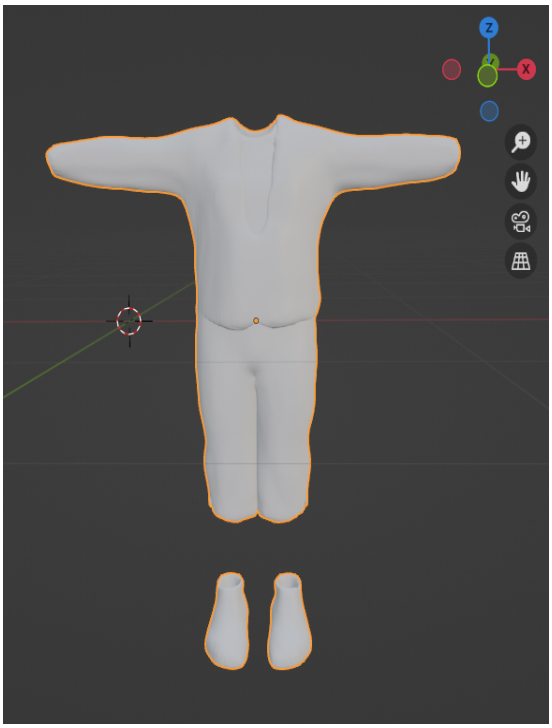


Figura 3.7: Ricostruzione vestiario che presenta discontinuità nella mesh.

le interazioni e i movimenti del vestiario rispetto all'armatura risultavano poco efficaci.

(a) Immagine in input. Fonte : https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcQoT5tnOPX2Y0Ee90ZtJzvMj9NSRZ_Fc2UVvf8dAmgvbQCvE1PW



(b) Ricostruzione.

Figura 3.8: Ricostruzione dimensionalmente errato.

Inizialmente, l'analisi della mesh del vestiario ha evidenziato un numero eccessivo di vertici disposti in maniera casuale. Per semplificare e alleggerire la mesh, è stato applicato un filtro di decimazione su Blender, ma questo non ha portato miglioramenti sostanziali. Dall'analisi della mesh è inoltre emerso che si trattava di una trimesh, come illustrato in figura 3.9. Pertanto, è stata ipotizzata la conversione della mesh in una quadmesh per migliorare l'efficacia dei movimenti e delle interazioni del vestiario. Il trimesh e il quadmesh sono due tipologie di mesh comunemente utilizzate nella grafica 3D, ognuna delle quali presenta caratteristiche specifiche che la rendono adatta a diversi ambiti di applicazione. La distinzione principale tra queste due mesh riguarda la forma delle facce che le compongono: il trimesh è costituito interamente da triangoli, mentre il quadmesh è composto da quadrilateri. Il trimesh è molto diffuso nei contesti in cui è importante avere una rappresentazione semplice e versatile della geometria. Essendo formato da triangoli, questo tipo di mesh può approssimare qualsiasi superficie 3D, anche quelle molto complesse o irregolari. La struttura a triangoli garantisce che la mesh possa mantenere la sua forma anche in presenza di curvature complesse, senza generare problemi topologici. I triangoli, essendo l'unità di base più semplice per descrivere una superficie. Nei trimesh,

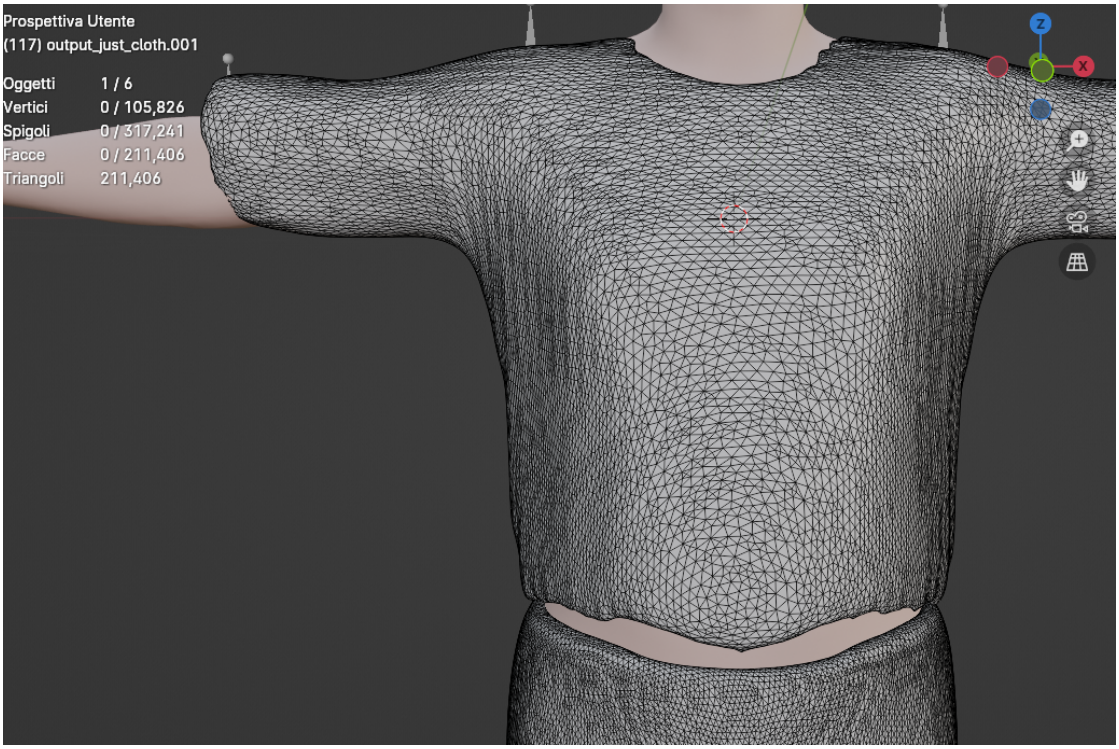


Figura 3.9: Visualizzazione dei vertici di una mesh relativa alla generazione del vestiario.

ogni punto di una superficie viene rappresentato in maniera uniforme senza richiedere particolari aggiustamenti, facilitando anche i calcoli relativi a collisioni e simulazioni fisiche, nei quali la semplicità di calcolo è spesso essenziale. Il quadmesh, invece, presenta una struttura molto diversa e offre vantaggi significativi nei contesti di modellazione e animazione. Essendo composto da quadrilateri, questo tipo di mesh crea una topologia più ordinata e regolare rispetto ai trimesh, particolarmente utile per la suddivisione e la deformazione delle superfici. Il quadmesh è preferito in applicazioni come il character design e l'animazione avanzata, in quanto supporta un controllo più preciso delle deformazioni, in cui è necessario adattare in modo naturale le forme durante il movimento. Il quadmesh riduce inoltre l'effetto di "artefatti" visivi che possono emergere con i trimesh, dove la presenza di triangoli può causare discontinuità. In sostanza, mentre i trimesh offrono vantaggi in termini di semplicità e stabilità nei calcoli computazionali, rendendoli ideali per applicazioni in tempo reale e per la gestione di superfici complesse senza deformazioni, i quadmesh sono preferiti in ambiti dove il controllo e la qualità della resa visiva sono cruciali. I quadmesh facilitano il lavoro di artisti e tecnici che devono applicare animazioni complesse o creare superfici dettagliate, in particolare quando è essenziale mantenere una topologia ordinata e coerente per effetti legati al movimento, per questo motivo si ipotizza un implementazione per la conversione in quadmesh. La modifica della mesh proposta presenta inoltre un miglioramenteo anche in termini di efficenza computazionale diminuendo di molto il numero di vertici nella mesh convertita come si osserva in figura

3.10 e oltre ad un miglioramento nell effetto nell'interazione con l'armatura del modello.

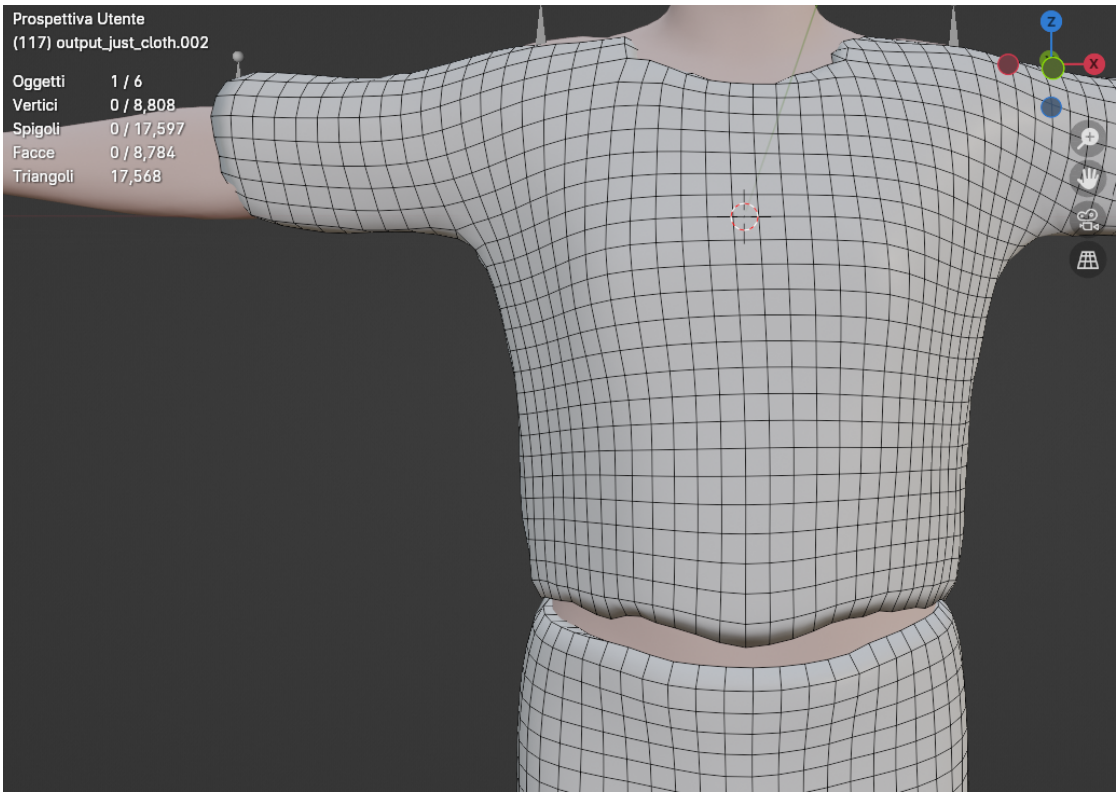
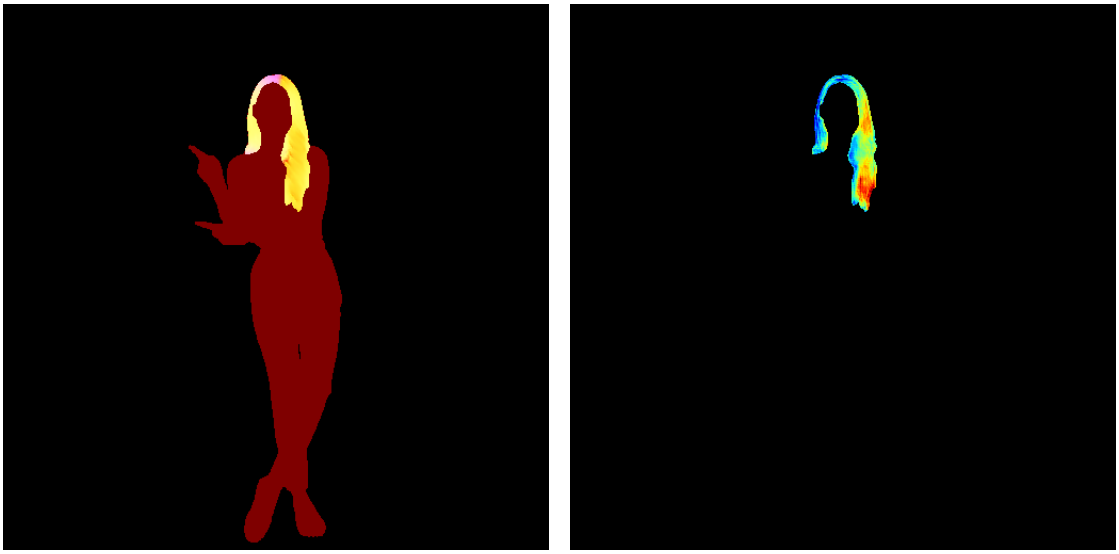


Figura 3.10: Visualizzazione dei vertici di una mesh convetita in quadmesh.

3.4 HairStep

Il metodo HairStep in un contesto di modellazione 3D dei capelli da una singola immagine RGB, rappresenta un significativo progresso nell'ambito della ricostruzione di capelli vista la complpessità del problema fornendo ottimi risultati Tale sfida è particolarmente complessa, poiché ottenere un dataset di immagini reali con geometrie accurate in 3D dei capelli è estremamente oneroso. Di conseguenza, i modelli sintetici sono ampiamente utilizzati, ma questo porta al problema del "domain gap", ovvero la discrepanza tra i dati sintetici e quelli reali che influisce negativamente sulla qualità della ricostruzione. HairStep introduce una rappresentazione intermedia composta da una strand map mappa delle ciocche) e una depth map come mostrato in figura 3.11. La strand map è una mappa vettoriale che rappresenta la direzione di crescita delle ciocche di capelli in 2D. A differenza delle mappe di orientamento comunemente utilizzate, che soffrono di rumore e ambiguità nella direzione di crescita, la strand map è una rappresentazione precisa e pulita ottenuta tramite annotazioni manuali o attraverso una rete neurale che imita questo processo. Si utilizza un dataset di riferimento chiamato HiSaZheng et al. [2023a], che



(a) Strand map. (b) Depth map.

Figura 3.11: Informazioni ricavate dall'immagine in input.

contiene immagini reali di capelli con annotazioni manuali delle ciocche, per supervisionare il modello durante l'addestramento. Le annotazioni sono state realizzate tracciando curve vettoriali dense lungo la direzione dei capelli, dal cuoio capelluto fino alle estremità. Per l'estrazione della strand map da immagini reali, viene impiegata una rete neurale di tipo U-NetRonneberger et al. [2015], che è addestrata utilizzando una combinazione di loss, come la pixel-wise L1 loss e la perceptual loss. La mappa risultante rappresenta le direzioni dei capelli su ogni pixel. Oltre alla strand map, si utilizza una depth map. Questo è essenziale per ottenere una ricostruzione completa e realistica dei capelli in 3D. Tuttavia, a causa della difficoltà di ottenere annotazioni di profondità precise per i capelli in immagini reali, è stato adottato un approccio di stima della profondità debolmente supervisionato, basato su annotazioni di profondità relative. Il dataset HiDaZheng et al. [2023a] fornisce annotazioni per coppie di pixel selezionate casualmente all'interno della regione dei capelli, per le quali viene indicato quale pixel appare più vicino rispetto all'altro. Questa annotazione viene utilizzata per addestrare un modello che predice una mappa di profondità completa per i capelli. Queste rappresentazioni non solo sono ricche di informazioni sufficienti per modellare accuratamente i capelli in 3D, ma sono anche facilmente inferibili da immagini reali. La strand map definisce la direzione della crescita dei capelli in maniera precisa, risolvendo uno dei principali problemi delle rappresentazioni precedenti basate su mappe di orientamento a 2D, che sono spesso influenzate dal rumore e dall'ambiguità direzionale ma forniscono importanti informazioni per la ricostruzione. La depth map, invece, fornisce informazioni relative alla profondità dei capelli rispetto alla fotocamera, elemento fondamentale per la ricostruzione tridimensionale. Una volta ottenute la strand map e la depth map, queste due rappresentazioni intermedie vengono combinate per guidare la ricostruzione della geometria dei capelli in 3D. Nello specifico,

la pipeline si basa su due campi impliciti per descrivere i capelli in 3D, il campo di occupazione e il campo di orientamento. Il campo di occupazione è una rappresentazione volumetrica che descrive la presenza o meno dei capelli in uno spazio 3D. Ogni punto nello spazio 3D può assumere un valore binario: 1 se si trova all'interno del volume dei capelli, 0 se si trova all'esterno. Durante l'addestramento, una grande quantità di punti viene campionata attorno alla superficie dei capelli per formare questo campo discreto. Questo campo è essenziale per definire i contorni del volume dei capelli nel modello 3D. Il campo di orientamento invece rappresenta la direzione di crescita dei capelli in 3D. A differenza delle mappe di orientamento a 2D utilizzate nei metodi precedenti, che non distinguono tra la radice e le punte dei capelli, il campo di orientamento tridimensionale cattura in modo completo la direzione dei capelli in ogni punto all'interno del volume. La pipeline prevede l'estrazione di orientamenti 3D unitari per punti densi lungo oltre 10.000 ciocche di capelli per ogni modello. Dopo la predizione dei campi di occupazione e di orientamento, la pipeline genera ciocche di capelli 3D utilizzando un metodo di crescita. Le ciocche vengono fatte crescere a partire dalle radici del cuoio capelluto secondo le direzioni fornite dal campo di orientamento, seguendo il metodo illustrato in NeuralHDHair (Wu et al. [2022a]), che garantisce la ricostruzione di modelli di capelli realistici. La pipeline implementa una versione modificata del modello NeuralHDHair, dove l'input principale non è la tradizionale mappa di orientamento, ma la rappresentazione ibrida pensata per HairStep, che combina la strand map e la depth map. Questa modifica ha permesso di ottenere miglioramenti significativi nella qualità della ricostruzione, eliminando gli artefatti e migliorando la fedeltà dei dettagli locali e globali dei capelli. Una delle innovazioni del lavoro è l'introduzione di nuove metriche per valutare la qualità dei modelli di capelli ricostruiti. Le principali metriche sono HairSale e HairRida oltre alla IoU di cui sono mostrati in tabella i risultati rispetto ai metodi presenti nello stato dell'arte 3.3.

Metodo	**IoU ↑**	**HairSale ↓**	**HairRida ↑**
NeuralHDHair (Orientation map)	77.56%	19.6	70.67%
NeuralHDHair (Strand map)	77.67%	**16 (-18.4%)**	72.37%
NeuralHDHair (HairStep)	77.22%	16.36 (-16.5%)	**76.79%**
DynamicHair (Orientation map)	56.39%	32.66	74.08%
DynamicHair (Strand map)	59.51%	26.53 (-18.8%)	73.42%
DynamicHair (HairStep)	59.14%	27.51 (-15.8%)	74.31%
HairNet (Orientation map)	57.15%	31.97	75.65%
HairNet (Strand map)	57.48%	28.6 (-10.5%)	74.81%
HairNet (HairStep)	57.01%	27.68 (-13.4%)	74.97%

Tabella 3.3: Confronto tra i metodi di ricostruzione dei capelli basato su IoU, HairSale, and HairRida con NeuralHDHair Wu et al. [2022b] e DynamicHair Wang et al. [2023].

HairSale misura l'errore medio angolare tra le direzioni di crescita dei capelli predette e quelle annotate. HairRida invece valuta l'accuratezza relativa della profondità attraverso un confronto tra le stime di profondità ottenute e le annotazioni di profondità relative. Gli esperimenti condotti su dataset reali e sintetici dimostrano la superiorità del metodo proposto rispetto alle tecniche presenti nello stato dell'arte, come HairNet (Zhou et al.

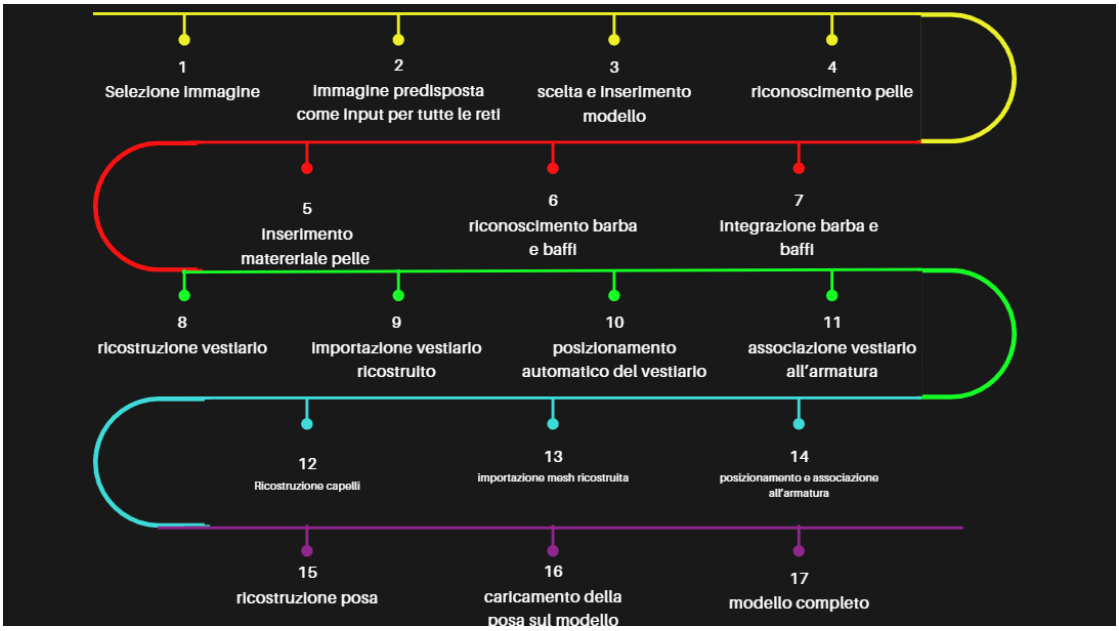


Figura 3.12: Pipeline di ricostruzione.

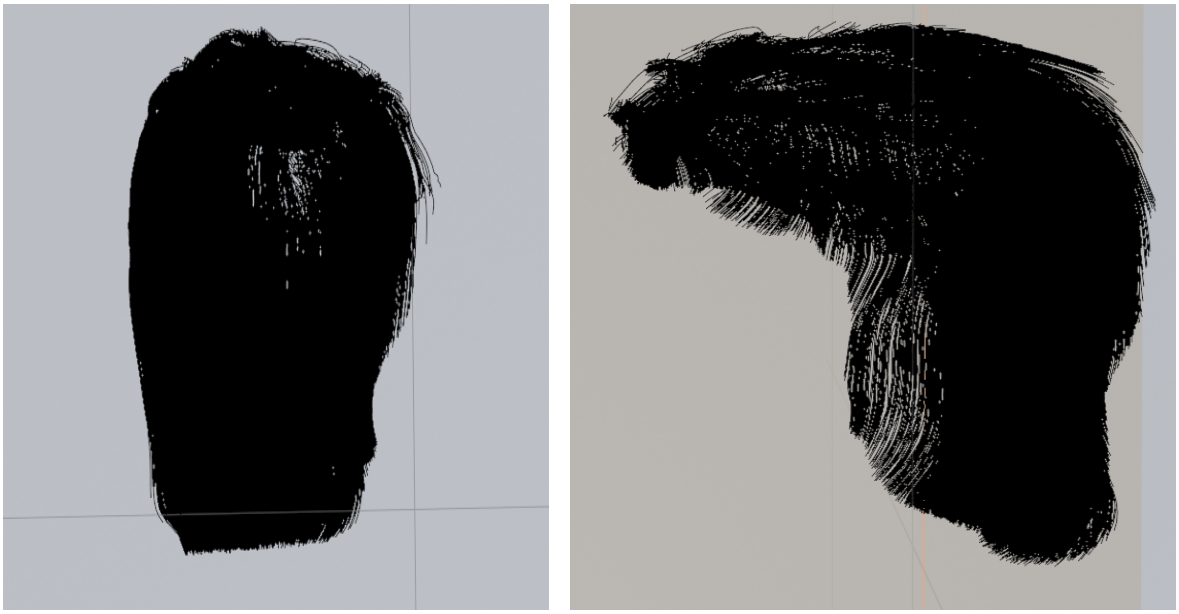
[2018b]) e DynamicHair (Yang et al. [2019]) per questo motivo si è scelto di utilizzare HairStep per la ricostruzione dei capelli.

3.4.1 Limiti

Le limitazioni di HairStep riguardano principalmente il divario tra i dati sintetici e quelli reali, nonostante l'introduzione della rappresentazione intermedia strand map e depth map. Anche se questa rappresentazione riesce a mitigare la differenza tra i domini, la ricostruzione 3D può soffrire in presenza di acconciature particolarmente intricate o non comuni, dove la capacità del modello di gestire complessità geometriche elevate risulta limitata. Le annotazioni debolmente supervisionate per la profondità possono introdurre imprecisioni, specialmente in dati reali, e produrre modelli che mancano di fedeltà rispetto alla realtà. Un ulteriore limite riscontrato è la precisione nella ricostruzione quando i capelli rappresentati sono corti come mostrato in figura 3.13. La parte posteriore dato l'utilizzo di una singola immagine non viene ricostuita correttamente come si può notare in figura 3.14 Un limite ulteriore si riscontra nei casi di capelli molto corti, dove la lunghezza è così ridotta da lasciare visibile la pelle sottostante, come mostrato in figura 3.15. In tali situazioni, la ricostruzione tende a non riconoscere correttamente quella zona come capelli, portando a una mesh parzialmente ricostruita. Inoltre, la presenza di elementi simili, come ciuffi d'erba, può confondere il modello che li identifica erroneamente come capelli, generando un output incorretto, come illustrato in figura 3.16, derivato dall'immagine 3.15 usata come input. Un'altra limitazione riguarda la dipendenza da dataset sintetici, che per quanto estesi, non riescono a catturare l'intera variabilità delle acconciature reali, influenzando la capacità del modello di generalizzare su nuove immagini. La qualità

Figura 3.13: Immagine di esempio utilizzata per la ricostruzione di capelli corti. Fonte : https://images.app.goo.gl/V7p6LH7ae8F2KoUW6

della mappa di profondità può risentire dell'adattamento debole ai dati reali, portando a ricostruzioni 3D che non rappresentano accuratamente la complessità delle acconciature osservate nell'immagine di input. Questi artefatti si manifestano soprattutto quando si tenta di modellare stili di capelli con occlusioni o variazioni di densità e forma, situazioni che la pipeline fatica a gestire pienamente.



(a) Ricostruzione capelli vista frontale. (b) Ricostruzione capelli vista laterale.

Figura 3.14: Esempio di limite di Hairstep.

3.5 Riconoscimento Barba

La classificazione della presenza di barba viene realizzata tramite una pipeline di machine learning avanzata, progettata per analizzare e classificare immagini (Chifffa [2024]). La CNN utilizzata svolge una classificazione simultanea della presenza di barba e cappello attraverso un'architettura a doppia uscita, dove ciascun output è dedicato a uno specifico compito di classificazione. Il processo inizia con un dataset di volti organizzati in quattro categorie: senza accessori, con cappello, con barba e con entrambi. Il dataset viene preprocessato per garantire la standardizzazione delle immagini, migliorando la precisione e la robustezza del modello. Durante l'addestramento, il dataset è diviso in training e validation set; la CNN è ottimizzata per minimizzare le funzioni di perdita associate a ciascuna classificazione. Gli algoritmi di ottimizzazione, come Adam (Kingma and Ba [2017]) e il gradient descent (Ruder [2017]), permettono al modello di apprendere rappresentazioni visive di cappelli e barbe. Nella fase di validazione, il modello viene valutato su un set di dati separato, non usato durante l'addestramento, per analizzare le prestazioni attraverso metriche standard quali precision, recall e F1 score per ciascuna classe. Questa validazione consente di valutare la capacità del modello di generalizzare e la sua affidabilità su dati non visti in precedenza.

3.5.1 Limiti

La maggiore problematica è legata al fallimento della classificazione che utilizza un classificatore Haar Arreola et al. [2019]. L'utilizzo di un classificatore Haar per il rilevamento

Figura 3.15: Immagine usata come input. Fonte : https://www.facebook.com/people/Football-page/61551889385832/

del viso è una tecnica consolidata per identificare volti in immagini, basata su un algoritmo di classificazione rapido e relativamente accurato per volti orientati frontalmente. Tuttavia, presenta limiti significativi per il rilevamento di volti con angolazioni diverse dal fronte, poiché le caratteristiche visuali cambiano e non corrispondono ai pattern appresi dal classificatore.

3.6 Riconoscimento carnagione

Lo script per il riconoscimento della pelle definisce un operatore che rileva automaticamente il colore dominante della pelle da un'immagine in input e applica tale colore come materiale su una mesh selezionata.Il processo è la funzione che accetta come input un percorso di immagine e tenta di caricare l'immagine usando OpenCV. Se l'immagine viene caricata correttamente, viene convertita da BGR a RGB. L'algoritmo utilizza quindi la funzionalità di rilevamento dei volti di OpenCV, basata su Haar Cascade, per trovare la posizione del volto all'interno dell'immagine convertita in scala di grigi. Se non viene

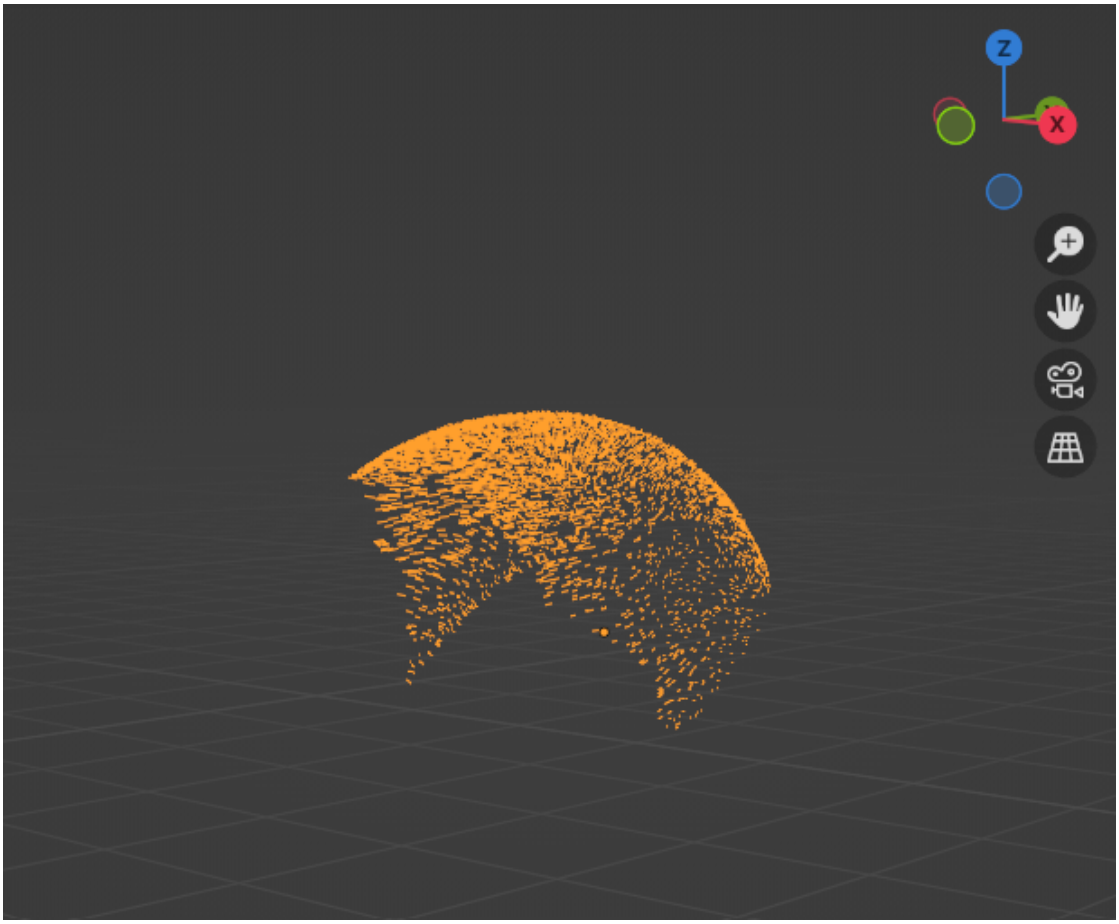


Figura 3.16: Ricostruzione problematica dell'immagine di input.

rilevato alcun volto, il processo si interrompe con un messaggio di avviso. In caso di successo, la regione del volto rilevata viene estratta e convertita in una serie di pixel RGB. A questo punto, viene applicata una maschera per selezionare solo i pixel che rientrano in un intervallo plausibile per il colore della pelle, definito da limiti inferiori e superiori in valori RGB. Questi pixel vengono filtrati e, se non ci sono pixel che corrispondono ai criteri di colore della pelle, il processo si interrompe. Se vengono identificati pixel della pelle, il colore dominante tra questi pixel viene calcolato utilizzando la libreria CounterPython Software Foundation [2023b], che permette di determinare il colore più frequente all'interno della selezione filtrata. Questo colore dominante viene restituito come un vettore RGB. Successivamente si utilizza questo colore per impostare il materiale della mesh selezionata in Blender. Dopo aver recuperato il percorso dell'immagine dalla scena di Blender e calcolato il colore dominante della pelle, la funzione verifica che l'oggetto selezionato sia una mesh valida. Se l'oggetto non ha materiali associati, ne viene creato uno nuovo in caso contrario, viene utilizzato il primo materiale già presente nella mesh. All'interno dell'albero dei nodi del materiale, il colore dominante viene applicato al nodo "Principled

BSDF" come colore di base, convertendo i valori RGB in un intervallo da 0 a 1 per essere compatibili con il sistema di Blender.

3.6.1 Limiti

Questo approccio per rilevare il colore della pelle e applicarlo a una mesh in Blender presenta alcune limitazioni che ne influenzano l'affidabilità e la precisione. L'algoritmo si basa sulle Haar Cascades di OpenCV per il rilevamento dei volti, una tecnica che risulta meno accurata in condizioni di illuminazione non uniforme o in presenza di volti parzialmente visibili. In questi casi, il rilevamento potrebbe non identificare correttamente il volto, causando un'interruzione del processo. Anche la scelta della maschera RGB per isolare i pixel della pelle introduce restrizioni: essendo definita con valori preimpostati, questa tecnica non risponde bene a variazioni di tonalità dovute all'illuminazione o a riflessi ambientali, il che può portare a una selezione insufficiente o erronea dei pixel rappresentativi della pelle. La stima del colore dominante basata su Counter risulta inoltre semplificata, poiché l'approccio ignora tecniche di clustering più sofisticate, come K-means Jin and Han [2010], e non considera sfumature o gradienti all'interno dell'area selezionata, limitando la precisione del colore finale calcolato. Il rischio che il colore applicato sia errato non è trascurabile ma è comunque semplice per l'utente modificare il materiale per ottenere il risultato desiderato.

3.7 Tecnologie utilizzate

3.7.1 Python

Python (Python Software Foundation [2023a]) è un linguaggio di programmazione ad alto livello noto per la sua sintassi semplice e leggibile. La sua popolarità è cresciuta esponenzialmente nel campo del machine learning e dell'intelligenza artificiale grazie alla vasta gamma di librerie e pacchetti che ne estendono le funzionalità. In particolare, Python si distingue per la sua capacità di gestire dati complessi, il che è essenziale per i progetti di machine learning. Uno degli aspetti che rende Python così adatto al machine learning è la disponibilità di librerie potenti come NumPy (Harris et al. [2020]) e Pandas (McKinney et al. [2010]). NumPy offre supporto per array multidimensionali e operazioni matematiche ad alte prestazioni, mentre Pandas facilita la manipolazione e l'analisi dei dati. Queste librerie sono fondamentali per la preparazione e la gestione dei dati, che è una fase cruciale in qualsiasi progetto di machine learning. Nell'ambito del machine learning stesso, librerie come Scikit-learn (Pedregosa et al. [2011]), TensorFlow (Abadi et al. [2016]) e PyTorch (Paszke et al. [2019a]) offrono strumenti per la costruzione, l'addestramento e la valutazione di modelli. Scikit-learn è particolarmente apprezzata per le sue implementazioni di algoritmi di apprendimento supervisionato e non supervisionato, rendendo facile l'implementazione di modelli tradizionali. TensorFlow e PyTorch, d'altra parte, sono utilizzati per il deep learning, consentendo agli sviluppatori di costruire reti neurali complesse e gestire grandi volumi di dati. Per quanto riguarda i progetti che coinvolgono la grafica 3D e la visualizzazione, Python offre pacchetti specifici che ampliano

ulteriormente le sue capacità. Una libreria di rilievo è Open3D (Zhou et al. [2018a]), progettata per l'elaborazione di dati 3D e la visualizzazione di nuvole di punti, mesh e volumi. Open3D permette di implementare facilmente algoritmi di machine learning su dati 3D, rendendolo ideale per applicazioni che vanno dalla ricostruzione 3D alla scansione di oggetti. Un altro strumento importante è trimesh (Dawson-Haggerty et al.), una libreria per la gestione e l'analisi di geometrie 3D. Trimesh consente di caricare, modificare e visualizzare modelli 3D, facilitando l'integrazione di algoritmi di machine learning in flussi di lavoro che richiedono una manipolazione complessa delle forme. Attraverso trimesh, gli sviluppatori possono combinare capacità di machine learning con la geometria 3D, creando applicazioni più avanzate e innovative. Infine, la capacità di Python di interagire con altre tecnologie e linguaggi, unita alla sua vasta comunità di sviluppatori e ricercatori, fa di questo linguaggio lo strumento migliore nel panorama attuale per lo sviluppo di un progetto riguardante l'intelligenza artificiale, soprattutto per quanto riguarda applicazioni avanzate nel dominio 3D.

3.7.2 Anaconda

Anaconda è una distribuzione open-source di Python e R, concepita principalmente per il machine learning e l'analisi dei dati. Questa piattaforma semplifica l'installazione e la configurazione di Python e dei pacchetti più utilizzati, fornendo un installer unificato che include sia il linguaggio di programmazione che le librerie necessarie. Uno degli elementi distintivi di Anaconda è il gestore di pacchetti Conda in figura 3.17, che facilita l'installazione, l'aggiornamento e la gestione delle librerie, oltre a garantire la corretta gestione delle dipendenze tra i vari pacchetti che risulta molto gravoso e problematico nella maggior parte dei progetti sviluppati in Python. Un'altra caratteristica fondamentale di Anacon-

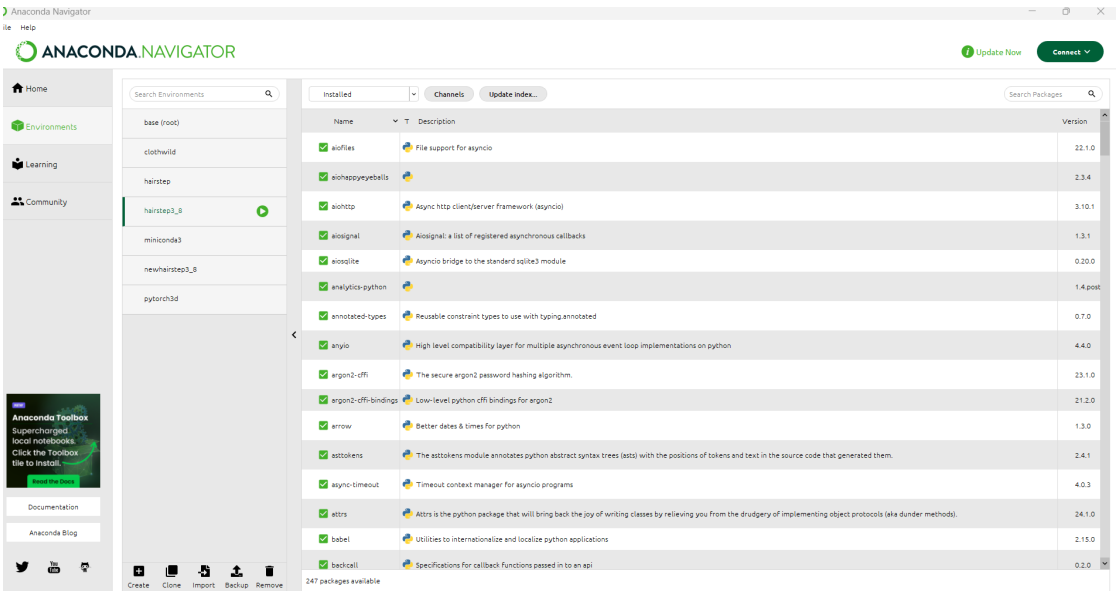


Figura 3.17: Interaccia di Anaconda .

da è la possibilità di creare ambienti virtuali isolat, consentendo agli utenti di lavorare su progetti con dipendenze diverse senza rischiare conflitti. Questa funzionalità si rivela particolarmente utile in contesti in cui è necessario utilizzare versioni specifiche di librerie o di Python stesso. Anaconda include anche una vasta gamma di pacchetti preinstallati, come NumPy, pandas, Matplotlib e scikit-learn, ma consente anche l'installazione di ulteriori pacchetti tramite Conda o pip. Anaconda è compatibile con vari sistemi operativi, tra cui Windows, macOS e Linux, rendendola una scelta versatile per gli utenti di diverse piattaforme. Grazie alla sua robustezza e alla sua ampia gamma di funzionalità, Anaconda rimane una delle scelte migliori per sviluppare progetti di ricerca in modo che siano facilmente riproducibili.

3.7.3 Virtual environment

Il virtual environment, abbreviato venv, è un modulo integrato in Python che consente agli sviluppatori di creare ambienti virtuali isolati per la gestione delle dipendenze dei progetti. Questa funzionalità è fondamentale per evitare conflitti tra diverse versioni di pacchetti e librerie, specialmente quando si lavora su più progetti contemporaneamente. Ogni ambiente virtuale creato con venv è indipendente, il che significa che le librerie installate in un ambiente non influenzano quelle presenti in un altro. L'utilizzo di un venv inizia con la creazione di un ambiente virtuale, il che può essere fatto facilmente utilizzando un comando da terminale. Questo comando genera una nuova directory, tipicamente chiamata venv, che contiene una copia dell'interprete Python e una directory dedicata per le librerie installate. Una volta creato, l'ambiente può essere attivato, il che permette agli sviluppatori di lavorare in un contesto dove le versioni delle librerie possono essere specificate e controllate in modo preciso. Attivare un ambiente virtuale modifica il percorso dell'interprete Python in modo che punti a quello dell'ambiente isolato, permettendo di installare pacchetti specifici per quel progetto senza modificare le installazioni globali di Python. Questa caratteristica si rivela particolarmente utile quando si utilizzano librerie che richiedono versioni diverse o quando si desidera testare il funzionamento di un progetto con specifiche dipendenze senza rischiare di compromettere l'integrità di altri progetti o applicazioni. Questa funzionalità mitiga una buona parte dei problemi derivanti dall'installazione di un progetto dato che la probabilità che un progetto sviluppato nel passato utilizzi versioni deprecate di alcuni pacchetti. La gestione delle dipendenze in venv avviene attraverso il gestore di pacchetti pip, che consente di installare, aggiornare e rimuovere librerie in modo semplice. Utilizzando un file chiamato requirements.txt, gli sviluppatori possono elencare tutte le dipendenze necessarie per un progetto. Questo file può essere utilizzato per ricreare facilmente l'ambiente in un'altra macchina, assicurando che il progetto funzioni correttamente anche in contesti diversi. L'isolamento degli ambienti virtuali facilita anche il test e la distribuzione dei progetti. Gli sviluppatori possono verificare la compatibilità delle loro applicazioni con diverse versioni di librerie o Python stesso, garantendo una maggiore stabilità e prevedibilità. Inoltre, durante la distribuzione di un'applicazione, è possibile fornire informazioni dettagliate su quali dipendenze sono necessarie e come installarle, migliorando così l'esperienza degli utenti finali e dei collaboratori.

3.7.4 Visual Studio Code

Visual Studio Code (Corporation [2024]), comunemente noto come VS Code, è un editor di codice sorgente sviluppato da Microsoft, progettato per essere un ambiente di sviluppo versatile e leggero. Rilasciato nel 2015, ha rapidamente guadagnato popolarità tra sviluppatori di vari livelli e discipline grazie alla sua interfaccia intuitiva, alla potenza di personalizzazione e alle numerose funzionalità integrate. Uno dei punti di forza di VS Code è la sua capacità di supportare una vasta gamma di linguaggi di programmazione tra cui Python. Grazie a un sistema di estensioni altamente sviluppato, gli utenti possono ampliare le funzionalità dell'editor per adattarlo a specifiche esigenze di sviluppo. Gli sviluppatori possono eseguire comandi direttamente all'interno dell'editor, senza

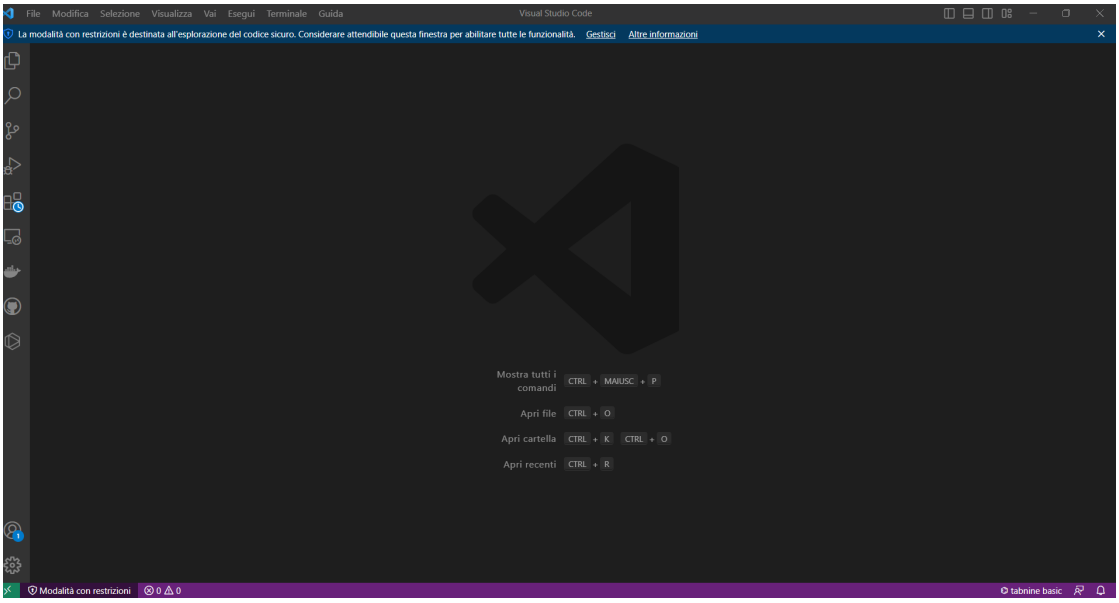


Figura 3.18: Interfaccia di Visual Studio Code.

dover passare a un terminale esterno. Questa funzionalità migliora notevolmente l'efficienza, consentendo di gestire operazioni di sviluppo e debugging in un ambiente unico e centralizzato. La gestione dei progetti è facilitata da funzionalità come il controllo delle versioni integrato, che supporta strumenti come Git (Chacon and Straub [2014]). Gli sviluppatori possono visualizzare modifiche, gestire branch e risolvere conflitti senza mai lasciare l'editor. Inoltre, VS Code offre un'interfaccia di debug robusta, permettendo di impostare punti di interruzione, eseguire codice passo-passo e analizzare variabili, il tutto in un ambiente visivo.

3.7.5 Pycharm

PyCharm è un ambiente di sviluppo integrato progettato specificamente per il linguaggio di programmazione Python, sviluppato da JetBrains. Riconosciuto per la sua interfaccia intuitiva e ricca di funzionalità, PyCharm offre strumenti avanzati per la scrittura,

il debug e il test del codice, rendendolo uno strumento molto utilizzato. Una delle caratteristiche distintive di PyCharm è il suo potente editor di codice mostrato in figura 3.19, che supporta il completamento automatico, la navigazione intelligente e il refactoring del codice. Questi strumenti aiutano a migliorare la produttività degli sviluppatori, riducendo il tempo necessario per scrivere e ottimizzare il codice. Inoltre, include un sistema di controllo della versione integrato, che consente di gestire facilmente le modifiche al codice e di collaborare con altri membri del team. PyCharm è dotato di strumenti

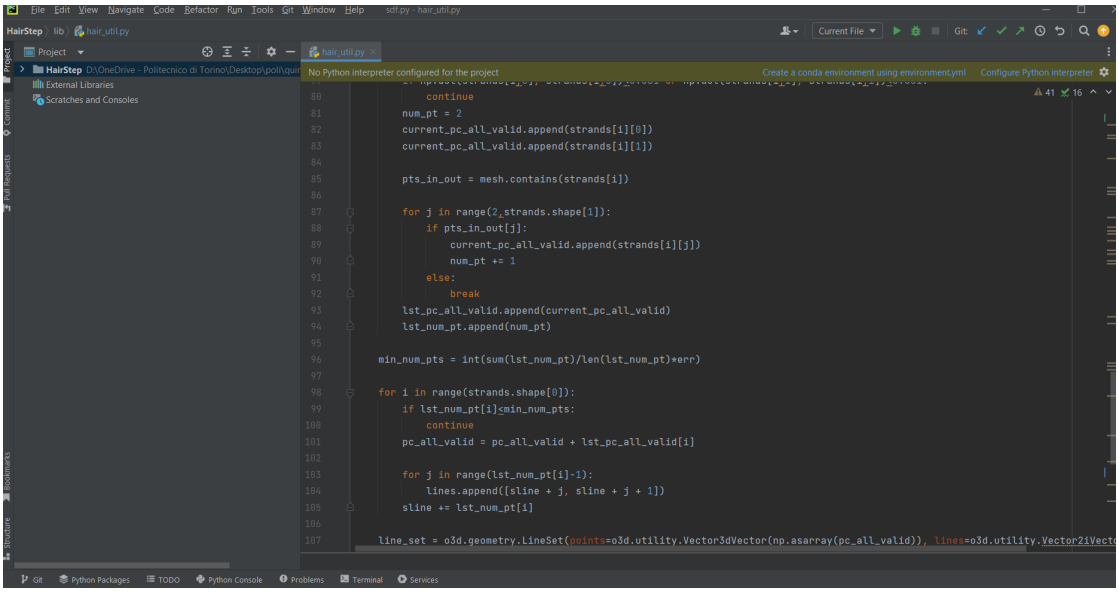


Figura 3.19: Interfaccia di Pycharm.

di debugging avanzati, che permettono agli utenti di eseguire il codice passo dopo passo, impostare punti di interruzione e monitorare le variabili in tempo reale. Questo facilita l'identificazione e la risoluzione di problemi, migliorando così l'affidabilità delle applicazioni sviluppate. Inoltre, PyCharm supporta il testing automatico e la creazione di suite di test, semplificando il processo di verifica della funzionalità del codice. Un altro aspetto importante di PyCharm è la sua integrazione con una vasta gamma di framework e librerie Python, come Pandas. Questa integrazione consente agli sviluppatori di sfruttare appieno le funzionalità di questi strumenti all'interno dell'IDE, migliorando l'efficienza dello sviluppo e facilitando la creazione di applicazioni web e analisi di dati. PyCharm offre anche strumenti per il supporto dello sviluppo con integrazioni per librerie come NumPy e Matplotlib (Hunter [2007]) per la visualizzazione dei dati. Gli utenti possono lavorare con notebook Jupyter direttamente all'interno dell'IDE, fornendo un ambiente coeso per l'analisi dei dati e la visualizzazione.

3.7.6 Blender

Blender (Blender Foundation [2024]) è un software di modellazione 3D, animazione, rendering e creazione di effetti visivi open source. Il codice sorgente di Blender è liberamente

accessibile e modificabile. Questo incoraggia una comunità attiva di sviluppatori che possono contribuire allo sviluppo del software e dei suoi add-on. L'accessibilità consente a chiunque di partecipare e di sfruttare Blender senza costi di licenza, rendendolo una scelta ottimale anche per la sua semplicità d'utilizzo e la sua interfaccia grafica molto intuitiva mostrata in figura 3.20. Blender offre un'API Python robusta che consente agli

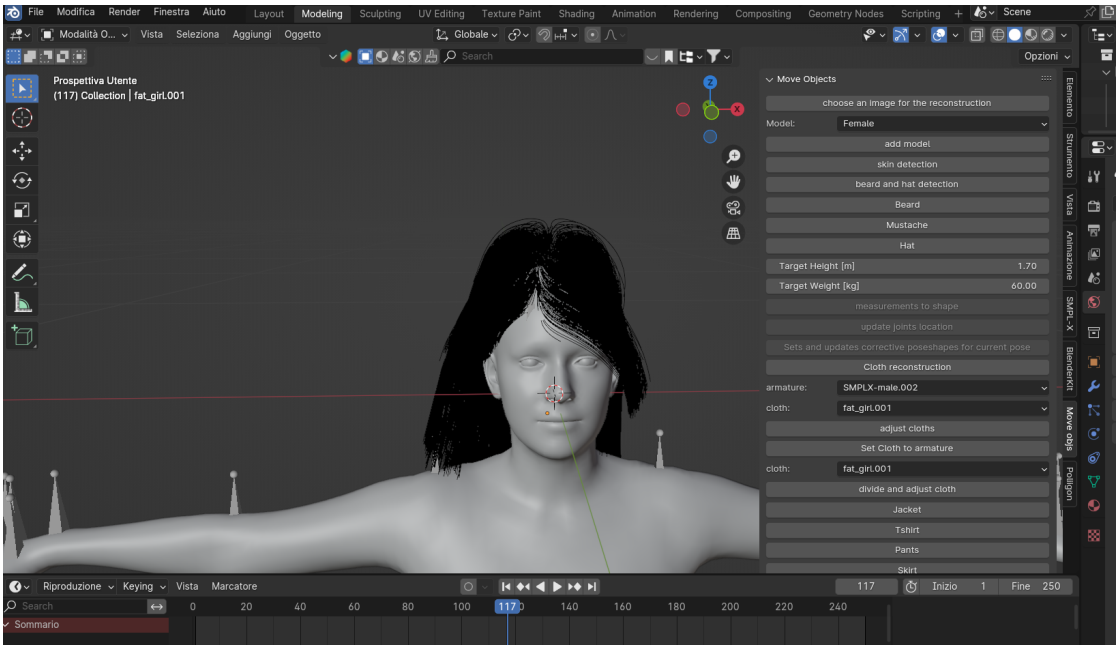


Figura 3.20: Interfaccia di Blender.

sviluppatori di creare add-on che possono estendere e personalizzare le funzionalità del software. Gli sviluppatori possono utilizzare Python per automatizzare compiti, creare nuovi strumenti, implementare flussi di lavoro personalizzati. Questa flessibilità rende Blender un ambiente ideale per lo sviluppo di add-on. Blender supporta l'importazione e l'esportazione di vari formati di file, consentendo l'integrazione con altri strumenti e flussi di lavoro. Gli sviluppatori possono creare add-on che facilitano l'interazione con software esterni senza preoccuparsi del formato dati utilizzato per gli oggetti essendo fornito il supporto per la maggior parte delle tipologie di file. Gli add-on consentono agli utenti di personalizzare Blender per soddisfare esigenze specifiche. Che si tratti di migliorare il flusso di lavoro, automatizzare processi ripetitivi o aggiungere nuovi strumenti.

3.7.7 Meshlab

MeshLab è un software open source dedicato alla gestione e all'elaborazione di mesh 3D. Sviluppato principalmente per l'elaborazione di dati ottenuti da scansioni 3D. La sua interfaccia intuitiva consente agli utenti di visualizzare, modificare e analizzare modelli 3D con facilità, in figura 3.21 l'interfaccia di MeshLab. Uno degli aspetti principali di MeshLab è la sua capacità di gestire una varietà di formati di file 3D, il che lo rende uno strumento

versatile per la lavorazione di modelli provenienti da diverse fonti. Grazie alla sua potenza di elaborazione, gli utenti possono eseguire operazioni complesse, come la pulizia delle mesh, la riduzione del numero di poligoni, la generazione di texture e la creazione di superfici triangolate a partire da nuvole di punti. La suite di strumenti di MeshLab

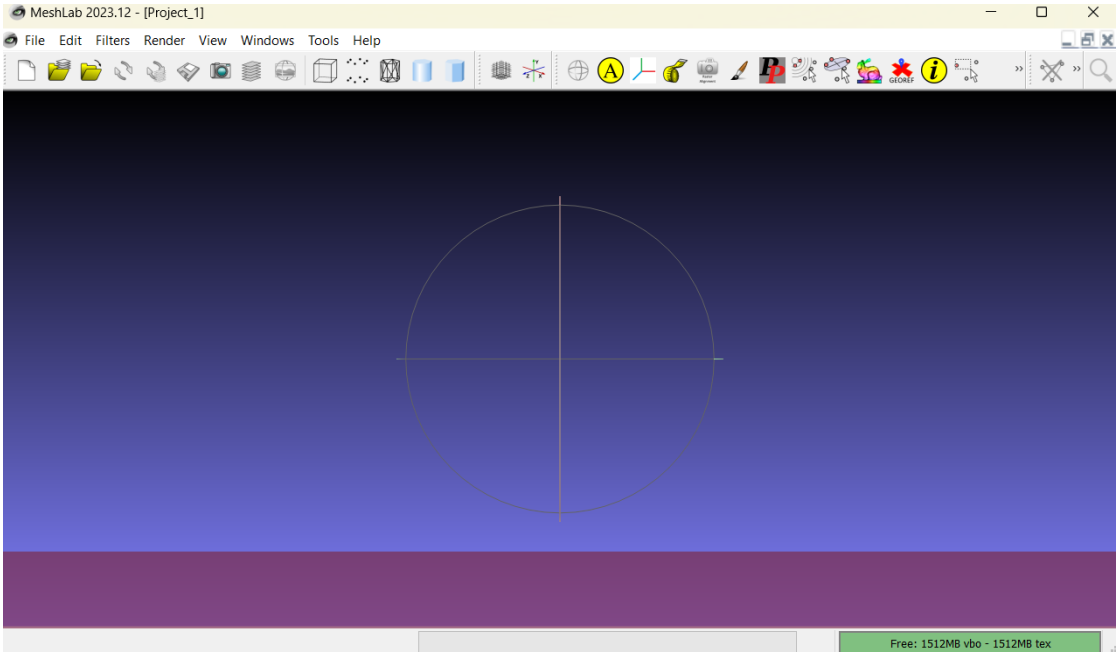


Figura 3.21: Interfaccia di Meshlab.

include filtri avanzati che permettono di migliorare la qualità dei modelli 3D. Questi strumenti possono essere utilizzati per rimuovere rumore, allineare mesh diverse e ottimizzare la topologia. Inoltre, MeshLab offre funzionalità di visualizzazione avanzate, consentendo agli utenti di esplorare i loro modelli in modo interattivo e di analizzare dettagli specifici. MeshLab si distingue anche per la sua capacità di eseguire analisi quantitative sui modelli. Gli utenti possono calcolare variabili come l'area, il volume e la curvatura delle mesh, fornendo informazioni preziose per la progettazione e l'analisi. Questa funzionalità è particolarmente apprezzata in contesti accademici e di ricerca, dove i dati quantitativi sono essenziali. Il software supporta anche l'importazione e l'esportazione di modelli in vari formati, facilitando così l'integrazione con altri strumenti di progettazione e modellazione. Gli utenti possono, ad esempio, preparare modelli per la stampa 3D, assicurandosi che siano ottimizzati..

3.7.8 Mixamo

Mixamo (Adobe Systems Incorporated [2024b]) è una piattaforma di Adobe (Adobe Systems Incorporated [2024a]), pensata per semplificare il processo di rigging e animazione dei modelli 3D. Mixamo consente di partire da una libreria di personaggi predefiniti o di caricare modelli personalizzati, rendendo accessibile l'animazione senza richiedere complesse

competenze tecniche. Una delle caratteristiche distintive della piattaforma è il rigging automatico, un processo che permette di trasformare un modello statico in un personaggio animabile attraverso un'analisi intelligente della struttura del modello e l'applicazione di uno scheletro virtuale. Per i modelli caricati dagli utenti, il processo di rigging comincia con la loro analisi da parte di Mixamo, che riconosce la mesh e individua le posizioni fondamentali delle articolazioni. Una volta caricato il modello nei formati supportati, come FBX, OBJ o ZIP, l'interfaccia, monstrata in figura 3.22, richiede all'utente di posizionare alcuni marker strategici su punti chiave del corpo. Questi marker, posizionati, ad esempio, sul mento, sui polsi, sulle ginocchia e sulle caviglie aiutano Mixamo a identificare con maggiore precisione la struttura anatomica del personaggio e a delineare la posizione esatta delle giunture. La precisione in questa fase è fondamentale per assicurare che il rigging automatico generi uno scheletro appropriato, che seguirà in modo naturale e realistico le proporzioni e l'anatomia specifica del modello caricato. Dopo aver confermato i marker, Mixamo procede con la creazione dell'armatura digitale del modello, unendo ogni giuntura in modo che le parti della mesh rispondano correttamente ai movimenti. Questa fase include anche l'assegnazione dei pesi, cioè la determinazione di come le diverse aree della mesh come le braccia, il busto, le gambe in modi che si muovano in relazione all'armatura. L'assegnazione dei pesi è una fase complessa, ma automatizzata, durante la quale Mixamo analizza la distribuzione delle articolazioni e crea transizioni fluide tra i segmenti del corpo. Questo consente alla mesh di mantenere forme naturali durante il movimento, evitando deformazioni innaturali o movimenti rigidi, che potrebbero risultare non realistici in fase di animazione. Il rigging automatico è progettato per rispondere dinamicamente ai dati specifici di ogni modello, il che rende Mixamo una risorsa efficace per trasformare rapidamente e in modo preciso una mesh in un personaggio 3D completamente riggato . Una volta completato il rigging, il sistema permette di visualizzare il risultato direttamente nella piattaforma, consentendo di applicare alcune animazioni predefinite per verificare la qualità del rigging e assicurarsi che i movimenti siano fluidi e privi di artefatti. Dopo questa verifica, il modello è pronto per essere scaricato, completo di armatura e pesi, per un uso immediato in progetti di animazione o videogiochi. Mixamo offre anche la possibilità di personalizzare i parametri delle animazioni, come la velocità e l'ampiezza dei movimenti, così che ogni animazione possa adattarsi perfettamente al ritmo e allo stile del progetto. In questo modo, Mixamo si distingue non solo per l'efficacia del rigging automatico, che consente di risparmiare ore di lavoro rispetto ai flussi di lavoro tradizionali, ma anche per la qualità dei risultati finali, che spesso risulta paragonabile a quella ottenuta con tecniche manuali. Questo approccio avanzato e intuitivo ha reso Mixamo uno strumento essenziale nella pipeline di produzione di contenuti 3D e un alleato prezioso per chi desidera integrare animazioni professionali senza la complessità e i tempi richiesti da software di rigging tradizionali.

3.7.9 GitHub

GitHub (GitHub [2024]) è una piattaforma ampiamente adottata per il controllo di versione e la collaborazione nello sviluppo software, creata per integrare Git e ottimizzare il lavoro di sviluppatori e team in un ambiente condiviso. Permette di gestire versioni multiple di progetti, offrendo strumenti per tenere traccia delle modifiche, gestire e risolvere

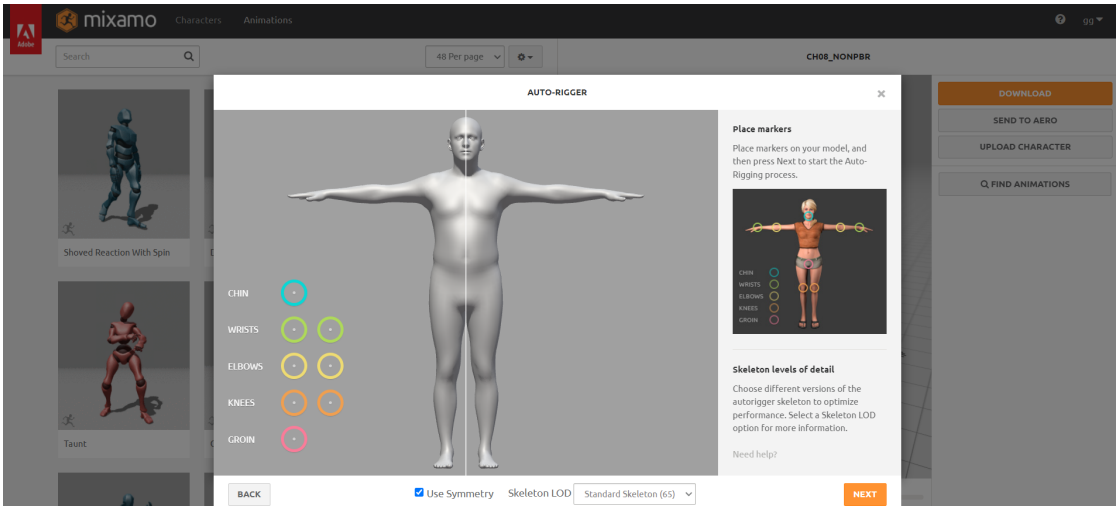


Figura 3.22: Interfaccia di Mixamo.

problemi, e monitorare la crescita e i progressi del codice tramite commit e versioni. Con l'uso di repository, gli sviluppatori possono lavorare in maniera organizzata su progetti, modificandoli e contribuendo a distanza tramite fork, clonazioni e pull request, che consentono la proposta di miglioramenti o correzioni senza modificare il progetto principale finché le modifiche non sono approvate. I repository possono essere pubblici o privati, adattandosi sia a progetti open source sia aziendali, rendendo GitHub una scelta versatile per iniziative di ogni tipo. Le funzionalità collaborative includono il tracciamento delle issue, ovvero problemi o richieste di miglioramenti che chiunque può aprire per segnalare bug o proporre idee. Gli sviluppatori possono commentare, discutere e risolvere queste issue direttamente sul sito, permettendo un lavoro più strutturato e coinvolgendo, nei progetti open source, anche gli utenti finali del software. Le pull request agevolano la revisione del codice, facilitando la valutazione delle modifiche da parte di altri membri del team prima della fusione nel codice principale. Questo sistema di controllo qualità integrato garantisce una coerenza nello sviluppo e assicura che solo le modifiche più accurate e pertinenti siano incorporate nel progetto. Un'altra funzione significativa è GitHub Actions, che permette l'integrazione e il deployment continui (CI/CD). Con Actions, gli sviluppatori possono creare flussi di lavoro automatizzati, come test di codice e rilascio di versioni, che si attivano in risposta a eventi specifici (come un push di commit o una pull request). Questi flussi possono essere personalizzati per includere test unitari, integrazioni e deployment automatico del codice, migliorando l'efficienza e riducendo il rischio di errori umani. GitHub favorisce la collaborazione globale e la condivisione di conoscenze, supportando una comunità open source vivace e in espansione. Organizzazioni e singoli sviluppatori possono condividere pubblicamente il proprio codice, rendendolo disponibile per l'apprendimento, la modifica e l'integrazione in altri progetti.

Capitolo 4

Implementazione del sistema

Nel capitolo corrente verranno illustrate le implementazioni pratiche dei metodi descritti in precedenza, evidenziando i processi seguiti e le modifiche apportate per raggiungere gli obiettivi prefissati.

4.1 Implementazione ricostruzione posa

SMPL-X offre un Add-on utilizzabile su blender che permettere di aggiungere nella scena un modello umano di genere maschile femminile o neutro, offrendo con un armatura per animare la mesh. Questo lavoro ha modificato alcuni script presenti nell Add-on di SMPL-X in modo da creare una pipeline che inizia scegliendo un'immagine a piacere di una persona in modo da ricostruirne l'avatar virtuale. Una volta scelta l'immmagine come l'immagine in figura 4.1 verrà automaticamente impostata come input di tutti gli script utilizzati nell'Add-on. Una volta scelto il modello che si preferisce da un elenco a discesa sarà possibile tramite un apposito pulsante inserirlo nella scena come mostrato in figura 4.2. L'esecuzione dell'intero processo di ricostruzione della posa è eseguibile da Python (Python Software Foundation [2023a]) eseguendo gli script direttamente da console o tramite l'uso di un framework come Visual Studio CodeCorporation [2024] ma per permettere una migliore esperienza ed ottimizzare il tempo di esecuzione è stato appositamente creato un file .bat per l'esecuzione automatica dei comandi necessari ricostuire la posa. Sfruttando la console python di blender viene eseguito il file .bat direttamente da blender ottenendo il risultato direttamente con una singola pressione di un bottone chiamato "pose reconstruction" come indicato in figura4.3, l'utente verrà ulteriormente avvisato con un messaggio che indica la corretta conclusione del processo come illustrato in figura 4.4. Un ulteriore personalizzazione come altezza e peso, in parametri di forma utilizzabili per il modello SMPL-X è stata inserita nell'Add-on per permettere all utente finale di personalizzare il proprio modello. Lo script è implementato come una classe, SMPLXMeasurementsToShape, che eredita da bpy.types.Operator, consentendo di eseguire l'operazione all'interno dell'interfaccia di Blender. Quando l'operatore viene eseguito, Blender verifica che l'oggetto attivo sia una mesh con un'armatura genitore, altrimenti l'operazione viene disabilitata. Questo controllo viene effettuato dalla funzione poll, che

Figura 4.1: Immagine utilizzata per la ricostruzione. Fonte: https://images.app.goo.gl/APmvAaRKzgwLi8Xh8

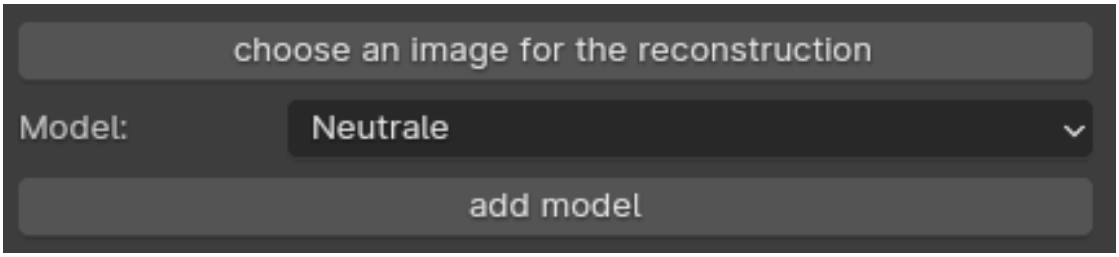


Figura 4.2: Pulsanti per la scelta dell'immagine e per l'aggunta di un modello.

rende disponibile l'operazione solo se vengono soddisfatti questi requisiti. La funzione execute avvia il processo. Inizialmente, lo script cerca i file JSON che contengono i regressori per calcolare i parametri betas a partire dalle misurazioni. Questi regressori sono suddivisi in tre categorie: femminile, maschile e neutro. Ciascuno di questi file contiene i coefficienti delle matrici necessarie per il calcolo dei parametri di forma in base all'altezza e al peso. Se i regressori non sono già stati caricati, lo script li carica dai file JSON

Figura 4.3: Pulsanti per l'esecuzione del modello e il carocamento della posa.

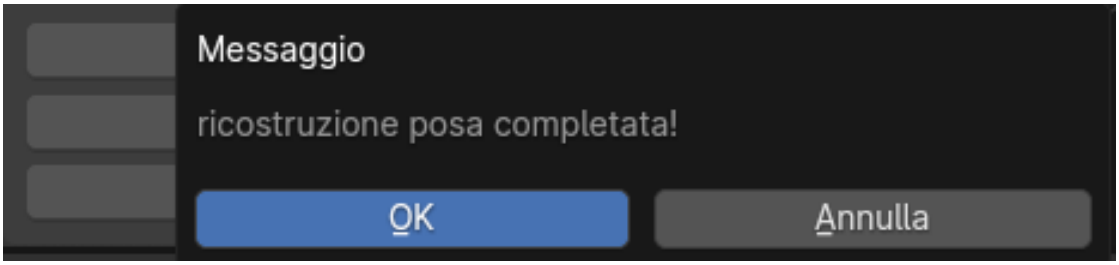


Figura 4.4: Messaggio relativo alla ricostruzione completata.

corrispondenti. Una volta selezionato il regressore appropriato, lo script procede a calcolare i valori dei parametri di forma. Questo calcolo si basa sulle misurazioni di altezza e peso fornite dall'utente tramite l'interfaccia di Blender come in figura4.5. In particolare, viene calcolata l'altezza in centimetri e il volume radice cubica del peso corporeo, poiché queste misurazioni sono utilizzate per stimare la variazione di forma. Utilizzando queste misurazioni, i parametri vengono calcolati moltiplicando la matrice di coefficenti per le misurazioni e aggiungendo il vettore che contiene i valori base per le deformazioni della forma. I valori calcolati vengono quindi applicati direttamente alle shape keys della mesh SMPL-X attiva in Blender. Ciascun shape key rappresenta una variazione specifica della forma del corpo, e lo script modifica questi valori per corrispondere ai parametri calcolati. Inoltre, lo script regola i limiti massimi e minimi degli shape keys in modo che possano rappresentare accuratamente il valore calcolato. Una volta applicati i parametri viene eseguita un'operazione per aggiornare la posizione delle giunture del modello SMPL-X, in modo che riflettano correttamente le nuove deformazioni della forma. Lo script di "Load pose" è progettato per caricare una posa del modello SMPL-X da un file .pkl e applicarla alla mesh attiva, consentendo la modifica dei parametri di forma e delle pose direttamente nell'ambiente di Blender. La sua implementazione utilizza le classi bpy.types.Operator e ImportHelper per facilitare il caricamento di file dall'utente e l'esecuzione del processo. la funzione è stata modificata inserire correttamente i parametri di tutte le ossa come anche il file .pkl prodotto in output è stato impostato in modo da trasportare le corrette informazioni relative alla posa e successivamente sono state fatte delle modifiche nello script per correggere le posizioni e le rotazioni. Il funzionamento dello script inizia con la verifica del tipo di oggetto attivo nella scena. Il controllo, realizzato nella funzione poll, consente di abilitare o disabilitare l'operazione solo se l'oggetto attivo è una mesh con un'armatura genitore o un'armatura stessa. Questo passaggio è necessario poiché la corretta applicazione delle pose richiede l'interazione tra la mesh e la sua armatura. Una volta che l'utente seleziona un file .pkl, contenente la posa da applicare, il contenuto del

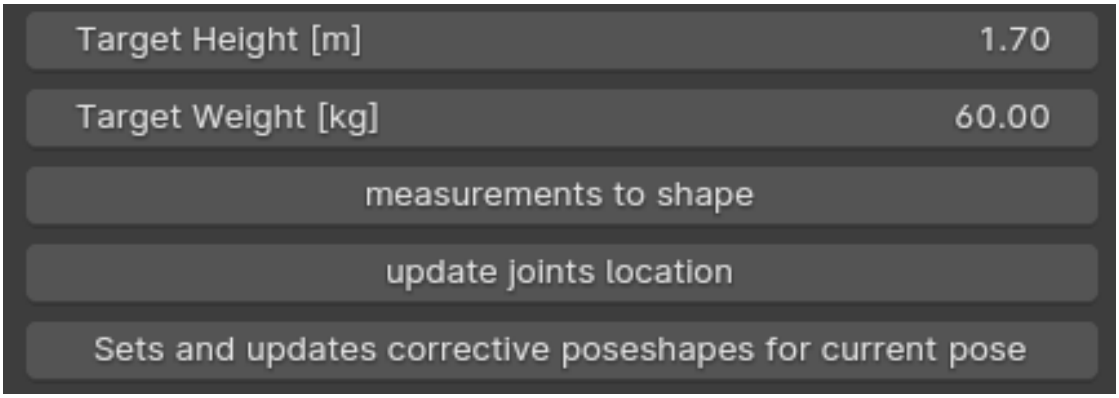
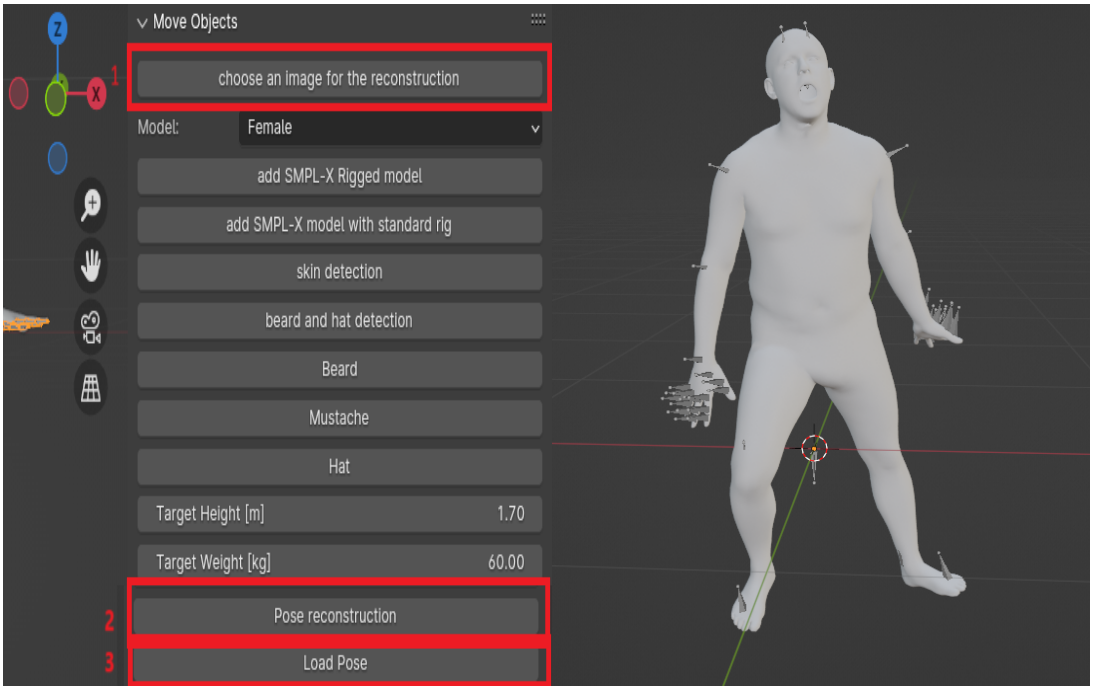


Figura 4.5: Interfaccia per la modifica di altezza e peso.

file viene caricato utilizzando la libreria pickle, con codifica latin1. I dati contenuti nel file includono parametri come la traslazione, l'orientamento globale, la posa del corpo e delle mani, oltre ai parametri di forma e alle espressioni facciali. Questi dati sono strutturati in array e rappresentano la base per l'ottimizzazione della posa all'interno della scena di Blender. Lo script modifica i parametri di forma della mesh attiva tramite gli shape keys. Ogni shape key rappresenta una variazione nella configurazione del modello, influenzando caratteristiche come la corporatura. Dopo l'applicazione dei parametri di forma, lo script aggiorna anche le posizioni delle giunture dell'armatura, assicurando che siano in accordo con la nuova forma del corpo. Per quanto riguarda l'applicazione della posa, i dati caricati vengono tradotti in trasformazioni di Rodrigues, un metodo matematico utilizzato per gestire rotazioni tridimensionali. Lo script applica queste trasformazioni a ciascuna articolazione del corpo, basandosi sui dati della posa estratti dal file .pkl reso comprensivo di tutte le informazioni relative alle ossa. Viene gestita anche la posa delle mani, combinando la posa caricata con una posa rilassata per ottenere un risultato più naturale. La gestione delle articolazioni, incluse quelle delle mani, viene eseguita tramite un ciclo for attraverso una lista di giunti predefiniti nel modello SMPL-X. Oltre alla gestione delle pose, lo script applica anche la traslazione del modello, spostando l'armatura nel mondo 3D di Blender secondo i valori contenuti nel file .pkl. Successivamente, vengono attivate le espressioni facciali attraverso gli shape keys associati alle diverse espressioni, come sorrisi o movimenti delle sopracciglia. Questo avviene regolando i valori delle chiavi di forma del viso in modo che le espressioni corrispondano ai dati presenti nel file. Infine, lo script attiva le corrective pose shapes, che sono progettati per correggere deformazioni indesiderate nelle pose estreme. Questo processo assicura che il modello mantenga un aspetto naturale anche quando assume pose articolari non convenzionali. Una volta completatao il processo si otterà un risultato come mostrato in figura 4.6 con una posa molto similare a quella nella fotografia utilizzata per la ricostruzione oltre ad espressioni facciali accurate.



(a) Pipeline di ricostruzione della posa. (b) Risultato del caricamento della posa.

Figura 4.6: processo completo di ricostuzione posa.

4.2 Implementazione ricostruzione vestiario

Durante i primi test sulla ricostruzione della posa e del vestiario, è emersa una problematica legata alla posa del modello su cui devono essere applicati i vestiti. La ricostruzione della posa effettuata da SMPLify-x utilizza OpenPose per ottenere le informazioni 2D dei giunti dall'immagine di input, mentre ClothWild si basa su Hand4Whole. OpenPose è progettato per rilevare le articolazioni chiave del corpo umano a partire da immagini o video, con particolare efficacia nella stima delle pose 2D, specialmente in scenari con più individui. Tuttavia, il suo utilizzo è limitato alla stima delle pose bidimensionali, senza occuparsi della ricostruzione 3D o del dettaglio delle mani e del viso. Hand4Whole, d'altro canto, è un modello specifico per la stima della mesh 3D del corpo umano, che include una ricostruzione dettagliata di corpo, mani e viso. Utilizza un'architettura Pose2Pose per combinare le informazioni delle articolazioni del corpo e delle mani, con l'obiettivo di ottenere una stima più precisa delle rotazioni articolari, in particolare per le mani, che sono complesse da ricostruire. Sebbene Hand4Whole sia orientato alla ricostruzione 3D, ClothWild sfrutta solo l'output 2D per la ricostruzione del modello posato. La principale differenza tra OpenPose e Hand4Whole risiede nel loro approccio: il primo è focalizzato sulla stima delle pose 2D ed è particolarmente efficace nel rilevamento in tempo reale di più individui, mentre il secondo è progettato per ricostruzioni 3D dettagliate, con un'enfasi particolare sulle mani e sulla connessione con il resto del corpo. Questa divergenza nei

metodi porta a risultati differenti a livello di posa, non macroscopici ma sufficientemente rilevanti da influire sulla successiva fase di ricostruzione del vestiario. Inizialmente è stato tentato un allineamento tra il modello posato ottenuto da SMPLify-x e la ricostruzione dei vestiti di ClothWild. Il primo approccio prevedeva lo spostamento di entrambe le mesh nell'origine, annullando le rotazioni. Tuttavia, è emerso che le due mesh presentavano rotazioni e scale intrinsecamente differenti come si osserva in figura 4.7 anche per pose semplici. Allineando manualmente gli oggetti su Blender, è risultato evidente che la differenza era legata alla ricostruzione della posa. Si è quindi optato per la ricerca di un algoritmo in grado di modificare e adattare una mesh per allinearla con un'altra. Un

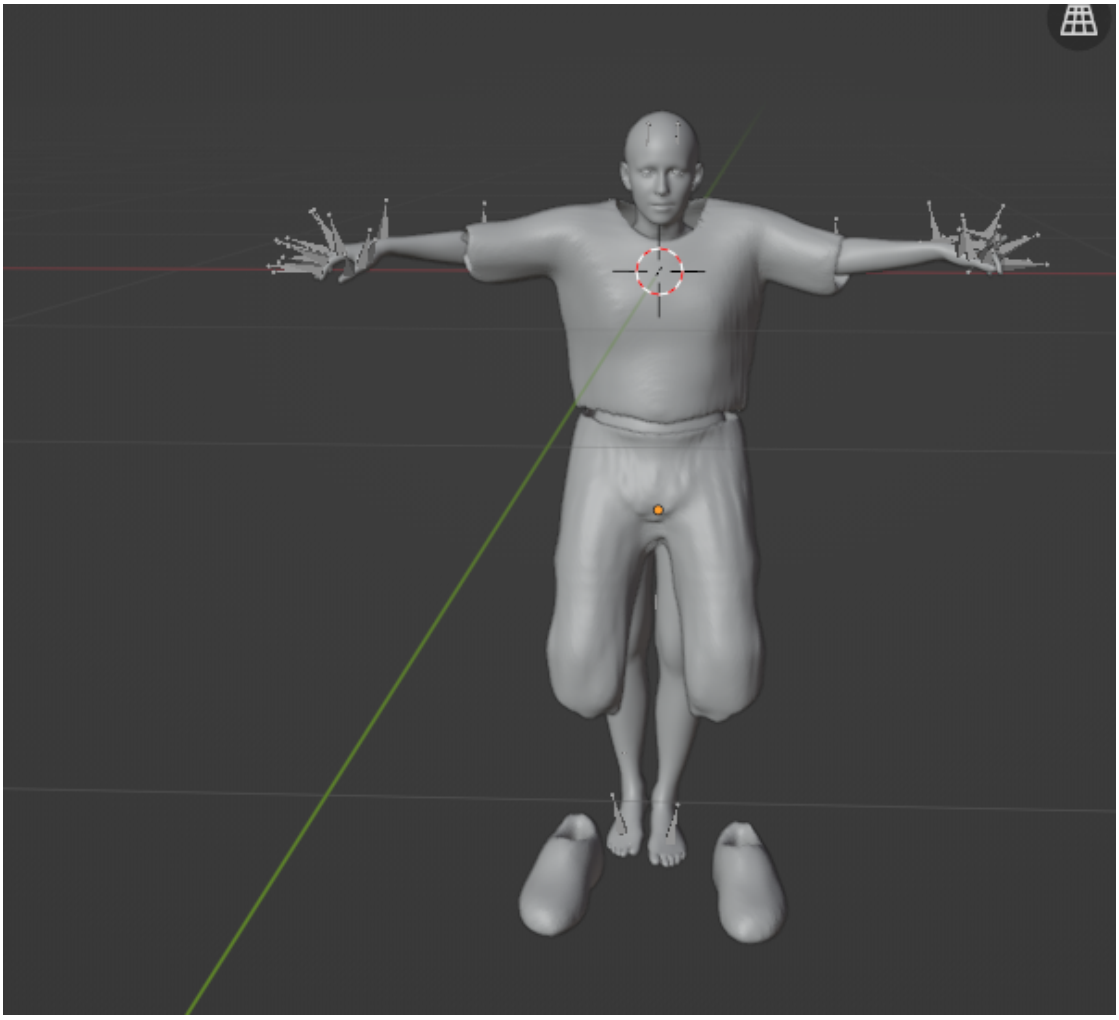


Figura 4.7: Ricostruzione posa e vestiti.

tentativo ulteriore è stato quello di replicare l'allineamento degli oggetti proposto da MeshLab (Cignoni et al. [2008]). L'allineamento degli oggetti in MeshLab è una procedura utile nel caso di mesh o nuvole di punti che necessitano di essere combinate o registrate

con precisione. Il processo si articola in diverse fasi, ognuna delle quali svolge un ruolo cruciale nell'ottenere un allineamento accurato. La prima fase consiste nel caricamento delle mesh da allineare. In questa fase, è importante assicurarsi di importate correttamente nella finestra principale del software. Una volta che i modelli sono caricati, è necessario scegliere quale delle due mesh fungerà da riferimento, ossia quella che resterà fissa durante tutto il processo. Questo passaggio si esegue tramite la selezione del layer corrispondente. Successivamente, si procede con un primo allineamento approssimativo. MeshLab offre uno strumento di registrazione manuale che consente di selezionare punti corrispondenti su entrambe le mesh. Questo primo allineamento, pur essendo grezzo, è essenziale poiché fornisce una base su cui l'algoritmo può lavorare nelle fasi successive. Una volta posizionati i punti corrispondenti, il software esegue una prima sovrapposizione delle mesh. Per affinare ulteriormente l'allineamento, viene applicato l'algoritmo ICP (Bai [2023]), noto come Iterative Closest Point. Questo metodo ottimizza la sovrapposizione delle due mesh, riducendo al minimo le distanze tra i punti più vicini. Affinché l'algoritmo funzioni correttamente, è necessario che l'allineamento iniziale non presenti errori grossolani; in caso contrario, l'ICP potrebbe non essere in grado di migliorare il risultato. Terminato l'allineamento, è possibile verificare la qualità del lavoro effettuato, sia visivamente, osservando la sovrapposizione delle mesh che risulta allineata nel miglior modo possibile ma risulta comunque non soddisfacente dato che sono presenti penetrazioni tra le mesh come si può vedere in figura 4.8. L'allineamento accurato di due mesh 3D rappresenta una sfida importante nella ricostruzione di oggetti tridimensionali e nell'analisi della posa corporea, specialmente in condizioni di visuali o risoluzioni diverse. La complessità del problema risiede nel dover assicurare la coerenza spaziale tra due set di dati che possono presentare deformazioni e variazioni dimensionali o di scala. A tale scopo, metodi come il feedback iterativo di allineamento e la fotometria si sono dimostrati strumenti efficaci. Il metodo PyMAF, ad esempio, utilizza un ciclo di feedback piramidale per migliorare progressivamente la qualità dell'allineamento delle pose. Attraverso un'iterazione continua tra diverse risoluzioni, PyMAF ottimizza l'allineamento della mesh da livelli più grezzi a livelli più dettagliati, permettendo un adattamento graduale che garantisce una convergenza più accurata delle pose. Questo approccio è particolarmente utile per la rappresentazione dettagliata della struttura corporea e delle espressioni, riducendo errori cumulativi e artefatti nella posa finale Zhang et al. [2021]. D'altra parte, il metodo di ottimizzazione fotometrica consente di allineare le mesh attraverso la minimizzazione della perdita fotometrica, definita come la somma delle differenze di intensità tra punti corrispondenti nelle immagini proiettate. Grazie all'uso di proiezioni da diverse prospettive, questo approccio consente di migliorare la coerenza visiva della mesh in condizioni multi-vista. La differenziazione tra proiezioni attraverso viste virtuali aiuta inoltre a stabilizzare il calcolo dei gradienti, essenziale per evitare disallineamenti durante l'ottimizzazione iterativa. Tale approccio è indicato per allineare dati 3D provenienti da scansioni o immagini da angolazioni differenti, superando la difficoltà di omogeneizzare dati acquisiti da fonti eterogenee Lin et al. [2019]. Uno dei metodi più recenti prevede l'uso di caratteristiche geometriche per l'allineamento. Tecniche come l'allineamento basato su SIFT (Lindeberg [2012]) o il SURF (Bay et al. [2006]), che sono state inizialmente sviluppate per immagini 2D, sono state estese al dominio 3D. L'idea è quella di rilevare punti di interesse che siano robusti rispetto a trasformazioni geometriche, rumore e variazioni di scala. Le corrispondenze

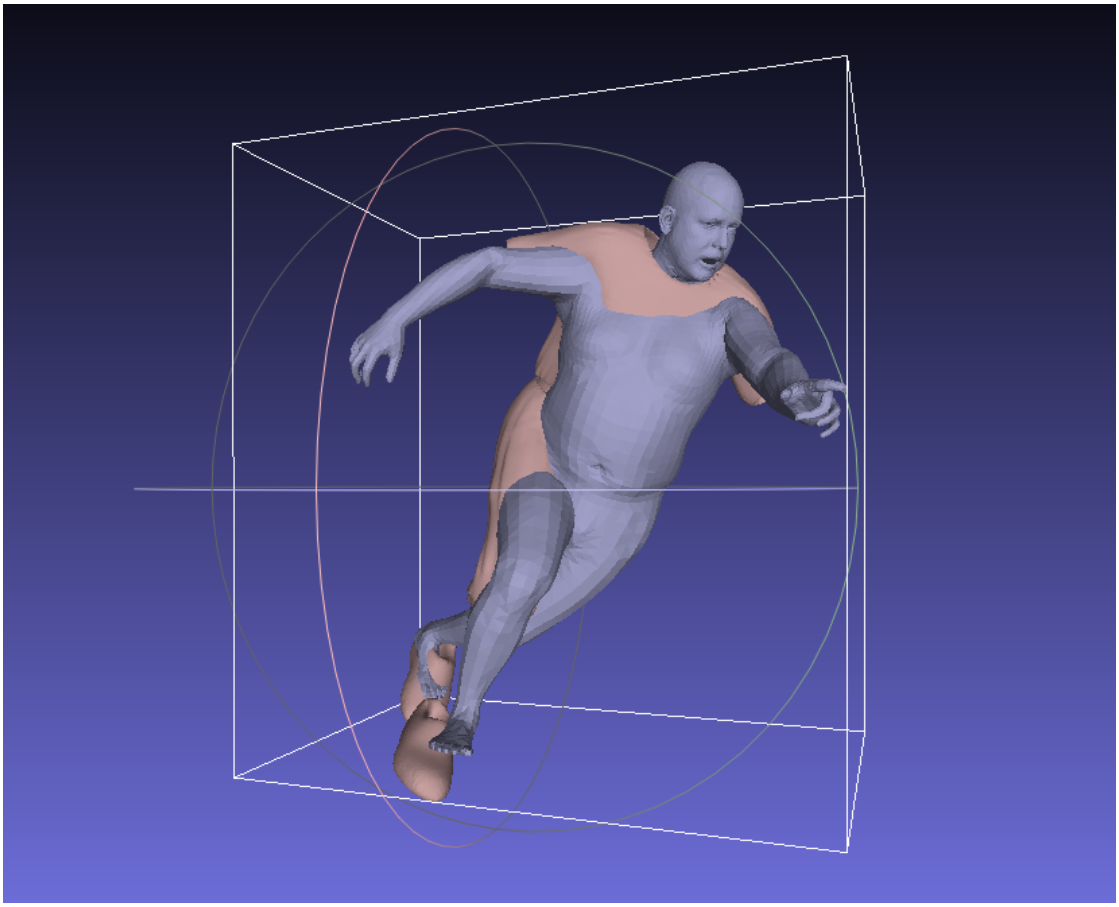


Figura 4.8: Allineamento tramite MeshLab.

tra questi punti vengono poi utilizzate per stimare la trasformazione iniziale tra le due mesh, migliorando la robustezza dell'allineamento rispetto alle differenze nella posizione e nell'orientamento iniziale. Un altro approccio interessante è l'FGR (Zhou et al. [2016]), che combina un allineamento basato su feature con una ricerca globale della migliore trasformazione rigida, riuscendo ad essere più veloce dell'ICP tradizionale e meno sensibile ai minimi locali. In sintesi, le metodologie analizzate nell'attuale stato dell'arte evidenziano che l'integrazione di strategie basate su feedback iterativo e ottimizzazione fotometrica rappresenta un approccio adattabile per ottenere un allineamento preciso nelle applicazioni di registrazione e ricostruzione tridimensionale. Tuttavia, si è deciso di esplorare ulteriori metodi per risolvere le problematiche riscontrate, ampliando il campo di analisi verso tecniche più semplici che possano offrire una maggiore efficacia rispetto alle limitazioni attuali. La problematica da risolvere rimane la differenza delle pose tra i due modelli, dopo una attenta valutazione si è scelto di semplificare il problema inserendo il modello SMPL-X in T-Pose come in figura 4.9 e di sviluppare uno script per produrre un file .json da utilizzare come input per ClothWild al posto di quello prodotto da Hand4Whole ma con lo stesso formato inserendo le informazioni della posa desiderata. Utilizzando

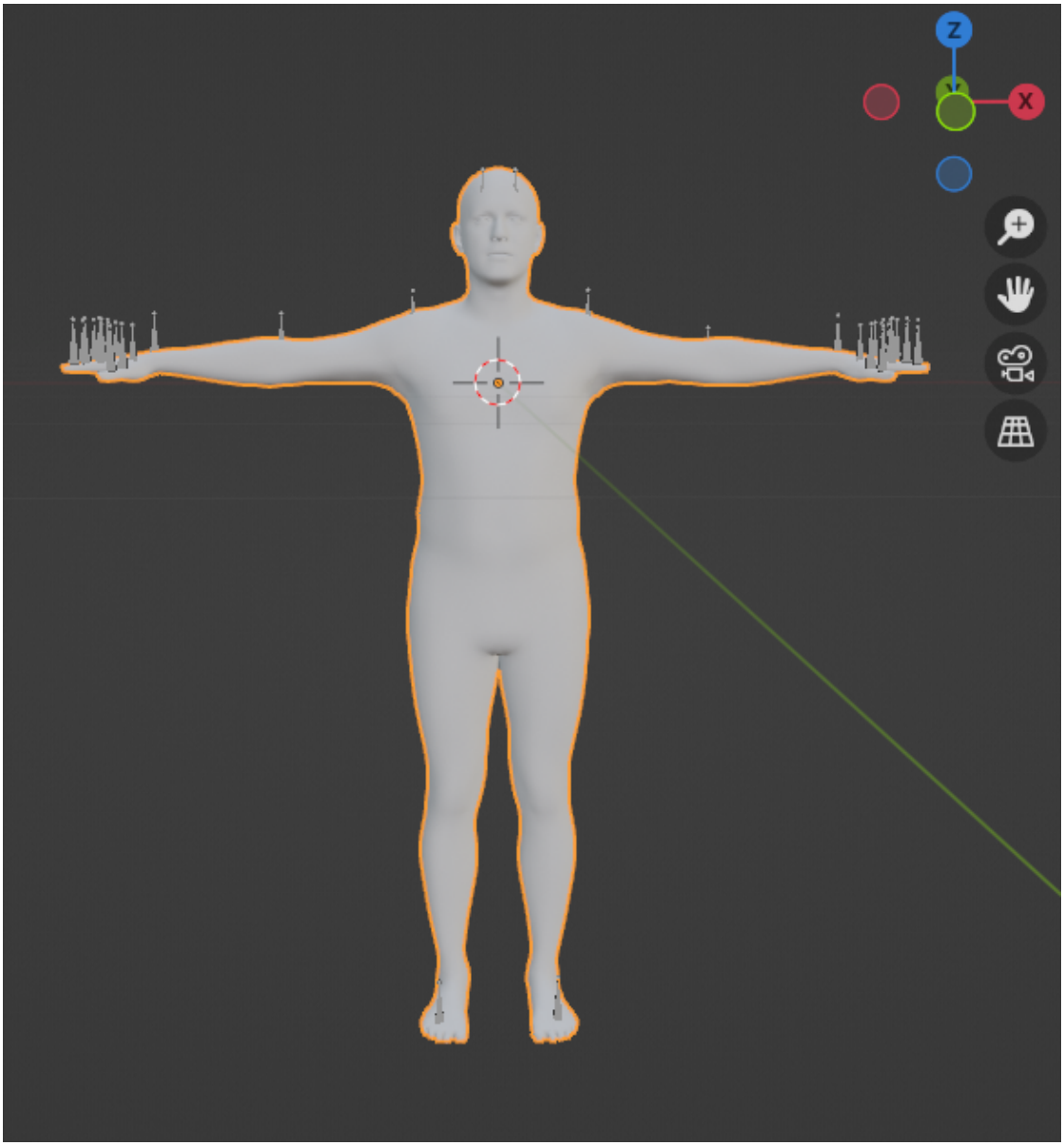


Figura 4.9: Modello SMPL-X in T-pose.

questo approccio il funzionamento della ricostruzione del vestiario varia leggermente, la classificazione delle tipologie di capi riconosciuti nell immagine rimane tale ma non viene utilizzato l'output di Hand4whole sostituendolo con lo script prodotto, in modo che la combinazione di Clothet e SMPLicit producano il vestiario classificato direttamente in T-pose come possiamo osservare in figura 4.10. Lo script sviluppato crea un file JSON contenente parametri specifici per una "T pose" utilizzando il modello SMPL. I parametri includono pose, forma e traslazione del corpo, nonché impostazioni per la fotocamera.

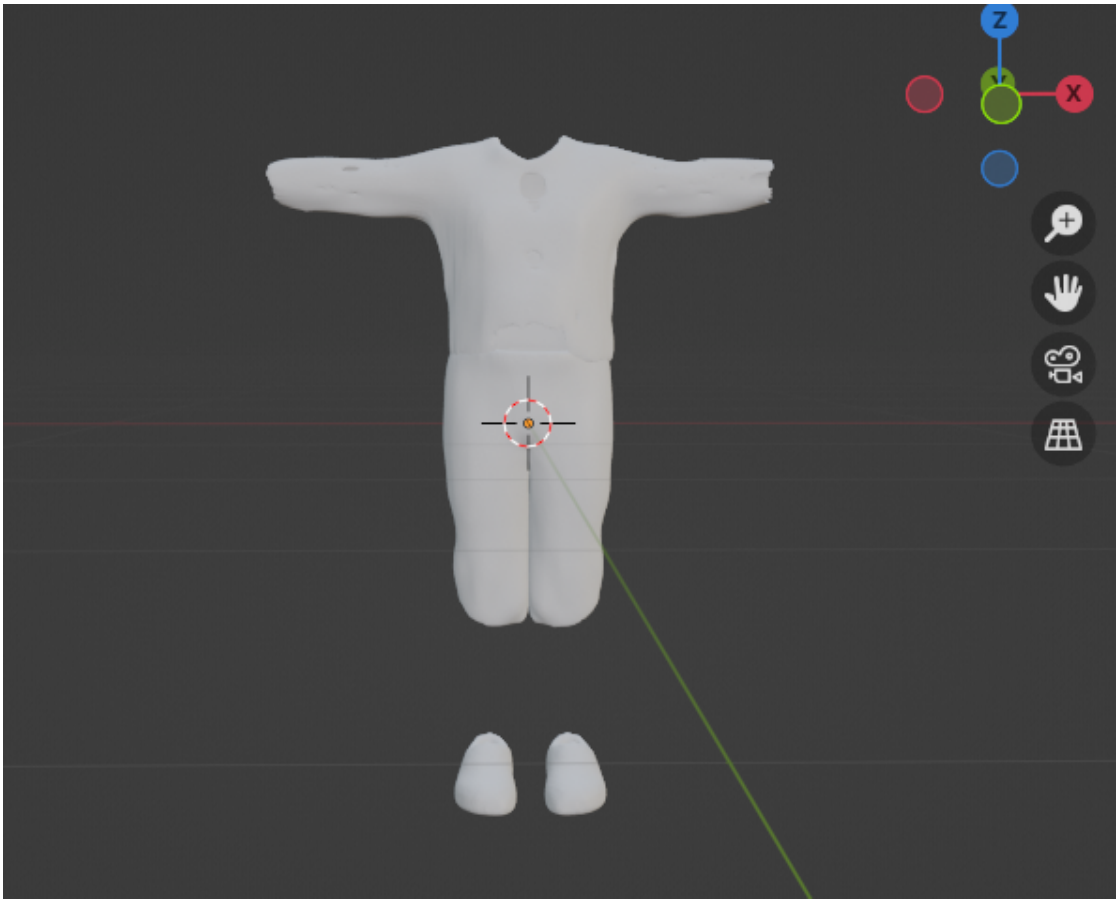


Figura 4.10: Vestiario posizionato in T-pose.

Per quanto riguarda la posa, viene inizializzata con un array di 72 zeri, rappresentante le rotazioni dei vari segmenti del corpo. Specifiche correzioni vengono applicate per ottenere una posa in cui le braccia sono orizzontali e le gambe dritte e orientate in avanti. Anche le rotazioni di ginocchia, caviglie e piedi sono impostate per ottenere una posizione naturale. La forma del corpo è descritta da un array di 10 zeri, mentre la traslazione del corpo è fissata a una posizione predefinita nello spazio in modo che sia nell'origine. I parametri della fotocamera includono il punto focale e il centro principale. Tutti questi dati vengono poi salvati in un file JSON, utilizzato per la posa del vestiario. Dopo l'acquisizione del file JSON contenente le informazioni relative alla posa, è stata apportata una modifica a ClothWild. Nella configurazione originale, l'output dell'esecuzione produceva una mesh unica che includeva sia il modello del corpo sia il vestiario. Il progetto è stato rielaborato per generare esclusivamente la sola mesh del vestiario, escludendo il modello del corpo. Questa scelta è stata motivata dall'intenzione di utilizzare il modello SMPL-X, che garantisce una maggiore accuratezza nella rappresentazione del corpo umano. L'esecuzione di ClothWild è stata integrata nell'Add-on di Blender precedentemente citato, permettendo

la ricostruzione del vestiario con un singolo pulsante. È stato creato un file .bat che contiene i comandi per eseguire l'intera pipeline di ricostruzione, partendo dal calcolo della posa tramite Hand4Whole fino alla generazione del file .obj contenente il vestiario. Questo file .bat viene eseguito direttamente dal pulsante "cloth reconstruction", come mostrato in figura 4.11, e al termine dell'esecuzione compare un messaggio di avviso per informare l'utente (figura 4.12). Una volta importato il file .obj relativo al vestiario ricostruito

Figura 4.11: funzione di ricostruzione vestiario.

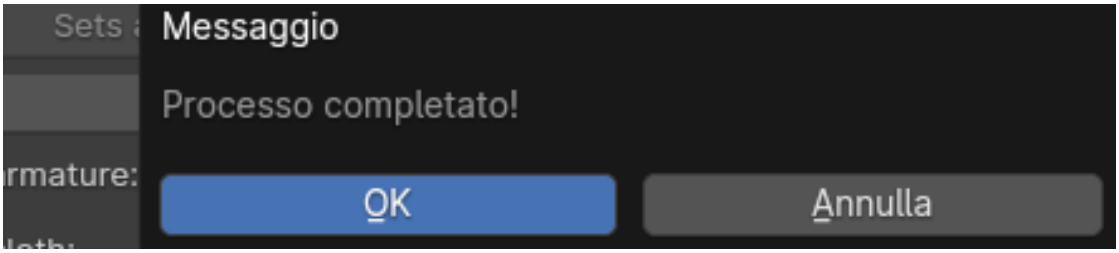


Figura 4.12: Messaggio di ricostruzione completata.

dall'immagine, è necessario posizionarlo correttamente sulla mesh del modello per evitare che questa operazione debba essere effettuata manualmente. A tale scopo, è stata implementata una funzione denominata "adjust cloths", come mostrato in figura 4.13 , che automatizza il processo di allineamento. È prevista la possibilità che la ricostruzione del vestiario avvenga con successo, ma che il risultato non risulti soddisfacente per l'utente, il quale potrebbe desiderare di sostituire il vestiario con modelli differenti. A tal proposito, sono stati utilizzati diversi modelli disponibili gratuitamente online su Turbosquid (tur), Free3D (fre) e Sketchfab (ske [2024]) che sono stati opportunamente modificati per essere compatibili con l'Add-on sviluppato. Per offrire all'utente un'ampia possibilità di personalizzazione, è stata selezionata una serie di modelli, uno per ciascuna classe presente in ClothWild. Ciò consente di adattare e personalizzare l'avatar in base alle preferenze individuali. Come illustrato in figura 4.14, un modello di t-shirt e un pantalone sono stati utilizzati per sostituire il vestiario ricostruito, dimostrando la possibilità di personalizzazione dell'avatar tramite l'Add-on. L'utilizzo dei modelli predefiniti è reso semplice grazie allo sviluppo di specifiche funzioni all'interno dell'Add-on, le quali permettono l'aggiunta del modello scelto direttamente sull'avatar. Come mostrato in figura 4.15, ogni pulsante è associato al nome del capo di abbigliamento corrispondente, facilitando la selezione e l'applicazione del vestiario personalizzato. L'allineamento del vestiario con la mesh e l'inserimento di una libreria di vestiti da poter utilizzare non sono le uniche funzionalità implementate. Poiché l'obiettivo del progetto è creare un avatar animabile, è necessario che il vestiario ricostruito segua le animazioni e le pose del modello. Questa operazione

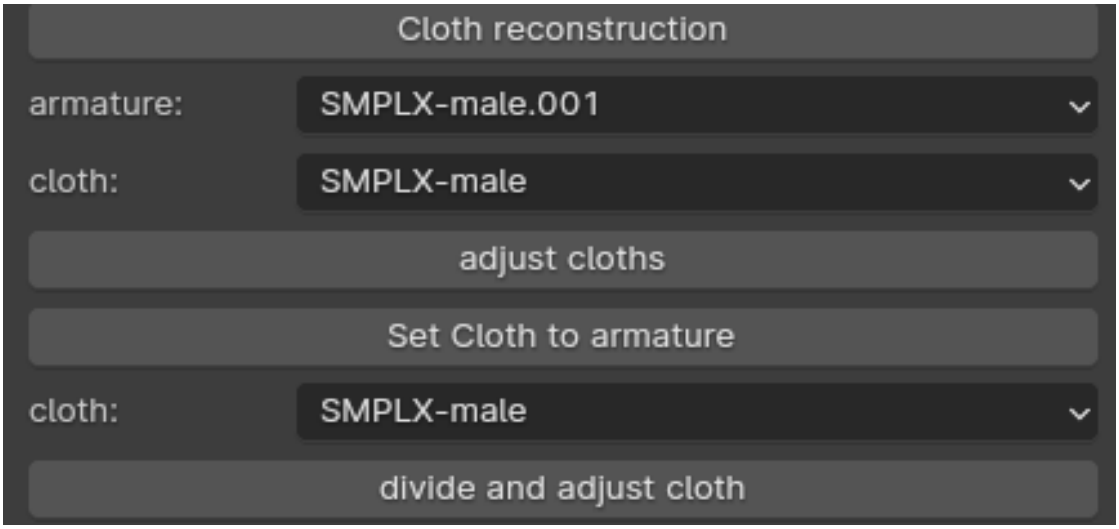


Figura 4.13: Interfaccia per la ricostruzione del vestiario.

viene realizzata tramite l'uso dei vertex group, attraverso la funzione "set cloth to armature", come illustrato in figura 4.13. I vertex group in Blender rappresentano un insieme di vertici selezionati e raggruppati per essere associati a determinate operazioni o modificatori, come l'assegnazione di pesi, materiali, o, più comunemente, il rigging. Ogni gruppo di vertici può essere creato manualmente dall'utente e personalizzato a seconda delle esigenze del progetto, o generato automaticamente quando si applica un'armatura denominata rig, a un modello. Il funzionamento di base dei vertex group è semplice, ogni gruppo contiene una selezione di vertici che viene etichettata con un nome specifico, permettendo a Blender di identificare questi vertici per applicare modificatori o trasformazioni. All'interno di un modello, un vertice può appartenere a più gruppi contemporaneamente, e ogni gruppo può avere un'influenza differente sui vertici, a seconda della funzione specifica per cui è utilizzato. Per assegnare i vertici a un gruppo, Blender consente l'uso del weight painting o la selezione diretta dei vertici in modalità di modifica. In weight painting, l'utente può dipingere manualmente il peso (influenzato da valori che vanno da 0 a 1) assegnato a ogni vertice rispetto a un gruppo, definendo quanto quel gruppo influenzerà la deformazione del modello durante il movimento. Il peso massimo (1) indica che un vertice sarà totalmente influenzato da quel gruppo, mentre un peso minore riduce proporzionalmente tale influenza. Nel contesto del rigging di un personaggio, i vertex group diventano particolarmente importanti quando si tratta di far muovere un vestito insieme all'armatura del modello riggato. Il processo di rigging di un modello implica l'associazione di un'armatura, composta da ossa, con il corpo del personaggio, permettendo al modello di deformarsi in modo realistico quando le ossa vengono mosse o ruotate. Questo processo può essere esteso anche al vestiario, garantendo che gli indumenti si muovano correttamente seguendo le deformazioni del corpo. L'associazione tra l'armatura e il vestito viene effettuata utilizzando i vertex group. Quando un vestito viene "attaccato" all'armatura, i vertici del

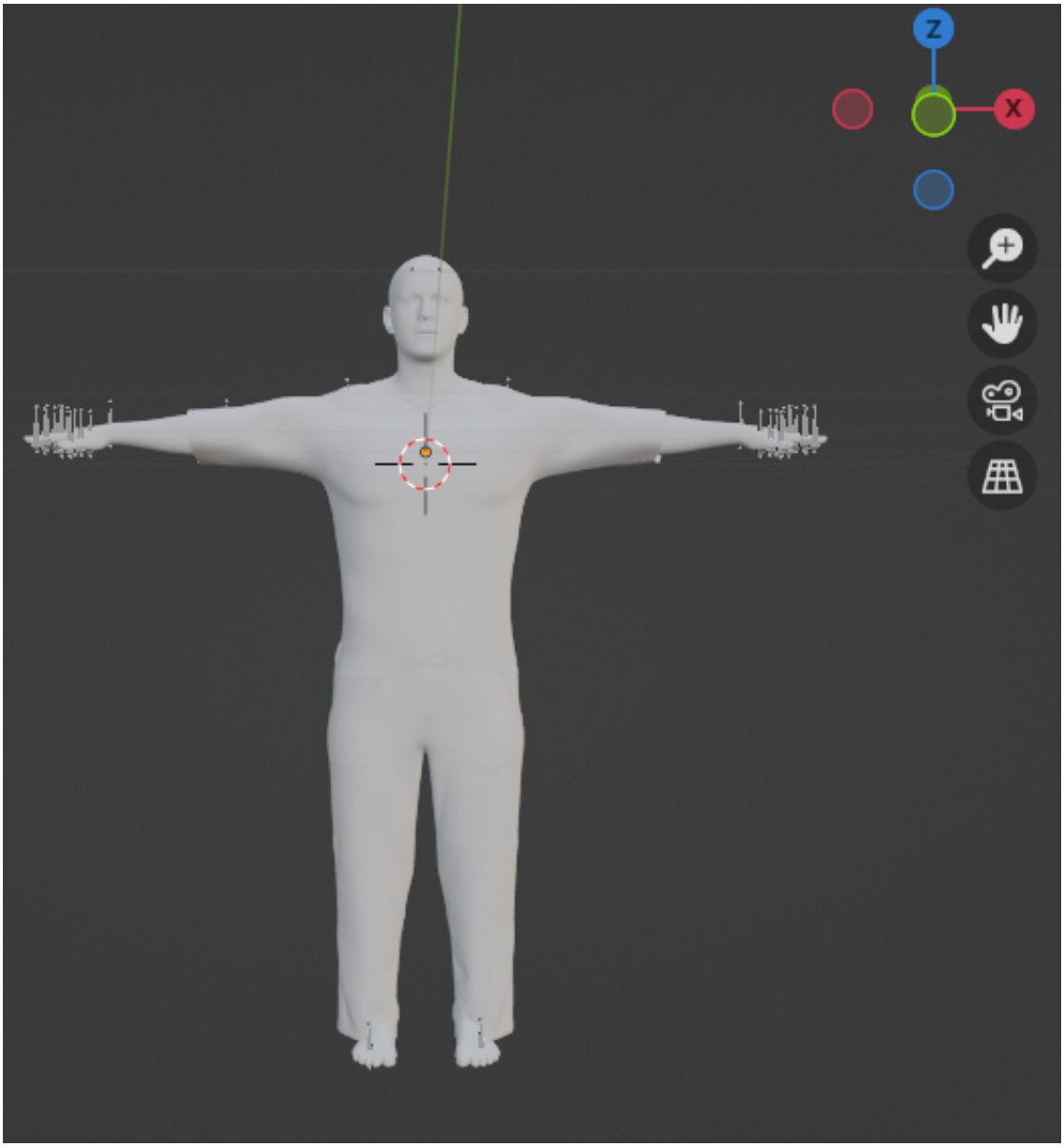


Figura 4.14: Esempio di modelli di t-shirt e pantalone aggiunti al modello.

vestito devono essere assegnati ai vertex group corrispondenti alle ossa del corpo sottostante. In questo modo, quando l'armatura si muove, il vestito seguirà il movimento in modo fluido, deformandosi in base ai pesi definiti nei vertex group come rappresentato in figura 4.16. Ad esempio, se si desidera che una manica di un vestito si muova insieme al braccio di un personaggio, i vertici che compongono la manica devono essere assegnati ai gruppi di vertici associati all'osso del braccio del rig del personaggio. Utilizzando il weight

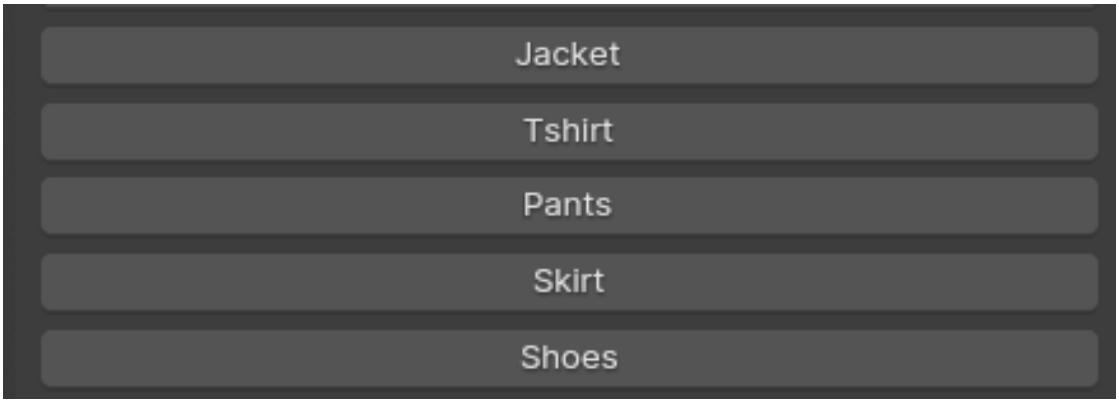


Figura 4.15: Funzioni per l'aggiunta di modelli di vestiti.

painting, è possibile garantire che i vertici vicino alla spalla abbiano un'influenza maggiore dall'osso della spalla e una minore dai gruppi di vertici associati al braccio inferiore, permettendo una deformazione realistica. Un ulteriore vantaggio dei vertex group in Blender è la possibilità di utilizzare lo skinning automatico per creare gruppi di vertici basati sull'influenza delle ossa dell'armatura. Questo processo, noto come "automatic weights", consente a Blender di generare automaticamente gruppi di vertici per il corpo del modello e per eventuali indumenti, basandosi sulla prossimità dei vertici alle ossa dell'armatura. Tuttavia, anche se questo metodo è efficace nella maggior parte dei casi, può richiedere aggiustamenti manuali tramite il weight painting per ottenere un risultato perfettamente realistico. La ricostruzione effettuata da ClothWild produce come risultato una singola mesh che rappresenta l'intero vestiario come l'esempio in figura 4.17. Per migliorare l'efficacia dell'applicativo, è stata sviluppata una funzione denominata "adjust and divide cloth", che suddivide la mesh dei vestiti in base alla geometria. Il risultato finale consiste in una mesh separata per ogni capo d'abbigliamento, ad eccezione delle scarpe, che saranno divise in due mesh distinte. È stata sviluppata una strategia che prevede, dopo la suddivisione della mesh, l'eliminazione di eventuali oggetti scollegati caratterizzati da un numero ridotto di vertici. Questo procedimento consente di rimuovere le parti di mesh scollegate e non significative, ottimizzando così lo spazio occupato dalla mesh complessiva. Un ulteriore controllo viene eseguito per eliminare eventuali errori, rimuovendo le mesh che contengono un numero molto basso di vertici e che non apportano alcun contributo significativo al vestiario, ottenendo così un risultato più pulito e accurato come mostratro in figura 4.18.

4.3 Implementazione ricostruzione dei capelli

È stata sviluppata una pipeline automatizzata che sfrutta la console Python all'interno di Blender per eseguire il codice di HairStep direttamente. Questa soluzione consente di ricostruire i capelli del modello scelto dall'utente mediante un singolo pulsante, come mostrato in figura 4.19 presente nell'add-on. Una volta completata l'esecuzione, il sistema

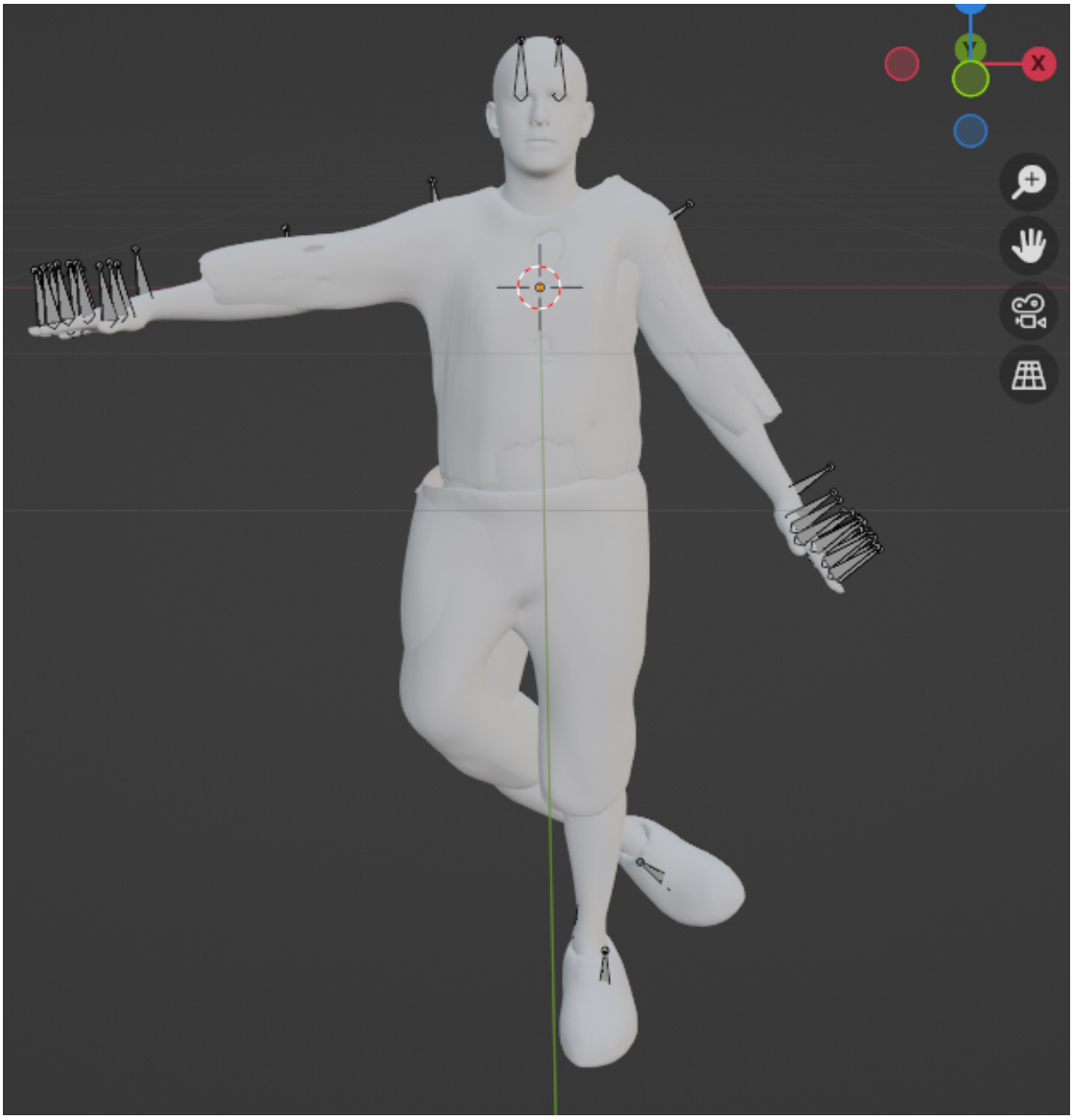


Figura 4.16: Esempio di uso di vertex group per modificare un vestito insieme al movimento dell'armatura.

visualizza un messaggio di feedback, visibile in figura 4.20, che informa l'utente sullo stato del processo. L'utente ha inoltre la possibilità di importare i capelli generati nella scena di Blender sotto forma di file .ply. Successivamente, una funzione specifica, rappresentata in figura 4.21, consente di posizionare accuratamente i capelli sul modello e di creare una parentela tra i capelli e l'oggetto. Questo passaggio permette di legare i capelli al movimento del modello, garantendo che si muovano in sincronia con esso durante la manipolazione. Questa metodologia permette di ricostruire capelli con strutture complesse come quella in

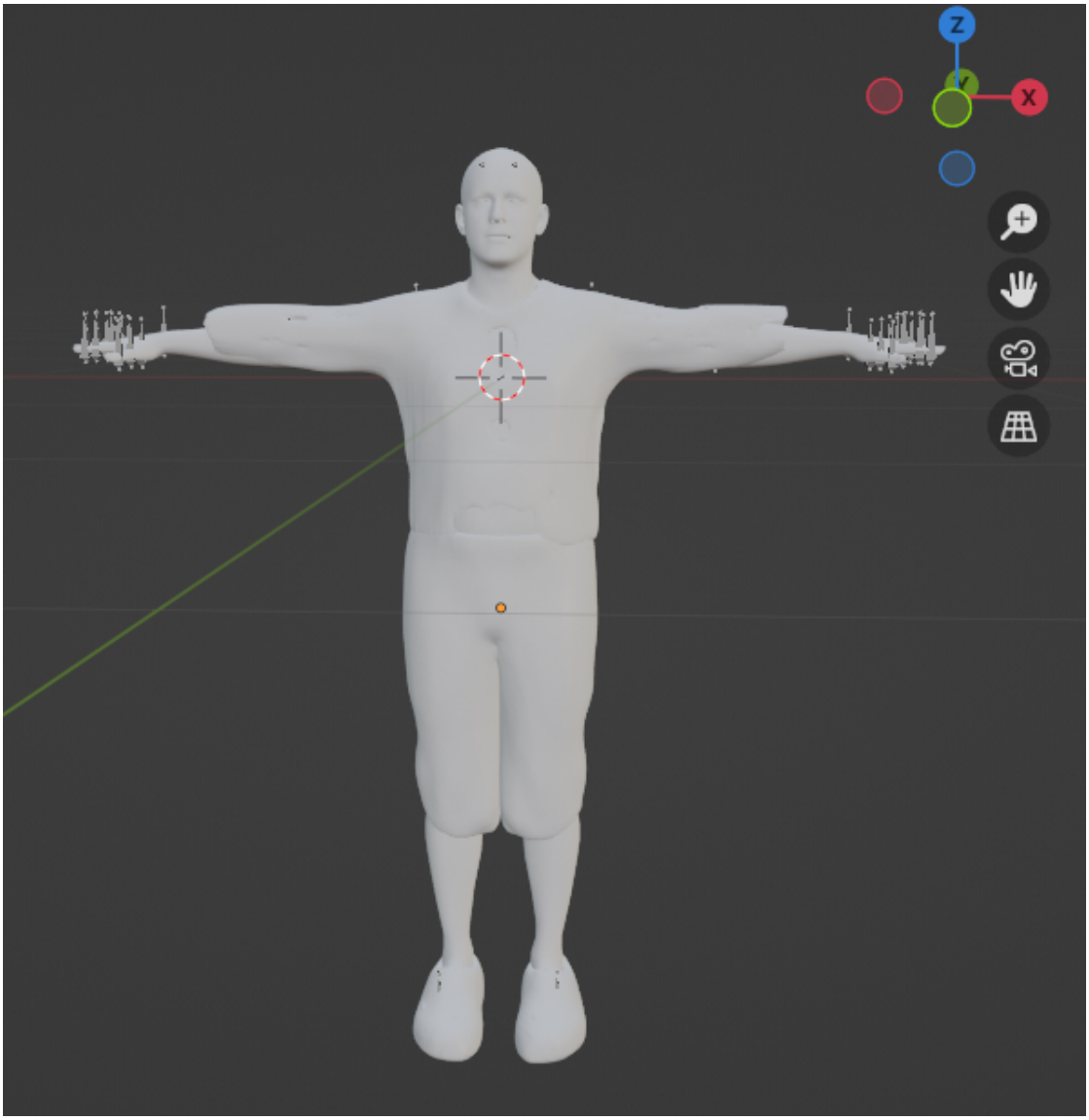


Figura 4.17: Vestiario in una mesh unica.

figura 4.22 L'implementazione automatizzata riducendo al minimo le operazioni manuali e ottimizzando il flusso di lavoro per la ricostruzione di capelli realistici 3D da una singola immagine ottenendo risultati realistici come mostato in figura 4.23 .

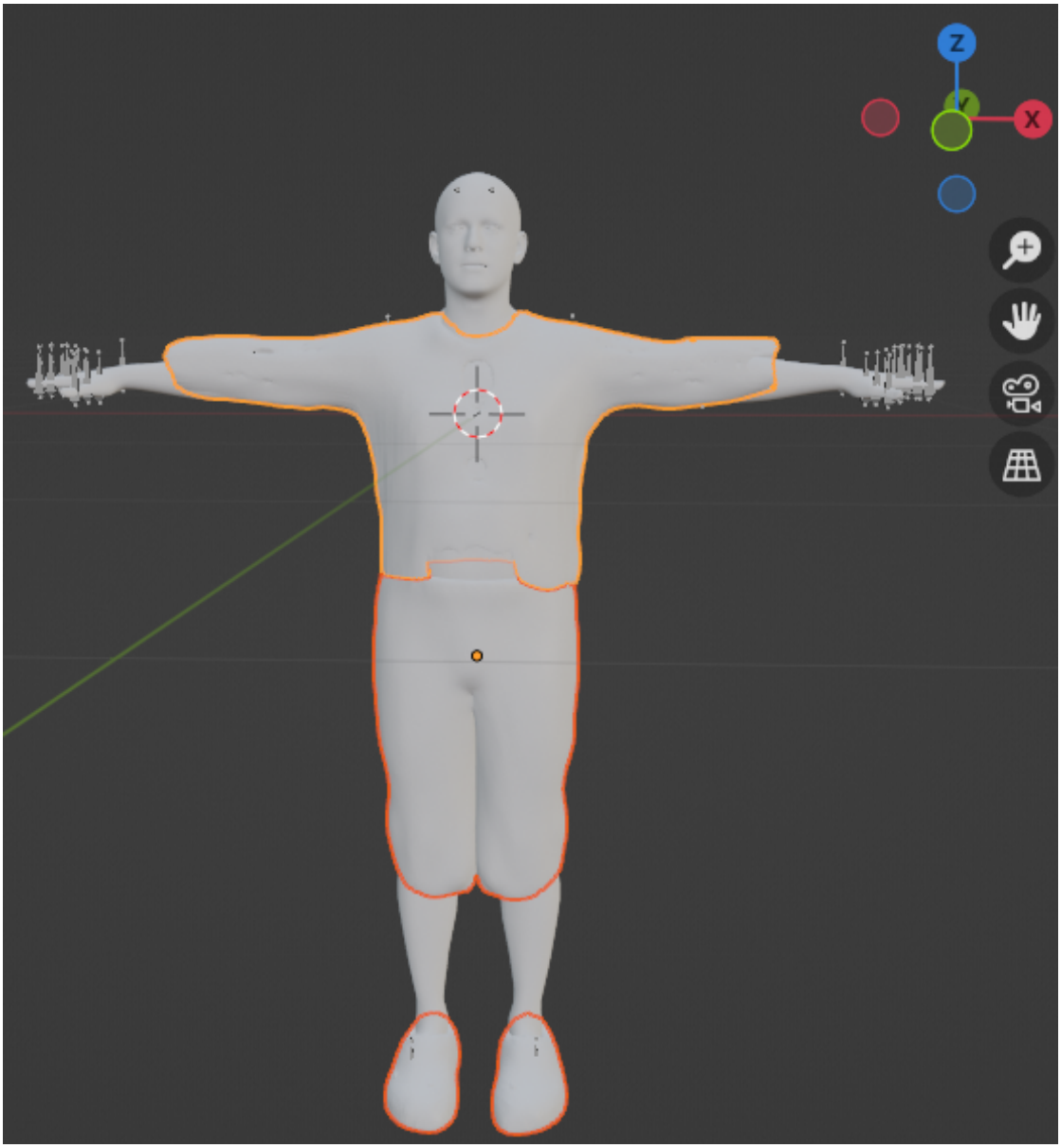


Figura 4.18: Vestiario diviso in mesh distinte per ogni capo di abbigliamento.

Figura 4.19: Bottone per eseguire la pipeline di ricostruzione dei capelli.

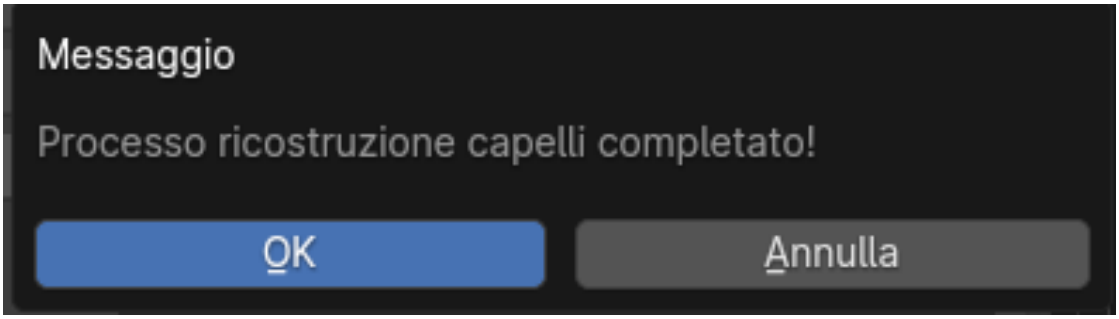


Figura 4.20: Messaggio che indica la conclusione della ricostruzione dei capelli.

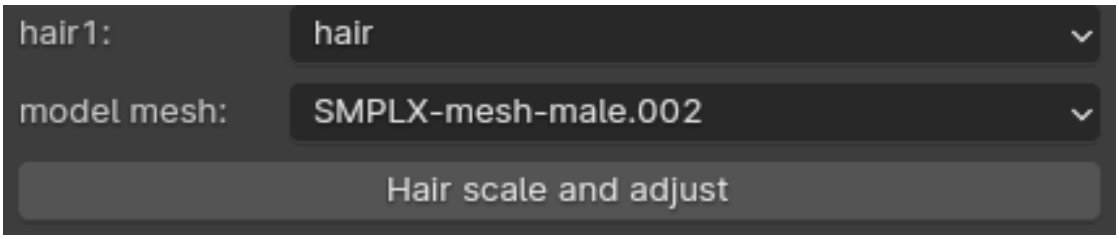
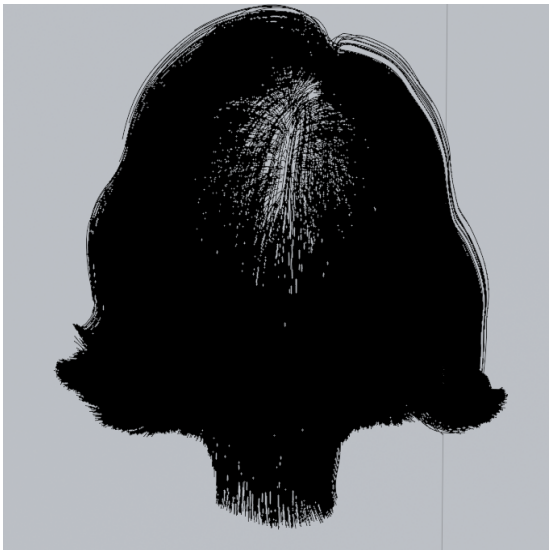


Figura 4.21: Messaggio che indica la conclusione della ricostruzione dei capelli.



(a) Ricostruzione capelli vista frontale.

(b) Ricostruzione capelli vista laterale.

Figura 4.23: Esempio di ricostuzione.

Figura 4.22: Immagine utilizzata per la ricostruzione. Fonte : https://github.com/GAP-LAB-CUHK-SZ/HairStep/tree/main/results/real_imgs/img

4.4 Implementazione riconoscimento della pelle

Lo script sviluppato nell add-on di blender permette automaticamente di impostare il risultato sulla mesh tramite l'utilizzo di un bottone nell add-on come in figura 4.24

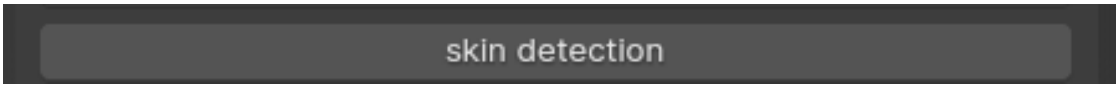


Figura 4.24: Interaccia per il riconoscimento della pelle.

Il risultato dell'esecuzione sarà la mesh del modello con l'inserimento del colore riconosciuto dall'immagine come riportato in figura 4.25

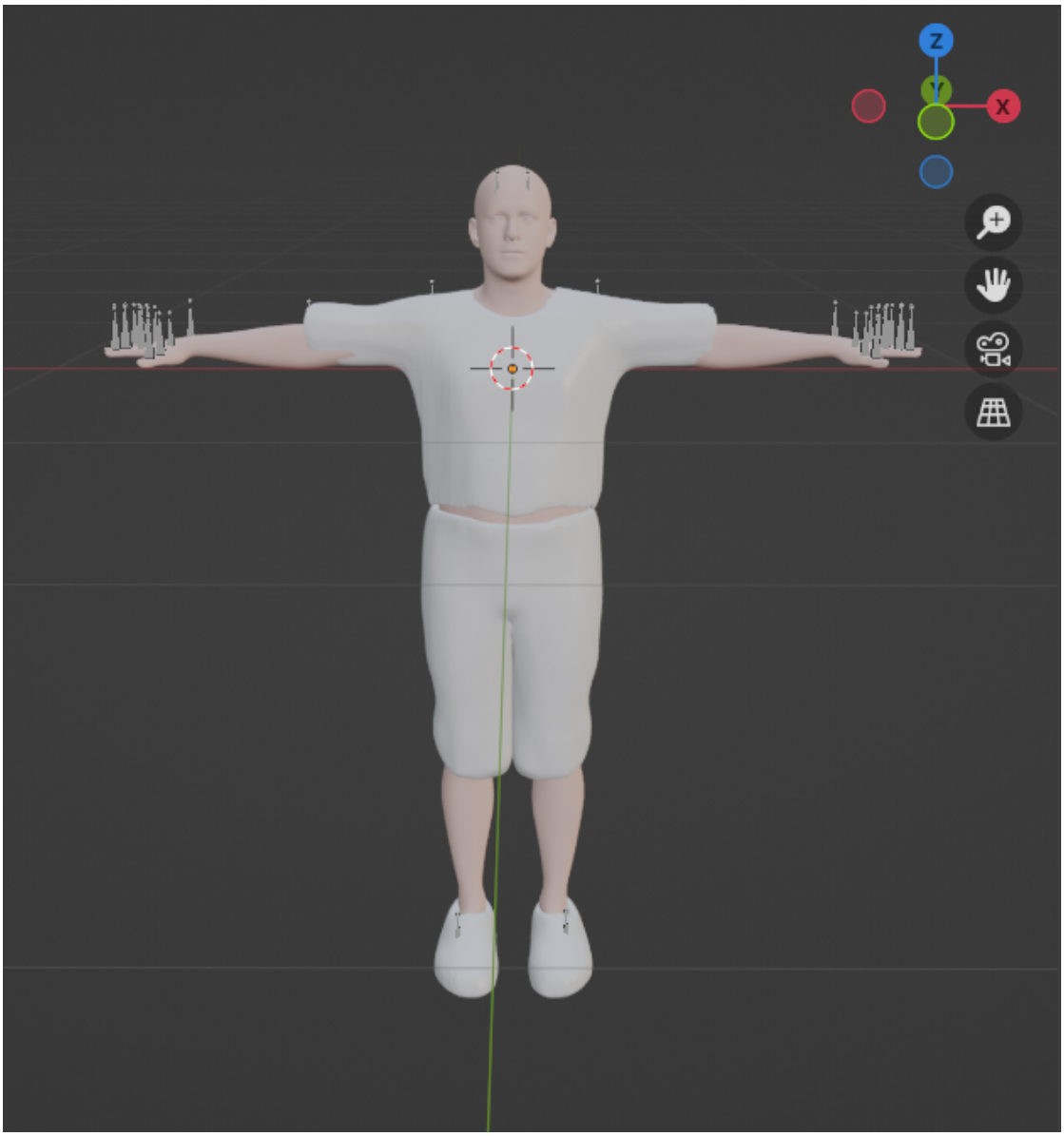


Figura 4.25: Mesh con colore applicato.

4.5 Implementazione riconoscimento della barba

Il progetto originale è stato modificato per mantenere unicamente la classificazione dell'immagine di input, rimuovendo ogni integrazione con la libreria OpenCV (Bradski [2000]), poiché non risulta necessario sovrapporre il risultato della classificazione sull'immagine stessa. La pipeline ora genera un file di output che riporta il risultato dell'inferenza. La classificazione di barba e cappello è stata integrata nell'add-on di Blender, dove l'inferenza può essere eseguita tramite un pulsante dedicato, come illustrato in figura 4.26. Al completamento dell'esecuzione, un messaggio con il risultato della classificazione viene mostrato all'utente, come in figura 4.27.

Figura 4.26: Interfaccia per l'esecuzione della classidficazione di barba e cappello.

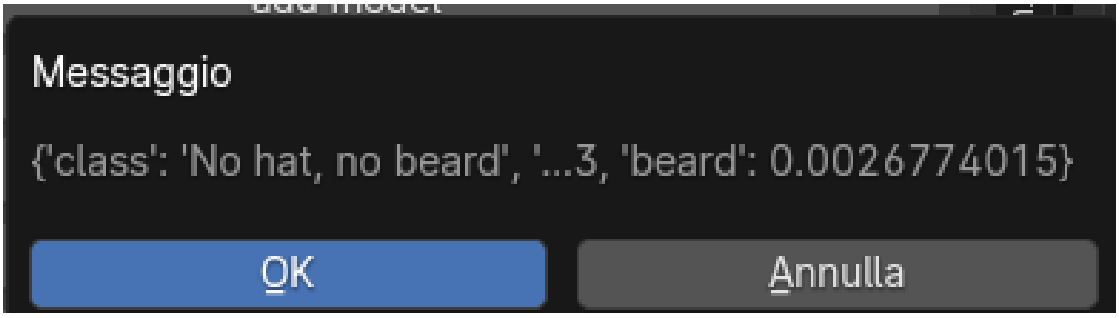


Figura 4.27: Messaggio contenente il risultato della classificazione.

Blender offre quindi un'interfaccia per personalizzare l'avatar con barba, baffi o cappelli, tramite una libreria predisposta, con ogni elemento selezionabile tramite un apposito pulsante, come si vede in figura 4.28.

Figura 4.28: Interfaccia per l'aggiunta dei modelli.

Si prevede la possibilità che la classificazione fallisca, in questo caso è implementato un sistema per analizzare un'immagine e determinare la presenza di barba o baffi basandosi

su una serie di punti chiave predefiniti per ogni regione del viso. L'esecuzione restituisce una struttura di dati che identifica coordinate specifiche per i punti di riferimento della mascella e dei baffi, simulando così il rilevamento automatico di landmark. Una volta ottenuti i landmark, il codice accede all'immagine in input, verificandone l'esistenza prima di procedere. Caricata l'immagine tramite la libreria Pillow (Clark [2015]) e convertita in scala di grigi, si esegue l'analisi della densità media di colore nelle aree di interesse, che restituisce la densità dei pixel per ogni coordinata dei baffi e della mascella.

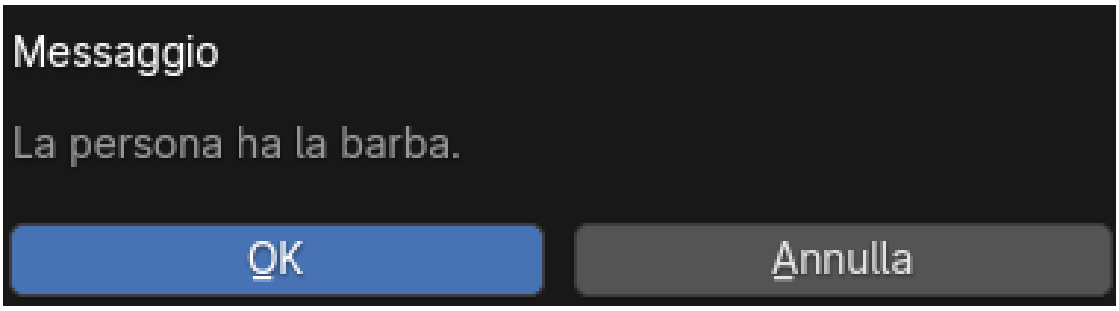


Figura 4.29: Messaggio in caso di classificazione alternativa.

La funzione calcola quindi la densità media delle aree rilevate e confronta i valori: se la densità dei baffi è significativamente inferiore rispetto alla mascella, il sistema conclude che ci sono baffi; al contrario, se la densità della mascella è inferiore, si indica la presenza di barba. In assenza di differenze significative, non si rilevano né barba né baffi. Infine si visualizza il risultato in Blender, mostrando un messaggio per comunicare l'esito della classificazione direttamente all'utente come in figura 4.29 in modo che anche in caso di fallimento del classificatore avvenga un controllo sulla presenza di barba e baffi.

Capitolo 5

Risultati

I risultati ottenuti sono valutati rispetto ad ogni parte della pipeline proposta, ognuna delle quali rispecchia l'output della ricerca di riferimento, in maniera coerente con le modifiche effettuate nei relativi progetti. Il risultato finale viene riportato nell add-on sviluppato in blender in cui è possibile eseguire le ricostruzioni direttamente dall'interfaccia riportata in figura 5.1. Il livello di dettaglio di SMPL-X permette una cattura espressiva che si

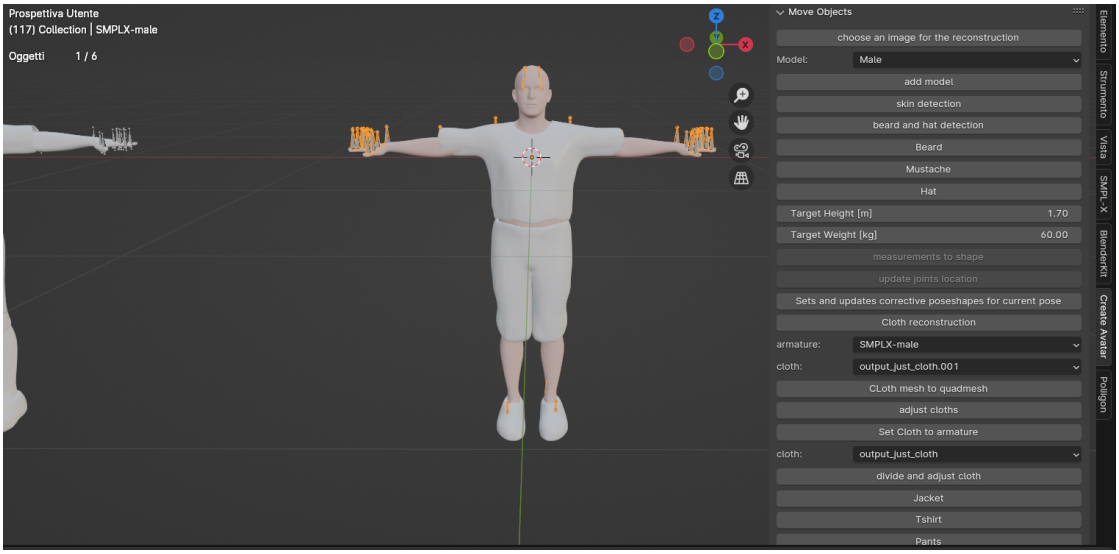
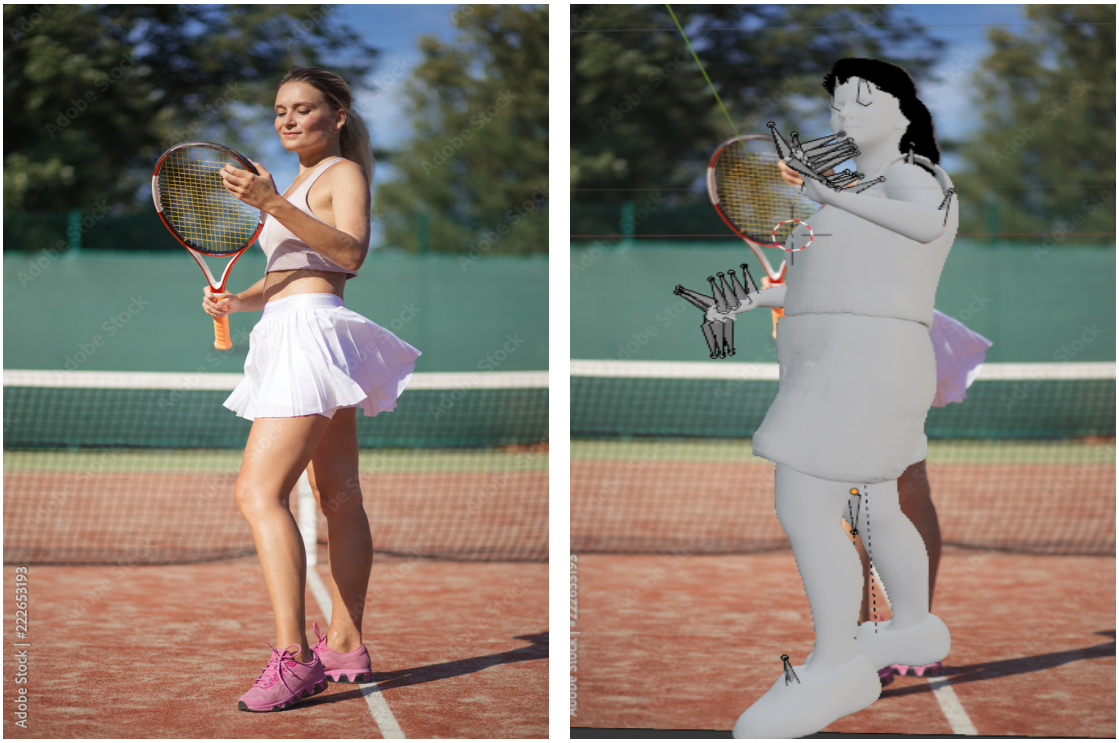


Figura 5.1: Interfaccia Add-on.

rivela robusta anche in condizioni di dati rumorosi, dove SMPL-X, potendo considerare il contesto dell'intero corpo, mantiene una stabilità superiore rispetto a modelli focalizzati solo su parti specifiche come mani o volto. La valutazione quantitativa dei risultati si basa sull'errore di confronto "vertex-to-vertex", che misura la distanza tra i vertici delle mesh 3D generate dal modello e i rispettivi dati di riferimento. Questo tipo di valutazione si rivela particolarmente rigorosa, poiché cattura in modo preciso gli errori di superficie e le rotazioni lungo le ossa del corpo, fornendo quindi un'indicazione più dettagliata rispetto

all'errore sui giunti 3D. L'errore sui giunti 3D, tuttavia, viene mantenuto come metrica di riferimento per permettere un confronto diretto con i modelli esistenti, e SMPL-X mostra comunque una riduzione dell'errore rispetto ai modelli precedenti. Inoltre, SMPL-X è stato sottoposto a uno studio ablativo per valutare il contributo di ciascuno dei suoi componenti alla precisione complessiva del modello. La versione con modello specifico per genere ha prodotto l'errore più basso, mentre l'utilizzo del modello neutrale di genere e l'assenza della penalizzazione per le collisioni hanno comportato un lieve aumento dell'errore. Tali risultati sottolineano come SMPL-X rappresenti un avanzamento significativo nella cattura tridimensionale espressiva del corpo umano, integrando corpo, mani e volto da immagini monoculari RGB in modo realistico. La ricostruzione del vestiario affidato a ClothWild presenta ottimi risulatati grazie anche alle modifiche effettuate sul progetto. La metodologia di valutazione è stata progettata per rispondere alle sfide poste dalle immagini "in-the-wild," che presentano grande variabilità in termini di pose e complessità visiva rispetto ai dataset sintetici su cui i modelli vengono tradizionalmente addestrati. Nell'analisi dei risultati, ClothWild dimostra una maggiore robustezza rispetto ai metodi esistenti, soprattutto nelle situazioni che coinvolgono pose complesse e occlusioni. Inoltre, l'adozione di una funzione di perdita basata su DensePos (Geng et al. [2022])e riduce sensibilmente le ambiguità nella supervisione 2D, migliorando l'accuratezza delle ricostruzioni 3D. Le metriche di valutazione includono la distanza di Chamfer (Fouard and Malandain [2003]), che misura la distanza media tra la superficie della ricostruzione e la superficie di riferimento; questa metrica è stata calcolata dopo un allineamento tra il modello ricostruito e quello di riferimento, utilizzando la proiezione 2D come base per l'associazione delle superfici corrispondenti. La seconda metrica, "Body-Cloth Correspondence", valuta la coerenza tra il modello del corpo e gli indumenti ricostruiti, verificando che le parti del corpo siano coperte dai rispettivi indumenti correttamente modellati. Entrambe le metriche mostrano come ClothWild superi i modelli di riferimento, sia nella precisione della geometria sia nella capacità di adattarsi a pose non convenzionali e a scenari con forti occlusioni. HairStep utilizzato per la ricostruzione dei capelli offre un risultato molto realistico di cui si analizzano i risultati e le metriche di valutazione. Per la valutazione dei risultati, viene utilizzato un confronto oggettivo tramite due nuove metriche oltre all' intersection over union (IoU). La prima, HairSale, calcola l'errore medio angolare nella direzione di crescita dei capelli, valutando la precisione delle ricostruzioni 3D a livello di orientamento. La seconda metrica, HairRida, valuta la precisione relativa della profondità per ogni regione, misurando l'accuratezza della rappresentazione spaziale. Queste misure forniscono un metodo di valutazione più rigoroso rispetto agli studi qualitativi tradizionali, spesso soggetti a errori interpretativi. I risultati sperimentali dimostrano che HairStep migliora la qualità della ricostruzione, garantendo una maggiore fedeltà alla struttura originale dei capelli e una buona generalizzazione ai dati reali rispetto ai metodi preesistenti. Tutto il progetto è stato valutato anche in base alla visual coherence, per consentire una verifica visiva della conformità del modello 3D rispetto all'immagine di partenza. La coerenza visiva rappresenta la corrispondenza tra l'immagine bidimensionale e la sua ricostruzione tridimensionale, assicurando che il modello finale rifletta accuratamente proporzioni, pose e dettagli della fonte originale come visibile in figura 5.2. L'allineamento geometrico, ad esempio, permette di sovrapporre il modello 3D all'immagine per valutare la corrispondenza delle strutture visive e degli angoli. Infine, il confronto dei contorni e



(a) Immagine di partenza.

(b) Ricostruzione.

Figura 5.2: Coerenza visiva.

delle caratteristiche principali del modello 3D con quelli dell'immagine conferma la corrispondenza visiva anche per i dettagli sottili, verificabili attraverso metriche quantitative e, quando necessario, tramite valutazioni umane. La visual coherence consente di verificare la verosimiglianza del risultato ottenuto con l'immagine utilizzata per la ricostruzione.

5.1 Valutazione della Qualità Percettiva dei Mesh Dinamici 3D

Negli ultimi anni, l'analisi quantitativa delle prestazioni in ambiti scientifici e tecnologici è diventata uno strumento cruciale per la valutazione e il miglioramento di modelli e algoritmi. In particolare, l'adozione di metriche standardizzate consente di confrontare in maniera oggettiva i risultati ottenuti e di individuare margini di ottimizzazione. Queste metriche permettono di quantificare in modo rigoroso l'accuratezza, la precisione e l'efficienza di sistemi complessi, favorendo la replicabilità e la trasparenza dei risultati. Le metriche presentate nel lavoro di Fakhri Torkhani (Torkhani et al. [2015]) si concentrano sulla creazione di specifiche metriche per valutare le prestazioni di modelli animati in maniera oggettiva. La scelta delle metriche è stata guidata dalla necessità di fornire una

rappresentazione dettagliata e completa delle capacità dei modelli esaminati, garantendo al contempo che le analisi siano coerenti con i requisiti scientifici e tecnici del settore.

Le metriche utilizzate, descritte nel dettaglio nei capitoli successivi, sono state selezionate per la loro rilevanza in quanto oggettive. Queste misurazioni consentono di definire standard per la valutazione e di confrontare le prestazioni con studi precedenti, favorendo così un progresso sistematico nel campo di applicazione.

5.1.1 Misura di distorsione percettiva basata sulle caratteristiche spaziali

La misura di distorsione percettiva basata sulle caratteristiche spaziali rappresenta il primo passo nella costruzione di una metrica oggettiva per la valutazione della qualità dei mesh dinamici 3D. Questa misura si basa sul concetto di ruvidità locale, che riflette la complessità geometrica della superficie di un mesh in un dato punto. La ruvidità locale è definita come il valore assoluto del Laplaciano della curvatura gaussiana, una proprietà geometrica che cattura le variazioni locali della forma. La distanza percettiva tra un vertice del mesh di riferimento v_{ij}^r e il corrispondente vertice v_{ij}^d del mesh distorto è descritta dalla formula:

$$e_{ij}^s = \frac{|LR_{ij}^r - LR_{ij}^d|}{LR_{ij}^r + LR_{ij}^d + C_{LR}} \tag{5.1}$$

In questa equazione, LR_{ij}^r e LR_{ij}^d rappresentano rispettivamente i valori di ruvidità locale nei due vertici, mentre $C_{LR} = 0.002$ è una costante introdotta per evitare instabilità numeriche in caso di denominatori molto piccoli. Queste distanze locali vengono poi combinate spazialmente attraverso una somma di Minkowski per calcolare la distanza percettiva per frame:

$$f_i^s = \left(\frac{1}{n_v} \sum_{j=1}^{n_v} (e_{ij}^s)^m \right)^{1/m} \tag{5.2}$$

dove $m = 3$ e n_v è il numero di vertici per frame. Questa distanza viene ulteriormente aggregata temporalmente, sempre utilizzando una somma di Minkowski, per ottenere la distanza percettiva globale puramente spaziale (SPD):

$$SPD = \left(\frac{1}{n_f} \sum_{i=1}^{n_f} (f_i^s)^n \right)^{1/n} \tag{5.3}$$

dove $n = 4$ e n_f rappresenta il numero totale di frame. La scelta dei parametri della somma di Minkowski riflette un approccio empirico basato sull'osservazione che le distorsioni più evidenti hanno un impatto maggiore sulla percezione umana.

5.1.2 Misura di distorsione percettiva spaziale ponderata dalla velocità

La misura di distorsione percettiva spaziale ponderata dalla velocità migliora la metrica includendo il parametro della velocità dei vertici, un elemento fondamentale per catturare

l'interazione tra movimenti dinamici e percezione visiva. Si basa sull'osservazione che le distorsioni in aree caratterizzate da elevata velocità di movimento sono generalmente meno visibili, mentre quelle in regioni statiche o quasi statiche risultano più evidenti. Per tenere conto di questo fenomeno, viene calcolata la velocità di ciascun vertice S_{ij}^P, normalizzata rispetto alla diagonale del bounding box del mesh, una misura scalabile che garantisce l'invarianza rispetto alla scala della scena. La ponderazione delle distorsioni locali segue una funzione lineare a tratti, descritta dalla formula:

$$e_{ij}^{st} = \begin{cases} w_1 e_{ij}^s, & \text{se } S_{ij}^P \leq T_1, \\ \left(w_1 + \frac{w_2 - w_1}{T_2 - T_1}(S_{ij}^P - T_1)\right) e_{ij}^s, & \text{se } T_1 < S_{ij}^P \leq T_2, \\ w_2 e_{ij}^s, & \text{se } S_{ij}^P > T_2 \end{cases} \tag{5.4}$$

In questa equazione, $T_1 = 0.01$, $T_2 = 0.05$, $w_1 = 1.0$ e $w_2 = 0.3$ sono parametri empirici determinati per rappresentare il peso relativo delle distorsioni in base alla velocità. Le velocità dei vertici vengono calcolate come la norma dei vettori di movimento normalizzati e si basano sul confronto tra le posizioni dei vertici nei frame consecutivi. Le distanze locali ponderate vengono poi combinate spazialmente utilizzando una somma di Minkowski per ciascun frame. dove $m = 3$. Successivamente, le distanze per frame vengono aggregate temporalmente, ancora una volta utilizzando una somma di Minkowski, per ottenere la distanza globale ponderata dalla velocità (SWSPD):

$$SWSPD = \left(\frac{1}{n_f}\sum_{i=1}^{n_f}(f_i^{st})^n\right)^{1/n} \tag{5.5}$$

dove $n = 4$. Questo approccio consente di modellare con maggiore precisione il modo in cui le distorsioni vengono percepite in relazione alla velocità del movimento, migliorando la correlazione con le valutazioni soggettive e catturando le sfumature della percezione visiva dinamica.

5.1.3 Integrazione delle caratteristiche temporali e della nuova metrica

L'integrazione delle caratteristiche temporali e della nuova metrica rappresenta un ulteriore miglioramento, includendo due elementi fondamentali: il contrasto basato sulla velocità e quello basato sulla direzione del movimento dei vertici. Questi parametri catturano le variazioni dinamiche che non possono essere rappresentate da misure puramente spaziali. Il contrasto basato sulla velocità, descritto dalla formula:

$$s_{ij}^s = \frac{||\vec{d}_{ij}^r|| - ||\vec{d}_{ij}^d||}{||\vec{d}_{ij}^r|| + ||\vec{d}_{ij}^d|| + C_S} \tag{5.6}$$

con $C_S = 0.002$, misura le differenze tra le norme dei vettori di movimento normalizzati del vertice di riferimento e del vertice distorto. In modo simile, il contrasto angolare considera le variazioni nella direzione del movimento ed è definito come:

$$a_{ij}^s = \frac{AG_{ij}^r - AG_{ij}^d}{AG_{ij}^r + AG_{ij}^d + C_{AG}} \tag{5.7}$$

dove AG_{ij}^r e AG_{ij}^d sono gli angoli dei vettori di movimento nei due vertici, e $C_{AG} = \frac{1}{32\pi}$. Entrambi i contrasti sono combinati spazialmente e temporalmente attraverso somme di Minkowski per ottenere due distanze globali: la distanza basata sulla velocità ($SPPD$) e quella basata sulla direzione (APD). Infine, la metrica finale ($DMPD$) combina le distanze spaziali e temporali in una formula unificata:

$$DMPD = \sqrt{w_1(SWSPD)^2 + w_2(SPPD)^2 + w_3(APD)^2} \tag{5.8}$$

dove $w_1 = 0.4$, $w_2 = 0.5$ e $w_3 = 0.1$ sono pesi empiricamente determinati dagli autori delle metriche. Questa combinazione consente di ottenere migliori prestazioni, in particolare nelle regioni con artefatti temporali dominanti, dimostrando il valore della metrica nella valutazione della qualità percettiva dei mesh dinamici. Sebbene la metrica rappresenti un importante passo avanti, il lavoro sottolinea la necessità di ulteriori miglioramenti, in particolare per quanto riguarda strategie di pooling più sofisticate e una comprensione più approfondita degli effetti di mascheramento visivo spaziotemporale.

Metrica	**Mesh di riferimento**	**Mesh abito lungo**	**Mesh maglietta e pantaloncini**	**Mesh gonna e canotta**
SPD	3.2281398	8.62533286	7.3208512	7.3903456
$SWSPD$	1.3947254	1.5101001	1.4755768	1.5128600
$DMPD$	3.1698475	4.2611222	3.5632937	3.9178504
$Vertici$	10475	123495	86958	89469

Tabella 5.1: Confronto tra metriche percettive per varie tipologie di mesh.

In conclusione, è stata adottata come riferimento una mesh ricostruita priva di vestiario, sottoposta a un'animazione di camminata. I valori delle metriche sono stati calcolati su tre modelli distinti, rispettivamente maschile, femminile e neutro, e successivamente ne è stata calcolata la media per ottenere una metriche di riferimento per il paragone dei risultati. Una volta ottenuti i valori relativi a SPD, $SWSPD$ e $DMPD$, il calcolo è stato ripetuto su diverse mesh che includevano vestiario di diverso tipo, utilizzando la medesima animazione per garantire la coerenza dell'analisi. I risultati ottenuti hanno evidenziato prestazioni migliori per mesh caratterizzate da un minor numero di vertici, mentre valori più elevati delle metriche indicano un lieve peggioramento delle prestazioni complessive ma comunque paragonabile al valore di riferimento.

Capitolo 6

Coclusioni e sviluppi futuri

6.1 Conclusioni

Il progetto presentato in questa tesi ha dimostrato come sia possibile ottenere avatar 3D animabili a partire da una singola immagine RGB, sfruttando l'integrazione di modelli di machine learning e tecniche di computer vision. Gli scopi prefissati, che miravano a sviluppare una pipeline capace di generare modelli tridimensionali personalizzabili e ad alto realismo, sono stati in gran parte soddisfatti. Gli avatar prodotti dalla pipeline mostrano una riproduzione accurata della posa, del vestiario e dei dettagli del viso, come capelli e la barba, rispondendo all'obiettivo di costruire un sistema accessibile e flessibile per l'utente finale. Grazie all'integrazione della pipeline all'interno di Blender come add-on, il sistema permette un'interazione intuitiva, offrendo all'utente la possibilità di importare l'immagine di partenza, generare l'avatar e apportare modifiche rapide direttamente nell'ambiente grafico, rispondendo così al requisito di usabilità e accessibilità. Nel confronto dei risultati, emergono punti di forza distintivi del sistema sviluppato, ma anche limitazioni che rappresentano aree di miglioramento per un utilizzo più ampio e una maggiore accuratezza. La combinazione di SMPLify-X per la generazione della posa e della struttura corporea, ClothWild per il vestiario e HairStep per la ricostruzione dei capelli ha fornito risultati efficaci e realistici, garantendo che l'avatar finale rispecchi fedelmente i dettagli del soggetto di riferimento. Il sistema si è dimostrato robusto nell'estrazione dei dettagli visivi anche in condizioni di input limitato, come l'utilizzo di una singola immagine, superando in parte le difficoltà dovute alla mancanza di dati di profondità o immagini multivista. Tuttavia, in presenza di occlusioni nell'immagine o di abiti complessi, si riscontrano talvolta artefatti e imprecisioni nella mesh generata, che influenzano la resa finale. In particolare, l'add-on risulta meno efficace nella gestione di abiti che presentano pieghe complesse o che non aderiscono strettamente alla superficie corporea. In questo contesto, il modello SMPLicit utilizzato da ClothWild si mostra meno efficace nel riprodurre fedelmente capi come gonne, mantelli o abiti larghi, dove sarebbe auspicabile una maggiore autonomia della mesh del vestiario dalla struttura corporea. Un ulteriore aspetto critico è la gestione delle espressioni facciali e delle texture del viso, sebbene accurata, non raggiunge la dinamicità desiderata per applicazioni che richiedono una

variabilità di espressioni o un realismo facciale ancora maggiore. In conclusione, questa tesi ha posto le basi per una pipeline solida e innovativa, aprendo interessanti prospettive di sviluppo per il futuro. I risultati raggiunti dimostrano la validità dell'approccio proposto e il suo potenziale applicativo, e offrono una solida base per ampliamenti successivi, mirati a migliorare la gestione delle espressioni facciali, l'adattabilità del vestiario e l'ottimizzazione computazionale. Tali migliorie potrebbero rendere il sistema ancora più versatile e accessibile, permettendo applicazioni in numerosi settori, come la realtà virtuale, l'intrattenimento il gaming e la moda digitale, contribuendo a ridefinire il ruolo degli avatar 3D come strumenti di interazione e personalizzazione sempre più realistici ed efficienti.

6.2 Sviluppi futuri

I futuri sviluppi della pipeline potrebbero portare miglioramenti significativi sia in termini di realismo che di personalizzazione, rendendo il sistema ancora più versatile e applicabile in contesti complessi. Un primo passo di rilievo sarebbe l'utilizzo di reti GAN o modelli di style transfer (Jing et al. [2018]) per generare la barba e i baffi riconosciuti automaticamente nell'immagine di input, offrendo una rappresentazione più dettagliata e naturale dei peli facciali. Questo approccio consentirebbe di gestire le variazioni di densità, lunghezza e texture della barba in modo più preciso, portando la fedeltà visiva a un livello superiore. Parallelamente, l'impiego di GAN per ottimizzare le texture cutanee migliorerebbe la rappresentazione della pelle, rendendola più autentica e dinamica, con capacità di risposta ai cambiamenti di illuminazione e riflettività, per un avatar più realistico in termini visivi. La simulazione fisica dei tessuti (Zhao [2024]) rappresenta un'ulteriore area di potenziale miglioramento, utile per conferire maggiore realismo al vestiario dell'avatar. Un sistema che modelli pieghe, ondulazioni e comportamenti fisici degli abiti permetterebbe di gestire in modo accurato gli effetti dei movimenti del corpo sugli indumenti, anche per capi voluminosi o non aderenti, come gonne, mantelli o abiti ampi. Questo avanzamento ridurrebbe gli artefatti visibili e migliorerebbe l'aspetto estetico del vestiario, in particolare durante l'animazione dell'avatar, dove la coerenza tra movimento e tessuto è fondamentale per mantenere il realismo. Un'estensione delle opzioni di personalizzazione potrebbe riguardare l'aggiunta di accessori come gioielli, borse e orecchini, identificabili direttamente dall'immagine di input tramite un classificatore dedicato. Questo componente, basato su un approccio simile a quello di ClothWild, riconoscerebbe e categorizzerebbe gli accessori presenti nell'immagine, consentendo di ricostruire e aggiungere automaticamente tali oggetti all'avatar. Tale sistema amplierebbe significativamente le possibilità di personalizzazione, permettendo di rappresentare l'individuo con maggiore fedeltà e autenticità Naik et al. [2020]. Sul fronte delle caratteristiche facciali, l'implementazione di un algoritmo di eye detection (Montazeri et al. [2016]) permetterebbe di riconoscere automaticamente il colore degli occhi del soggetto, applicando una texture corrispondente agli occhi del modello presente nell'immagine in input e consentendo all'utente la possibilità di selezionare un colore alternativo da una libreria predefinita. Questa funzionalità offrirebbe un ulteriore livello di dettaglio, con la possibilità di adattare anche particolari come il colore degli

occhi per una rappresentazione più precisa dell'avatar Infine, l'integrazione di un algoritmo di color detection (Navada et al. [2014]) potrebbe consentire di prelevare fedelmente le colorazioni dall'immagine di input in modo da applicare i pigmenti alla mesh ricostruita, preservando l'aspetto originale del soggetto. Il sistema identificherebbe le tonalità predominanti per vestiario, pelle e accessori, applicando automaticamente tali colori alla mesh 3D. Successivamente, in Blender, l'utente potrebbe personalizzare ulteriormente questi colori, adattandoli alle proprie preferenze. Questa funzione contribuirebbe a migliorare la coerenza dell'avatar rispetto al riferimento visivo e aumenterebbe l'autonomia dell'utente nel gestire le preferenze estetiche. L'adozione di queste nuove funzionalità amplificherebbe le capacità della pipeline, rendendo l'intero processo di ricostruzione 3D ancora più realistico, dinamico e personalizzabile. La possibilità di controllare dettagli complessi come texture cutanee, comportamento del vestiario, accessori, colore degli occhi e delle superfici del modello renderebbe il sistema versatile e adatto a un numero sempre più vasto di applicazioni, includendo non solo ambiti come la realtà virtuale e l'intrattenimento, ma anche la moda digitale, l'educazione e l'assistenza virtuale. Questi sviluppi consoliderebbero il sistema come strumento di riferimento per la generazione di avatar 3D personalizzati e immersivi, allineandosi con le esigenze sempre più sofisticate del settore tecnologico e creativo.

Bibliografia

Free3d. https://free3d.com.

Turbosquid. https://www.turbosquid.com.

Sketchfab. https://sketchfab.com, 2024.

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

Jascha Achenbach, Thomas Waltemate, Marc Erich Latoschik, and Mario Botsch. Fast generation of realistic virtual humans. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, VRST '17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450355483. doi: 10.1145/3139131.3139154. URL https://doi.org/10.1145/3139131.3139154.

Adobe Systems Incorporated. Adobe, 2024a. URL https://www.adobe.com/.

Adobe Systems Incorporated. Mixamo, 2024b. URL https://www.mixamo.com/#/.

Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models, 2018. URL https://arxiv.org/abs/1803.04758.

Luis Arreola, Gesem Gudiño, and Gerardo Flores. Object recognition and tracking using haar-like features cascade classifiers: Application to a quad-rotor uav, 2019. URL https://arxiv.org/abs/1903.03947.

Hao Bai. Icp algorithm: Theory, practice and its slam-oriented taxonomy. *Applied and Computational Engineering*, 2(1):10–21, March 2023. ISSN 2755-273X. doi: 10.54254/2755-2721/2/20220512. URL http://dx.doi.org/10.54254/2755-2721/2/20220512.

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, pages 404–417, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-33833-8.

Blender Foundation. *Blender - A 3D Creation Suite*. Blender Foundation, Amsterdam, The Netherlands, 2024. URL https://www.blender.org. Version 4.1.

G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields, 2019. URL https://arxiv.org/abs/1812.08008.

Scott Chacon and Ben Straub. *Pro git*. Apress, 2014.

Chifffa. Hat and beard classifier. https://github.com/Chifffa/hat_beard_classifier, 2024. Accessed: 2024-10-24.

Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In Vittorio Scarano, Rosario De Chiara, and Ugo Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008. ISBN 978-3-905673-68-5. doi: 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136.

Alex Clark. Pillow (pil fork) documentation, 2015. URL https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf.

Enric Corona, Albert Pumarola, Guillem Alenyà, Gerard Pons-Moll, and Francesc Moreno-Noguer. Smplicit: Topology-aware generative model for clothed people. In *CVPR*, 2021.

Microsoft Corporation. *Visual Studio Code*, 2024. URL https://code.visualstudio.com/. Accessed: 2024-10-18.

Dawson-Haggerty et al. trimesh. URL https://trimesh.org/.

Céline Fouard and Grégoire Malandain. Systematized calculation of optimal coefficients of 3-d chamfer norms. In Ingela Nyström, Gabriella Sanniti di Baja, and Stina Svensson, editors, *Discrete Geometry for Computer Imagery*, pages 214–223, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-39966-7.

Jiaqi Geng, Dong Huang, and Fernando De la Torre. Densepose from wifi, 2022. URL https://arxiv.org/abs/2301.00250.

Inc. GitHub. *GitHub*, 2024. URL https://github.com/. Accessed: 2024-10-18.

Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018. doi: 10.1109/CVPR.2018.00762.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre

Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007.

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7): 1325–1339, 2014. doi: 10.1109/TPAMI.2013.248.

Boyi Jiang, Juyong Zhang, Yang Hong, Jinhao Luo, Ligang Liu, and Hujun Bao. Bcnet: Learning body and cloth shape from a single image, 2020. URL https://arxiv.org/abs/2004.00214.

Jiaxi Jiang, Paul Streli, Huajian Qiu, Andreas Fender, Larissa Laich, Patrick Snape, and Christian Holz. Avatarposer: Articulated full-body pose tracking from sparse motion sensing, 2022. URL https://arxiv.org/abs/2207.13784.

Xin Jin and Jiawei Han. *K-Means Clustering*, pages 563–564. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. doi: 10.1007/978-0-387-30164-8_425. URL https://doi.org/10.1007/978-0-387-30164-8_425.

Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review, 2018. URL https://arxiv.org/abs/1705.04058.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Zorah Laehner, Daniel Cremers, and Tony Tung. Deepwrinkles: Accurate and realistic clothing modeling, 2018. URL https://arxiv.org/abs/1808.03417.

Hanchao Li and Xinguo Liu. Strand-accurate multi-view facial hair reconstruction and tracking. *The Visual Computer*, 40(7):4713–4724, July 2024. ISSN 1432-2315. doi: 10.1007/s00371-024-03465-5. URL https://doi.org/10.1007/s00371-024-03465-5.

Suixian Li, Kaida Xiao, and Pingqi Li. Spectra reconstruction for human facial color from rgb images via clusters in 3d uniform cielab* and its subordinate color space. *Sensors*, 23(2), 2023. ISSN 1424-8220. doi: 10.3390/s23020810. URL https://www.mdpi.com/1424-8220/23/2/810.

Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. URL https://doi.org/10.1145/3130800.3130813.

Yunhao Liang, Ziang Ye, Wen Liu, and Huchuan Lu. Scape: A simple and strong category-agnostic pose estimator. *arXiv preprint arXiv:2407.13483*, 2024. URL https://arxiv.org/abs/2407.13483.

Chen-Hsuan Lin, Oliver Wang, Bryan C. Russell, Eli Shechtman, Vladimir G. Kim, Matthew Fisher, and Simon Lucey. Photometric mesh optimization for video-aligned 3d object reconstruction, 2019. URL https://arxiv.org/abs/1903.08642.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. URL https://arxiv.org/abs/1405.0312.

Tony Lindeberg. *Scale Invariant Feature Transform*, volume 7. 05 2012. doi: 10.4249/scholarpedia.10491.

Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989. ISSN 1436-4646. doi: 10.1007/BF01589116. URL https://doi.org/10.1007/BF01589116.

Matthew Loper. Chumpy: A python framework for differentiable optimization. https://github.com/mattloper/chumpy, 2014. URL https://github.com/mattloper/chumpy. Version 0.69.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.

Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics*, 36(4):1–14, July 2017. ISSN 1557-7368. doi: 10.1145/3072959.3073596. URL http://dx.doi.org/10.1145/3072959.3073596.

Mitra Montazeri, Mahdieh Montazeri, and Saeid Saryazdi. Eye detection in digital images: challenges and solutions, 2016. URL https://arxiv.org/abs/1601.04871.

Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Accurate 3d hand pose estimation for whole-body 3d human mesh estimation, 2022a. URL https://arxiv.org/abs/2011.11534.

Gyeongsik Moon, Hyeongjin Nam, Takaaki Shiratori, and Kyoung Mu Lee. 3d clothed human reconstruction in the wild, 2022b. URL https://arxiv.org/abs/2207.10053.

Umut Müftüoğlu, Georgios Pavlakos, Tim Von Marcard, Mayu Otani, Angjoo Kanazawa, and Michael J. Black. Human body prior, 2024. URL https://pypi.org/project/human-body-prior/.

Roshan G. Naik, Parth Singh, and Prem Kumar Kalra. Putting jewellery and accessories on a 3d face model generated from 2d image. 2020. URL https://api.semanticscholar.org/CorpusID:229215676.

Bhagya R Navada, K V Santhosh, S Prajwal, and Harikishan B Shetty. An image processing technique for color detection and distinguish patterns with similar color: An aid for color blind people. In *International Conference on Circuits, Communication, Control and Computing*, pages 333–336, 2014. doi: 10.1109/CIMCA.2014.7057818.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019a. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019b. URL https://arxiv.org/abs/1912.01703.

Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

Jérôme Perret and Emmanuel Vander Poorten. Touching virtual reality: a review of haptic gloves. 06 2018.

Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael Black. Clothcap: Seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, 36(4), 2017. URL http://dx.doi.org/10.1145/3072959.3073711. Two first authors contributed equally.

Python Software Foundation. *Python: A programming language*, 2023a. URL https://www.python.org. Version 3.11.4, accessed October 2023.

Python Software Foundation. *Python Collections Library: Counter*. Python Software Foundation, 2023b. URL https://docs.python.org/3/library/collections.html#collections.Counter. Accessed: 2023-10-26.

Grégory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

Vivien Rolland, Moshiur R. Farazi, Warren C. Conaty, Deon Cameron, Shiming Liu, Lars Petersson, and Warwick N. Stiller. Hairnet: a deep learning model to score leaf hairiness, a key phenotype for cotton fibre yield, value and insect resistance. *Plant Methods*, 18(1):8, January 2022. ISSN 1746-4811. doi: 10.1186/s13007-021-00820-8. URL https://doi.org/10.1186/s13007-021-00820-8.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), November 2017.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Gemma Rotger, Francesc Moreno-Noguer, Felipe Lumbreras, and Antonio Agudo. Single view facial hair 3d reconstruction. In Aythami Morales, Julian Fierrez, José Salvador Sánchez, and Bernardete Ribeiro, editors, *Pattern Recognition and Image Analysis*, pages 423–436, Cham, 2019. Springer International Publishing. ISBN 978-3-030-31332-6.

Sebastian Ruder. An overview of gradient descent optimization algorithms, 2017. URL https://arxiv.org/abs/1609.04747.

Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization, 2020. URL https://arxiv.org/abs/2004.00452.

A. Erman Tekkaya and Celal Soyarslan. *Finite Element Method*, pages 508–514. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-642-20617-7. doi: 10.1007/978-3-642-20617-7_16699. URL https://doi.org/10.1007/978-3-642-20617-7_16699.

Fakhri Torkhani, Kai Wang, and Jean-Marc Chassery. Perceptual quality assessment of 3d dynamic meshes: Subjective and objective studies. *Signal Processing: Image Communication*, 31:185–204, 2015. ISSN 0923-5965. doi: https://doi.org/10.1016/j.image.2014.12.008. URL https://www.sciencedirect.com/science/article/pii/S0923596514001866.

Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision (ECCV)*, sep 2018.

Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV (11)*, volume 11215 of *Lecture Notes in Computer Science*, pages 55–71. Springer, 2018. ISBN 978-3-030-01252-6. doi: 10.1007/978-3-030-01252-6_4. URL https://arxiv.org/abs/1804.01654.

Ziyan Wang, Giljoo Nam, Tuur Stuyck, Stephen Lombardi, Chen Cao, Jason Saragih, Michael Zollhoefer, Jessica Hodgins, and Christoph Lassner. Neuwigs: A neural dynamic model for volumetric hair capture and animation, 2023. URL https://arxiv.org/abs/2212.00613.

Keyu Wu, Yifan Ye, Lingchen Yang, Hongbo Fu, Kun Zhou, and Youyi Zheng. Neuralhdhair: Automatic high-fidelity hair modeling from a single image using implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1526–1535, 2022a. doi: 10.1109/CVPR52688.2022.00155.

Keyu Wu, Yifan Ye, Lingchen Yang, Hongbo Fu, Kun Zhou, and Youyi Zheng. Neuralhdhair: Automatic high-fidelity hair modeling from a single image using implicit neural representations, 2022b. URL https://arxiv.org/abs/2205.04175.

Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J. Black. Icon: Implicit clothed humans obtained from normals, 2022. URL https://arxiv.org/abs/2112.09127.

Lingchen Yang, Zefeng Shi, Youyi Zheng, and Kun Zhou. Dynamic hair modeling from monocular videos using deep neural networks. *ACM Trans. Graph.*, 38(6), November 2019. ISSN 0730-0301. doi: 10.1145/3355089.3356511. URL https://doi.org/10.1145/3355089.3356511.

Hongwen Zhang, Yating Tian, Xinchi Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop, 2021. URL https://arxiv.org/abs/2103.16507.

Tuanshan Zhang and Haoran Ma. Clothnet: sensitive semantic segmentation network for fabric defect detection. *Textile Research Journal*, 93(1-2):103–115, 2023. doi: 10.1177/00405175221114927. URL https://doi.org/10.1177/00405175221114927.

Zhiwei Zhao. A physics-embedded deep learning framework for cloth simulation, 2024. URL https://arxiv.org/abs/2403.12820.

Yujian Zheng, Zirong Jin, Moran Li, Haibin Huang, Chongyang Ma, Shuguang Cui, and Xiaoguang Han. Hairstep: Transfer synthetic to real using strand and depth maps for single-view 3d hair modeling. https://github.com/GAP-LAB-CUHK-SZ/HairStep, 2023a.

Yujian Zheng, Zirong Jin, Moran Li, Haibin Huang, Chongyang Ma, Shuguang Cui, and Xiaoguang Han. Hairstep: Transfer synthetic to real using strand and depth maps for single-view 3d hair modeling, 2023b. URL https://arxiv.org/abs/2303.02700.

Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 766–782, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46475-6.

Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing, 2018a. URL https://arxiv.org/abs/1801.09847.

Yi Zhou, Liwen Hu, Jun Xing, Weikai Chen, Han-Wei Kung, Xin Tong, and Hao Li. Hairnet: Single-view hair reconstruction using convolutional neural networks, 2018b. URL https://arxiv.org/abs/1806.07467.