

POLITECNICO DI TORINO

Laurea magistrale in Ingegneria del Cinema e dei Mezzi di Comunicazione



Tesi di Laurea Magistrale

Cattura volumetrica per la produzione televisiva: analisi delle potenzialità e limitazioni

Relatori:

Andrea Bottino

Francesco Strada

Alberto Ciprian

Davide Zappia

Candidati:

Giada Ferreri

Carolina Icardi

In collaborazione con



Ottobre 2024

Abstract

La seguente tesi si propone di individuare e testare sistemi per la realizzazione di video volumetrici di buona qualità, che possano essere integrati in produzioni televisive Rai. In un panorama multimediale sempre più ibrido tra reale e virtuale, è fondamentale disporre di rappresentazioni tridimensionali realistiche di persone. Il video volumetrico grazie all'utilizzo del live action, invece della computer grafica e dell'animazione, permette di facilitare le rappresentazioni tridimensionali di contenuti dinamici. Inoltre, la possibilità di modificare l'illuminazione e i movimenti di camera dopo la fase di ripresa offre una maggiore flessibilità creativa durante fase di post-produzione.

La cattura volumetrica a basso costo è ancora in fase emergente e, per questo motivo, presenta alcune limitazioni. I recenti sviluppi nel deep learning stanno migliorando e rendendo più accessibile questa tecnologia, ma il processo per una completa democratizzazione e un alto livello di accessibilità è ancora lungo.

La ricerca si è orientata verso l'utilizzo di diversi software, permettendo l'analisi di molteplici soluzioni tecnologiche: algoritmi di deep learning, sensori di profondità e LiDAR. Sono state testate le applicazioni Volu, Depthkit e V3LCamera e l'algoritmo open source del Gaussian Splatting 4D.

Le catture volumetriche, ottenute dalle tre differenti applicazioni, sono state inserite all'interno di ambienti virtuali al fine di poterne valutare l'integrazione e la bontà delle ricostruzioni volumetriche. Dopo un'analisi delle diverse tecnologie condotta in maniera indipendente, si è ritenuto opportuno svolgere un confronto diretto. Volu si è dimostrata la migliore per il rapporto qualità-prezzo e fotorealismo. Depthkit e V3LCamera hanno mostrato tempistiche ridotte in fase di elaborazione, ma limiti in termini di realismo.

In seguito, è stata analizzata la cattura volumetrica in real time, ottenendo risultati buoni solamente con un angolo di mobilità visiva di pochi gradi. Nonostante sia stata rilevata una latenza accettabile, le ricostruzioni soffrono di numerosi artefatti, che peggiorano all'aumentare della distanza del soggetto dallo strumento di cattura.

Il Gaussian Splatting 4D ha, invece, richiesto un'analisi indipendente a causa della mancata possibilità di integrazione in motori grafici. Questa tecnologia è risultata promettente, ma non ancora ottimizzata per un utilizzo televisivo. Lo scenario potrebbe evolvere in futuro apportando numerosi vantaggi alle produzioni televisive,

considerandone i costi trascurabili. Tali assunzioni sono state ulteriormente confermate da test soggettivi, condotti su 20 persone.

Indice

Capitolo 1 - Introduzione	1
1.1 Contesto e motivazioni	6
1.2 Obiettivi della tesi	9
1.3 Requisiti Rai	10
Capitolo 2 - Stato dell'arte	12
2.1 Formati	12
2.1.1 Mesh	12
2.1.2 Nuvola di punti	13
2.1.3 Voxel	13
2.1.4 NeRF	14
2.1.5 Gaussian Splatting	15
2.2 Panoramica sulla pipeline del video volumetrico	17
2.2.1 Cattura	17
2.2.2 Compressione	17
2.2.3 Trasmissione	18
2.2.4 Rendering	19
2.2.5 Visualizzazione	20
2.3 Tecniche per la cattura volumetrica	21
2.3.1 Array di camere calibrate	21
2.3.2 Camera singola	24
2.4 Casi studio	26

Capitolo 3 - Tecnologie utilizzate	34
3.1 Volu.....	34
3.1.1 Storia azienda e costi.....	34
3.1.2 Funzionamento tecnologia	36
3.2 Depthkit	41
3.2.1 Storia azienda e costi.....	41
3.2.2 Funzionamento tecnologia	43
3.3 V3LCamera.....	50
3.3.1 Storia azienda e costi.....	50
3.3.2 Funzionamento tecnologia	51
3.4 Gaussian Splatting 4D	58
3.4.1 Storia ed evoluzione delle reti neurali per il volumetrico	58
3.4.2 Funzionamento tecnologia	59
 Capitolo 4 - Valutazione delle tecnologie	 71
4.1 Volu.....	71
4.2 Depthkit	83
4.3 V3LCamera.....	91
4.4 Confronti tra tecnologie	102
4.5 Confronto con il progetto europeo XReco.....	106
4.6 Gaussian Splatting 4D	108
 Capitolo 5 - Risultati sperimentali	 119
5.1 Analisi soggettive.....	119
5.1.1 Setup di cattura.....	119

5.1.2 Setup test	122
5.1.3 Discussione risultati	124
5.1.3.1 Volu, Depthkit e V3LCamera	124
5.1.3.2 Gaussian Splatting 4D	133
5.1.3.3 Real time.....	149
5.2 Analisi oggettive	153
5.2.1 Parametri utilizzati	153
5.2.2 Risultati	156
5.3 Analisi soggettive e oggettive 4D-GS a confronto	158
 Capitolo 6 - Conclusioni	 160
6.1 Considerazioni finali.....	160
6.2 Limiti	161
6.3 Sviluppi futuri.....	162
 Appendice.....	 165
Acronimi.....	172
Lista delle figure.....	174
Lista dei grafici.....	181
Lista delle tabelle	184
Bibliografia e sitografia.....	185

Capitolo 1

Introduzione

Il panorama dei servizi multimediali, negli ultimi anni, ha subito significative trasformazioni. Il video bidimensionale è progredito gradualmente verso la realizzazione dei video panoramici (video a 360°) ed è infine approdato al video volumetrico. Questa evoluzione è stata anche dettata dal desiderio del pubblico di fruire una realtà replicata, oltrepassando i limiti imposti dagli schermi bidimensionali.

Il video a 360° viene realizzato utilizzando una telecamera a 360° o molteplici telecamere 2D disposte in una sfera. Il vantaggio principale si riscontra nel realismo ottenuto che permette di non registrare nessun *uncanny valley effect* negli spettatori [1]. Infatti, il video a 360° riproduce attori reali e oggetti fisici senza la necessità di ricorrere alla computer grafica, a differenza di quanto accade in esperienze di realtà virtuale ed aumentata. D'altra parte, lo spettatore, in un video a 360°, non può muoversi liberamente nell'ambiente né interagire con oggetti o attori riprodotti.

La *volumetric capture* è, invece, una tecnologia che permette di catturare nello spazio reale per poi ricostruire in 3D un ambiente, un oggetto o una persona nel tempo. Il video volumetrico grazie all'utilizzo del live action, invece della computer grafica e dell'animazione, permette di facilitare le rappresentazioni tridimensionali di contenuti dinamici. Il live action ripreso viene trasformato in modello 3D generando nuvole di punti o mesh tridimensionali.

L'idea alla base del video volumetrico non è affatto recente e nasce dalla volontà di replicare il più fedelmente possibile il mondo reale in digitale. Gli iconici ologrammi di Star Wars e Blade Runner hanno ispirato e contribuito ad alimentare la volontà di replicare la realtà con estremo livello di dettaglio, superando i limiti e le costrizioni imposti dal 2D [2]. Il video volumetrico permette di rivoluzionare il modo in cui vengono usufruiti i contenuti video. Infatti, permette allo spettatore di sentirsi veramente presente nell'ambiente, vivendo un'esperienza molto più coinvolgente e accattivante. I video volumetrici si distinguono dai video tradizionali per la loro capacità di offrire un'esperienza senza precedenti di immersione spazializzata e interattività a sei gradi di

libertà (DoF) (figura 1.1). Ciò include tre dimensioni della posizione di visione (X, Y, Z) e tre dimensioni dell'orientamento della visione ($yaw, pitch, roll$) [2]. Per avere sei gradi di libertà, tutti gli oggetti della scena devono avere una rappresentazione tridimensionale. Per ottenere un modello 3D realistico, è necessario acquisire informazioni da molteplici punti di vista e ricavare le opportune informazioni sulla geometria, sulla texture e sul moto.

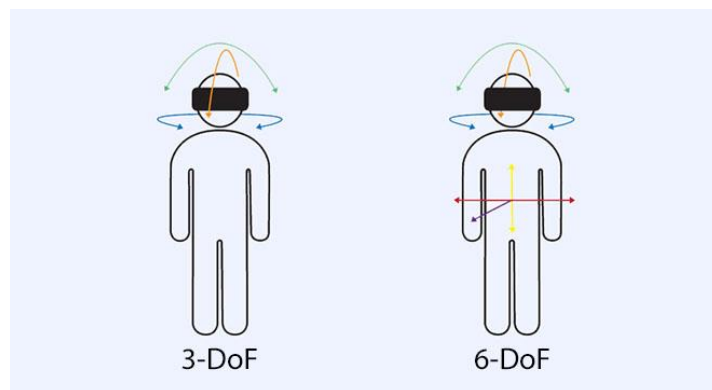


Figura 1.1: Tre gradi di libertà vs sei gradi di libertà

Fonte: <https://is.gd/VUkByn>

Una caratteristica distintiva di un video volumetrico è la libertà concessa allo spettatore di percepire l'immagine dal punto di vista da lui scelto piuttosto che dal punto di vista scelto per lui dal regista, consentendo nuove forme di narrazione audiovisiva. La scelta delle dimensioni dell'inquadratura, degli angoli di ripresa, dei movimenti della telecamera e delle posizioni degli attori rispetto alla telecamera spetta ora allo spettatore. Di conseguenza, il processo di osservazione non può più essere descritto come passivo.

I servizi video volumetrici e le tecnologie sottostanti hanno un enorme potenziale nel rivoluzionare le applicazioni multimediali del futuro. L'avanzamento nella computer grafica e nel processamento delle informazioni hanno avuto un ruolo cruciale nel passaggio dal video bidimensionale al video volumetrico tridimensionale. Nonostante il rapido sviluppo, la tecnologia relativa al video volumetrico presenta ancora un enorme potenziale di crescita e innovazione.

L'acquisizione volumetrica è il primo passo per produrre video volumetrici. Poiché il video volumetrico è rappresentato da contenuti 3D, il processo di acquisizione è molto diverso dai tradizionali video bidimensionali. La cattura volumetrica è una tecnologia che registra sia il colore del pixel che la sua posizione nello spazio 3D. Si tratta, quindi, di

una cattura della topologia, piuttosto che di una proiezione 2D, di attori, set, oggetti di scena, oggetti e ambienti. Nonostante l'uso del termine "volumetrico", l'acquisizione volumetrica utilizzata non cattura il volume dell'oggetto di per sé [1]. Infatti, la cattura volumetrica registra le informazioni sulla topologia dell'oggetto o della persona catturata, senza alcuna informazione sull'interno. Il processo di cattura avviene utilizzando una varietà di dati di input: dati RGB, dati di profondità e dati LiDAR.

I video, ottenuti tramite cattura volumetrica, possono essere successivamente trasmessi attraverso Internet utilizzando varie reti di accesso, tra cui Ethernet, WiFi o reti cellulari. Una volta trasmessi, i video volumetrici possono essere visualizzati su vari dispositivi, inclusi desktop, dispositivi mobili e *Head Mounted Display* (HMD). Questi ultimi offrono un'esperienza più coinvolgente per gli utenti, i quali possono immergersi in un'esperienza virtuale fotorealistica e navigare liberamente nei contenuti video da qualsiasi posizione e angolo di visione.

Con la rapida evoluzione sperimentata dalla realtà estesa (XR), la domanda di contenuti volumetrici di alta qualità è aumentata in modo significativo. Tra questi contenuti si individuano non solo video pubblicitari e musicali, ma anche programmi TV e altre forme di contenuto. Infatti, le tecniche manuali per generare avatar e ambienti tridimensionali realistici richiedono sforzi sostanziali da parte di artisti e sviluppatori. I sistemi di acquisizione volumetrica possono facilitare questo processo, ottenendo queste informazioni direttamente dal mondo reale [3].

Di seguito, vengono riportati alcuni esempi particolarmente significativi di utilizzi di video volumetrici. Nel 2019, ai Billboard Music Awards, Madonna e i suoi numerosi ologrammi hanno inaugurato il primo evento live televisivo che combina riprese real time con la cattura volumetrica [4]. La riapparizione dopo quattro anni di assenza dal palco della cantante è stata decisamente spettacolare e scenica: sulle note di "Medellin", quattro ologrammi di Madonna hanno popolato e animato il palco. I modelli volumetrici erano stati precedentemente catturati, gli effetti pre-processati e infine riprodotti in live. Il risultato finale è una performance in realtà aumentata, visibile solo per gli spettatori da casa, in cui i cantanti, Madonna e Maluma, e i ballerini in live action dividono il palco con i video volumetrici (figura 1.2).



Figura 1.2: Madonna ai Billboards Music Awards (2019)

Fonte: <https://is.gd/jdnzza>

Un ulteriore esempio concreto dell'utilizzo di questa tecnologia si può individuare nel video musicale 'My Universe', rilasciato nel 2021. Nel video in questione, sia i Coldplay che i BTS sono stati catturati volumetricamente alle estremità opposte del pianeta e successivamente le loro riproduzioni tridimensionali sono state intrecciate perfettamente [5]. Inoltre, l'esibizione live dei Coldplay, alla finale di The Voice USA, è stata resa possibile sfruttando la riproduzione volumetrica dei BTS precedentemente acquisita (figura 1.3). La cattura volumetrica in questione è avvenuta in una struttura mobile costituita da un insieme di 106 camere. Otto milioni di spettatori sono rimasti affascinati dalla performance a The Voice, mentre il video musicale ha raggiunto 300 milioni di visualizzazioni. Questo è un chiaro esempio di come la convergenza della cattura volumetrica e la virtual production in real time stiano rivoluzionando la creazione di contenuti broadcast e di intrattenimento cinematografico [6].



Figura 1.3: Live The Voice USA

Fonte: <https://is.gd/8Hj1k0>

In alcune scene di “The Matrix Resurrections” si può rintracciare l’applicazione della cattura volumetrica. Infatti, le scene in questione sono state registrate in uno studio volumetrico (Volucap) che è riuscito a raggiungere questo obiettivo costruendo diversi sistemi di telecamere volumetriche. In primo luogo, sono state realizzate delle soluzioni portatili di cattura volumetrica e successivamente è stato costruito, in una piscina profonda sette metri, il primo studio volumetrico subacqueo al mondo (figura 1.4). Ciò ha reso possibile creare, in un secondo momento, nuovi movimenti di telecamera, guardare dietro gli attori, mescolare volti e performance di attori diversi e progettare nuovi effetti visivi [7].



Figura 1.4: Studio subacqueo di cattura volumetrica Volucap

Fonte: <https://is.gd/PghttpA>

Lo studio e lo sviluppo di sistemi di cattura volumetrica, dunque, è in corso da anni ma le prestazioni raggiunte da queste tecnologie non sempre soddisfano le esigenze dei professionisti del settore a causa di problemi di qualità e limiti di tempo. Infatti, ci sono alcune barriere dovute all’ingente costo dell’hardware e al massiccio lavoro di post-produzione. Sono state condotte numerose ricerche per cercare di semplificare il processo di acquisizione ed elaborazione dei dati e rendere più accessibile la tecnologia, ma non è stato ancora stabilito uno standard. Inoltre, la volumetric capture può fornire ottimi risultati usando sofisticati sistemi offline, mentre la cattura real time, specialmente se non si usano complessi setup, è particolarmente difficile. Le recenti scoperte nelle tecniche di deep learning e nelle architetture di rete neurale stanno cambiando questa situazione ma, nonostante l’eccezionale qualità ed affidabilità raggiunta da questi approcci, la maggior parte dei modelli all’avanguardia presenta ancora degli svantaggi.

1.1 Contesto e motivazioni

Al giorno d'oggi gli ambienti virtuali offrono esperienze coinvolgenti da qualsiasi prospettiva scelta. Sebbene l'attuale generazione di ambienti virtuali possa effettivamente trasmettere dettagli fotorealistici per un'ampia gamma di oggetti e scene, la rappresentazione degli esseri umani non raggiunge lo stesso livello di qualità. Tale limitazione incrementa il fenomeno di uncanny valley, effetto in grado di destare sensazioni spiacevoli di repulsione e disagio nello spettatore di fronte alla vista di umanoidi riprodotti in computer grafica. Per tale motivo, spesso non si riesce a suscitare una sospensione dell'incredulità nello spettatore. Il realismo degli avatar, inoltre, è fondamentale per aumentare la comunicazione, l'empatia, la fiducia, il pensiero critico e il processo decisionale nell'istruzione, nella formazione e nell'intrattenimento in ambienti immersivi. Al fine di sperimentare la vera telepresenza, mancano dei fattori umani critici necessari come il contatto visivo, le microespressioni facciali, il linguaggio naturale del corpo, il coinvolgimento da persona a persona e la partecipazione effettiva [8]. Il processo di *motion capture* richiederebbe molto tempo e non riuscirebbe a rappresentare tutti i movimenti dettagliati di un attore e dei suoi vestiti. Ciò può, invece, essere ottenuto tramite il video volumetrico.

Si prevede che il video volumetrico consentirà nuovi casi d'uso nel settore dell'intrattenimento (ad esempio replay sportivi, concerti e giochi), nel patrimonio culturale, nell'istruzione, nella comunicazione e nel commercio. Infatti, attualmente esistono numerosi studi di acquisizione volumetrica che vengono utilizzati per la produzione commerciale (figura 1.5).



Figura 1.5: Cattura volumetrica utilizzata per una partita di basket

Fonte: <https://is.gd/tauKIU>

Negli ultimi anni il concetto di studio televisivo si è gradualmente esteso e non è più solo focalizzato alla produzione di video convenzionali, ma anche alla realizzazione di ambienti virtuali. Gli studi di acquisizione volumetrica utilizzano molte telecamere per massimizzare la qualità della ricostruzione virtuale. L'utilizzo di più telecamere, tuttavia, aumenta anche i costi computazionali e i tempi di elaborazione, generando spesso un collo di bottiglia, soprattutto per la ricostruzione in tempo reale. È importante, quindi, ridurre il numero di telecamere garantendo al tempo stesso una qualità di ricostruzione sufficiente per l'utilizzo dei programmi televisivi. Un ulteriore fattore che può influire negativamente nell'utilizzo pratico della cattura volumetrica negli studi televisivi sono i possibili problemi operativi (ad esempio il maggiore rischio di guasti del sistema) che, dunque, devono essere ridotti.

La cattura volumetrica è una tecnologia emergente e, per tale motivo, presenta ancora diverse problematiche che aprono, però, la possibilità a ricerche e sviluppi futuri. A causa della natura complessa che contraddistingue il video volumetrico, si ha una mancanza di procedure di test e set di dati standardizzati e ciò rappresenta un'opportunità significativa per i ricercatori.

I video volumetrici sono in genere di dimensioni decisamente maggiori rispetto alle loro controparti 2D; ciò rappresenta una sfida significativa per le risorse informatiche e di larghezza di banda nelle infrastrutture odierne. In aggiunta, la compressione dei video volumetrici ha ricevuto relativamente poca attenzione e, quindi, presenta numerose opportunità per ulteriori esplorazioni. Inoltre, il rendering dei video volumetrici, nonostante il significativo aumento della potenza di calcolo dei dispositivi mobili, è ancora un compito molto impegnativo.

L'unione di modelli linguistici di grandi dimensioni (LLM), come GPT-4, con la tecnologia video volumetrica apre un mondo di opportunità che amplia i confini della creazione, dell'interazione e della personalizzazione dei contenuti [2]. Quando integriamo il video volumetrico con i LLM, possiamo aggiungere una funzionalità extra: la reattività. La fusione di LLM e tecnologia video volumetrica è una potente combinazione che può migliorare significativamente le esperienze digitali. Inoltre, l'integrazione dei contenuti generati dall'intelligenza artificiale (AIGC) con video volumetrici presenta numerose opportunità di ricerca. L'AIGC ha il potenziale per migliorare il processo di creazione di video volumetrici generando contenuti 3D di alta

qualità utilizzando meno telecamere o da input guidati da testo, riducendo così la complessità e i costi di produzione. Inoltre, AIGC può inserire dati mancanti o correggere errori nei filmati acquisiti, migliorando in definitiva la qualità e il realismo dell'output finale.

Al momento, si può affermare che la possibilità di spostare liberamente la propria attenzione tra i mondi digitali e il mondo reale sta diventando sempre più familiare a molte persone. In questo contesto, stanno emergendo nuove espressioni artistiche e comunicative che sono in grado di collegare il mondo reale e quello virtuale.

1.2 Obiettivi della tesi

La tesi si propone di individuare e testare sistemi per la realizzazione di video volumetrici, che siano integrabili in programmi televisivi. Nello specifico, si desidera ricostruire digitalmente una persona e i suoi movimenti nel modo più accurato possibile.

La Rai, come la maggior parte delle produzioni televisive, non dispone di studi appositi per la cattura volumetrica. Costruire un ambiente ad hoc per la realizzazione di video volumetrici è molto costoso e complesso: è dunque necessario trovare alternative che permettano di ottenere buoni risultati senza dover utilizzare attrezzature altamente specializzate. La ricerca si deve orientare al trovare soluzioni disponibili sul mercato, che siano a costo ridotto e permettano di realizzare catture volumetriche anche in setup non specifici. I video volumetrici ottenuti, però, devono raggiungere una qualità simile a quella richiesta dalla distribuzione televisiva broadcast. Raggiungere questo obiettivo non è semplice: la ricerca su sistemi a costo ridotto è piuttosto limitata e presenta numerose problematiche e sfide da superare. È fondamentale, inoltre, valutare in quali contesti l'integrazione di video volumetrici in programmi televisivi possa risultare vantaggiosa e se possa contribuire a semplificare e ridurre i costi della pipeline di produzione.

Durante la fase di ricerca, sono state cercate anche soluzioni per la cattura volumetrica in tempo reale. Gli studi sull'argomento, però, sono spesso datati e la qualità ottenuta non sempre soddisfa gli standard televisivi. Nonostante ciò, la cattura in tempo reale permetterebbe di ridurre al minimo i tempi di elaborazione e potrebbe essere un'ottima alternativa per rispettare gli stringenti tempi televisivi. Per questo motivo, si è deciso di analizzare anche questa tipologia di cattura volumetrica.

Una volta individuate delle possibili soluzioni che rispecchino i criteri prima citati, si dovrà provvedere a testarle in diversi contesti e a immaginarne possibili applicazioni concrete. Per ciascuna tecnologia, dovranno essere analizzati aspetti positivi e negativi, che ne evidenzino potenzialità ed eventuali criticità. A fronte di quest'analisi, i diversi sistemi potranno essere confrontati tra loro e si potrà decretare l'alternativa migliore. Il risultato auspicabile è trovare un metodo in grado di sfruttare al massimo il potenziale della cattura volumetrica per arricchire i programmi televisivi, mantenendo al contempo un processo produttivo efficiente e sostenibile.

1.3 Requisiti Rai

La ricerca sulla cattura volumetrica svolta ha dovuto tenere conto di determinati requisiti Rai. In primo luogo, la limitazione sul budget ha orientato la ricerca verso applicazioni che disponessero di versioni di prova gratuite. Ciò ha portato a una scrematura delle tecnologie presenti sul mercato in quanto, essendo un campo di ricerca agli albori e complesso, la quantità di denaro richiesta per il loro utilizzo è spesso molto onerosa. Gli sviluppi futuri molto probabilmente renderanno maggiormente accessibile questa tecnologia ma, oggi, le principali catture volumetriche vengono realizzate in studi di produzione specializzati e molto costosi. Inoltre, i motori grafici come Unity e Unreal Engine sono ancora una novità nel settore della produzione televisiva. Di conseguenza, c'è una carenza di personale qualificato con le competenze necessarie per avviare una produzione su larga scala che sfrutti pienamente le potenzialità offerte da queste tecnologie. Per tale motivo, la ricerca si è orientata verso l'individuazione di soluzioni più accessibili e meno complesse al fine di ottenere una cattura volumetrica facilmente utilizzabile, senza dover ricorrere obbligatoriamente ad attrezzature specializzate.

La qualità richiesta per una produzione televisiva deve rispettare determinati standard. Infatti, il pubblico televisivo è abituato a fruire programmi che rispecchiano fedelmente la realtà. Tale caratteristica, dunque, deve essere mantenuta anche nella produzione di catture volumetriche. In quest'ultimo campo, raggiungere il fotorealismo risulta, però, molto complicato. Infatti, tutte le tecnologie hanno sempre dovuto seguire numerose evoluzioni per poter raggiungere un livello di qualità elevato. Considerando le numerose ricerche che si stanno svolgendo negli ultimi anni, la cattura volumetrica potrà ottenere, probabilmente in un futuro non lontano, un fotorealismo molto buono. Fino a tale momento, gli studi si devono attenere ad una produzione meno realistica e pensare a utilizzi alternativi. La Rai presenta una distribuzione notevole di programmi televisivi in cui vengono sfruttati VFX e la virtual production (ad esempio Green Meteo e Clorophilla). Proprio in tali programmi, si potrebbe inserire in maniera ottimale una cattura volumetrica meno realistica. Tale scelta potrebbe generare nello spettatore di tali programmi, solitamente rientrante nella categoria di bambino, un effetto sorpresa e di stupore. L'utilizzo di una cattura volumetrica in questi contesti, inoltre, permetterebbe una notevole riduzione di costi in quanto non sarebbe più necessario modellare ed animare un personaggio 3D, operazione che richiede una quantità di tempo e denaro non

indifferente. Infatti, la cattura volumetrica richiederebbe una semplice registrazione della persona in live action e un successivo compositing nella scena, riducendo costi e tempistiche.

Inoltre, i programmi televisivi di performance di canto e ballo potrebbero essere resi maggiormente spettacolari utilizzando gruppi di ballo catturati volumetricamente. Tale esempio di utilizzo si è dimostrato decisamente efficace in molti casi di produzioni televisive mondiali, come è stato possibile notare negli esempi esposti in precedenza. Tale soluzione permetterebbe, oltre ad una riduzione dei costi, una sicurezza relativa alla bontà della performance, considerando che la cattura volumetrica verrebbe realizzata in momenti precedenti rispetto alla distribuzione del programma televisivo.

All'interno di produzioni televisivi più realistiche, le catture volumetriche potrebbero rendere più semplice la realizzazione di un programma televisivo in cui i vari presentatori televisivi siano situati in differenti luoghi del mondo. Infatti, ciò renderebbe possibile contenere i costi relativi agli spostamenti della troupe e dell'hardware. L'integrazione con gli ambienti virtuali è resa realistica dalla possibilità di effettuare il *relighting* sulle catture volumetriche, opportunità non esistente per le riprese bidimensionali effettuate su green screen.

Infine, molte tecnologie low budget presenti sul mercato permettono di ottenere catture volumetriche non propriamente tridimensionali, piuttosto rientranti nella categoria del 2.5 D. Tale caratteristica, in realtà, non è un problema per una produzione televisiva, in quanto i programmi televisivi difficilmente necessitano di effettuare una rotazione della camera di 360 gradi intorno al soggetto ripreso. Infatti, la caratteristica principale richiesta per tali scopi è la sensazione di tridimensionalità che una cattura volumetrica è in grado di trasmettere e non la vera e propria tridimensionalità.

Capitolo 2

Stato dell'arte

In questo capitolo si offre una panoramica sulla realizzazione di video volumetrici. In primo luogo, vengono analizzati i formati di rappresentazione più in utilizzo al giorno d'oggi. In seguito, si studiano le diverse fasi della pipeline per la produzione di video volumetrici e come ciascuna sia influenzata dal tipo di formato utilizzato. Infine, viene approfondita la fase d'acquisizione, analizzandone potenzialità e limiti. Vengono, inoltre, valutate varie soluzioni commerciali e open source, adottate dalle aziende del settore.

2.1 Formati

I video tradizionali presentano dei formati ormai consolidati e standardizzati. D'altra parte, i video volumetrici mostrano molteplici forme di rappresentazione in continua evoluzione. I formati dei video volumetrici possono essere suddivisi in due macrocategorie: rappresentazioni esplicite ed implicite. Le prime si basano su formati specifici e riconosciuti, mentre quelle implicite sfruttano le reti neurali per rappresentare i contenuti implicitamente [2].

I formati utilizzati per i video volumetrici sono ciò che guida la scelta e lo sviluppo delle diverse tecniche di elaborazione dei relativi dati.

2.1.1 Mesh

Una *mesh* è un oggetto caratterizzato da facce, spigoli e vertici (figura 2.1). Questa rappresentazione permette di ottenere, a livello visivo, un vero e proprio volume anche se, in realtà, l'oggetto all'interno risulta vuoto.

Il corrispettivo formato è ottimale per una rappresentazione dettagliata anche di oggetti più complessi. Questa caratteristica, però, può comportare diversi problemi di latenza in caso di trasmissione in tempo reale dei video volumetrici.

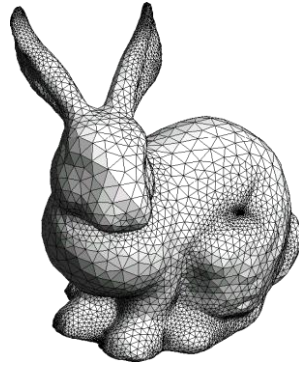


Figura 2.1: Esempio di mesh

Fonte: <https://is.gd/FvyV0V>

2.1.2 Nuvola di punti

La rappresentazione in questione permette di riprodurre un oggetto tramite molteplici punti disposti nello spazio (figura 2.2). Ogni punto non solo detiene informazioni spaziali ma, nella maggior parte dei casi, anche i dati relativi ai canali RGB. La nuvola di punti si ottiene tramite strumenti che siano in grado di acquisire, a partire da un oggetto reale o da una persona, le sue informazioni di colore e di profondità. I dispositivi, più utilizzati per ottenere questa tipologia di rappresentazione, sono le camere RGB-D, i laser oppure la tecnologia LiDAR.

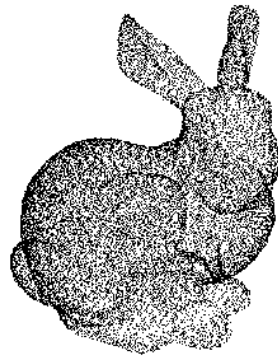


Figura 2.2: Esempio di nuvola di punti

Fonte: <https://is.gd/BRBNmu>

2.1.3 Voxel

Il *voxel* rappresenta il corrispettivo tridimensionale del pixel ed è utilizzato per i contenuti volumetrici (figura 2.3). L'unità di base è un cubo che può essere disposto nello spazio

tridimensionale. Questa rappresentazione, però, non permette di mantenere i dettagli e, al tempo stesso, può occupare molta memoria.

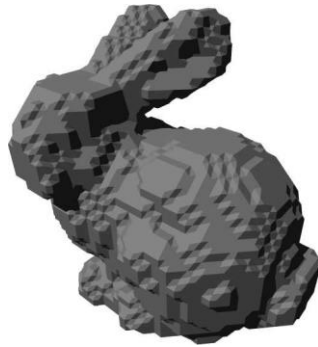


Figura 2.3: Esempio di voxel

Fonte: <https://is.gd/KeSoqC>

2.1.4 NeRF

Il *deep learning* ha permesso, negli ultimi anni, di ricostruire rappresentazioni volumetriche statiche a partire da un insieme di immagini. Queste ultime devono ricoprire, almeno parzialmente, le varie aree della scena di cui si vuole avere una ricostruzione volumetrica. Le porzioni della scena non visibili nelle immagini, invece, sono ricostruite dall'intelligenza artificiale che cerca di stimarne le viste sintetiche in base alle informazioni che possiede.

Il video introduce un'ulteriore dimensione e l'applicazione delle tecniche *NeRF* ad esso è risultata una sfida significativa. Il tempo è stato considerato come un input aggiuntivo che ha richiesto la suddivisione del processo di apprendimento della rete neurale in due fasi principali. In primo luogo, viene codificata la scena in uno spazio canonico e successivamente questa rappresentazione canonica viene mappata nella scena deformata in un istante temporale preciso [9]. Questo processo sfrutta una funzione 6D: $\Phi(x, y, z, \theta, \phi, t) = \sigma, c$ [2]. Quest'ultima permette alla rete neurale di mappare le coordinate in ingresso relative alla posizione nello spazio (x, y, z) , alla direzione di vista (θ, ϕ) e al tempo (t) in molteplici viste bidimensionali che dipendono dal punto di vista e che sono caratterizzate da densità di volume (σ) e colore (c) . Alcune tecniche NeRF permettono di ricostruire le scene dinamiche utilizzando un unico video, ottenuto da una sola camera che si muove nello spazio.

I NeRFs rendono possibile la ricostruzione di una geometria ad alta risoluzione e un rendering fotorealistico della scena da nuovi punti di vista (figura 2.4), ma richiedono risorse computazionali elevate.



Figura 2.4: Ricostruzione ottenuta con la tecnica NeRF da diversi punti di vista

Fonte: <https://is.gd/DzzK5N>

2.1.5 Gaussian Splatting

Un'alternativa ai NeRFs è una tecnica, sviluppata recentemente, chiamata *Gaussian Splatting*. Quest'ultima rende possibile incrementare la velocità di rendering ad un livello pari a quello di rendering in tempo reale. I dati in ingresso, richiesti da questo algoritmo, sono gli stessi dei NeRF. Infatti, il Gaussian Splatting richiede un insieme di immagini da cui verrà generata una nuvola di punti. Quest'ultima sarà, in un secondo momento, convertita in un insieme di gaussiane 3D. Questo spazio, costituito da molteplici gaussiane, sarà sottoposto a un processo di ottimizzazione che permetterà di ottenere una rappresentazione volumetrica più fedele possibile alla realtà (figura 2.5). Durante questa fase, si duplicano o si suddividono o si spostano le gaussiane 3D che sono state posizionate in modo errato [10] in modo da poter ottenere una ricostruzione più coerente e veritiera. Successivamente, le gaussiane 3D ottimizzate verranno proiettate su un piano 2D in fase di rendering.

I grandi vantaggi apportati da questa tecnica hanno spinto la ricerca ad estendere il Gaussian Splatting per la ricostruzione volumetrica di scene dinamiche. In questo caso, le gaussiane 3D verranno spostate nello spazio e modificate nella loro forma per ogni lasso temporale considerato [11].



Figura 2.5: Due ricostruzioni ottenute con il Gaussian Splatting. A sinistra la gaussiane sono completamente opache.

A destra ogni gaussiane ha il proprio valore di trasparenza.

Fonte: <https://is.gd/dN9BEU>

2.2 Panoramica sulla pipeline del video volumetrico

In questo paragrafo viene introdotta una panoramica sull'intera pipeline del video volumetrico, i cui principali passaggi sono cinque: cattura, compressione, trasmissione, rendering e visualizzazione. Ogni stadio presenta caratteristiche derivanti dalla pipeline dei video tradizionali ma anche molte innovazioni, studiate appositamente per questo ambito emergente. Il video volumetrico è un campo ancora poco esplorato, considerando la sua nascita molto recente. Infatti, si presume che nei prossimi anni ci saranno molteplici avanzamenti che permetteranno di rendere più efficiente l'intera pipeline di produzione. Per tale motivo, ogni stadio presenta grandi possibilità di ricerca e innovazione.

2.2.1 Cattura

La cattura è il primo processo della pipeline in questione e deve essere svolta correttamente per riuscire ad ottenere un video volumetrico di qualità in fase finale. L'acquisizione risulta differente rispetto alla corrispettiva dei video bidimensionali, in quanto l'output richiesto non è più rappresentato da una semplice matrice di pixels [2]. I dati ottenuti in fase di cattura, infatti, non contengono unicamente informazioni di colore, ma presentano anche un'indicazione sulla profondità. Quest'ultima informazione è fondamentale per la ricostruzione tridimensionale della scena catturata. Al fine di ottenere questi dati, nella maggior parte dei casi sono necessari dispositivi più strutturati e complessi di quelli utilizzati per la cattura bidimensionale. Negli ultimi anni, l'intelligenza artificiale sta iniziando a rendere più accessibile questa tecnologia permettendo di realizzare video volumetrici sfruttando strumenti di cattura meno sofisticati e più economici.

La cattura volumetrica permette, a prescindere dalla tecnica utilizzata, di ottenere le informazioni sulla topologia di un oggetto o di una persona, ma non sulla loro struttura interiore.

2.2.2 Compressione

Il passaggio successivo è rappresentato dalla compressione dei dati ottenuti in fase di cattura. L'output ottenuto è rappresentato da informazioni più strutturate e complesse dei corrispettivi video bidimensionali e, per tale ragione, si necessita di uno studio

approfondito delle possibili tecniche di compressione applicabili a tali dati. Infatti, non risulta possibile applicare direttamente gli standard di compressione dei video tradizionali.

I metodi di compressione dei video volumetrici possono essere categorizzati in base alla rappresentazione dei dati tridimensionali ottenuti. In primo luogo, si può individuare la compressione utilizzata per le nuvole di punti che prevede la codifica basata sulla loro trasformata oppure la codifica basata sulla loro predizione temporale.

La compressione applicata alle mesh, invece, si basa sulla quantizzazione e su una codifica entropica dell'informazione.

I NeRFs, considerando che sono rappresentati da modelli di rete neurale, utilizzano diversi metodi ispirati alle tecniche di *Model Compression* [2]. Queste ultime risultano essere tecniche estremamente utilizzate per ridurre le dimensioni dei modelli di deep learning e il loro spazio di archiviazione [12].

Per quanto riguarda il Gaussian Splatting, la maggior parte dello spazio di archiviazione è occupato dai parametri gaussiani e spesso risultano essere ridondanti [13]. Per tale motivo, sono state sviluppate varie tecniche che permettono di comprimerli. Indipendentemente dall'output ottenuto, la compressione dei dati volumetrici è fondamentale. Infatti, questa tipologia di dati contiene spesso informazioni ridondanti e lo spazio occupato è decisamente oneroso, soprattutto se confrontato con le dimensioni dei file dei classici video bidimensionali.

2.2.3 Trasmissione

La trasmissione è ciò che permette all'utente finale di fruire dei contenuti volumetrici a partire da una previa cattura e compressione. I dati volumetrici, contenendo molteplici informazioni, sono molto pesanti e di grandi dimensioni. Per tale motivo, la loro trasmissione è una sfida significativa a cui si è cercato di rispondere tramite diverse tecniche.

Alcune tecniche di trasmissione utilizzate permettono di ridurre la quantità di informazione trasmessa basandosi sul fatto che solamente una porzione di video volumetrico verrà effettivamente visualizzata dall'utente. Questo comportamento si può attribuire alla limitata visibilità ottenibile con gli strumenti di visualizzazione più

utilizzati, ovvero gli schermi tradizionali. In questo modo, il sistema permette di ridurre la qualità del restante volume che non ricade del frustum visivo dell'utente.

Ulteriori metodi di trasmissione si basano sulla possibilità di codificare il video a diversi livelli di qualità. Le condizioni di rete disponibili orienteranno la scelta di trasmissione ad un livello di qualità più alto oppure più basso. Questa tecnica può permettere di ottenere miglioramenti significativi in termini di prestazione di trasmissione.

Le tecniche di trasmissione basate sulla segmentazione semantica e sul riconoscimento degli oggetti, invece, vengono utilizzate per l'estrazione delle informazioni semantiche a partire da un video. Queste ultime permettono di conferire maggiore importanza alle porzioni della scena più rilevanti e, dunque, concentrarsi prevalentemente sulla trasmissione di esse [2].

2.2.4 Rendering

Il processo di rendering è ciò che permette di trasformare i dati geometrici, a partire da un modello 3D o da una scena, in un'immagine o in un video. Il processo in sé influenza notevolmente l'esperienza complessiva e, per tale motivo, è fondamentale cercare un approccio che possa migliorarla.

Uno dei metodi più tradizionali consiste nel considerare ogni singolo punto della nuvola di punti come se fosse un pixel. Un ulteriore approccio permette, invece, di connettere i vari punti di una mesh fino a formare un triangolo, il quale verrà successivamente sottoposto a una fase di rasterizzazione su una superficie bidimensionale. Un'evoluzione di tali tecniche consiste nello sfruttare le reti neurali per calcolare il valore e il gradiente della funzione lungo ogni punto del raggio emesso a partire dalla posizione della camera. Ciò permette di aumentare l'immersività della visione anche grazie a un rendering con illuminazione e riflessioni più realistiche [2]. Tuttavia, questo metodo presenta ancora una latenza non indifferente. Un aumento di velocità nella fase di rendering è stato introdotto da una nuova tecnica utilizzata per il Gaussian Splatting. Quest'ultima si basa su un algoritmo che permette di proiettare direttamente sul piano 2D le gaussiane 3D, considerando il contributo di ogni gaussiana per ogni pixel (figura 2.6). In questo modo, si riesce ad ottenere un rendering in tempo reale anche ad alte risoluzioni [11].

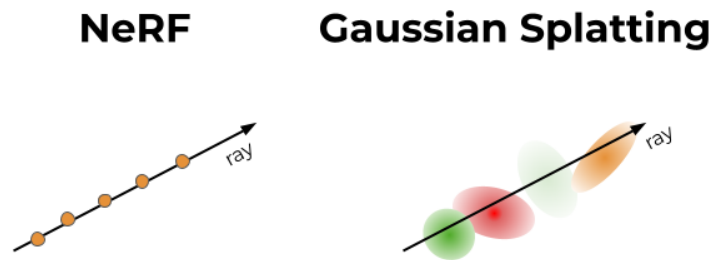


Figura 2.6: Confronto del processo di rendering del NeRF e del Gaussian Splatting

Fonte: <https://is.gd/1TZiex>

2.2.5 Visualizzazione

La visualizzazione dei video volumetrici può avvenire sia su schermi tradizionali sia su HMD (Head Mounted Display) (figura 2.7). Nel primo caso, l'utente si può muovere all'interno del video volumetrico mantenendo una visuale limitata dallo schermo bidimensionale. L'utilizzo degli HMD, invece, garantisce una notevole differenza rispetto alla visualizzazione dei video bidimensionali. Questi strumenti sfruttano la visione stereoscopica, la quale permette di fondere le due immagini dei due occhi in una unica. Grazie a questi dispositivi, l'utente ha la possibilità di immergersi nella scena e, in alcuni casi, di navigare in essa. Queste caratteristiche permettono di superare la limitata visibilità e mobilità dei video tradizionali.



Figura 2.7: Esempio di Head Mounted Display (HMD)

Fonte: <https://is.gd/Wx7b4u>

2.3 Tecniche per la cattura volumetrica

La cattura è il primo passo per la realizzazione di un video volumetrico. Questa fase è una delle più cruciali e deve essere accuratamente studiata. L'analisi di questo processo è fondamentale per comprendere quali siano le diverse possibilità che si possono intraprendere e i relativi vantaggi e svantaggi. Dai primi tentativi, agli inizi degli anni duemila, fino a oggi, diverse tecniche e metodologie si sono evolute, combinate e affinate con l'obiettivo di restituire le migliori catture volumetriche possibili. I recenti progressi nella fotografia computazionale hanno permesso di migliorare e agevolare il processo di realizzazione di video volumetrici. Sono nati nuovi efficienti metodi di cattura, interpolazione e organizzazione in strutture dati coerenti.

Le metodologie di cattura sono varie. Una prima suddivisione si può operare considerando il numero di camere impiegate: si può scegliere di utilizzare array di camere calibrate oppure una singola camera.

2.3.1 Array di camere calibrate

Una matrice di camere viene disposta intorno al soggetto da catturare (figura 2.8). I vari punti di vista catturati devono essere poi combinati per fornire un'informazione soddisfacente del soggetto nella sua totalità.



Figura 2.8: Array di camere

Fonte: <https://is.gd/8zhtVO>

È quindi indispensabile calibrare e sincronizzare le camere prima di iniziare il processo di cattura. La calibrazione può essere eseguita utilizzando dei *marker*, che vengono sfruttati per l'allineamento spaziale delle camere. È questo il metodo che viene solitamente utilizzato, ma è possibile sfruttare approcci non basati sui marker [2]. In

questo caso, durante la fase di calibrazione viene posizionata al centro della stanza una struttura fisica, ad esempio delle scatole di cartone. Viene poi stimata la posizione delle camere in funzione delle coordinate di quelle della struttura. Le posizioni e l'orientamento delle camere devono rimanere fissi per tutta la durata della cattura, in modo da unire sempre correttamente i dati *raw* delle camere calibrate. Ottenere dati allineati nello spazio non è però sufficiente, ma devono essere anche sincronizzati temporalmente. La sincronizzazione avviene sia a livello hardware che software. Le camere vengono collegate via cavo le une alle altre e viene stabilito quale assumerà il ruolo di *master*, mentre le altre saranno gli *slaves*. Il master invia il segnale di clock agli slaves, che vengono così sincronizzati. La sincronizzazione software è necessaria per garantire che il server e i pc collegati alle camere siano sincronizzati gli uni con gli altri. Il PTP (Precision Time Protocol) è il protocollo che, in genere, viene sfruttato per la sincronizzazione dei dispositivi [2]. La calibrazione e la sincronizzazione garantiscono che i frame possano essere uniti in un unico e corretto video volumetrico.

In questa tipologia di cattura possono essere usate diverse tipi di tecnologie e dispositivi, come camere RGB, RGB-D, infrarossi e LiDAR.

Le camere RGB (figura 2.9) catturano solo l'informazione di colore e sfruttano algoritmi avanzati di computer vision e machine learning per combinare i dati delle immagini, ottenute da diversi punti di vista.



Figura 2.9: Camera RGB per la volumetric capture dell'IO Industries

Fonte: <https://is.gd/CcqcAP>

I dispositivi RGB-D (figura 2.10) uniscono, frame per frame, le informazioni di colore a quelle di profondità. Il calcolo di quest'ultima viene stimato da appositi sensori, che restituiscono una mappa di profondità [14].



Figura 2.10: Intel® RealSense™ Depth Camera D455

Fonte: <https://is.gd/I6bu2d>

Le informazioni di profondità e di colore (figura 2.11) devono essere allineate, affinché la ricostruzione avvenga in modo coerente.

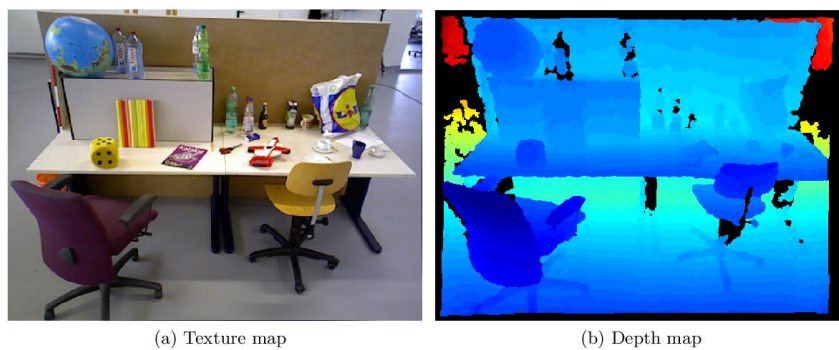


Figura 2.11: A sinistra mappa della texture. A destra mappa di profondità

Fonte: <https://is.gd/DvtYXc>

In primo luogo, vengono processati i dati per rimuovere il rumore e correggere le distorsioni. In seguito, vengono allineati grazie alle informazioni ottenute nella calibrazione, che restituisce i parametri intrinseci ed estrinseci della camera. Il sistema di coordinate della mappa di profondità viene convertito in quello del colore e i valori di profondità sono assegnati al pixel corrispondente nell'immagine. Il risultato sono immagini RGB-D, che contengono informazioni sul colore e sulla profondità pixel per pixel e che vengono sfruttate per la ricostruzione 3D della scena. Ciascuna immagine, però, ha un campo visivo limitato e rivela solo una porzione dell'ambiente o del soggetto; è quindi necessario unirle per ottenere un risultato completo. Anche in questo caso, sono d'aiuto i parametri ottenuti con la calibrazione, che permettono di stabilire un sistema di coordinate globale per tutte le sotto-scene. Queste ultime sono infine unite e si ottiene una ricostruzione completa [2].

L'uso di camere RGB-D semplifica e velocizza il processo di ricostruzione tridimensionale. Le informazioni di profondità ottenute permettono di ridurre il bisogno di complessi algoritmi, al contrario fondamentali nella cattura RGB. L'acquisizione RGB-

D permette inoltre di ridurre il numero di camere utilizzate e ottenere comunque un risultato soddisfacente e coerente. Al contrario, la cattura RGB richiede, in genere, un numero superiore di camere, a parità di risultato.

Un video volumetrico può essere realizzato anche sfruttando la tecnologia LiDAR (figura 2.12). Il LiDAR è un sensore attivo che consente di riflettere le caratteristiche di uno scenario e ottenerne un modello tridimensionale. Il tempo di volo (TOF) è usato per l'elaborazione di una mappa di distanza degli oggetti nella scena [15]. In genere, gli scanner LiDAR vengono usati insieme a camere RGB tradizionali per unire i dati di profondità e le informazioni di colore.

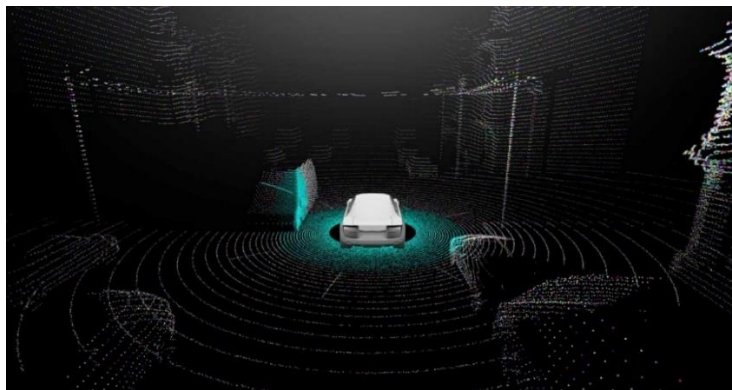


Figura 2.12: Tecnologia LiDAR

Fonte: <https://is.gd/e0DpWa>

L'approccio di cattura con array calibrati di camere permette di ottenere ottimi risultati, ovvero dei video volumetrici estremamente realistici. Il costo dipende dal tipo e dal numero di camere utilizzate e questo tipo di setup è, in genere, complesso e dispendioso. Per tale motivo, viene utilizzato da grandi produzioni, che hanno a disposizione budget elevati. Tuttavia, i recenti sviluppi nel deep learning e nella computer vision stanno rendendo sempre più accessibile la realizzazione di video volumetrici per piccole produzioni e prosumer, con a disposizione budget limitati. Infatti, è oggi possibile realizzare video volumetrici anche con una singola camera RGB.

2.3.2 Camera singola

Ci sono diversi metodi per realizzare un video volumetrico con un'unica camera; due tra le più importanti tecniche sono la *Structure From Motion (SfM)* e la *Single-View Depth Estimation* [2].

La Structure from Motion (SfM) permette di ottenere un modello tridimensionale di una scena o di un oggetto, a partire da una serie di immagini bidimensionali catturate da diversi punti di vista [16]. Si basa sui principi della fotogrammetria per calcolare le posizioni 3D (x,y,z) a partire da due o più immagini. La SfM include diversi step, tra cui l'estrazione e il matching di feature, stima delle pose delle camere, la triangolazione e la compensazione a stelle proiettive [2] (figura 2.13). L'estrazione delle features comporta l'individuazione di punti ad alta frequenza nelle immagini, punti delle immagini in cui il colore varia rapidamente, ad esempio spigoli e bordi. Viene poi eseguito il matching: vengono cercate nelle diverse immagini le features precedentemente identificate, che corrispondono allo stesso punto tridimensionale nello spazio. La stima delle pose delle camere consente di stabilire la posizione e l'orientamento della camera di ogni immagine in relazione alla scena o all'oggetto. A partire dal matching e dalla stima delle pose delle camere è possibile calcolare la posizione 3D di ciascuna feature: si intersecano i raggi di ciascuna camera che passano per la feature e ne si ottiene così la localizzazione nello spazio [17]. Infine, la compensazione a stelle proiettive (*bundle adjustment*) è una procedura che permette di affinare i parametri delle camere e la posizione delle features. Si cerca di minimizzare l'errore di proiezione, l'errore geometrico che misura la differenza tra le posizioni osservate e quella previste delle features [18].

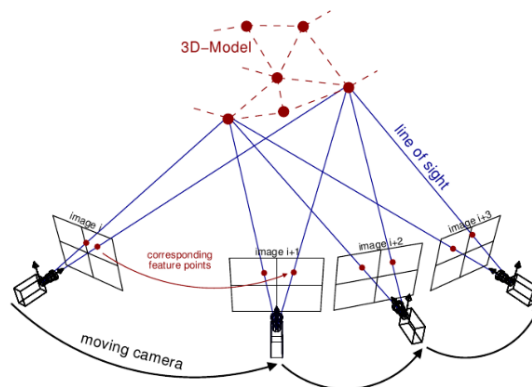


Figura 2.13: Schema Structure From Motion

Fonte: <https://is.gd/LK4hSe>

La Single-Image Depth Estimation (SIDE) calcola la profondità di una scena, a partire da una singola immagine [19]. Per ottenere un video volumetrico con questa tecnica la scena viene ripresa da uno o più punti di vista e, per ciascun frame, viene elaborata una densa mappa di profondità. Le *CNN* (*Convolutional Neural Networks*) sono lo standard per

questo processo: sono infatti in grado di ricavare informazioni di profondità della scena servendosi di una singola immagine RGB [20]. Grazie alle informazioni di profondità e ai parametri intrinseci della camera, è possibile generare una nuvola di punti 3D. La nuvola di punti è successivamente processata con algoritmi di ricostruzione delle superfici che permettono di ottenere una mesh 3D. Infine, il modello viene texturizzato sfruttando tecniche di texture mapping.

I NeRFs e il Gaussian Splatting sono in grado di interpretare la tridimensionalità della scena a partire da un dataset di immagini; nel Gaussian Splatting le camere vengono calibrate con l'SfM e l'insieme di gaussiane viene inizializzato con la nuvola di punti generata dal processo SfM [10]. I recenti progressi nel deep learning, il potenziamento dei motori grafici, gli algoritmi di ottimizzazione e nuove tecniche di rendering hanno contribuito alla nascita e alla diffusione di questi nuovi formati di rappresentazione, democratizzando e rendendo sempre più accessibile la realizzazione di video di volumetrici. I modelli di deep learning si dimostrano inoltre in grado di estrarre informazioni geometriche da semplici immagini anche in situazioni complesse. In aggiunta, la stima della profondità non viene più eseguita a livello hardware, con appositi strumenti costosi, ma può essere demandata a una fase successiva. Questo permette di ridurre ulteriormente la complessità e il costo della fase di cattura.

2.4 Casi studio

I video volumetrici stanno assumendo un ruolo sempre più rilevante nel panorama della realtà virtuale e aumentata, nonché nelle produzioni cinematografiche e televisive. Diverse importanti realtà, come Microsoft e Intel, hanno deciso di investire nella loro realizzazione. Negli ultimi anni, sia a livello europeo che mondiale, gli studi di cattura volumetrica hanno visto una crescita significativa. I setup sono, nella maggior parte dei casi, costosi e le strutture complesse: vengono usate array di decine o centinaia di camere calibrate in specifici ambienti opportunamente allestiti e illuminati. Qui di seguito vengono riportate alcune delle più importanti realtà che si occupano di cattura volumetrica e le tecnologie che utilizzano.

Microsoft Mixed Reality Studios

Gli studi di Mixed Reality di Microsoft (figura 2.14) nascono nel 2010 a Washington e ora hanno sede a San Francisco. Lo studio più importante ha 106 camere a infrarossi e RGB, grazie alle quali avviene la cattura di video volumetrici estremamente realistici. Sono inoltre stati introdotti due stage mobili, con un numero minore di camere (64), leggeri e facilmente trasportabili. Azure, la piattaforma di cloud computing sviluppata da Microsoft, ha permesso di processare un numero maggiore di dati e raggiungere una velocità di streaming pari a quella di piattaforme leader del settore, come Netflix [21]. Sono stati realizzati progetti importanti, tra cui la performance di Madonna nel 2019 ai Billboard Music Awards, e diversi rilevanti progetti educativi e formativi.



Figura 2.14: Microsoft Mixed Reality Capture Studio

Fonte: <https://is.gd/agyQ87>

Intel Studios

Lo studio di cattura volumetrica di Intel è una cupola geodetica di circa 930 m² (figura 2.15) e si trova a Los Angeles. Si attesta il primato del più grande hub multimediale immersivo al mondo. Vengono usate 96 camere 5K per catturare l'azione in scena e complessi algoritmi convertono trilioni di pixel in un ambiente tridimensionale [22]. Ogni camera è collegata a una batteria dei server Intel posti in loco. Per la connessione viene sfruttata una rete di circa otto chilometri di cavi a fibra ottica e si ottiene una velocità di trasmissione dell'ordine dei terabytes al minuto. I dati da elaborare dai server sono così numerosi che è stato necessario allestire una stanza isolata perché il rumore non interferisse con la registrazione dell'audio in studio [23]. Qui sono stati filmati

contemporaneamente venti attori e ballerini per una performance di “Grease”. Questo progetto è il frutto di una collaborazione con Paramount.

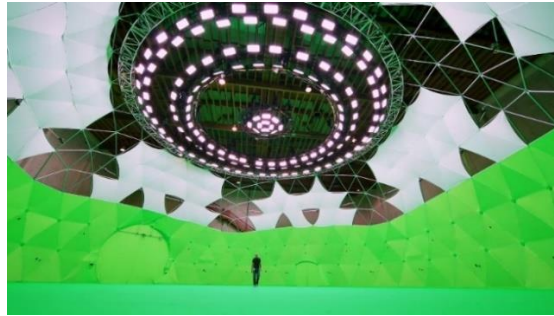


Figura 2.15: Intel Studios

Fonte: <https://is.gd/dNhEPV>

Infinite Realities

Questa azienda propone un nuovo sistema di cattura volumetrica con l'obiettivo di ottenere acquisizioni volumetriche ad alta qualità [24]. L'innovazione, apportata da questa realtà, consiste nel processamento e nella distribuzione di contenuti ottenuti tramite l'utilizzo del Gaussian Splatting 3D e 4D. A tale scopo, nel gennaio del 2024 Infinite Realities ha acquisito una licenza commerciale *3DGS* (Gaussian Splatting 3D) completa con Inria per consentire l'offerta di servizi basati sul *4DGS* (Gaussian Splatting 4D) a clienti in tutto il mondo. Successivamente, l'azienda ha trascorso gli ultimi 10 mesi a sviluppare una pipeline software 4DGS personalizzata, utilizzando il loro cluster GPU per elaborare rapidamente centinaia di migliaia di frame 4DGS. Tutto ciò è stato anche possibile grazie ai loro sistemi di acquisizione volumetrica all'avanguardia, basati su 176 camere sincronizzate e su uno spazio illuminato correttamente.

Infinite Realities afferma di poter catturare qualsiasi cosa, dalle interazioni complesse di persone per scene contenenti effetti visivi, a scene dinamiche di combattimenti veloci, sport, recitazione, e persino catturare e archiviare ricordi di familiari per il futuro (figura 2.16).



Figura 2.16: Cattura volumetrica realizzata da Infinite Realities studio

Fonte: <https://is.gd/UmB8eE>

4DViews

4DViews è un'azienda, con sede in Francia, specializzata nell'acquisizione ed elaborazione di video volumetrici. La realtà non si concentra solo sulla realizzazione di catture volumetriche realistiche, ma offre pacchetti preconfezionati a studi creativi e professionali. È stato sviluppato un formato di video volumetrico compatibile con diverse piattaforme hardware e software. 4DViews offre una tecnologia proprietaria per l'acquisizione (4DCapture) e un sistema di acquisizione di video volumetrici (Holosys). Holosys (figura 2.17) è un sistema che fornisce hardware e software necessari per una cattura volumetrica totalmente trasportabile, che consente un veloce allestimento per le catture volumetriche in loco. Il volume di cattura è di 5 metri in larghezza e 2,4 in altezza, il frame rate può raggiungere i 60 fps e la risoluzione delle texture è a 5k [25]. Il formato di output è un formato proprietario. La capacità di registrazione arriva fino a 110 minuti. È infine possibile l'integrazione dei video volumetrici su Unity e Unreal Engine tramite appositi plugins.



Figura 2.17: Holosys by 4DViews

Fonte: <https://is.gd/tjOA8s>

NHK STRL Meta Studio

Meta Studio (figura 2.18), sviluppato presso NKH STRL, è uno studio in grado di fornire contenuti volumetrici e potenziali flussi di lavoro per la produzione di contenuti 2D, 3D e AR/VR, soprattutto per il broadcasting.

Nella produzione di programmi televisivi convenzionali, artisti reali recitano su un set in un vero studio illuminato da luci reali e le riprese vengono eseguite utilizzando una vera telecamera. Esistono, in tali situazioni, limiti alle modifiche che possono essere gestite in post-produzione e, in alcuni casi, potrebbe essere necessario ripetere le riprese. La cattura volumetrica, invece, permette, una volta terminato il lavoro di acquisizione, di annullare le modifiche, eseguite al computer, un numero qualsiasi di volte.

Nella produzione di Meta Studio, si ritiene che la luce del soggetto fotografico possa essere registrata così com'è durante le riprese e possa essere successivamente modificata arbitrariamente. La libertà, resa possibile sia nel lavoro con l'illuminazione che con la macchina da presa, espande notevolmente le possibilità nella produzione. Infatti, la libertà nel posizionamento della camera da presa può anche permettere di ottenere video da angolature e posizioni in cui una camera reale non potrebbe essere posizionata.

Al fine di ottenere tali risultati, lo studio utilizza 24 camere 4K RGB e sensori inseriti in una cupola emisferica con un diametro di 8 metri e un'altezza di 5 metri. Le camere permettono una ricostruzione tridimensionale del soggetto, ovvero si ottiene una nuvola di punti che, però, contiene errori di misura. Questi ultimi vengono compensati durante il processo di ricostruzione tramite una *Deep Neural Network (DNN)* [26].

L'utilizzo di più telecamere, comune in molti studi di cattura volumetrica, aumenterebbe i costi computazionali e il tempo di elaborazione, diventando un problema per la ricostruzione in tempo reale. Meta Studio, per tale motivo, ha scelto di ridurre il numero di telecamere assicurando al tempo stesso una qualità sufficiente della ricostruzione per l'utilizzo dei programmi televisivi.

Inoltre, le camere con i sensori di profondità hanno spesso una risoluzione minore rispetto a quelle RGB e, per tale ragione, Meta Studio ha optato per la scelta di camere senza sensore di profondità per l'acquisizione dei suoi dati [27].



Figura 2.18: Esterno ed interno di NKH Meta Studio

Fonte: <https://is.gd/vBapfT>

Function4D

Function4D è un sistema di cattura volumetrica in real time che utilizza un numero ridotto di sensori RGBD consumer (fino a tre) [28]. La sfida è ottenere ottime catture volumetriche in scenari complessi e dinamici, senza la necessità di setup ad alta tecnologia e complicate metodologie d'acquisizione. Il sistema, a differenza di altri, nonostante la semplicità, è in grado di gestire situazioni difficili, come l'interazione di una persona con oggetti, il cambiamento di vestiti, movimenti veloci e interazioni tra più persone.

Function4D sfrutta la fusione volumetrica e usa delle *deep implicit functions (DIF)* per la ricostruzione volumetrica. La fusione volumetrica unisce le informazioni di profondità raccolte da diverse angolazioni in un'unica rappresentazione 3D coerente [29], mentre le DIF forniscono una rappresentazione 3D che decompone lo spazio in un insieme strutturato di funzioni implicite apprese [30].

Per generare output completi e consistenti nel tempo, gli attuali metodi di fusione volumetrica uniscono il maggior numero possibile di dati di profondità nel tempo. Ciò comporta una forte dipendenza dal tracking, che però può non essere accurato quando ci sono notevoli cambi nella topologia e oclusioni. Al contrario, le *deep implicit functions (DIF)* sono ottime per completare le superfici, ma, a causa del numero ridotto di informazioni di profondità e del rumore elevato nei sensori RGBD consumer, non sono in grado di ricavare informazioni dettagliate e continue nel tempo [28]. Function4D

combina la fusione volumetrica e l'uso di deep implicit functions (DIF), perché, queste due tecniche, unite, sono in grado di compensarsi e migliorare il risultato finale.

Una volta sincronizzati gli input RGBD, viene sfruttata la *dynamic sliding fusion (DSF)* (figura 2.19), che permette di unire frames vicini per ottenere un risultato senza rumore e continuo nel tempo. La pipeline di fusione volumetrica viene ridisegnata e permette di ottenere risultati con rumore minimizzato, topologia consistente e informazioni temporali continue. A partire dai risultati ottenuti con la DSF, vengono usate le deep implicit functions per eliminare le dipendenze dal tracking di lunghi periodi di tempo. Le deep implicit functions sono utili per preservare alcuni dettagli geometrici e i risultati di texturizzazione.

Per ottenere risultati soddisfacenti in tempo reale vengono messi in atto specifici accorgimenti, che permettono di rendere Function4D più veloce di ordini di grandezza rispetto agli altri approcci.

I risultati ottenuti dimostrano che Function4D supera i metodi esistenti in termini di visualizzazione, capacità di generalizzazione, qualità della ricostruzione ed efficienza di esecuzione. Sebbene Function4D fornisca ottime e dettagliate ricostruzioni per le zone visibili, la generazione di superfici e di texture accurate per regioni completamente occluse rimane una sfida. Il lavoro futuro potrebbe ampliare l'uso delle deep implicit functions per includere osservazioni temporali e integrare le informazioni RGB per migliorare la ricostruzione geometrica in presenza di materiali specifici, che i sensori di profondità fanno fatica a rilevare.

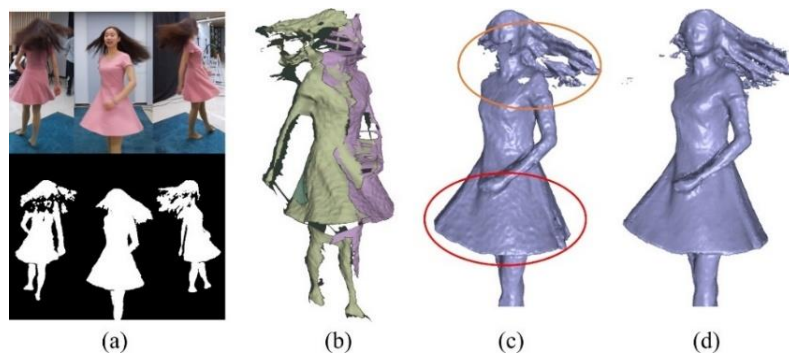


Figura 2.19: Valutazione della fusione scorrevole dinamica.

Fonte: <https://is.gd/Jflm7K>

Contesto economico

Gli esempi esposti mostrano gli studi più importanti a livello mondiale e una realtà minore, occupati nella realizzazione di video volumetrici. È importante evidenziare l'importanza del mercato dei video volumetrici, che è in continua espansione, soprattutto negli ultimi anni. Infatti, la dimensione del mercato del video volumetrico era di 2,0 miliardi nel 2022 e si stima che raggiungerà 7,6 miliardi nel 2028, con una CAGR (tasso annuo di crescita composto) del 28,6% [31]. Inoltre, è cruciale sottolineare come i recenti avanzamenti tecnologici stiano contribuendo a rendere la cattura volumetrica una tecnologia sempre più accessibile come è stato possibile riscontrare nell'esempio, citato precedentemente, di Fuction4D. Ulteriori aziende (come DepthKit, Volograms e Velox) stanno contribuendo alla democratizzazione della cattura volumetrica.

Capitolo 3

Tecnologie utilizzate

In questo capitolo si offre una panoramica sulle soluzioni tecnologiche utilizzate per la realizzazione dei video volumetrici. Per ogni tecnologia, viene analizzata la storia della relativa azienda e i costi di utilizzo. Successivamente, si presenta una descrizione dettagliata della soluzione tecnologica da un punto di vista tecnico. In tale sezione, infatti, vengono analizzati i formati di output, la pipeline e le limitazioni a cui prestare attenzione durante la fase di cattura.

3.1 Volu

3.1.1 Storia azienda e costi

Volograms è una compagnia di AI, con sede principale a Dublino, impegnata nella cattura volumetrica di persone, soprattutto a livello consumer. Infatti, il loro obiettivo consiste nel permettere a chiunque la creazione di contenuti per il VR e l'AR e renderlo semplice così come registrare un contenuto video bidimensionale [32]. L'azienda ha ottenuto, nel 2022, l'Auggie Awards per il miglior utilizzo dell'intelligenza artificiale. Tale premio è uno dei più importanti riconoscimenti nel campo della realtà aumentata e virtuale.

L'applicazione Volu, sviluppata da Volograms, è scaricabile su dispositivi mobile o tablet Apple con il sistema operativo iOS 14 o iOS 15. Per la generazione di modelli Vologram è inoltre necessaria la compatibilità con ARKit. L'applicazione è disponibile anche per i device realizzati da un produttore di dispositivi Android che supportano ARCore. L'applicazione in questione, dunque, permette di registrare contenuti volumetrici senza la necessità di setup professionali.

L'azienda mette a disposizione una versione di prova gratuita che permette, però, una registrazione limitata a cinque secondi. Video volumetrici con una durata maggiore possono essere acquistati a prezzi che variano in base ai secondi di video scelti. Infatti, è

possibile registrare un video volumetrico di 30 secondi ad un costo di 89 euro, un video di 60 secondi a 174 euro mentre 120 secondi di cattura volumetrica costano 329 euro. Un'ulteriore possibilità consiste nel sottoscrivere un abbonamento mensile che dà al cliente una maggiore libertà di registrazione e condivisione dei video volumetrici.

La tecnologia in questione, portata avanti da Volograms, ha elaborato più di 3 milioni di modelli 3D e permette, inoltre, le comunicazioni AR per tutti i tipi di aziende e professionisti. Infatti, i contenuti volumetrici di Volograms hanno anche alimentato campagne per aziende come HUGO BOSS, Mars Wrigley e Vodafone (figura 3.1).



Figura 3.1: Cattura volumetrica ottenuta con l'app Volu per Hugo Boss

Fonte: <https://is.gd/GZEWwi>

Inoltre, Volograms è stata chiamata per aiutare a creare ambientazioni immersive in contesti culturali, come per la National Gallery di Londra (figura 3.2) o per altre applicazioni di realtà aumentata all'interno dei musei.



Figura 3.2: Cattura volumetrica ottenuta con l'app Volu per la National Gallery di Londra

Fonte: <https://is.gd/XOcg0M>

Ulteriori applicazioni interessanti per questa tecnologia sono state individuate nelle produzioni broadcast. In particolare, sono state effettuate acquisizioni di squadre sportive

e poi distribuite alle televisioni (figura 3.3). La tecnologia AI di Volograms ha consentito loro di ottenere modelli 3D e persino sequenze video volumetriche, senza la necessità di portare tutti i giocatori in uno studio di acquisizione volumetrica. Questo utilizzo ha riscontrato negli spettatori un aumento di realismo e immersività nella scena grazie alla possibilità di illuminare il soggetto successivamente e di avere ombre dinamiche, portando l'esperienza broadcast ad un altro livello.

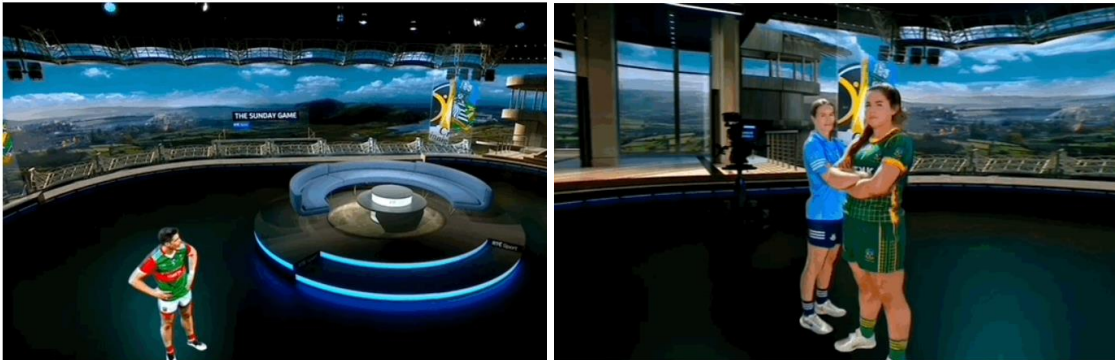


Figura 3.3: Utilizzi di catture volumetriche ottenute con l'app Volu nello studio televisivo Fox Sports

Fonte: <https://is.gd/UclyCB>

3.1.2 Funzionamento tecnologia

Applicazione

L'applicazione presenta tre pagine principali (figura 3.4). La schermata iniziale si presenta simile a una schermata di un social media in quanto l'utente ha la possibilità di vedere i contenuti volumetrici in tendenza al momento e trovare ispirazione per le sue acquisizioni. La schermata successiva permette all'utente di realizzare le proprie catture volumetriche e vederne il risultato. Inoltre, in questa sezione si ha la possibilità di selezionare un Vologram, catturato precedentemente, e inserirlo nell'ambiente reale circostante. Questa possibilità è resa disponibile grazie alla scansione di una superficie dell'ambiente circostante inquadrato dallo smartphone. È inoltre possibile applicare alcuni effetti alla cattura volumetrica e rendere il Vologram più creativo. L'utente può, dunque, registrare la propria cattura volumetrica ed inserirla in qualunque ambiente. I risultati di questa combinazione saranno visibili in una schermata dedicata e potranno essere condivisi con amici e conoscenti.

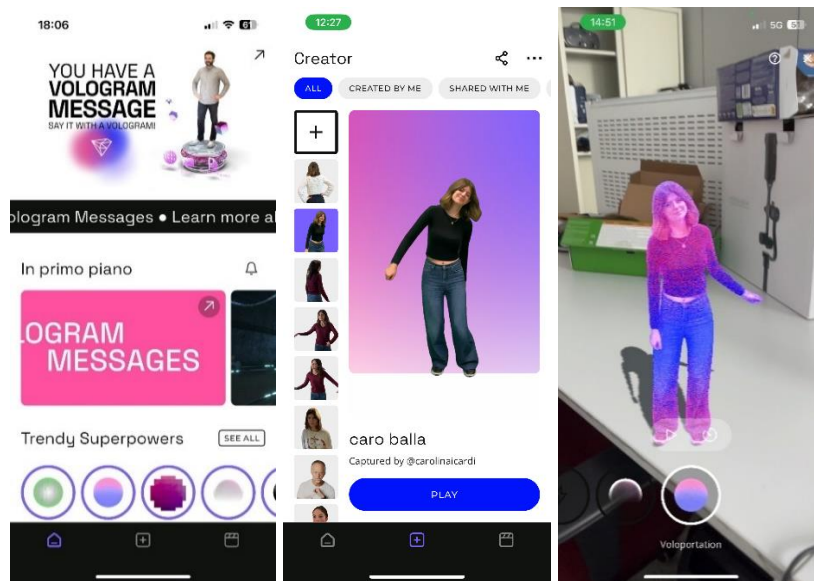


Figura 3.4: Le tre diverse schermate dell'app Volu

Limitazioni cattura

L'app Volu ha alcune limitazioni e requisiti da rispettare per ottenere un risultato volumetrico soddisfacente. In primo luogo, si può eseguire la cattura di una sola persona alla volta e il suo corpo deve essere completamente inquadrato. Lo sfondo deve essere statico e, all'interno della scena inquadrata, non devono comparire altri oggetti in movimento o animali. Inoltre, lo spazio deve essere ben illuminato, ma non vi è la necessità di utilizzare un green screen. Infatti, viene solo richiesto che lo sfondo non sia dello stesso colore dei vestiti o della pelle della persona inquadrata. Per quanto riguarda il soggetto, i movimenti non devono essere troppo veloci e deve essere posizionato frontalmente alla telecamera del telefono con cui si esegue la cattura. Nonostante possano essere seguiti tutti questi consigli, l'app non permette di ricostruire tutti dettagli del corpo umano e potrebbe avere dei problemi nella ricostruzione delle mani, di particolari acconciature e dei tacchi.

Pipeline

L'applicazione Volu permette di registrare un video utilizzando semplicemente la fotocamera del telefono. In seguito, inizia la parte di elaborazione, al termine della quale viene restituito il modello Vologram. Questa applicazione sfrutta un algoritmo di

intelligenza artificiale che permette di ricostruire la parte posteriore del soggetto non ripresa dalla camera.

La pipeline di tale applicazione include la segmentazione semantica, la stima fotometrica delle normali, la ricostruzione volumetrica monoculare e la generazione di texture posteriori [33]. Durante la fase di segmentazione semantica, viene assegnata un'etichetta di classe ai vari pixel tramite degli algoritmi di deep learning e ciò permette l'elaborazione delle informazioni visive. La stima fotometrica successiva rende possibile ottenere le normali della superficie della persona catturata per poi ottenere la ricostruzione volumetrica. Infine, le texture posteriori vengono generate utilizzando le informazioni posizionali e semantiche che permettono di aiutare e rendere più semplice il processo di predizione (figura 3.5). Il metodo utilizzato si basa su un algoritmo di traduzione image-to-image con reti adattate a predire la vista posteriore di un essere umano. La vista occlusa della persona, quindi, è in un primo momento predetta in uno spazio bidimensionale in quanto l'algoritmo image-to-image permette di effettuare una mappatura tra l'immagine in input e quella in output. Successivamente, viene utilizzata l'immagine predetta per migliorare la texture del modello 3D predetto [34] effettuando una mappatura tra i pixel dell'immagine RGB e la superficie 3D del corpo. In questo modo, si riesce a ricostruire il lato non visibile di una persona utilizzando una singola immagine e non utilizzando modelli parametrici. Questo sistema è stato realizzato in collaborazione con il Trinity College di Dublino, ma la prospettiva dell'azienda è quella di direzionare il loro lavoro verso l'utilizzo delle *diffusion networks* e altri modelli di generazione di immagini. Le *diffusion networks* vengono utilizzate per la generazione di immagini e per generare dati simili ai dati su cui sono addestrati [35].

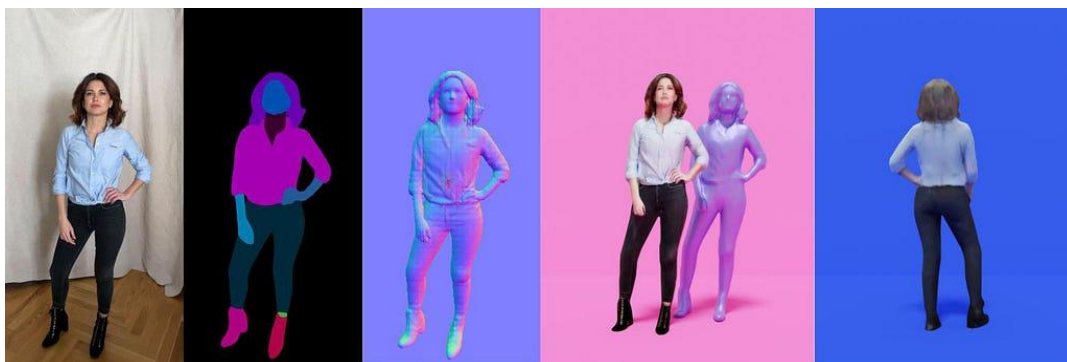


Figura 3.5: Da sinistra verso destra vengono mostrate la segmentazione semantica, la stima fotometrica delle normali, la ricostruzione monoculare volumetrica e la sintesi della texture occlusa

Fonte: <https://is.gd/QW12so>

Prima di sviluppare la tecnologia di ricostruzione 3D monoculare, in Volograms hanno iniziato a catturare le persone in 3D utilizzando una tecnica di acquisizione volumetrica multicamera nel loro studio di Dublino e in altri studi nel mondo. Ciò ha consentito di raccogliere un enorme dataset di modelli 3D di alta qualità con un'ampia varietà di movimenti, generi, tipi di corporatura, etnie e capi di abbigliamento diversi. Sono stati realizzati tali modelli 3D di alta qualità da molti punti di vista diversi, con diverse combinazioni di illuminazione e con diverse risoluzioni. Inoltre, sono stati generati dati aggiuntivi per addestrare diversi modelli di intelligenza artificiale all'interno della loro pipeline.

I modelli tridimensionali ottenuti sono caratterizzati da un'etichettatura semantica automatica. Infatti, ogni vertice della mesh viene classificato e inserito in una delle venti classi diverse. Questo sistema di etichettatura permette di identificare i capi di abbigliamento, le parti del corpo ed altre caratteristiche della persona catturata.

Formati di output

Dopo la cattura, l'applicazione processerà il filmato e il risultato verrà restituito dopo un tempo molto variabile: da qualche minuto a qualche ora. La cattura volumetrica può essere visionata direttamente sull'applicazione oppure la si può scaricare sul proprio telefono. Dopo aver esportato il file Volu dall'applicazione, si ottiene un file zip contenente un file mp4 e due file nel formato proprietario vols (figura 3.6). Il primo file contiene la sequenza di texture mentre gli altri due includono l'intestazione e la sequenza in sé.




 header.vols	File VOLS	1 KB	No
 sequence_0.vols	File VOLS	10.928 KB	No
 texture_1024_h264	MP4 Video File (VLC)	4.935 KB	No

Figura 3.6: Files ottenuti con l'esportazione dall'app Volu

Integrazione con i motori grafici

L'azienda Volograms ha reso possibile l'integrazione dei modelli volumetrici all'interno dei progetti immersivi rilasciando molteplici plugin.

I Volograms possono essere inseriti su Blender. Per l'import, è necessario convertire l'output, precedentemente analizzato, in una sequenza di files obj, una sequenza di texture in formato jpg e una sequenza di file mlt (figura 3.7).

 output_frame_00000000	Tipo: File JPG Dimensioni: 1024 x 1024	Dimensione: 364 KB
 output_frame_00000000	Tipo: File MTL	Ultima modifica: 14/05/2024 11:04 Dimensione: 164 byte
 output_frame_00000000	Tipo: 3D Object	Ultima modifica: 14/05/2024 11:04 Dimensione: 441 KB
 output_frame_00000001	Tipo: File JPG Dimensioni: 1024 x 1024	Dimensione: 372 KB
 output_frame_00000001	Tipo: File MTL	Ultima modifica: 14/05/2024 11:04 Dimensione: 164 byte
 output_frame_00000001	Tipo: 3D Object	Ultima modifica: 14/05/2024 11:04 Dimensione: 441 KB

Figura 3.7: Files ottenuti tramite il plugin per l'integrazione su Blender

In questo modo, si riuscirà a visionare il risultato volumetrico ottenuto da Volu anche su Blender. Infatti, ulteriori plugin permettono di unificare i vari files obj e ottenere un'unica sequenza.

Esiste, inoltre, un plugin rilasciato sempre da Volograms che permette di importare direttamente l'output, ottenuto dall'applicazione Volu, su Unity. Infatti, non sarà necessario effettuare alcuna conversione e i due file vols e il file mp4 saranno importati direttamente sul *game engine*. Il plugin è stato rilasciato anche per le versioni più recenti di Unity. In quest'ultimo l'output volumetrico potrà essere manipolato come qualunque altro asset. L'unica limitazione, in questo contesto, è l'impossibilità di visionare la cattura volumetrica se non nella modalità 'play'. Tale costrizione può risultare molto limitante considerando che, in questo modo, non si riesce ad avere un riscontro in tempo reale delle modifiche effettuate sull'*asset*.

Per quanto riguarda Unreal Engine, Volograms ha rilasciato un plugin per importare il file di Volu su questo game engine ma è disponibile solo per Unreal 4. Al giorno d'oggi, Unreal Engine è disponibile nella versione 5 e, per tale motivo, il plugin risulta essere molto vecchio e poco utilizzabile.

Inoltre, i risultati ottenuti da Volu possono essere integrati anche in piattaforme per la WebAR, utilizzate per creare complesse esperienze di AR come 8thWall e Blippar [32]. Infine, l'integrazione dei contenuti volumetrici di Volograms si può riscontrare anche con Three.js, una libreria JavaScript cross-browser e un'interfaccia di programmazione utilizzata per creare e visualizzare grafica computerizzata 3D animata in un browser Web.

3.2 Depthkit

3.2.1 Storia azienda e costi

Depthkit è una soluzione software realizzata dieci anni fa da Scatter, una compagnia con sede a New York. A giugno del 2024 è stata acquisita da Evercoast, un'azienda di New York impegnata nella cattura volumetrica. Questa acquisizione ha permesso di unire le forze per offrire molteplici soluzioni di tecnologie video tridimensionali e per ampliare i confini della creazione di contenuti 3D.

Depthkit permette di catturare video volumetrici, a livello professionale e consumer, tramite l'utilizzo di un PC e di alcuni sensori di profondità. In questo modo, vengono ottenute nell'unico momento della ripresa sia le informazioni di colore sia le informazioni di profondità del soggetto. Progettato per essere robusto e affidabile, Depthkit è ben consolidato tra i creatori volumetrici di tutto il mondo.

L'applicazione desktop richiede un PC in cui sia installato Windows e un collegamento con una camera RGB-D. I sensori di profondità supportati sono le Microsoft Kinect Azure e le Microsoft Kinect v2, ma viene raccomandato l'utilizzo della camera con sensore di profondità Orbbec Femto Bolt [36].

Sono disponibili tre diversi abbonamenti Depthkit tra cui scegliere. Depthkit Core, con un costo pari a 365 euro all'anno, è il punto di accesso più veloce e accessibile alla creazione volumetrica. Depthkit Cinema, invece, permette di ottenere acquisizioni volumetriche con una risoluzione fino a 8K. Questa opzione presenta un prezzo annuale pari a 3600 euro circa. Infine, Depthkit Studio è la soluzione video volumetrica professionale più avanzata e il suo costo annuale è pari a circa 27 500 euro. Questa opzione consente di connettere facilmente fino a dieci sensori di profondità Orbbec Femto Bolt o Azure Kinect a un singolo PC per creare acquisizioni volumetriche. Depthkit Studio presenta, inoltre, la possibilità di effettuare il live streaming delle catture volumetriche e l'estrazione automatica della silhouette della persona inquadrata (figura 3.8).



Figura 3.8: A sinistra setup per cattura volumetrica in live streaming. A destra il risultato della cattura volumetrica in live streaming

Fonte: <https://is.gd/veAJng>

Depthkit Core offre una versione di prova gratuita che permette di catturare i video volumetrici con una durata limitata a trenta secondi ed esportarne solamente cinque secondi (figura 3.9).

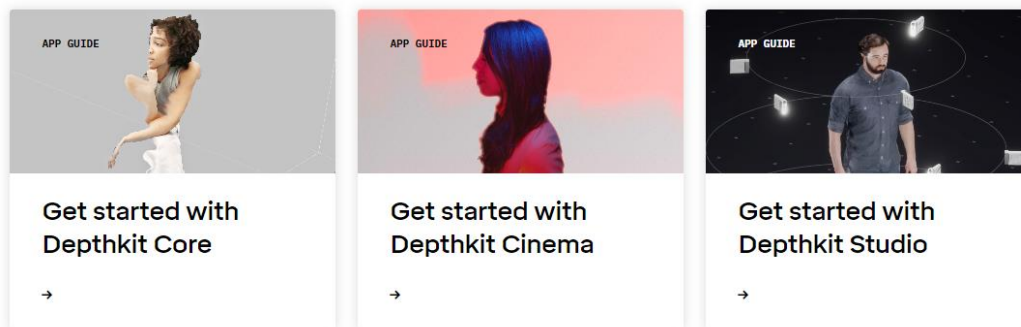


Figura 3.9: Pacchetti disponibili per Depthkit

Fonte: <https://is.gd/zIcOnZ>

Depthkit è stato adoperato per la realizzazione di diverse tipologie di contenuto: dai film di Hollywood, ai documentari di registi vincitori di premi Oscar e a innumerevoli anteprime in festival indipendenti, tra cui Sundance e Tribeca. È stato utilizzato per creare esperienze interattive di realtà aumentata per artisti importanti come gli U2 ed Eminem nei principali concerti negli stadi. Dietro le quinte, viene utilizzato dai più grandi marchi nella produzione video e nell'intrattenimento immersivo. Ne è un esempio il cortometraggio 'Zero days VR' (figura 3.10), vincitore di molteplici premi.

Il rendering ottenuto da Depthkit, in particolare nella versione Depthkit Studio, permette di ottenere un risultato abbastanza fotorealistico, ma spesso è utilizzato anche all'interno di pipeline con effetti visivi.



Figura 3.10: A sinistra video musicale 'Rap God' di Eminem. A destra il cortometraggio 'Zero days VR'.

Fonte: <https://is.gd/MaKBc9>

3.2.2 Funzionamento tecnologia

Applicazione

L'applicazione desktop Depthkit presenta due schermate principali: una relativa all'acquisizione della clip e una che permette la manipolazione del video volumetrico ottenuto (figura 3.11).

La prima schermata consente di gestire le configurazioni di ripresa e del sensore utilizzato per la cattura.

Il pannello relativo alle registrazioni, invece, contiene la lista di tutte le acquisizioni effettuate all'interno di un progetto e l'utente ha la possibilità di visionarle nella viewport 3D. In quest'ultima, si può visualizzare la cattura volumetrica sotto forma di mesh e di nuvola di punti, oppure nella modalità wireframe. Queste modalità di visualizzazione possono essere combinate con diverse tipologie di texturing (Normale, Shaded o Textured). Inoltre, la viewport 3D mostra le pre-visualizzazioni del colore e della profondità della cattura ottenuta, ovvero i dati grezzi dal sensore. Ciò è utile per assicurarsi che le impostazioni del sensore, come l'esposizione, siano state impostate correttamente o per identificare rapidamente i buchi nei dati di profondità.

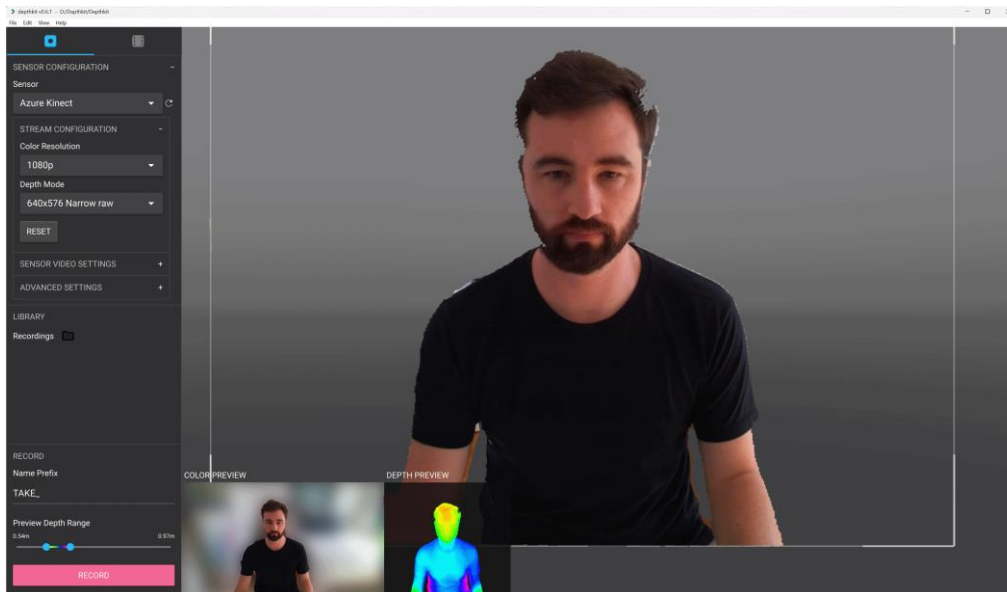


Figura 3.11: Interfaccia applicazione desktop

Fonte: <https://is.gd/voWZwb>

Nel software in questione, inoltre, si ha la possibilità di restringere il campo visivo e inserire una *matte*, realizzata con software esterni. Il restringimento del campo visivo permette di modificare il range di profondità spostando il *near* e il *far plane* (figura 3.12). Questi ultimi due piani rappresentano la minima e la massima distanza catturata dal sensore. Combinando l'uso di una *matte* e il restringimento visivo, si riesce ad ottenere un output contenente unicamente il soggetto interessato escludendo gli elementi circostanti.

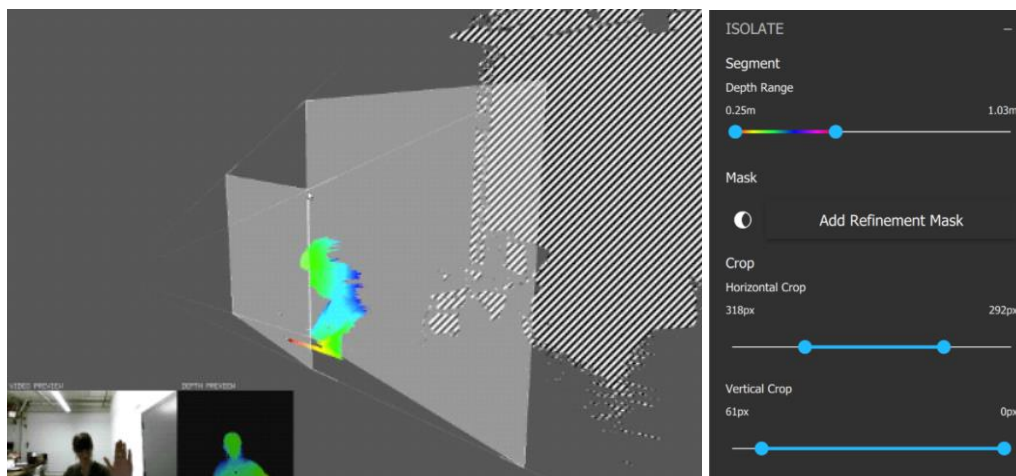


Figura 3.12: A sinistra la visualizzazione del near e del far planes nella viewport 3D. A destra la finestra dell'applicazione per l'inserimento della *matte* e di parametri vari per l'isolamento del soggetto.

Fonte: <https://is.gd/voWZwb>

Dopo aver isolato il soggetto all'interno della clip, è possibile aggiustare e modificare alcuni parametri per migliorare il risultato finale (figura 3.13).

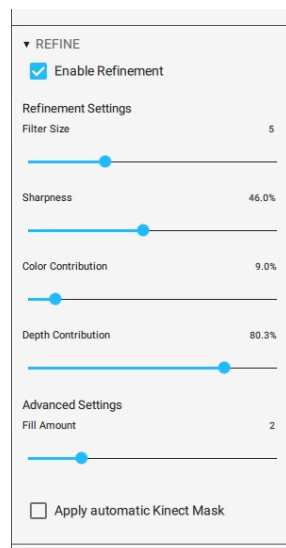


Figura 3.13: Parametri che permettono il miglioramento della clip volumetrica

Limiti cattura

Depthkit è in grado di individuare uno o più corpi nei dati di profondità tramite il rilevamento dell'indice corporeo. Quest'ultimo funge da maschera del corpo, permettendo di nascondere lo sfondo per le esportazioni dei soggetti. Il rilevamento dell'indice corporeo non riconoscerà corpi parziali o soggetti troppo vicini al sensore. Un controllo della corretta acquisizione lo si può avere tramite la pre-visualizzazione all'interno dell'applicazione desktop. Particolare attenzione deve essere anche prestata all'illuminazione in quanto se l'ambiente non fosse illuminato abbastanza, potrebbero non essere acquisiti correttamente i dati del corpo. Inoltre, si deve tenere in considerazione il movimento del soggetto in modo da non tagliare le dita delle mani e dei piedi durante la regolazione dei near e far planes. A tale fine, potrebbe essere utile fissare o contrassegnare lo spazio di acquisizione in modo che il soggetto abbia un'idea dello spazio di acquisizione previsto.

Per sfruttare a pieno il flusso di lavoro di mascheramento di Depthkit, le registrazioni devono essere catturate seguendo alcune accortezze. In primo luogo, è consigliabile riprendere il soggetto su uno sfondo rimovibile (ad esempio un green screen) e illuminato in modo uniforme. Ciò renderà più semplice isolare il soggetto dallo sfondo all'interno

della clip utilizzando strumenti convenzionali come il chromakey. Altri metodi che non richiedono uno schermo verde, come il *rotoscoping* e strumenti basati sul machine learning, sono utilizzabili, ma spesso sono alternative più complesse e impegnative. In questi ultimi casi, potrebbe essere utile effettuare una cattura dello sfondo verde senza il soggetto della durata di qualche secondo.

La tecnologia che fa funzionare le Kinect si basa sullo spettro della luce infrarossa. Per tale motivo, l'utilizzo di luci ad infrarossi nella scena può disturbare il sensore, peggiorando l'acquisizione dei dati di profondità. Sono, di conseguenza, consigliate le luci LED o fluorescenti.

Un'ulteriore raccomandazione durante la cattura consiste nell'evitare l'uso di materiali trasparenti o riflettenti; ciò significa occhiali, specchi, finestre, pelle lucida, jeans cerati, ecc. Questi materiali degradano, infatti, i dati di profondità in quanto le loro proprietà non permettono di riflettere la luce nell'obiettivo della fotocamera.

Pipeline

Il flusso di lavoro di Depthkit è simile alla produzione video, ma l'output è 2.5D. Sul set, Depthkit acquisisce dati sul colore e sulla profondità da un sensore di profondità. Quest'ultimo è un dispositivo specializzato in grado di determinare la distanza tra sé e il soggetto di interesse. I principali tipi di fotocamere di profondità includono sensori stereo, sensori *Time-of-Flight (ToF)* e sensori di luce strutturata. Ciascun tipo funziona secondo principi diversi ed è adatto per applicazioni specifiche. Le fotocamere di profondità con sensori stereo utilizzano due o più obiettivi per catturare diverse visualizzazioni di una scena. Confrontando queste visualizzazioni, la fotocamera può calcolare le informazioni sulla profondità in base alla disparità tra le immagini. I sensori ToF, invece, misurano la profondità emettendo un impulso luminoso e calcolando il tempo impiegato dalla luce per rimbalzare dopo aver colpito un oggetto. La differenza di tempo fornisce informazioni precise sulla profondità. Infine, i sensori di luce strutturata proiettano uno schema di luce noto su una scena. Analizzando le distorsioni di questo modello quando si riflette sugli oggetti, la fotocamera può dedurre informazioni sulla profondità. Per gestire le differenti condizioni luminose, le moderne fotocamere di profondità sono dotate di sensori a infrarossi e algoritmi avanzati che consentono loro di funzionare efficacemente in varie condizioni di illuminazione [37]. I sensori di profondità Kinect v2, Azure Kinect e Orbbec

Femto Bolt, utilizzabili con l'applicazione Depthkit, si basano sulla tecnologia Time-of-Flight (ToF) (figura 3.14).

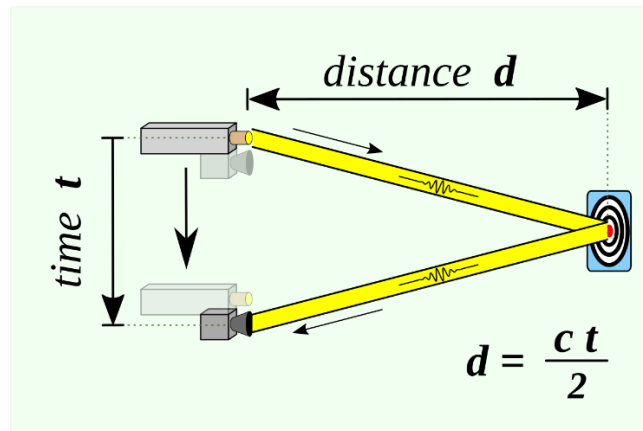


Figura 3.14: Schema funzionamento sensore Time-of-Flight (ToF)

Fonte: <https://is.gd/Zx3BjK>

Per ottenere risultati ancora migliori, si può optare per un abbinamento del sensore di profondità ad una fotocamera professionale. In quest'ultimo caso, si devono montare i sensori di profondità alle fotocamere in modo stabile tramite dei supporti per il rigging. Prima di procedere all'accoppiamento della fotocamera e del sensore, si deve eseguire la correzione della distorsione introdotta dalle lenti della fotocamera.

L'anteprima in tempo reale della profondità e del colore della cattura permette di avere la sicurezza che l'acquisizione stia avvenendo correttamente. Successivamente possono essere utilizzati gli algoritmi di perfezionamento, disponibili nell'applicazione desktop, sui dati di profondità delle riprese effettuate. Il controllo sulla qualità della cattura volumetrica, infatti, è reso possibile dall'algoritmo di miglioramento delle immagini di Depthkit. Questo algoritmo genera una mappa di profondità raffinata che si adatta alla risoluzione del video a colori, permettendo l'esportazione dei risultati alla massima qualità possibile.

L'applicazione Depthkit, dunque, sfrutta la tecnologia delle camere di profondità e si basa sulla tecnologia proprietaria *Deferred Surface Reconstruction*, in attesa di brevetto. Grazie a tale tecnologia proprietaria, la mesh non viene generata al momento dell'acquisizione, mantenendo il flusso di lavoro dinamico e agile. Inoltre, il livello di dettaglio e la densità della geometria possono essere regolati in fase di esecuzione.

Formati output

Una volta ottenuto il risultato voluto a livello visivo, si può procedere con l'esportazione che permette molteplici formati:

- Il formato “**combined per pixel video**” include in un singolo file video sia i dati di profondità che quelli di colore ed è ottimizzato per la riproduzione grazie ai pacchetti di espansione Depthkit per Unity. Questo formato, inoltre, sfrutta le pratiche video standard, come l'incorporamento dell'audio e la compressione. Scegliendo questa tipologia di formato si ottengono tre files diversi:
 1. “Combined per pixel video”: è un file mp4 e rappresenta i flussi di colore e di profondità ed è ottimizzato per Unity.
 2. “Poster image: un singolo fotogramma estratto dalla clip e visualizzato in Unity quando la clip non viene riprodotta.
 3. File di metadati: è un file txt e contiene i dati di profondità e acquisizione ed è necessario con il videoclip per la riproduzione nel motore di gioco.
- Il formato “**Combined Per Pixel Image Sequences**” produce una cartella con una sequenza di immagini nello stesso layout del video Combined per Pixel e un file di metadati di accompagnamento.
- Il formato “**Textured Geometry Sequence**” produce una sequenza, composta da tre file per fotogramma:
 1. OBJ/PLY: i dati geometrici del frame
 2. Immagine PNG/JPG: i dati colore della texture del frame
 3. Materiale: un file materiale che mappa la texture con la geometria
- Il formato “**Textured Background Geometry**” esporta una singola mesh che può essere ideale per ambienti uniformi e sfondi statici. Questo formato genera più file, come il formato “Textured Geometry Sequence” descritto precedentemente, ma esporta solo un singolo fotogramma medio, anziché una sequenza.

Integrazione con i motori grafici

Tramite un plugin, è possibile importare il risultato dell'export nel formato “combined per pixel video” su Unity. Questo plugin professionale consente di sfruttare lo Shader

Graph e il Visual Effects Graph di Unity. Inoltre, permette di utilizzare la pipeline di rendering universale (URP) e la pipeline di rendering ad alta definizione (HDRP). Tale plugin permette anche, tramite la modifica di molteplici parametri messi a disposizione, di perfezionare ulteriormente le catture volumetriche in modo da ottenere il miglior risultato possibile.

Le clip esportate da Depthkit, inoltre, possono essere incorporate sul Web in Three.js. Nella versione di Depthkit Studio, le acquisizioni esportate come OBJ con texture sono pronte per essere importate direttamente in Arcturus HoloEdit. Da lì, le acquisizioni possono essere analizzate, compresse, codificate e renderizzate proprio come qualsiasi altra clip volumetrica supportata da HoloSuite.

Depthkit Studio, inoltre, consente di pubblicare video volumetrici in tempo reale, localmente su un'applicazione in esecuzione sul proprio computer, o anche di trasmetterli su Internet ad un pubblico che utilizza applicazioni 3D interattive in qualsiasi parte del mondo. Tale funzionalità di livestreaming si può anche ottenere direttamente su Zoom, Google Meet, Facebook Live o Twitch, tramite l'utilizzo di una semplice webcam.

3.3 V3LCamera

3.3.1 Storia azienda e costi

Velox XR, azienda sviluppatrice dell'applicazione V3LCamera, conta in totale una decina di impiegati e ha sede a Woking (UK), non molto distante da Londra. Velox è stata fondata nel 2019 da Alex Grona, uno specialista tecnico, che ha lavorato all'EA Games e alla Sony. La realtà vuole rivoluzionare la virtual production, rendendola più semplice e accessibile e dare vita al metaverso in tempo reale. Velox XR si occupa sia della fase di cattura che di elaborazione di video volumetrici. Lo scanning in tempo reale e l'integrazione dei dati nei workflows della virtual production permettono di semplificarne nettamente il processo. I video volumetrici ottenuti tramite l'applicazione di Velox XR, V3LCamera, possono essere sfruttati in diversi contesti, tra cui la VR, AR e i videogiochi. Velox XR, inoltre, permette di ottenere video volumetrici in real time, registrati tramite webcam.

Con V3LCamera, sviluppatori e content creators possono agevolmente acquisire e creare rappresentazioni digitali di persone reali. V3LCamera si può scaricare dall'App Store ed è compatibile con tutti i dispositivi mobile Apple dotati di scanner LiDAR e con sistema operativo iOS 17, o successivi [37]. Quest'app è molto recente; infatti, è stata rilasciata a giugno 2023. Nonostante sia da poco sul mercato, ha già suscitato molto interesse e curiosità: infatti, il plugin Velox è già stato scaricato più di 50000 volte [38].

Per il processo di acquisizione non sono necessarie attrezzature aggiuntive: non è richiesto un green screen, né uno specifico studio e neanche la calibrazione delle camere. Per poter vedere ed elaborare successivamente i video catturati con i dispositivi mobile, sono disponibili tre plugins per Unreal: Velox Neuro, Velox Player e Velox Player Plus. Il primo e il secondo sono gratuiti, mentre Velox Player Plus ha un costo pari a 563,55 € (figura 3.15). Sul sito di Velox XR, vengono forniti tutorials e la documentazione necessaria perché tutti possano comprendere facilmente come utilizzarli.

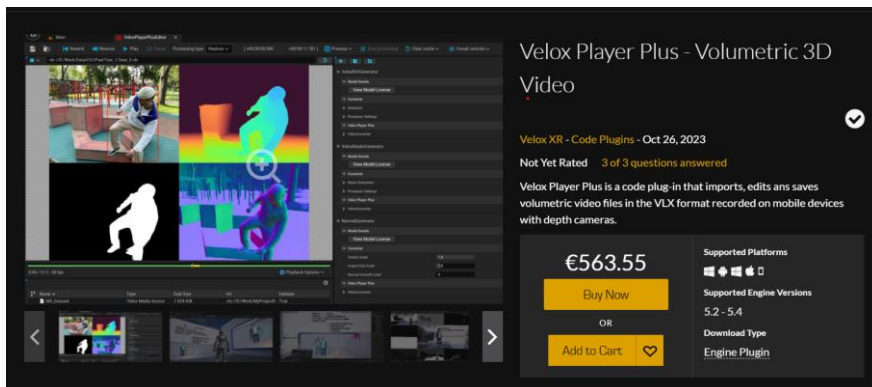


Figura 3.15: Pagina di Epic Games in cui è possibile acquistare il plugin Velox Player

Velox Player permette solo di importare su specifici motori grafici i video volumetrici precedentemente registrati, mentre Velox Player Plus e Velox Neuro introducono funzionalità più avanzate, grazie alle quali è possibile ottenere risultati più realistici e soddisfacenti.

Velox, pur essendo una nuova realtà, ha già avviato importanti progetti e collaborazioni, come testimoniato dalla partecipazione alla SXSW (South By Southwest), un importante evento annuale di interactive media, festival musicali e conferenze che si tiene metà marzo a Austin, in Texas [39]. In quest'occasione, Velox ha girato contenuti 3D dal vivo alla UK House e alla British Music Embass.

3.3.2 Funzionamento tecnologia

Applicazione

L'applicazione V3LCamera presenta un'unica schermata iniziale (figura 3.16), in cui è possibile registrare un video volumetrico. Cliccando sull'icona del punto interrogativo in alto a sinistra, si ricevono le informazioni necessarie per capire il funzionamento dell'applicazione. Prima di tutto bisogna fissare l'origine: viene chiesto a chi registra il video di cliccare su un punto nello schermo; l'origine viene generalmente scelta a livello del terreno e vicino alla persona da catturare volumetricamente. La schermata presenta inoltre una finestra con una serie di parametri, modificabili a piacimento. Si può quindi cambiare la massima profondità, il livello di compressione e scegliere se registrare in HD o 4K. Una volta applicati i dovuti accorgimenti, si può iniziare a registrare cliccando sul bottone in basso a destra. Terminata l'acquisizione, non è possibile rivedere il video, ma viene direttamente restituito un file con formato proprietario VLX, che si potrà poi importare su motori grafici grazie ai plugins citati in precedenza.

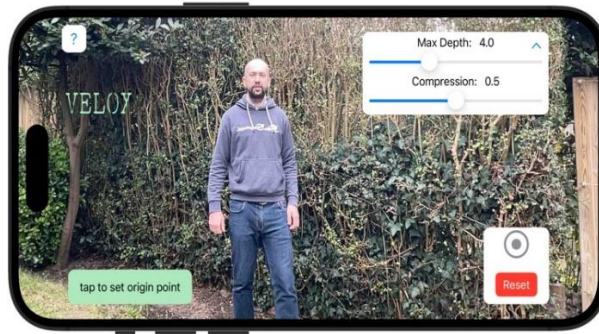


Figura 3.16: Screen della schermata di V3LCamera

Fonte: <https://appadvice.com/app/v3lcamera/1662079944>

Limitazioni cattura

Né sul sito di Velox XR, né sull'app V3LCamera, vengono fornite chiare indicazioni sugli accorgimenti necessari per la cattura. Si specifica soltanto che si può registrare dovunque e senza la necessità di complessi sistemi, ma non vengono fornite indicazioni su quante persone si possono catturare in contemporanea, se possono essere inseriti anche oggetti esterni, eventualmente anche in movimento e quali movimenti possono risultare problematici per il sistema.

Inoltre, non poter rivedere il video registrato prima del processamento può risultare problematico, perché non sempre è chiaro se la registrazione è avvenuta in modo corretto e se il video volumetrico restituito, di conseguenza, sarà soddisfacente.

Pipeline

I tre plugin per Unreal citati in precedenza presentano tre modalità e approcci differenti per l'elaborazione e il processamento di un video volumetrico.

Velox Player è un plugin per UE5 che permette di importare i video volumetrici in formato VLX. I frame del video registrato sono trasformati in una geometria animata con delle texture, completamente compatibile con la pipeline di rendering real time di Unreal Engine. Nello specifico, questo plugin implementa l'interfaccia *Media Player Framework* che permette di trasformare ogni fotogramma in una mesh 3D dinamica texturizzata. [40]. Rilasciando il file VLX all'interno di un progetto con questo plugin si ottengono tre file:

- Un file di tipo *Media Source* con il riferimento al file VLX specificato
- Un file di tipo *Media Player* per gli assets media source

- Un *Actor Blueprint* per il rendering dell'output del media player

Dopo aver registrato con la camera dell'applicazione V3LCamera, il dataset ha dati solo nei canali di colore e profondità e sono le uniche utilizzate dal plugin Velox Player.

A maggio questo plugin era scaricabile gratuitamente dal Market Place di Epic Games, ma ora si possono scaricare solo Velox Player Plus e Velox Neuro.

Velox Player Plus Editor permette di generare altre informazioni oltre quelle di profondità e colore e modificare i dati esistenti (figura 3.17). Il soggetto, con questo plugin, viene separato dall'ambiente circostante e può essere quindi successivamente integrato in un ambiente CG o con altri elementi in CG. Per effettuare il rotoscoping, vengono sfruttate e combinate tra loro diverse tecniche, come *human detection* e avanzate procedure per la generazione di maschere e normali.

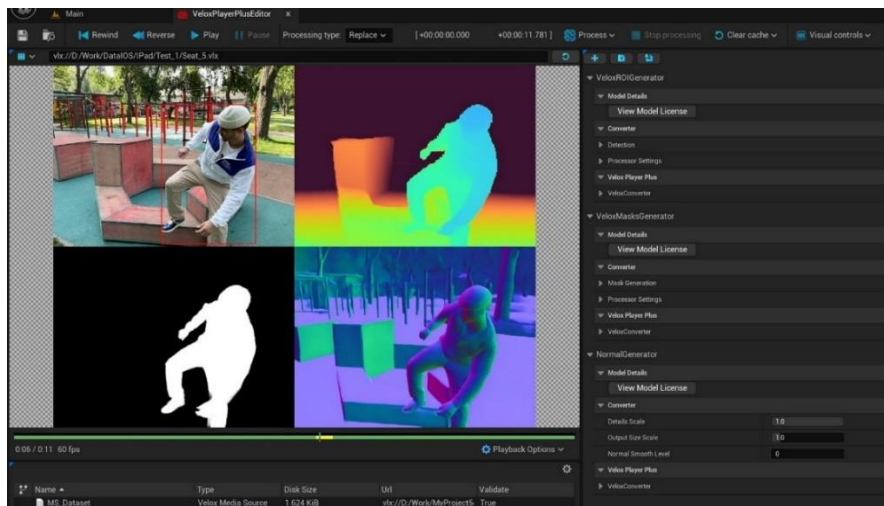


Figura 3.17: Screenshot di Unreal che mostra i passaggi necessari per ottenere un video volumetrico di una persona con il plugin Velox Player Plus

Fonte: <https://is.gd/ym34nv>

La human detection è basata su YOLOv7 [40]. YOLOv7 è lo stato dell'arte tra gli identificatori di oggetti in tempo reale e supera in velocità e accuratezza gli altri detectors [41]. Il codice sorgente è disponibile su Github [42] ed è stato utilizzato da Velox XR per l'identificazione in real time di persone e oggetti ripresi in scena (figura 3.18).

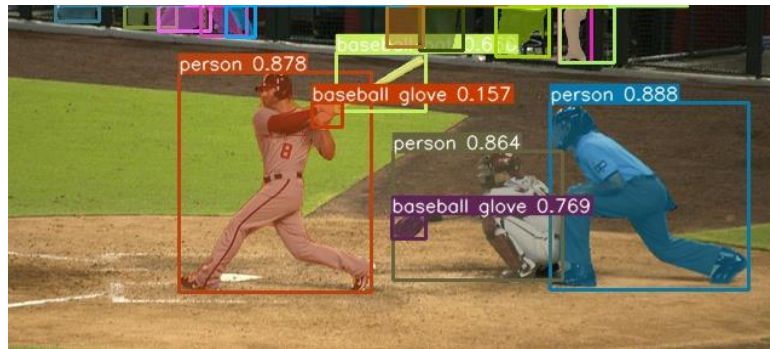


Figura 3.18: Instance segmentation

Fonte: <https://is.gd/eFMFe4>

La generazione di maschere avviene basandosi sul codice tratto da una ricerca del 2021 [43], in cui viene elaborato un metodo per realizzare maschere di persone, in tempo reale, a partire da video ad alta risoluzione. Questo metodo permette di raggiungere prestazioni estremamente all'avanguardia. A differenza dei metodi tradizionali che trattano in maniera indipendente ogni fotogramma del video, quest'approccio utilizza un'architettura ricorrente, un tipo di rete neurale progettata specificatamente per la gestione di dati sequenziali o temporali, come i video. Vengono sfruttate le informazioni temporali tra fotogrammi vicini per migliorare la coerenza temporale e la qualità complessiva dei ritagli.

Nel caso in cui le maschere generate in questo modo non siano ancora soddisfacenti, si possono usare quelle basate sul modello *Segment Anything Model (SAM)*, un nuovo modello di intelligenza artificiale di Meta Ai che permette di "ritagliare" con un banale click qualunque tipo di oggetto [44].

Per la stima della profondità monoculare, invece, viene usato un approccio che non sfrutta un unico dataset per l'addestramento dei modelli, ma che utilizza dati da fonti complementari per migliorare i risultati [45].

Le normali vengono generate a partire dai dati ricavati dal canale di profondità. Si può anche scegliere di seguire i risultati del paper "Rethinking Inductive Biases for Surface Normal Estimation" [46]. In questo metodo, per la stima delle normali, viene usata la direzione del raggio per-pixel e viene codificata la relazione tra le normali delle superficie apprendendo la loro rotazione relativa rispetto ai pixel vicini.

Per conferire un look particolare al canale colore viene sfruttato AdaAttN (figura 3.19), un'architettura innovativa per il trasferimento di un certo stile in immagini e video. Quest'approccio introduce una normalizzazione adattiva basata sull'attenzione, che

permette di considerare sia le caratteristiche superficiali sia quelle profonde di immagini e video [47].

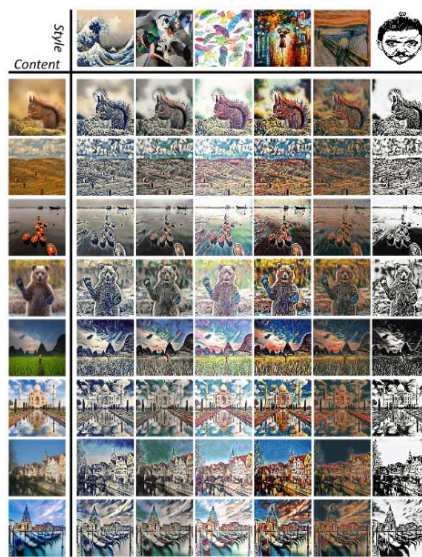


Figura 3.19: Esempi di AdaAttN

Fonte: <https://is.gd/UdmSQq>

Per tutte le operazioni necessarie per separare il soggetto della cattura volumetrica dallo sfondo viene in realtà usato il plugin Velox Neuro.

Velox Neuro è un plugin con codice open source che permette di integrare algoritmi di machine learning in Unreal Engine (figura 3.20). Il plugin sfrutta la libreria nativa di ONNX (Open Neural Network Exchange) Runtime per eseguire l'inferenza su modelli di machine learning in tempo reale. L'inferenza comporta l'esecuzione di un modello AI dopo che è stato addestrato su un dataset specifico per l'apprendimento e successivamente testato su un dataset di validazione [48]. In questo contesto, ci si riferisce all'utilizzo di un modello precedentemente addestrato per fare previsioni o prendere decisioni su nuovi dati. Velox Neuro permette di caricare e utilizzare modelli salvati nel formato ONNX, standard open source per i modelli di machine learning [49].

Il plugin consente di eseguire inferenze su piattaforme Windows e supporta l'accelerazione hardware tramite CUDA per le GPU NVIDIA e DirectML per le GPU compatibili con DirectX 12. Questa integrazione è particolarmente utile quando è necessario eseguire inferenze in tempo reale, ad esempio nel caso di rilevamento di oggetti e di persone, generazione di maschere, stima della profondità e animazione di mesh in real time.

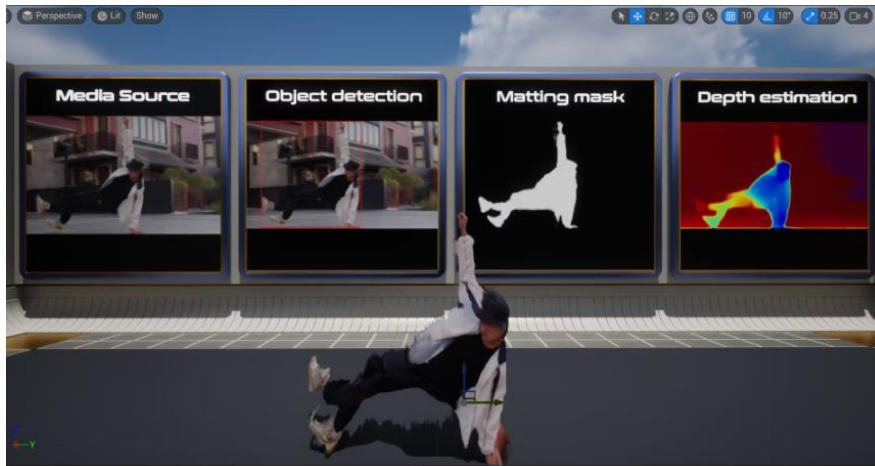


Figura 3.20: Esempio della viewport del progetto demo di Velox Neuro

Fonte: <https://is.gd/iNYq4K>

Formati di output

Una volta terminata la registrazione del video dall'app, verrà restituito istantaneamente un file VLX, un formato proprietario, basato sulla tecnologia di Velox XR. Non è possibile aprire questo file dal telefono e il risultato della cattura non si può visionare. Bisogna quindi trasferire il file VLX sul pc in cui è installato Unreal Engine e i plugin di Velox XR, trattati in precedenza. Per vedere il video volumetrico bisogna importare il video in un progetto Unreal e, a seconda del plugin utilizzato, verranno restituiti determinati dati. Si trovano informazioni sul tracking, che fornisce le indicazioni sulla posizione e sull'orientamento della camera. Sono contenuti inoltre il video compresso in formato RGB e la mappa di profondità in scala di grigio. Vengono forniti i dati di time code di sincronizzazione dei frame. Queste sono le funzionalità di base per il plugin di Velox Player, ma sono disponibili anche dati aggiuntivi per gli altri due plugin, tra cui i canali per la generazione di maschere e normali e per il tracking dei soggetti umani.

Integrazione con sistemi operativi e motori grafici

Velox Player Plus si può scaricare su Mac e su Windows 10 e Windows 11, Velox Neuro invece non può essere scaricato su Mac. Velox Player Plus funziona anche su dispositivi Android (figura 3.21).

Tecnologie utilizzate

	Windows	Mac	iOS	Android
V3LCamera	No	No	Yes, iOS 17 and above	No
Velox Player (Plus)	Yes Windows 10 and above	Yes	No	Yes
Velox Neuro	Yes Windows 10 and above	No	No	No

Figura 3.21: Screenshot del sito ufficiale di Velox in cui vengono specificate le compatibilità di sistema

Fonte: <https://is.gd/wrnvkC>

L'unico motore grafico in cui è possibile vedere ed elaborare le catture volumetriche di Velox è Unreal Engine, le cui versioni supportate per l'integrazione sono le 5.2, la 5.3 e la 5.4.

3.4 Gaussian Splatting 4D

3.4.1 Storia ed evoluzione delle reti neurali per il volumetrico

Il rendering di nuove viste (*Novel View Synthesis, NVS*) ha un ruolo cruciale nella computer vision 3D ed è fondamentale in diversi contesti e applicazioni, come la VR, AR e produzioni cinematografiche. La NVS ha l'obiettivo di generare immagini di una scena da un nuovo punto di vista, a partire da una collezione di immagini scattate da punti di vista conosciuti.

Le rappresentazioni più comuni per le scene tridimensionali sono le meshes e i punti, perché sono rappresentazioni esplicite e permettono una rasterizzazione veloce. I metodi di *Radiance Fields* hanno, però, recentemente rivoluzionato il paradigma di rappresentazione di modelli tridimensionali. I Neural Radiance Fields (NeRFs) generano una rappresentazione volumetrica (chiamata radiance field), che è in grado di restituire il colore e la densità di ogni punto della scena tridimensionale sfruttando funzioni implicite [50]. Quest'approccio permette di ottenere il rendering di nuove viste ottimizzando un Multi-Layer Perceptron (MLP), una rete neurale che mappa i dati di input in un insieme appropriato di dati di output. I NeRFs restituiscono ottime rappresentazioni accurate, a discapito, però, di lunghi tempi di training e, specialmente, di rendering. È stato però introdotto un nuovo tipo di rappresentazione, che consente di accelerare i tempi di rendering, ottenendo il real time: il Gaussian Splatting 3D. Il rendering volumetrico dei NeRFs viene sostituito con un metodo più efficiente, in cui i punti delle gaussiane 3D vengono proiettati direttamente sul piano bidimensionale. Il Gaussian Splatting non solo permette di accelerare i tempi rendering, ma poiché fornisce una rappresentazione esplicita, rende più semplice e immediata la manipolazione della scena tridimensionale [11].

Le gaussiane 3D sono una rappresentazione esplicita, sotto forma di nuvole di punti. Ogni gaussiana è caratterizzata da attributi di posizione, colore, opacità, fattore di rotazione e fattore di scala. Per ogni pixel, il colore e l'opacità di tutte le gaussiane sono combinati grazie a una specifica equazione che restituisce il risultato finale.

Gli approcci citati si concentrano sulle rappresentazioni di scene statiche, ma si può pensare ad estenderli anche a quelle dinamiche. Questo processo non è semplice, anzi presenta diverse problematiche e interessanti sfide. Le ricerche effettuate in questo ambito

sono molto recenti, ma, nonostante ciò, sono già stati sviluppati diversi approcci e soluzioni che testimoniano un forte interesse per la tematica e i risultati ottenuti sono promettenti. Tra le ricerche si sottolinea l'importanza di "HyperNerf" e "Gaussian Splatting 4D". Entrambi i metodi restituiscono buoni risultati, ma il *Gaussian Splatting 4D* riduce notevolmente i tempi di rendering e permette di rappresentare meglio dettagli intricati degli oggetti, in particolare per la rappresentazione di scene in interni [11]. La velocità di rendering e training sono molto importanti, specialmente se si tratta di applicazioni in ambito cinematografico, in cui è fondamentale l'immediatezza e la velocità di elaborazione dei contenuti. Per questo motivo, si è scelto di analizzare in maniera più approfondita il Gaussian Splatting 4D. È importante sottolineare che, nonostante i NeRFs e il Gaussian Splatting siano molto recenti, la ricerca, anche se già piuttosto approfondita, ha un ancora ampio margine di espansione e miglioramento, sia nel 3D che nel 4D.

3.4.2 Funzionamento tecnologia

Algoritmo e funzionamento del Gaussian Splatting 4D

La principale difficoltà di questo metodo risiede nel modellare in modo soddisfacente movimenti complicati dei punti, a partire da un input sparsi, radi. Nel 3D-GS l'insieme di gaussiane viene inizializzato con la nuvola di punti generata dal processo della SfM; allo stesso modo anche il 4D-GS sfrutta questo processo iniziale [10]. Si potrebbe pensare di costruire le gaussiane 3D a ogni marca temporale (timestamp). Un *timestamp* è una sequenza di caratteri che indicano una data e/o un orario preciso [51]. Se la scena è dinamica, i timestamp possono essere visti come fotogrammi o istanti in cui la scena cambia. Elaborare le gaussiane tridimensionali a ogni marca temporale non è conveniente perché richiederebbe molta memoria e spazio d'archiviazione. Dynamic3DGS [52] segue quest'approccio e tiene traccia della posizione e della varianza di ciascuna gaussiana 3D a ogni timestamp t_i . Il consumo di memoria è lineare ($O(tN)$, dove N è il numero delle gaussiane 3D e t i vari timestamps). Invece, se i movimenti e le deformazioni delle gaussiane vengono rappresentati con un'efficiente rete di campi di deformazione gaussiana (*Gaussian deformation field network*), si riesce a mantenere una rappresentazione compatta con bassi tempi di training e rendering [11]. In questo caso, il

consumo di memoria dipende solo dal numero di gaussiane presenti nella scena (N) e dai parametri del campo di deformazione (F). La complessità è quindi di ordine lineare, $O(N+F)$. Grazie al Gaussian deformation field network, è possibile trasformare, ad ogni timestamp, le canoniche gaussiane 3D in quelle nelle nuove posizioni con le relative nuove forme (figura 3.22). Il processo di trasformazione permette quindi di rappresentare sia il movimento che la deformazione delle gaussiane. Il campo di deformazione F è composto da due moduli principali: un encoder e un decoder. Inoltre, l'ottimizzazione permette di affinare i risultati ottenuti.

L'**encoder** spaziale e temporale H include un modulo di sei piani multi-risoluzione e un MLP, una rete neurale a strati in questo caso di dimensioni ridotte.

Per la codifica delle informazioni si parte dal presupposto che gaussiane vicine condividano informazioni spaziali e temporali simili. Dato un gruppo di gaussiane 3D, viene calcolato il centro delle coordinate di ciascuna e il suo timestamp. Dalle informazioni estratte si ricava il valore medio delle gaussiane (x,y,z) e ne si memorizzano le diverse combinazioni spaziali e temporali: $\{(x, y),(x, z),(y, z),(x, t),(y, t),(z, t)\}$. In seguito, un piccolo MLP viene utilizzato per combinare le caratteristiche spaziali e temporali. In uscita, si ottiene un vettore con le informazioni finali, che viene decodificato per la predizione delle deformazioni delle gaussiane [11].

Il **decoder**, a partire dalle caratteristiche finali restituite dall'encoder, predice la deformazione delle gaussiane 3D. Questo decoder è costituito da una serie di MLP separati, ciascuno specializzato in uno specifico tipo di deformazione: posizione (ΔX), rotazione (Δr) e scala (Δs) [11]. Una volta applicate queste trasformazioni, si ottengono i nuovi parametri (X',r',s') per le gaussiane deformate.

Per quanto riguarda l'**ottimizzazione**, si parte con l'inizializzazione delle gaussiane 3D statiche con la SfM e, attraverso un processo iterativo, vengono apprese le gaussiane che rappresentano in modo accurato le parti in movimento della scena.

Le gaussiane vengono infine proiettate sul piano dell'immagine per creare la rappresentazione finale. Il processo di splatting implica la distribuzione delle Gaussiane nello spazio dell'immagine, tenendo conto delle loro dimensioni e intensità per contribuire correttamente all'immagine finale.

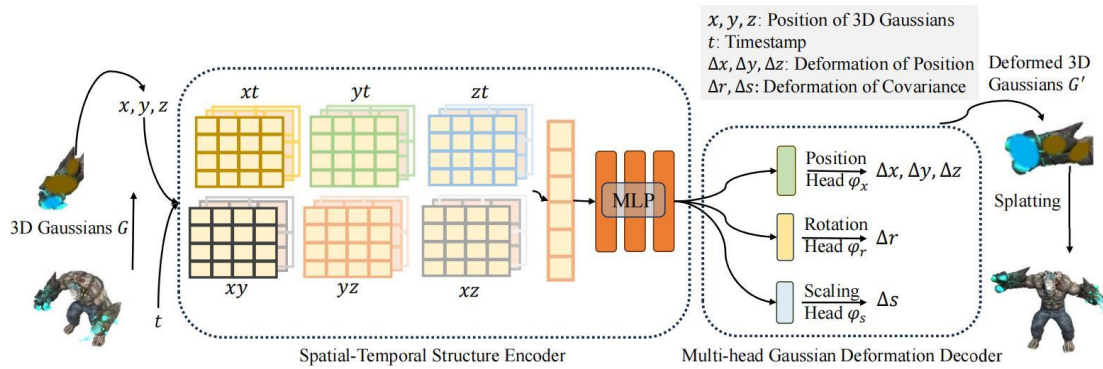


Figura 3.22: Pipeline completa del Gaussian Splatting 4D

Fonte: <https://is.gd/6cUIO2>

Limitazioni

Nonostante il 4D-GS permetta di ottenere buoni risultati e raggiunga il real time nel rendering, gli stessi ricercatori sottolineano alcune problematiche. Prima di tutto, movimenti ampi non riescono a essere modellati efficacemente e possono restituire un risultato non soddisfacente. Inoltre, lo sfondo, in alcuni casi, potrebbe essere non rappresentato in modo accurato, con un numero di punti insufficiente, se non addirittura con nessun punto. Infine, la stima imprecisa della posizione delle camere può causare artefatti e degradare il risultato finale. In aggiunta, è ancora difficile per 4D-GS separare il movimento delle gaussiane statiche e dinamiche nei setting monocamerali senza alcuna supervisione aggiuntiva, cioè senza l'uso di dati di profondità o telecamere multiple.

Nei setting con una sola camera, è più complesso ottenere ottimi risultati e potrebbero verificarsi dei problemi, soprattutto nel training di alcune scene particolarmente complesse. Inoltre, anche quando vengono sfruttate più camere nella fase di cattura, se la scena presenta molti movimenti ampi, come nei dataset di sport, il 4D-GS non riesce a offrire buoni risultati in tempi ridotti. Anche quando viene utilizzata una sola camera, il campo di deformazione potrebbe non essere in grado di modellare efficacemente movimenti ampi e drastici cambi di scena e restituire un risultato sfocato.

Nel momento in cui si decide di processare un proprio dataset, si devono adottare alcuni accorgimenti per ottenere una rappresentazione soddisfacente. Innanzitutto, bisogna considerare le limitazioni appena citate per evitare situazioni che potrebbero potenzialmente causare artefatti visivi. Inoltre, poiché inizialmente viene sfruttata la SfM,

è opportuno cercare di inquadrare scene con diversi oggetti e molti punti di contrasto per facilitare la ricerca e l'individuazione delle features.

Funzionamento codice

Il paper [11] rimanda a una pagina Github con il codice open source associato [53]. È possibile anche eseguire il codice in Colab e non solo in locale.

Prima di tutto, è necessario verificare che, nell'ambiente utilizzato, siano installate tutte le librerie e le dipendenze necessarie, con le versioni esatte, consultabili nei requisiti della pagina.

I comandi da eseguire sono diversi a seconda che si voglia effettuare una ricostruzione di scene sintetiche o di registrazioni di spazi reali e, inoltre, sono differenti anche in base al numero di camere utilizzate. Ci sono diversi dataset già forniti che possono essere usati per la ricostruzione con le gaussiane, sia sintetici che reali. Prima di iniziare a eseguire comandi, i dataset devono essere organizzati come specificato nel Github.

Inoltre, viene offerta la possibilità di processare un proprio video, catturato con una o più camere RGB. Se si sceglie di usare questo tipo di dataset, è necessario installare *nerfstudio* e seguire la loro pipeline COLMAP per la stima delle posizioni delle camere (figura 3.23).

```
pip install nerfstudio
# computing camera poses by colmap pipeline
ns-process-data images --data data/your-data --output-dir data/your-ns-data
```

Figura 3.23: Comandi per l'installazione di nerfstudio e il processamento delle immagini con colmap

Fonte: <https://is.gd/57T9cs>

Per il processamento con Colmap, possono essere forniti come dati in ingresso sia i frames del video che il video stesso. Questa fase può richiedere abbastanza tempo, da minuti a ore, a seconda della lunghezza del filmato, del numero delle camere e della risoluzione. Se viene utilizzata una sola camera vengono restituiti i seguenti file (figura 3.24).

colmap	26/07/2024 10:55	Cartella di file	
images	26/07/2024 10:55	Cartella di file	
images_2	26/07/2024 10:55	Cartella di file	
images_4	26/07/2024 10:55	Cartella di file	
images_8	26/07/2024 10:55	Cartella di file	
sparse_pc	10/06/2024 16:37	3D Object	1.957 KB
transforms.json	10/06/2024 16:37	Adobe.AfterEffect...	639 KB

Figura 3.24: Output di Colmap

L'unica cartella che viene utilizzata dal 4D-GS è "colmap", al cui interno si trovano le cartelle riportate nell'immagine sottostante. In "images" si trovano i vari frame del video processato, mentre "sparse" contiene alcuni risultati necessari per il training successivo. Colmap immagazzina le informazioni estratte in un unico file database SQLite (figura 3.25).

images	26/07/2024 10:55	Cartella di file	
sparse	26/07/2024 10:55	Cartella di file	
database	10/06/2024 16:00	Data Base File	352.412 KB

Figura 3.25: Contenuto della cartella "colmap"

All'interno della cartella "sparse" sono contenuti i seguenti file (figura 3.26).

cameras.bin	10/06/2024 16:37	File BIN	1 KB
images.bin	10/06/2024 16:37	File BIN	25.839 KB
points3D.bin	10/06/2024 16:37	File BIN	8.693 KB
points3D	11/06/2024 12:14	3D Object	1.788 KB
project	10/06/2024 16:31	Impostazioni di co...	2 KB

Figura 3.26: Contenuto della cartella sparse

Di default, Colmap utilizza un formato di file binario (leggibile dalla macchina velocemente) per archiviare modelli sparsi. Le informazioni sono divise in tre file, uno contenente i dati relativi alle camere, uno alle immagini e uno ai punti 3D. Qualunque cartella contenente questi tre file costituisce un modello sparso [54].

L'output restituito con più camere viene ottenuto eseguendo il seguente comando (figura 3.27) e restituisce in uscita le cartelle specificate in figura (figura 3.28).

```
bash multipleviewprogress.sh (youe dataset name)
```

Figura 3.27: Comando per il processamento con più camere

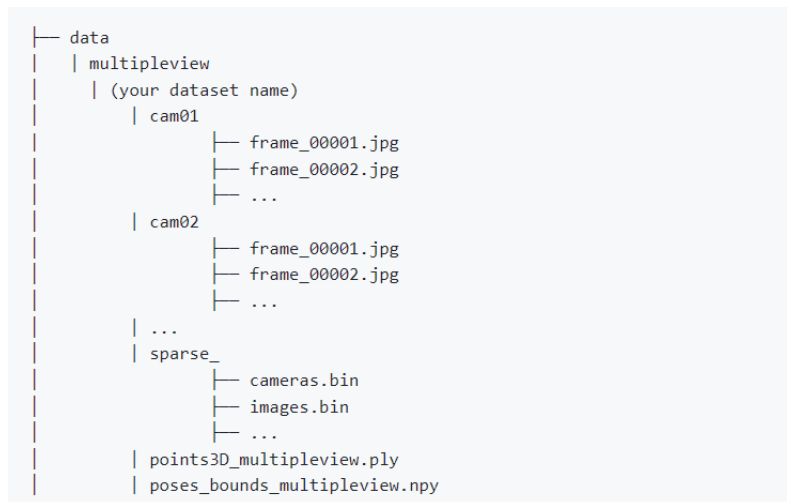


Figura 3.28: Output del processamento con più camere

Fonte: <https://is.gd/57T9cs>

A questo punto, terminata la fase di pre-processing, si può passare alla fase di training, che viene eseguita tramite questo comando (figura 3.29).

```
C:\Users\carol> python train.py -s data/multipleview/(your dataset name) --port 6017 --expname "multipleview/(your dataset name)" --configs arguments/multipleview/(your dataset name).
```

Figura 3.29: Comando generico per il training

Una volta terminato il training, è possibile visionare il risultato ottenuto tramite un viewer. Il viewer che viene presentato nel Github, però, non riproduce il movimento, ma mostra come risultato il primo frame del video e non consente quindi di vedere la scena dinamica e capire la bontà della sua rappresentazione. In questo viewer ci sono diverse possibilità di visualizzazione: la scena può essere visionata tramite le gaussiane splattate (figura 3.30), sotto forma di nuvola di punti (figura 3.31), o con ellissoidi (figura 3.32).

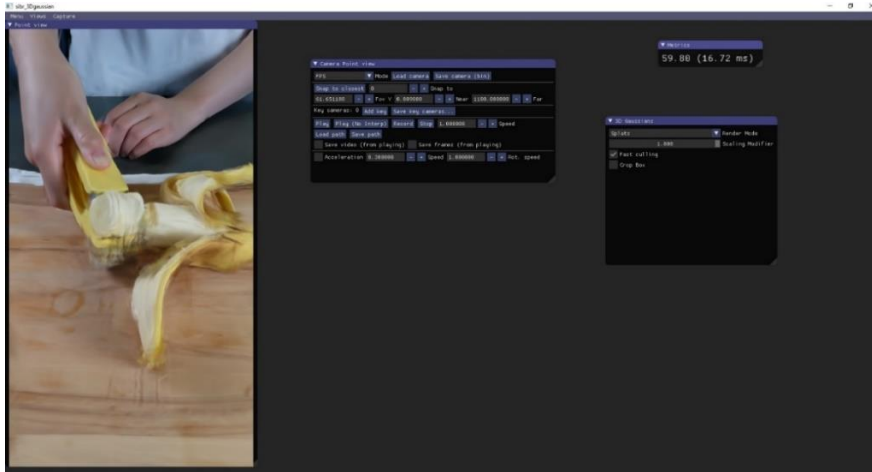


Figura 3.30: Screenshot del viewer collegato al paper con la rappresentazione offerta dalle gaussiane splattate

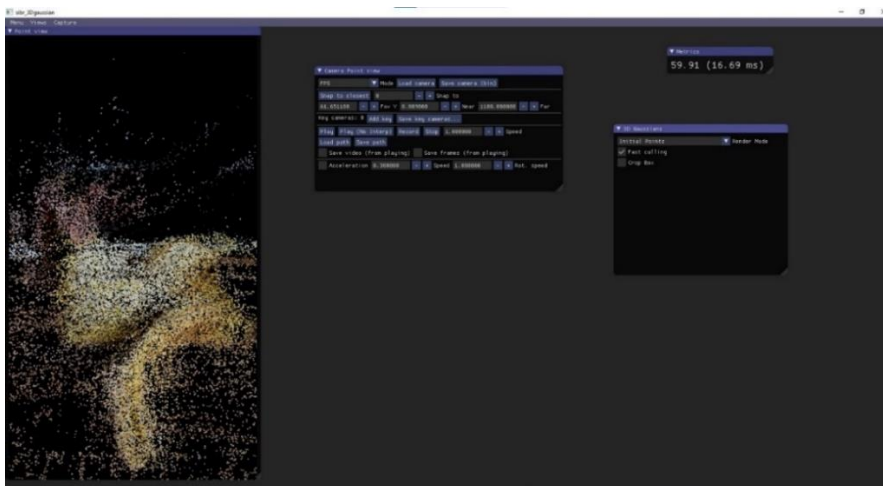


Figura 3.31: Screenshot del viewer collegato al paper con la rappresentazione sotto forma di nuvola di punti

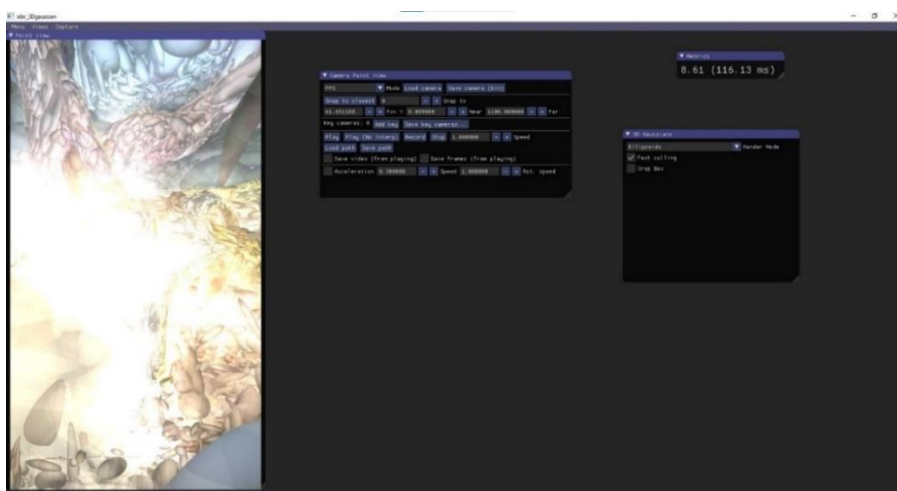


Figura 3.32: Screenshot del viewer collegato al paper con la rappresentazione con ellissoidi

È però possibile usare il viewer su browser (figura 3.33) di un altro Github [55] che consente di vedere la scena in movimento. La latenza di questo sistema di visualizzazione è molto alta, anche perché è su browser e soffre anche i ritardi della rete. Il viewer in questione, inoltre, presenta numerose limitazioni nella fase di esportazione in quanto permette di posizionare le camere nello spazio in modo arbitrario ma il video renderizzato che si ottiene non è dinamico.



Figura 3.33: Screenshot del viewer da browser

Se invece si desidera renderizzare la scena negli stessi punti di vista di quelli delle riprese si può sfruttare il seguente comando (figura 3.34), personalizzandolo a dovere.

```
C:\Users\carol>python render.py --model_path "output/dnerf/bouncingballs/" --skip_train --configs arguments/dnerf/bouncingballs.py
```

Figura 3.34: Esempio di comando per il training

Si possono eseguire degli script specifici che restituiscono le metriche per la valutazione finale del modello (figura 3.35).

```
C:\Users\carol>python metrics.py --model_path "output/dnerf/bouncingballs/"
```

Figura 3.35: Esempio di comando per la valutazione finale del modello

Formati di output

Nell'immagine qui sotto (figura 3.36) sono riportati i risultati ottenuti dopo il training e il rendering.

coarsetest_render	26/07/2024 10:52	Cartella di file	
coarsetrain_render	26/07/2024 10:52	Cartella di file	
finetest_render	26/07/2024 10:52	Cartella di file	
finetrain_render	26/07/2024 10:52	Cartella di file	
point_cloud	26/07/2024 10:52	Cartella di file	
test	26/07/2024 10:52	Cartella di file	
video	26/07/2024 10:52	Cartella di file	
cfg_args	08/07/2024 16:34	File	3 KB
per_view.json	17/07/2024 14:12	Adobe.AfterEffect...	23 KB
results.json	17/07/2024 14:12	Adobe.AfterEffect...	1 KB

Figura 3.36: Output del training e del rendering di un video con il 4D-GS

Le cartella “coarsetrain_render” e “coarsetest_render” contengono i risultati del training a bassa risoluzione (figure 3.37 e 3.38). In questa fase, il modello viene addestrato su dati a bassa risoluzione per ottenere una prima rappresentazione grossolana della scena.

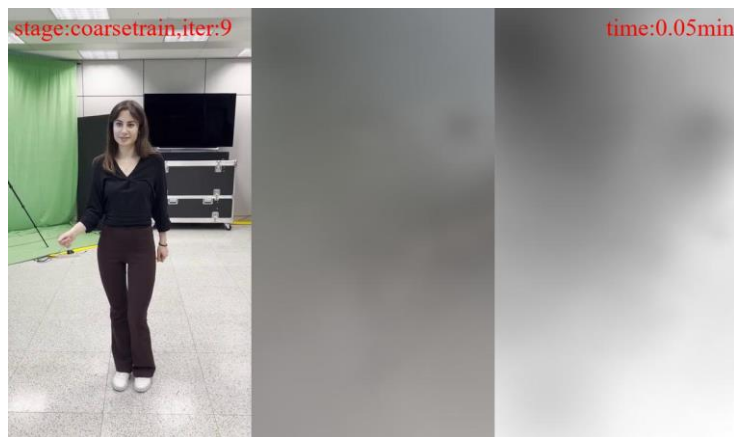


Figura 3.37: Esempio di risultato nella cartella "coarsetrain_render"



Figura 3.38: Esempio di risultato ottenuto in "coarsetest_render"

Le immagini qui sopra sono il risultato ottenuto all'iterazione numero nove. I risultati delle iterazioni successivi (figura 3.39) mostrano un miglioramento progressivo, che però si concentra sempre su una rappresentazione approssimata della struttura e della geometria globale della scena.



Figura 3.39: Risultato di "coarstrain_render" all'ultima iterazione

Le cartelle "finetrain_render" e "finetest_render", invece, mostrano i risultati ad alta risoluzione, in cui si nota un miglioramento della qualità e del dettaglio delle rappresentazioni (figure 3.40 e 3.41).



Figura 3.40: Risultato dell'ultima interazione in "finetrain_render"

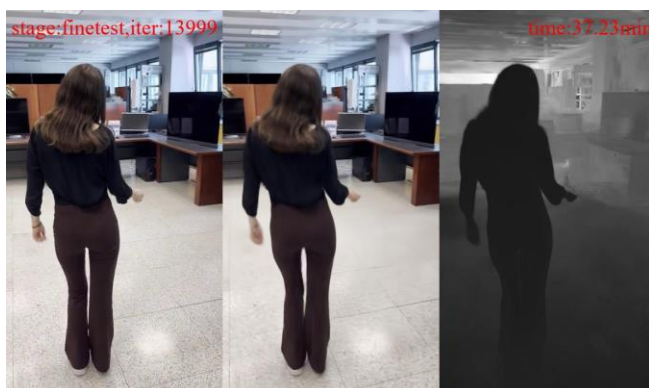


Figura 3.41: Risultato dell'ultima interazione in "finetest_render"

La cartella “point_cloud” contiene la rappresentazione della scena sotto forma di nuvola di punti. Il file “cfg_args” è molto utile, in quanto contiene le configurazioni e i parametri utilizzati durante l’esecuzione del codice, come i parametri di training e di rendering. I file “per_view.json” e “results.json” presentano rispettivamente i dati e metadati per ogni vista e una panoramica dei risultati finali, con specifiche metriche di valutazione.

Infine, “test” e “video” sono relative ai risultati finali ottenuti dal rendering. La cartella “video” contiene il video e i frame restituiti dopo la ricostruzione della scena con le gaussiane.

Integrazione con i motori grafici

Al momento non esistono plugin che permettano di importare il gaussian splatting 4D su qualsiasi motore grafico. Esistono diversi plugin, a pagamento e no, che permettono di importare i risultati del gaussian splatting 3D. Importare frame per frame la nuvola di punti e poi unire i vari risultati con un apposito plugin non si è rivelato un approccio efficace. L’integrazione, al momento, non è quindi ancora possibile, ma potrà sicuramente esserlo tra qualche tempo: diverse realtà ci stanno infatti lavorando. In particolare, si evidenzia un articolo [56] in cui le gaussiane 4D sono state integrate su Unreal Engine 5; viene dichiarato che verrà rilasciato il plugin che hanno utilizzato. L’articolo è stato pubblicato a maggio di quest’anno e l’uscita del plugin sembra imminente.

Tabella riassuntiva

APPLICAZIONE	TECNOLOGIA	DISPOSITIVO INPUT	TEMPO ELABORAZIONE	RICOSTRUZIONE	OUTPUT	INTEGRAZIONE
Volu	AI	Camera RGB	Da qualche minuto a qualche ora	3D	Proprietario: file vols + file mp4 per le texture	Blender, Unity, Unreal Engine 4, 8th Wall, Blippar, Three.js
Depthkit	Camera RGB-D	Kinect v2, Kinect Azure e Orbecc Femto Bolt	Qualche minuto	2.5D	Combined per pixel video, Combined per pixel image sequences, Textured geometry Sequence, Textured background geometry	Unity, Three.js, Arcturus HoloEdit
V3LCamera	LiDAR	Camera Apple dotata di LiDAR	Qualche minuto. Tempo reale con il plugin VeloxNeuro	2.5D	Proprietario: file vlx	Unreal Engine 5.2, 5.3, 5.4
Gaussian Splatting 4D	AI	Camera RGB	Da un'ora a qualche ora	3D	Video RGB, nuvola di punti e Gaussiane 4D	Nessuna

Tabella 3.1: Confronto tra le varie soluzioni tecnologiche

Capitolo 4

Valutazione delle tecnologie

In questo capitolo viene mostrata una valutazione sulle soluzioni tecnologiche utilizzate per la realizzazione dei video volumetrici. A tale scopo, vengono analizzate potenzialità e limitazioni di ciascuna tecnologia. Inoltre, vengono esplicitati i possibili ambiti di utilizzo, considerando il realismo ottenuto dalle varie catture volumetriche e la possibilità di effettuare il relighting su di esse in fase di post-produzione.

Infine, le applicazioni Volu, Depthkit e V3LCamera vengono confrontate tra di loro, inserendo le varie catture volumetriche nello stesso ambiente virtuale. Il Gaussian Splatting 4D, invece, non consente l'integrazione della ricostruzione volumetrica in motori grafici e, per tale motivo, viene analizzato indipendentemente.

4.1 Volu

Per la realizzazione di video volumetrici è stata sfruttata l'applicazione Volu installata su un Iphone 13. L'applicazione Volu, come visto in precedenza, permette di esportare un file zip, al cui interno ci sono un file mp4 e due file nel formato proprietario vols. Il primo file contiene la sequenza di texture mentre gli altri due includono l'intestazione e la sequenza di modelli 3D che, se posti uno di seguito all'altro, restituiscono il video volumetrico. I risultati ottenuti sono stati importati su Blender e Unity. Per poter importare la sequenza di obj su Blender è stato necessario installare un plugin apposito che permettesse di unire i file obj in un unico modello animato nel tempo, mentre su Unity esiste un plugin di Volograms che automatizza le operazioni.

I requisiti da rispettare per ottenere un video volumetrico soddisfacente non includono l'utilizzo di un green screen, motivo per cui, nei primi test, si è deciso di riprendere un soggetto in movimento in una stanza ben illuminata senza il green screen in background (figura 4.1). Si notano però numerosi artefatti visivi.



Figura 4.1: Alcuni render da Blender frontali a figura intera di una cattura volumetrica realizzata con l'app Volu senza l'utilizzo di green screen

Il soggetto non viene scontornato in maniera ottimale e i bordi sono un po' sfarfallanti. In particolare, se le gambe sono unite e le braccia abbastanza aderenti al corpo, il risultato ottenuto presenta difetti evidenti. Inoltre, la ricostruzione delle mani è piuttosto problematica; appaiono infatti, in alcuni frame, come un unico blocco, in cui le dita non si riescono a distinguere. Questo problema si verifica soprattutto quando i movimenti sono veloci. Il sistema inoltre presenta alcune difficoltà nel separare le scarpe dai pantaloni; il modello non offre quindi una chiara e ben visibile separazione tra i jeans e le scarpe e tende a fondere le due in un tutt'uno non molto realistico (figura 4.2). Inoltre, i piedi non sono entrambi allo stesso livello e questo causa compenetrazioni con il terreno in alcuni frame. Allo stesso modo anche la separazione tra la manica e la mano sul braccio è sfumata, non netta e peggiora il realismo del modello (figura 4.3).



Figura 4.2: A sinistra dettaglio piedi. A destra dettaglio gambe



Figura 4.3: Dettaglio braccia e mani

Si nota un miglioramento nella resa delle mani quando il movimento è lento e sono poste di fronte alla camera, come mostrato in figura (figura 4.4).



Figura 4.4: A sinistra render a figura intera. A destra dettaglio buona ricostruzione della mano.

Le espressioni facciali sono realistiche, ma, anche in questo caso, se i movimenti della testa sono troppo veloci si creano alcuni artefatti visivi (figura 4.5). Il risultato, tutto sommato, però, è soddisfacente.



Figura 4.5: Viso ed espressioni facciali

Il video volumetrico ottenuto presenta comunque un buon realismo e tridimensionalità. Infatti, Volu ricostruisce totalmente la figura umana, anche nella parte posteriore che non è stata ripresa (figura 4.6), e questo permette di muoversi nello spazio più liberamente, ottenendo comunque una rappresentazione convincente. La parte posteriore, però, dato che viene completamente ricostruita senza avere alcun riferimento nel video, non è ottima e non è possibile una rotazione a 360° intorno al soggetto che restituisca un risultato plausibile. Ciò nonostante, è importante sottolineare che difficilmente, in applicazione pratiche, è richiesta una ripresa totale intorno al soggetto e, quindi, anche se la riproduzione non è soddisfacente non dovrebbe creare particolari problematiche. Al tempo stesso, Volu è una delle poche applicazioni che ricostruisce la persona nella sua interezza e questo permette di raggiungere un livello di realismo e tridimensionalità decisamente elevato, non ottenibile se non venisse realizzata la ricostruzione della parte non ripresa in fase di cattura. In aggiunta, l'ombra che si ottiene in questo modo è accurata, perché la ricostruzione posteriore permette di ottenere un risultato più preciso e attendibile.



Figura 4.6: Alcuni render da Blender posteriori a figura intera di una cattura volumetrica realizzata con l'app Volu senza l'utilizzo di green screen

Come si può notare in figura, la parte posteriore presenta numerose problematiche: si possono notare delle pessime ricostruzioni soprattutto per i capelli e per le texture degli abiti.

Per ottenere risultati migliori sono state effettuate numerose altre prove. Innanzitutto, ci si è concentrati sul cercare un modo per migliorare il risultato posteriore: una prima idea è stata far girare il soggetto su se stesso in fase di ripresa. Si era ipotizzato che il sistema, avendo informazioni sulla parte posteriori in alcuni momenti del video, potesse essere in grado di ricostruire meglio il soggetto nella sua totalità. Questa supposizione, però, si è rivelata errata, perché il risultato nella parte posteriore migliora solo negli istanti in cui è inquadrato dalla camera (figura 4.7), ma in tutti gli altri continua a presentare le stesse problematiche. Inoltre, così facendo, anche il viso e la parte frontale della persona non sono ricostruiti in modo soddisfacente quando non sono inquadrati. La prova non ha quindi prodotto i risultati sperati, anzi il modello ottenuto è peggiore, in quanto non è particolarmente accurato né nella parte frontale, né in quella posteriore del soggetto. In sostanza l'algoritmo di intelligenza artificiale non è in grado di ricostruire il modello e poi farlo muovere liberamente in quanto, per ogni frame, viene ricostruito un nuovo oggetto OBJ e non vengono sfruttate le informazioni precedentemente ricavate.



Figura 4.7: Modello renderizzato di fronte, di profilo e di dietro dalla stessa posizione delle riprese



Figura 4.8: Modello renderizzato di fronte, di profilo e di dietro dalla posizione opposta rispetto alle riprese

Come si può vedere in figura (figura 4.8), la parte non inquadrata dalla camera non viene ricostruita in modo soddisfacente. Si nota che gli arti vengono rappresentati con maggiore fedeltà rispetto ai capelli e al volto, che non presenta neanche i tipici connotati umani. Si deduce che nessun miglioramento è stato apportato, anzi la ricostruzione del viso si dimostra specialmente problematica.

Un'ulteriore prova è stata fatta provando a registrare con due Iphone simultaneamente un soggetto riprendendolo di fronte e di dietro (figura 4.9). I due risultati avrebbero dovuto poi essere uniti in modo da ottenere un soggetto ricostruito molto bene da entrambe le

prospettive. Questo approccio, però, incontra varie problematiche. In primo luogo, non è possibile ottenere una sincronizzazione delle camere, in quanto la registrazione deve essere effettuata direttamente dall'applicazione e non tramite un caricamento successivo di riprese. Inoltre, ogni frame produce un OBJ: l'unione delle sequenze di OBJ del soggetto ripreso frontalmente con le sequenze del soggetto ripreso posteriormente deve essere portato avanti per ogni singolo OBJ e richiede una precisione elevatissima. Infine, le camere devono avere essere posizionate e angolate con estrema precisione, perché i due risultati siano coerenti e sia possibile unirli. Il lavoro necessario è quindi complesso e richiede molto tempo e precisione; per tali motivi, questo approccio non risulta essere utilizzabile ai fini di una produzione televisiva.



Figura 4.9: Ricostruzione frontale e posteriore

Un'osservazione interessante del risultato ottenuto è l'ottima resa di vestiti con pattern e texture complesse.

Appurato che il precedente metodo non introduce miglioramenti significativi, si è deciso di provare a effettuare riprese in cui il soggetto è posto davanti a un green screen. L'utilizzo di un green screen dovrebbe migliorare il risultato, specialmente sui bordi. In effetti, si nota che i bordi non sono più sfarfallanti e diventano più definiti (figura 4.10).



Figura 4.10: Risultato del video volumetrico registrato con il green screen in background

Migliora, inoltre, anche la resa dei capelli e si riducono i difetti e gli artefatti quando le gambe sono unite e le braccia aderiscono al busto. I capelli, anche quando sono in movimento, hanno una buona resa. Si evidenzia anche un'ottima rappresentazione dei dettagli (figura 4.11): in particolare, i pantaloni hanno un'ottima qualità; le pieghe e la morbidezza del tessuto contribuiscono a rendere estremamente plausibile la ricostruzione volumetrica.

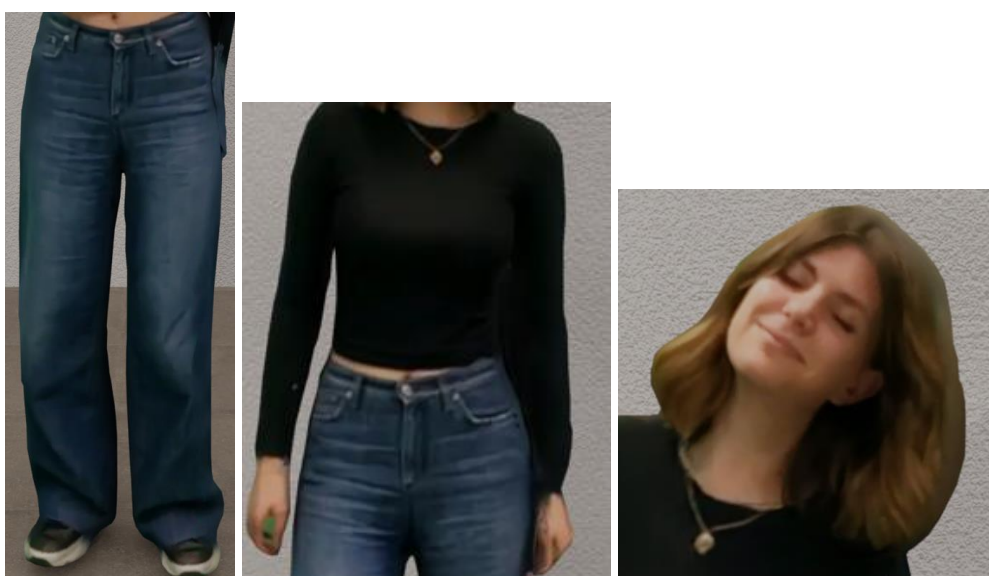


Figura 4.11: Dettagli di gambe, braccia e capelli

Non si notano particolari miglioramenti nella ricostruzione delle mani, che, se il movimento è veloce, non sono molto realistiche (figura 4.12). È opportuno però sottolineare che, nel complesso, il risultato non è fastidioso e non in tutti i frame le mani non vengono ricostruite in maniera apprezzabile.



Figura 4.12: Dettaglio delle mani

Un difetto introdotto dal green screen è lo *spill*, che sporca la figura con una tonalità verdastra (figura 4.13). In questo caso, lo spill è localizzato principalmente sui capelli e sulle braccia. Tuttavia, il risultato è, in generale, migliore e più definito e per questo motivo si è deciso di effettuare le catture volumetriche successive servendosi sempre di un green screen.

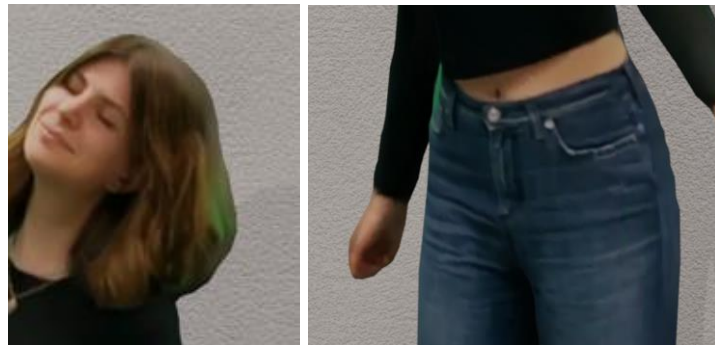


Figura 4.13: Esempi di spill

Se si muove la camera di un angolo di 60° intorno al soggetto si ottengono i risultati in figura (figura 4.14). Il senso di tridimensionalità persiste, ma il realismo nel viso si riduce abbastanza. Infatti, la faccia sembra più schiacciata e il naso non è ricostruito perfettamente. In generale, però, il risultato è buono.



Figura 4.14: Render con la camera angolata in totale di 60°

In seguito, si è deciso di indagare se il modello reagisse ai cambiamenti di illuminazione. Si dimostra che il modello si adatta alle luci a cui è esposto. Questo avviene sia che il video volumetrico venga importato su Blender, sia su Unity, modificando gli opportuni parametri. La reazione del modello ai cambiamenti di luminosità è molto importante, perché ammette la possibilità di inserire la ricostruzione volumetrica in ambienti nettamente diversi da quello in cui è avvenuta la registrazione e ottenere comunque risultati plausibili. Qui di seguito si riportano tre diversi tipi di illuminazioni, a cui è stato sottoposto il soggetto (figura 4.15).



Figura 4.15: Rendering a diverse luminosità, da sinistra a destra con luminosità crescente

Si nota che a bassa e media luminosità il risultato è più realistico e meglio integrato con l'ambiente. Una luminosità non eccessiva, infatti, permette di mascherare piccoli difetti e contribuisce a rendere più omogenea la scena. Anche nello scenario con luminosità più elevata, comunque, si ha una buona resa: alcune imperfezioni, ad esempio nel tessuto dei pantaloni, sono più evidenti, ma, tutto sommato, il realismo non diminuisce in maniera significativa.

Un'altra prova per vedere come reagisce il soggetto ricostruito ai cambiamenti di luce è stata svolta cambiando il colore dell'illuminazione e vedendo come questo influenzasse il risultato finale. Il soggetto reagisce anche a questi cambiamenti, come illustrato in figura (figura 4.16).

Inoltre, si nota che il modello presenta anche un'ombra dinamica che si adatta alle diverse situazioni di illuminazione. Questo aspetto è notevole e contribuisce a integrare perfettamente la persona nell'ambiente virtuale.



Figura 4.16: Rendering con luci di diversi colori

Infine, Volu è un'applicazione particolarmente interessante perché offre anche la possibilità di integrare i video volumetrici nell'ambiente che ci circonda al momento, direttamente dal cellulare (figura 4.17). In aggiunta, è possibile applicare degli effetti visivi in real time. Gli effetti sono predefiniti e scaricabili dalla schermata principale dell'app. L'inserimento di VFX è interessante e permette di vedere la cattura volumetrica sotto una nuova luce. I video volumetrici possono essere sfruttati non solo in chiave

realistica, ma anche diventare uno strumento creativo ed espressivo. Non sempre viene raggiunto un risultato attendibile, ma, con l'aggiunti di effetti visivi, è possibile rendere la cattura volumetrica un potente strumento artistico. Per una produzione televisiva, può essere utile l'idea di applicare degli effetti spettacolari e reinterpretare il video volumetrico. L'applicazione offre alcuni effetti predefiniti, ma è possibile anche crearne ad hoc e applicarli agli OBJ.



Figura 4.17: Alcuni effetti predefiniti che offre l'app Volu

4.2 Depthkit

Come visto in precedenza, ci sono diverse versioni di Depthkit che si possono utilizzare per realizzare video volumetrici. I costi, però, sono elevati e si è scelto di usufruire della prova gratuita offerta per la versione Depthkit Core. Ci sono alcune limitazioni sull'export e sulla fase di cattura, ma è utile per capire le potenzialità di questa tecnologia.

Per le riprese, si può utilizzare un unico sensore di profondità: nel nostro caso, è stata usata una Kinect v2. La Kinect v2 è stata collegata tramite cavo USB a un PC, su cui era stato precedentemente installato il software Depthkit.

La scena ripresa dal sensore di profondità viene vista in real time, grazie al software. Quest'aspetto è molto vantaggioso perché permette di capire se la cattura sta avvenendo in maniera corretta, se la luminosità della scena è sufficiente e se il sensore di profondità sta funzionando opportunamente.

I video sono stati registrati con il green screen come sfondo. Infatti, una volta che la Kinect v2 restituisce il risultato di colore e profondità, il soggetto deve essere isolato. Depthkit consente di inserire una maschera animata per ottenere solo il video volumetrico della persona (figura 4.18). La maschera viene realizzata su un software diverso: la scelta migliore, in termini di velocità e qualità, per noi, si è rivelata Adobe After Effects, che tramite l'applicazione dell'effetto *keylight* permette di realizzare una buona maschera in tempi estremamente ridotti.

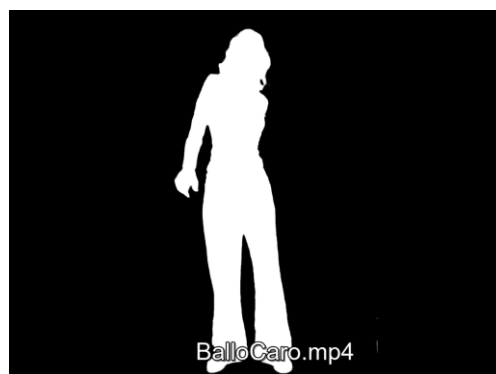


Figura 4.18: Video mp4 di una maschera animata

Una volta applicata la maschera alla cattura su Depthkit, sono stati modificati alcuni parametri per ottenere risultati con meno buchi d'informazione e più definiti (figura 4.19).

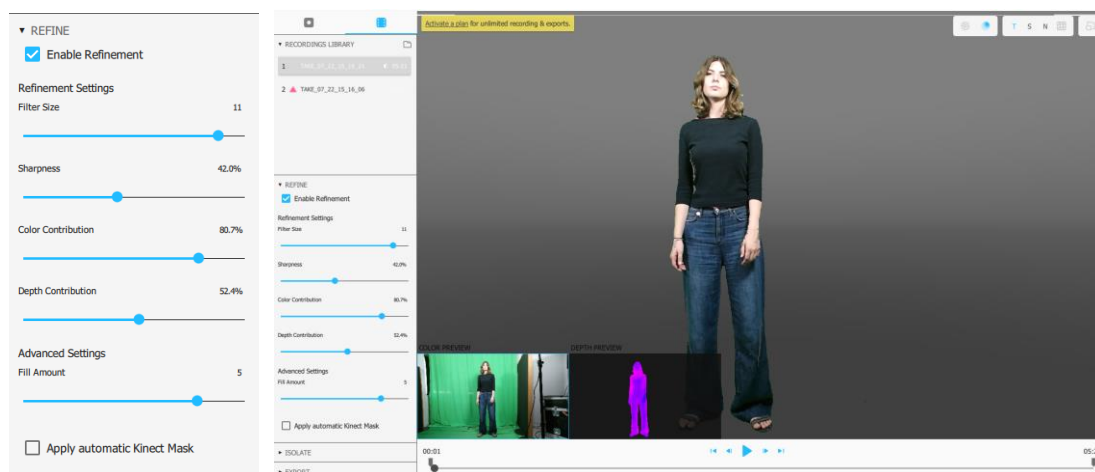


Figura 4.19: Parametri disponibili per migliorare il risultato ottenuto e schermata totale di Depthkit

Quando si è arrivati a un risultato soddisfacente, il video volumetrico è stato esportato nel formato opportuno per l'integrazione su Unity. Per sfruttare l'apposito plugin scaricabile nella prova gratuita per Depthkit Core, il formato di export che da utilizzare è "Combined per pixel video". Si ottengono in output un video con il video volumetrico e la relativa mappa di profondità, un'immagine png di un frame e un file txt di metadati (figura 4.20).

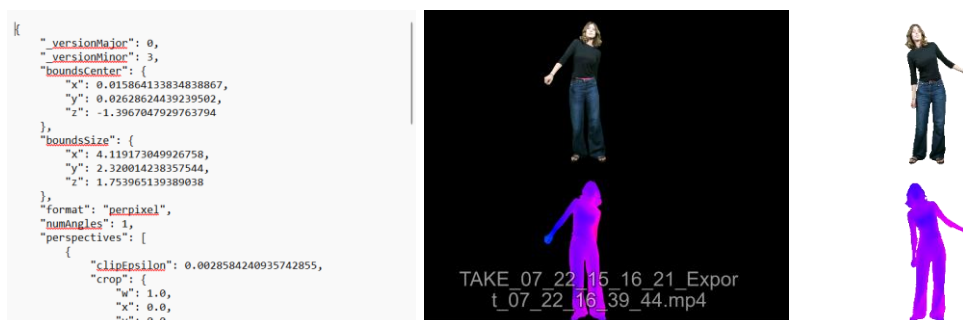


Figura 4.20: Output ottenuti esportando nel formato "Combined per pixel video"

Questi tre file, quando importati su Unity restituiscono il video volumetrico. Il file png diventa la copertina statica che compare quando non si è in modalità "play". Se non ci fosse non sarebbe possibile vedere un'anteprima del risultato ottenuto e non si vedrebbe immediatamente l'esito degli aggiustamenti, perché sarebbe necessario prima modificare e poi vedere la ricostruzione volumetrica in modalità "play". Questo file è dunque fondamentale perché permette di vedere in tempo reale le modifiche applicate. Il plugin per Unity presenta numerosi parametri, modificabili a piacimento, che permettono di ottenere l'output migliore possibile.

Una volta effettuati gli opportuni cambiamenti, si può passare alla modalità "play" per vedere se il video volumetrico è soddisfacente.

È bene notare che l'output restituito non è una ricostruzione tridimensionale completa della persona; la parte posteriore non viene ricostruita e l'output è 2.5 D. Non è possibile immaginare quindi applicazioni in cui la camera ruota a 360° intorno al soggetto, ma viene restituito comunque un senso di tridimensionalità sufficiente per movimenti di camera più limitati.

Sono stati quindi effettuati dei test per stimare la bontà della rappresentazione e valutare entro quale angolo la camera virtuale debba rimanere per ottenere ancora risultati plausibili.

In una prima prova, è stato ripreso un soggetto posizionato davanti a un green screen. I movimenti sono semplici e non troppo veloci: le gambe sono praticamente statiche, la dinamicità si concentra nella parte superiore del corpo, principalmente nelle braccia. Il risultato in figura (figura 4.21) è stato ottenuto modificando opportunamente i parametri messi a disposizione.



Figura 4.21: Output Depthkit su Unity a figura intera

Il risultato, visto di fronte, è abbastanza buono e anche il realismo viene preservato. Le mani sono piuttosto definite e le dita chiaramente identificabili (figura 4.22). I vestiti sono realistici e ben dettagliati. I bordi, però, sono estremamente sfarfallanti e, in alcune parti, vengono rimosse porzioni della figura umana. Quest'aspetto è particolarmente evidente nei capelli (figura 4.23).



Figura 4.22: Dettaglio sulle mani



Figura 4.23: A sinistra dettagli sui capelli. A destra dettagli sui bordi

Le espressioni facciali sono abbastanza plausibili, ma se il movimento della testa è veloce il risultato peggiora (figura 4.24).



Figura 4.24: Dettaglio sul viso in movimento

È stato poi registrato, nelle stesse modalità, e processato un altro video volumetrico con un diverso soggetto e movimento, simile a quello visto in precedenza nell'analisi di Volu. In quest'analisi si è cercato inoltre di capire entro quale angolo debba rimanere la camera virtuale per ottenere dei risultati soddisfacenti.



Figura 4.25: Figura intera ripresa di fronte

Quando la camera virtuale è posta di fronte alla persona, i risultati sono tutto sommato buoni, pur presentando difetti simili a quelli visti per l'altra ricostruzione volumetrica. In questo caso, in cui sono stati indossati dei pantaloni più larghi e quindi più soggetti al movimento, si sono verificati alcuni artefatti anche negli indumenti, come si può vedere in figura (figura 4.25). Se ci sposta lateralmente invece, il risultato perde la sua tridimensionalità e, di conseguenza, il suo realismo.



Figura 4.26: Risultato ottenuto con la camera posta di lato

Come si può vedere in figura (figura 4.26), il soggetto ricostruito non è realistico: ci sono buchi e parti mancanti e la figura si espande nello spazio, cioè non ci sono chiari e definiti bordi che la delimitano. Inoltre, è importante sottolineare che, quando le braccia si trovano davanti al busto, e si ruota la camera, la ricostruzione presenterà dei buchi nella parte del busto coperta dal braccio. Questo accade perché il sensore di profondità non riesce a ricavare le informazioni su ciò che è occluso.

Tutte queste imperfezioni peggiorano la qualità del video volumetrico. Si conclude che è necessario limitare i movimenti di camere e ridurre la loro ampiezza. Il video volumetrico presenta artefatti, anche se la camera viene angolata di poco, ma si ritiene che fino ad un'ampiezza di circa 60° in totale si possano ottenere risultati ancora più o meno apprezzabili. Una ricostruzione come quella in figura (figura 4.27) si può ritenere ancora accettabile.



Figura 4.27: Risultato ottenuto con la camera posta leggermente di lato

Un altro punto emblematico dell'analisi sono le interazioni delle ricostruzioni volumetriche con le luci e i suoi cambiamenti. In primo luogo, è bene notare che inserendo i risultati di DepthKit Core in un ambiente illuminato si creano ombre dinamiche che seguono e rispettano i movimenti della persona. Queste ombre, però, non ricalcano in pieno la silhouette del soggetto catturato; infatti, poiché la parte posteriore non è del tutto ricostruita e i bordi che delimitano la figura molto frastagliati, l'ombra presenta numerose irregolarità che inficiano sul realismo della rappresentazione (figura 4.28).



Figura 4.28: Esempio di ricostruzione di un'ombra

In seguito, si è deciso di analizzare le interazioni con le luci. Si nota che l'illuminazione sul soggetto non varia (figura 4.29), non è quindi influenzata dalle fonti luminose nella scena virtuale, ma solo da quelle in fase di ripresa. Quest'aspetto rende meno realistica la scena e il livello di integrazione tra la persona e l'ambiente virtuale si riduce.



Figura 4.29: Risultati ottenuti a diverse luminosità, da sinistra a destra con luminosità crescente

Non ci sono variazioni neanche se il soggetto viene illuminato con luci colorate (figura 4.30). Ancor di più in questo caso, si sottolinea come questa mancanza vada a peggiorare il risultato ottenuto.



Figura 4.30: Risultato ottenuto cambiando il colore delle luci

Un aspetto molto interessante di Depthkit è la possibilità di inserire in modo estremamente semplice dei VFX sulla ricostruzione volumetrica. Inoltre, sulla pagina web di Depthkit sono forniti tutorial, che rendono possibile la realizzazione di semplici effetti anche per chi non è esperto. Non vengono forniti esempi solo per Unity, ma anche per altri software, come Adobe After Effects e Houdini [57]. Si è deciso di provare ad applicare un semplice VFX sulla ricostruzione volumetrica trattata fino ad adesso (figura 4.31). Il software scelto è After Effects, che tramite l'utilizzo del plug-in Plexus, consente di effettuare immediate e suggestive trasformazioni sulla sequenza di OBJ importati. Applicare effetti visivi ai video volumetrici permette di semplificare la pipeline dei VFX, che richiederebbe, in precedenza, la modellazione e l'eventuale animazione di un modello 3D, al contrario non necessari se si parte dall'output di Depthkit. È cruciale sottolineare come spesso questa tecnologia non venga utilizzata per video volumetrici realistici, ma per ottenere risultati più artistici e simbolici. Si può quindi considerare Depthkit anche come utile strumento espressivo e creativo.



Figura 4.31: Risultato di un video volumetrico con effetti ottenuto su After Effects

4.3 V3LCamera

Il plugin di base dell'applicazione, Velox Player, permette l'inserimento del file in formato proprietario su Unreal Engine. Siccome la human detection non è inclusa in tale plugin, il risultato visionabile sul game engine include anche lo spazio circostante presente in fase di registrazione. In questo caso, la riproduzione del video volumetrico avviene insieme all'ambiente circostante.

Come punto di partenza per l'analisi di tale applicazione, è stato registrato un video dall'applicazione con un Iphone 13 Pro, dotato di LiDAR, e inserito su Unreal Engine 5.3 tramite il plugin di base Velox Player. In questo caso, è stato possibile analizzare non solo la cattura volumetrica del soggetto ma anche quella relativa all'ambiente intorno in quanto il risultato volumetrico visionabile considera tutto ciò che è stato catturato in fase di acquisizione. Infatti, tale plugin presenta un unico pannello in cui si può notare come sia presente solamente la stima della profondità della scena ripresa (figura 4.32) e non ci sia nessun pannello relativo al rotoscoping.

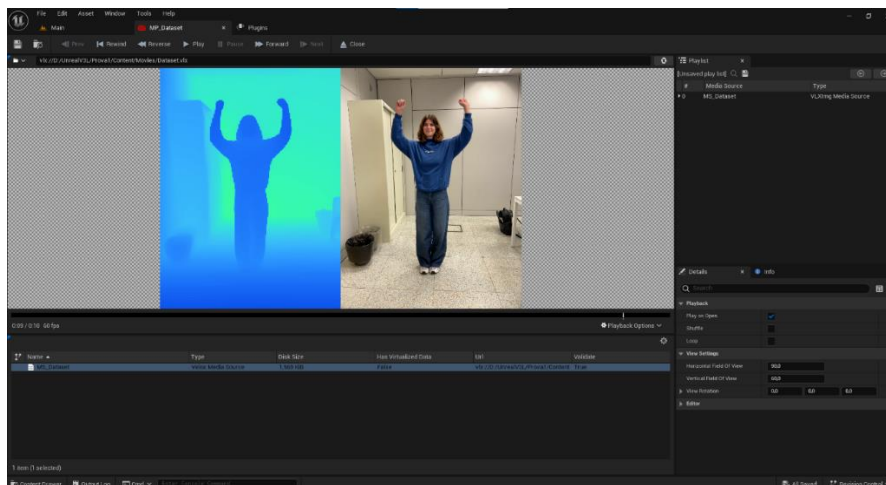


Figura 4.32: Pannello relativo alla stima della profondità della cattura volumetrica

Nonostante non sia disponibile in questo plugin la possibilità di effettuare una human detection, la figura umana risulta separata dallo sfondo (figura 4.33). Tale buco di informazioni, generato lungo il bordo del soggetto, segue il movimento della persona inquadrata ed è visibile anche se la cattura volumetrica viene visionata frontalmente. Per tale motivo, il risultato è difficilmente utilizzabile e, inoltre, risulta complicato integrarlo in ambienti virtuali.



Figura 4.33: Buco di informazione intorno al soggetto inquadrato

Inoltre, muovendosi nell'ambiente del game engine (figura 4.34) si può notare che la riproduzione volumetrica viene ricostruita fino a 30 gradi a sinistra e 30 gradi a destra, rispetto alla posizione centrale di registrazione. Tale caratteristica rende possibile un angolo di visibilità complessiva di 60 gradi.

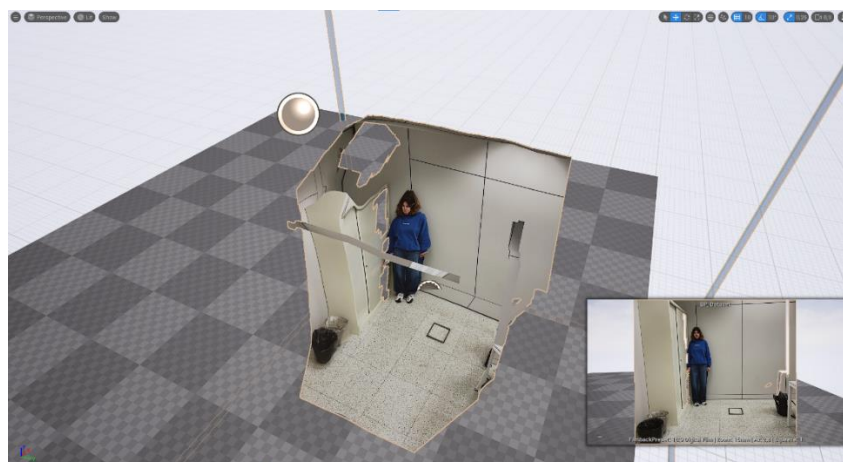


Figura 4.34: Ricostruzione volumetrica del soggetto e dell'ambiente

Superati tali gradi di mobilità, gli artefatti relativi alla cattura diventano notevolmente visibili, generando una degradazione del risultato ottenuto (figura 4.35). Infatti, si iniziano a presentare buchi di informazione non recuperabili: i volumi del soggetto e dell'ambiente subiscono una notevole deformazione rendendo il risultato irrealistico.

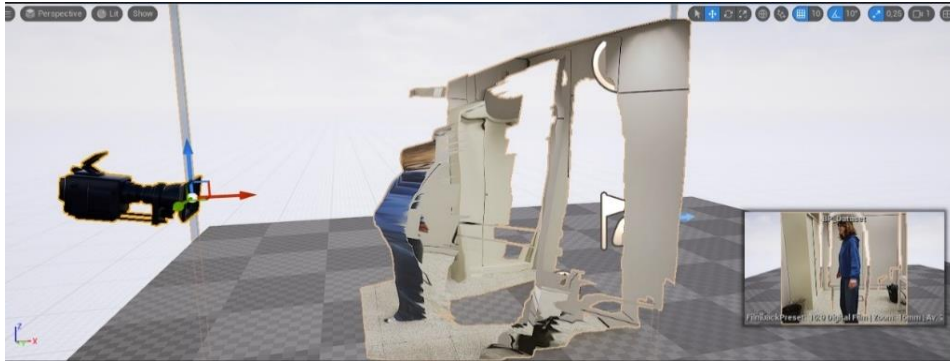


Figura 4.35: Artefatti presenti con un angolo di mobilità maggiore di 60 gradi

Il file inserito nell'ambiente virtuale possiede già una videocamera che segue il movimento della camera dell'Iphone, utilizzata in fase di acquisizione. In questo modo, il risultato può essere visionabile unicamente dalla prospettiva utilizzata durante la cattura. Tale camera può essere successivamente aggiunta ad un sequencer per poter azionare il video volumetrico nello spazio tridimensionale e si possono modificare i relativi parametri per ottenere il risultato voluto.

Per poter ampliare le possibilità di utilizzo, si può inserire nell'ambiente virtuale una nuova camera, di cui si può decidere arbitrariamente il movimento. Sfruttando tale strategia, si riescono ad utilizzare i dati volumetrici ottenuti grazie alla tecnologia LiDAR. Infatti, l'aggiunta di una nuova camera con nuovi movimenti nello spazio permette di esplorare la cattura volumetrica anche da prospettive non visibili durante la fase di acquisizione. In tale modo, si riesce ad ottenere un valore aggiuntivo rispetto ad un semplice video bidimensionale.

Considerando le limitazioni appena esposte, la cattura volumetrica ottenuta è risultata difficilmente integrabile in ambienti virtuali e, per tale motivo, l'analisi si è successivamente orientata all'utilizzo del plugin Velox Neuro. Quest'ultimo rende possibile isolare il soggetto dall'ambiente circostante e, in questo modo, si ha la possibilità di inserirlo in un ambiente virtuale arbitrario. Tale funzionalità, però, risulta non sempre estremamente precisa e, in certi casi, il rotoscoping effettuato presenta degli artefatti in prossimità dei bordi del soggetto (figura 4.36).



Figura 4.36: A sinistra un corretto rotoscoping. A destra un rotoscoping che presenta artefatti.

In particolare, se il soggetto inquadrato interagisce con degli oggetti reali allora questi non verranno rimossi (figura 4.37). Gli oggetti vengono individuati e identificati dall’algoritmo, che provvede a scontornarli opportunamente.

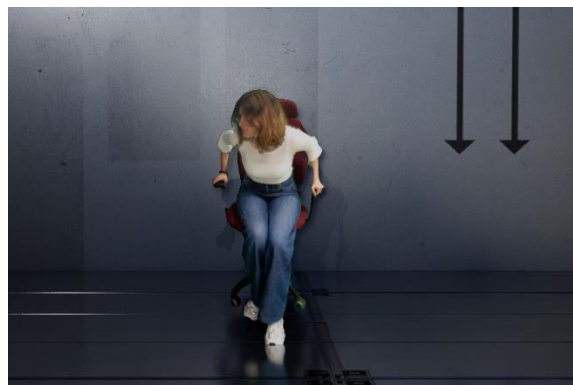


Figura 4.37: Cattura volumetrica in cui non vi è stata l’eliminazione dell’oggetto reale con cui ha interagito il soggetto

Le problematiche, relative al rotoscoping lungo il bordo del soggetto, si sono riscontrate a causa di un mancato utilizzo di un green screen che rende più complicato il riconoscimento del soggetto inquadrato per l’algoritmo. Per tale motivo, si è reso necessario indirizzare l’analisi verso catture volumetriche realizzate con l’utilizzo di green o blue screen. Ciò ha reso possibile ottenere risultati migliori. Infatti, i bordi del soggetto non risultano influenzati dagli oggetti circostanti, i quali vengono completamente eliminati nell’output finale. Inoltre, il plugin mette a disposizione molteplici materiali specificatamente realizzati per casi d’uso specifici (figura 4.38), tra

cui un materiale da utilizzare in caso di registrazione con green screen. Quest'ultimo permette un migliore rotoscoping del soggetto e un risultato maggiormente utilizzabile.

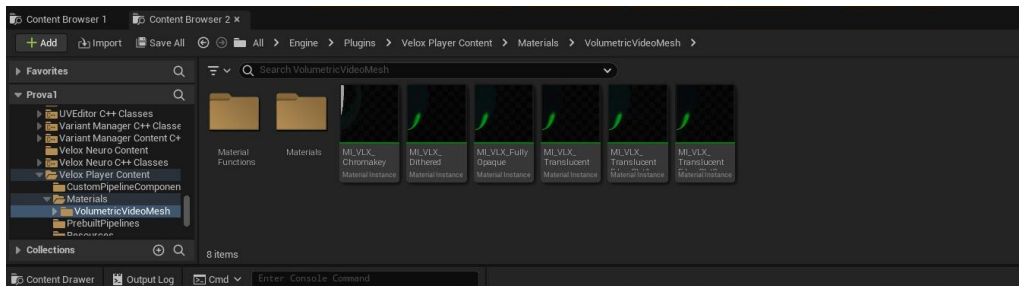


Figura 4.38: Materiali disponibili e utilizzabili con le catture volumetriche realizzate con V3LCamera

D'altra parte, l'algoritmo spesso elimina anche informazioni relative al corpo della persona (figura 4.39). Di conseguenza, il risultato ottenuto, per quanto possa essere migliore rispetto a quello ottenuto con il plugin base, non riesce comunque ad ottenere una qualità ottimale.



Figura 4.39: Eliminazione parti del corpo del soggetto catturato

Un'ulteriore limitazione si individua nella parte superiore della testa. Infatti, i capelli sembrano essere dimezzati di volume. Inoltre, i piedi presentano un difetto ottico in quanto la profondità conferita ad essi non risulta essere corretta, generando nello spettatore l'impressione di bidimensionalità. Ciò ha contribuito ad una diminuzione complessiva del realismo (figura 4.40).



Figura 4.40: Artefatti che generano bidimensionalità nella cattura volumetrica

In generale, il risultato ottenuto possiede poca tridimensionalità e, come nel caso del plugin base, l'angolo di visibilità risulta essere limitato a 60 gradi (figura 4.41).



Figura 4.41: Ricostruzione volumetrica con un angolo di visibilità di 30 gradi a sinistra rispetto alla posizione centrale della camera utilizzata per la registrazione

Le catture volumetriche, ottenute con V3LCamera, rispondono ai cambiamenti di luce dell'ambiente virtuale. Infatti, una modifica alla tinta dell'illuminazione genera un

cambiamento nel colorito della pelle e negli indumenti indossati dal soggetto, come si può notare dalla figura (figura 4.42).



Figura 4.42: Confronto della stessa cattura volumetrica con due illuminazioni differenti a livello di tinta

Il cambiamento dell'intensità luminosa dell'ambiente virtuale genera modifiche sulla texture del soggetto. In particolare, ad alte luminosità si può notare una diminuzione del realismo del soggetto poiché i colori della pelle del soggetto e dei suoi vestiti risultano opachi e poco intensi. Confrontando invece tra di loro le ricostruzioni ottenute a basse e medie luminosità, si possono riscontrare poche differenze. Nella figura (figura 4.43) si confrontano le risposte della ricostruzione volumetrica utilizzando un'illuminazione ambientale costante nei tre diversi casi ma modificando il valore di una point light frontale e di una point light posteriore. In particolare, si analizza una luminosità bassa, media e alta.



Figura 4.43: Confronto della stessa cattura volumetrica con diverse intensità luminosa. A sinistra una bassa luminosità, in centro luminosità media e a destra luminosità alta

La cattura volumetrica è, inoltre, in grado di generare ombre dinamiche nell'ambiente virtuale. Infatti, tali ombre modificano la loro inclinazione e posizione in base a dove viene collocata la fonte luminosa (figura 4.44). Ciò significa che l'ombra acquisita durante la fase di registrazione non ha alcuna influenza in quanto viene completamente eliminata.

Inoltre, la cattura volumetrica posizionata nelle circostanze di materiali riflettenti provocherà riflessioni su di essi. Queste ultime saranno dinamiche e, dunque, seguiranno il movimento del soggetto.



Figura 4.44: Ombre dinamiche e riflessioni

I cambiamenti dinamici dei video volumetrici appena esposti, soprattutto se confrontati con i video bidimensionali, permettono di aumentare le possibilità di utilizzo in quanto sono in grado di rispondere alle variazioni applicate negli ambienti virtuali. Ciò contribuisce ad un maggiore realismo complessivo e ad una migliore integrazione all'interno degli ambienti virtuali. Nonostante tale vantaggio, non risulta possibile applicare VFX direttamente sull'output ottenuto. Tale limitazione è dovuta al fatto che la cattura volumetrica è in formato proprietario ed è possibile inserirla nell'ambiente virtuale solamente tramite il plugin apposito, rendendo difficile la sua manipolazione diretta.

Real time

Utilizzando il plugin open source Velox Neuro, è possibile collegare una webcam e vedere i risultati di ricostruzione volumetrica su Unreal Engine in tempo reale. Questo plugin permette anche di scontornare le figure in tempo reale, eseguendo l'*object detection* e la *matting mask* (figura 4.45).



Figura 4.45: Object detection, matting mask e depth estimation nella cattura volumetrica real time

Il plugin in questione è in grado di eseguire una ricostruzione di diversi soggetti contemporaneamente nel caso in cui la webcam inquadri molteplici persone (figura 4.46).

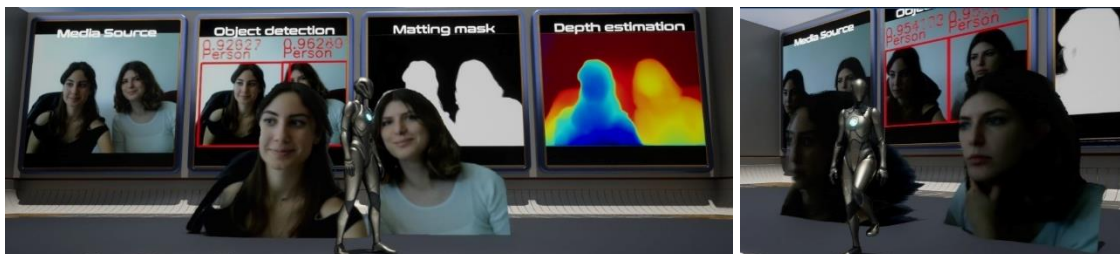


Figura 4.46: Vista frontale e laterale della ricostruzione volumetrica in tempo reale di molteplici soggetti contemporaneamente

D'altra parte, il plugin permette anche di isolare una sola figura abilitando una funzione, disponibile tra le impostazioni. In questo caso, l'algoritmo individua il soggetto più vicino e genera la ricostruzione solo di esso, eliminando tutto ciò che non risulta essere tale soggetto.

Anche in questo caso, i risultati risultano essere buoni solamente con un angolo di mobilità visiva di pochi gradi. Un ulteriore peggioramento della ricostruzione è stato individuato nel momento in cui il soggetto si allontana dalla webcam (figura 4.47) in

quanto la cattura volumetrica risulta essere meno nitida e contenente un numero maggiore di buchi di informazione.



Figura 4.47: A sinistra ricostruzione volumetrica a mezzo busto. A destra ricostruzione volumetrica a figura intera.

Le ulteriori limitazioni individuate risultano essere le medesime riscontrate nella cattura volumetrica offline. Nel caso del real time, però, risultano essere accentuate. Infatti, i bordi del soggetto sono maggiormente ritagliati, effetto dovuto anche all'angolo di visibilità disponibile della webcam. Quest'ultima, infatti, va ad incidere sia sulla risoluzione della cattura volumetrica sia sulla porzione di spazio catturato.

Una grande problematica si riscontra nella ricostruzione delle mani (figura 4.48). Infatti, l'algoritmo non riesce a scontornarle ottimamente se non si trovano di fronte al corpo della persona inquadrata o se le dita sono distanti tra di loro. Ciò che si genera, infatti, è uno scontornamento eccessivo che influenza la qualità della ricostruzione volumetrica complessiva. Una distanza ridotta tra le dita della mano, invece, comporta una riduzione degli artefatti visibili; tuttavia, il risultato che si ottiene non è soddisfacente.



Figura 4.48: Ricostruzione problematica delle mani

Infine, se il soggetto si posiziona lateralmente rispetto alla webcam la ricostruzione volumetrica genererà degli artefatti in prossimità delle sporgenze del viso (figura 4.49), in particolar modo nella zona del naso. Quest'ultimo, infatti, non verrà ricostruito e ciò renderà poco veritiero il profilo del soggetto.



Figura 4.49: Artefatti generati nel profilo del soggetto inquadrato

Per quanto riguarda la reattività del sistema, diversi tentativi hanno reso possibile affermare che la latenza generata è dell'ordine dei millisecondi, risultato accettabile per un sistema in real time.

4.4 Confronti tra tecnologie

Dopo un'analisi delle diverse tecnologie condotta in maniera indipendente, si ritiene opportuno svolgere un approfondimento confrontandole le une con le altre. Dalle considerazioni precedenti si può desumere che Volu permette di ottenere i risultati più realistici; le ricostruzioni sono 3D e non 2.5D, la gestione dinamica delle ombre e delle luci è ottima. Tuttavia, si è deciso di eseguire una ricerca più approfondita, in cui sono stati confrontati video volumetrici registrati a parità di condizioni di illuminazione, movimenti del soggetto e modalità di ripresa. In questo modo, è possibile confrontare in maniera più puntuale Volu, Depthkit e V3LCamera.

In primo luogo, si è deciso di realizzare video volumetrici con movimenti delle braccia lenti. Dalle analisi precedenti è infatti emerso che tutte e tre le tecnologie non riescono a ricostruire in modo ottimale movimenti repentini. Viene però introdotto un elemento di difficoltà in più: il soggetto tiene in mano un microfono. Si desidera valutare come i diversi sistemi reagiscano alle interazioni della persona con oggetti di scena. Inoltre, la persona, in fase di ripresa, pronuncia una breve frase, che permette di studiare come le diverse tecnologie siano in grado di riprodurre i cambiamenti nelle espressioni facciali associati al parlato. Si è deciso di registrare un video con queste caratteristiche, perché simile a quello che potrebbe essere utile a una produzione televisiva.

Infine, i risultati sono stati inseriti nello stesso ambiente virtuale, ma su due motori grafici diversi: i video volumetrici di Volu e Depthkit sono stati importati su Unity, mentre quelli di V3LCamera su Unreal Engine. L'ambiente, grazie a un apposito plugin, è stato esportato da Unreal a Unity e sono state quindi mantenuti uguali l'ambiente, la tipologia di illuminazione e i movimenti di camera del rendering. Qui sotto (figura 4.50) vengono mostrati i risultati di un rendering frontale in cui il soggetto assume una posa simile nei tre casi. Si nota che Volu e Depthkit non riescono a ricostruire bene la mano che impugna il microfono e il microfono stesso. La ricostruzione di Volu è specialmente problematica: sembra che gli algoritmi sfruttati da questa tecnologia riconoscano solo la figura umana e non siano in grado di ricostruire oggetti, che, se interagiscono con la persona, vanno a degradare la qualità del video volumetrico. Il problema di Depthkit, invece, è la difficoltà a stimare correttamente la profondità quando il soggetto tiene in mano il microfono. Al contrario, la ricostruzione realizzata tramite l'app di Velox, V3LCamera, non presenta alcun tipo di difetto (figura 4.51).



Figura 4.50: Da sinistra a destra rendering con camera frontale di Volu, Depthkit e V3LCamera

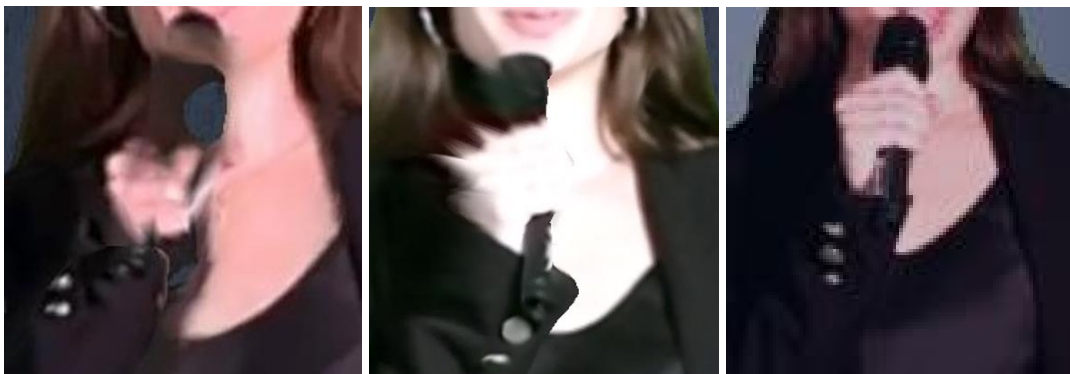


Figura 4.51: Da sinistra a destra dettagli sulla mano che impugna il microfono di Volu, Depthkit e V3LCamera

Si nota inoltre, che nonostante i movimenti non fossero particolarmente veloci, Volu ha alcune difficoltà nel ricostruire la mano sinistra.

Per quanto riguarda le espressioni facciali, dovute ai movimenti della bocca nel parlato, si evidenzia che tutti i sistemi sono stati in grado di ottenere buoni risultati.

Successivamente, l'analisi si è orientata allo studio di movimenti rapidi al fine di comprendere quale tecnologia fosse in grado di generare meno artefatti in tale circostanza.

In particolare, è stato richiesto al soggetto di muovere velocemente la mano, simulando un saluto (figura 4.52). Tale movimento potrebbe essere spesso richiesto all'interno delle produzioni televisive e, per tale motivo, è risultato necessario comprendere quali sono i risultati migliori ottenibili con le tecnologie analizzate.

Volu e Depthkit producono un risultato che presenta notevoli artefatti in corrispondenza della mano. Le due tecnologie, infatti, generano una riproduzione della mano poco verosimile, in quanto viene ricostruita come un blocco unico. In quest'ultimo la separazione tra le dita viene meno, contribuendo a generare una sensazione di poca veridicità del risultato. Inoltre, Volu produce un alone di sfocatura nelle circostanze della mano, probabilmente dovuto al suo rapido movimento. Dall'altra parte, Depthkit restituisce una texture non accurata della mano, aumentando il distacco con la realtà. V3LCamera, invece, produce un risultato migliore ma non perfetto. Infatti, lo scontornamento della mano, soggetta al movimento veloce, viene eseguito correttamente ma presenta delle imperfezioni in alcuni punti. In questi ultimi, è possibile notare come la rimozione del green screen, posto dietro il soggetto, non sia avvenuta in modo preciso, rendendo il risultato meno ottimale (figura 4.53).



Figura 4.52: Da sinistra a destra rendering con camera frontale di Volu, Depthkit e V3LCamera



Figura 4.53: Da sinistra a destra dettagli sulla mano che esegue un movimento veloce di Volu, Depthkit e V3LCamera

Oltre alla ricostruzione frontale, è bene analizzare i risultati ottenuti quando la camera si muove intorno al soggetto (figura 4.54). A seguito delle osservazioni fatte in precedenza, alla camera virtuale è stato fatto compiere in totale un angolo di 60°. Ciò ha permesso di confrontare le varie tecnologie, comprendendo quale fosse maggiormente integrabile all'interno degli ambienti virtuali.



Figura 4.54: Da sinistra a destra rendering con la camera leggermente angolata Volu, Depthkit, e V3LCamera

Si può, dunque, notare ed affermare che le maggiori verosimiglianza e tridimensionalità siano rintracciabili nelle catture volumetriche restituite da Volu. Quest'ultima applicazione incontra alcune difficoltà nella fluidità di movimenti corporei veloci e nell'interazione con degli oggetti, ma rende possibile una maggiore integrazione all'interno degli ambienti virtuali. Di conseguenza, le catture volumetriche ottenute con Volu hanno maggiore potenziale di utilizzo.

4.5 Confronto con il progetto europeo XReco

Dal confronto precedente emerge che la tecnologia migliore, tra quelle analizzate, è Volu. Si è ritenuto poi opportuno confrontare i suoi risultati con alcuni di quelli ottenuti nel progetto *XReco*. Quest'ultimo è un progetto che unisce 20 partner, tra cui la Rai, provenienti da 12 Paesi [58], uniti dalla volontà di trasformare la tecnologia XR. Uno dei principali obiettivi del progetto è l' "holoportation", che dovrebbe permettere di ottenere delle ricostruzioni umane realistiche in real time. Nel progetto, oltre alle tecniche di acquisizione, vengono studiate la compressione e l'ottimizzazione, in diverse condizioni computazionali e di rete. In particolare, il centro di ricerca *i2cat*, con sede in Catalogna, Spagna, ha sviluppato una tecnologia di holoportation. Le informazioni ricavate da tre camere RGB-D sono usate per ottenere rappresentazioni volumetriche realistiche di persone.

Sono state elaborate una serie di soluzioni e strategie per ottenere le ricostruzioni in tempo reale, ma, per il confronto con Volu, saranno usati alcuni risultati ottenuti da *i2cat* non in real time. Sono stati analizzati cinque secondi di clip, in quanto Volu permette di registrare gratuitamente solo per questo tempo. Per permettere un confronto più puntuale, sono stati eseguiti movimenti equivalenti al video di riferimento e si è cercato di ricostruire un'ambientazione simile (figura 4.55).



Figura 4.55: A sinistra ricostruzione volumetrica ottenuta con holoportation. A destra cattura volumetrica ottenuta con Volu.

La ricostruzione ottenuta da *i2cat* mostra una qualità inferiore rispetto a quella di Volu. La riproduzione delle gambe non è completa e i piedi non sono totalmente rappresentati. La testa presenta notevoli artefatti e buchi d'informazione. La figura non è quindi molto realistica, ma mantiene un senso di tridimensionalità. La ricostruzione volumetrica

ottenuta con Volu, invece, è completa e ben integrata all'interno dell'ambiente virtuale. Si creano, però, alcune imperfezioni sul volto quando non è posto di fronte alla camera, mentre la ricostruzione di i2cat non sembra presentare questo problema (figura 4.56).



Figura 4.56: A sinistra ricostruzione del volto di profilo di holoportation. A destra ricostruzione volto di profilo di Volu.

Ciò nonostante, è indubbio che Volu restituisca modelli più realistici e dettagliati, senza buchi d'informazioni. Questo conferisce alle ricostruzioni un alto grado di plausibilità e realismo, aspetti importanti per le produzioni televisive.

Il confronto tra Volu e la tecnologia sviluppata i2cat permette di evidenziare i punti di forza e di debolezza delle due. Volu si afferma come la soluzione ideale per progetti che necessitano di ricostruzioni ad alta qualità, mentre i2cat è interessante per gli studi sulla compressione e sulla cattura volumetrica in tempo reale.

4.6 Gaussian Splatting 4D

Il paper, relativo al Gaussian Splatting 4D, presenta un collegamento al profilo Github in cui è condiviso il codice utilizzabile liberamente dagli utenti. Per riuscire a compilare tale codice con successo, è necessario soddisfare i requisiti relativi al codice del Gaussian Splatting 3D, condiviso su Github. Tra questi è risultato fondamentale l'installazione di Colmap, la cui pipeline rende possibile il processamento delle immagini in ingresso. Infatti, tale pipeline permette di ottenere dei dati su cui l'algoritmo del Gaussian Splatting 4D può lavorare ed effettuare il training. I numerosi problemi sorti, durante l'installazione di Colmap, hanno reso necessario l'utilizzo del Docker Hub. Quest'ultimo ha reso possibile installare Colmap e procedere verso l'utilizzo dell'algoritmo del Gaussian Splatting 4D.

In primo luogo, è stato necessario comprendere quali limitazioni presentasse il processamento delle immagini. Per uno studio iniziale, i vari test effettuati sono stati portati avanti utilizzando una risoluzione delle immagini non superiore a 406x720 con i formati PNG e JPEG. Successivamente, la risoluzione è stata aumentata fino a 1080x1920, consentendo un risultato migliore di ricostruzione. L'algoritmo che effettua il processamento delle immagini, però, è stato in grado di restituire il risultato solo se la camera, in fase di registrazione, effettuava dei movimenti nello spazio e non rimaneva fissa in una sola posizione.

Successivamente, i dati ottenuti dal processamento delle immagini sono stati utilizzati nella fase di training. Il risultato di quest'ultimo può essere visualizzato attraverso due diversi viewer (figura 4.57) o tramite un render, con le limitazioni esplicitate precedentemente.

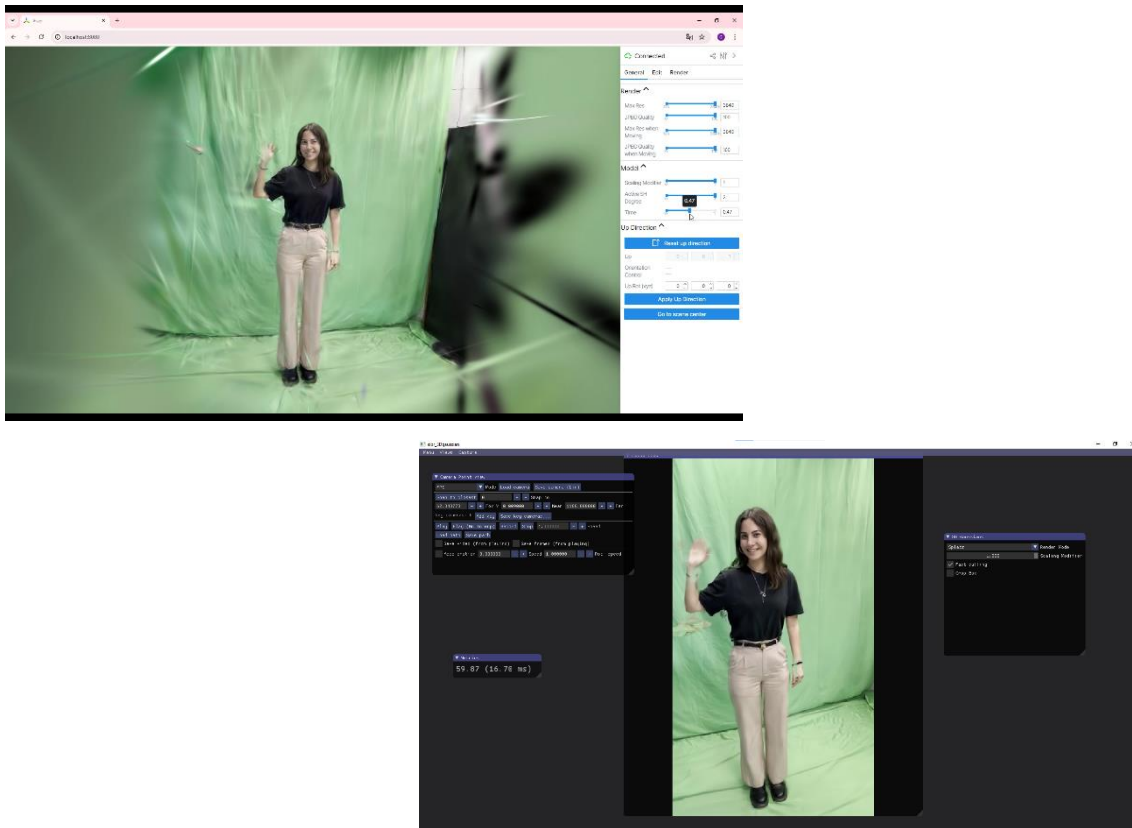


Figura 4.57: Viewer dinamico e statico

Dopo aver compreso le varie limitazioni a cui prestare attenzione, l'analisi si è inizialmente focalizzata a comprendere le potenzialità dell'algoritmo analizzando la ricostruzione di vari oggetti in movimento (figura 4.58). La ricostruzione di questi ultimi ha ottenuto dei risultati buoni se il movimento delle mani, che permetteva l'animazione degli oggetti in scena, non risultava essere troppo veloce. Inoltre, si è potuto dedurre che un contrasto abbastanza elevato tra gli oggetti può influire positivamente sulla loro ricostruzione.



Figura 4.58: A sinistra render con movimenti veloci. A destra render con movimenti controllati.

Successivamente, la ricerca è stata orientata verso la ripresa di persone intere partendo da un'acquisizione effettuata con la videocamera di un Iphone 13. In questo ultimo caso, è stato sfruttato un green screen posto nella parte posteriore al soggetto. Se il soggetto si muoveva velocemente, la ricostruzione perdeva di concretezza e fedeltà. Ciò si è potuto verificare soprattutto nel movimento delle braccia e delle mani (figura 4.59).



Figura 4.59: Degradazione delle informazioni visive a causa di movimenti veloci delle braccia e delle mani

Se il soggetto, invece, effettuava dei movimenti più controllati l'algoritmo riusciva a ricostruire in maniera più fedele il movimento e, dunque, si riusciva ad ottenere un render più realistico (figura 4.60).



Figura 4.60: Ricostruzione più fedele delle braccia e delle mani a causa di movimenti più controllati

Successivamente, si è cercato di indirizzare l'analisi verso la ricerca di alcune soluzioni utili a gestire i movimenti corporei veloci. A tale fine, sono stati analizzati i vari parametri

modificabili. Infatti, il training si basa su alcuni parametri, definiti iperparametri modificabili, dichiarati in un file python. Gli *iperparametri* sono valori che caratterizzano un modello ed influiscono sul funzionamento e sulle prestazioni del modello stesso. Quando l'intero training set è sottoposto al modello, allora si ha una *epoch*. Il training set potrebbe essere troppo grande per essere elaborato tutto in una volta e, per tale motivo, si può suddividere il training set in sottogruppi uniformi, chiamati *batch*. Il numero di esempi contenuti in ogni batch è chiamato *batch size*. Il numero di batch necessari a completare un epoch viene, invece, definito *iterazione*. Il numero di epoch ed il batch size influiscono sulla velocità di addestramento di un modello, ma anche sul suo modo di perfezionarsi [59]. La batch size è uno degli iperparametri più importanti nell'addestramento al deep learning e ha un impatto diretto sull'accuratezza e sull'efficienza computazionale del processo di addestramento [60]. È stato, dunque, necessario effettuare dei test per poter comprendere come e quanto questi iperparametri influissero sul risultato finale.

Inizialmente, è stata incrementata la batch size e successivamente è stato eseguito il training con tale cambiamento. La differenza più sostanziale può essere individuata nella ricostruzione delle mani, come si può osservare nella figura (figura 4.61). Infatti, il frame a sinistra risulta essere ottenuto con un valore di batch size minore (pari a 2) rispetto al valore di batch size del frame di destra (pari a 3). Tale modifica non è andata ad incidere sui tempi di training.



Figura 4.61: A sinistra ricostruzione batch size pari a 2. A destra batch size pari a 3.

In un secondo momento, è stato effettuato un ulteriore test con l'aumento del valore della batch size e con l'aumento del numero di iterazioni. Anche in questo caso, la differenza maggiormente visibile risulta essere nella ricostruzione delle mani in movimento. Nella

figura (figura 4.62), infatti, si può vedere un confronto in cui a sinistra si trova il render dell'immagine senza modifiche dei parametri (batch size pari a 2 e numero di iterazioni pari a 14 000) mentre a destra il risultato ottenuto aumentando il valore di batch size (pari a 20) e di iterazioni (pari a 20 000). Nonostante le modifiche degli iperparametri abbiano permesso di ottenere risultati migliori, i tempi relativi al training hanno registrato un aumento di due ore. Ciò potrebbe limitarne l'utilizzo in contesti televisivi, in cui la rapidità di elaborazione è cruciale.



Figura 4.62:A sinistra ricostruzione batch size pari a 2 e iterazioni pari a 14 000. A destra batch size pari a 20 e iterazioni pari a 20 000.

Le ricostruzioni, fino a questo momento analizzate, hanno utilizzato dei video in ingresso raffiguranti un soggetto posto davanti ad un green screen. Nonostante ciò, non è stata effettuata alcuna operazione volta a separare il soggetto dallo sfondo. Infatti, l'algoritmo del Gaussian Splatting 4D sfruttato non prevede la possibilità di effettuare rotoscoping: ciò che viene restituito è un ambiente tridimensionale rappresentante tutto ciò che è stato catturato durante la fase di ripresa iniziale. Per tale motivo, si è cercata una soluzione manuale per poter ottenere unicamente la ricostruzione del soggetto desiderato. In primo luogo, è stata effettuata un'operazione di *chroma key* sul video originale. Successivamente, si è cercato di effettuare il processamento dei frames di tale video tramite la pipeline di Colmap. Quest'ultima operazione non è andata a buon fine in quanto Colmap necessita di un video in ingresso da cui poter ricavare le informazioni utili alla ricostruzione dello spazio tridimensionale circostante e dei movimenti di camera. Per tale motivo, è stato eseguito successivamente un processamento del video originale senza scontornamento tramite la pipeline di Colmap. I dati della scena tridimensionale ottenuti sono stati mischiati con le immagini scontornate e tutte queste informazioni sono state fornite all'algoritmo del Gaussian Splatting 4D. Quest'ultimo ha comunque ricostruito

tutto l'ambiente catturato durante la registrazione e i pixels che, nelle immagini in ingresso, erano trasparenti sono stati renderizzati con il colore nero (figura 4.63). Tali tentativi, hanno fatto emergere una limitazione non valicabile di tale algoritmo, almeno per il momento.

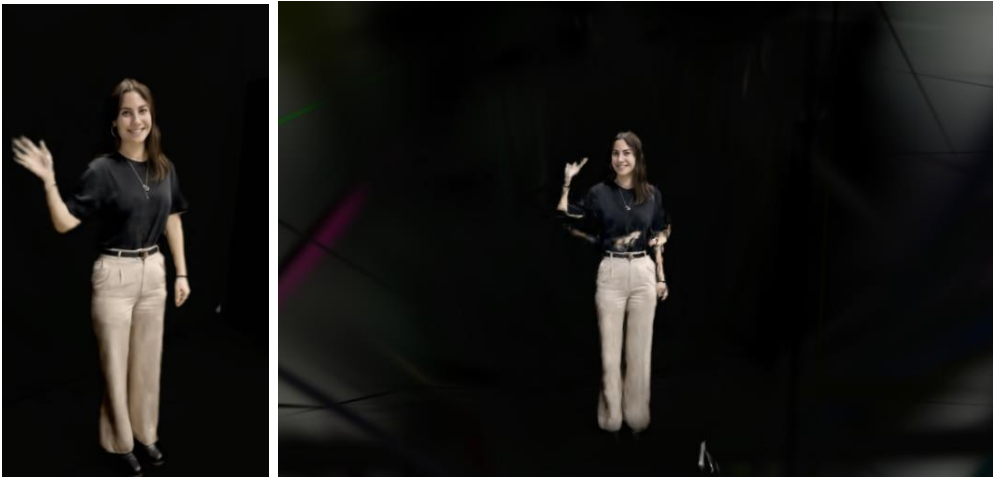


Figura 4.63: Render della ricostruzione ottenuta con il canale alpha e vista dal viewer dinamico

In seguito, i test sono stati orientati alla possibilità di girare intorno al soggetto durante le riprese per comprendere se, in questo modo, il soggetto venisse ricostruito più fedelmente anche nella parte posteriore. In questo caso, non è stato utilizzato alcun green screen. Il risultato volumetrico, visionabile dal viewer, non è stato ottimale in quanto la parte posteriore non è stata sempre migliorata negli istanti temporali in cui non era visibile dalla videocamera. Inoltre, il movimento più rapido della camera, per poter girare intorno al soggetto, ha peggiorato i risultati.

I primi test sono stati effettuati muovendo la camera intorno al soggetto di 180 gradi, ottenendo una ricostruzione molto approssimativa della parte posteriore del soggetto. In seguito, sono stati eseguiti alcuni test muovendo la camera intorno al soggetto ripreso di 360 gradi. La ricostruzione della parte posteriore della persona inquadrata risulta essere maggiormente realistica, anche negli istanti temporali in cui la camera non inquadra tale porzione del soggetto e affida la ricostruzione interamente all'algoritmo di AI (figura 4.64).



Figura 4.64: A sinistra ricostruzione posteriore con movimento di camera di 180 gradi nel viewer. A destra ricostruzione posteriore con movimento di camera di 360 gradi nel viewer.

Inoltre, durante questi test è stato richiesto al soggetto ripreso di muovere la testa e lo sguardo, ottenendo dei risultati poco accettabili nella ricostruzione (figura 4.65). Gli artefatti generati sono dovuti al fatto che sia il soggetto inquadrato che la camera compivano un movimento notevole. Per tale motivo, si è reso necessario in seguito far mantenere lo sguardo in camera durante le riprese. In aggiunta a ciò, è stato notato che anche il movimento veloce delle braccia e delle mani generava notevoli problemi in fase di ricostruzione e ciò ha spinto la successiva ricerca verso movimenti più controllati.

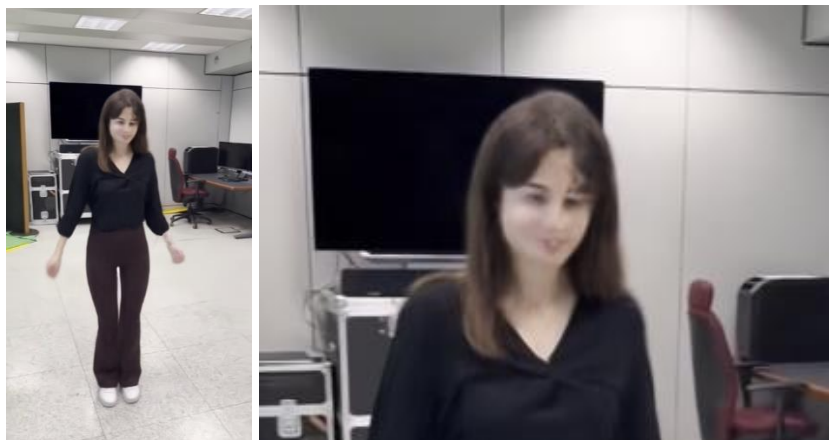


Figura 4.65: Degradazione delle informazioni nella ricostruzione del volto.

L'analisi successiva è stata condotta rivolgendo lo sguardo verso l'utilizzo di camere professionali. Infatti, è stata utilizzata una Sony $\alpha 7R$ IV per comprendere se ciò potesse influenzare la bontà del risultato finale. La ricostruzione, in tale caso, presentava colori più fedeli alla realtà e un maggior numero di dettagli. Ciò è dovuto al fatto che con una videocamera professionale si possa avere maggiore libertà sui parametri fotografici (ISO, apertura diaframma, tempo di otturazione, bilanciamento del bianco, ecc...) e un sensore

di dimensione maggiore. Per quanto il risultato ottenuto possa essere ritenuto più soddisfacente, i tempi di processamento e di training sono aumentati.

Al fine di avere un confronto abbastanza veritiero, sono state effettuate le riprese sia con la camera dell'Iphone 13 sia con la camera Sony α 7R IV (figura 4.66) cercando di eseguire, per quanto possibile, gli stessi movimenti di camera e facendo compiere al soggetto gli stessi spostamenti.



Figura 4.66: A sinistra render ricostruzione video registrato con l'Iphone 13. A destra render video registrato con la Sony α 7R IV

Nonostante i miglioramenti appena esposti, le ricostruzioni presentano comunque delle limitazioni nel caso di movimenti veloci (figura 4.67). Tale problematica, dunque, non risiede nella minore risoluzione del video in ingresso ma rappresenta una limitazione non superabile. L'unico modo per risolvere questo problema è limitare i movimenti veloci durante la fase di registrazione.



Figura 4.67: Render movimenti veloci delle braccia e delle mani

Analizzando lati positivi e negativi riscontrati nell'utilizzo della camera professionale, si è scelto di indirizzare l'analisi successiva verso l'utilizzo costante di video registrati con la Sony $\alpha 7R$ IV.

Ulteriori problematiche che sono state analizzate riguardano gli indumenti che il soggetto potrebbe indossare durante le riprese. In primo luogo, lo studio si è focalizzato sulla ricostruzione di vestiti larghi e particolarmente soggetti al movimento. In tale caso, l'algoritmo del Gaussian Splatting ha reso possibile una ricostruzione quasi perfetta non generando praticamente nessuna deformazione, come si può vedere nella ricostruzione della gonna in figura (figura 4.68).



Figura 4.68: Render della ricostruzione del movimento della gonna

In secondo luogo, si sono svolti dei test sulla ricostruzione delle superfici riflettenti, tipiche degli occhiali da vista o da sole. Anche in questo caso, l'algoritmo del Gaussian Splatting permette di ottenere risultati soddisfacenti. I test, eseguiti per tale scopo, hanno sfruttato dei video in ingresso contenenti molteplici superfici di materiale riflettente, come si può vedere in figura (figura 4.69). In quest'ultima è chiaramente visibile come sia le lenti degli occhiali da vista sia le finestre abbiano ottenuto una ricostruzione ottimale.

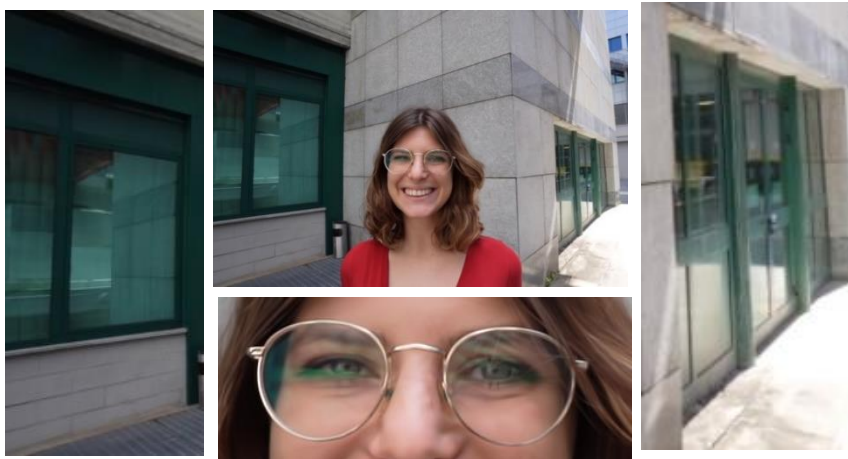


Figura 4.69: Render della ricostruzione di superfici riflettenti

In questo ultimo test è stato possibile, inoltre, effettuare un'analisi rivolta alle espressioni facciali. Queste ultime, infatti, hanno ottenuto un'ottima ricostruzione, probabilmente dovuta al fatto che il viso era ben illuminato e che lo sguardo del soggetto non facesse notevoli spostamenti.

Le tempistiche necessarie ad ottenere tali risultati non sono indifferenti in quanto l'algoritmo del Gaussian Splatting 4D è di recente pubblicazione e, quindi, non ancora completamente ottimizzato. Infatti, si prevede che nel futuro si presenteranno notevoli miglioramenti nelle prestazioni e nelle tempistiche. I vari video, di cui è stata eseguita una ricostruzione tramite tale algoritmo, hanno subito una fase di processamento e una fase di training. Il processamento delle immagini, eseguito tramite Colmap, ha avuto tempistiche abbastanza variabili, ma sempre inferiori rispetto a quelle richieste per il training. Per rendere più concreta tale asserzione, si può considerare un video di 5 secondi registrato a 30 fps, ovvero 150 frames in ingresso. Il processamento di tale video è stato eseguito in 35 minuti mentre il training in 40 minuti tramite la scheda grafica NVIDIA RTX A5000. Il video in questione riprendeva un soggetto che eseguiva una breve camminata in un luogo interno.

La fase di render, invece, risulta essere notevolmente più veloce permettendo di ottenere il risultato in quasi sempre meno di due minuti.

Si sono inizialmente confrontati due video con una risoluzione iniziale pari a 406x720, renderizzati con la scheda grafica NVIDIA RTX A5000. Il primo video, contenente 55 mila gaussiane circa, ha raggiunto 35.6 FPS durante la fase di rendering. L'ulteriore video analizzato presentava quasi 59 mila gaussiane e gli FPS di rendering registrati sono stati

pari a 34 circa, permettendo di osservare una diminuzione degli FPS del 4.5% (figura 4.70). Tale analisi permettere di mettere in luce una correlazione fondamentale: il numero di gaussiane incide sugli FPS, ovvero sulla velocità con cui si riesce ad ottenere il render. Dunque, un numero di punti maggiore aumenta le tempistiche e diminuisce gli FPS, influenzando negativamente il processo di rendering.

```
point nums: 55318 [26/07 15:06:50]  
Rendering progress: 100%  
FPS: 35.646194174606244 [26/07 15:06:56]  
  
point nums: 58694 [26/07 15:17:25]  
Rendering progress: 100%  
FPS: 33.72216749611151 [26/07 15:17:36]
```

Figura 4.70: In alto FPS rendering del video con 55 mila gaussiane. In basso FPS rendering del video contenente 58 mila gaussiane.

Un'ulteriore variazione dei tempi di rendering e dei relativi FPS è stata riscontrata nei video registrati tramite una camera professionale. Infatti, è stato eseguito il render di due video di circa 10 secondi, uno registrato con la camera dell'Iphone e l'altro con la fotocamera Sony α 7R IV. In questi due video sono state replicati gli stessi movimenti di camera e degli attori e la registrazione è avvenuta nello stesso ambiente. Il video registrato con la camera dello smartphone, con una risoluzione 406x720 a 30 fotogrammi al secondo, ha eseguito il render a circa 15 FPS. D'altra parte, il video ottenuto con la camera professionale, con una risoluzione 1920x1080 a 30 fotogrammi al secondo, è stato renderizzato a circa 6 FPS. Tale analisi permette di affermare che un aumento nella risoluzione del video iniziale incide notevolmente sulle prestazioni di rendering.

Capitolo 5

Risultati sperimentali

In questo capitolo vengono valutate le varie soluzioni tecnologiche adottate tramite l'utilizzo di metriche soggettive e oggettive. Al fine di valutare soggettivamente le catture volumetriche ottenute, sono stati eseguiti dei test soggettivi per tutte e quattro le tecnologie. La valutazione oggettiva, invece, è stata possibile effettuarla unicamente per il Gaussian Splatting 4D. Gli esiti ottenuti, in entrambe le casistiche, hanno permesso di evidenziare quale tecnologia riesca ad ottenere i risultati migliori e quale i peggiori.

5.1 Analisi soggettive

La ricerca successiva si è focalizzata sul comprendere la bontà dei risultati ottenuti tramite la somministrazione di alcuni test soggettivi. Un importante obiettivo di questi ultimi si può individuare nel comprendere se le catture volumetriche fossero utilizzabili per una produzione televisiva. Per tale motivo, si è deciso di integrare le catture volumetriche in uno studio televisivo virtuale. I risultati tridimensionali ottenuti dal Gaussian Splatting 4D, invece, sono stati testati senza nessun inserimento in un ambiente virtuale a causa della mancata possibilità di integrazione con motori grafici. Infine, si è scelto di analizzare le possibilità offerte dalla cattura volumetrica in tempo reale e quali vantaggi e svantaggi apportassero.

5.1.1 Setup di cattura

Setup di cattura Volu, Depthkit e V3LCamera

In primo luogo, è stato necessario effettuare la registrazione dei video su cui sarebbero stati effettuati i test soggettivi. A tale scopo, è stata utilizzato un Iphone 13 PRO, posizionato su un cavalletto, su cui erano state scaricate le applicazioni Volu e V3LCamera (figura 5.1). Per le catture volumetriche realizzate con l'applicazione Depthkit, invece, è stata utilizzata la Kinect v2, collegata a un PC su cui è stato possibile memorizzare le registrazioni effettuate.

Il soggetto è stato posizionato frontalmente rispetto allo strumento di cattura ed è stato posto davanti ad un green screen, necessario per il successivo rotoscoping. In seguito, sono stati posizionati due pannelli LED SuperPanel Dual-Color Lupo (figura 5.1) a $+60^\circ$ e -60° circa rispetto alla posizione centrale dello strumento di cattura, entrambe ad una temperatura di 5600 K con una percentuale di dimmer pari a 100.



Figura 5.1: Setup per i video per i test di Volu, V3LCamera e Depthkit

In tutti e tre i casi lo strumento di cattura non è stato sottoposto a nessun movimento durante la fase di registrazione. Tale scelta è stata dovuta al fatto che, nella fase di analisi precedente, i risultati migliori sono stati ottenuti quando lo strumento di cattura rimaneva fisso.

Durante la fase di registrazione, al soggetto è stato richiesto di eseguire dei movimenti controllati e semplici per riuscire ad ottenere una cattura volumetrica migliore.

Sono state realizzate tre diverse registrazioni: una per ogni applicazione considerata. Tale scelta è stata dovuta dal fatto che non è stato possibile effettuare una singola registrazione e processarla in seguito nei vari software in quanto ogni applicazione richiedeva una registrazione interna all'applicazione.

Setup di cattura Gaussian Splatting 4D

La registrazione dei video, per il successivo processamento tramite l'algoritmo del Gaussian Splatting 4D, ha richiesto un setup differente. In primo luogo, non si è potuto mantenere lo strumento di cattura fisso ma si sono dovuti effettuare minimi movimenti nell'ambiente per rendere possibile la raccolta delle informazioni tridimensionali sullo spazio, necessarie per il processamento dei video. Inoltre, è stata utilizzata una fotocamera Sony $\alpha 7R$ IV (figura 5.2) in quanto il precedente studio sull'algoritmo in questione ha

presentato notevoli miglioramenti utilizzando una fotocamera professionale rispetto a una fotocamera di uno smartphone.

Per la realizzazione dei test soggettivi, relativi alla valutazione dell'algoritmo del Gaussian Splatting 4D, è stato necessario registrare quattro differenti video. Infatti, in tal caso, è stato fondamentale valutare la bontà delle ricostruzioni in situazioni critiche. Due video sono stati registrati in interno; in uno di questi il soggetto è stato posto davanti ad un green screen, mentre l'altro video registrato in interno presenta un soggetto che si muove liberamente nello spazio. Nel video in cui vi è la presenza del green screen sono stati utilizzati due pannelli LED SuperPanel Dual-Color Lupo a $+60^\circ$ e -60° circa rispetto alla posizione centrale della fotocamera (figura 5.2).



Figura 5.2: A sinistra setup per uno dei video realizzato per il test del Gaussian Splatting. A destra la fotocamera professionale utilizzata.

Gli ulteriori due video, invece, sono stati realizzati in esterno permettendo di valutare le possibili problematiche introdotte dagli agenti esterni (luce solare, riflessioni, ecc...). Ognuno di questi video introduce una difficoltà per la ricostruzione da parte del Gaussian Splatting 4D. In particolar modo, è stato possibile valutare la ricostruzione di movimenti veloci di gamba e braccia, riflessioni, espressioni facciali e movimento dei vestiti.

5.1.2 Setup test

Una volta ultimate le riprese, si è deciso di unire i diversi risultati in un unico video per rendere più fruibili i test. Le ricostruzioni volumetriche ottenute con Volu, Depthkit e V3LCamera vengono inizialmente presentate uno di seguito all'altro, intervallate da qualche secondo di schermata grigia per segnalare la transizione. Quest'ultima permette all'occhio di rilassarsi e ridurre l'influenza visiva del risultato precedente su quello successivo [61]. In questa prima fase, i video, seppur brevi (5 secondi ciascuno), vengono mostrati una sola volta. Successivamente, viene introdotto ogni risultato con un breve titolo su schermata grigia (figura 5.3), ad esempio "Clip 1", e, immediatamente dopo, mostrato il video in loop per due volte. La durata eccessivamente breve, infatti, non permetterebbe a chi visiona il video di analizzarlo a fondo, trovarne eventuali difetti e pregi. Per questo motivo, si è deciso di mostrarlo per due volte. Alla fine del video in loop, appare sullo schermo un codice QR (figura 5.4), da inquadrare con lo smartphone, che rimanda alla pagina di Google Forms in cui vengono poste sei domande relative al video appena visto. Questo processo viene ripetuto per i risultati ottenuti con le tre diverse tecnologie, mantenendo invariate le domande del questionario.



Figura 5.3: A sinistra schermata con titolo iniziale. A destra risultato proposto

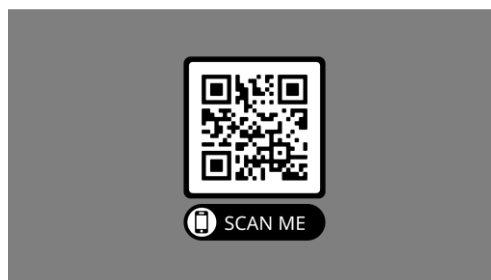


Figura 5.4: Schermata con QR code da scannerizzare

In seguito, vengono proposti i risultati ottenuti con il Gaussian Splatting 4D. Come visto in precedenza, non è ancora possibile importare le ricostruzioni ottenute su motori grafici, e, di conseguenza, non è possibile inserire gli output del 4D-GS nell'ambiente virtuale

usato negli altri casi. Si è deciso, allora, di cambiare la modalità di test: in questo caso, vengono confrontati il video RGB registrato con la camera e la sua ricostruzione con le gaussiane. Viene quindi mostrato il video originale (figura 5.6), introdotto opportunamente da una schermata grigia con un titolo (figura 5.5) e poi il video ricostruito (figura 5.6), separato dal precedente sempre dalla scritta su sfondo grigio. A questo punto, compare la schermata con il QR code con i questionari. Questo processo viene ripetuto per i quattro video. Le domande poste non sono sempre le stesse, ma in ciascun video si sono voluti analizzare diversi aspetti e, di conseguenza, sono stati posti quesiti mirati.



Figura 5.5: Esempio di schermata d'introduzione

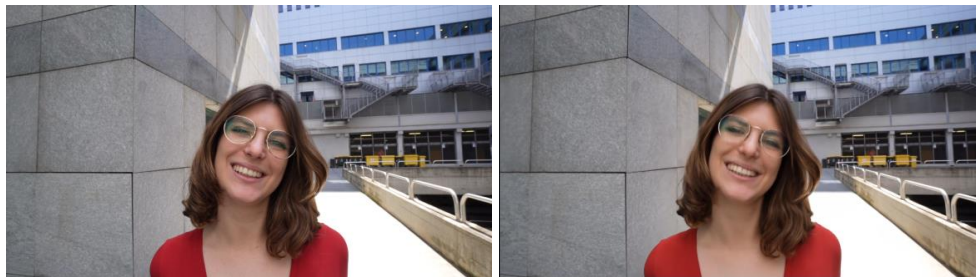


Figura 5.6: A sinistra video originale. A destra video ricostruito

Infine, il test presenta una parte più interattiva in cui viene richiesto di provare la cattura volumetrica in tempo reale. È stata usata una webcam Logitech Streamcam che registra a 60 fps, ha come risoluzione 1080p e apertura f/2.0. Chi esegue il test, vedrà inserita la propria ricostruzione in real time in un ambiente virtuale e potrà muoversi liberamente, avvicinarsi e allontanarsi dalla webcam e valutare come il sistema risponda e, in seguito, rispondere a un questionario con domande specifiche per la cattura in tempo reale.

I test sono stati svolti da 20 persone, di genere sia maschile che femminile, all'interno degli uffici di Rai CRITS. A tutti è stata fornita una breve introduzione sull'argomento di ricerca e una descrizione più approfondita su cosa dovessero osservare e valutare nelle varie fasi del test. Per i questionari si è deciso di usare la *Absolute Category Ranking (ACR)*, in cui ci sono cinque livelli della scala in ordine decrescente (vedi appendice):

Excellent, Good, Fair, Poor, Bad [62]. Per quanto riguarda il questionario per la cattura volumetrica in real time, invece, si è deciso di introdurre anche una domanda aperta con l'intento di lasciare maggiore libertà di espressione agli utenti, perché, in quest'ultimo test, gli utenti potevano agire più liberamente, scegliendo i movimenti da eseguire.

5.1.3 Discussione risultati

5.1.3.1 Volu, Depthkit e V3LCamera

Volu

La ricostruzione di Volu (figura 5.7) è stata inserita in un ambiente virtuale su Unity; è stato aggiunto un movimento di camera con un angolo totale di circa 60 gradi. Le valutazioni dei partecipanti ai test, che hanno osservato il risultato da diversi punti di vista, hanno evidenziato punti di forza e di debolezza della tecnologia.

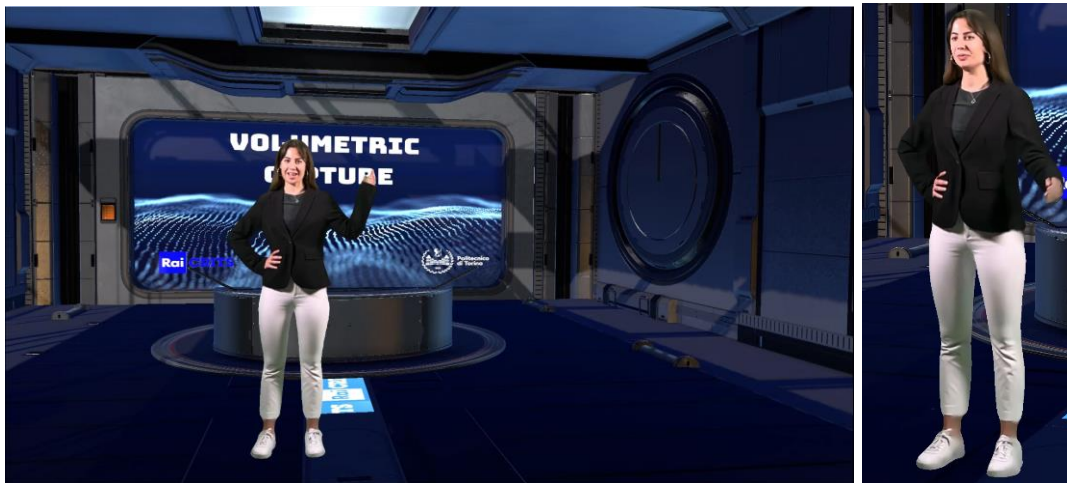


Figura 5.7: Ricostruzione di Volu

Il risultato peggiore è stato individuato nella ricostruzione delle braccia e delle mani in movimento. Infatti, il 15% dei soggetti ha valutato scarsa tale ricostruzione e solamente il 5% l'ha giudicata eccellente (grafico 5.1).

Risultati sperimentali

Come valuti il movimento delle braccia e delle mani?

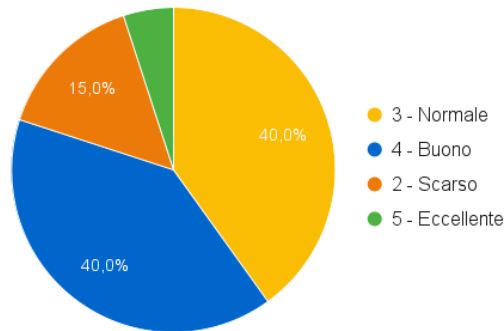
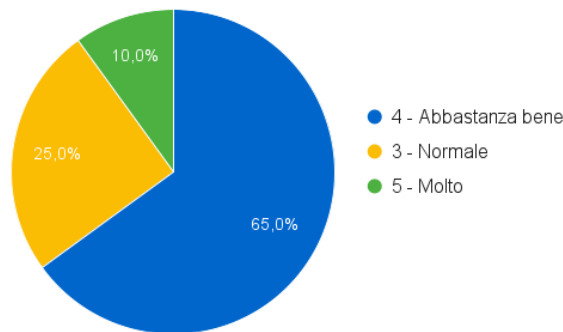


Grafico 5.1: Quesito di valutazione sulla ricostruzione del movimento delle braccia e delle mani

L'integrazione del soggetto con l'ambiente virtuale e la ricostruzione delle espressioni facciali hanno ottenuto risultati notevolmente buoni (grafico 5.2). Infatti, si può notare nella figura come nessun voto sia ricaduto nei due livelli più bassi.

Quanto è integrato il soggetto con l'ambiente circostante?



Come valuti le espressioni facciali?

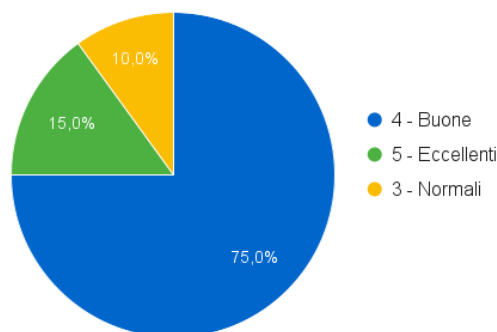


Grafico 5.2: In alto quesito di valutazione sull'integrazione del soggetto con l'ambiente virtuale. In basso quesito di valutazione sulla ricostruzione delle espressioni facciali.

Ciò rappresenta un ottimo raggiungimento in quanto questi due aspetti conferiscono un maggiore realismo complessivo e, di conseguenza, una maggiore possibilità di utilizzo. Infatti, il 40% dei tester ha affermato di trovare molto realistica la ricostruzione volumetrica del soggetto, mentre il 50% l'ha ritenuta abbastanza realistica (grafico 5.3). Tali percentuali permettono di affermare che, per il 90% delle persone sottoposte al test, il realismo è una caratteristica che contraddistingue la cattura volumetrica ottenuta dall'applicazione Volu.

Quanto è realistico il soggetto?

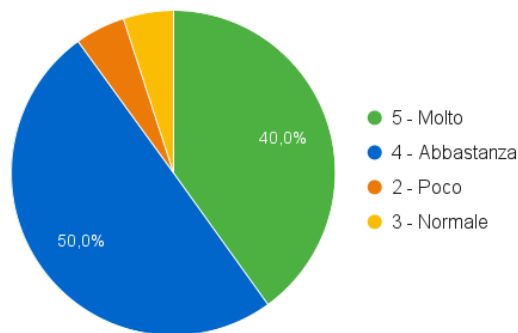


Grafico 5.3: Quesito di valutazione sul realismo del soggetto ricostruito

I risultati dei vari aspetti, analizzati dai soggetti sottoposti al test, permettono di affermare che tale applicazione produce un risultato facilmente integrabile in una produzione televisiva. Tale asserzione è ulteriormente confermata da un quesito posto ai tester stessi in cui nessuno ha ritenuto impossibile utilizzare la cattura volumetrica in contesto televisivo (grafico 5.4).

Potrebbe funzionare per una produzione televisiva?

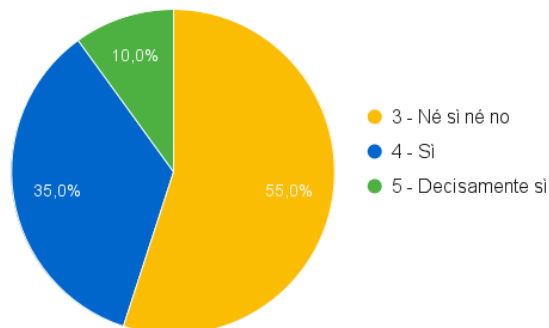


Grafico 5.4: Quesito di valutazione sulla possibilità di utilizzo in una produzione televisiva.

I risultati dei test soggettivi, relativi all'applicazione Volu, hanno ottenuto dei risultati molto buoni. Considerando la scala ACR utilizzata, si può individuare come nessun quesito abbia riportato una percentuale di voto nel livello più basso. Questa considerazione porta a concludere che l'applicazione riesce ad ottenere una risposta positiva nei diversi aspetti considerati.

Depthkit

Il risultato restituito da Depthkit (figura 5.8) è stato inserito nello stesso ambiente di Volu, su Unity. La camera compie, anche in questo caso, un angolo di circa 60 gradi intorno alla figura.

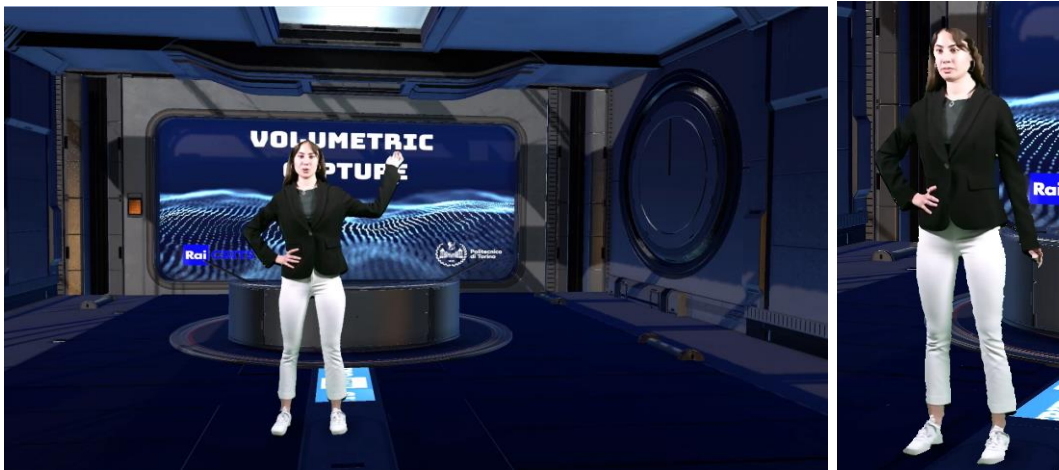


Figura 5.8: Ricostruzione di Depthkit

Il video di Depthkit è stato valutato 'pessimo' dal 75% dei tester e 'normale' dal 25% (grafico 5.5). In tale domanda, nessun soggetto sottoposto al test ha dato una risposta positiva.

Come valuti in generale il video?

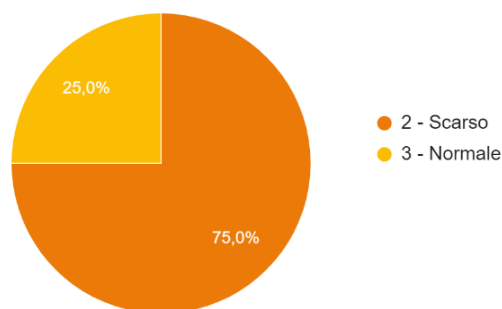
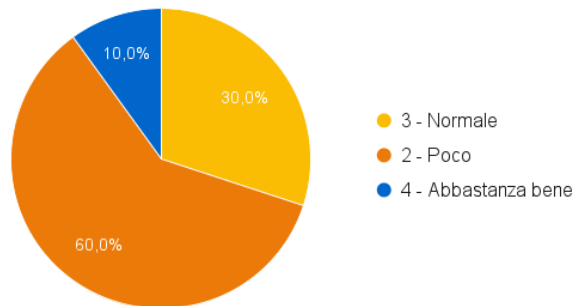


Grafico 5.5: Quesito di valutazione sul video in generale

L'integrazione della cattura volumetrica con l'ambiente circostante è stata ritenuta bassa dal 60% dei tester mentre una piccola percentuale, ovvero il 10%, l'ha ritenuta abbastanza buona (grafico 5.6). Per quanto riguarda le espressioni facciali, la maggioranza dei tester è rimasta neutrale nella risposta, ma nessuno le ha valutate né ottime né pessime (grafico 5.6). In entrambi i quesiti, però, si ha uno sbilanciamento generale delle risposte verso il negativo.

Quanto è integrato il soggetto con l'ambiente circostante?



Come valuti le espressioni facciali?

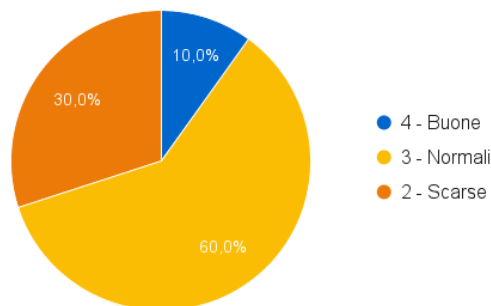


Grafico 5.6: In alto quesito di valutazione sull'integrazione del soggetto con l'ambiente virtuale. In basso quesito di valutazione sulla ricostruzione delle espressioni facciali.

L'analisi di tali risposte porta a concludere che Depthkit non riesce a restituire un risultato con grandi prospettive di utilizzo. In particolar modo, il realismo non è un elemento che caratterizza la cattura volumetrica ottenuta con Depthkit. Infatti, tale cattura volumetrica è stata ritenuta utilizzabile, nel modo in cui è stata presentata, per una produzione televisiva solamente dal 10% dei tester (grafico 5.7). La maggioranza, al contrario, si è espressa in maniera negativa a riguardo.

Potrebbe funzionare per una produzione televisiva?

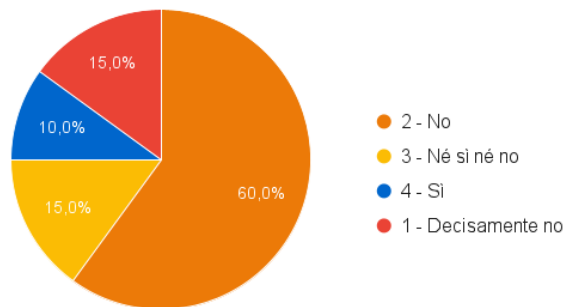


Grafico 5.7: Quesito di valutazione sulla possibilità di utilizzo in una produzione televisiva.

In conclusione, la valutazione soggettiva dell'applicazione di Depthkit non ha prodotto risultati molto positivi. Infatti, in nessun quesito sono state riscontrate risposte nel livello più alto, corrispondente ad una valutazione eccellente. Si potrebbe, dunque, pensare di utilizzare la cattura volumetrica in questione in un contesto differente, magari mascherando i difetti tramite l'utilizzo di effetti visivi.

V3LCamera

L'output di V3LCamera (figura 5.8) è stato inserito in un ambiente equivalente a quello di Volu e Depthkit, ma su Unreal Engine. Oltre agli oggetti di scena, anche il movimento di camera è stato preservato.



Figura 5.9: Ricostruzione con V3LCamera

Nella valutazione del realismo del soggetto si possono individuare percentuali di risposte in ogni livello, ma la metà dei tester ha ritenuto poco realistico il soggetto. Solamente il 5% dei soggetti sottoposti al test ha ritenuto che il soggetto della cattura volumetrica avesse un realismo molto alto (grafico 5.8).

Quanto è realistico il soggetto?

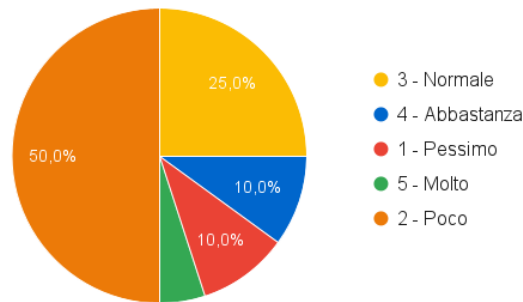


Grafico 5.8: Quesito di valutazione sul realismo del soggetto ricostruito

Per quanto riguarda le espressioni facciali, si può notare come il giudizio su di esse non sia completamente negativo. Infatti, il 25% dei tester le ha valutate buone e meno della metà dei soggetti del test (il 45%) ha ritenuto, invece, che fossero scarse (grafico 5.9). Infine, la valutazione del movimento delle braccia e delle mani ha ottenuto una valutazione peggiore, ottenendo un 15% di risposte nel livello di valutazione più bassa della scala ACR (grafico 5.10).

Come valuti le espressioni facciali?

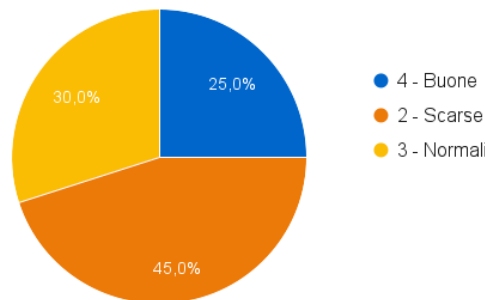


Grafico 5.9: Quesito di valutazione sulla ricostruzione delle espressioni facciali.

Come valuti il movimento delle braccia e delle mani?

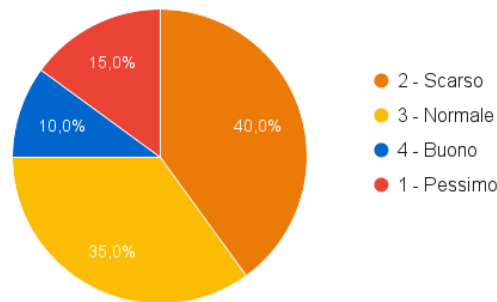


Grafico 5.10: Quesito di valutazione sulla ricostruzione del movimento delle braccia e delle mani

L'applicazione V3LCamera, dunque, permette di ottenere dei risultati volumetrici poco integrabili con gli ambienti virtuali, ma che permettono una ricostruzione dei dettagli abbastanza buona, ad esempio le espressioni facciali. L'utilizzo per una produzione televisiva, però, richiede che la cattura volumetrica rispetti determinate caratteristiche, tra cui anche il realismo e l'integrabilità con ambienti virtuali. A tal proposito, il 45% dei tester ha affermato che questa cattura volumetrica non potrebbe essere, in maniera più assoluta, utilizzata in un contesto televisivo. Il 20% ha, invece, ritenuto che i vari difetti riportati dalla cattura volumetrica non andassero ad incidere sul possibile utilizzo in televisione (grafico 5.11).

Potrebbe funzionare per una produzione televisiva?

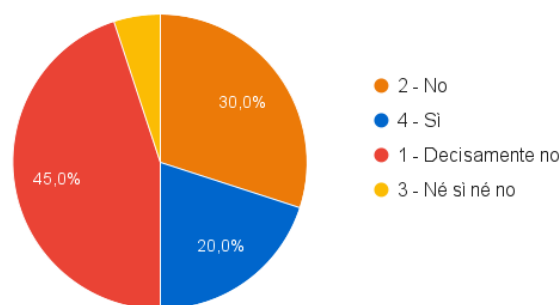


Grafico 5.11: Quesito di valutazione sulla possibilità di utilizzo in una produzione televisiva.

In definitiva, le catture volumetriche, realizzate con l'applicazione V3LCamera, hanno ottenuto risposte molto variegata ma, in quasi tutte le domande, la media si è attenuta ai livelli bassi.

Confronto e risultati finali

I test per le tre applicazioni considerate si sono svolti ponendo ai tester le stesse domande. In tal modo, è stato possibile confrontare successivamente le risposte tra loro per comprendere quale applicazione fosse migliore rispetto alle altre e in quali aspetti.

Una visione generale dell'istogramma in figura (grafico 5.12) permette di comprendere la notevole prevalenza dell'applicazione Volu sulle altre due applicazioni analizzate sotto ogni punto di vista. Infatti, la media delle risposte si aggirano, per ogni domanda posta ai tester, ad un valore di 3 e 4. Depthkit e V3LCamera, invece, mostrano risultati molto simili tra di loro anche se è possibile riscontrare una leggera prevalenza di Depthkit. L'unica domanda in cui V3LCamera è riuscita ad ottenere un risultato migliore si individua nel quesito relativo al realismo del soggetto ricostruito. Le ricostruzioni di Depthkit, in questo caso, sfruttano i dati di profondità restituiti dalla Kinect v2, uscita nel 2009 e ormai fuori produzione. Nonostante la tecnologia datata, Depthkit permette di generare delle ricostruzioni volumetriche di qualità migliore rispetto a quelle ottenute con la tecnologia LiDAR dell'applicazione V3LCamera. Entrambe le applicazioni, però, hanno ottenuto valori mediamente bassi e ciò porta ad affermare che un loro utilizzo sia notevolmente complicato. Nonostante ciò, si potrebbe valutare un loro impiego differente che permetta di nascondere i difetti. Infatti, si potrebbe decidere di non puntare al realismo e inserirli in una pipeline diversa. Le catture volumetriche, infatti, offrono numerosi vantaggi che possono essere sfruttati anche in contesti non realistici.

Volu, invece, presenta dei risultati molto buoni e ciò permette di affermare che tale applicazione sia in grado di fornire un risultato utilizzabile in un contesto televisivo. Il realismo, in questo caso, ottiene una media di valori sopra il livello 4 della scala ACR. Ciò permette di concludere che tale applicazione è in grado di generare una cattura volumetrica realistica ed utilizzabile come tale, senza ausilio di effetti visivi.

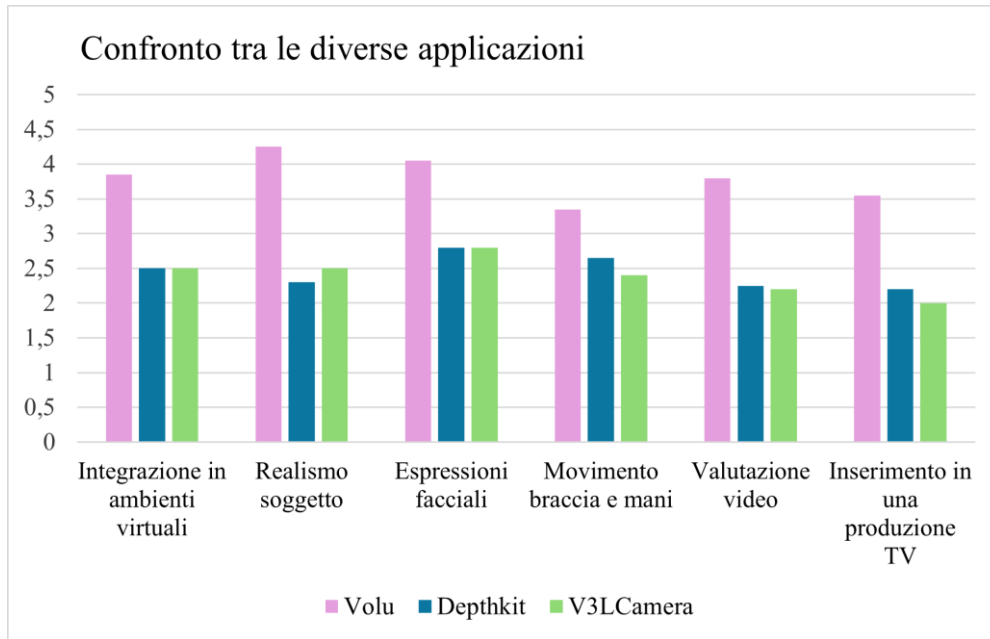


Grafico 5.12: Istogramma per il confronto delle applicazioni Volu, Depthkit e V3LCamera

5.1.3.2 Gaussian Splatting 4D

Una volta terminata la valutazione delle tre tecnologie precedenti, le modalità di test cambiano e gli utenti devono confrontare un video originale con quello ottenuto tramite la ricostruzione con il 4D-GS. I questionari presentano alcune domande uguali, in cui viene richiesto di valutare quanto la rappresentazione del soggetto sia realistica, fornire una valutazione globale della ricostruzione, considerare la somiglianza al video originale e analizzare le espressioni facciali. Gli altri quesiti, invece, variano da video a video; per ciascuno, infatti, si è ritenuto necessario sottolineare alcuni aspetti specifici, che hanno permesso di individuare punti di forza e di debolezza delle ricostruzioni tramite il Gaussian Splatting 4D.

Sono stati realizzati quattro video, due in interno e due all'esterno.

Video green screen

Il primo video è stato girato in interni, con un green screen sullo sfondo, e presenta lo stesso soggetto, che compie gli stessi movimenti dei video volumetrici realizzati con Volu, Depthkit e V3LCamera (figura 5.10). In questo caso, l'obiettivo era valutare come il 4D-GS ricostruisse un video equivalente a quelli visti in precedenza. I movimenti sono

abbastanza lenti e controllati e anche le espressioni del volto non cambiano repentinamente.



Figura 5.10: A sinistra video green screen originale. A destra video green screen ricostruito con il 4D-GS

I risultati ottenuti nel questionario mostrano che il soggetto ricostruito è stato considerato realistico da tutti i tester: il 65% ha votato il punteggio più alto della scala e il 35% il secondo più alto (grafico 5.13).

Quanto risulta realistico il soggetto?

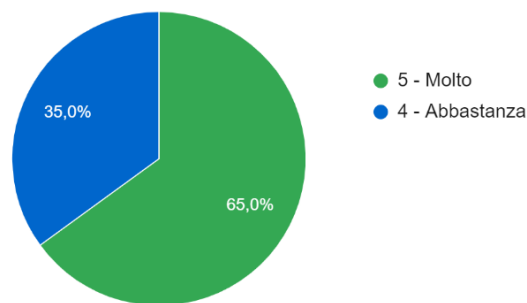


Grafico 5.13: Quesito di valutazione sul realismo del soggetto ricostruito nel video su green screen

Anche la valutazione generale del video si aggiudica punteggi molto positivi. Nessun utente ha votato “scarso” o “pessimo” e il 70% considera “buono” il risultato ottenuto (grafico 5.14). Inoltre, il 65% ha votato il punteggio più alto della scala nella domanda in cui viene chiesto di valutare la somiglianza al video originale (grafico 5.15).

Come valuti il video in generale?

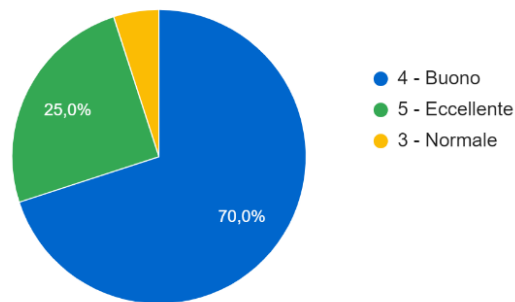


Grafico 5.14: Quesito di valutazione in generale del video su green screen

Quanto è simile al video originale?

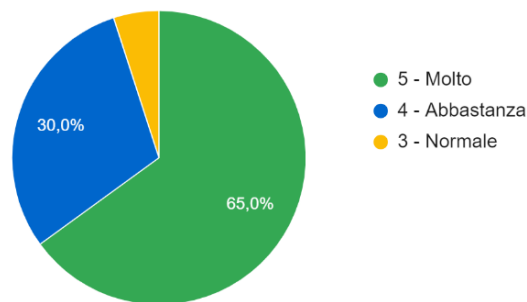


Grafico 5.15: Quesito di valutazione sulla somiglianza tra video originale e ricostruito del video su green screen

In particolare, si evidenzia che le espressioni facciali sono valutate in maniera molto positiva dagli utenti (grafico 5.16). Le espressioni facciali, in questo caso, erano semplici e il sistema è riuscito a ricostruirle bene. Anche il movimento generale del soggetto viene valutato in maniera positiva (grafico 5.17). Si riscontrano, invece, esiti leggermente peggiori per il movimento delle braccia e delle mani, in cui il 30% dei voti è “scarso” o “normale” (grafico 5.18).

Come valuti le espressioni facciali?

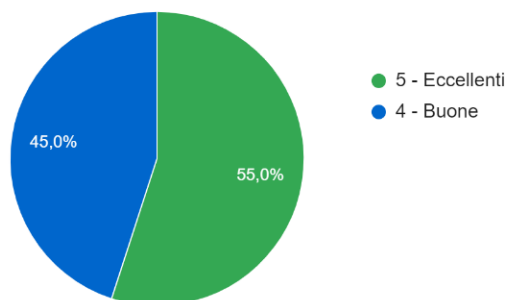


Grafico 5.16: Quesito di valutazione sulle espressioni facciali del video su green screen

Come valuti il movimento del soggetto in generale?

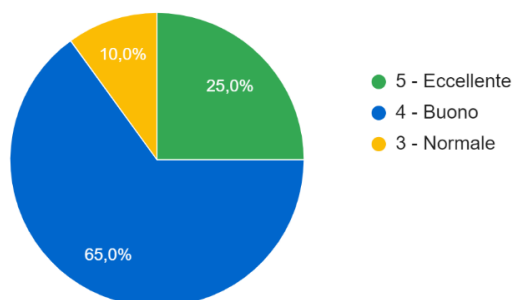


Grafico 5.17: Quesito di valutazione sui movimenti del soggetto del video su green screen

Come valuti il movimento delle mani e delle braccia?

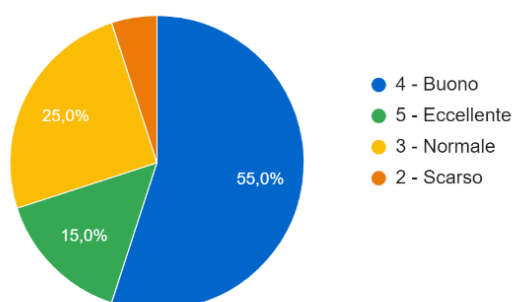


Grafico 5.18: Quesito di valutazione sul movimento di braccia e gambe del video su green screen

Video viso

Il secondo video proposto (figura 5.11) mira a valutare come il 4D-GS sia in grado di rispondere a veloci cambiamenti nelle espressioni facciali e la bontà di rappresentazioni delle riflessioni, visibili negli occhiali indossati. Il video è stato girato all'esterno, in una giornata soleggiata. Si vuole infatti anche considerare se le valutazioni migliorano o peggiorano a seconda che il video sia girato in ambienti al chiuso o in spazi all'aperto.

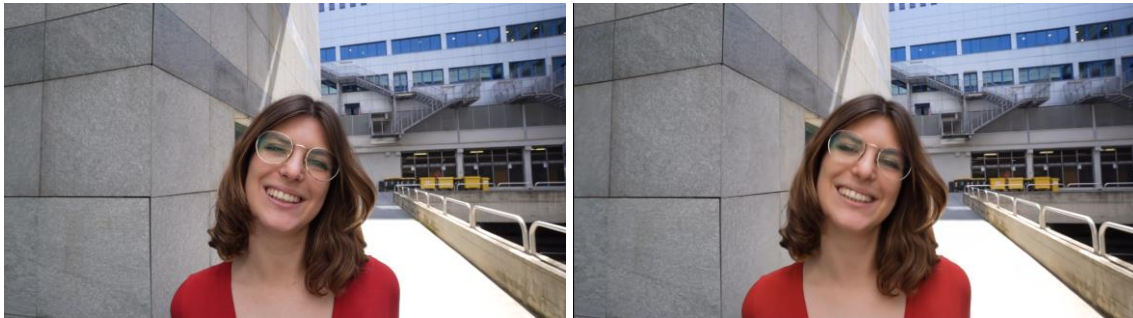


Figura 5.11: A sinistra video viso originale. A destra video viso ricostruito con il 4D-GS

Dalle valutazioni ottenute emerge che la ricostruzione del soggetto è stata considerata realistica da quasi tutti i tester, di cui il 35 % ha votato il livello massimo della scala ACR (grafico 5.19). Anche la valutazione globale del video è positiva: l'90% degli utenti ritiene il video ottenuto "normale" o "buono" (livello 3 e 4 della scala). Si nota che nessun utente non considera né eccellente, né pessimo il risultato ottenuto (grafico 5.20).

Quanto risulta realistico il soggetto?

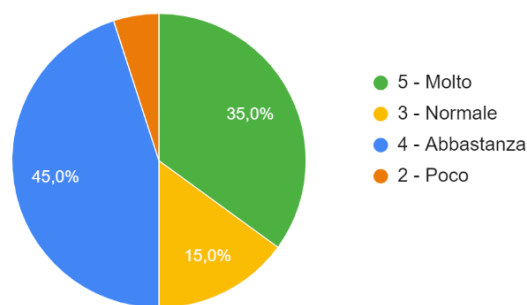


Grafico 5.19: Quesito di valutazione sul realismo del soggetto ricostruito nel video sulle espressioni facciali

Come valuti il video in generale?

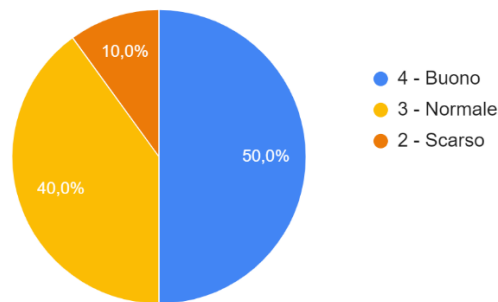


Grafico 5.20: Quesito di valutazione in generale del video sulle espressioni facciali

Il 70% dei tester ritiene che il video sia “molto” o “abbastanza” simile all’originale. Solo il 5% considera il video poco simile a quello di partenza e nessuno ha votato il livello più basso della scala. La valutazione globale risulta, quindi, positiva (grafico 5.21).

Quanto è simile al video originale?

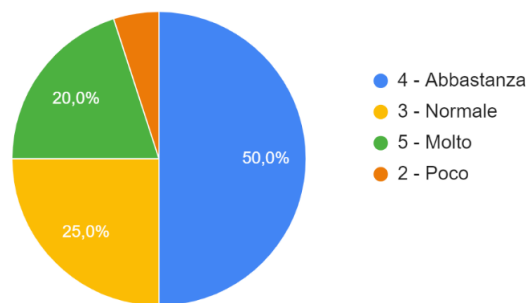


Grafico 5.21: Quesito di valutazione sulla somiglianza tra video originale e ricostruito del video sulle espressioni facciali

Il movimento del soggetto, in questo caso, si riferisce in particolare alle movenze della testa. I risultati ottenuti sono abbastanza positivi: nessuno ha votato pessimo e il 45% ritiene che i movimenti ricostruiti siano buoni e il 10% eccellenti (grafico 5.22).

Come valuti il movimento del soggetto in generale?

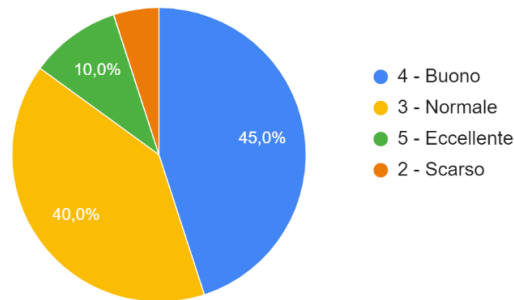


Grafico 5.22: Quesito di valutazione sul movimento del soggetto in generale nel video sulle espressioni facciali

In aggiunta, la resa delle espressioni facciali è stata considerata “buona” dal 70% dei partecipanti al test ed “eccellente” dal 10%. Anche in questo caso nessuno ha votato il livello più basso della scala (grafico 5.23).

Come valuti le espressioni facciali?

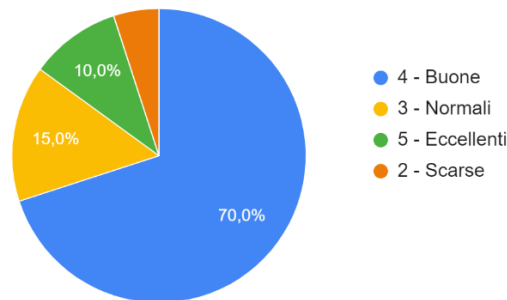


Grafico 5.23: Quesito di valutazione sulle espressioni facciali nel video del viso

La riproduzione degli occhiali ottiene una valutazione più bassa rispetto alle precedenti (grafico 5.24). L’obiettivo è valutare se l’algoritmo del 4D-GS è in grado di interpretare e ricostruire in modo accurato le riflessioni. Il 20% dei partecipanti ritiene il risultato non soddisfacente. Al contrario, il 10% ritiene che sia eccellente e la maggior parte (70%) ha votato i livelli 3 e 4 della scala. Si deduce che il Gaussian Splatting 4D riesce a interpretare e riprodurre, sebbene non perfettamente, le superfici riflettenti.

Come valuti la riproduzione degli occhiali?

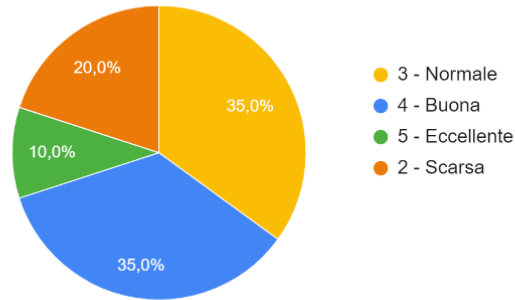


Grafico 5.24: Quesito di valutazione sulla riproduzione degli occhiali del video sulle espressioni facciali

Video camminata

Il terzo video vuole valutare se e in che modo il 4D-GS riesca a ricostruire movimenti complessi, come quelli di una camminata, in cui il soggetto cambia notevolmente la sua posizione nello spazio. Una difficoltà aggiunta è l'ambientazione all'esterno, in cui è stato girato il video. Si desidera, inoltre, evidenziare la capacità di ricostruzione dei vestiti, soprattutto quando in movimento. Infatti, la persona nel video indossa una gonna che si muove molto durante la camminata (figura 5.12).

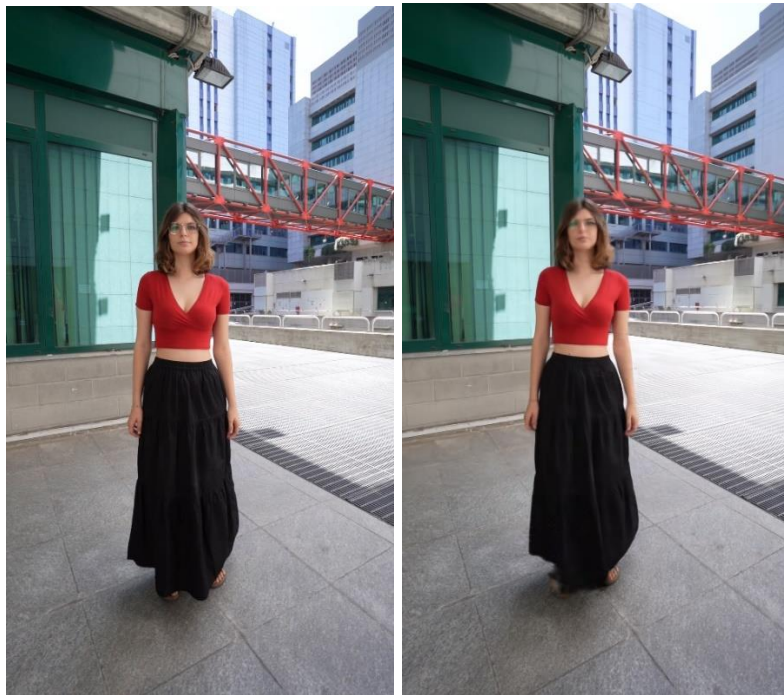


Figura 5.12: A sinistra video camminata originale. A destra video camminata ricostruito con il 4D-GS

Il 65% delle persone ritiene che il soggetto riprodotto sia abbastanza o molto realistico: ha votato, cioè, i livelli 4 e 5 della scala ACR (grafico 5.25). Anche la valutazione complessiva (grafico 5.26) e la somiglianza al video originale (grafico 5.27) ottengono buoni punteggi, seppur si noti una percentuale abbastanza consistente (nel primo caso il 20%, nel secondo 15%) che considera il risultato “scarso”.

Quanto risulta realistico il soggetto?

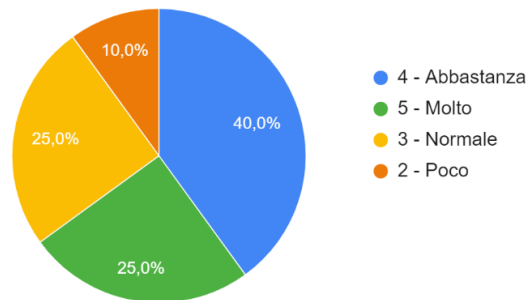


Grafico 5.25: Quesito di valutazione sul realismo del soggetto ricostruito nel video della camminata

Come valuti il video in generale?

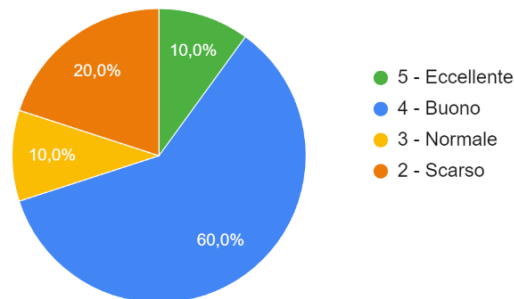


Grafico 5.26: Quesito di valutazione in generale del video della camminata

Quanto è simile al video originale?

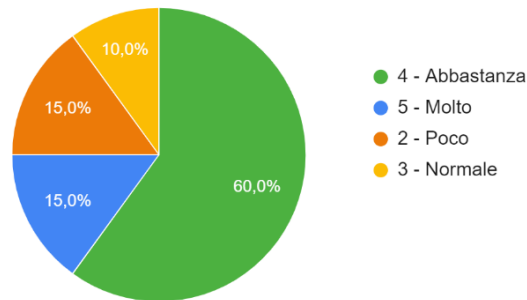


Grafico 5.27: Quesito di valutazione sulla somiglianza tra video originale e ricostruito del video della camminata

Le espressioni facciali sono considerate dal 50% “buone”, ma è bene notare che una percentuale consistente (il 20%) le considera scarse (grafico 5.28). Il movimento ampio del soggetto potrebbe aver introdotto alcuni difetti anche nel volto.

Come valuti le espressioni facciali?

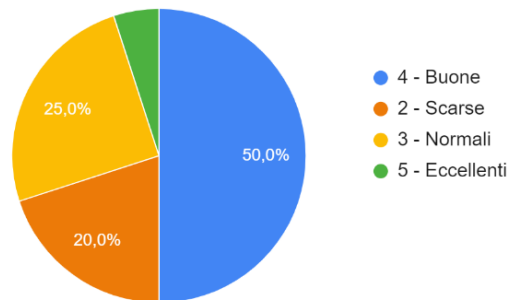


Grafico 5.28: Quesito di valutazione sulle espressioni facciali del video della camminata

La camminata viene considerata fluida e realistica. Il 45% ha votato il livello 4 della scala e il 30% il massimo livello, il 5 (grafico 5.29). Anche il movimento più generico del soggetto, in cui vengono inclusi il movimento della testa, dei capelli e dei vestiti, ha soddisfatto i tester; solo il 10% lo ritiene “scarso” e nessuno “pessimo”, ultimo livello della scala (grafico 5.30).

Quanto è fluido il movimento della camminata?

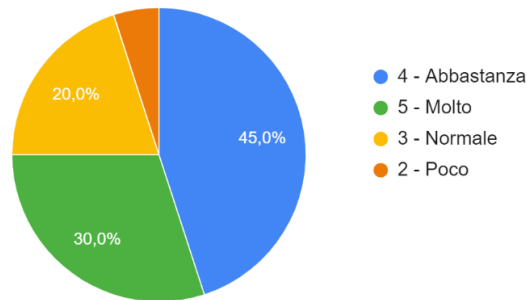


Grafico 5.29: Quesito di valutazione sulla fluidità del movimento nel video della camminata

Come valuti il movimento del soggetto in generale?

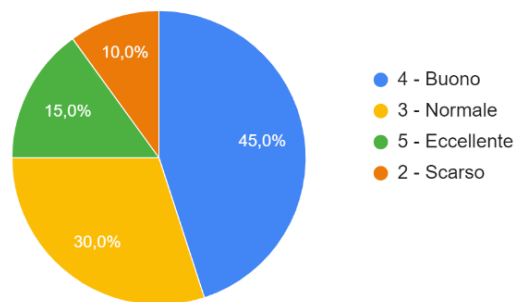


Grafico 5.30: Quesito di valutazione sul movimento in generale nel video della camminata

Infine, la ricostruzione dei vestiti e del loro movimento è stata considerata “eccellente” dal 35% di coloro che hanno partecipato al test e “buona” dal 55%. Il risultato ottenuto è notevole, considerando anche che nessuno ha votato gli ultimi due livelli della scala: nessuno considera la riproduzione degli abiti non soddisfacente (grafico 5.31).

Come valuti la ricostruzione dei vestiti?

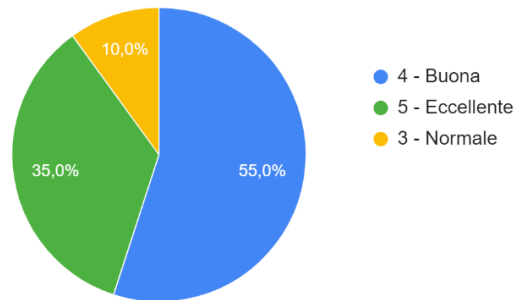


Grafico 5.31: Quesito di valutazione sulla ricostruzione dei vestiti nel video della camminata

Video ginnastica

L'ultimo video, girato in spazi chiusi, presenta un soggetto che compie un ampio movimento sul posto (figura 5.13). La persona salta ripetutamente, muovendo braccia e gambe. In questo caso, si vuole valutare se il 4D-GS sia in grado di ricostruire movimenti veloci senza introdurre eccessivi artefatti visivi.



Figura 5.13: A sinistra video ginnastica originale. A destra video ginnastica ricostruito con il 4D-GS

La ricostruzione del soggetto ha ottenuto punteggi molto alti. Nessuno ha votato i due livelli più bassi della scala e ben il 50% dei partecipanti ha votato il livello massimo (grafico 5.32). Allo stesso modo, anche il video nella sua totalità viene valutato in maniera positiva: il 55% ritiene il risultato ottenuto “buono” e il 15% “eccellente” (grafico 5.33). A conferma di un risultato soddisfacente, anche la somiglianza al video originale si aggiudica un’ottima valutazione, con alte percentuali nei livelli più alti della scala (grafico 5.34).

Quanto risulta realistico il soggetto?

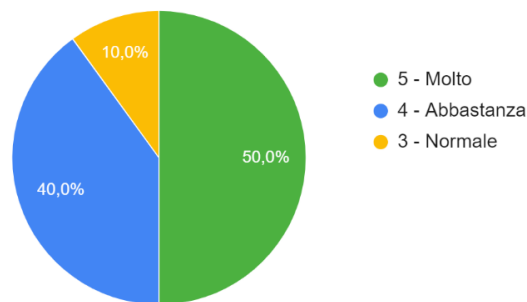


Grafico 5.32: Quesito di valutazione sul realismo del soggetto ricostruito nel video della ginnastica

Come valuti il video in generale?

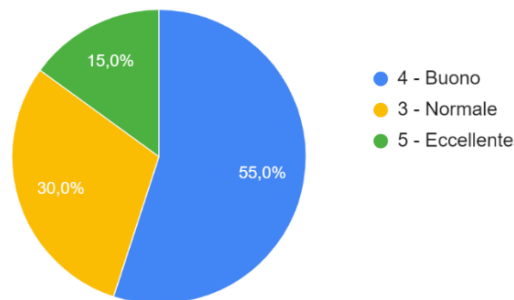


Grafico 5.33: Quesito di valutazione in generale del video della ginnastica

Quanto è simile al video originale?

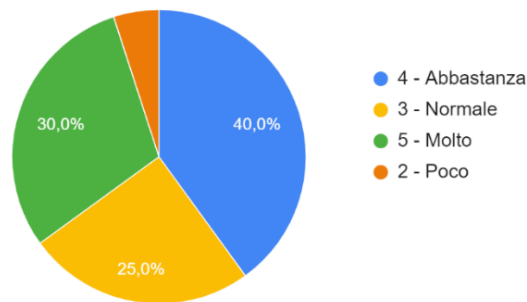


Grafico 5.34: Quesito di valutazione sulla somiglianza tra video originale e ricostruito del video della ginnastica

Il movimento del soggetto, nella sua totalità, ha ottenuto valutazioni soddisfacenti; il 50% lo ritiene “buono”, il 25% “eccellente” e il 25% “normale” (grafico 5.35). Anche il movimento delle mani e delle braccia (grafico 5.36) e quello delle gambe (grafico 5.37), presi singolarmente, sono stati considerati realistici da chi ha risposto al questionario. Il primo, però, ha ottenuto una valutazione leggermente più bassa, in cui il 5% ha ritenuto “scarso” il risultato ottenuto.

Come valuti il movimento del soggetto in generale?

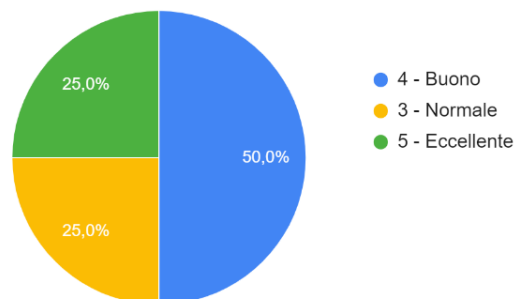


Grafico 5.35: Quesito di valutazione sul movimento generale del soggetto nel video della ginnastica

Come valuti il movimento delle mani e delle braccia?

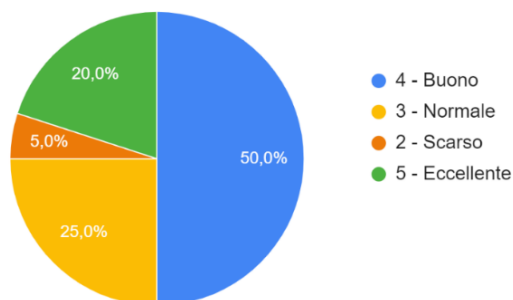


Grafico 5.36: Quesito di valutazione sul movimento di mani e braccia nel video della ginnastica

Come valuti il movimento delle gambe?

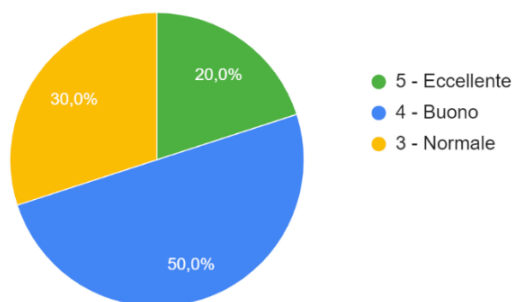


Grafico 5.37: Quesito di valutazione sul movimento delle gambe nel video della ginnastica

Infine, non sembra che la velocità del movimento peggiori il risultato delle espressioni facciali; infatti, il 65% dei tester sostiene che siano “buone” (grafico 5.38).

Come valuti le espressioni facciali?

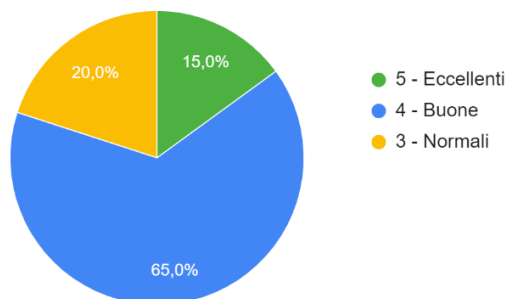


Grafico 5.38: Quesito di valutazione sulle espressioni facciali nel video della ginnastica

Confronto e risultati finali

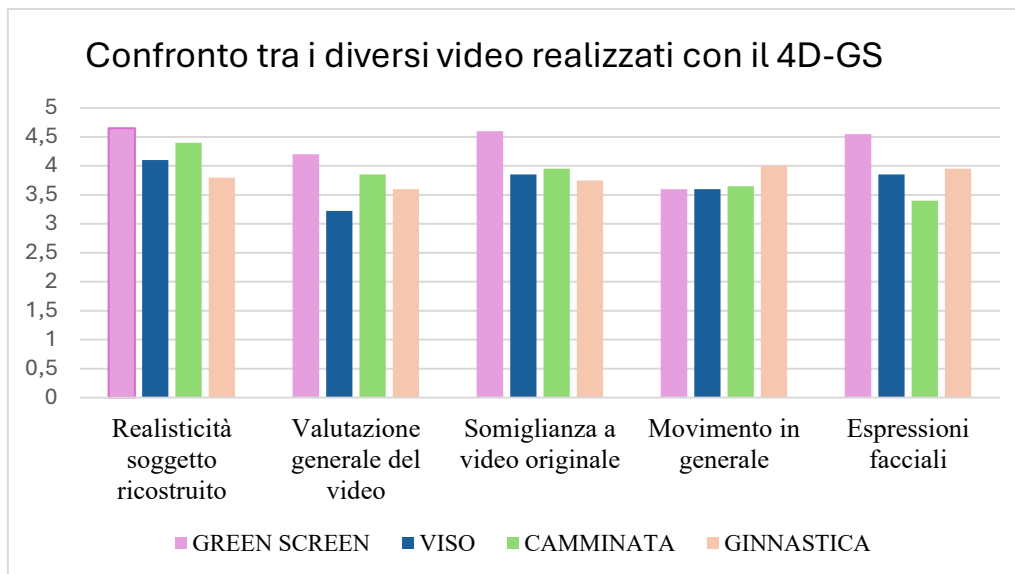


Grafico 5.39: Istogramma per il confronto tra i diversi video ottenuti con il gaussian splatting 4D

Dall'istogramma (grafico 5.39) emerge che il video che ha ottenuto valutazioni più alte è quello girato su green screen. Il risultato non è sorprendente in quanto le riprese sono avvenute in ambiente al chiuso, con un'opportuna illuminazione, e i movimenti svolti non sono eccessivamente complessi e veloci, così come le espressioni facciali. Il punteggio di valutazione del movimento in generale non è altissimo: questo accade perché, in questo video, la ricostruzione con il 4D-GS non riesce a interpretare perfettamente la mano e il braccio quando in moto.

Si nota un punteggio più basso rispetto agli altri nella valutazione complessiva del video con la ricostruzione del viso. Il 4D-GS mostra alcune difficoltà nella ricostruzione di volti in primo piano, soprattutto quando le espressioni facciali sono molto marcate e cambiano rapidamente.

Le espressioni facciali del video della camminata sono valutate come le peggiori. Il 4D-GS non riesce a ricostruire in maniera estremamente efficace il volto, quando il corpo si muove molto nello spazio.

Il video della ginnastica ha ottenuto il punteggio più alto nella valutazione del movimento in generale. Questo risultato non è scontato, in quanto il sistema potrebbe avere difficoltà nella ricostruzione di salti e ampie mosse. In questo specifico caso, invece, è stato giudicato in maniera positiva.

Non si nota una discrepanza notevole nei punteggi tra i video girati all'interno e all'esterno, anche se quelli girati al chiuso presentano votazioni leggermente più alte.

In conclusione, si può affermare che i risultati ottenuti siano buoni e hanno reso possibile evidenziare potenzialità e possibili problematiche nell'utilizzo del Gaussian Splatting 4D per la ricostruzione di figure umane in movimento.

5.1.3.3 Real time

L'ultimo test, con relativo questionario, riguarda la cattura volumetrica in tempo reale, ottenuta sfruttando il plugin Velox Neuro con l'uso della webcam. La riproduzione in tempo reale dei partecipanti è stata inserita nell'ambiente virtuale usato anche per quelle non in real time e i tester hanno potuto muoversi e testare il sistema liberamente. Sono poi state poste alcune domande per valutare la qualità complessiva della ricostruzione; in aggiunta, è stata inserita una domanda aperta in cui coloro che hanno preso parte al test hanno annotato aspetti positivi e negativi riscontrati.

Il movimento riprodotto non ha ottenuto valutazioni altissime: il 35% ha votato il livello 2 della scala e il 5% il più basso. Il 30% ritiene, però, "buono" il risultato ottenuto. Nessuno ha votato il livello più alto della scala (grafico 5.40).



Grafico 5.40: Quesito di valutazione sul movimento riprodotto in real time

La latenza del sistema non è stata considerata eccessiva, solo il 15% dei partecipanti al test sostiene che il sistema sia "poco" reattivo. Il 50% ha invece votato il secondo livello più alto della scala (grafico 5.41). Si conclude che questa cattura volumetrica in tempo reale riesce a non introdurre un eccessivo ritardo e preservare il real time.

Quanto reputi reattivo il sistema?

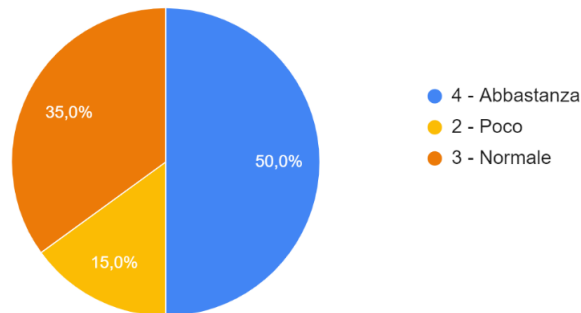


Grafico 5.41: Quesito di valutazione sulla reattività del sistema in real time

Per il 35% dei tester il realismo della riproduzione digitale non è sufficiente e hanno votato infatti gli ultimi due livelli della scala di valutazione. Il 40% mantiene una valutazione media, il 20% ritiene la ricostruzione “buona” e solo il 5% “eccellente”. I punteggi non sono quindi altissimi (grafico 5.42). Il sistema di Velox in tempo reale introduce infatti alcune problematiche, non riscontrate in quello non in real time, che causano una diminuzione del realismo.

Quanto reputi realistica la tua riproduzione digitale?

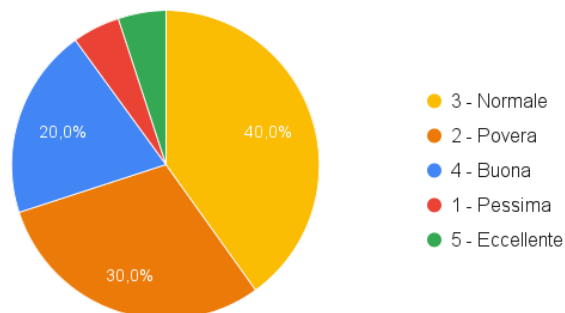


Grafico 5.42: Quesito di valutazione sul realismo della riproduzione digitale in real time

È stato suggerito ai partecipanti al test di avvicinarsi e allontanarsi dalla webcam per valutare se il sistema migliorasse o peggiorasse al variare della distanza (grafico 5.43). Il 30% ritiene che avvicinarsi alla camera non introduca un perfezionamento degno di nota. Un altro 30%, invece, sostiene che la riproduzione digitale migliori “abbastanza” quando si è più vicini.

Quanto migliora la tua riproduzione digitale man mano che ti avvicini alla webcam?

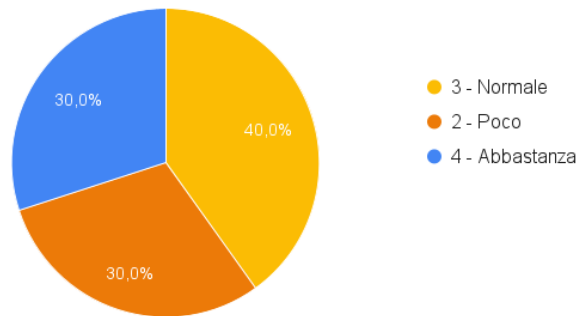


Grafico 5.43: Quesito di valutazione sul miglioramento della riproduzione digitale in real time se ci si avvicina alla webcam

Infine, è stato chiesto ai partecipanti se il risultato ottenuto potesse funzionare per una produzione televisiva (grafico 5.44). Il 45% si esprime in maniera negativa e solo il 20% risponde di sì. Nessuno ha votato il livello più alto della scala. Si deduce che la qualità della ricostruzione digitale non venga ritenuta sufficiente dalla maggioranza per un utilizzo televisivo.

Potrebbe funzionare per una produzione televisiva?

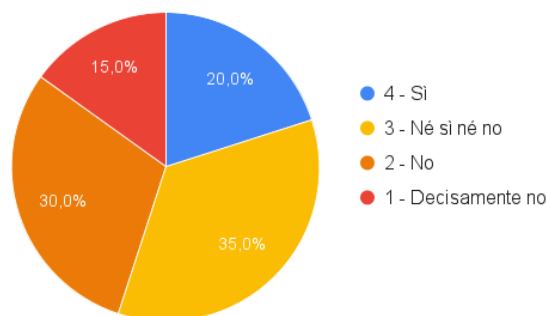


Grafico 5.44: Quesito di valutazione sulla possibilità di utilizzo in una produzione televisiva

Dalle domande aperte emerge che la riproduzione volumetrica è considerata generalmente soddisfacente, nonostante siano emerse alcune aree potenzialmente da migliorare. Tra gli aspetti positivi, si evidenzia la reattività del sistema, le ottime riflessioni dinamiche e le ombre e, in generale, una buona sensazione di realismo. Tuttavia, è opportuno riportare anche i principali difetti riscontrati dai partecipanti al test: la figura viene descritta come “schiacciata”, specialmente nella zona della testa, e i tester ritengono che non tutti i movimenti delle braccia e delle gambe vengano catturati

correttamente, specialmente se molto rapidi. La ricostruzione viene definita “abbastanza piatta”, priva di volume, senza contorni netti e definiti. Inoltre, viene sottolineato come il sistema non riesca sempre a riprodurre opportunamente le mani, ad esempio, quando le dita non sono molto distanti, non riesce a ritagliarle bene e fornire una rappresentazione convincente. Emerge infine una scarsa qualità nei dettagli del viso e dei capelli.

In conclusione, per aumentare il realismo della rappresentazione, il sistema dovrebbe fornire una ricostruzione più dettagliata, meglio delineata, che restituisca un maggiore senso di tridimensionalità e in grado di catturare meglio i movimenti veloci.

5.2 Analisi oggettive

Le tre applicazioni utilizzate (Volu, Depthkit e V3LCamera) hanno permesso di generare delle catture volumetriche nei vari formati proprietari, precedentemente analizzati. Tale caratteristica è limitante per una valutazione tramite metriche oggettive. Infatti, i formati delle tre applicazioni sono diversi tra loro e, dunque, non confrontabili tramite delle metriche consolidate e uniche. Inoltre, le metriche oggettive sono facilmente utilizzabili su output in formato non proprietario. Il Gaussian Splatting 4D, invece, mette a disposizione la possibilità di valutare oggettivamente il risultato ottenuto.

5.2.1 Parametri utilizzati

Il github, relativo al codice del Gaussian Splatting 4D, mette a disposizione un file python che permette di ottenere i valori di differenti metriche. Queste ultime sono in grado di dare una valutazione oggettiva sulla bontà del video ricostruito in questione. Di seguito verranno analizzate nel dettaglio le metriche sfruttate a tale scopo.

SSIM

La *Structural Similarity Index Measure (SSIM)* è una metrica percettiva che permette di calcolare la somiglianza tra due immagini e di quantificare il degrado dell'immagine. Si tratta di una metrica di riferimento completa che richiede due immagini dalla stessa acquisizione: un'immagine di riferimento e un'immagine elaborata [63]. SSIM è un modello basato sulla percezione che considera il degrado dell'immagine come un cambiamento percepito nelle informazioni strutturali, incorporando anche importanti fenomeni percettivi [64].

Per valutare la qualità dell'immagine, l'SSIM viene solitamente applicata solo alla luminanza, sebbene possa essere applicata anche a valori di colore (ad esempio RGB) o cromatici (ad esempio YCbCr). La metrica SSIM fornisce come risultato un valore decimale compreso tra -1 e 1, dove 1 indica perfetta somiglianza, 0 indica nessuna somiglianza e -1 indica anticorrelazione.

MS-SSIM

Una forma più avanzata di SSIM, chiamata *Multiscale SSIM (MS-SSIM)*, viene eseguita su più scale attraverso un processo di downsampling multi-step. La metrica MS-SSIM combina le informazioni sulla luminanza, al livello di risoluzione più elevato, con le informazioni sulla texture e sul contrasto, a più risoluzioni o scale ridotte. Le diverse scale tengono conto della variabilità nella percezione dei dettagli dell'immagine causata da diversi fattori. Tra questi ultimi si possono individuare la distanza di visione dall'immagine, la distanza tra la scena e il sensore e la risoluzione del sensore di acquisizione dell'immagine. In questo modo, MS-SSIM è in grado di catturare la complessità della percezione visiva umana. Quando un'immagine presenta differenze di pixel massime, il valore di MS-SSIM è uguale a zero. Se le differenze dell'immagine diminuiscono, il valore della metrica aumenta [65]. È stato dimostrato che funziona altrettanto bene o, meglio, di SSIM con vari database di immagini e video.

D-SSIM

La *Structural Dissimilarity Index Measure (DSSIM)* può essere derivata da SSIM [64] e costituisce una metrica utilizzata per calcolare la differenza tra due immagini. La SSIM, invece, viene utilizzata per il calcolo della somiglianza tra l'immagine originale e quella elaborata. La formula, utilizzata per il calcolo del DSSIM, si definisce come mostrato in figura (figura 5.14).

$$DSSIM(x, y) = \frac{1 - SSIM(x, y)}{2}$$

Figura 5.14: Formula per il calcolo del DSSIM

Fonte: <https://is.gd/FHCcGa>

Basandosi su tale formula, più il valore risultante è vicino a 0 più le immagini sono simili, mentre se il valore è vicino ad 1 allora le immagini considerate saranno molto diverse.

PSNR

Il *Peak Signal-to-noise Ratio (PSNR)* è una misura adottata per valutare la qualità di una immagine compressa o elaborata rispetto all'originale. Tale indice è definito come il

rapporto tra la massima potenza di un segnale e la potenza di rumore che può compromettere la qualità dell'immagine elaborata [66]. Il PSNR è solitamente espresso in termini di scala logaritmica di decibel. Un valore più alto di PSNR indica che l'immagine elaborata sarà stata ricostruita meglio per corrispondere all'immagine originale e, inoltre, permette di affermare che l'algoritmo ricostruttivo ha svolto un lavoro migliore. Tipici valori di PSNR variano da 20 a 40. Un incremento di 0,25 dB viene in genere considerato un'ottimizzazione significativa, apprezzabile dal punto di vista percettivo umano [67].

Il limite principale di questa metrica è che si basa strettamente su un confronto numerico non tenendo in conto nessun fattore biologico del sistema visivo umano, a differenza di ciò che avviene utilizzando l'indice di somiglianza strutturale (SSIM).

LPIPS

La metrica *Learned Perceptual Image Patch Similarity (LPIPS)* calcola la somiglianza percettiva tra due immagini. In particolare, LPIPS sfrutta la suddivisione in patch dell'immagine, ovvero in sotto-regioni, ed è basata su dei modelli di deep-learning pre-addestrati. È stato dimostrato che questa misura corrisponde bene alla percezione umana. Un punteggio LPIPS basso significa che le patch delle immagini sono percettivamente simili [68].

Questa metrica può utilizzare differenti reti neurali, tra cui vgg e alex. La rete **vgg** (Visual Geometry Group) è una rete neurale convoluzionale profonda (CNN), utilizzata nei modelli innovativi di identificazione degli oggetti [69]. La rete **alex** è sempre un modello di rete neurale convoluzionale profonda (CNN) che ha rappresentato un significativo avanzamento nel riconoscimento automatico delle immagini[70].

5.2.2 Risultati

Di seguito, vengono riportati i risultati dei quattro video ricostruiti con il 4D-GS, analizzati attraverso le metriche di qualità dell'immagine precedentemente discusse: SSIM, MS-SSIM, D-SSIM, PSNR e LPIPS (versione vvg e alex). I valori sono riportati fino alla terza cifra decimale (tabella 5.1).

VIDEO	↑ SSIM	↑ MS-SSIM	↓ D-SSIM	↑ PSNR	↓ LPIPS-VGG	↓ LPIPS-ALEX
GREEN SCREEN	0,941	0,971	0,014	37,488	0,237	0,075
VISO	0,923	0,965	0,018	34,677	0,195	0,195
CAMMINATA	0,905	0,963	0,019	31,234	0,201	0,096
GINNASTICA	0,905	0,955	0,023	31,243	0,219	0,108

Tabella 5.1: Confronto tra i valori delle metriche oggettive per i vari video ricostruiti con il Gaussian Splatting 4D

Tutti i video analizzati presentano valori SSIM superiori a 0,9, evidenziando una notevole somiglianza visiva. Questa metrica ha come massimo 1, che indica perfetta somiglianza. I risultati ottenuti indicano quindi una buona affinità strutturale tra i video ricostruiti e quelli di riferimento. Si nota, inoltre, che il video del “green screen” presenta un valore più alto (0,941) rispetto agli altri (“viso” a 0,923, “camminata” e “ginnastica” entrambi a 0,905).

Tutti i video hanno valore MS-SSIM molto alti, con un intervallo che va da 0,955 a 0,971. Questo conferma che la qualità percettiva è alta. Anche in questo caso, il video “green screen” ottiene il punteggio migliore.

Anche i valori del D-SSIM sono buoni; in questo caso, più ci si avvicina a 0 e più il video ricostruito è simile a quello originale. Si nota una leggera variazione, da 0,014 (“green screen”) a 0,023 (“ginnastica”). Il video “green screen” ha di nuovo ottenuto il punteggio migliore, mentre “ginnastica” il peggiore.

I valori di PSNR mostrano una maggiore discrepanza tra i vari video ricostruiti rispetto alle metriche precedenti. Il video “green screen” ha un PSNR di 37,488, seguito da “viso” con 34,677, “ginnastica” con 31,243 e “camminata” con 31,234. Si ricorda che un PSNR più alto indica una ricostruzione meno rumorosa e una maggiore qualità dell'immagine. Anche secondo la metrica LPIPS-alex il video “green screen” è quello che presenta la ricostruzione più simile all'originale. Al contrario, i valori di LPIPS-vgg indicano il video

“green screen” (0,237) come il peggiore e “viso” (0,195) come il migliore. Tutti i risultati, comunque, sono relativamente bassi. Questo suggerisce che la somiglianza percettiva globale rimane abbastanza buona.

In conclusione, il video “green screen” ha ottenuto i punteggi migliori in quasi tutte le metriche. Si può ipotizzare che uno sfondo semplice e movimenti controllati permettano di ottenere ricostruzioni di qualità superiore. Anche il video “viso” ottiene buoni risultati, con un PSNR e SSIM leggermente inferiori a “green screen”, ma comunque indicativi di una buona ricostruzione. Infine, i video “camminata” e “ginnastica” hanno i punteggi più bassi in diverse metriche. Questo risultato si può attribuire alla maggiore complessità di movimento, che introduce motion blur e rende più difficile la gestione dei dettagli fini.

5.3 Analisi soggettive e oggettive 4D-GS a confronto

Per fornire una valutazione più completa del 4D-GS, è necessario confrontare i risultati emersi dai test soggettivi (vedi grafico 5.39) con quelli oggettivi (vedi tabella 5.1).

Il video “green screen” ha ottenuto ottime valutazioni in entrambi i casi. L’unica anomalia riguarda il valore di LPIPS-vgg, che è inferiore rispetto agli altri video, nonostante le altre metriche suggeriscano il contrario. Tuttavia, la buona qualità emersa dai test è confermata da tutte le altre metriche dell’analisi oggettiva.

Nel video “viso”, nonostante le valutazioni soggettive siano meno positive a causa della difficoltà del sistema nel ricostruire espressioni facciali complesse, le metriche oggettive mostrano comunque buoni risultati. Il miglior valore di LPIPS-vgg suggerisce che la riproduzione digitale potrebbe essere percepita come più realistica di quanto in realtà emerso dalle valutazioni soggettive. In generale, però, le metriche oggettive sono abbastanza in linea con quelle soggettive.

Le valutazioni soggettive nel video “camminata” sono in parte confermate dalle metriche oggettive. L’SSIM e il PSNR sono i più bassi, a conferma del fatto che un ampio movimento del corpo rende difficile una ricostruzione precisa. Al contrario, i valori di LPIPS-vgg e LPIPS-alex sono i secondi migliori, suggerendo un maggior realismo percepito. Anche la valutazione sulla realistica del soggetto ricostruito, ottenuta dai test soggettivi, ha ottenuto il secondo punteggio più alto dopo “green screen”, confermando un forte plausibilità della ricostruzione. Si conclude che, nonostante il sistema non riesca a riprodurre in maniera perfetta il movimento e le espressioni facciali, il risultato, tutto sommato, è convincente.

Le metriche oggettive mostrano che il video “ginnastica” è tra i meno performanti. L’SSIM e il PSNR sono relativamente bassi. Questo conferma ulteriormente una difficoltà del sistema a gestire la ricostruzione del soggetto, soprattutto dei dettagli, durante movimenti complessi. I punteggi raggiunti da questo video nelle valutazioni soggettive non sono comunque altissimi, a conferma di un risultato non ottimale. Ciò nonostante, sembra che la qualità percepita della ricostruzione sia comunque superiore rispetto a quella misurata oggettivamente. Questo suggerisce che la rappresentazione appare, per gli utenti, migliore di quanto le metriche suggeriscano.

In conclusione, le valutazioni soggettive sono generalmente confermate dalle metriche. Ci sono alcune minori eccezioni, ma, tutto sommato, si nota una consistente corrispondenza tra risultati soggettivi e oggettivi.

Capitolo 6

Conclusioni

6.1 Considerazioni finali

Questa tesi ha permesso di mettere in luce i vantaggi e gli svantaggi che la cattura volumetrica può apportare all'interno di una produzione Rai e in quali contesti sia possibile integrarla. La possibilità di effettuare in post-produzione il relighting e i movimenti di camera permettono una maggiore libertà artistica e una riduzione di costi. Inoltre, tale tecnologia permette di eliminare la fase di modellazione ed animazione 3D, in quanto tutto il processo si basa semplicemente su una cattura in live action. Questo rende la procedura di costruzione di avatar fotorealistici più efficiente e veloce.

La tecnologia che si presta maggiormente ad un utilizzo in un contesto televisivo è quella relativa all'applicazione Volu. Quest'ultima garantisce, infatti, una cattura volumetrica fotorealistica accettabile per gli standard televisivi. V3LCamera e Depthkit, invece, restituiscono una ricostruzione volumetrica dopo un tempo di elaborazione molto breve, ma poco realistica. Il loro utilizzo, dunque, potrebbe orientarsi a contesti differenti. Ad esempio, si potrebbe pensare di sfruttare degli VFX per mascherare i difetti della cattura. Tutte e tre le applicazioni presentano dei costi relativamente elevati. L'applicazione più conveniente è V3LCamera ma, considerando i risultati ottenuti, la scelta basata sul rapporto qualità-prezzo ricade sempre su Volu. I test soggettivi hanno ulteriormente confermato tali constatazioni.

L'analisi condotta sulla cattura volumetrica in real time ha permesso di individuare diverse problematiche, limitanti per un utilizzo in una produzione televisiva.

Infine, i risultati volumetrici ottenuti dall'algoritmo Gaussian Splatting 4D possono essere considerati soddisfacenti, anche in alcune situazioni critiche.

6.2 Limiti

Lo stato dell'arte attuale della cattura volumetrica presenta diverse limitazioni. In primo luogo, i risultati ottenibili dalle applicazioni a basso costo raggiungono risultati di qualità nettamente inferiore rispetto alle catture volumetriche prodotte dai grandi studi di produzione. Inoltre, la bassa concorrenza sul mercato attuale di applicazioni per la cattura volumetrica permette di mantenere i costi relativamente alti. Un'ulteriore limitazione si riscontra nei tempi di elaborazione in quanto la produzione televisiva ha spesso dei vincoli stringenti in tale ambito. La soluzione migliore sarebbe il real time ma, per adesso, i risultati sono di qualità così bassa da renderli inutilizzabili in programmi televisivi.

Infine, il Gaussian Splatting 4D potrebbe essere uno strumento interessante considerati i suoi costi praticamente nulli, grazie alla disponibilità di codice open source. La sua inutilizzabilità al giorno d'oggi si individua principalmente nella mancata possibilità di integrazione in motori grafici, ovvero l'impossibilità di inserire il contenuto volumetrico in un ambiente virtuale. In aggiunta a ciò, non è possibile isolare la figura rispetto all'ambiente in cui è stata catturata e ciò compromette i vantaggi di una cattura volumetrica di un soggetto. La possibilità di accedere al codice per apportare modifiche a piacimento è sicuramente un grande vantaggio e punto di forza. D'altra parte, è difficile, per utenti non esperti, capire l'algoritmo ed essere in grado di sfruttarlo al meglio. Il Gaussian Splatting 4D, allo stato attuale, richiede, a chi desidera utilizzarlo, diverse competenze informatiche, oltre che GPU e workstation performanti. Inoltre, anche la qualità della ricostruzione dovrebbe essere migliorata e i tempi di elaborazione notevolmente ridotti per poter pensare a un utilizzo di questa tecnologia in produzioni televisive e cinematografiche.

6.3 Sviluppi futuri

È cruciale sottolineare come i recenti progressi nel deep learning abbiano determinato significativi avanzamenti nelle tecnologie impiegate per la cattura volumetrica e una considerevole riduzione dei costi. Si prevede che l'accessibilità continuerà a crescere, con un aumento delle soluzioni a basso costo disponibili sul mercato e un progressivo miglioramento della qualità delle ricostruzioni volumetriche.

Nella ricerca futura potrebbe essere interessante approfondire la cattura volumetrica in tempo reale. Il processo per ottenerla è già esistente, ma i requisiti di qualità necessari per una produzione televisiva attualmente non vengono rispettati. Ulteriori sviluppi e studi potrebbero introdurre cambiamenti significativi e permettere il connubio tra ridotta velocità di elaborazione e ottima qualità, ideale per la produzione televisiva.

Per quanto riguarda la cattura volumetrica non in tempo reale, si ritiene che Volu sia l'applicazione con il più alto potenziale di sviluppo futuro. L'azienda Volograms ha dichiarato che introdurrà nel proprio sistema le diffusion networks. Questo cambiamento rappresenterà sicuramente un avanzamento promettente, che potrà portare a una serie di miglioramenti significativi. Le diffusion networks (vedi capitolo 3, paragrafo 3.1.2) potrebbero consentire a Volograms di ottenere ricostruzioni volumetriche più dettagliate, texture più realistiche e maggiore coerenza visiva. Si presume, quindi, che la qualità globale dei modelli ottenuti migliorerà. Si prevede inoltre che i tempi di elaborazione possano essere ulteriormente ottimizzati, rendendo la tecnologia più facilmente integrabile in contesti produttivi che richiedono tempi di elaborazione ridotti, come quelli televisivi. In conclusione, si consiglia di continuare a osservare gli sviluppi di Volograms, che si prospetta sempre più interessante per il futuro della cattura volumetrica.

Una delle direzioni più promettenti riguarda sicuramente l'approfondimento del Gaussian Splatting 4D, tecnologia emergente e, per tale ragione, con ancora un ampio margine di miglioramento. La controparte statica è una tecnica di rendering e ricostruzione 3D estremamente nuova e, infatti, la ricerca che ha dato inizio alla sua diffusione risale ad agosto 2023 [10]. Nonostante la sua recente introduzione, il suo potenziale rivoluzionario è emerso immediatamente e già oggi si possono osservare numerose applicazioni e studi che lo sfruttano. Tale soluzione tecnologica, applicata alla ricostruzione di scene in movimento, è ancora più recente: i primi studi risalgono a dicembre 2023 [11] e, quindi, la ricerca e gli studi sull'argomento sono agli albori. Ciò nonostante, negli ultimi mesi,

sono stati pubblicate numerose ricerche che sfruttano quest'algoritmo per le rappresentazioni digitali umane. Il Gaussian Splatting 3D, utilizzato per la ricostruzione di avatar successivamente animabili, ha già dimostrato notevole potenziale grazie alle ricerche che ne esplorano le applicazioni nelle ricostruzioni virtuali di persone [71] [72] [73]. Tra gli studi più interessanti ne emerge uno sulla generazione di modelli umani 3D realistici a partire da un prompt di testo [74]. La possibilità di creare modelli unici e personalizzati sfruttando solo qualche parola di testo e in tempi brevi è un avanzamento significativo, reso possibile dall'utilizzo del Gaussian Splatting.

La ricerca sul Gaussian Splatting 4D è meno numerosa, ma da non sottovalutare. L'integrazione della quarta dimensione, a differenza dell'equivalente 3D, consente di ottenere modelli umani già in movimento, annullando la necessità di animare avatar digitali. Questa tecnologia potrebbe apportare significativi cambiamenti all'intera pipeline di produzione audiovisiva, permettendo l'adozione di soluzioni a costo ridotto, ma di alta qualità.

Infine, l'aggiunta di VFX alle ricostruzioni digitali, in questo lavoro di tesi soltanto accennata, potrebbe essere approfondita, nell'ottica di analizzare usi anche insoliti dello strumento. Per una produzione televisiva, può essere necessario cercare una maggiore spettacolarità nei programmi trasmessi e le ricostruzioni volumetriche con l'aggiunta di effetti visivi potrebbero rappresentare un originale modo per ottenerla.

In definitiva, il futuro della cattura volumetrica appare estremamente promettente. Le continue innovazioni e la crescente accessibilità di questa tecnologia permetteranno di migliorare sempre di più le rappresentazioni digitali offerte.

Appendice

Questionario sottoposto all'utente per le applicazioni (Volu, V3LCamera e Depthkit)

Cattura volumetrica - Applicazioni

B I U ↻ ✖

Tesi di laurea magistrale Ingegneria del Cinema e dei mezzi di comunicazione

Quanto è integrato il soggetto con l'ambiente circostante? *

5 - Molto

4 - Abbastanza bene

3 - Normale

2 - Poco

1 - Per niente

Quanto è realistico il soggetto? *

⋮

5 - Molto

4 - Abbastanza

3 - Normale

2 - Poco

1 - Pessimo

Come valuti le espressioni facciali? *

5 - Eccellenti

4 - Buone

3 - Normali

2 - Scarse

1 - Pessime

Come valuti il movimento delle braccia e delle mani? *

- 5 - Eccellente
- 4 - Buono
- 3 - Normale
- 2 - Scarso
- 1 - Pessimo

Come valuti in generale il video? *

- 5 - Eccellente
- 4 - Buono
- 3 - Normale
- 2 - Scarso
- 1 - Pessimo

Potrebbe funzionare per una produzione televisiva? *

- 5 - Decisamente si
- 4 - Sì
- 3 - Né sì né no
- 2 - No
- 1 - Decisamente no

Questionario sottoposto all'utente per il real time

Cattura volumetrica - Real time

Tesi di laurea magistrale Ingegneria del Cinema e dei mezzi di comunicazione

Come valuti il tuo movimento riprodotto? *

- 5 - Eccellente
- 4 - Buono
- 3 - Normale
- 2 - Povero
- 1 - Pessimo

Quanto reputi reattivo il sistema? *

- 5 - Molto
- 4 - Abbastanza
- 3 - Normale
- 2 - Poco
- 1 - Per niente

Quanto reputi realistica la tua riproduzione digitale? *

- 5 - Eccellente
- 4 - Buona
- 3 - Normale
- 2 - Povera
- 1 - Pessima

Quanto migliora la tua riproduzione digitale man mano che ti avvicini alla webcam? *

- 5 - Molto
- 4 - Abbastanza
- 3 - Normale
- 2 - Poco
- 1 - Per niente

Potrebbe funzionare per una produzione televisiva? *

- 5 - Decisamente si
- 4 - Sì
- 3 - Né sì né no
- 2 - No
- 1 - Decisamente no

Scrivi qui quali aspetti positivi e negativi hai notato

Testo risposta lunga

Questionario sottoposto all'utente per il Gaussian Splatting 4D

Cattura volumetrica - Gaussian Splatting

Tesi di laurea magistrale Ingegneria del Cinema e dei mezzi di comunicazione

Quanto è fluido il movimento della camminata? *

- 5 - Molto
- 4 - Abbastanza
- 3 - Normale
- 2 - Poco
- 1 - Per niente

Quanto risulta realistico il soggetto? *

- 5 - Molto
- 4 - Abbastanza
- 3 - Normale
- 2 - Poco
- 1 - Per niente

Come valuti le espressioni facciali? *

- 5 - Eccellenti
- 4 - Buone
- 3 - Normali
- 2 - Scarse
- 1 - Pessime

Come valuti il movimento del soggetto in generale? *

- 5 - Eccellente
- 4 - Buono
- 3 - Normale
- 2 - Scarso
- 1 - Pessimo

Quanto è simile al video originale? *

- 5 - Molto
- 4 - Abbastanza
- 3 - Normale
- 2 - Poco
- 1 - Per niente

Come valuti la ricostruzione dei vestiti? *

- 5 - Eccellente
- 4 - Buona
- 3 - Normale
- 2 - Scarsa
- 1 - Pessima

Come valuti il movimento delle mani e delle braccia? *

- 5 - Eccellente
- 4 - Buono
- 3 - Normale
- 2 - Scarso
- 1 - Pessimo

Come valuti la riproduzione degli occhiali? *

- 5 - Eccellente
- 4 - Buona
- 3 - Normale
- 2 - Scarsa
- 1 - Pessima

Come valuti il movimento delle gambe? *

- 5 - Eccellente
- 4 - Buono
- 3 - Normale
- 2 - Scarso
- 1 - Pessimo

Come valuti il video in generale? *

- 5 - Eccellente
- 4 - Buono
- 3 - Normale
- 2 - Scarso
- 1 - Pessimo

Acronimi

2D: Bi-dimensional, Bidimensionale

2.5D: 2.5 Dimensional, Due dimensioni e mezzo

3D: Three-dimensional, Tridimensionale

3D-GS, 3D Gaussian Splatting, Gaussian Splatting 3D

4D: Four-dimensional, Quadridimensionale

4D-GS, 4D Gaussian Splatting, Gaussian Splatting 4D

ACR: Absolute Category rating, Valutazione della categoria assoluta

AI: Artificial Intelligence, Intelligenza artificiale

AR: Augmented Reality, Realtà Aumentata

AIGC: Artificial Intelligence Generated Content, Contenuti generati dall'intelligenza artificiale

CAGR: Compounded Average Growth Rate, Tasso annuo di crescita composto

CG: Computer Graphics, Computer grafica

CNN: Convolutional Neural Network, Rete neurale convoluzionale

D-SSIM: Structural Dissimilarity Index Measure, La misura dell'indice di dissomiglianza strutturale

DIF: Deep Implicit Function, Funzione implicita profonda

DNN: Deep Neural Network, Rete neurale profonda

DoF: Degree of Freedom, Gradi di libertà

DSF: Dynamic Sliding Fusion, Fusione dinamica scorrevole

FPS: Frames Per Second, Fotogrammi al secondo

GPU: Graphics Process Unit, Unità di elaborazione grafica

GPT: Generative Pre-trained Transformer, Trasformatore generativo pre-addestrato

GS: Gaussian Splatting, Splatting gaussiano

- HDRP:** High Definition Render Pipeline, Pipeline di render ad alta definizione
- HMD:** Head-Mounted Display
- LiDAR:** Laser Imaging Detection and Ranging, Rilevazione e misurazione della luce laser
- LLM:** Large Language Model, Modello di apprendimento specializzato nella comprensione del linguaggio
- LPIPS:** Learned Perceptual Image Patch Similarity, Similitudine percettiva appresa di patch di immagini
- MS-SSIM:** Multiscale Structural Similarity Index Measure, La misura dell'indice di somiglianza strutturale multiscala
- MLP,** Multilayer Perceptron, Perceptrone multistrato
- NeRF(s):** Neural Radiance Field(s), Campo di radianza neurale
- NVS:** Novel View Synthesis, Sintesi nuove viste
- PSNR:** Peak Signal-to-Noise Ratio, Rapporto segnale rumore di picco
- PTP:** Precision Time Protocol, Protocollo di precisione temporale
- RGB:** Red Green Blue, ROSSO VERDE BLU
- RGB-D:** Red Green Blue- Depth, ROSSO VERDE BLU- PROFONDITÀ
- SAM:** Segment Anything Model, Modello Segment Anything
- SfM:** Structure from Motion, Struttura dal movimento
- SIDE:** Single-Image Depth Estimation, Stima della profondità da un'unica immagine
- SSIM:** Structural Similarity Index Measure, Misura dell'indice di somiglianza strutturale
- ToF:** Time of flight, Tempo di volo
- URP:** Universal Render Pipeline, Pipeline di render universale
- VFX:** Visual Effects, Effetti visivi
- VR:** Virtual reality, Realtà virtuale
- XR:** Extended Reality, Realtà estesa

Lista delle figure

Figura 1.1: Tre gradi di libertà vs sei gradi di libertà	2
Figura 1.2: Madonna ai Billboards Music Awards (2019)	4
Figura 1.3: Live The Voice USA.....	4
Figura 1.4: Studio subacqueo di cattura volumetrica Volucap	5
Figura 1.5: Cattura volumetrica utilizzata per una partita di basket.....	6
Figura 2.1: Esempio di mesh.....	13
Figura 2.2: Esempio di nuvola di punti	13
Figura 2.3: Esempio di voxel.....	14
Figura 2.4: Ricostruzione ottenuta con la tecnica NeRF da diversi punti di vista	15
Figura 2.5: Due ricostruzioni ottenute con il Gaussian Splatting. A sinistra la gaussiane sono completamente opache. A destra ogni gaussiana ha il proprio valore di trasparenza.	16
Figura 2.6: Confronto del processo di rendering del NeRF e del Gaussian Splatting...	20
Figura 2.7: Esempio di Head Mounted Display (HMD)	20
Figura 2.8: Array di camere.....	21
Figura 2.9: Camera RGB per la volumetric capture dell'IO Industries.....	22
Figura 2.10: Intel® RealSense™ Depth Camera D455	23
Figura 2.11: A sinistra mappa della texture. A destra mappa di profondità	23
Figura 2.12: Tecnologia LiDAR.....	24
Figura 2.13: Schema Structure From Motion.....	25
Figura 2.14: Microsoft Mixed Reality Capture Studio	27
Figura 2.15: Intel Studios	28
Figura 2.16: Cattura volumetrica realizzata da Infinite Realities studio	29
Figura 2.17: Holosys by 4DViews	29
Figura 2.18: Esterno ed interno di NKH Meta Studio.....	31
Figura 2.19: Valutazione della fusione scorrevole dinamica.....	32
Figura 3.1: Cattura volumetrica ottenuta con l'app Volu per Hugo Boss	35
Figura 3.2: Cattura volumetrica ottenuta con l'app Volu per la National Gallery di Londra	35

Figura 3.3: Utilizzi di catture volumetriche ottenute con l'app Volu nello studio televisivo Fox Sports.....	36
Figura 3.4: Le tre diverse schermate dell'app Volu	37
Figura 3.5: Da sinistra verso destra vengono mostrate la segmentazione semantica, la stima fotometrica delle normali, la ricostruzione monoculare volumetrica e la sintesi della texture occlusa	38
Figura 3.6: Files ottenuti con l'esportazione dall'app Volu.....	39
Figura 3.7: Files ottenuti tramite il plugin per l'integrazione su Blender.....	40
Figura 3.8: A sinistra setup per cattura volumetrica in live streaming. A destra il risultato della cattura volumetrica in live streaming.....	42
Figura 3.9: Pacchetti disponibili per Depthkit.....	42
Figura 3.10: A sinistra video musicale 'Rap God' di Eminem. A destra il cortometraggio 'Zero days VR'.	43
Figura 3.11: Interfaccia applicazione desktop	44
Figura 3.12: A sinistra la visualizzazione del near e del far planes nella viewport 3D. A destra la finestra dell'applicazione per l'inserimento della matre e di parametri vari per l'isolamento del soggetto.....	44
Figura 3.13: Parametri che permettono il miglioramento della clip volumetrica.....	45
Figura 3.14: Schema funzionamento sensore Time-of-Flight (ToF).....	47
Figura 3.15: Pagina di Epic Games in cui è possibile acquistare il plugin Velox Player	51
Figura 3.16: Screen della schermata di V3LCamera.....	52
Figura 3.17: Screenshot di Unreal che mostra i passaggi necessari per ottenere un video volumetrico di una persona con il plugin Velox Player Plus	53
Figura 3.18: Instance segmentation.....	54
Figura 3.19: Esempi di AdaAttN	55
Figura 3.20: Esempio della viewport del progetto demo di Velox Neuro.....	56
Figura 3.21: Screenshot del sito ufficiale di Velox in cui vengono specificate le compatibilità di sistema.....	57
Figura 3.22: Pipeline completa del Gaussian Splatting 4D.....	61
Figura 3.23: Comandi per l'installazione di nerfstudio e il processamento delle immagini con colmap.....	62

Figura 3.24: Output di Colmap	63
Figura 3.25: Contenuto della cartella "colmap"	63
Figura 3.26: Contenuto della cartella sparse	63
Figura 3.27: Comando per il processamento con più camere	63
Figura 3.28: Output del processamento con più camere	64
Figura 3.29: Comando generico per il training	64
Figura 3.30: Screenshot del viewer collegato al paper con la rappresentazione offerta dalle gaussiane splattate.....	65
Figura 3.31: Screenshot del viewer collegato al paper con la rappresentazione sotto forma di nuvola di punti.....	65
Figura 3.32: Screenshot del viewer collegato al paper con la rappresentazione con ellipsoidi.....	65
Figura 3.33: Screenshot del viewer da browser	66
Figura 3.34: Esempio di comando per il training	66
Figura 3.35: Esempio di comando per la valutazione finale del modello	66
Figura 3.36: Output del training e del rendering di un video con il 4D-GS.....	67
Figura 3.37: Esempio di risultato nella cartella "coarsetrain_render"	67
Figura 3.38: Esempio di risultato ottenuto in "coarsetest_render".....	67
Figura 3.39: Risultato di "coarsetrain_render" all'ultima iterazione	68
Figura 3.40: Risultato dell'ultima interazione in "finetrain_render"	68
Figura 3.41: Risultato dell'ultima interazione in "finetest_render"	68
Figura 4.1: Alcuni render da Blender frontali a figura intera di una cattura volumetrica realizzata con l'app Volu senza l'utilizzo di green screen	72
Figura 4.2: A sinistra dettaglio piedi. A destra dettaglio gambe.....	73
Figura 4.3: Dettaglio braccia e mani	73
Figura 4.4: A sinistra render a figura intera. A destra dettaglio buona ricostruzione della mano.	73
Figura 4.5: Viso ed espressioni facciali.....	74
Figura 4.6: Alcuni render da Blender posteriori a figura intera di una cattura volumetrica realizzata con l'app Volu senza l'utilizzo di green screen	75
Figura 4.7: Modello renderizzato di fronte, di profilo e di dietro dalla stessa posizione delle riprese.....	76

Figura 4.8: Modello renderizzato di fronte, di profilo e di dietro dalla posizione opposta rispetto alle riprese.....	76
Figura 4.9: Ricostruzione frontale e posteriore.....	77
Figura 4.10: Risultato del video volumetrico registrato con il green screen in background	78
Figura 4.11: Dettagli di gambe, braccia e capelli.....	78
Figura 4.12: Dettaglio delle mani.....	79
Figura 4.13: Esempi di spill.....	79
Figura 4.14: Render con la camera angolata in totale di 60°	80
Figura 4.15: Rendering a diverse luminosità, da sinistra a destra con luminosità crescente	80
Figura 4.16: Rendering con luci di diversi colori	81
Figura 4.17: Alcuni effetti predefiniti che offre l'app Volu	82
Figura 4.18: Video mp4 di una maschera animata	83
Figura 4.19: Parametri disponibili per migliorare il risultato ottenuto e schermata totale di Depthkit	84
Figura 4.20: Output ottenuti esportando nel formato "Combined per pixel video"	84
Figura 4.21: Output Depthkit su Unity a figura intera	85
Figura 4.22: Dettaglio sulle mani	86
Figura 4.23: A sinistra dettagli sui capelli. A destra dettagli sui bordi.....	86
Figura 4.24: Dettaglio sul viso in movimento.....	86
Figura 4.25: Figura intera ripresa di fronte	87
Figura 4.26: Risultato ottenuto con la camera posta di lato	87
Figura 4.27: Risultato ottenuto con la camera posta leggermente di lato	88
Figura 4.28: Esempio di ricostruzione di un'ombra	88
Figura 4.29: Risultati ottenuti a diverse luminosità, da sinistra a destra con luminosità crescente	89
Figura 4.30: Risultato ottenuto cambiando il colore delle luci	89
Figura 4.31: Risultato di un video volumetrico con effetti ottenuto su After Effects ...	90
Figura 4.32: Pannello relativo alla stima della profondità della cattura volumetrica....	91
Figura 4.33: Buco di informazione intorno al soggetto inquadrato	92
Figura 4.34: Ricostruzione volumetrica del soggetto e dell'ambiente.....	92

Figura 4.35: Artefatti presenti con un angolo di mobilità maggiore di 60 gradi	93
Figura 4.36: A sinistra un corretto rotoscoping. A destra un rotoscoping che presenta artefatti.....	94
Figura 4.37: Cattura volumetrica in cui non vi è stata l'eliminazione dell'oggetto reale con cui ha interagito il soggetto.....	94
Figura 4.38: Materiali disponibili e utilizzabili con le catture volumetriche realizzate con V3LCamera	95
Figura 4.39: Eliminazione parti del corpo del soggetto catturato	95
Figura 4.40: Artefatti che generano bidimensionalità nella cattura volumetrica.....	96
Figura 4.41: Ricostruzione volumetrica con un angolo di visibilità di 30 gradi a sinistra rispetto alla posizione centrale della camera utilizzata per la registrazione.....	96
Figura 4.42: Confronto della stessa cattura volumetrica con due illuminazioni differenti a livello di tinta.....	97
Figura 4.43: Confronto della stessa cattura volumetrica con diverse intensità luminosa. A sinistra una bassa luminosità, in centro luminosità media e a destra luminosità alta .	97
Figura 4.44: Ombre dinamiche e riflessioni.....	98
Figura 4.45: Object detection, matting mask e depth estimation nella cattura volumetrica real time	99
Figura 4.46: Vista frontale e laterale della ricostruzione volumetrica in tempo reale di molteplici soggetti contemporaneamente	99
Figura 4.47: A sinistra ricostruzione volumetrica a mezzo busto. A destra ricostruzione volumetrica a figura intera.....	100
Figura 4.48: Ricostruzione problematica delle mani.....	100
Figura 4.49: Artefatti generati nel profilo del soggetto inquadrato	101
Figura 4.50: Da sinistra a destra rendering con camera frontale di Volu, Depthkit e V3LCamera	103
Figura 4.51: Da sinistra a destra dettagli sulla mano che impugna il microfono di Volu, Depthkit e V3LCamera.....	103
Figura 4.52: Da sinistra a destra rendering con camera frontale di Volu, Depthkit e V3LCamera	104
Figura 4.53: Da sinistra a destra dettagli sulla mano che esegue un movimento veloce di Volu, Depthkit e V3LCamera	104

Figura 4.54: Da sinistra a destra rendering con la camera leggermente angolata Volu, Depthkit, e V3LCamera.....	105
Figura 4.55: A sinistra ricostruzione volumetrica ottenuta con holoportation. A destra cattura volumetrica ottenuta con Volu.	106
Figura 4.56: A sinistra ricostruzione del volto di profilo di holoportation. A destra ricostruzione volto di profilo di Volu.....	107
Figura 4.57: Viewer dinamico e statico	109
Figura 4.58: A sinistra render con movimenti veloci. A destra render con movimenti controllati.....	109
Figura 4.59: Degradazione delle informazioni visive a causa di movimenti veloci delle braccia e delle mani	110
Figura 4.60: Ricostruzione più fedele delle braccia e delle mani a causa di movimenti più controllati.....	110
Figura 4.61: A sinistra ricostruzione batch size pari a 2. A destra batch size pari a 3..	111
Figura 4.62: A sinistra ricostruzione batch size pari a 2 e iterazioni pari a 14 000. A destra batch size pari a 3 e iterazioni pari a 20 000.	112
Figura 4.63: Render della ricostruzione ottenuta con il canale alpha e vista dal viewer dinamico	113
Figura 4.64: A sinistra ricostruzione posteriore con movimento di camera di 180 gradi nel viewer. A destra ricostruzione posteriore con movimento di camera di 360 gradi nel viewer.	114
Figura 4.65: Degradazione delle informazioni nella ricostruzione del volto.	114
Figura 4.66: A sinistra render ricostruzione video registrato con l'Iphone 13. A destra render video registrato con la Sony α 7R IV	115
Figura 4.67: Render movimenti veloci delle braccia e delle mani	115
Figura 4.68: Render della ricostruzione del movimento della gonna.....	116
Figura 4.69: Render della ricostruzione di superfici riflettenti	117
Figura 4.70: In alto FPS rendering del video con 55 mila gaussiane. In basso FPS rendering del video contenente 58 mila gaussiane.	118
Figura 5.1: Setup per i video per i test di Volu, V3LCamera e Depthkit.....	120
Figura 5.2: A sinistra setup per uno dei video realizzato per il test del Gaussian Splatting. A destra la fotocamera professionale utilizzata.	121

Figura 5.3: A sinistra schermata con titolo iniziale. A destra risultato proposto	122
Figura 5.4: Schermata con QR code da scannerizzare	122
Figura 5.5: Esempio di schermata d'introduzione	123
Figura 5.6: A sinistra video originale. A destra video ricostruito	123
Figura 5.7: Ricostruzione di Volu.....	124
Figura 5.8: Ricostruzione di Depthkit.....	127
Figura 5.9: Ricostruzione con V3LCamera.....	129
Figura 5.10: A sinistra video green screen originale. A destra video green screen ricostruito con il 4D-GS	134
Figura 5.11: A sinistra video viso originale. A destra video viso ricostruito con il 4D-GS	137
Figura 5.12: A sinistra video camminata originale. A destra video camminata ricostruito con il 4D-GS.....	140
Figura 5.13: A sinistra video ginnastica originale. A destra video ginnastica ricostruito con il 4D-GS.....	144
Figura 5.14: Formula per il calcolo del DSSIM.....	154

Lista dei grafici

Grafico 5.1: Quesito di valutazione sulla ricostruzione del movimento delle braccia e delle mani	125
Grafico 5.2: In alto quesito di valutazione sull'integrazione del soggetto con l'ambiente virtuale. In basso quesito di valutazione sulla ricostruzione delle espressioni facciali.	125
Grafico 5.3: Quesito di valutazione sul realismo del soggetto ricostruito.....	126
Grafico 5.4: Quesito di valutazione sulla possibilità di utilizzo in una produzione televisiva.....	126
Grafico 5.5: Quesito di valutazione sul video in generale.....	127
Grafico 5.6: In alto quesito di valutazione sull'integrazione del soggetto con l'ambiente virtuale. In basso quesito di valutazione sulla ricostruzione delle espressioni facciali.	128
Grafico 5.7: Quesito di valutazione sulla possibilità di utilizzo in una produzione televisiva.....	129
Grafico 5.8: Quesito di valutazione sul realismo del soggetto ricostruito.....	130
Grafico 5.9: In alto quesito di valutazione sull'integrazione del soggetto con l'ambiente virtuale. In basso quesito di valutazione sulla ricostruzione delle espressioni facciali.	130
Grafico 5.10: Quesito di valutazione sulla ricostruzione del movimento delle braccia e delle mani	131
Grafico 5.11: Quesito di valutazione sulla possibilità di utilizzo in una produzione televisiva.....	131
Grafico 5.12: Istogramma per il confronto delle applicazioni Volu, Depthkit e V3LCamera	133
Grafico 5.13: Quesito di valutazione sul realismo del soggetto ricostruito nel video su green screen	134
Grafico 5.14: Quesito di valutazione in generale del video su green screen.....	135
Grafico 5.15: Quesito di valutazione sulla somiglianza tra video originale e ricostruito del video su green screen.....	135
Grafico 5.16: Quesito di valutazione sulle espressioni facciali del video su green screen	136
Grafico 5.17: Quesito di valutazione sui movimenti del soggetto del video su green screen.....	136

Grafico 5.18: Quesito di valutazione sul movimento di braccia e gambe del video su green screen	136
Grafico 5.19: Quesito di valutazione sul realismo del soggetto ricostruito nel video sulle espressioni facciali.....	137
Grafico 5.20: Quesito di valutazione in generale del video sulle espressioni facciali	138
Grafico 5.21: Quesito di valutazione sulla somiglianza tra video originale e ricostruito del video sulle espressioni facciali	138
Grafico 5.22: Quesito di valutazione sul movimento del soggetto in generale nel video sulle espressioni facciali	139
Grafico 5.23: Quesito di valutazione sulle espressioni facciali nel video del viso	139
Grafico 5.24: Quesito di valutazione sulla riproduzione degli occhiali del video sulle espressioni facciali.....	140
Grafico 5.25: Quesito di valutazione sul realismo del soggetto ricostruito nel video della camminata.....	141
Grafico 5.26: Quesito di valutazione in generale del video della camminata	141
Grafico 5.27: Quesito di valutazione sulla somiglianza tra video originale e ricostruito del video della camminata	142
Grafico 5.28: Quesito di valutazione sulle espressioni facciali del video della camminata	142
Grafico 5.29: Quesito di valutazione sulla fluidità del movimento nel video della camminata.....	143
Grafico 5.30: Quesito di valutazione sul movimento in generale nel video della camminata.....	143
Grafico 5.31: Quesito di valutazione sulla ricostruzione dei vestiti nel video della camminata.....	144
Grafico 5.32: Quesito di valutazione sul realismo del soggetto ricostruito nel video della ginnastica	145
Grafico 5.33: Quesito di valutazione in generale del video della ginnastica	145
Grafico 5.34: Quesito di valutazione sulla somiglianza tra video originale e ricostruito del video della ginnastica	146
Grafico 5.35: Quesito di valutazione sul movimento generale del soggetto nel video della ginnastica	146

Grafico 5.36: Quesito di valutazione sul movimento di mani e braccia nel video della ginnastica.....	147
Grafico 5.37: Quesito di valutazione sul movimento delle gambe nel video della ginnastica.....	147
Grafico 5.38: Quesito di valutazione sulle espressioni facciali nel video della ginnastica	147
Grafico 5.39: Istogramma per il confronto tra i diversi video ottenuti con il gaussian splatting 4D	148
Grafico 5.40: Quesito di valutazione sul movimento riprodotto in real time.....	149
Grafico 5.41: Quesito di valutazione sulla reattività del sistema in real time.....	150
Grafico 5.42: Quesito di valutazione sul realismo della riproduzione digitale in real time	150
Grafico 5.43: Quesito di valutazione sul miglioramento della riproduzione digitale in real time se ci si avvicina alla webcam.....	151
Grafico 5.44: Quesito di valutazione sulla possibilità di utilizzo in una produzione televisiva.....	151

Lista delle tabelle

Tabella 3.1: Confronto tra le varie soluzioni tecnologiche.....	70
Tabella 5.1: Confronto tra i valori delle metriche oggettive per i vari video ricostruiti con il Gaussian Splatting 4D.....	156

Bibliografia e sitografia

- [1] K. Pietroszek e C. Eckhardt, «Volumetric capture for narrative films», nov. 2020, pp. 1–3. doi: 10.1145/3385956.3422116.
- [2] Y. Jin, K. Hu, J. Liu, F. Wang, e X. Liu, «From Capture to Display: A Survey on Volumetric Video», 11 settembre 2023, *arXiv*: arXiv:2309.05658. doi: 10.48550/arXiv.2309.05658.
- [3] J. Usón e J. Cabrera, «Analysis and Development of Deep Learning Depth Estimation Techniques for Volumetric Capture and Free Viewpoint Video», apr. 2024, pp. 520–523. doi: 10.1145/3625468.3652913.
- [4] «Madonna’s groundbreaking augmented reality performance at Billboard Music Awards | Dimension». Consultato: 30 luglio 2024. [Online]. Disponibile su: <https://dimensionstudio.co/news/madonna-billboard/>
- [5] «Coldplay x BTS: “My Universe” | Dimension». Disponibile su: <https://dimensionstudio.co/work/coldplay-bts-my-universe-holograms/>
- [6] «The Volumetric Future: Coldplay x BTS on The Voice | Dimension». Disponibile su: <https://dimensionstudio.co/news/coldplay-bts-voice-my-universe-holograms/>
- [7] «The Matrix Resurrections», Volucap. Disponibile su: <https://volucap.com/portfolio-items/the-matrix-resurrections/>
- [8] J. Heagerty *et al.*, «HoloCamera: Advanced Volumetric Capture for Cinematic-Quality VR Applications», *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, fasc. 5, pp. 2767–2775, mag. 2024, doi: 10.1109/TVCG.2024.3372123.
- [9] A. Pumarola, E. Corona, G. Pons-Moll, e F. Moreno-Noguer, «D-NeRF: Neural Radiance Fields for Dynamic Scenes», 27 novembre 2020, *arXiv*: arXiv:2011.13961. doi: 10.48550/arXiv.2011.13961.
- [10] B. Kerbl, G. Kopanas, T. Leimkuehler, e G. Drettakis, «3D Gaussian Splatting for Real-Time Radiance Field Rendering», *ACM Trans. Graph.*, vol. 42, fasc. 4, pp. 1–14, ago. 2023, doi: 10.1145/3592433.
- [11] G. Wu *et al.*, «4D Gaussian Splatting for Real-Time Dynamic Scene Rendering», 15 luglio 2024, *arXiv*: arXiv:2310.08528. doi: 10.48550/arXiv.2310.08528.

- [12] Z. Li, H. Li, e L. Meng, «Model Compression for Deep Neural Networks: A Survey», *Computers*, vol. 12, fasc. 3, Art. fasc. 3, mar. 2023, doi: 10.3390/computers12030060.
- [13] S. Niedermayr, J. Stumpfegger, e R. Westermann, «Compressed 3D Gaussian Splatting for Accelerated Novel View Synthesis», 22 gennaio 2024, *arXiv*: arXiv:2401.02436. doi: 10.48550/arXiv.2401.02436.
- [14] «What are RGBD cameras? Why RGBD cameras are preferred in some embedded vision applications?», e-con Systems. Disponibile su: <https://www.e-consystems.com/blog/camera/technology/what-are-rgb-d-cameras-why-rgb-d-cameras-are-preferred-in-some-embedded-vision-applications/>
- [15] «What is LiDAR and How Does it Work? | Synopsys». Disponibile su: <https://www.synopsys.com/glossary/what-is-lidar.html>
- [16] «What is Structure from Motion? - MATLAB & Simulink - MathWorks Italia». Disponibile su: <https://it.mathworks.com/help/vision/ug/what-is-structure-from-motion.html>
- [17] «Camera Triangulation for depth and distance analysis», Medium. Disponibile su: <https://medium.com/@rohinfablabz/camera-triangulation-for-depth-and-distance-analysis-6e9da94cc9d7>
- [18] «Tutorial #A10 - Compensazione a stelle proiettive (bundle adjustment)», 3Dflow. Disponibile su: <https://www.3dflow.net/it/compensazione-a-stelle-proiettive-bundle-adjustment/>
- [19] A. Mertan, D. J. Duff, e G. Unal, «Single Image Depth Estimation: An Overview», *Digital Signal Processing*, vol. 123, p. 103441, apr. 2022, doi: 10.1016/j.dsp.2022.103441.
- [20] R. Garg, V. K. B.G., G. Carneiro, e I. Reid, «Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue», in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, e M. Welling, A c. di, Cham: Springer International Publishing, 2016, pp. 740–756. doi: 10.1007/978-3-319-46484-8_45.
- [21] «Microsoft Mixed Reality Capture Studios create holograms to educate and entertain», Source. Disponibile su: <https://news.microsoft.com/source/features/work-life/microsoft-mixed-reality-capture-studios-create-holograms-to-educate-and-entertain/>

- [22] «Intel Studios Showcases Volumetric Production at 77th Venice...», Intel. Disponibile su: <https://www.intel.com/content/www/us/en/newsroom/news/studios-volumetric-production-venice-film-festival.html>
- [23] «Intel Studios' Volumetric Capture Space Can Record 20 People at Once». Disponibile su: <https://variety.com/2019/digital/features/intel-studios-volumetric-capture-holograms-ar-vr-1203358126/>
- [24] «Spatial Capture – Infinite-Realities». Disponibile su: <https://www.ir-ltd.net/ir/4dgs/>
- [25] «4Dviews - Volumetric video capture technology». Disponibile su: <https://www.4dviews.com/>
- [26] «Realistic Capture and Production Using “Meta Studio” | Broadcast Technology», NHK STRL. Disponibile su: <https://www.nhk.or.jp/strl/english/publica/bt/91/3.html>
- [27] H. Morioka, T. Misu, T. Sugino-shita, e H. Mitsumine, «NHK Meta Studio: A Compact Volumetric TV Studio for 3-D Reconstruction», *IEEE Transactions on Broadcasting*, vol. 69, fasc. 1, pp. 2–9, mar. 2023, doi: 10.1109/TBC.2022.3210367.
- [28] T. Yu, Z. Zheng, K. Guo, P. Liu, Q. Dai, e Y. Liu, «Function4D: Real-time Human Volumetric Capture from Very Sparse Consumer RGBD Sensors», 6 maggio 2021, *arXiv*: arXiv:2105.01859. doi: 10.48550/arXiv.2105.01859.
- [29] J. Choe, S. Im, F. Rameau, M. Kang, I. Kweon, e I. Kweon, *VolumeFusion: Deep Depth Fusion for 3D Scene Reconstruction*. 2021.
- [30] K. Genova, F. Cole, A. Sud, A. Sarna, e T. Funkhouser, «Local Deep Implicit Functions for 3D Shape», 11 giugno 2020, *arXiv*: arXiv:1912.06126. doi: 10.48550/arXiv.1912.06126.
- [31] «Volumetric Video Market Size, Share, Industry Report, Revenue Trends and Growth Drivers», MarketsandMarkets. Disponibile su: <https://www.marketsandmarkets.com/Market-Reports/volumetric-video-market-259585041.html>
- [32] «Volograms — AI-powered 3D volumetric holograms». Disponibile su: <https://www.volograms.com/>
- [33] «Medium Article». Disponibile su: <https://www.volograms.com/medium-article?726032e17385>

- [34] J. González Escribano, S. Ruano, A. Swaminathan, D. Smyth, e A. Smolic, «Texture improvement for human shape estimation from a single image», ago. 2022. doi: 10.56541/SOWW6683.
- [35] «Introduction to Diffusion Models for Machine Learning», News, Tutorials, AI Research. Disponibile su: <https://www.assemblyai.com/blog/diffusion-models-for-machine-learning-introduction/>
- [36] «Depthkit». Disponibile su: <https://www.depthkit.tv/>
- [37] «VELOX». Disponibile su: <https://v3lox.com/#contact>
- [38] «In a world first, two innovative UK startups join forces at SXSW to capture and share immersive 3D interviews with just a single device», Music Technology UK. Disponibile su: <https://musictechnology.uk/two-innovative-uk-startups-join-forces-at-sxsw/>
- [39] «South by Southwest», *Wikipedia*. 13 luglio 2024. Disponibile su: https://en.wikipedia.org/w/index.php?title=South_by_Southwest&oldid=1234201348
- [40] «New Advanced settings», Google Docs. Disponibile su: https://docs.google.com/document/d/1VRg--EoyDMH0MvI5aVRewUeqe6DTyvunCSx0S9bUSrQ/edit?usp=sharing&usp=embed_facebook
- [41] C.-Y. Wang, A. Bochkovskiy, e H.-Y. M. Liao, «YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors», 6 luglio 2022, *arXiv*: arXiv:2207.02696. doi: 10.48550/arXiv.2207.02696.
- [42] K.-Y. Wong, *WongKinYiu/yolov7*. (1 agosto 2024). Jupyter Notebook. Disponibile su: <https://github.com/WongKinYiu/yolov7>
- [43] S. Lin, L. Yang, I. Saleemi, e S. Sengupta, «Robust High-Resolution Video Matting with Temporal Guidance», 25 agosto 2021, *arXiv*: arXiv:2108.11515. doi: 10.48550/arXiv.2108.11515.
- [44] «Segment Anything | Meta AI». Disponibile su: <https://segment-anything.com/>
- [45] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, e V. Koltun, «Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer», 25 agosto 2020, *arXiv*: arXiv:1907.01341. doi: 10.48550/arXiv.1907.01341.

- [46] «Rethinking Inductive Biases for Surface Normal Estimation». Disponibile su: <https://arxiv.org/html/2403.00712v1>
- [47] S. Liu *et al.*, «AdaAttN: Revisit Attention Mechanism in Arbitrary Neural Style Transfer», presentato al Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 6649–6658. Disponibile su: https://openaccess.thecvf.com/content/ICCV2021/html/Liu_AdaAttN_Revisit_Attention_Mechanism_in_Arbitrary_Neural_Style_Transfer_ICCV_2021_paper.html
- [48] «Inference in Machine Learning and Deep Learning: What is it? How to use it? Ultimate Guide 2023», Medium. Disponibile su: <https://londondataconsulting.medium.com/inference-in-machine-learning-and-deep-learning-what-is-it-how-to-use-it-ultimate-guide-2023-152fed531edc>
- [49] «Velox Neuro - Neural Network Inference Engine in Code Plugins - UE Marketplace», Unreal Engine. Disponibile su: <https://www.unrealengine.com/marketplace/en-US/product/velox-neuro-machine-learning-inference-engine>
- [50] A. S. A. Rabby e C. Zhang, «BeyondPixels: A Comprehensive Review of the Evolution of Neural Radiance Fields», 18 marzo 2024, *arXiv*: arXiv:2306.03000. doi: 10.48550/arXiv.2306.03000.
- [51] «Marca temporale», *Wikipedia*. 25 gennaio 2024. Disponibile su: https://it.wikipedia.org/w/index.php?title=Marca_temporale&oldid=137531260
- [52] J. Luiten, G. Kopanas, B. Leibe, e D. Ramanan, «Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis», 18 agosto 2023, *arXiv*: arXiv:2308.09713. doi: 10.48550/arXiv.2308.09713.
- [53] *hustvl/4DGaussians*. (16 agosto 2024). Jupyter Notebook. HUST Vision Lab. Disponibile su: <https://github.com/hustvl/4DGaussians>
- [54] «Output Format — COLMAP 3.11-dev documentation». Disponibile su: <https://colmap.github.io/format.html>
- [55] «yzslab/gaussian-splatting-lightning: A 3D Gaussian Splatting framework with various derived algorithms and an interactive web viewer». Disponibile su: <https://github.com/yzslab/gaussian-splatting-lightning>
- [56] «Animated Gaussian Splatting in Unreal Engine 5». Disponibile su: <https://80.lv/articles/animated-gaussian-splatting-in-unreal-engine-5/>

- [57] «Depthkit Tutorials». Disponibile su: <https://www.depthkit.tv/tutorials>
- [58] V. Tensil, «Home», XReco. Disponibile su: <https://xreco.eu/>
- [59] C. Casadei, «IA: Un po' di nozioni prima della pratica», maggiolidevelopers. Disponibile su: <https://www.developersmaggioli.it/blog/ia-un-po-di-nozioni-prima-della-pratica/>
- [60] «Epochs, Batch Size, Iterations - How they are Important». Disponibile su: <https://www.sabrepc.com/blog/Deep-Learning-and-AI/Epochs-Batch-Size-Iterations>
- [61] «BT.500 : Methodologies for the subjective assessment of the quality of television images». Disponibile su: <https://www.itu.int/rec/R-REC-BT.500-15-202305-I/en>
- [62] «Absolute Category Rating», *Wikipedia*. 22 maggio 2024. Disponibile su: https://en.wikipedia.org/w/index.php?title=Absolute_Category_Rating&oldid=1225151175
- [63] «SSIM: Structural Similarity Index | Imatest». Disponibile su: <https://www.imatest.com/docs/ssim/>
- [64] «Structural similarity index measure», *Wikipedia*. 29 febbraio 2024. Disponibile su: https://en.wikipedia.org/w/index.php?title=Structural_similarity_index_measure&oldid=1210955022
- [65] «Multi-Scale Structural Similarity — Data Quality Metrics 0.1 documentation». Disponibile su: https://quality.nfdi4ing.de/en/latest/image_quality/MultiScale_Structural_Similarity.html
- [66] «Peak Signal-to-Noise Ratio as an Image Quality Metric». Disponibile su: <https://www.ni.com/en/shop/data-acquisition-and-control/add-ons-for-data-acquisition-and-control/what-is-vision-development-module/peak-signal-to-noise-ratio-as-an-image-quality-metric.html>
- [67] «Peak signal-to-noise ratio», *Wikipedia*. 11 luglio 2024. Disponibile su: https://it.wikipedia.org/w/index.php?title=Peak_signal-to-noise_ratio&oldid=140147303
- [68] «Learned Perceptual Image Patch Similarity (LPIPS) — PyTorch-Metrics 1.4.1 documentation». Disponibile su:

- https://lightning.ai/docs/torchmetrics/stable/image/learned_perceptual_image_patch_similarity.html
- [69] S. Bangar, «VGG-Net Architecture Explained», Medium. Disponibile su: <https://medium.com/@siddheshb008/vgg-net-architecture-explained-71179310050f>
- [70] «AlexNet», *Wikipedia*. 4 luglio 2023. Disponibile su: https://it.wikipedia.org/w/index.php?title=AlexNet&oldid=134309335#cite_note-:1-1
- [71] «HUGS: Human Gaussian Splats», Apple Machine Learning Research. Disponibile su: <https://machinelearning.apple.com/research/hugs>
- [72] «ASH: Animatable Gaussian Splats for Efficient and Photoreal Human Rendering». Disponibile su: <https://vcai.mpi-inf.mpg.de/projects/ash/>
- [73] S. Hu e Z. Liu, «GauHuman: Articulated Gaussian Splatting from Monocular Human Videos», 5 dicembre 2023, *arXiv*: arXiv:2312.02973. doi: 10.48550/arXiv.2312.02973.
- [74] X. Liu *et al.*, «HumanGaussian: Text-Driven 3D Human Generation with Gaussian Splatting», 14 marzo 2024, *arXiv*: arXiv:2311.17061. doi: 10.48550/arXiv.2311.17061.

