UNIVERSITÀ
POLITECNICA
DELLE MARCHE

Facoltà di Ingegneria
Corso di Laurea Magistrale in
Ingegneria Informatica e dell'Automazione

Neural Radiance Fields: Architettura e Pipeline per il Cultural Heritage

Neural Radiance Fields:
Architecture and Pipeline in Cultural Heritage

Relatore: Chia.mo
Prof. Primo Zingaretti

Tesi di Laurea di:
David Caprari

Anno Accademico 2022-2023

UNIVERSITÀ
POLITECNICA
DELLE MARCHE

Facoltà di Ingegneria
Corso di Laurea Magistrale in
Ingegneria Informatica e dell'Automazione

Neural Radiance Fields: Architettura e Pipeline per il Cultural Heritage

Neural Radiance Fields:
Architecture and Pipeline in Cultural Heritage

Relatore: Chia.mo
Prof. Primo Zingaretti

Tesi di Laurea di:
David Caprari

Anno Accademico 2022-2023

Università Politecnica delle Marche
Facoltà di Ingegneria
Corso di Laurea Magistrale in
Ingegneria Informatica e dell'Automazione
Via Brecce Bianche – 60131 Ancona (AN), Italy

A Elena

Sommario

I campi di radianza neurale sono una nuova tecnologia per la rappresentazione di scene digitali in uno spazio tridimensionale. Questo lavoro vuole, a seguito di uno studio sperimentale che valuti scelte e tecnologie impiegate in alcune delle implementazioni più importanti per lo stato dell'arte, applicare queste nuove soluzioni in diversi ambienti pratici, in particolare in quelli legati alla conservazione del patrimonio culturale. Ciò per verificare quanto è rappresentabile da questi approcci e quali possano essere le migliori scelte tecniche, tra quelle a disposizione, per ottenere il più alto risultato possibile di rappresentazione contestualmente al Cultural Heritage. Per fare ciò, alcune architetture verranno testate con dataset e soluzioni allo stato dell'arte per Computer Graphics e Deep Learning. Dopodiché verrà effettuato un confronto tra le migliori e più recenti architetture disponibili su tre diversi dataset riportanti scene da sintetizzare legate al patrimonio artistico e culturale della regione Marche. Verranno quindi stilati diversi approcci relativi a diverse condizioni di acquisizione delle immagini di partenza.

Abstract

Neural Radiance Fields are a new technology regarding representation of digital scenes in a three-dimensional space. After an experimental study evaluating choices and technologies used in some of the most important implementations for the state of the art, this work has the objective of applying these new solutions in different practical environments, in particular in those related to the conservation of the Cultural Heritage. This to verify what these tool can represent and what might the best technical choices be, among those available, in obtaining the best representation result in the defined environment. In order to do this, some architectures will be tested with datasets and solution from the state of the art of Computer Graphics and Deep Learning. Afterwards, a report will be conducted confronting the best and most recent available architectures on three different datasets showing scenes to be synthesized related to the artistic and cultural heritage of the Marche region. Different approaches related to different conditions of the starting images acquisition will then be drawn up.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

Uno degli obiettivi fondanti della Computer Graphics, cioè di quella disciplina dell'informatica che si occupa della manipolazione di immagini e video attraverso i calcolatori digitali, è la sintesi di immagini fotorealistiche. Tradizionalmente, le immagini sintetiche ottenute a partire da una scena digitale tridimensionale vengono ricavate utilizzando algoritmi di rendering quali la rasterizzazione o il ray tracing, i quali assumono come input rappresentazioni matematiche digitali della geometria e dei materiali in uso nella scena. Questi input definiscono quindi la scena stessa, i suoi attributi e come e cosa deve essere riportato nell'immagine in output; ci si riferisce a ciò come ad una rappresentazione della scena, soprattutto nel caso in cui gli oggetti da rappresentare siano molteplici.

Il design digitale della scena è spesso un processo umano che richiede tempo ed abilità per la risoluzione. Coniugando queste esigenze pratiche con quelle di sempre maggiore fotorealismo degli output ottenuti sono nati processi di image-based modeling che si basano su differenze tra la rappresentazione della scena digitale e la scena reale stessa, simili all'idea di un manuale o automatico ricalco di un lucido che dia un'idea di come sia la realtà visiva. Negli ultimi anni, a partire dalla possibilità di definire una differenza palpabile e numerica tra la scena reale e ciò che si vuole ottenere in uno spazio digitale, sono nati diversi strumenti che relegano a processi di intelligenza artificiale basati su reti neurali il compito di ricostruire il task di modellazione e rendering. Questi strumenti, che molto spesso si basano su un processo di addestramento basato proprio su quella differenza, rientrano in un grande sottogruppo della Computer Graphics definito come Neural Rendering.

Gli approcci appena introdotti di Neural Rendering combinano idee derivanti dalla Computer Graphics classica e dal Machine Learning per creare algoritmi in grado di sintetizzare immagini attraverso spazi digitali tridimensionali a partire da osservazioni del mondo reale. Nel campo della CG, negli ultimi anni si è notata una vera e propria esplosione delle ricerche legate a questo argomento[1] e in generale di

tutte le tecniche di Deep Learning, sottogruppo del ML che fa uso di Reti Neurali Profonde (Deep Neural Networks).

In questo ramo di ricerca, uno degli eventi più importanti nell'ultimo lustro è legato alla prima presentazione dei Neural Radiance Fields (NeRF)[2], campi di radianza neurale che di questo lavoro sono colonna portante.

Questi strumenti, che con il massivo aumento di ricerche correlate hanno sviluppato una vera e propria famiglia, sfruttano un connubio di idee legate alla CG e al DL. In particolare, si assume che una rete neurale profonda possa essere in grado di rappresentare un'intera scena digitale tridimensionale a partire da una serie, molto spesso piccola e limitata, di immagini scattate ad un oggetto o un contesto. Le capacità di generalizzazione e ricostruzione della rete possono quindi essere sfruttate per effettuare la sintesi di immagini nuove e sintetiche in pose statiche che non erano state fornite durante il processo di addestramento. Processo che, basandosi sulla differenza tra quanto viene generato e quanto era atteso, porta quindi alla possibilità di addestrare questo sistema di intelligenza artificiale in modo che sia più preciso e accurato nella sintesi.

Con l'aumento dei ricercatori occupati ad affinare sempre più questi approcci, c'è stato un forte incremento dei possibili impieghi che queste tecniche possono ricoprire. Dalla tutela dell'ambiente ai beni culturali, dal cinema al campo dei videogiochi, dalla progettazione tecnica alla valutazione qualitativa delle strutture. Quando una pipeline di lavoro richiede la progettazione digitale di un oggetto o parti di una scena, i NeRF possono essere utilizzati per semplificare o sostituire processi della stessa, facilitando il lavoro e, quando eseguiti opportunamente, aumentando la resa grafica dell'output.

Come ogni nuova tecnologia, però, sono presenti diverse limitazioni e svantaggi pratici che non è sempre semplice superare. Dalla ingente richiesta di risorse alle complesse richieste specifiche, la lista di complessità a volte risulta troppo lunga per un utilizzo pratico e semplice nella vita di tutti i giorni da parte di un tecnico del settore. Questo lavoro riporta quindi l'intento di unirsi alla ricerca svolta su questa frontiera della CG con l'obbiettivo di migliorarne alcuni processi tecnico-pratici e semplificare l'utilizzo di questi potenti strumenti.

Ai fini di quanto descritto, si è scelto di studiare, comprendere ed infine applicare queste tecnologie al Cultural Heritage, con l'idea che questi strumenti possano svolgere un ruolo chiave nella preservazione dei beni materiali che rappresentano l'eredità culturale della nostra comunità. In particolare, a seguito di uno studio tecnologico del metodo, se ne vedrà l'applicazione a due edifici storici ed un monumento. Questo poiché, una rappresentazione tridimensionale accurata dell'interazione tra l'uomo e

l'ambiente nell'ambito del patrimonio materiale, può contribuirne al mantenimento ed alla presentazione.

1.1 Contesto e motivazione

Negli ultimi decenni sono stati sviluppati ed affinati diversi algoritmi con l'obbiettivo di emulare i modelli matematici di generazione delle immagini delle camere reali. Di questi, particolare attenzione in questo lavoro la avranno i modelli di illuminazione della scena; modelli che sono strettamente basati sulla fisica della luce visibile e su come questa si diffonda attraverso la scena per poi giungere all'occhio, o la camera, dell'osservatore. In particolare, a partire da modelli di fonti luminose, questi metodi di illuminazione della scena digitale consentono in coda l'intero processo di rendering, in quanto permettono di ottenere la quantità di luce riemessa dalle superfici. Come processo a valle, afferrato se la superficie tridimensionale sarà visibile o meno nella rappresentazione bidimensionale, resterà da individuarne la trama, o texture, e mostrare l'insieme di queste superfici, illuminate correttamente, come output finale.

Questo processo di generazione di immagini, rendering appunto, oltre a richiedere un'elevata corrispondenza qualitativa tra la realtà e la scena che la rappresenta, richiede molto spesso una capacità computazionale che può aumentare esponenzialmente in base alla qualità richiesta. Per questo motivo si è sempre prestata molta attenzione a questi algoritmi e diverse necessità legate ad una miriade di industrie hanno fatto sì che la ricerca di software renderer sempre più efficienti, supportati da hardware sempre più capace, fosse uno dei contesti chiave dell'informatica moderna.

Parallelamente allo sviluppo di questo ramo della CG, la legge di Moore legata a molta della componentistica hardware, e quindi l'aumento totale della capacità computazionale a disposizione, ha portato ad una esplosione dei dati che è possibile raccogliere[3] e da qui a diversissimi strumenti utilizzati per la loro analisi. In particolare, quando gli algoritmi digitali sono in grado di effettuare correzioni interne legate allo stato dei dati presi in analisi è possibile parlare di Machine Learning. Tra le tante famiglie facenti capo al ML, quella del Deep Learning sfrutta algoritmi sufficientemente generici capaci di approssimare il comportamento statistico di diversi modelli matematici attraverso un processo di apprendimento.

Processi che erano strettamente legati al settore del ML hanno quindi iniziato a permeare sempre di più diversi altri campi: nel caso della CG, uno dei settori più importanti in cui resta questo scambio di approcci è proprio il Neural Rendering già introdotto.

1.2 Obbiettivi e principale contribuzione

Il lavoro di tesi parte dall'esigenza ingegneristica di comprendere al meglio sia teoricamente che praticamente lo strumento per la generazione e visualizzazione di Neural Radiance Fields.

A questo proposito uno degli obiettivi fondanti è un attento studio dei moduli software che compongono diverse implementazioni. Si parte quindi dalla necessità di comprendere la struttura organizzativa ed informativa dei dataset più comuni e come le immagini vengono preprocessate da parte della pipeline generativa. A seguire è necessario comprendere in che modalità il contenuto informativo delle immagini, i canali rosso, verde e blu insieme ai metadati interni, vengono utilizzati per la sintesi tridimensionale e come vengano adattati per il passaggio attraverso gli strati di processo successivi. Lo studio deve poi spostarsi sull'architettura neurale dello strumento, affinché sia chiaro quali siano state le scelte implementative e come questi strumenti rispondono ingegneristicamente a ciò che gli viene richiesto fare. Contemporaneamente, è importante comprendere in che modo funzioni il processo di addestramento dell'architettura e quali sono gli accorgimenti utilizzati per aumentare le performance finali. A questo punto, manca comprendere in che modo sia possibile rappresentare l'output dell'architettura in uno spazio virtuale dinamico e con quali modalità sia poi possibile restituire, della scena 3D sintetizzata, una rappresentazione simile a quelle più utilizzate, quali mesh o nuvole di punti.

Al fine di effettuare questo studio comprensivo dell'intera pipeline di processo, è stato necessario organizzare diversi esperimenti, ognuno che comprenda uno o più punti della pipeline, sia per verificare che lo stato dell'arte sia replicabile opportunamente, ma anche per provare la risposta dell'architettura a singole modifiche, in modo da valutarne le singole componenti. Alle modifiche che comportano questi esperimenti non viene esplicitamente richiesto di migliorare lo stato dell'arte o di contribuire in modo significativo alla ricerca, bensì di dare una chiara idea al ricercatore sull'apporto dei singoli concetti alla base teorica complessiva.

A completare il lavoro, è stato obiettivo fondamentale della ricerca l'applicazione delle conoscenze così derivate in modo da soddisfare opportunamente un caso di studio. Applicazione che quindi vede centrale il risultato dell'acquisizione di due video e un insieme di scatti fotografici da parte di un drone di tre diversi siti culturali presenti nella regione Marche. In particolare, la tecnologia verrà applicata ad un video del santuario di Macereto, complesso religioso posto a circa 1000 metri s.l.m. nel versante occidentale dei Monti Sibillini a Visso; ad un video di alcuni ruderi del castello di Magalotti nel comune di Fiastra; infine, sempre nella zona dei Monti Sibillini, ad una serie di scatti fotografici della statua al centro di Piazza A. Gentili

a San Ginesio. Si ritiene che la sperimentazione di queste soluzioni nell'ambito del Cultural Heritage sia importante nel mantenimento del patrimonio culturale, sia a livello conservativo dell'immagine, sia in quanto permette studi sullo stato di salute dei monumenti e delle strutture.

1.3 Struttura della tesi

Questo lavoro è composto da questa introduzione e da una presentazione dello stato dell'arte in cui verranno descritti quali sono gli approcci classici e come e con quali architetture è possibile risolvere il task di generazione dei campi di radianza neurale. Per definire questi ultimi vi sarà la necessità di introdurre diverse nozioni legate al settore del Deep Learning e in particolare andranno introdotte le reti neurali definite Multi Layer Perceptron che verranno collocate alla base delle architetture più complesse che vogliamo presentare. Da qui, una volta discussa una primitiva architettura di NeRF e loro software accessori, saranno introdotte più brevemente un piccolo numero di alcune tra le implementazioni successive che negli ultimi anni si sono susseguite alla primissima idea, apportando forti miglioramenti e nuove capacità.

Inquadrato lo stato dell'arte e i filoni in cui la ricerca si sta concentrando contemporaneamente alla stesura di questo lavoro, nel capitolo successivo a quello appena descritto verranno riportate le descrizioni di alcune delle implementazioni software utilizzate sulle quali sono stati predisposti diversi esperimenti volti a verificare la validità delle scelte fatte per alcuni moduli della pipeline NeRF. Con queste prove si è voluto passare in esame ogni componente principale della pipeline: la raccolta dati, la trasformazione dei dati iniziali in feed per l'architettura neurale, l'architettura neurale stessa, parte del postprocessing del campo di radianza al fine di ottenere output riconducibile alle tecniche più diffuse di progettazione 3D.

Successiva a questa prima serie di esperimenti sulla pipeline generica di utilizzo, si è scelto di applicare lo strumento a diversi casi d'uso reali. In particolare, si vedranno i risultati che gli strumenti basati su campi di radianza neurale riescono ad ottenere dai tre siti storici citati in precedenza. Si vedranno quindi una serie di confronti basandosi su diverse pipeline, al fine di trovare una soluzione che possa garantire elevata qualità di rappresentazione con un discreto accesso da parte di tecnici interessati allo strumento.

Terminata la sezione riguardante esperimenti e applicazione degli strumenti, nel successivo capitolo verrà riportata una discussione dei risultati in cui, oltre a questi ultimi, si passerà in esame anche il metodo stesso.

Infine, a concludere il lavoro, verranno riportate le conclusioni su quanto svolto e diverse osservazioni legate al futuro di questa ricerca, con idee e stimoli per il lettore.

Capitolo 2

Stato dell'arte

Al fine di introdurre al meglio una serie di strumenti che saranno poi fondamentali per descrivere le radici di questo lavoro, è necessario fare un excursus preliminare nell'ambito del ML e del DL. In particolare, sarà necessario introdurre delle tecniche di ML basate su modelli di apprendimento, alcuni concetti alla base dell'Intelligenza Artificiale, le reti neurali MLP e alcuni processi legati al loro addestramento.

A seguire si passerà nel campo della CG pura per definire una serie di termini e tecniche base e, tra i sistemi di illuminazione globali delle scene tridimensionali, la radiosità insieme alla quale verranno introdotte brevemente alcune tecniche di ray tracing. Questo affinché si abbiano abbastanza nozioni sugli approcci classici da effettuare parallelismi con gli approcci neurali che verranno introdotti successivamente.

In coda, per l'appunto, sulla base di quanto definito, verranno introdotti i Neural Radiance Fields (NeRF) nella loro implementazione classica e primitiva. Ne verrà descritto l'approccio teorico, l'architettura, le tecnologie pratiche e ne verranno presentate le funzionalità. Introdurre la versione software capostipite risulta essere molto educativo in quanto consente di definire agilmente una base tecnologica per buona parte delle evoluzioni successive. In questo modo, terminata la disquisizione dell'implementazione originale, si passerà ad una carrellata delle successive. Instant-NGP, Mip-NeRF, Mip-NeRF-360, Ref-NeRF e NeRF-W sono solo alcuni degli acronimi utilizzati per definire diverse architetture per i NeRF; quelle appena citate vengono riportate in quanto sono molto vicine alla base teorica di questo lavoro che si basa fortementr sugli articoli di ricerca che le definiscono.

Terminata l'introduzione più tecnica degli strumenti citati, verrà fatto anche un passaggio presso alcuni articoli al fine di fare una breve e incompleta revisione dei più nuovi approcci, anche questi basati su quanto introdotto, per completare una consistente panoramica dello stato dell'arte e per fornire una sufficiente prospettiva di comprensione di queste tecniche.

2.1 Machine Learning e Deep Learning

Se definiamo un algoritmo come una sequenza finita di operazioni al fine di completare una serie di quesiti facenti parte tutti della stessa classe[4], allora possiamo definire un algoritmo di Machine Learning (ML) come uno che sia in grado di modificare le sue performance in risposta ad un adattamento alle condizioni in cui viene applicato[5].

La modifica del comportamento della macchina molto spesso avviene in un certo range di comportamento che viene definito a priori. Questo range limitato non pregiudica che si possano adattare sottomoduli dell'algoritmo a task differenti o che le stesse architetture possano essere trasferite su contesti simili: idee, convenzioni e strutture si muovono per tutta la base di conoscenza.

Nel contesto dell'Intelligenza Artificiale, cioè quella disciplina che studia e regola i processi informatici capaci di replicare o simulare le capacità logiche e comportamentali dell'essere umano, il ML non è tutto. La fetta più piccola della torta è presa da sistemi in grado di basarsi su basi di conoscenza da cui astrarne di nuova: spesso si tratta di sistemi basati su assunti logici del primo ordine in grado di effettuare abduzione ed inferenza di nuove regole logiche da aggiungere alla base.

Allo stesso modo, potremmo dire che la restante fetta si basa su assunti e assiomi più matematico-statistici. Nella maggior parte dei casi gli algoritmi di ML utilizzano uno o più modelli che vengono applicati per simulare o adattarsi ad un comportamento reale più o meno astratto; l'algoritmo verrà quindi definito in relazione alla risposta dei modelli agli input che è possibile fornirvi. Questo modello matematico così introdotto andrà in qualche modo adattato al task con valori e caratteristiche interne che si può assumere divergano nel caso lo stesso modello sia adattato ad un altro task similare. Questi parametri definiscono, per l'appunto, i comportamenti matematici e statistici delle feature sulle quali i modelli si basano. Feature che possono essere chiaramente evidenti e conosciute, nel caso di modelli di ML stretto, o possono essere apprese automaticamente dal modello.

In ogni caso e per la totalità di questi strumenti, il processo di apprendimento sullo spazio delle feature e quindi ciò che è richiesto per fornire all'algoritmo la capacità di inferenza della regola richiesta prende nome di training, addestramento. La totalità degli algoritmi di ML ha quindi necessità di adattare i modelli precedentemente introdotti ai dati con cui poi svolgerà il compito richiesto. Si parla quindi della necessità di riempire un gap informativo, cioè di fare in modo che questi strumenti, che inizialmente non conoscono abbastanza, siano poi in grado, in base alle informazioni che vengono apprese in fase di addestramento, di avere abbastanza conoscenza

insita da dare una risposta convincente. Risposta, output dell'algoritmo, che va strettamente legata al task in risoluzione.

Essendo strumenti a base statistica che molto spesso si basano sulla previsione di un comportamento sconosciuto, in fase di traning è necessaria la definizione di alcune metriche. Questi valori serviranno quindi poi, nella fase di valutazione della capacità operativa dello strumento, per caratterizzarne le performance. Nel caso puramente statistico a cui è votato l'utilizzo dei modelli appena definiti, le metriche si basano sulla verità della previsione. Una composizione di queste può comportare la disponibilità di un più ampio numero di parametri di valutazione con il fine di ottenere un processo di validazione più preciso.

Nel caso di questo lavoro però, gli approcci e le architetture che verranno utilizzati si occupano di effettuare sintesi di immagine. Le metriche da utilizzare saranno quindi specifiche, ma fondamentali per i processi di addestramento e valutazione.

Quanto detto finora vale nel caso in cui sia possibile determinare precedentemente alla fase di training una serie di feature chiare e facilmente definibili. Nel momento in cui questo non sia possibile o non sia conveniente, si può ricorrere a dei metodi in qualche modo fortemente aspecifici in grado non solo di riempire il gap informativo tra le caratteristiche peculiari del problema da risolvere e il task richiesto, bensì di apprendere le proprietà stesse su cui derivare l'apprendimento. Si parla quindi di feature astratte e gli strumenti in grado di discernere conoscenza in questi casi sono spesso ascrivibili a quelli di Deep Learning.

Questi algoritmi tentano di replicare il comportamento neuronale celebrale umano: ad imitare la biologia umana, l'elemento base fondamentale di queste strutture è il neurone. Replica digitale con una sua soglia di attivazione che, per convenzione, viene posto in strutture definite layer, consequenziali tra loro, che compongono reti neurali artificiali, Artificial Neural Networks. Si parla di Deep Learning quando le architetture hanno a disposizione un diverso numero di strati interni interposti tra quello di input e quello di output. Attraverso l'utilizzo di queste strutture l'informazione viene decomposta in diversi livelli di rappresentazione, dove ogni livello può corrispondere ad una serie di caratteristiche, concetti, feature: i concetti di più alto livello, che spesso coincidono con il task richiesto al modello di intelligenza artificiale, vengono definiti sulla base dei concetti di più basso livello. Possiamo quindi impropriamente assegnare a questi diversi livelli informativi uno degli strati della nostra architettura. Questo impropriamente poiché essendo strumenti che principalmente lavorano a scatola chiusa, non si ha sempre la certezza di cosa effettivamente rappresenti a livello informativo lo strato o la regione della rete. Informazione che quindi sarà definita dalle interconnessioni tra neuroni che vengono definite da un certo peso per singola connessione.

Sebbene esistano diverse tecniche complesse che sfruttano reti con un gran numero di neuroni e parametri con connessioni e strati di diversa tecnologia, e sebbene questi modelli possano essere utilizzati in una serie di diverse tecniche e approcci, quanto richiesto alla comprensione di questo lavoro risulta essere relativamente semplice.

Introduciamo quindi le reti neurali a percettrone multistrato[6][7] o Multi-Layer Perceptron. Reti che sono antesignane rispetto a tantissime delle architetture più articolate utilizzate negli ultimi anni e nella loro prima trattazione introducono tanti concetti che si sono poi trasmessi ed evoluti con la ricerca in questo ambito. Fondamentali nell'ambito di ricerca che si vuole trattare, sono alla base della totalità degli strumenti per la generazione di campi di radianza neurale. Come ne suggerisce il nome, sono architetture neurali artificiali composte da una serie di strati: il primo composto dai neuroni di input che riceveranno una serie di valori iniziali di attivazione a partire dai dati che si ha a disposizione, a seguire uno o più strati di neuroni definiti nascosti (hidden layer) ed infine uno strato di neuroni di output. Ogni singolo neurone dell'architettura sarà quindi logicamente composto da una funzione di attivazione che ne definisce lo stato in base alla sommatoria degli ingressi di attivazione che riceve e una serie di connessioni pesate in uscita. La funzione di attivazione molto spesso è una funzione non prettamente lineare: nel caso del lavoro in discussione si citerà e verrà ripresa spesso l'Unità di Rettificazione Lineare o ReLU, cioè un neurone con funzione di attivazione data soltanto dalla parte positiva dell'argomento. Altre funzioni di attivazione che verranno citate sono la funzione sigmoide logistica, la tangente iperbolica e la Leaky-ReLU, rettificatore lineare con perdita nei valori negativi.

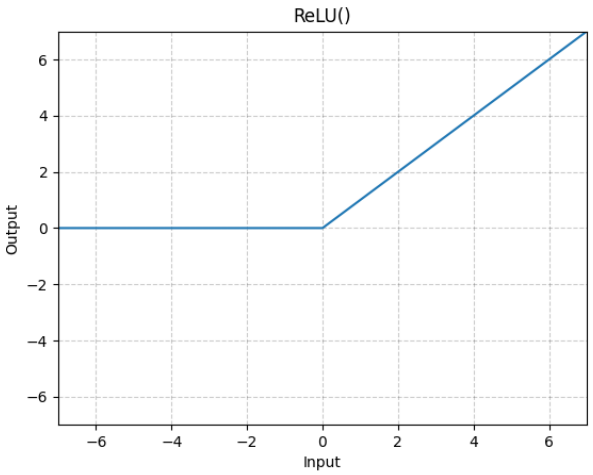

Figura 2.1: Funzione di attivazione che ammette solo valori positivi.

Altri attivatori che si renderanno utili negli esperimenti corpo di questo lavoro di tesi sono alcune variazioni rispetto a quelli già introdotti: SELU, Mish e RReLU. SELU

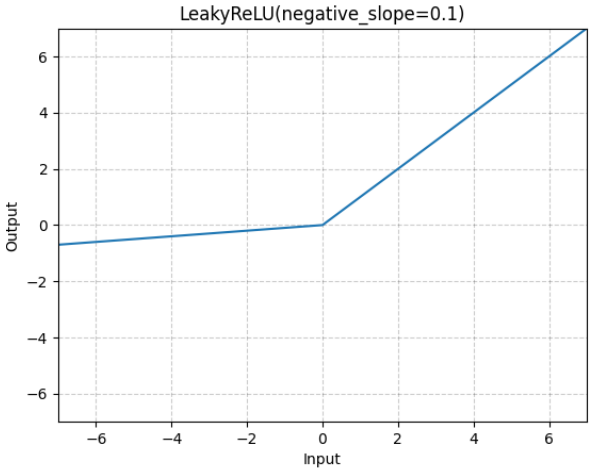

Figura 2.2: Funzione di attivazione che ammette valori positivi e ammette perdita nei negativi.

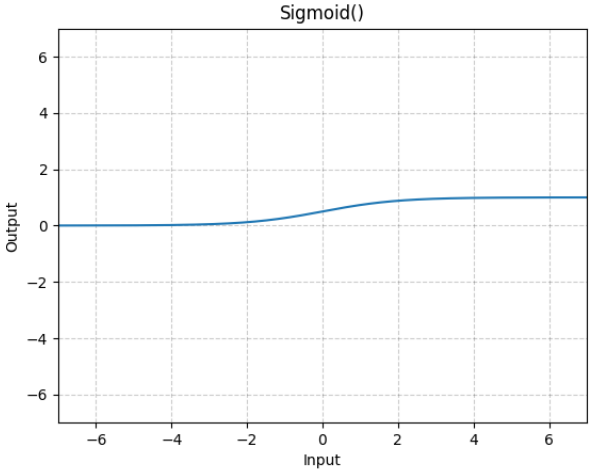

Figura 2.3: Funzione di attivazione a sigmoide logistica.

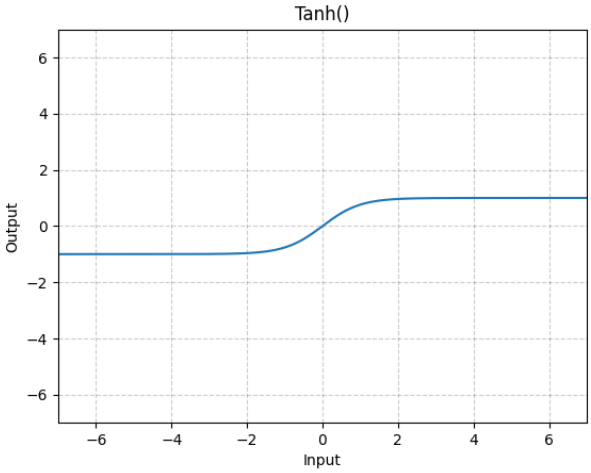

Figura 2.4: Funzione di attivazione a tangente iperbolica.

viene utilizzato in alcune architetture con l'obbiettivo di ottenere una normalizzazione delle reti in maniera automatica[8]. In maniera simile, Mish è una funzione di attivazione che ha l'obiettivo di ottenere una normalizzazione della discesa del gradiente in particolare in correlazione con i più classici ottimizzatori di rete[9]. Infine Randomized-ReLU è un attivatore che aumenta empiricamente le proprietà aleatorie della perdita di Leaky-ReLU, ottenendo a parità di architettura ottimi risultati in alcuni task di classificazione ben noti in letteratura[10].

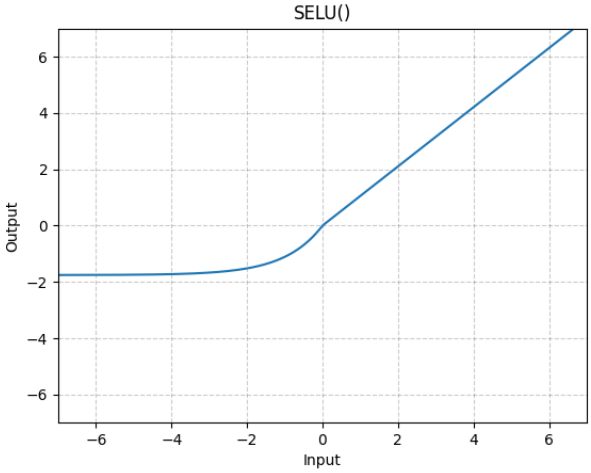

Figura 2.5: Funzione di attivazione che ha come obiettivo concorrere alla self-normalization delle reti neurali artificiali.

Verrà argomentato successivamente come la semplicità delle MLP porta alcuni vantaggi nel loro utilizzo nel caso dei NeRF ed in particolare una loro complessità teorica non elevata non ne pregiudica le capacità di rappresentazione e sintesi delle scene 3D che sono oggetto di questo lavoro di tesi.

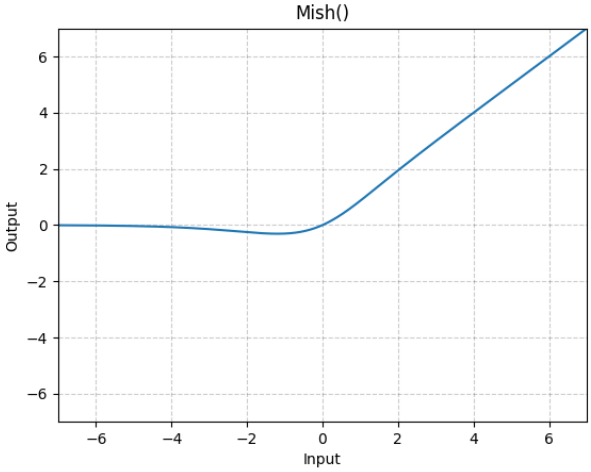

Figura 2.6: Funzione di attivazione che ha come obiettivo la normalizzazione della backpropagation in aiuto agli ottimizzatori.

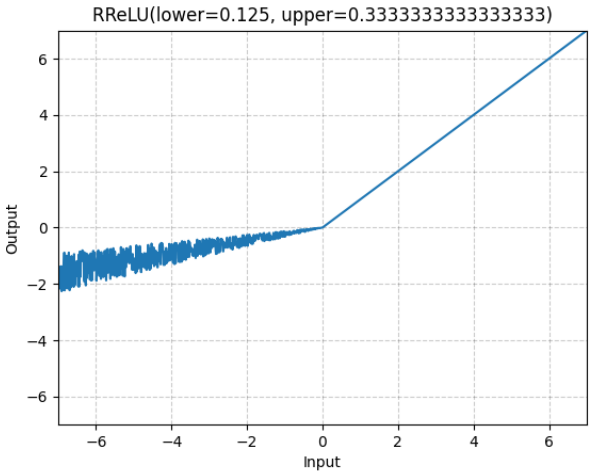

Figura 2.7: Funzione di attivazione con comportamento casuale in perdita negativa.

2.2 Computer Vision e Computer Graphics

Computer Vision e Computer Graphics, due discipline sorelle, strettamente interconnesse e protagoniste nell'informatica moderna, saranno protagoniste di questa sezione in quanto, al fine di definire un metro di paragone per gli strumenti di radianza neurale, fonderanno il corpus di questo lavoro.

Una delle definizioni più complete di Computer Vision la caratterizza come disciplina che studia l'insieme di metodi per l'acquisizione, la lavorazione e l'analisi di immagini digitali da cui può essere effettuata estrazione di dati multidimensionali al fine di produrre informazione numerica o simbolica. Materia che tocca un campo interdisciplinare, si occupa di come i calcolatori digitali possano estrarre informazione di alto livello da immagini e, in una prospettiva totalmente ingegneristica, di come si possano replicare i processi di acquisizione del sistema visivo umano[11].

La disciplina è strettamente legata anche a quella dell'intelligenza artificiale. Lo Human Visual System e la sua replicazione sono spesso stati al centro di diverse ricerche stante il grande interesse nel processo di descrizione della realtà visuale, una delle interfacce che più caratterizza la percezione umana della realtà. I primi impulsi tecnologici sono stati l'estrazione di bordi ed il riconoscimento di oggetti, quest'ultimo task fortemente legato alla definizione di Multi Layer Perceptron già citato. È riconosciuto, però, che per una maggiore comprensione dell'immagine bidimensionale fosse necessario estrarne informazione anche riguardo alle strutture tridimensionali presenti nell'immagine. Necessità che sono state raccolte dalla Computer Graphics che si qualifica quale la disciplina inclusa nell'ambito dell'informatica che si occupa di studiare metodi e tecniche di sintesi e manipolazione di informazioni visuali. La computer graphics si occupa di un campo molto ampio, esteso dalla rappresentazione digitale bidimensionale semplice alla generazione e manipolazione di contenuti multidimensionali nonché della loro rappresentazione.

Per questo lavoro di tesi ci si ritroverà a citare e descrivere moltissimi strumenti delle due materie, sia in ambito bidimensionale che tridimensionale. Parte imperante dello studio fatto risiede nella sintesi di spazi tridimensionali a partire da immagini 2D; è risultato quindi necessario introdurre come siano effettivamente rappresentabili i due concetti, in modo da avere più chiara la trattazione successiva.

Sebbene sia una piccola parte della disciplina, una delle tecniche più utili che verrà citata ed utilizzata spesso all'interno di questa trattazione è la ricerca ed individuazione di feature puntuali all'interno dell'immagine e il conseguente matching, in modo da ricostruire lo spazio di acquisizione relativo tra le due. Questa tecnica, definita come Feature Detection and Matching, si basa sullo stabilire quante più corrispondenze

tra due immagini diverse della stessa scena. Si basa quindi sull'individuazione di una serie chiara e distinta di punti chiave e la definizione di una regione nell'intorno di questi. Si passa poi all'estrazione e normalizzazione del contenuto della regione, la definizione di un descrittore che riassuma le caratteristiche di ogni singolo punto e, una volta svolto il tutto, del matching con algoritmi che generalmente sfruttano la ricerca di una distanza minima tra punti chiave e descrittori delle due immagini[12].

Questa tecnica permette, spesso con alta precisione e in presenza di un maggior numero di immagini, di ricostruire una triangolazione tra gli oggetti della scena e le posizioni di osservazione, o posa di scatto dell'immagine. Con strumenti che descriveremo in un'altra sezione, quali COLMAP[13][14] o la funzione di allineamento del software Metashape, sarà quindi possibile, a partire da un certo numero di immagini, ricostruire uno spazio delle feature tridimensionale che sarà in grado di fornirci le posizioni e angolazioni di scatto. Parametri questi fondamentali nella generazione di NeRF.

Per quanto riguarda invece la rappresentazione di scene 3D, la maggiore complessità teorica viene gestita con una maggiore complessità pratica: vanno definiti una serie di elementi base con cui è possibile generare il modello 3D all'interno di un ambiente virtuale. La scena sarà quindi composta da diversi modelli geometrici che definiscono la forma e la struttura degli oggetti all'interno di essa, da texture che vengono applicate ai modelli geometrici per definire l'aspetto visivo degli oggetti, da modelli di fonti luminose utilizzate per illuminare la scena e influenzare l'aspetto degli oggetti e dai modelli di telecamere che definiscono il punto di vista dell'osservatore virtuale nella scena.

Ai fini del contenuto di questo lavoro è necessario introdurre con più dettaglio due famiglie di modelli di illuminazione globale della scena tridimensionale che applicano la luce delle sorgenti luminose alle superfici geometriche e all'ambiente della stessa.

Il Ray-Tracing[15] si basa sull'inseguimento di raggi luminosi, considerando solo quelli che raggiungono l'osservatore, invertendone la traiettoria. È possibile associare ad ogni pixel dell'immagine tanti raggi da invertire in partenza. Questi raggi primari vengono quindi propagati nella direzione della scena, fino a che non incontrino una superfice che ne modifichi la direzione, identificando diversi raggi secondari. Viene quindi avviata una tecnica ricorsiva dove lo stesso processo di inversione viene eseguito per i raggi secondari, promuovendoli a primari. Il criterio di stop è dato dalla condizione di incontro del raggio con una sorgente luminosa, dalla quale viene estratta la luminosità emessa. A questo punto la ricorsione viene risolta e vengono calcolati i parametri dei raggi a seguire. Solitamente è possibile definire un numero di ricorsioni massime da effettuare in base al quale verrà inserita la condizione, sull'ultima ricorsione, di identificare tanti raggi secondari quante sono le sorgenti

luminose, in modo da avere completamento dell'algoritmo. La tecnica risulta essere molto potente a livello qualitativo, ma richiede mediamente più capacità di calcolo rispetto alle tecniche basate su illuminazione locale. Negli ultimi anni, la tecnica ha trovato molto utilizzo in ambiti cinematografico e videoludico portando allo sviluppo di hardware dedicati sempre più potenti. Attualmente, diversi hardware grafici hanno al loro interno architetture sviluppate proprio per facilitare il calcolo del Ray-Tracing, con engine dedicati.

L'altra tecnica altrettanto diffusa, la Radiosity[16], si basa sul calcolo della radianza, cioè il tasso di emissione energetica delle singole superfici. Il calcolo viene effettuato dividendo la scena tridimensionale in una serie di patch, piccole regioni, che vengono trattate singolarmente come sorgenti luminose. Il numero di patch può essere adattato alle esigenze della scena. Vengono calcolati i fattori di forma delle patch che portano alla risoluzione di un numero uguale alla loro quantità di equazioni lineari nello stesso numero di incognite. Sebbene sia già immaginabile come il calcolo risulti essere computazionalmente oneroso, il risultato risulta essere utilizzabile da qualunque punto di vista. Inoltre, la risoluzione di sistemi di equazioni collegate attraverso le variabili di emissione permette la possibilità di definire una certa dominanza cromatica, cioè di un effetto dato dal colore di oggetti nelle vicinanze che possono influenzare quelli delle superfici parzialmente riflettenti.

Nel caso di quest'ultimo modello, la radianza è centrale per l'argomento trattato in questo lavoro: gli strumenti che verranno introdotti a breve si occupano di ricavare con tecniche di intelligenza artificiale la radianza emessa dalla scena, definendo appunto il campo di radianza neurale soggetto della trattazione. Per farlo, si avvalgono di tecniche che sfruttano l'emissione di raggi e la determinazione del colore dato dalle superfici intercettate. Si parlerà quindi di strumenti che sono in grado di percepire e riportare direttamente i parametri di illuminazione globale della scena, sfruttando tecniche basate sulle due famiglie di modelli, affiancando il tutto alla gestione data da algoritmi che sfruttano la capacità di generalizzazione delle reti neurali. Il fatto che vengano utilizzati modelli statistici quali le reti neurali MLP per il calcolo di modelli reali che, in maniera più classica, vengono ricavati risolvendo serie di sistemi di equazioni lineari, fa rientrare il concetto nel campo del Neural Rendering definito nel capitolo introduttivo. Il tutto per avere accesso ad una nuova tecnica di rendering che cerca di rendere utili le recenti novità tecnologiche.

2.3 Neural Radiance Fields

Un NeRF (Neural Radiance Field) è un modello di grafica computerizzata che viene utilizzato per rappresentare oggetti in uno spazio tridimensionale e scene complesse[2]. È stato introdotto nel 2020 come un approccio basato su reti neurali per la sintesi di immagini fotorealistiche.

La chiave del funzionamento dello strumento è la rappresentazione volumetrica di un campo di radianza neurale. Campo che rappresenta una funzione continua nello spazio tridimensionale e associa ad ogni punto una serie di proprietà, come colore e densità volumetrica (o opacità del punto). Invece, quindi, di modellare direttamente la geometria degli oggetti, un NeRF modella la varianza e la sua variazione nello spazio.

Per crearlo è necessario un insieme di dati di addestramento che consiste in coppie di punti nello spazio 3D e i corrispondenti colori dei punti osservati dalle diverse angolazioni. Questi dati di training possono provenire da immagini di una scena reale catturate da diverse prospettive, oppure da modelli 3D sintetici. Durante la fase di addestramento, una architettura NeRF apprende i parametri di una rete neurale profonda MLP che può stimare il colore e l'opacità per qualsiasi punto dello spazio 3D. La rete prende quindi in input le coordinate spaziali di un punto e ne restituisce le proprietà nel campo di radianza. Questo processo consente di catturare le relazioni complesse tra i punti nello spazio e di generare una rappresentazione dettagliata e realistica di una scena.

Terminato l'addestramento, lo strumento può essere utilizzato anche per generare nuove immagini da punti di vista arbitrari utilizzando la rappresentazione appresa del campo di radianza. Il NeRF calcola, infatti, la radianza dei raggi che partono dall'osservatore virtuale e incontrano le superfici emittenti attraverso la scena. Questo permette la generazione di immagini sintetiche ma realistiche di oggetti reali in modo efficiente, a partire da qualsiasi punto di vista desiderato.

In questo capitolo si vedrà quindi una descrizione di alcune delle architetture NeRF più rilevanti, definendone caratteristiche e differenze. Si partirà dall'architettura capostipite per poi scendere nelle piccole e grandi rivoluzioni tecniche che hanno portato ad una sempre maggiore diffusione dello strumento. La trattazione delle singole sezioni dipende quindi fortemente dagli articoli di ricerca pubblicati a riguardo.

2.3.1 NeRF

Questa soluzione, spesso definita semplicemente NeRF o Vanilla NeRF, è riconosciuta come il primo vero e proprio utilizzo riuscito di campo di radianza neurale nella computer graphics. Prima, sebbene si fosse già tentata la rappresentazione implicita di uno spazio 3D come ottimizzazione di un'architettura neurale a partire da signed distance functions o occupancy fields. Allo stesso modo, concorrentemente erano state definite diverse architetture neurali con lo stesso obiettivo.

Di fatto, questa architettura, prima ad ottenere risultati comparabili con altri metodi di rappresentazioni di computer graphics, rappresenta la scena tridimensionale come un vettore funzionale pentadimensionale con input composto da un punto 3D, $x = (x, y, z)$ e una direzione di visualizzazione 2D, $d = (\theta, \psi)$, del quale output è dato da un colore emesso $c = (r, g, b)$ e da una densità volumetrica σ. La direzione all'interno della scena è definita comunemente con un vettore cartesiano tridimensionale unitario. La funzione di scena pentadimensionale è quindi approssimata da una rete MLP $F_\Phi : (x, d) \rightarrow (c, \sigma)$ e l'ottimizzazione dei pesi Φ mappa ad ogni coordinata di input 5D una corrispondente densità volumetrica e un colore emesso nella direzione di visualizzazione.

A livello di architettura la rappresentazione rimane consistente alla visualizzazione del campo da pose multiple incoraggiando la rete MLP a predirre la funzione di densità σ come una funzione della sola posizione x mentre viene richiesto che il colore c sia predetto a partire sia dalla posizione che dalla direzione di visualizzazione. Per ottenere questo risultato, l'architettura processa le coordinate 3D attraverso otto strati fully-connected della rete MLP con ReLU come attivatori e produce il vettore di densità σ e altri 256 vettori di feature. A concludere il passaggio per la rete, l'uscita degli otto strati viene passata per un nono strato che ha come output il colore RGB dipendente dalla direzione di visualizzazione.

I ricercatori, assieme a colleghi di studi concorrenti[17], evidenziano come però, un'architettura così costruita performi con risultati scarsi alle alte variazioni di segnale in colore e geometria. Le reti neurali profonde hanno un bias che le inclina ad apprendere meglio funzioni a bassa frequenza rispetto a quelle con più frequente variazione. Inoltre, il mappare gli input su uno spazio a più alta dimensionalità con l'utilizzo di funzioni ad alta frequenza migliora la capacità di apprendimento delle reti di queste funzioni. Viene quindi introdotto un encoder, definito encoder posizionale, simile a quanto fatto nella ricerca riguardante i transformer, altra importante architettura neurale[18]. Di fatto quindi, ogni parametro in input della funzione in ingresso F_Ψ subisce l'applicazione di una funzione $\gamma(p) = (sin(2^0\pi p), cos(2^0\pi p), \ldots, sin(2^{(L-1)}\pi p, cos(2^{(L-1)}\pi p))$. Questo encoder po-

sizionale, rappresentato dalla funzione $\gamma(.)$, viene applicato ai parametri suddetti normalizzati sia del punto tridimensionale x che della direzione di visualizzazione d. L'ordine dell'encoder, il parametro L, viene settato a 10 per la posizione, a 4 per la direzione di visualizzazione.

Al fine di massimizzare i risultati dati dalla necessità di effettuare neural rendering di una funzione di volume, viene scartata l'ipotesi di effettuare un campionamento dello spazio in N punti. Vengono perciò definite due reti separate, una definita "coarse" ed una "fine"[1] per la rappresentazione del volume. Viene quindi valutata precedentemente la funzione più sparsa e successivamente, in base al risultato ottenuto, quella più fine in modo da definire un campionamento discreto non uniforme, ad aumentare il numero di campioni in zone in cui ci si aspetta più contenuto visibile. Si considera infatti una parte significativa della scena come spazialmente vuota, della quale non è così importante apprendere i parametri.

L'architettura viene quindi utilizzata per generare un render volumetrico riportando il colore di ogni raggio passante per la scena usando tecniche già note[19]. La funzione di densità può essere interpretata come una probabilità differenziale di un raggio terminante su di una particella di dimensioni infinitesime nella posizione x. Il singolo colore sarà dato dalla funzione $C(r) = \int_{tn}^{tf} T(t)\sigma(r(t))c(r(t), d)dt$ dove $T(t) = \exp(-\int_{tn}^{t} \sigma(r(s))ds)$ dove tn e tf sono i valori massimi e minimi della sezione di spazio infinitesimo. La funzione $T(t)$ definisce quindi la probabilità che un raggio che viaggia da tn a tf non colpisca nessun'altra particella. È necessario quindi calcolare l'integrale $C(r)$ per ogni raggio emesso da ogni singola vista per la posa richiesta. Per il calcolo viene utilizzata una quadratura deterministica con un campionamento stratificato, in modo che la rete MLP possa essere valutata in posizioni continue tra loro nel corso del training.

Punto chiave della trattazione è che la funzione seguente, $\hat{C}(r) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))c_i$, dove $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j\delta_j)$, risulta essere differenziabile all'interno dello spazio tridimensionale, differenziabilità che consente una più semplice convergenza dello strumento neurale. Per come è costruita l'architettura, è possibile ottenere una rappresentazione volumetrica da una sola rete; quindi, viene ottimizzata la singola scena a partire dalle immagini e le pose corrispondenti. Ad ogni iterazione di ottimizzazione viene effettuato un campionamento random dei raggi, che poi vengono processati con il campionamento gerarchico definito sopra. Per ogni set di raggi viene utilizzata la tecnica di volume rendering già descritta al fine di ottenere una rappresentazione di quanto viene portato in input all'architettura. La loss su cui viene addestrato lo strumento è quindi data dalla somma degli errori quadrati tra i raggi renderizzati e quelli originali per ogni canale di colore, sia per la rete sparsa

[1] Traducibili come sparsa e fine.

che per quella fine. Infine, i parametri di addestramento quali il numero di raggi per batch, sono definiti in base all'hardware a disposizione, mentre si consiglia un numero di campionamenti per raggio di 64 in caso sparso e 128 nel caso fine. Per completare il cerchio riguardo l'addestramento, viene utilizzato il diffuso ottimizzatore Adam.

I dati di addestramento riportati nell'articolo di ricerca, e poi ripresi da moltissimi altri studi conseguenti, si basano su scene sintetiche realizzate con lo strumento di progettazione grafica Blender, da cui vengono estratte le camera matrices, matrici di posizionamento dell'osservatore nello spazio. Quindi, oltre alle singole immagini, che nel caso del Synthetic Dataset presentato dal paper, sono 100 per il training, 100 per validation e 100 per evaluation, all'interno di un file JSON vengono riportate coordinate di acquisizione delle viste e orientamento. Nel caso invece di scene reali, si cita già l'utilizzo dello strumento COLMAP che verranno descritte successivamente.

I risultati ottenuti vengono valutati secondo alcune metriche, quali PSNR, SSIM e LPIPS che verranno introdotte anch'esse più avanti. I ricercatori dello studio riportano in conclusione, come già citato, buonissime capacità di rappresentazione delle scene, sebbene si facciano i conti con forti drawback dati dal tempo di addestramento dello strumento. Ai fini dei risultati riportati si cita la richiesta di hardware di alto livello quali una nVidia v100 per la alta disponibilità di VRAM e di almeno 12 ore per l'addestramento, meglio se nell'ordine di un giorno. Limitazioni tecnologiche notevoli che, come si vedrà, saranno superate da altri studi che apporteranno lievi ma sostanziali modifiche.

D'altra parte, rispetto ad altre tecniche di rendering di scene tridimensionali, si hanno diversi importanti vantaggi quali la possibilità di racchiudere la scena all'interno di un minimo ingombro in memoria dato soltanto dai pesi dei neuroni della rete (5MB per questa architettura), la capacità di sintesi di immagini da pose nuove e gli evidenti passi avanti fatti, al tempo della pubblicazione della ricerca, nella capacità di rendering neurale dello strumento.

I concetti di renderer legato al volume rendering come descritto, l'encoder posizionale, l'architettura a MLP multipla, l'idea di concezione del dataset e il funzionamento di base introdotti nell'articolo di ricerca descritto vengono ripresi fortemente da moltissimi articoli basati su questa tipologia di strumenti.

2.3.2 Instant-NGP

Come già indicato nel paragrafo precedente, le primitive grafiche neurali parametrizzate da reti neurali completamente connesse possono essere costose da addestrare e valutare. I ricercatori che hanno introdotto la seguente architettura per campi di radianza neurale riescono a ridurre fortemente i tempi di addestramento con una serie di migliorie dall'alta efficacia.

Viene citata l'importanza degli encoder precedentemente introdotti come caratteristiche fondamentali delle architetture NeRF: non solo rendono possibile l'apprendimento di feature a più alta frequenza attraverso un mapping delle caratteristiche reali ad uno spazio a più alte dimensioni, ma quando sono costruite come insieme di parametri apprendibili, si fanno carico della gran parte dell'apprendimento richiesto al task. Questo fa sì che si possano utilizzare reti MLP più piccole e rapide, sebbene l'utilizzo di encoder di questo tipo potrebbe pregiudicare l'uso di una serie di ottimizzazioni hardware indicate per il training di reti neurali. Inoltre, encoder di questo tipo potrebbero basarsi su euristiche o modifiche strutturali, ancora meno semplici da gestire computazionalmente.

Viene introdotto quindi un encoder a griglia hash multirisoluzione[20] (Multiresolution Hashgrid Encoder). Questo modulo vuole essere adattivo, efficiente ed indipendente al task di utilizzo, tant'è che ne viene citata l'applicazione in contesti anche esterni alle architetture NeRF; contesto in cui comunque ci si soffermerà maggiormente. In particolare, l'utilizzo della tabella hash per l'apprendimento delle caratteristiche spaziali risolve parte dei problemi che le NeRF di Mildenhall risolvevano con una architettura doppia "coarse" e "fine": a risoluzione sparsa la tabella manterrà una mappatura 1:1 tra punti della griglia e array di entry, mentre a risoluzione più fine l'array verrà trattato come tabella hash in grado di gestire le collisioni che, portando al calcolo della media dei gradienti di addestramento, fa sì che i gradienti più rilevanti per la funzione di addestramento prevalgano sugli altri. Questa struttura quindi dà priorità maggiore alle zone con dettagli più fini rispetto alle aree più sparse, aumentando la velocità di convergenza. Si assume infatti che nelle zone sparse, più vuote, ci sia meno da apprendere che nelle zone a più alta densità informativa della scena tridimensionale. L'aumento dimensionale del positional encoder delle NeRF classiche viene mantenuto utilizzando tabelle hash legate a diverse risoluzioni dell'immagine, parametrizzando quindi la risposta dell'immagine a diverse frequenze.

Inoltre, l'utilizzo di strumenti quali le tabelle hash a complessità computazionale costante, non richiede la gestione di flusso informativo e scala bene, dal punto di vista di operazioni di basso livello, con le moderne GPU. È quindi possibile,

utilizzando l'algoritmo proposto, effettuare query alle tabelle hash di ogni risoluzione contemporaneamente.

Dal punto di vista implementativo di questi hashgrid encoders, si vuole risolvere il problema di codificare l'input di una rete neurale MLP. Tale input, che ora dipenderà da una struttura che ha parametri addestrabili, passerà attraverso una rete neurale, anch'essa con parametri che è possibile addestrare. Si ha quindi la necessità di definire un processo di addestramento che ne tenda conto. Per questo motivo, i parametri dell'encoder Θ sono disposti su L livelli, ognuno dei quali può contenere fino a T vettori di feature con dimensionalità F.

Ogni livello è indipendente dagli altri e concettualmente immagazzina i vettori di feature ai vertici di una griglia di cui la risoluzione è data precedentemente e assume una progressione geometrica tra la risoluzione più sparsa e quella più fine. Ogni vertice della griglia sarà quindi mappato ad un vettore di feature che avrà dimensione massima fissata ad al massimo T. Come indicato prima, nel caso in cui la densità richiesta sia minore di T, la mappatura tra i vertici della griglia e T seguirà un andamento 1:1, nel caso in cui questo non sia possibile, si sfrutteranno le capacità di indexing delle tabelle hash. Capacità che ovviamente richiedono l'utilizzo di una funzione di hashing, sebbene non venga garantita in questo caso la gestione delle collisioni in quanto questa viene relegata alle capacità di apprendimento della rete neurale che vi è al di sotto nella pipeline. La funzione di hashing utilizzata sfrutta concezioni spaziali effettuando uno XOR bit-wise dei valori in ingresso con grandi numeri primi[21].

Per ogni punto dell'immagine e per ogni griglia multirisoluzione, questi vengono interpolati con i valori associati ai vertici nella griglia all'interno delle tabelle hash suddette. Valori che sono quindi i parametri che verranno addestrati in fase di training. Dopodiché, i vettori risultanti dell'interpolazione tra l'input all'encoder e i parametri appena definiti vengono concatenati per realizzare finalmente l'input da passare alle reti MLP in coda al processo di encoding. A questo punto, un'altra modifica sostanziale sta nella modifica della doppia rete MLP "coarse" e "fine" con due reti più piccole fornite di soli due hidden layer di 64 neuroni, una per la resa della densità dell'immagine e l'altra per la resa del colore. La density-net si occuperà quindi di ricevere in ingresso i parametri in arrivo dall'encoder posizionale, mentre la color-net riceverà l'uscita della density concatenata all'encoding risolto per i parametri direzionali dell'immagine, al fine di preservare il colore dell'infinitesimo di spazio in base alla direzione di visualizzazione. In particolare, questa architettura riporta l'utilizzo di un encoder basato su armoniche sferiche che però descriveremo in un'architettura successiva, essendone elemento chiave per la resa qualitativa.

A questo punto, tali modifiche non solo garantiscono un lieve miglioramento delle

metriche qualitative rispetto all'architettura NeRF classica ma ne riducono il tempo di convergenza di diversi ordini di grandezza. In particolare, si cita come avvenga il passaggio su una nVidia GTX 3090 da ore di calcolo all'ottenimento della convergenza in circa 5 minuti sul Synthetic Dataset. Si nota anche come ci sia un passaggio da una scheda grafica tipicamente utilizzata per ambienti di alto profilo professionale o di ricerca come la v100 ad una scheda di pregiato livello ma disponibile anche in ambito consumer. Si riconosce infatti, dalla pubblicazione di questo articolo, un'esplosione dell'utilizzo di questi strumenti diventati grazie a questa ricerca rapidi da addestrare e con hardware richiesto semplice da reperire. Proprio per questo motivo, per questo lavoro di tesi, una delle architetture preferite per lo sviluppo di diversi esperimenti e relative pipeline è proprio quella introdotta in questa sezione, mentre le altre utilizzate ne sono forti derivazioni.

Nonostante ciò, le forti semplificazioni adottate a livello architetturale garantite dalla disponibilità dell'encoder posizionale hanno portato alla scelta di MLP più piccole che, viene citato nello studio, in caso di riflessioni all'interno delle scene, faticano ad ottenere i risultati ottenuti da altre architetture come Mip-NeRF o Ref-NeRF. Oltretutto, il problema delle riflessioni rimarrà tale in quanto ad oggi si fa fatica a rendere in maniera ottimale la densità nell'intorno delle superfici: ci aspetteremmo di poter delimitare dei bordi spazialmente densi se percorriamo un raggio dall'origine presso l'osservatore fino al raggiungimento dell'oggetto, ma in realtà le architetture NeRF tendono a rendere le superfici come risultato volumetrico di una nuvola che, presa dalla posa richiesta, ne replichi le caratteristiche. Questa difficoltà di resa delle superfici pregiudica parzialmente la capacità di ottenere diversi output, quali ad esempio le mesh che risultano difficili da poggiare sulle superfici e da chiudere.

2.3.3 Mip-NeRF

Una delle prime migliorie a cui hanno lavorato alcuni dei ricercatori ideatori dell'articolo di ricerca relativo all'architettura NeRF originale consiste in una modifica alla base dello strumento: invece di inviare raggi attraverso la scena per campionarla, si inviano tronchi di coni[22]. L'idea è riprendere uno degli approcci già utilizzati nella CG, cioè il mipmapping. Per diminuire l'aliasing, nell'approccio classico applicato ad una texture, si rendono disponibili una serie di downsample della stessa immagine e, in base alla dimensione della proiezione dell'ingombro del pixel all'interno della scena, si preleva l'appropriata immagine a più bassa risoluzione. Così facendo, si applica un filtraggio a precedere, in quanto parte del processo di antialiasing viene spostato in una fase precedente alla computazione della scena, anziché a concludere il processo, in parte richiedendo meno capacità di calcolo per la conclusione della pipeline di rendering.

Nel caso dei NeRF, la soluzione che porta Mip-NeRF (multum in parvum Neural Radiance Fields) estende le precedenti architetture nel rappresentare la radianza prefiltrata per un continuo spazio di scalature dell'immagine. L'input a questa architettura è quindi una gaussiana tridimensionale che rappresenta la regione nella quale si vuole integrare la radianza. Si può effettuare il render di un pixel prefiltrato interrogando Mip-NeRF ad intervalli lungo un tronco di cono, utilizzando gaussiane che approssimino il cono in corrispondenza del pixel. Al fine di effettuare questo processo, viene introdotto un nuovo componente, definito come Integrated Positional Encoder che, rispetto a quello definito nell'architettura NeRF originale, consente il calcolo di feature a partire da una regione di spazio, rispetto quindi ad un singolo punto. Di fatto, questo encoder si basa fortemente sulla definizione di una gaussiana multivariata che viene utilizzata per definire le feature in uscita. Caratteristiche puntiformi che non sono altro che le coordinate in uscita di un encoder posizionale NeRF distribuite secondo la campana di gauss tridimensionale appena definita.

A livello architetturale, la necessità di un'architettura a doppia rete per il calcolo del campo di radianza viene superata in quanto, l'IPE definito nel paragrafo precedente, unito all'invio di fusti di coni consente di codificare esplicitamente la variazione di risoluzione dell'immagine. È quindi possibile semplificare l'architettura, utilizzando una sola rete MLP con un maggior numero di neuroni se richiesto. In particolare, viene effettuato un sampling di dimensione costante a 128 elementi per tutto il tronco di cono, non avendo necessità di passare da un sampling più sparso ad uno più denso.

I risultati ottenuti da questa soluzione migliorano nettamente quanto fatto da NeRF. In particolare, il processo di prefiltraggio aumenta la qualità ad alta risoluzione, mantenendosi più costante ai già buoni risultati a bassa risoluzione. L'architettura

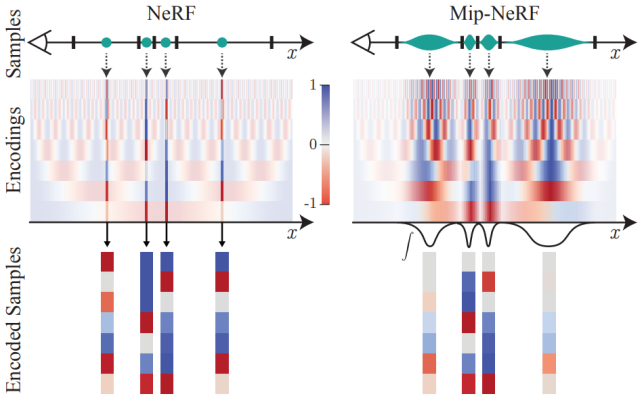

Figura 2.8: Confronto monodimensionale tra la risposta del Positional Encoder[2] e l'Integrated Positional Encoder[23]. L'utilizzo dell'IPE riduce l'aliasing integrando le feature del PE su ogni intervallo in modo che le features a dimensioni ad alta frequenza tendono ad un infinitesimo comparate alla dimensione dell'intervallo di integrazione, risultando in feature che codificano la dimensione dell'intervallo implicitamente.

leggermente semplificata risulta anche più veloce nel calcolo del campo di radianza di circa il 7%, sebbene comunque i risultati rimangano nell'ordine delle ore[22].

2.3.4 Ref-NeRF

Si è notato come buona importanza nei risultati qualitativi la abbiano anche le superfici riflettenti, poiché consentono di aumentare il realismo oggettivo della scena. Molto spesso si hanno infatti diversi livelli di riflessione da parte degli oggetti, dalle superfici leggermente meno opache fino a superfici quasi completamente riflettenti o specchi. Si è già notato come sia NeRF che Instant-NGP soffrano qualitativamente rispetto a, ad esempio, Mip-NeRF in ambito di resa della riflessione. Per questo motivo, un leggero step in avanti rispetto a quest'ultima viene dato da Ref-NeRF[23]. In particolare, un diverso approccio alla radianza e piccole modifiche architetturali consentono l'ottenimento di alcuni tra i migliori risultati qualitativi alla data di stesura di questo documento.

Ref-NeRF vuole quindi strutturare ciò che viene emesso dalle superfici come radianza in radianza entrante nella superficie, diffusione di colore, ruvidezza del materiale e tinta speculare; insieme di parametri che si presta meglio per una più fluida interpolazione della scena rispetto alla sola radianza parametrizzata rispetto alla direzione di visualizzazione. Grazie a questa riparametrizzazione, Ref-NeRF è in grado di rendere con risultati migliori gli accenni luminosi delle riflessioni e le riflessioni stesse. Questa serie di parametri ora definiti consente, in aggiunta, di modulare la scena modificando un'insieme di moltiplicatori interni all'architettura: è quindi possibile aumentare la riflessione delle superfici, modificarne la ruvidezza e la diffusione di tinta. Si effettua quindi un passaggio dalla semplice radianza emessa, ad una sua riparametrizzazione dipendente dalla direzione di visualizzazione e la normale locale alla superficie. In questo modo è possibile addestrare la rete su parametri che variano non più soltanto in base alla posa dell'immagine, ma anche in base alla riflessione dei raggi all'interno della scena. Riflessione dei raggi che però non può essere soltanto condizionata dalla direzione di visualizzazione in quanto, materiali con ruvidezza diversa rispondono diversamente all'angolo di incisione della luce.

Viene introdotta quindi una tecnica definita come Integrated Directional Encoder che utilizza, invece delle funzioni sinusoidali viste nel caso del Positional Encoder di NeRF, una serie di armoniche sferiche, complemento matematico che risulterebbe stazionario sulle superfici sferiche e quindi aumenterebbe le capacità di apprendimento delle componenti di riflessione. Inoltre, attorno al vettore principale di riflessione, si fa in modo che l'architettura a MLP sottostante possa apprendere una distribuzione di vettori definita come distribuzione di von Mises-Fisher. Distribuzione vettoriale che quindi risponderà alle armoniche sferiche e che si definirà come input ad una serie di MLP. In particolare, le coordinate spaziali verranno passate alla prima spatial-net che risponderà con una serie di parametri tra cui le normali da passare alla funzione per il calcolo della riflessione. Direzione di riflessione che insieme alla

densità servirà da input alla directional-net condivisa con Mip-NeRF che poi fornirà l'output da comporre a ciò che rimane degli output della spatial-net per generare l'output dell'architettura. Torna anche qui l'approccio a doppia rete con l'ennesima variazione rispetto a quanto definito inizialmente per le NeRF.

Sebbene l'architettura più complessa rispetto a Mip-NeRF abbia ovvi tradeoff relativi al tempo di calcolo del campo di radianza neurale, la soluzione risulta ottenere i migliori risultati qualitativi e viene spesso citata come punto di riferimento per la qualità di rappresentazione [24].

2.3.5 NeRF in the Wild

Gli approcci descritti fino a questo momento hanno dimostrato di ottenere ottimi risultati in condizioni controllate: le immagini che forniranno la scena sono catturate in un breve periodo di tempo durante il quale le condizioni di luminosità sono costanti e il contenuto della scena rimane statico. Con NeRF in the Wild[25], i ricercatori che vi hanno lavorato, si approcciano per primi in questo ambito alla concezione che il mondo reale sia geometricamente, materialmente e fotometricalmente non statico. Se ne può concludere, quindi, che la radianza non sia affatto costante.

In questo processo si tendono a separare le caratteristiche quali l'esposizione, l'illuminazione, il tempo meteorologico e il post-processing dell'immagine dalla sua geometria, in modo che l'architettura neurale possa apprendere con successo sulle sole caratteristiche morfologiche. Per farlo, le caratteristiche citate vengono apprese, per ogni immagine di input, in uno spazio latente a basse dimensioni. Dopodiché, separati gli elementi transitori da quelli statici del dataset, il modello reale viene reso dall'unione del campo di radianza neurale con un secondo campo volumetrico strettamente collegato ad un campo di incertezza. Il campo di incertezza ("uncertainty field") viene utilizzato per ridurre gli effetti degli oggetti transitori nella scena e catturare parte del rumore di osservazione. Questo processo consente quindi, avendo a disposizione una rappresentazione separata degli elementi statici della scena rispetto a quelli dinamici, di rappresentare soltanto il contenuto statico e quindi, banalmente, rimuovere oggetti indesiderati e temporanei. Utilizzando dataset fototuristici è quindi possibile rimuovere passanti, piccole impalcature o disturbi fotografici.

Ai fini di ottenere quanto descritto, viene applicato un approccio di Generative Latent Optimization[26], con il quale è possibile estrarre le rappresentazioni latenti dell'immagine: si può quindi più semplicemente predirre la geometria della scena, nonché definire uno spazio in cui possano essere considerate le condizioni di illuminazione. Sfruttando quindi moduli dell'architettura che possano rendere separatamente l'immagine statica e i transitori e assumendo che non tutte le predizioni di pixel siano ugualmente affidabili, è possibile estrarre informazioni riguardo le occlusioni o le semplici variazioni di colore tra due immagini della stessa scena. Il tutto mantenendo l'approccio a doppia MLP dove la rete a densità più sparsa sfrutta una loss che è dipendente soltanto dalla rappresentazione latente del componente del modello.

Sebbene l'architettura implementi diverse tecniche quali il multi-heading delle MLP, grazie a questo modello di architettura è possibile sfruttare dataset non strutturati di immagini: non sono più richieste condizioni costanti di cattura. Ciò apre l'utilizzo di architetture NeRF a dati ottenuti dal web, aumentando di molto le possibilità di utilizzo di questi strumenti in alcune condizioni più complesse rispetto a quelle definite

finora. In aggiunta, l'utilizzo di modelli latenti consente di poter gestire l'aspetto della scena digitale: con applicazioni di questo strumento è possibile modificare le condizioni metereologiche e di illuminazione dei render in uscita.

2.3.6 Altre architetture

Nel caso di scene tridimensionali definite unbounded, cioè quelle in cui la camera virtuale dell'osservatore può essere orientata in diverse direzioni e l'oggetto rappresentato può esistere ad ogni distanza dall'osservatore, gli approcci definiti precedentemente tendono a faticare nella resa dell'interezza della scena. In ampie scene come queste l'utilizzo di modelli come quelli già descritti riporta tre problematiche: la parametrizzazione, cioè definire uno spazio opportuno che possa contenere grandi scene; l'efficienza, la capacità di rendere opportunamente scene più grandi senza aumentare di troppo la dimensione della rete MLP alla base; l'ambiguità di rappresentazione, il contenuto di una scena unbounded può stare a diverse distanze ed essere osservato da un piccolo numero di raggi, rendendone più difficile la ricostruzione. Mip-NeRF-360[27] sfrutta una contrazione dello spazio opportuna per contrarre efficacemente lo spazio, la reintroduzione di una doppia MLP per una miglior gestione della qualità e quindi dell'efficienza e per risolvere l'ambiguità e la nuvolosità di alcune superfici introduce un regolarizzatore di densità. Come architettura richiede più capacità di calcolo per la computazione del campo di radianza rispetto agli approcci basati su Instant-NGP, ma viene definita come uno degli approcci più capaci qualitativamente nel caso di scene molto grandi o unbuonded.

Per quanto riguarda invece una platea di strumenti che possano essere definiti particolarmente interessanti, si ha piacere nel citarne alcuni: TetraNeRF[28] riesce a ricostruire un campo di radianza a partire da input definiti da point cloud, il ché è interessante vista l'inversione del processo classico; Instruct-NeRF2NeRF[29] sfrutta modelli di Latent Diffusion per modificare la scena o aggiungere dettagli inesistenti al momento delle acquisizioni attraverso un prompt testuale; K-Planes[30] si occupa di unificare una serie di intuizione per la rappresentazione latente di immagini e la deformazione spaziale del campo nel caso del movimento per NeRF dinamici; LERF[31] implementa campi di radianza nei quali vengono incorporate informazioni testuali per la descrizione degli oggetti riportati. Questi esempi a sottolineare quanto sia diverso ed in espansione il campo della ricerca in questi argomenti.

Capitolo 3

Sintesi di Neural Radiance Fields

In questo capitolo si vedranno metodi, strumenti e tecniche pratiche utilizzate per la sintesi dei campi di radianza neurale attraverso studi operativi ed esperimenti. Ai fini della discussione del metodo, verranno introdotte diverse risorse web, tool e piattaforme che hanno permesso questa parte: vista la complessità dell'argomento, non è stato ritenuto necessario reimplementare gli strumenti richiesti, bensì si è scelto di sfruttare codice già redatto o porting degli strumenti originali. In particolare, per la parte riguardante gli esperimenti sulla pipeline NeRF legati ad un approccio generico, si è scelto di utilizzare un porting di Instant-NGP per il tool PyTorch[1] in modo da sfruttare le caratteristiche di rapidità di generazione del campo della specifica architettura, rendendone possibile un utilizzo più immediato rispetto al codice originale. Nella fase di esperimenti seguente, verrà invece riportato uno strumento, attualmente allo stato dell'arte per quanto riguarda l'insieme di risorse disponibili all'interno della stessa piattaforma, che può essere utilizzato per la sintesi di campi di radianza a partire da diverse architetture neurali e diversi processi.

Introdotte le piattaforme software utilizzate, è necessario descrivere le metodologie con cui verranno valutati i risultati qualitativi ottenuti in termini di resa della scena tridimensionale. Verranno introdotte una serie di metriche, basate sia sull'analisi del segnale che sulla similarità soggettiva, che verranno sfruttate per dare un'idea di confronto qualitativo tra le scene.

Al termine della discussione sulle metodologie, verranno riportate delle descrizioni riguardo i dati utilizzati per il testing e la generazione di scene. Se per i test iniziali riguardo i moduli della pipeline di generazione verranno utilizzati dataset standardizzati, nella sezione riguardante l'applicazione degli strumenti in esame al patrimonio culturale descriveremo anche le metodologie per le quali, a partire da semplici immagini o video, è possibile elaborare un dataset adeguato.

[1] Framework per lo sviluppo di software legato a Machine Learning e Deep Learning disponibile presso: https://pytorch.org/.

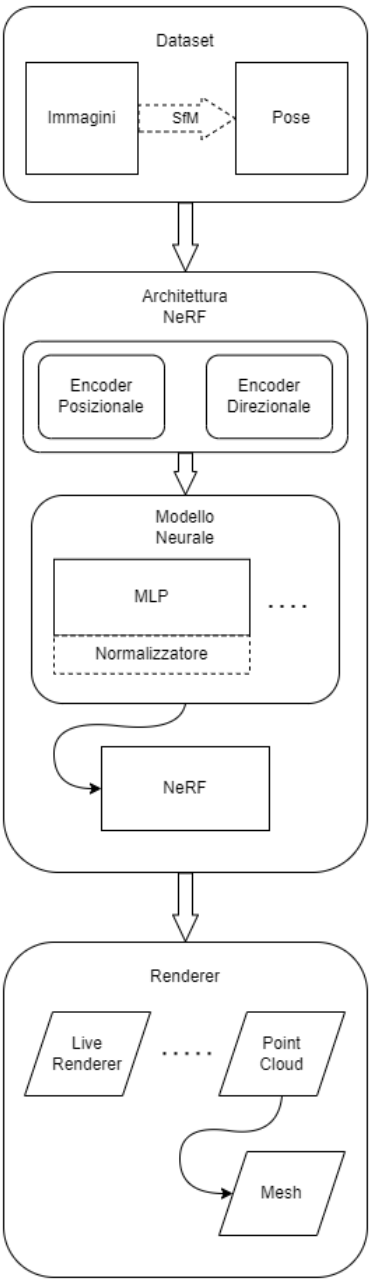

Figura 3.1: Pipeline del processo NeRF. Il processo parte da un dataset di immagini che può già includere le pose di scatto; se queste non sono presenti vengono utilizzati strumenti di Structure from Motion per ricavarle. I dati vengono trasmessi all'architettura NeRF che li processa con uno strato di encoder. Il modello neurale, che può contenere un diverso numero di MLP, prende in ingresso l'output degli encoder e, quando presente, utilizza una funzione di normalizzazione per l'accumulo della densità. Attraverso il processo di training viene generato un campo NeRF. Questo può essere visualizzato attraverso tecniche di rendering, da un Live Renderer volumetrico all'accumulo dell'output in profondità in point cloud. Da questa è possibile utilizzare tecniche di generazione di mesh.

3.1 Software utilizzato

Ai fini di prendervi dimestichezza, valutare le capacità della soluzione e fondare un'adeguata base pratica di conoscenza per lo sviluppo della tesi che segue, si è scelto di utilizzare un porting dell'implementazione originale di Instant-NGP come prima base di partenza. Nella comunità open source e di ricerca legata all'utilizzo di queste soluzioni, quella che è sembrata migliore sia dal punto di vista di risultati citati, che per la trasparenza di progettazione e la fedeltà alla soluzione originale è la soluzione descritta come torch-ngp[32]. Il repository contiene il codice per l'utilizzo di un renderer volumetrico, un'implementazione in Python di Instant-NGP, una GUI per la gestione, la visualizzazione e l'esplorazione del risultato, nonché codice relativo ad altre implementazioni che non sono state utilizzate e non verranno citate. Il codice messo a disposizione nella sua implementazione raggiunge, nell'utilizzo delle sue capacità massime, il 99% dei risultati qualitativi legati all'analisi del segnale di immagine dell'implementazione originale. il software risulta inoltre completare la generazione del campo di radianza, a parità di numero di iterazioni, in un tempo che è compatibile con quello citato nell'articolo legato a Instant-NGP[20], ma diverso in quanto su hardware differente.

Per una breve descrizione di cosa fa il codice, il repository mette a disposizione un renderer interrogabile con delle posizioni spaziali e vettori direzionali attraverso una GUI che, a suo tempo, è in grado di interrogare l'architettura neurale per la generazione dell'immagine a partire dal campo di radianza. Architettura che è possibile addestrare con delle chiamate opportune ad un trainer che a sua volta è in grado di effettuare il data-gathering del dataset fornito in ingresso. È quindi possibile visualizzare il processo di addestramento esplicitamente, visualizzando la risposta delle reti neurali all'interno della stessa scena 3D che muta con l'apprendimento dei pesi, ad un piccolo costo computazionale. L'output del processo è quindi un file contenente i pesi dell'architettura neurale completamente addestrata. A richiesta, si può richiedere l'output del campo in formato di point cloud ed è possibile applicare la tecnica dei marching cubes[33] per ottenere in output una resa della mesh dell'oggetto. L'implementazione quasi interamente in python del tool rende molto immediata la modifica del processo, sia a livello metodologico che architetturale. Facilità di modifica che non si aveva nel caso dell'implementazione originale in linguaggio CUDA/C++ che avrebbe richiesto una più minuziosa attenzione ai componenti di più basso livello, che in python è stata relegata alle opportune librerie.

Lo strumento seguente, di notevole importanza in quanto ha come attivi contributori molti dei ricercatori legati al mondo delle NeRF e che in parte ne sono ideatori, è il framework Nerfstudio[34]. Questo, completamente a sorgente aperta, consente l'utilizzo di alcune tra le più importanti architetture NeRF, nonché tutte quelle citate

in questo lavoro ed inoltre rende disponibile tutta una serie di tool chiave per il preprocessing e il postprocessing dei dati. In particolare, attraverso implementazioni di diversi algoritmi di Structure from Motion, consente la generazione di dataset in formato utilizzabile a partire da collezioni fotografiche o filmati; attraverso strumenti legati alla visualizzazione di scene 3D e rendering, consente la generazione di filmati composti da viste sintetiche per una più semplice e meno onerosa valutazione soggettiva delle scene più grandi e complesse; infine, essendo uno strumento strettamente legato alla ricerca di settore, mette a disposizione strumenti efficaci per la generazione e valutazione oggettiva di parametri di qualità delle scene. Questa serie di motivazioni lo rende uno strumento più che indicato nel suo utilizzo. Replicando quindi, per ogni architettura, quanto è in grado di fare lo strumento precedente, tra le tante mette a disposizione NeRF originale, Instant-NGP, Mip-NeRF, NeRF-W ed una implementazione propria, anch'essa a sorgente aperta, ma in costante divenire, Nerfacto.

Al momento della stesura di questo documento, Nerfacto combina diverse tecniche all'avanguardia all'interno di un singolo modello, quali: rifinitura della posa della camera, condizionamento dell'aspetto per ogni immagine, campionamento propositivo (simile a quello NeRF basato su densità di dettaglio), contrazione della scena, hashgrid-encoder. Tra le varie tecniche, la pipeline propaga l'algoritmo di apprendimento fino alla direzione di acquisizione della camera nel dataset, modificando leggermente gli angoli delle pose in fase di training in caso di necessità, al fine di aumentare la resa qualitativa della scena. Il che rende l'architettura particolarmente indicata nel caso dell'acquisizione attraverso metodi a basso costo (foto con smartphone, droni, camere video, etc.). Il campo così generato in Figura 3.2 sfrutta, similmente a Instant-NGP, un encoder posizionale a griglia hash multirisoluzione e un encoder direzionale basato su armoniche sferiche. La risposta viene fornita con due MLP, una che si occupa di fornire l'uscità della scena in densità, l'altra in colore in formato RGB.

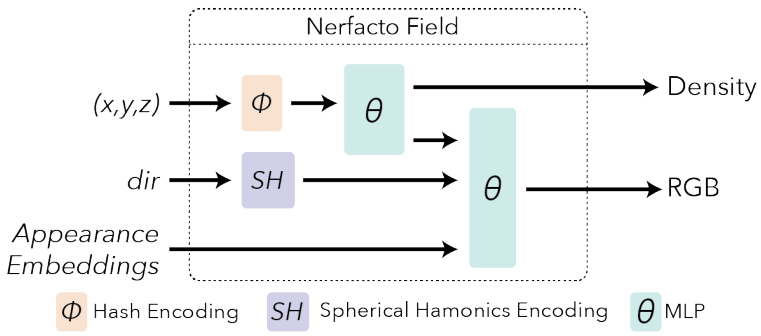

Figura 3.2: Composizione dell'architettura per la generazione ed interrogazione di un campo di radianza neurale generato con l'architettura Nerfacto.

A corredo dei test, come elementi software aggiuntivi, è stato utilizzato codice

COLMAP[13], per l'estrazione delle pose da dataset reali assieme, per lo stesso motivo, alla suite software Agisoft Metashape e codice per la generazione di mesh secondo l'algoritmo dei Deep-Marching Tetrahedra[35]. Per l'estrazione dei dati di volo dei droni che sono stati utilizzati nell'ambito degli esperimenti riguardanti il patrimonio culturale, è stata utilizzata la suite exiftool[2].

[2]Tool per l'estrazione di dati a partire dai campi EXIF delle immagini digitali, disponibile presso: https://exiftool.org/.

3.2 Dataset

I primi esperimenti sono realizzati con una serie di dataset già conosciuti in letteratura ed utilizzati da diversi articoli di ricerca riguardo altrettante architetture. Il primo, che contribuisce all'iconografia del camioncino giocattolo simbolo di Nerfstudio e delle architetture NeRF in genere, è stato introdotto dal paper NeRF originale e prende nome di Synthetic renderings of objects[2], abbreviato in Synthetic Dataset o, essendo realizzato con il software di progettazione grafica Blender, in Blender Dataset. Esso è composto da immagini di 8 oggetti che riportano geometrie complesse e materiali non-Lambertiani[3]. Sei di questi oggetti sono renderizzati in viste prese nell'emisfero superiore, due di questi su di una sfera completa. Per ogni oggetto sono disponibili 100 viste per il training, 100 per l'evaluation del modello, 100 per il testing, tutte ad una dimensione di 800x800 pixel.

Figura 3.3: Esempio di alcune immagini facenti parte della scena Lego del dataset Synthetic.

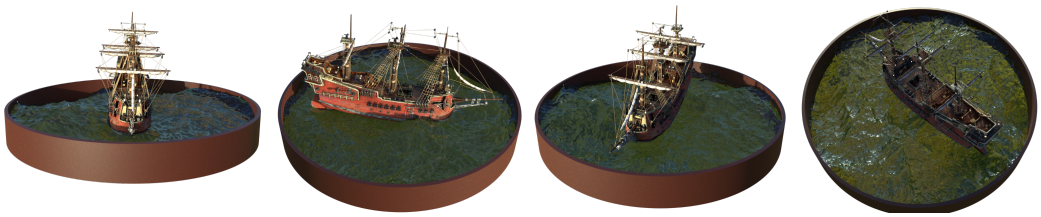

Figura 3.4: Esempio di alcune immagini facenti parte della scena Ship del dataset Synthetic.

Per verificare le capacità di sintesi delle architetture in caso di immagini reali scontornate vengono utilizzate alcune scene dal dataset Tanks&Temples[36], in particolare le scene predisposte per il training: Barn, Caterpillar, Church, Courthouse, Ignatius, Meeting Room, Truck. Nel caso di questo dataset, non c'è consistenza nel numero di immagini di training, evaluation e test, sebbene le immagini siano sempre di un numero minimo nell'ordine delle diverse decine fino ad un numero massimo vicino alle tre centinaia. Le scene citate, oltre a riportare come oggetti principali

[3]Materiali che non riportano riflessione lambertiana, quindi non hanno una riflessione diffusa distribuita equamente nello spazio.

diverse strutture, in alcuni casi quali Meeting Room, risultano utili per testare le capacità di resa di scene con zone ad alta riflessione, come superfici a specchio o vetro, punto chiave delle architetture per campi di radianza. Le immagini facenti parte del dataset non sempre riportano l'oggetto principale esattamente nel centro.

Figura 3.5: Esempio di alcune immagini facenti parte della scena Truck del dataset Tanks&Temples.

Per valutare la resa delle architetture testate nel caso di un numero minore di pose a disposizione è stato utilizzato il dataset legato alla ricerca nel campo del Local light Field Fusion[37], LLFF Dataset. Queste scene, oltre a rappresentare una sfida al calcolo dato da poche informazioni, riporta anche oggetti del mondo vegetale, con foglie e piccoli dettagli che possono contribuire al test delle capacità di antialiasing degli strumenti generativi.

Figura 3.6: Esempio di alcune immagini facenti parte della scena Room, a sinistra, e della scena Fern, a destra, del dataset LLFF.

I dataset riportati finora sono ottimi nel caso di test preliminari alle architetture, il fatto che però siano composti da immagini acquisite in condizioni strettamente controllate li rende meno simili a quanto possa essere ottenibile in condizioni più libere, vicine all'acquisizione diretta dal mondo reale. Inoltre, la grandissima parte delle scene tridimensionali che ne risultano sono scene strettamente bounded, cioè, legate alla rappresentazione del singolo oggetto, che contengono poche informazioni per l'esplorazione. Per questo motivo, in letteratura vengono citati diversi algoritmi per la ricerca delle coordinate di acquisizione delle immagini. COLMAP[13] e Hierarchical Localization[38] sono solo alcuni degli strumenti open source disponibili per questo fine, mentre per quanto riguarda software proprietario, uno strumento importante e già citato, è Agisoft Metashape[4]. Nel task di Structure From Motion, gli strumenti sono in grado di ricostruire angolazione e posizione delle camere virtuali a partire da un processo di triangolazione inversa delle feature estraibili dalle immagini. Se poi le pose vengono convertite in matrici di trasformazione delle camere opportune,

[4]Strumento software per la gestione di scene di immagini e fotogrammetria disponibile presso: https://www.agisoft.com/.

è possibile fornire queste informazioni alle architetture NeRF in modo che queste siano capaci di ricostruire la scena 3D con le metodologie già citate.

Con quanto appena descritto, nel caso di prove degli algoritmi nell'ambito del Cultural Heritage, i software di SfM sono stati elementi chiave per la ricostruzione di informazioni che, al momento dell'acquisizione delle immagini e video, non era stato previsto avrebbero dovuto essere annotate. In particolare, sono state utilizzate le implementazioni interne a Nerfstudio di COLMAP e il software Metashape. Sono quindi state ricostruite le camera matrix di immagini acquisite in tre siti culturalmente rilevanti: di quelle estrapolate da un video del santuario di Macereto, complesso religioso posto a circa 1000 metri s.l.m. nel versante occidentale dei Monti Sibillini a Visso; quelle estrapolate da un video di alcuni ruderi del castello di Magalotti nel comune di Fiastra; infine, sempre nella zona dei Monti Sibillini, ad una serie di scatti fotografici della statua al centro di Piazza A. Gentili a San Ginesio. Nel caso di video, si parla di filmati in risoluzione 4k a 30fps, di cui vengono opportunamente estratti solo i frame chiave, per raggiungere un numero di immagini che si aggiri tra le 250 e le 350. Nel caso della galleria fotografica di San Ginesio, invece, sono raccolti circa 140 frame. Di tutte queste, si lascerà la premura agli script di Nerfstudio dell'effettuare uno split 90-10 per separare immagini di training da quelle per la valutazione delle capacità dei modelli.

Figura 3.7: Esempio di alcuni frame estratti dal video della scena ambientata al santuario di Macereto.

Figura 3.8: Esempio di alcuni frame estratti dal video della scena ambientata al castello Magalotti.

Figura 3.9: Esempio di alcune immagini facenti parte della collezione fotografica della scena ambientata in piazza A. Gentili a San Ginesio.

3.3 Metriche

Per valutare i risultati ottenuti in ogni prova, è necessario riportare diverse metriche che, nel caso complesso di valutazioni delle immagini, siano in grado di dare un'idea dei risultati complessivi della resa di sintesi. Partendo dalla più semplice analisi del segnale video, si utilizza una metrica legata alle caratteristiche segnale-rumore del confronto tra due immagini: Peak Signal-to-Noise Ratio (PSNR) è una metrica molto diffusa per quantificare la massima potenza di un segnale contrapposta alla massima potenza di rumore che può pregiudicare la fedeltà di rappresentazione. Questa metrica si basa sul calcolo dell'errore quadratico medio per ogni punto della differenza tra le due immagini in input, riportata in scala logaritmica data dal valore massimo tra i pixel.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

$$PSNR = 20 \log_{10}(MAX_I) - 10 \log_{10}(MSE)$$

dove (m, n) sono le dimensioni di una immagine I priva di rumore, K la sua approssimazione con rumore e MAX_I il massimo valore legato a un pixel dell'immagine.

Se PSNR misura la distanza a livello di segnale tra due immagini, la Structural Similarity Index Measure (SSIM)[39] è un metodo per la predizione della qualità percepita di un'immagine digitale o cinematica, risultando questa nella misura della similarità tra due immagini. Viene calcolata su diverse patch dell'immagine di dimensione NxN, x e y:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c1)(\sigma_x^2 + \sigma_y^2 + c2)}$$

dove μ_x è il valor medio dei pixel di x, μ_y è il valor medio dei pixel di y, σ_x^2 e σ_y^2 le varianze, σ_{xy} la covarianza, $c_1 = (k_1 L)^2$ e $c_2 = (k_2 L)^2$ due variabili per la stabilizzazione della divisione al denominatore dove L è il range dinamico dei valori dei pixel $2^{bitsperpixel} - 1$ e $k_1 = 0.01$, $k_2 = 0.03$.

L'ultima metrica che viene riportata è invece data dalla capacità di alcune reti deep discriminanti di rappresentare feature come metrica di percezione umana. Quello che fanno Zhang et al.[40] per la metrica LPIPS è utilizzare alcuni strati di una rete neurale VGG[41] dalle caratteristiche fortemente convoluzionali, già addestrata in task generativi e discriminativi in architetture GAN o simil GAN, come estrattore di feature: per calcolare una distanza d_0 tra due patch x, x_0, data una rete F, vengono calcolati gli embedding deep dei due patch, normalizzate le attivazioni dei neuroni della rete per la dimensione dei canali, viene scalato ogni canale per un vettore

di scalatura w e viene estratta la distanza l_2. A questo punto, si media per la dimensione spaziale e su tutti gli strati della rete. Da qui, una piccola rete neurale G viene addestrata per predirre il giudizio percettivo h dalle coppie di distanze (d_0, d_1). Questa metrica risulta più capace nella resa di caratteristiche soggettive dell'HVS date dall'interpretazione del segnale visivo da parte dell'essere umano. In particolare, in condizioni dove le metriche PSNR e SSIM faticano nella resa di una distanza oggettiva, quali tra un'immagine e un semplice blur della stessa, la capacità discriminativa degli strati della rete VGG utilizzata risulta efficace. Verrà quindi considerata come una metrica più vicina alle caratteristiche definite come soggettive di un osservatore umano, non digitale.

Capitolo 4

Esperimenti e Risultati

In ottica di quanto già descritto nel capitolo precedente, a seguire a questa introduzione, verrà riportata una descrizione degli esperimenti mentre di seguito saranno indicati i risultati ottenuti. Alla descrizione di ogni esperimento corrisponderà la presentazione dei risultati ottenuti. Una discussione di quanto riportato sarà corredata, per ogni singolo caso, da tabelle per la definizione dei risultati numerici codificati nelle varie metriche introdotte e, quando possibile, da immagini per mostrare parte dell'output ottenuto in forma di campi di radianza neurale. Come serie di esperimenti si è scelto di effettuarne su ogni sezione del processo NeRF: a partire da come è possibile generare i dati da fornire agli strumenti, a come e con quale processo si possono ottenere output in forma di mesh. Verranno quindi introdotte delle sottosezioni a definire quanto detto.

4.1 Esperimenti sugli encoder

4.1.1 Descrizione

Uno degli elementi che ha sicuramente rivoluzionato l'ambito dei campi NeRF è l'introduzione da parte di Muller et al. del multi-hashgrid encoder in Instant-NGP. Il fatto che abbia a disposizione una serie di parametri addestrabili esterni alla MLP e che questi possano essere modificati in maniera più efficiente rispetto alla richiesta di convergenza della sola rete rende il processo di addestramento diversi ordini di grandezza più veloce. Oltre a questo, la capacità di risolvere in parte il problema di un efficiente campionamento dei raggi all'interno della scena semplifica di molto l'impianto a doppia rete MLP introdotto da NeRF originale. Con questo in mente, tenendo quindi conto dei notevolissimi miglioramenti che l'utilizzo di questo encoder introduce e data la complessità pratica di questo algoritmo, si è scelto, in questa serie di test, di non modificarlo. Allo stesso tempo, ispirati dalla qualità

raggiunta parallelamente da Ref-NeRF con un encoder molto più vicino a livello pratico al Positional Encoder di NeRF originale, ci si è chiesto quanto effettivamente fosse efficiente questa scelta. In particolare, ci si chiede quanto l'apprendimento di parametri che sono suscettibili al training possa incidere sulla resa della scena.

Gli esperimenti che verranno introdotti in questa sezione si basano quindi sulla modifica dell'encoder direzionale, in verde, che coincide con Ref-NeRF nel caso di Instant-NGP: l'encoder basato su Spherical Harmonics. In particolare, viste le evidenti differenze tra le architetture dei due approcci, ci si chiede quanto questo incida nei campi di radianza neurale generati con Instant-NGP. Si è scelto quindi di lavorare con torch-ngp, software nel quale la modifica agli encoder è accessibile e si basa sulla semplice modifica del codice Python relativo.

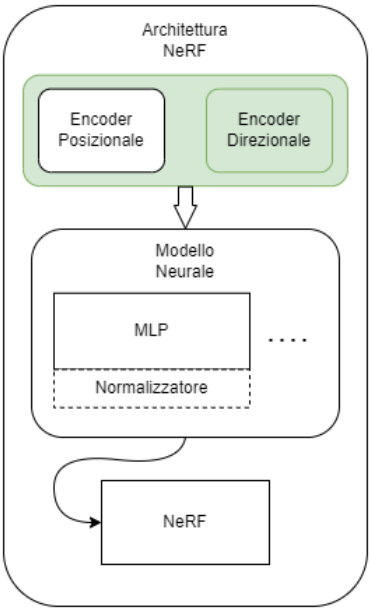

Figura 4.1: Diagramma di azione degli esperimenti sugli encoder. La modifica viene fatta all'implementazione dell'encoder direzionale. Questo coinvolge, a cascata, i parametri delle reti MLP. L'esperimento viene valutato sul campo NeRF generato.

Si è scelto quindi di generare una serie di tre test da mettere poi a confronto. Il primo si basa sull'addestramento e valutazione di diversi campi di radianza neurale con l'architettura Instant-NGP non modificata. Il secondo test introduce un encoder definibile come "Dummy": si è scelto di introdurre un nuovo encoder, relativamente poco complesso, che ci si auspicava ottenesse risultati decisamente peggiori rispetto a quello di default. Questo encoder, definibile semplicemente come encoder moltiplicativo, moltiplica gli angoli di yaw, pitch e roll per dei parametri che vengono appresi, e quindi modificati durante l'addestramento. Ciò vorrebbe risultare in una modifica degli angoli di visualizzazione e ci si auspicava portasse ad una deformazione della

scena, in quanto, sebbene le capacità di generalizzazione della rete MLP possano essere in grado di compensarla, i tre diversi parametri di scala hanno possibilità di divergere e quindi deformare il campo spazialmente. La semplice moltiplicazione del vettore $d = [\rho, \theta, \psi]$ per il vettore $k = [k_\rho, k_\theta, k_\psi]$ avviene durante il passo forward dell'encoder, mentre, essendo definiti i valori di k come parametri addestrabili esterni, a mo' dell'hashgrid encoder, nel passo backward della backpropagation è il motore python ad occuparsi dell'addestramento e modifica dei parametri. L'ultimo test, invece, applica una funzione passthrough all'encoder direzionale: l'architettura neurale viene addestrata con direttamente in input gli angoli di acquisizione delle pose in radianti.

Per avere un'idea della resa dei diversi algoritmi a diverse condizioni di stress, il test viene svolto sulle scene del Synthetic Dataset con l'aggiunta di alcune altre prese da Tanks&Temples e LLFF Dataset. Le scene da testare saranno così: Lego, Chair, Drums, Ficus, Hotdog, Materials, Mic e Ship da Synthetic; Barn, Caterpillar, Family e Truck da Tanks&Temples; Fern e Room da LLFF. Si porrà particolarmente l'accento sul fatto che l'encoder basato su armoniche sferiche viene introdotto proprio per aumentare la resa della scena in presenza di riflessioni e superfici non opache: Materials, Ship e Room sono le scene con più evidenti superfici riflettenti. La scena Materials è composta da una serie di superfici quasi sferiche di diversi materiali, alcuni di questi con alta capacità di riflessione. La scena Ship riporta un modellino di nave a vela che naviga in un piccolo specchio d'acqua leggermente mossa quindi sono presenti diversi riflessi sulla superficie del liquido. Infine, la scena Room ha tra gli elementi principali che la compongono uno schermo spento posto al centro della stanza che riflette il contenuto di quanto è nella direzione opposta all'osservatore.

Tutte le scene vengono valutate dopo 30'000 step di addestramento, ogni step coincidente con il passaggio per l'architettura del contenuto informativo di una immagine; passaggio dell'intero dataset della scena che sarà quindi ripetuto più volte in base al numero di immagini presenti. L'architettura viene quindi settata per il raggiungimento del risultato massimo, quindi con mappatura degli errori in fase di realizzazione, tensori a 32bit, utilizzo dell'architettura CUDA anche per il casting dei raggi nella scena e il precaricamento in GPU delle immagini.

I risultati numerici ottenuti sono riportati in Tabella 4.1. La generazione di campi di radianza è stata svolta su di una macchina con processore Intel i7-9750H e nVidia RTX 2060 che supporta una rolling release basata su Arch aggiornata a Maggio 2023 con software richiesto aggiornato nello stesso periodo. Ogni scena richiede un tempo variabile per l'addestramento che va dai 20 ai 40 minuti, in base alla dimensione in pixel delle immagini.

4.1.2 Risultati

Facendo riferimento alla Tab.4.1, una delle prime osservazioni che è possibile fare è relativa ai risultati ottenuti in base alla metrica. Si ricorda che, nel caso della metrica PSNR ciò che viene valutato è strettamente legato ad una corrispondenza fisica tra il segnale di due immagini: quella originale utilizzata come valutatore e un render della scena ottenuto mantenendo la stessa posa della precedente. Si ha quindi una metrica che misura la correlazione fisica tra i due segnali individuando le differenze come rumore. La metrica assume quindi valore migliore tanto più ne è grande il valore espresso, valore peggiore se più basso. D'altra parte, la metrica LPIPS si comporta in maniera opposta: tanto più è basso il valore migliore ne sarà il risultato riportato e viceversa. Questa misura però, attraverso feature legate all'apprendimento profondo, la verosimiglianza di un render della scena rispetto alla sua controparte di valutazione ad occhi umani.

Ciò detto, è possibile notare come per molte delle scene tridimensionali che hanno come soggetto principale degli oggetti con materiali prettamente opachi, i tre encoder introdotti ottengano valori di PSNR tutto sommato strettamente paragonabili. In particolare, l'encoder Dummy risulta equivalente o leggermente migliore dell'encoder basato su armoniche sferiche in tutte quelle scene in cui vi sono poche riflessioni, pochi dettagli o dettagli distinguibili ad una risoluzione elevata: è il caso, ad esempio, della scena Chair in cui elemento principale lo fa il tessuto della sedia che presenta trame riconoscibili soltanto a definizioni particolarmente elevate e non vi sono riflessioni. Perde performance, però, nel caso di oggetti reali o se sono presenti elevati dettagli: nel caso di Mic, dove oggetto principale della scena è un microfono da registrazione, la caratteristica texture metallica del microfono non viene resa in maniera impeccabile. Nel caso di riflessioni, ad esempio per Room, i risultati sono molto inferiori. Quanto appena descritto è visualizzabile nella serie in Fig.4.2. Sono qui evidenti le maggiori capacità di rappresentazione dello Spherical Harmonics, che ha proprio come scopo principale quello di essere il più performante possibile in presenza di superfici riflettenti. La scena materials, in qualche modo, risulta essere un outlier rispetto a quanto appena definito: l'encoder dummy, in questa iconica prova frequentemente utilizzata in letteratura per dare idea delle capacità di resa delle riflessioni, sebbene non abbia fondamenti teorici adeguati quanto quello utilizzato dalle architetture Ref-NeRF e Instant-NGP, risulta essere migliore sia per quanto riguarda il rapporto segnale rumore, sia per quello che concerne le proprietà soggettive umane. È probabile che la risposta al quesito in cui ci si chiede quali siano le motivazioni risieda in approcci di deformazione che meglio riportano gli oggetti contenuti nella scena e la resa che questo encoder con pochissime basi teoriche riesce a ottenere degli oggetti con meno capacità riflessive. Ne concludiamo quindi che l'encoder che era stato introdotto per fornire maggiore rumore e peggiorare

Scena	Spherical Harmonics		Dummy		Passthrough	
	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS
chair	33.83	0.013	33.90	0.013	33.91	0.013
drums	25.46	0.069	25.42	0.070	25.45	0.069
ficus	32.34	0.020	32.38	0.020	32.53	0.019
hotdog	34.53	0.034	34.49	0.036	34.62	0.034
lego	34.88	0.011	34.67	0.011	34.67	0.011
materials	28.26	0.049	28.41	0.046	28.38	0.048
mic	35.14	0.009	34.98	0.010	35.18	0.009
ship	22.89	0.335	22.83	0.346	22.87	0.340
barn	26.76	0.273	26.48	0.280	26.55	0.279
caterpillar	25.16	0.147	24.99	0.151	24.82	0.152
family	21.04	0.159	32.01	0.060	32.50	0.058
truck	26.91	0.129	26.80	0.130	26.77	0.132
fern	23.76	0.225	23.74	0.230	23.51	0.233
room	33.08	0.049	32.13	0.060	32.41	0.053

Tabella 4.1: Risultati dell'esperimento con test dell'Encoder Direzionale. Sono riportate le metriche di PSNR e LPIPS ottenute per ogni singola scena ed ogni differente encoder.

volutamente le capacità rappresentative delle architetture, in realtà sia paragonabile all'utilizzo delle armoniche sferiche a patto di un costante peggioramento, seppur minimo, delle proprietà soggettive.

Figura 4.2: Confronto tra alcune immagini esportate dai NeRF ottenuti con Spherical Harmonics a sinistra e Dummy Encoder a destra. Nel caso della scena Chair non vi sono differenze immediatamente rilevabili, nel caso della scena Mic l'encoder con armoniche sferiche riesce maggiormente nella resa del dettaglio della corona metallica finale del microfono.

L'encoder definito Passthrough, che in realtà sta a riportare la totale assenza di manipolazione dei dati in ingresso e quindi, l'assenza di un vero e proprio encoder, evidenzia il trend di attuale peggioramento delle metriche di valutazione di segnale nel caso di utilizzo dello Spherical Harmonics. A questo riguardo ricordiamo che, nel caso di Ref-NeRF, l'Integrated Positional Encoder che utilizzava armoniche sferiche, in realtà, piuttosto che lavorare con i raggi di visualizzazione della scena, lavora con

un'architettura che rappresenta la radianza non solo come radianza parametrizzata rispetto alla direzione di visualizzazione, ma come insieme di parametri dati anche dalla capacità e possibile riflessione dei raggi di visualizzazione da parte delle superfici. Instant-NGP invece considera la radianza al pari dell'approccio classico NeRF, cioè come funzione degli angoli di visualizzazione. Non si esclude quindi che, per quanto riguarda metriche che si avvicinano il più possibile a componenti fisiche del modello tridimensionale generato, il fenomeno osservato non sia dovuto ad una assenza strutturale e architetturale di moduli che effettuino la riparametrizzazione discussa. Sebbene quanto appena detto, l'aumento dimensionale dato dall'utilizzo delle armoniche sferiche potrebbe di fatto contribuire, sempre nel caso di scene con poca riflessione, a miglior resa del dettaglio superficiale che, si ricorda, essere più difficile da percepire per la metrica PSNR rispetto a LPIPS. Il paragone invece non tiene nel caso di superfici riflettenti: sia le metriche di segnale che quelle soggettive riportano risultati discretamente migliori.

Se ne conclude che, sebbene la scelta di utilizzare un encoder di questo tipo nel caso di Instant-NGP non sia né qualitativamente né computazionalmente efficace quanto l'utilizzo del multi-hashgrid, il costante leggero miglioramento che si ottiene in molte delle scene giustifica il suo utilizzo che si ricorda avere un costo computazionale relativamente basso, paragonabile a quello del Positional Encoder dell'architettura NeRF. Non si ha quindi necessità di riportare le differenze nel tempo di addestramento e overfit delle reti sulle singole scene in quanto le differenze risulterebbero irrisorie e, con molta probabilità, principalmente dovute a fattori non controllabili quali ad esempio il throttling delle componenti hardware. Va fatto notare che il risultato chiaramente outlier della scena family ottenuta con SH sia probabilmente dovuto ad un bug del sistema utilizzato in combinazione con l'architettura selezionata per il test. In particolare, il bug potrebbe essere ascrivibile ad un errore di memoria grafica durante il processo di valutazione dei risultati, poiché un'ispezione visiva soggettiva da parte dell'autore di questo lavoro non ha dato evidenza di percepibili variazioni tra le scene della stessa famiglia. Si fa notare inoltre come nella totalità delle scene ottenute, le singole variazioni metriche risultano essere quasi totalmente indistinguibili dall'occhio umano attento; sicuramente non percepibili da un osservatore esterno non preparato o su di un supporto diverso dalla GUI di sistema legata allo strumento utilizzato.

4.2 Esperimenti sul modello Neurale

4.2.1 Descrizione

Questa serie di esperimenti si concentra sulle reti neurali alla base delle architetture NeRF per la resa di scene tridimensionali a partire da input bidimensionale. Le riflessioni chiave vengono effettuate sull'utilizzo di semplici MLP che, in molti task della CV e della CG, sono state superate da architetture spesso convoluzionali, con memoria, attention learning e quindi strutture molto più complesse. Se un primo approccio verrà tentato nella sostituzione delle reti MLP, un secondo si concentrerà maggiormente nello studio interno delle ANN, portando a modifiche mirate dovute a processi interni alle architetture NeRF.

Viene reso noto il fatto che, in principio a questa serie di esperimenti, è stata tentato un rapido passaggio con architetture più complesse delle semplici MLP. In particolare, primo e unico test di questa tipologia è stato svolto utilizzando una Concurrent Neural Network delle stesse dimensioni delle MLP: 3 strati, stesso numero di parametri in ingresso, 15 per la rete sigma dati da caratteristiche spaziali e risposta multidimensionale dell'encoder e 20 per la rete color dati da 16 output della rete direzionale e risposta dell'encoder, con 64 neuroni per gli hidden layer. Sono stati utilizzati gli stessi attivatori, ottimizzatori e parametri di configurazione. Il test è stato effettuato sulla caratteristica scena Lego, valutando la metrica del PSNR e il tempo di risoluzione dell'addestramento in 30'000 step. Gli scarsi risultati ottenuti in un tempo molto più alto, uniti alle citazioni di articoli in cui le MLP vengono definite adeguate, superando architetture più complesse, hanno portato al termine di questo filone di esperimenti.

Nonostante ciò, i campi di radianza neurale generati con le tecniche viste finora condividono la stessa soluzione per la resa delle superfici generate da volumi. In particolare, attraverso il passaggio nelle reti dei raggi campionati da due o più immagini differenti è possibile, attraverso la propagazione di questi secondo più punti di vista, individuare la superficie di un volume sfruttando la triangolazione: nell'ipotesi che si stia osservando una superficie completamente nera su sfondo bianco in vista frontale nella prima immagine, nel caso della seconda vista, ad angoli di posa differenti, i raggi individueranno l'oggetto nero nello spazio e sarà possibile definire una profondità da entrambe le posizioni di visualizzazione. A questo punto però, nonostante sia possibile individuare il confine tra il nostro volume nero e il bianco dello sfondo, tutti i raggi che, nella seconda immagine, continueranno a visualizzare un'entità nera nell'intorno spaziale della superficie, continueranno a triangolare una profondità valida più interna all'oggetto. Ciò significare che la

densità superficiale dell'oggetto verrà distribuita per tutta la profondità dell'oggetto stesso in base alla visibilità nelle immagini fornite come input: è ovvio che un maggior numero di immagini prese da pose diverse consente di adagiare la densità con maggiore precisione all'interno del volume occupato dal cubo. Nonostante ciò, la densità volumetrica non potrà essere differenziata dalla densità di superficie: questo comporta che nel nostro campo di radianza neurale l'oggetto sarà definito come una densità superficiale piena. Da qui nasce la problematica di nuvolosità già verificata in letteratura[22][23] che si applica soprattutto alle superfici riflettenti e ai bordi delle scene in quanto, solitamente, non è possibile avere a disposizioni immagini che definiscano esternamente, o anteriormente nel caso delle riflessioni, i confini di volume esterni dell'oggetto. Le architetture rendono quindi gli oggetti di sfondo o riflettenti come nuvole di densità che, prese dal punto di vista dell'immagine originale, appaiono come è richiesto all'oggetto. Questo concetto porta ad una importante differenza tra gli output in forma di point cloud delle architetture NeRF, rispetto agli output in point cloud generati da metodi più classici come la fotogrammetria. Si avranno quindi point cloud più dense all'interno dei volumi rispetto alle point cloud quasi esclusivamente superficiali ricavate da metodi di tracciamento laser. Questo, oltre che a minimi problemi di resa della scena tridimensionale che molto spesso sono risolvibili con approcci quantitativi con maggior numero di immagini o qualitativi con migliori architetture, porta ad una riconosciuta difficoltà nell'ottenere mesh qualitativamente valide. La presenza di densità interna al volume porta, spesso, a difficoltà nella resa delle superfici esterne sulle quali adagiare o costruire la mesh dell'oggetto[33].

L'intera problematica viene spesso mitigata a livello architetturale con funzioni di normalizzazione o troncamento dei valori interni ai volumi[2][23]. Effettuando il campionamento di densità per la lunghezza di un raggio inviato all'interno della scena tridimensionale contenente un oggetto opaco si vuole che la densità nello spazio sia il più possibile impulsiva in corrispondenza della superficie, invece che, come avviene in casi non filtrati, distribuita più o meno omogeneamente all'interno del volume con valori positivi prossimi allo zero. Riguardo ciò, le architetture basate su Instant-NGP, tra cui anche torch-ngp, sfruttano un troncamento esponenziale dei valori interni ai volumi, con l'obbiettivo di aumentare la resa impulsiva della superficie. L'idea alla base dell'esperimento di questa sezione consiste nel modificare i parametri interni alla rete in modo che sia più facile per il filtro appena introdotto agire nell'individuazione del bordo superficiale dell'oggetto. In particolare, una delle idee alla base di questo concetto è ammettere densità negative: gli attivatori dei neuroni delle MLP che fondano le architetture sono esclusivamente ReLU. Questo attivatore setta a zero il contributo dei pesi negativi e fornisce risposta lineare per quelli positivi, risultando in una convergenza a zero dei pesi negativi stessi, aumentando la probabilità che la rete sia interamente definita da valori positivi. Si sceglie quindi di effettuare prove sfruttando diversi attivatori, al solito verificando la resa in PSNR e SSIM dei

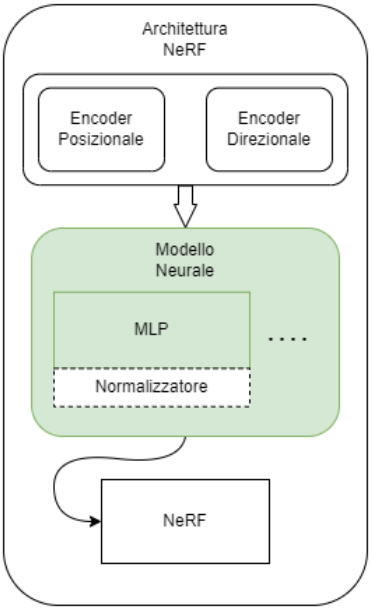

Figura 4.3: Diagramma di azione degli esperimenti sul modello neurale. La modifica viene fatta all'implementazione delle reti neurali alla base dell'architettura: sostituendo le MLP o modificandone gli attivatori. Si agisce sulla zona in verde. L'esperimento viene valutato sul campo NeRF generato.

render delle scene 3D. Il codice viene modificato e tutti gli attivatori delle reti density e color vengono modificati appropriatamente. In una prima fase vengono testati diversi attivatori su alcune scene 3D quali Lego, Room, Barn, Drums, Materials, Mic e Ship prese dai dataset Synthetic, Tanks&Temples e LLFF. Gli attivatori utilizzati nell'esperimento sono ReLU, Leaky-ReLU, Tanh, SELU, Mish e RReLU e vengono testati, per semplicità, solo su PSNR. Questo perché si nota che in casi di ovvie difficoltà di resa della scena, PSNR e LPIPS tendono a a rappresentare la qualità uniformemente. Gli attivatori sono stati scelti per valutazione del loro comportamento nella zona intorno alle x corrispondenti a 0: Leaky-ReLU in Fig.2.1 per l'introduzione della perdita, Tanh in Fig.2.4 per il comportamento linearizzabile al cambio di quadranti, SELU in Fig.2.5 per le capacità di self-normalization e l'andamento negativo, Mish in Fig.2.6 per l'andamento negativo nell'intorno di 0 che per valori minori delle x torna ad essere nullo, RReLU in Fig.2.7 per il comportamento randomico.

Individuato l'attivatore che più si avvicina alle performance ottenute, il problema è stato riportato al solo dataset Synthetic per il test di PSNR e LPIPS in un confronto tra l'attivatore di default e il migliore del test precedente, Leaky-ReLU. Al fine di consentire un miglior risultato all'algoritmo introdotto con il nuovo attivatore, mantenendo comunque quanto già valido per ReLU, si è scelto di aumentare il numero di step di addestramento da 30'000 a 40'000 in quanto è stato supposto un maggior

gap informativo dato dall'introduzione della perdita. Per l'intera sezione vengono mantenute costanti le chiamate al software in modo che questo, esattamente come nei test della sezione precedente, performi alla massima qualità. Tutti i test vengono effettuati sulla stessa macchina della sezione precedente, con stessa architettura software e distribuzione GNU/Linux.

4.2.2 Risultati

Per quanto riguarda la modifica all'architettura neurale con sostituzione della MLP con una Concurrent NN. La sorpresa legata al fatto che un'architettura fondata su concetti così semplici sia più che sufficiente nel rappresentare un task grafico così complesso come la radianza emessa dalle superfici trova alcune osservazioni nella sua moderazione. La prima osservazione è legata al fatto che sebbene il task sia complesso, la richiesta ad una singola architettura di riportare una sola scena, cioè di avere overfit in fase di training della rete sulla scena, semplifica di molto le esigenze rappresentative. La seconda osservazione è legata al fatto che, molto spesso, le architetture più complesse già citate nel caso delle CV e CG hanno come compito quello di risolvere task che sono legati a molte rappresentazioni; il problema dell'overfit non si pone. Stesso concetto che, presente come problematica relativa al dataset e alla perdita della discesa del gradiente nei task più comuni del ML e del DL, porta all'avvenuta modifica di alcuni parametri legati agli ottimizzatori che agiscono sulle reti NeRF[20].

I risultati ottenuti in Tab.4.2 sono quindi prevedibili e chiaramente discutibili: una maggior complessità della rete legata alla sua struttura interna e alla necessità di effettuare una backpropagation più complessa, portano alla maggior durata del processo di addestramento. Maggior durata che è strettamente collegata, anche, alla convergenza più tardiva. Un maggior numero di pesi per neurone, che è il caso del confronto tra MLP e Concurrent, porta ad una necessità di più step di addestramento per ottenere lo stesso risultato di convergenza all'overfit della scena. A riguardo però, la supposizione di maggior complessità, sebbene validata dai risultati ottenuti, non esclude che architetture più complesse non possano ottenere risultati migliori, magari se opportunamente ottimizzate. Non si esclude infatti che maggior capacità di rappresentazione o maggiore facilità nell'estrazione di informazione da parte di reti tipicamente più complesse non possano trovarsi ad ottenere risultati migliori nella generazione di un campo di radianza. In ogni caso, reti più complesse avranno sicuramente necessità di una maggiore occupazione di VRAM, nonché di una maggior difficoltà nella loro interrogazione da parte del renderer che, per ogni pixel, avrà necessità di richiedere molti più calcoli. Questo, in parte, aumentando la difficoltà di utilizzo di questi sistemi.

Architettura	PSNR	Tempo
MLP	35.05	26'35"
Concurrent-NN	31.76	1h48'21"

Tabella 4.2: Risultati dell'esperimento con test sull'architettura della rete neurale mettendo a confronto MLP con una rete Concurrent. Vengono riportati PSNR ed il tempo di raggiungimento di 30'000 step.

Per quanto riguarda invece i test sugli attivatori, espressi in Tab.4.3, il risultato dipende strettamente dalla risposta della rete. L'individuazione di una densità, che è punto chiave della rete sigma, è un problema di individuazione di un numero strettamente positivo. Inoltre, si può assumere che, maggiore il numero di raggi che nello stesso punto dello spazio individuano una occupazione, maggiore sarà la probabilità che quel punto tridimensionale sia effettivamente occupato. Risulta essere ovvio quindi come questo problema sia un problema che ben si adatta ad un andamento strettamente positivo, lineare. Quanto descritto è esattamente coincidente con l'andamento della funzione di attivazione di una unità ReLU. Con questa osservazione si giustificano quindi le motivazioni che portano alla definizione di questo attivatore come quello di default per tutte le architetture NeRF. In maniera similare, l'utilizzo di una tangente iperbolica come funzione di attivazione è molto consona quando il task in risoluzione tende ad avere caratteristiche che ne fanno oscillare i valori chiave nell'intervallo $[-1, 1]$. Il che quindi ci da idea della motivazione di fondo alla base del più scarso risultato ottenuto da questo attivatore nel caso in cui è stato portato. Mentre si può supporre che l'attivatore SELU ottenga un così basso punteggio dovuto al fatto che la sua funzione di attivazione unisce caratteristiche lineari positive a caratteristiche fortemente logaritmiche negative, e quindi salti la capacità di ottenere una automatica normalizzazione della rete, Randomized-ReLU introduce una componente randomica che non sembra auspicare una migliore convergenza.

Il risultato particolare ottenuto dall'attivatore Mish che tende ad essere quasi completamente sovrapponibile con ReLU ha l'alta probabilità di essere dato dalla buona somiglianza che c'è tra le due funzioni di attivazione. Nei casi delle scene Room e Mic, l'impiego di questa funzione di attivazione sembra dare più fedeltà di rappresentazione al dettaglio nel caso della scena con microfono sebbene, in presenza di riflessione nello schermo all'interno di room, ReLU sembra essere migliore nella resa della riflessione. Leaky-ReLU, che è ovviamente molto simile come funzione all'attivatore di default, conferma lo stesso trend di minore capacità di resa della riflessione con un minimo vantaggio nella resa dei dettagli. Per le scene selezionate, Leaky-ReLU ottiene risultati in PSNR addirittura leggermente migliori con l'unica eccezione proprio data dalla scena Room. Questo a parzialmente confermare l'ipotesi iniziale sulla resa effettiva del connubio attivatore-normalizzazione della densità in

Scena	ReLU	Leaky-ReLU	Tanh	SELU	Mish	RReLU
lego	35.05	35.11	34.58	34.18	35.02	34.54
room	33.64	33.45	32.13	31.27	32.83	-
barn	26.84	26.95	26.74	24.44	27.06	26.66
drums	25.61	25.72	25.44	25.08	25.65	25.48
materials	28.67	28.76	28.27	27.96	28.75	28.43
mic	35.19	35.49	35.00	34.37	35.67	34.25
ship	22.90	22.90	22.63	22.43	22.88	22.66
avg	**29.70**	**29.77**	**29.26**	**28.53**	**29.69**	**-**
Δ su ReLU	-	0.07	-0.44	-1.16	-0.01	-

Tabella 4.3: Risultati dell'esperimento con test sui diversi attivatori. Viene riportata la metrica di PSNR ottenuta per ogni singola scena e ogni singolo attivatore.

corrispondenza delle superfici.

Si passa quindi ad una valutazione ristretta al singolo dataset Synthetic del confronto tra reti con attivatori ReLU e reti con Leaky-ReLU. Quanto è chiaro in Tab.4.4 è che la differenza notabile tra i due approcci, prolungando gli step totali d'addestramento e testando il dataset nella sua interezza, rimane molto risibile, sebbene rispetto al test precedente ci sia in ogni caso un miglioramento qualitativo. Mentre l'incremento in positivo delle metriche in entrambi i casi, viste le condizioni a contorno dell'esperimento mantenute costanti, è sicuramente dato dall'incremento del numero di step che hanno quindi contribuito alla riduzione del gap informativo tra lo strumento addestrato e la realtà, l'appiattimento delle performance tra le due soluzioni può avere diverse spiegazioni. Non si escludono bias legati alla scelta delle scene nella fase precedente come non si esclude che l'estrema somiglianza tra le due funzioni di attivazione non possa portare a risultati dissimili: quanto di migliore si ottiene nel caso Leaky-ReLU in PSNR e nel caso ReLU in LPIPS non può essere escluso sia dato da fluttuazioni puramente aleatorie. Risulta inoltre anche difficile individuare e giustificare casi specifici dati dalle singole condizioni della scena trattata se non nel caso della scena Ship in cui, per quanto già detto nel paragrafo precedente, si nota una leggerissima perdita di qualità della rete con attivatori leaky probabilmente dovuta alle riflessioni nell'acqua. Tende quindi a manifestarsi empiricamente lo stesso fenomeno già descritto nel paragrafo precedente.

In ogni caso e per quanto visto in questa sezione, Leaky-ReLU si mantiene un'alternativa molto valida alle più comuni soluzioni predefinite, ReLU. Risulta capace di ottenere, con consistenza, un errore dovuto al segnale dell'immagine prodotta leggermente migliore, sebbene però incontri, con alta probabiltà, diverse problematiche nella resa delle riflessioni; ottiene allo stesso modo risultati paragonabili nelle altre

Scena	ReLU		Leaky-ReLU	
	PSNR	LPIPS	PSNR	LPIPS
chair	35.33	0.010	35.30	0.010
drums	26.36	0.062	26.45	0.058
ficus	32.28	0.019	32.28	0.021
hotdog	35.13	0.031	35.13	0.035
lego	36.27	0.009	36.30	0.008
materials	29.25	0.044	29.31	0.043
mic	35.56	0.007	35.57	0.007
ship	23.22	0.30	23.12	0.314
avg	**31.67**	**0.060**	**31.68**	**0.062**

Tabella 4.4: Risultati dell'esperimento con test tra ReLU e Leaky-ReLU. Vengono riportate le metriche di PSNR e LPIPS per ogni scena del dataset Synthetic ed ogni attivatore.

metriche. Il risultato, dovuto sicuramente ad una elevata somiglianza con la soluzione di default e ad un ragionamento teorico alla base che questi risultati parzialmente validano, rimane però troppo contenuto per giustificare una sostituzione consistente dell'attivatore classico con la nuova intuizione. Rimane, infatti, un test limitato alla singola implementazione e sarebbe di elevato interesse proseguire l'analisi applicando l'esperimento a un maggior numero di condizioni a contorno diverse.

4.3 Esperimenti sulla generazione di Mesh e Point Cloud

4.3.1 Descrizione

Un'estensione del tema trattato nella sezione precedente è la generazione di output a partire dai campi di radianza neurale. Nel caso in cui come output venga scelta la semplice interrogazione alla scena tridimensionale oppure un render video della scena, ad esempio nella quale ci si possa muovere al suo interno, sebbene si ottengano risultati di ottima qualità, una delle difficoltà d'approccio alla pipeline NeRF è proprio il passaggio da scene volumetriche di questo tipo agli altri sistemi più comuni. In particolare, ad oggi è difficile rendere i campi di radianza neurale all'interno dei software di progettazione di più comodo e radicato uso. Nonostante comunque esistano diversi plugin per il passaggio da NeRF a Blender, Unity e Unreal Engine[34][1], la mancanza di un sistema di rappresentazione che possa fare da ponte a questo nuovo strumento e metodologie più classiche incontra diverse difficoltà. Le stesse difficoltà che risiedono nella realizzazione di nuvole di punti a partire da densità volumetriche NeRF da utilizzare per poi applicare algoritmi di marching cubes o di individuazione di superfici di Poisson. Come risultato, questi algoritmi riportano mesh che molto spesso subiscono l'effetto cloudy che è stato descritto ampiamente nella sezione precedente, riportando quindi difficoltà nel porting dei modelli ottenuti con questa tecnica a causa della perdita qualitativa di consistenza della superficie dell'oggetto da rappresentare. Con la stessa filosofia per la quale sono state introdotte le architetture NeRF, cioè quella di sostituire modelli matematici per la risoluzione di task di CG con modelli statistici quali le ANN, sono state valutate diverse soluzioni in letteratura per la sostituzione degli approcci più classici.

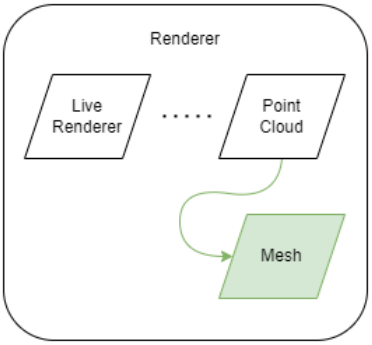

Figura 4.4: Diagramma di azione degli esperimenti sul processo di generazione di mesh. Si agisce modificando l'algoritmo per il passaggio da nuvola di punti a mesh tridimensionale, in verde. L'esperimento viene valutato sul risultato in output.

[1]Tre dei software di progettazione grafica più utilizzati.

In particolare, la soluzione prescelta che fa da base agli esperimenti che verranno introdotti in questa sezione è Deep Marching tetrahedra, una tecnica che utilizza approcci tipici del Deep Learning per definire il passaggio da una point cloud e una forma primitiva, quali una sfera o una rappresentazione stilizzata dell'oggetto, ad una mesh che si vuole ottenere il più simile possibile agli spazi ingombrati dalla point cloud di partenza. Al fine di questo esperimento si vedrà la rappresentazione in mesh e point cloud della scena Lego realizzata con il framework Nerfstudio; point cloud che, assieme alla superficie sferica di partenza, verrà utilizzata con l'algoritmo di generazione di mesh dmtet, Deep Marching TETrahedra. Viste le proprietà osservabili valutabili semplicemente del caso e viste le difficoltà che si avrebbero nell'utilizzare metriche oggettive, ai fini di questo esperimento verranno riportati soltanto risultati grafici, da valutare in maniera empirica. Si cita comunque la letteratura nell'individuare nelle superfici di Poisson la migliore soluzione in risposta a questo problema nel caso di pipeline NeRF[34]. È stato necessario, però, affinché fosse possibile ottenere il completo funzionamento dello strumento, uniformare le componenti del dataset relative alle trasformazioni matriciali delle pose delle camere manualmente, allontanandosi dai tre split previsti inizialmente di train, valutazione e test, passando ai semplici training e valutazione con composizione 90-10 invece che 50-50. Train e valutazione sono quindi uniti nello stesso contenitore, lasciando la scelta allo strumento del 10% delle immagini e pose da utilizzare come valutatori. Questa manipolazione è resa possibile dalla semplice considerazione soggettiva che verrà fatta del risultato in mesh, distante dalla necessità di dover individuare parametri numerici.

Per adattare quanto richiesto ad uno script utilizzabile per la generazione di mesh è stato necessario sfruttare diverse risorse tra cui codice Kaolin[2] e strumenti di conversione di formato[3]. Per la risoluzione dell'addestramento degli strumenti deep alla base dell'algoritmo sono stati terminati 30'000 step di addestramento di una rete MLP. La macchina sulla quale sono stati effettuati i test coincide con quella utilizzata nelle sezioni precedenti.

[2]Kaolin è uno strumento facente parte del framework nVidia Optimus. Software poprietario a questo indirizzo: https://developer.nvidia.com/kaolin. Mette però a disposizione una serie di repository in cui è possibile rinvenire il codice di esempio utilizzato in questa sezione[42].

[3]Disponibile qui, corredato da codice open-source: https://products.aspose.app/3d/conversion/ply-to-usd. Effettua un trascurabile scambio degli assi X e Z, comportando una rotazione arginabile con una rotazione della scena.

4.3.2 Risultati

Lo strumento Deep Marching Tetrahedra risulta particolarmente interessante in applicazioni in cui si abbia già a disposizione un volume che può fungere da scheletro della point cloud: lo strumento, infatti, in un processo iterativo suscettibile all'addestramento della rete deep MLP alla sua base, adatta un volume iniziale al riempimento della point cloud. La problematica riscontrabile nel processo di addestramento del caso riportato è proprio data dall'assenza naturale di questa componente di base. Nel caso di un campo di radianza neurale da riportare in forma di mesh, la definizione di uno scheletro esula da quanto si ha a disposizione: sarebbe quindi richiesta una sua produzione manuale oppure un processo di estrazione di tali informazioni. Sono quindi richiesti volumi che definiscono in maniera approssimata la forma degli oggetti nella scena. Ipotizzando un processo di estrazione automatico basato su concetti simili a quelli presentati, questo potrebbe tradursi, al termine di una ricorsione, nel risolvere esattamente quanto fatto: fornire una base di partenza altamente generica data da un volume sferico posizionato nel centro della scena. Per quanto visibile in Fig.4.5, nonostante i 30'000 step di addestramento, alcune componenti della mesh riconducibili al punto di partenza in forma di sfera sono facilmente individuabili nelle insolite bombature delle superfici.

La differenza di colore notabile tra i diversi approcci è data dalle capacità dell'algoritmo Nerfstudio di adagiare il contenuto in colore della radianza trasformato in texture sui punti individuati nel passaggio a point cloud. Da qui, applicare un algoritmo di posa simile consente di ottenere informazioni sul colore anche nel file object che definisce la mesh. Lo stesso strumento[34], all'interno della sua documentazione, consiglia fortemente l'utilizzo delle superfici di Poisson come metodo per la generazione finale del grafo 3D: si è quindi scelto, in quanto il risultato ottenuto dall'algoritmo dmtet non risulta entusiasmante, di evitare test ulteriori se già ben documentati. Se ne conclude però quanto la suite Nerfstudio sia veramente all'avanguardia come completezza di utilizzo e quanto, sebbene in sviluppo, sia sulla strada giusta per consentire un sempre più ampio utilizzo di strumenti come le NeRF, declinate nelle varie e diverse architetture e i loro task specifici. Ciò ovviamente alla luce di quanto ottenuto: sebbene la mesh generata non sia perfetta o paragonabile a quanto sarebbe in grado di realizzare un 3D artist, il processo è sicuramente molto rapido e, in qualche modo, può fornire un'ottima base di partenza proprio all'artista incaricato di lavorare con queste strutture 3D. Uno dei parametri chiave che può contribuire fortemente a questo è la capacità degli approcci NeRF di mantenere le proporzioni degli oggetti riportati attraverso una adeguata triangolazione delle acquisizioni e opportune indicazioni riguardo la camera che ha effettuato la cattura. Come ultima ossevazione della sezione, si nota come la resa della point cloud osservata dall'esterno sia notevole, caratteristica propria e riconosciuta degli approcci NeRF.

La problematica più grande nella generazione della mesh presentata è proprio legata al punto di contatto tra oggetto e superficie in cui sono presenti diverse assenze di chiusure. Questo a causa del già citato fenomeno di point cloud dense, che complicano il lavoro degli algoritmi attuali nel passaggio a mesh. Nonostante ciò, diversi dei dettagli più complessi da presentare, come i famosi punti di aggancio dei mattoncini lego, vengono resi con buona qualità e bassa approssimazione.

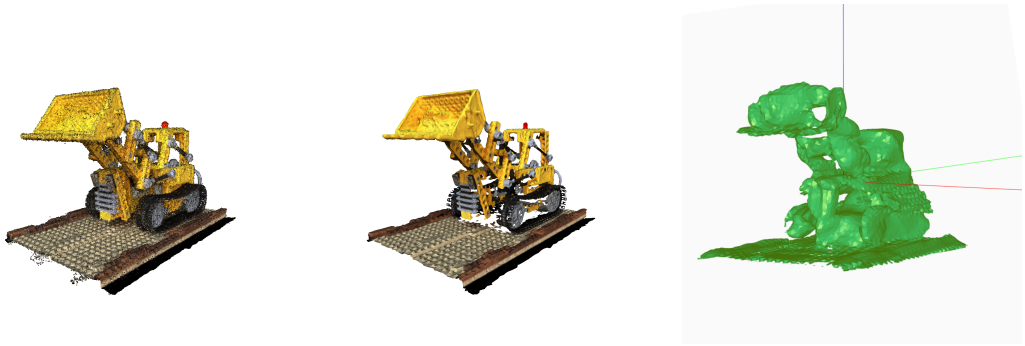

Figura 4.5: Confronto tra la pointcloud esportata con Nerfstudio della scena Lego da cui sono state esportate, al centro, la mesh con Nerfstudio e superfici di poisson e, a destra, la mesh con Deep Marching Tetrahedra.

4.4 Esperimenti nel Cultural Heritage

4.4.1 Descrizione

L'ultima e più corposa serie di esperimenti ha come obiettivo il condensare la conoscenza fin qui appresa nella risoluzione di un task più pratico, che abbia più a che fare con il mondo reale e che soprattutto possa essere replicato in maniera relativamente agile senza specifiche conoscenze riguardo le proprietà intrinseche degli strumenti ad architettura NeRF. Nel dettaglio, da ogni serie di esperimenti svolti finora sono state estratte osservazioni che hanno contribuito alla stesura e progettazione di quanto verrà descritto in questa sezione. Gli esperimenti svolti sugli encoder hanno portato alla scelta di una architettura neurale che supportasse encoder capaci di ottenere ottima qualità di rappresentazione senza però avere forti svantaggi legati alla richiesta computazionale ed in particolare al tempo di addestramento in overfit della singola scena. Gli esperimenti sull'architettura neurale hanno portato alla scelta di una soluzione che potesse essere il più possibile correttamente dimensionata, cioè capace di rappresentare quanto richiesto con un elevato numero relativo di parametri e neuroni ma senza incorrere in architetture sovradimensionate o eccessivamente complesse che avrebbero pregiudicato resa, ottenimento e portabilità della soluzione addestrata. Gli esperimenti svolti sulla generazione di output a partire dagli strumenti a disposizione hanno confermato come, sebbene con qualche importante drawback, le soluzioni fornite da Nerfstudio siano potenti, efficaci e adatte alla maggior parte dei task. Osservazione appena descritta che, per l'appunto, rimane obbiettivo principale del framework.

Detto ciò, nella scelta di una soluzione che possa essere addestrata in tempi discreti ottenendo al tempo stesso qualità che sono un buon punto di riferimento per lo stato dell'arte, la scelta è ricaduta sulle architetture basate su Instant-NGP e Nerfacto. La scelta viene svolta in un'ottica generalista: il task da risolvere vuole essere quello della semplice replica della scena tridimensionale e, per quanto riguarda la base di informazioni di partenza, non presenta particolari necessità o complicazioni. Per questo motivo, tutte le architetture molto lodevoli create per la risoluzione di task specifici, come alcune di quelle già citate, sebbene interessanti dal punto di vista della ricerca, non sarebbero adatte per la risoluzione del task in definizione. Le altre architetture citate come Mip-NeRF o Ref-NeRF risultano potenti ed adatte al compito descritto ma hanno il fortissimo svantaggio legato alle richieste computazionali ed al tempo di addestramento che, sebbene per qualità raggiungibili almeno equivalenti[23], avrebbero pregiudicato il numero di test effettuabili. Questo semplicemente perché il singolo test avrebbe richiesto un tempo diversi ordini di grandezza maggiore.

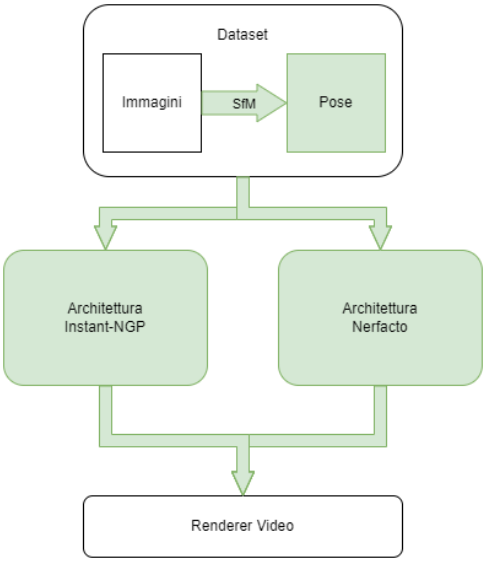

Figura 4.6: Diagramma di azione degli esperimenti sul processo per la generazione di NeRF nel Cultural Heritage. Il processo, questa volta più ampio, testa diversi processi di Structure from Motion e diverse architetture NeRF. I risultati vengono valutati sui risultati delle metriche applicate ai campi NeRF e alcuni video render.

Le due architetture proposte integrano, per questi motivi, il multi-resolution hashgrid encoder introdotto per Instant-NGP e, nel caso di Nerfacto, vengono utilizzati i tweak descritti nella sezione in cui è stata introdotta. I due approcci risultano essere quindi decisamente similari, motivo per il quale, all'interno delle prove che stanno per essere riportate, sono indicati diversi confronti tra i due sistemi. Sistemi che performano similmente sia in approcci bounded che unbounded eliminando il doversi porre il problema da parte dell'utilizzatore meno esperto. Oltretutto, eccetto alcune piccole differenze che verranno discusse in fase di discussione dei risultati nel capitolo seguente, dal punto di vista di capacità di calcolo necessarie per entrambe le architetture si parla di high-end consumer products, quindi macchine che, sebbene possano rivelarsi costose, sono generalmente disponibili al mercato consumer con ampie capacità. L'insieme di queste osservazioni avvicina ancor di più l'utilizzo di questi strumenti da parte di un tecnico non eccessivamente formato, in questo modo la problematica di avvicinamento a questi strumenti scema e un maggiore utilizzo da parte del pubblico aumenta le possibilità raggiungibili.

Descritte le scelte che hanno portato alle due architetture selezionate, sono state ritenute ininfluenti le scarse limitazioni che si hanno anche per la scelta del dataset. È risaputo in letteratura che i campi di radianza neurale non soffrono eccessivamente se generati a partire da scene sufficientemente illuminate[27] e, per quanto già definito, non si hanno particolari necessità legate alla condizione di bounding della scena. Per quanto perciò descritto, viste anche le possibilità di ricostruire pose con strumenti

di Structure from Motion, sono stati scelti dataset che potessero rivelarsi adeguati al task mettendo, allo stesso tempo, alla prova le capacità rappresentative degli strumenti. Viste le relative poche limitazioni, sono stati scelti tre dataset derivanti da acquisizioni svolte da droni in volo in zone con oggetti o costruzioni facenti parte del patrimonio culturale regionale.

Visto che, inoltre, i droni utilizzati hanno la necessità di mantenere un tracciamento di posizione, angoli di volo e relative accelerazioni per mantenere un volo fluido e controllabile, si è scelto di sfruttare questi dati nel tentativo di superare, quando possibile, gli strumenti di ricerca pose quali COLMAP e Metashape. Questo poiché gli strumenti di Structure from Motion, sebbene decisamente potenti, rispetto a condizioni estramemente controllate, tendono a definire un discreto errore nella generazione degli angoli di posa che tende a propagarsi fino ad inficiare sulla qualità del campo di radianza generato. Motivazione che spinge Nerfacto, che per definizione vuole essere uno strumento più generale rispetto a Instant-NGP, possiede a livello architetturale diversi accorgimenti per il miglioramento delle pose in fase di addestramento ed in risposta a queste.

I tre luoghi su cui generare scene tridimensionali corrispondono a quelli già citati: il santuario di Macereto, il castello Magalotti e la piazza di San Ginesio. A questi faremo quindi corrispondere i dataset macereto, magalotti e sanginesio. I dati estratti da Macereto derivano da un intero video di circa 100 secondi in formato .mp4 con risoluzione 4k e 30fps. Da questo video vengono estratti attraverso uno script apposito 300 frame ai quali, grazie al fatto che il drone utilizzato è capace di memorizzare una serie di informazioni sul volo, vengono associati dati di volo relativi alla posizione GPS del drone durante l'acquisizione. Questi dati vengono quindi processati e viene effettuata una trasformazione di posizioni in formato EPSG:4326 (VGS84) per la rappresentazione di coordinate latitudinali e longitudinali sul globo terrestre al formato EPSG:332633 (UTM) per la rappresentazione su mappe cartesiane, assunti quindi gli assi X e Y. Supposta la direzione del drone rispetto all'asse Nord mantenuta costante, ruotando i dati in base alla direzione del drone e l'orientamento latitudinale e longitudinale, non vengono memorizzati, poiché non disponibili, i dati di volo di Yaw, Pitch e Roll. L'assenza degli angoli della camera funge però da test del limite di funzionamento dei meccanismi di ottimizzazione delle pose interni all'architettura Nerfacto. Il dataset magalotti, similare, viene estratto considerando i primi 1130 frame di un video 4k a 30fps che riportano il decollo in linea retta del drone in oggetto, mantenendo l'imbardata, il rollio e il beccheggio costanti. Di questi, vengono estratti 366 frame. Il fatto che i parametri degli angoli di volo siano rimasti costanti consente, anche in assenza di dati su (Y, P, R), di ipotizzarne la stazionarietà considerando solo il movimento tridimensionale del drone. Infine, il dataset sanginesio viene generato a partire da 140 immagini scattate durante un volo, contenenti stavolta tutti i dati relativi agli angoli di volo, che verranno quindi approssimati agli angoli della camera.

Per testare la pipeline ed individuare la migliore condizione di utilizzo di questi oggetti in grandi spazi, per tutti e tre i dataset, i dati di navigazione del drone vengono, per test paralleli, sostituiti ai dati delle camera matrix ottenuti con gli strumenti COLMAP e Metashape separatamente. In questo modo, si testano le condizioni di robustezza sia delle misurazioni del drone utilizzato, sia degli algoritmi di ricerca delle pose. La sezione si divide quindi in tre serie di esperimenti, ognuna realizzata con uno dei dataset a disposizione.

Per macereto vengono generati inizialmente sei campi di radianza neurale: tre con Instant-NGP relativi al sub-dataset ottenuto considerando le pose fornite dai sensori, a quello ottenuto generando le pose con COLMAP e a quello ottenuto generando le pose con Metashape; tre con Nerfacto, riprendendo gli stessi sub-dataset. Se la generazione delle pose a partire dai dati di navigazione richiede un tempo triviale, Metashape richiede un tempo medio di 18 minuti, COLMAP un tempo medio di 36 minuti. Di queste scene generate, per ognuna viene effettuata una valutazione di PSNR, SSIM e LPIPS sfruttando la funzione apposita di Nerfstudio. Dopodiché, per una valutazione soggettiva del risultato ed in particolare della qualità della resa grafica viene effettuato, di ogni scena, un video render realizzato interpolando su di alcuni punti chiave definiti nella GUI Nerfstudio. Questo consente di ottenere diversi render con la stessa traiettoria di volo virtuale sulle sei scene tridimensionali generate. I risultati dell'esperimento sono consultabili in Tab.4.5

I dataset magalotti e sanginesio, con risultati in Tab.4.6 e Tab.4.7, vengono utilizzati allo stesso modo: vengono realizzate sei scene tridimensionali da campi di radianza neurale per mettere a risalto il confronto tra approcci manuali e SfM sulle due architetture scelte. Nel caso del dataset relativo al castello Magalotti, l'alta frequenza di campionamento delle immagini, a fornire quindi frame simili, sembra inficiare negativamente sulla capacità di ricostruzione di pose di COLMAP che, oltre ad impiegare una media di 51 minuti, riesce a fornire soltanto 247 frame correttamente allineati, scartandone altri; nessun problema per Metashape che impiega 19 minuti per la risoluzione del task di generazione; ancora meno per l'approccio manuale che anche qui impiega un tempo di computazione triviale. Nel caso del dataset relativo alla piazza di San Ginesio, gli strumenti mantengono un tempo computazionale che rimane lineare rispetto al dataset macereto visto il minor numero di immagini: 8 minuti per Metashape, 17 per COLMAP. Con gli stessi obiettivi sperimentali vengono effettuate le valutazioni suddette con le metriche introdotte e con osservazioni soggettive dei video render prodotti.

Visto il numero di scene da produrre e i processi computazionalmente complessi da risolvere, per questi esperimenti viene utilizzata una macchina di tipo cluster con circa 100GiB di RAM e una scheda grafica RTX A6000 con 50GiB di VRAM e TDP massimo di 300W.

Per concludere la serie di esperimenti si è scelto di validare quanto ottenuto confrontando gli output realizzati con quanto possibile ottenere con architetture Nerfacto di maggiori dimensioni, con maggiori parametri e più neuroni: eccetto le scene tridimensionali di macereto e sanginesio con pose ottenute da dati di navigazione, sono state replicate le scene Nerfacto utilizzando l'opzione Nerfacto-Big di Nerfstudio. La risoluzione di queste architetture richiede capacità computazionali attualmente disponibili con schede grafiche nVidia top di gamma del mercato consumer, o schede grafiche per professionisti o cluster quali quelle presenti nella macchina utilizzata. Si esce quindi da un discorso di semplice ottenimento dei risultati, ma col fine di validare quanto ottenibile con strumenti hardware più accessibili. Con le stesse modalità dei casi appena introdotti, anche di queste architetture viene fatta una valutazione in base alle metriche introdotte, riportata nelle Tab.4.8, Tab.4.9 e Tab.4.10 e, sempre per una valutazione soggettiva del risultato, vengono effettuati dei video render delle stesse traiettorie suddette.

4.4.2 Risultati

All'interno di questa sezione verranno inizialmente riportate osservazioni strettamente legate alla scena ed ai test sul dataset presi in analisi, dopodiché la trattazione tornerà generale nell'analisi degli esperimenti nel loro insieme. Viene introdotta nuovamente, in questa sezione, la metrica SSIM che si ricorda essere una metrica di similarità tra patch confrontate a diverse risoluzioni di due immagini, una originale, una generata dal NeRF. La metrica riporta un valore più alto se migliore, più basso se peggiore; andamento coerente con PSNR, opposto a LPIPS.

Quanto svolto per il dataset Macereto dipende fortemente dalle informazioni che si hanno a disposizione: i frame video sono associati ad una struttura che riporta soltanto i dati riguardanti la posizione GPS del drone. Non avendo a disposizione ulteriori informazioni riguardo l'orientamento del drone nello spazio, sono state fatte diverse assunzioni: analizzando il video soggettivamente, le variazioni d'angolo rispetto all'asse di partenza sono sembrate trascurabili, così come beccheggio e rollio. Inoltre, le trasformazioni con proiezioni richieste per il passaggio da sistema GPS basato su latitudine e longitudine ad un sistema metrico sono svolte con la massima precisione. I dati in Tab.4.5 mostrano come, nel caso della scena addestrata con le pose ricavate dai dati di navigazione del drone, le assunzioni fatte siano in realtà smontate. La scena risulta fortemente rumorosa e, con una esplorazione manuale attraverso la GUI di Nerfstudio, è possibile notare come solo alcune zone siano visivamente consistenti l'una rispetto all'altra. Questa considerazione ha portato ad un confronto visivo delle pose ottenute con quelle ricavate dagli strumenti di SfM e se ne conclude che, sebbene visivamente non sembrasse ci fosse un'ampia variazione

degli angoli di imbardata, specialmente in alcuni tratti della traiettoria di volo questa era consistente. Si può notare anche come, sebbene Nerfacto sia predisposta di tecniche che dovrebbero mitigare il forte errore introdotto dato dall'assunzione fatta, questo non riesce a riportare l'errore in un range accettabile. Ne risultano un più basso rapporto segnale-rumore PSNR e più scarse metriche di similarità SSIM e LPIPS, sia per le scene addestrate con Instant-NGP che per quelle addestrate con la architettura introdotta proprio con Nerfstudio.

D'altra parte, gli strumenti basati su SfM ottengono risultati che possiamo definire simili. La differenza più importante però, non presentabile con dati numerici, risiede nel processo di orientamento ed estrazione della direzione Nord relativa della scena. La differenza sostanziale, infatti, la fa il processo di ricerca ottimale delle corrispondenze tra feature dei frame: l'implementazione di COLMAP interna a Nerfstudio, dopo aver effettuato l'estrazione delle feature caratteristiche da ogni immagine, va alla ricerca di una serie di immagini consistente che sia la migliore disponibile sulla quale far partire il processo di allineamento a valle. Questo fa sì che COLMAP cominci ad ordinare le immagini con immagine di partenza, che quindi definisce l'orientamento iniziale, che è casuale rispetto alle immagini del dataset, dove la casualità è strettamente legata alle proprietà intrinseche del set di immagini. Ne risulta che per lo stesso set e le stesse feature estratte, COLMAP trovi sempre lo stesso sub-set di immagini di partenza e quindi orienti il dataset sempre allo stesso modo. Il che, però, risulta in un orientamento Nord relativo altrettanto casuale della scena NeRF rispetto, se non altro, all'ordine di acquisizione dei frame in input all'algoritmo. Metashape, invece, avvia il suo processo di allineamento dalla prima immagine fornita, facendo sì che l'orientamento iniziale sia strettamente legato all'orientamento della prima immagine acquisita. Inoltre, dal mero punto di vista delle performance, per la risoluzione del task citato nella prima parte di questa sezione di individuazione di due dataset con SfM, Metashape impiega 13'28" mentre COLMAP 38'09". Non si esclude che la differenza possa essere data da un diverso numero di feature estratte per ogni immagine, o da altri parametri interni ai due strumenti, sebbene comunque, per quanto la funzione di allineamento di Metashape sia protetta dalle licenze del software, si presuppone che gli algoritmi siano simili. Dal punto di vista dei risultati ottenuti, anche se finora le performance computazionali hanno riportato differenze, per quanto riguarda la qualità oggettiva della scena, i risultati sono pienamente sovrapponibili assumendo che le minime differenze ottenute siano completamente aleatorie. Il fatto che entrambe le architetture convergano agli stessi risultati sottolinea ancora come sebbene le pose individuate siano numericamente differenti, possano però essere relativamente equivalenti a meno di una rototraslazione della scena e quindi di tutte le camera matrix.

L'analisi prettamente soggettiva dei render video esportati, d'altra parte, mette in flebile vantaggio quanto ottenuto con Metashape: sebbene i singoli frame chiave

Macereto	Instant-NGP			Nerfacto		
	Drone	COLMAP	Metashape	Drone	COLMAP	Metashape
PSNR	18.51	23.24	23.43	17.63	22.40	22.37
SSIM	0.627	0.729	0.733	0.614	0.676	0.674
LPIPS	0.544	0.352	0.351	0.533	0.406	0.406

Tabella 4.5: Risultati dell'esperimento con confronto tra Instant-NGP e Nerfacto nei dati estratti dal dataset Macereto con pose ricavate dall'estrazione dei parametri di volo del drone, da COLMAP e da Agisoft Metashape.

della scena siano prettamente equivalenti, in fase di movimento il campo di radianza neurale addestrato con il software proprietario sembra rispondere con più morbidezza e, in entrambe le architetture, appare meno aliasing di movimento. Ciò può essere dato da fenomeni di motion blur legati all'aliasing che sono ridotti nella sintesi di immagini lontane dalle acquisizioni del dataset nel caso Metashape, a simboleggiare una probabile leggerissima riduzione del rumore nelle pose.

Nel confronto tra le due architetture invece, sebbene Instant-NGP sembri ottenere risultati migliori dal punto di vista metrico, per quanto riguarda un'analisi data dall'esplorazione della scena e dalla presa visione dei video esportati, i due risultati sembrano equivalenti. Non si evidenziano particolari fenomeni di clouding in un caso o nell'altro, probabilmente a causa del fatto che gli strumenti di SfM utilizzati, fornendo pose altamente precise, saturano le capacità di reallinamento e adjustment delle architetture Nerfacto. Se ne conclude che gli approcci siano equivalenti sia dal unto di vista di tempo computazionale richiesto, aggirandosi entrambe le soluzioni intorno ai 18' per i 30'000 step, sia per quanto riguarda la qualità raggiunta.

Figura 4.7: Macereto: confronto tra Instant-NGP, a sinistra, e Nerfacto, a destra, nella resa di un campo largo dalle pose generate con Metashape. Nerfacto riesce in una migliore resa degli alberi nello sfondo, Instant-NGP in una migliore resa della pavimentazione erbosa e della recinzione in legno. Il rumore sulla destra è assenza di informazione e viene codificato con valori RGB casuali; Nerfacto ne effettua a livello di architettura uno smoothing.

Nel caso gemello del dataset Magalotti, dove il punto di partenza è sempre un video

Figura 4.8: Macereto: confronto tra Instant-NGP, a sinistra, e Nerfacto, a destra, nella resa del dettaglio del portone di ingresso in una angolazione lontana da quelle presenti nel dataset. Si nota come il risultato Nerfacto sia più morbido, con più blur ma meno artefici.

4K in cui si ha a disposizione il tracciato GPS ma non l'orientamento del drone e della sua camera, viene effettuata un'assunzione su alcune proprietà dello specifico volo: in particolare, riportando il video iniziale un frammento relativo al decollo ascensionale del quadricottero in cui, caso voglia, che non vi si riscontra visivamente una variazione degli angoli di volo, come è già stato descritto, viene selezionato un frammento del video iniziale. Di questo frammento vengono estratti 366 frame video che vengono opportunamente trasformati in immagini in cui, data l'assunzione di staticità degli angoli di visualizzazione, è stato assunto un'orientamento costante della camera, con sola traslazione nello spazio tridimensionale. Inoltre, riportando i frame scelti in decollo, la variazione più importante nelle coordinate è data soprattutto dall'altezza dal suolo. Altezza che ovviamente deriva dall'altitudine GPS, parametro che è risaputo, allo standard attuale europeo, essere di risoluzione affidabile con precisione superiore al metro, meglio se nell'ordine delle decine. Dati questi presupposti, sebbene in termini di PSNR i risultati siano peggiori, ci si trova ad avere risultati strettamente paragonabili nelle metriche di similarità SSIM e LPIPS tra quanto appreso in NeRF dai dati direttamente ricavati dal geopositioning del drone. Questo risultato diventa ancor più interessante se l'analisi, da generica in entrambe le architetture, viene portata nel caso Nerfacto: qui si ha una forte corrispondenza tra i risultati, che siano essi legati ai dati estratti dal drone o con SfM. Risultati intra-architettura così simili che hanno probabilità di essere dovuti al fatto che il dataset porta un numero consistente di immagini che riportano però un frammento molto minimo della sfera di visualizzazione: il momento del decollo avviene inquadrando un'unica costruzione che viene quindi ripresa da terra fino a superiormente al tetto.

Quanto descritto fa sì che in questo caso, l'estrazione di dati di volo dal drone sia altamente confrontabile con tecniche basate su SfM. Inoltre, per l'estrazione delle pose da questo dataset, data una forte somiglianza tra le immagini, vista la minima variazione in posizione e dato un loro maggior numero, gli algoritmi di SfM risultano più lenti: se Metashape rimane comunque accettabile in un tempo vicino ai 32'35",

COLMAP impiega 1h36'22" riuscendo ad allineare un numero minore di immagini, sebbene data la alta similarità e il grande numero di frame, i risultati non sembrano risentirne particolarmente. Quanto descritto si contrappone, invece ad un tempo di estrazione dati e calcolo delle camera matrix da parte dello script di generazione del dataset Drone che rientra invece nell'ordine dei secondi e, non solo calcolando le matrici, ma estraendo anche i frame dal video. Se ne conclude quindi che possano esistere condizioni per le quali una corretta ed oculata progettazione dell'acquisizione possa dare subito risultati rilevanti evitando, magari in caso di dataset molto grandi e più estesi e complessi di questo, lunghi processi di ricerca pose con SfM.

Un'analisi che invece vuole paragonare le due architetture, conferma quanto visto prima: le architetture Nerfacto ottengono punteggi in metriche leggermente inferiori ad Instant-NGP, dove una più marcata differenza è presente nel segnale immagine, mentre ve ne è meno nelle metriche di similarità. Vengono quindi replicate le osservazioni fatte per il dataset precedente: gli strumenti di SfM probabilmente funzionando a dovere saturano le capacità di ottimizzazione pose di Nerfacto, facendo venir meno una delle soluzioni più efficaci interne all'architettura. Inoltre, sebbene Instant-NGP sia più dispendiosa a livello di memoria durante il training con picchi di differenza che si attestano intorno ai 2GiB di VRAM, la qualità raggiunta è strettamente similare, in tempi anche essi molto poco distanti. Per le due architetture si rimane, infatti, nell'ordine dei 20' di addestramento sulla macchina con A6000 descritta nel capitolo precdente.

Figura 4.9: Magalotti: confronto tra architetture Nerfacto a partire da dati COLMAP, a sinistra, e dati Metashape, a destra, su di un campo largo della scena. Come è possibile notare, le maggiori difficolta di calcolo avute dallo strumento COLMAP relative a questo dataset si traducono anche in una deformazione e stretch orizzontale della scena, non presente per i dati estratti con Metashape. La deformazione è data sia da una minore precisione, sia da diversi errori nella definizione dei parametri intrinsechi della camera, quali quanti dipendono dalla curvatura della lente.

L'ultima osservazione che è possibile estrarre dai dati in Tab.4.6 è attinente ad un confronto puro tra COLMAP e Metashape che da vantaggio a COLMAP. Come nel caso precedente, però, un'analisi visuale e soggettiva dei render video esportati da idea di come quanto ricavato dallo strumento a sorgente chiusa porti

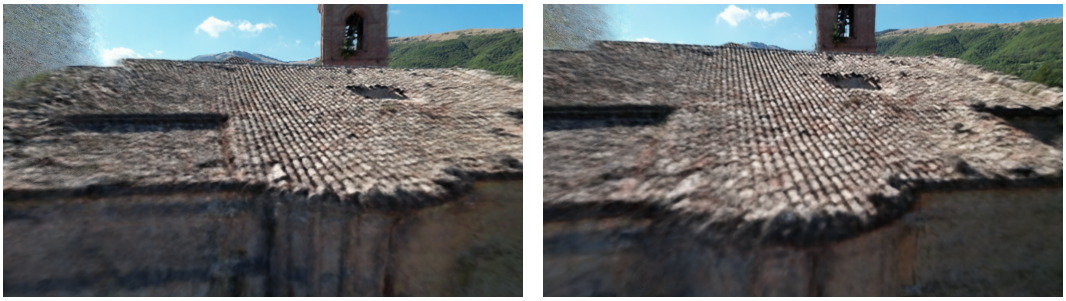

Figura 4.10: Magalotti: confronto tra architetture Nerfacto a partire da dati COLMAP, a sinistra, e dati Metashape, a destra, di un dettaglio del tetto della costruzione oggetto del dataset. La deformazione della scena fa sì che vi sia distorsione anche nella generazione degli output render a partire dalla stessa posa nello spazio. Nonostante ciò, è possibile notare come vi sia meno aliasing nella soluzione Metashape e la scena sia lievemente più morbida in corrispondenza delle superfici non correttamente rappresentate, come la zona del tetto sulla sinistra delle due immagini.

ad un minore aliasing di movimento, portando quindi ad una maggiore morbidezza dei video esportati e quindi ad una migliore qualità video percepibile. Quanto appena descritto, che è accaduto anche nel dataset precedente, potrebbe darsi da una maggiore efficacia di uno strumento di SfM rispetto all'altro di individuare in maniera migliore e separatamente la posizione di visualizzazione o gli angoli della camera dell'osservatore. Assunto corretto quanto ipotizzato, una soluzione che potrebbe essere valida è relativa al fatto che COLMAP tenda ad essere più preciso nell'individuazione degli angoli delle camere, mentre dall'altra parte Metashape potrebbe essere più efficace nell'individuazione della posizione dell'osservatore per le singole immagini. L'ipotesi trova supporto proprio nel confronto tra i dati numerici e quanto osservabile nei render video continui: un migliore angolo di visualizzazione porta, soprattutto per immagini che hanno elementi principali relativamente lontani dall'osservatore, a delle metriche numeriche migliori in quanto c'è una maggiore corrispondenza tra la scena generata e le immagini di valutazione; una migliore posizione di acquisizione da parte di Metashape, legata però ad un peggior risultato negli angoli, può portare a risultati altrettanto validi qualitativamente, che soffrono però meno di aliasing, sia per la maggior precisione spaziale di acquisizione, sia per il fatto che maggiori difficoltà di ripresa degli angoli potrebbero introdurre del motion blur. Fenomeno che, se in termini ridotti, aumenta parzialmente la qualità percepita del video da parte dell'HVS in quanto avvicina il video stesso alla percezione umana del movimento. Quanto osservabile visualmente nei dati estratti dai parametri del drone a livello di render video si pone a metà tra i due approcci SfM: non vi è eccessivo aliasing nel movimento dell'osservatore, sebbene questo possa essere dato anche da un leggero ma maggiore rumore nella rappresentazione delle superfici che, in questo caso, perdono leggermente di definizione rispetto agli altri due approcci. Quest'ultima osservazione è confermata anche dai risultati numerici citati precedentemente.

Magalotti	Instant-NGP			Nerfacto		
	Drone	COLMAP	Metashape	Drone	COLMAP	Metashape
PSNR	19.99	21.33	21.10	19.38	19.12	18.90
SSIM	0.508	0.550	0.552	0.501	0.490	0.483
LPIPS	0.589	0.525	0.530	0.521	0.521	0.537

Tabella 4.6: Risultati dell'esperimento con confronto tra Instant-NGP e Nerfacto nei dati estratti dal dataset Magalotti con pose ricavate dall'estrazione dei parametri di volo del drone, da COLMAP e da Agisoft Metashape.

San Ginesio	Instant-NGP			Nerfacto		
	Drone	COLMAP	Metashape	Drone	COLMAP	Metashape
PSNR	10.27	19.90	17.25	9.95	18.03	16.18
SSIM	0.407	0.594	0.573	0.399	0.505	0.500
LPIPS	0.997	0.543	0.550	0.877	0.626	0.632

Tabella 4.7: Risultati dell'esperimento con confronto tra Instant-NGP e Nerfacto nei dati estratti dal dataset San Ginesio con pose ricavate dall'estrazione dei parametri di volo del drone, da COLMAP e da Agisoft Metashape.

Le osservazioni fatte per le prime due scene si riflettono anche sulla terza, San Ginesio: sebbene in questo caso si abbiano a disposizione i dati di puntamento del drone e orientamento della camera, purtroppo questi sembrano essere troppo rumorosi. Le scene risultanti basate sia su Instant-NGP che su Nerfacto sono fortemente rumorose, il che ne consegue che anche le metriche che le descrivono siano inferiori rispetto a quando vengono applicati sistemi di SfM. Vengono generati quindi campi di radianza che riflettono le stesse caratteristiche di quanto era stato descritto per lo stesso caso per il dataset Macereto. Vi sono poche nuvole consistenti l'una con l'altra, forti deformazioni e rumore generico quasi totalmente diffuso.

Per quanto riguarda invece i risultati delle scene da SfM, la metrica PSNR tende ad essere maggiore con gli approcci COLMAP rispetto a Metashape, sebbene quelle di similarità siano paragonabili. Anche in questo caso si fa riferimento a quanto detto precedentemente sull'ipotesi di maggior efficacia di COLMAP nell'individuazione degli angoli e quindi nella resa delle diverse superfici della scena. Con l'uso di entrambe le architetture i fenomeni di aliasing sono fortemente limitati: le ringhiere che è possibile identificare nella scena vengono rese con buona qualità in entrambi i casi. Ulteriori riflessioni ricalcherebbero quanto già detto per i dataset precedenti: il fatto che molte delle osservazioni fatte siano riscontrabili anche in questo test validano maggiormente le ipotesi fatte precedentemente riguardo la saturazione degli strumenti di miglioramento di Nerfacto e le ottime capacità di resa di Instant-NGP, al costo di maggiori esigenze nel calcolo.

Figura 4.11: San Ginesio: estrazione dello stesso frame senza crop dello sfondo, a sinistra, con crop dello sfondo, al centro, in densità rispetto al crop dello sfondo a destra. Il dettaglio parzialmente sfocato rispetto allo sfondo è costante e dato dal fatto che, essendo la scena molto ampia, il sistema di bounding di Nerfacto concentra la generazione di dettagli sullo sfondo. Se la scena fosse stata definita bounded, si avrebbe comportamento inverso.

Figura 4.12: San Ginesio: estrazione dello stesso frame con crop a sinistra e in profondità rispetto al crop a destra. Anche qui la statua risulta sfocata, ma vi è alto dettaglio nella profondità della figura.

Le maggiori esigenze computazionali di una delle due architetture permettono quindi anche il confronto di quanto riportato con architetture Nerfacto-Big. Queste, oltre ad includere un maggior numero di parametri a causa delle reti MLP scelte di dimensioni maggiori, richiedono un numero di step più consistente per completarne l'addestramento. Visto che il tempo di training aumenta considerevolmente e con queste premesse si aggira intorno ad 1h30', l'esperimento non è stato ripetuto per le scene Drone che hanno riportato rumore troppo consistente. Si è quindi scelto di ripetere il tutto con il dataset con dati derivati dai dati di volo solo nel caso Magalotti. Questo poiché quanto citato nei paragrafi precedenti fa notare che sia l'unico caso in cui si ottenga comunque una scena di qualità. Nei risultati, riportati nelle Tab.4.8-4.9-4.10 è possibile notare come, sebbene il PSNR tenda ad essere migliore rispetto ai casi di addestramento con Nerfacto standard, le metriche di similarità sono più vicine. L'analisi visuale fatta sui video render non evidenzia particolari miglioramenti dati dall'aumento di parametri. Questo è probabilmente riconducibile al fatto che lo scopo con la quale è stata ingegnerizzata questa variante dell'architettura non sta nel rappresentare i dettagli della scena con maggiore qualità, bensì rappresentare scene più ampie. Se ne conclude che le scene trattate, sebbene riprendano spazi all'aperto decisamente grandi, subendo una scalatura interna, non generano risultati diversi da

Macereto	Nerfacto-Big	
	COLMAP	Metashape
PSNR	22.87	22.95
SSIM	0.680	0.686
LPIPS	0.442	0.438

Tabella 4.8: Risultati dell'esperimento su Nerfacto-Big nei dati estratti dal dataset Macereto.

Magalotti	Nerfacto-Big		
	Drone	COLMAP	Metashape
PSNR	19.24	19.14	19.25
SSIM	0.489	0.488	0.488
LPIPS	0.583	0.582	0.585

Tabella 4.9: Risultati dell'esperimento su Nerfacto-Big nei dati estratti dal dataset Magalotti.

NeRF di oggetti o scene che, nella realtà, sono spazialmente più piccole. Il confronto con Instant-NGP riporta le stesse osservazioni legate alla qualità ancora comparabile sebbene fossero attese delle migliorie. La riflessione sulle architetture Nerfacto-Big può essere chiusa facendo notare la quasi totale sovrapposizione di resa tra i due tool di SfM nei casi di Macereto, Magalotti e San Ginesio, soprattutto per quanto riguarda le metriche di similarità.

La serie di esperimenti fatti nel Cultural Heritage si ritiene quindi valida e i risultati visivi ne sono una prova. I tool utilizzati si adattano molto bene al task e riescono, in particolare, a rendere con fine dettaglio anche i materiali e i dettagli più fini come i mattoni, le pietre, gli alberi di sfondo e le incisioni sulla roccia. Il test riporta anche come uno strumento come Nerfacto riesca, con spese minime in termini di capacità computazionale, hardware richiesto e soprattutto complessità di acquisizione e postprocessing, a rendere scene con estrema fedeltà visiva: i risultati ottenuti in così poco tempo sono fortemente distanti da strumenti più classici quali il

San Ginesio	Nerfacto-Big	
	COLMAP	Metashape
PSNR	19.52	16.80
SSIM	0.552	0.532
LPIPS	0.593	0.604

Tabella 4.10: Risultati dell'esperimento su Nerfacto-Big nei dati estratti dal dataset San Ginesio.

design di mesh o la progettazione CAD. Assunta quindi l'acquisizione delle immagini, fare il pre-processing, la generazione del campo di radianza e il post-processing è ascrivibile a qualche ora se si sceglie una pipeline come quelle descritte, considerando un numero di immagini simile a quanto riportato. Il che consente un utilizzo di questi strumenti a tutto tondo e, citando la capacità di esportare diversi parametri quali il campo di densità spaziale, point cloud o mesh, possono assumere un ruolo sempre più importante nell'ambito della preservazione del patrimonio.

Capitolo 5

Conclusioni e sviluppi futuri

5.1 Conclusioni

Data una prima indagine e dati i confronti delle soluzioni NeRF rispetto allo stato dell'arte precedente alla loro introduzione, è già chiaro fin da subito quanto le capacità di sintesi di strumenti di questo tipo possano portare a grande innovazione nel campo della Computer Graphics e del Neural Rendering. Il tema principale è ovviamente riconducibile alla sintesi di scene tridimensionali a partire da ambienti reali. Questo porta gli strumenti neurali ad avere forti ed ampissime implicazioni in tutto ciò che ha a che fare con il design e la resa di oggetti in ambito digitale: si passa dalla sintesi e la progettazione di modelli 3D per la produzione industriale, all'ingegneria edile e civile, alla Computer Graphics per il cinema o per il marketing e ovviamente ad altri esempi non forniti in cui il design 3D la fa da padrone. In qualche modo, è possibile introdurre pipeline NeRF ogni volta che ci si trova in un ambiente di progettazione grafica per risolvere uno dei tantissimi task attinenti ad argomenti appena citati e non soltanto. E nonostante l'ambito applicativo sia davvero esteso, questo non significa che allo stato attuale della tecnologia questa possa essere pronta a superare diverse delle tantissime sfide che è possibile porvi. Questo lavoro di tesi ha proprio questa idea come indagine di fondo: ci si chiede quanto e come le soluzioni riportate ed ampiamente descritte possano essere applicate all'esterno di approcci che attualmente sono strettamente confinati e legati alla ricerca. Allo stato attuale legato alla scrittura di questo documento, si contano sulle dita di una mano gli approcci che veramente sfruttano le capacità delle architetture NeRF e le associano ad una più generica progettazione grafica. Oltre alla suite Nerfstudio già citata, che però richiede buona destrezza d'uso, sono veramente poche le soluzioni che lavorano con questi oggetti software e, quando lo fanno, incontrano alcune difficoltà parzialmente dovute a diverse esigenze computazionali ed ingegneristiche che non sono sempre di semplice risoluzione. Le principali problematiche, come già ben documentato, sono legate alla richiesta di hardware di discreto livello e alle elevate esigenze computazionali sia in

fase di generazione di questi modelli tridimensionali che in fase di esplorazione e visualizzazione del risultato.

Nonostante ciò, nello span temporale che tocca questo lavoro, dal momento iniziale e parzialmente inesperto del primo approccio, alla più matura analisi finale svolta con gran cognizione di causa, è possibile notare più di una discreta e sostanziale evoluzione dello strumento. Repository utilizzati inizialmente, quali torch-ngp[32] e Instant-NGP[20] sebbene fossero in pieno sviluppo nei primi istanti di ricerca e preparazione, sono, allo stato attuale, oramai soltanto aggiornati per fornire compatibilità a software e driver più recenti. La suite Nerfstudio, che inizialmente consentiva di utilizzare un numero di quattro diverse implementazioni in versione stabile, ora consente l'utilizzo, anche con plug-in esterni, di almeno una ventina di diverse soluzioni, la maggior parte di queste dedicate alla risoluzione di un task specifico di cui, inizialmente, non c'era traccia nemmeno negli sviluppi futuri degli articoli di ricerca più ambiziosi. Come già citato in diverse occasioni, la spinta tecnologica e di ricerca che vi è dietro questi strumenti rimane consistente e si inizia a perdere il conto dei singoli spunti tecnologici che approdano e aggiornano continuamente lo stato dell'arte.

Nel tentativo di validare, però, quelle tecnologie che più abbiano contribuito alla ricerca di settore, questo lavoro conferma l'importanza di quelli che potremmo definire tre piloni principali su cui grandissima parte della trattazione scientifica si basa al momento della stesura di quest'ultimo capitolo. L'articolo principale che cita per primo i campi NeRF[2] è sicuramente uno di questi: tutt'ora, a più di due anni dalla pubblicazione dell'articolo, in diverse repository e in entrambe quelle utilizzate, esiste codice software utilizzato e puntualmente riportato della pubblicazione originale; il dataset Synthetic, definito a volte Blender dataset, introdotto nell'articolo, non è soltanto immagine e logo della community di ricercatori che ad oggi vi lavora, ma anche il principale benchmark utilizzato di tutte le nuove soluzioni. Il secondo pilastro, quello che veramente ha reso disponibile queste soluzioni ad una platea mondiale di ricercatori decisamente ampia, è invece la pubblicazione legata ad Instant-NGP[20]. L'encoder Multiresolution Hashgrid Encoder, oltre ad introdurre una soluzione decisamente interessante per altri ambiti, rivoluziona completamente il tempo di addestramento di questi strumenti e ne aumenta in maniera decisamente esponenziale l'usabilità. Allo stato attuale, con una macchina di media gamma, è possibile generare una mesh di un oggetto presente sulla scrivania dell'utilizzatore in meno di un'ora di calcolo computazionale. Questo, oltre ad aumentare l'interesse nello strumento, è carburante per la ricerca in quanto semplifica enormemente i tempi di sviluppo ed approfondimento. L'ultimo pilone, sempre più uno standard de facto delle spinte di ricerca, è proprio Nerfstudio. I nuovissimi approcci che nascono in ambito NeRF, spesso anche non citati in questo lavoro, sfruttano sempre più le componenti software oramai standardizzate dal progetto. Repository che, allo stato attuale, include come sviluppatori attivi quasi la totalità dei ricercatori

autori delle architetture citate. Come mostrato, per di più, lo strumento sebbene pecchi di disponibilità d'uso anche per coloro senza particolari capacità di settore, è decisamente completo e definisce senza sforzi un'intera pipeline, a partire dalla semplice ideazione del modello, all'export di render pronti per essere caricati nel web o utilizzati con altri software di progettazione.

Inoltre, per non complicare ulteriormente la trattazione, a fondare una ancora migliore base qualitativa della ricerca in questo ambito ci sono diversissime soluzioni che riescono ad includere in questo tipo di pipeline diversi concetti chiave dell'informatica attuale e del campo dell'AI: NeRF che generano immagini in base a quanto definito in un prompt con strumenti di Latent Diffusion; campi collegati a Transformer Generativi in grado di catturare l'informazione specifica sul loro contenuto a partire dall'analisi di una breve descrizione concettuale; strumenti in grado di sostituire la radianza nel visibile ad altri fenomeni fisici di emissione quali l'analisi visuale dei segnali radio o di onde elettromagnetiche generiche. Quanto vi è da fare in questo campo è sicuramente molto e ci si aspetta che l'esplosione di ricerche a riguardo continui ancora per diverso tempo.

Entrando più nel dettaglio di questo lavoro, i risultati ottenuti provano quanto questi strumenti possano essere importanti in diversi campi. Oltre a giustificarne parte delle caratteristiche e a fornire spunti di riflessione riguardo le loro implementazioni, soprattutto per quanto riguarda le scelte legate agli encoder utilizzati e alla loro architettura, è stato provato come sia possibile definire una potente pipeline nell'ambito della preservazione del patrimonio culturale. Dai risultati ottenuti infatti si definisce come, anche a basso costo, sia possibile utilizzare strumenti di Structure from Motion per impostare un dataset su cui un'architettura Nerfacto o Instant-NGP possa lavorare per fornire ottimi risultati. La scena così ottenuta può facilmente sostituire quelle ottenute con rilevazioni attualmente molto più complesse e costose che necessitano hardware specifico quali tracciatori laser: è possibile muoversi all'interno di una piazza con uno smartphone e, dopo un calcolo opportuno con NeRF, ottenere dati di profondità, aspetto e struttura visitabili e consultabili nel giro di pochissimo tempo. Quanto questo lavoro vuole provare però è anche legato alla qualità e output dei risultati ottenuti che sicuramente può avvicinare queste tecniche, tra cui quelle di fotogrammetria, a dei risultati di riproduzione della realtà decisamente notevoli.

Sebbene fosse prevedibile, gli approcci studiati risultano essere estremamente ottimizzati e, a meno di rivoluzioni interne del metodo utilizzato per la rappresentazione del campo di radianza o dei moduli software che compongono le architetture più diffuse, risulta essere molto difficile affinare le capacità rappresentative ottenute da questi modelli di strumenti. Si fa notare, inoltre, quanto sia comunque possibile approcciarsi a questi metodi semplicemente e, allo stesso tempo, ottenere soluzioni meno legate alla ricerca e sviluppo e più vicini ad un'implementazione di produzione:

le basi software di pacchetti di Machine Learning e gestione dei tensori GPU sui quali queste architetture si basano supportano in maniera totale diverse soluzioni meno adibite alla ricerca e considerate più solide quali l'utilizzo di linguaggi di programmazione strettamente efficienti come il C++/CUDA o il C#.

La problematica della generazione di output completamente usabili rimane comunque uno dei temi principali riguardo alle difficoltà di approccio degli strumenti NeRF. In questa ricerca è stato fatto notare come alcuni dei più promettenti approcci sviluppati negli ultimi anni non siano poi efficaci nella risoluzione della generazione di output in formato mesh. Deep Marching Tetrahedra risulta, oltre che una soluzione ostica dovuta alla difficoltà nel reperimento del codice software che ne implementa correttamente l'algoritmo alla base, uno strumento non all'altezza dei metodi quali Marching Cubes o superfici di Poisson che, nel caso di quest'ultime implementate nella suite Nerfstudio, risultano essere il massimo della qualità ottenibile nel caso riportato. Una comune assenza di soluzioni di questo tipo sottolinea però la difficoltà ingegneristica che risiede alla base di questo task.

Avendone analizzato attentamente e a diverse riprese l'architettura e le capacità, si conclude che a meno di esigenze non trattate in questo lavoro, Instant-NGP utilizzata come architettura per campi NeRF nella preservazione del patrimonio sia probabilmente la punta di diamante degli strumenti di digitalizzazione a disposizione. Quanto appena affermato è supportato dalla costanza con cui la soluzione si è rivelata la migliore sotto diversi aspetti di valutazione qualitativa o, nel singolo caso in cui non lo fosse, strettamente paragonabile al miglior risultato ottenuto. D'altra parte, alcune note complessità implementative di driver, hardware e manipolazioni richieste suggeriscono un parigrado approccio basato su Nerfacto che risulta essere sempre più semplice ed immediato nell'utilizzo, mantenendo allo stesso tempo degli standard qualitativi molto elevati. Lo stesso discorso vale anche nel caso in cui i dataset a disposizione non siano immagini di scene strettamente unbounded e ampie come quelle prese nei casi del Cultural Heritage di questo lavoro, ma anche dataset di oggetti più con scenari più stretti, piccoli e più vicini alla vita di tutti i giorni in cui non si ha sempre a disposizione un drone con cui acquisire diversi scatti di alta qualità. Quanto quindi affermato per la buona equivalenza delle due architetture testate perde leggermente: Nerfacto risulta essere un approccio molto più elastico a diverse problematiche di bounding della scena e, sebbene in questo lavoro non siano particolarmente emersi numericamente, i tweak che possiede possono fare la differenza nella resa dei piccoli dettagli.

L'analisi portata avanti non tocca soltanto le architetture software utilizzate per la risoluzione dei vari task citati, bensì anche molte delle problematiche legate alla scelta di uno o più elementi della pipeline. In particolare, nel caso in cui si ritenga necessario utilizzare strumenti dal codice sorgente aperto per ottenere risultati che

mantengano sicuramente una delle più alte qualità raggiungibili, COLMAP per lo SfM è la soluzione ideale. Allo stesso tempo, nel caso in cui sia necessario utilizzare il campo NeRF prodotto come strumento per la lavorazione con precisione geografica, ad esempio nel caso in cui ci si trovi a dover risolvere un task che richiede che la scena prodotta sia correttamente geolocalizzata nello spazio anche in base al suo orientamento,che non sia deformata e che sia richiesto sia processata in un tempo relativamente corto, la miglior soluzione percorribile è quella di fornire alla suite Nerfstudio un dataset con le pose delle camere preventivamente generate con Agisoft Metashape. Il tool, oltre a fornire la possibilità di mantenere orientamento e geolocalizzazione delle immagini, e quindi della scena generata, consente anche un approccio più rapido e una più facile gestione di problemi che possono sorgere riguardo all'allineamento degli scatti. Nonostante i due approcci siano sicuramente notevoli, la soluzione che si candida ad essere migliore nella risoluzione del task, ma che allo stesso tempo richiede una serie di accorgimenti specifici che potrebbero portar il processo al di fuori di quanto vi è richiesto, è l'utilizzo di scatti tracciati e controllati già nel mondo reale. Si è visto come, nel caso Magalotti, un'alta ma non estrema precisione dei dati a disposizione ha portato a risultati paragonabili con quelli ottenuti a partire dallo SfM. Se ne conclude che, se si ha a disposizione strumenti molto più vicini alla fotogrammetria classica, quindi tracciamento nello spazio delle coordinate della camera, ciò può portare a risultati ovviamente migliori.

L'obiettivo contributivo di questo lavoro, oltre a fornire al lettore una base teorica stretta e pratica degli approcci NeRF e a presentarne uno studio intenso relativo alle scelte modulari e architetturali, è quello della definizione della miglior pipeline attuale a partire dai dati presentati e con le esigenze richieste da un'ipotesi di preservazione culturale di quanto oggetto delle immagini a disposizione. Nei termini definiti, si ritiene che le varie ipotesi appena riportate, considerando tutte le assunzioni fatte riguardo i contenuti presentati, siano sufficienti a dichiarare l'obiettivo raggiunto. Sono state infatti toccate condizioni diverse, con ampi o più ristretti dati a disposizione, cercando di tenere da conto anche le capacità computazionali impiegate ed impiegabili. Sono stati inoltre raggiunti alti livelli qualitativi da parte delle rappresentazioni tridimensionali, a validare ancora di più gli studi informatico-architetturali e le pipeline impiegate.

L'ultima conclusione riporta come, un approccio ingegneristico che può basarsi sia sui campi NeRF che sulle soluzioni più classiche di fotogrammetria può essere quanto di più conveniente sia suggerito nel caso in cui una alta qualità, sia delle superfici che dei dati a disposizione, sia fortemente richiesta. La soluzione pratica data dal connubio di tutti questi strumenti può risultare nella miglior scelta possibile.

5.2 Sviluppi futuri

Data l'estensione della trattazione, diversi sono i temi che sarebbe stato interessante approfondire maggiormente, sebbene si sia proceduto con l'obiettivo di massima attenzione. L'argomento è inoltre in costate divenire, perciò qualsiasi spunto sarà fornito in questa sezione è strettamente legato al fatto che andrà sicuramente correlato ad un attento lavoro di ricerca: come è già stato citato all'interno di questo lavoro, durante la stesura di questo documento la ricerca nei campi di radianza neurale si trova in un periodo molto florido e non passa paio di mesi in cui una piccola ma sostanziale innovazione non venga introdotta.

Seguendo invece quanto citato all'interno di questa trattazione, uno dei primi spunti che sarebbe interessante approfondire è strettamente legato a quanto svolto nel contesto della modifica all'architettura. È infatti probabile che ulteriori ricerche nel contesto di una architettura neurale che soddisfi in maniera più piena le esigenze NeRF siano legate ad approfondimenti nel campo delle reti Multi-Layer Perceptron. È stato notato a più riprese quanto le caratteristiche di questi strumenti siano efficienti ed efficaci nel contesto NeRF, ma non si esclude che architetture più pratiche e complesse possano raggiungere risultati quanto meno equivalenti. Ciò richiederebbe però spunti teorici che giustifichino la soluzione. Allo stesso modo, interessante sarebbe valutare in maniera più completa, a livello di altre implementazioni e con dataset diversi, la soluzione con Leaky-ReLU di Instant-NGP. È stato provato quanto sia comparabile alla soluzione standard e sarebbe interessante prevedere un esperimento che riesca a validare la miglior efficacia di questa tipologia di attivatore associato ad una funzione di filtraggio opportuna. In questo modo si potrebbe contribuire a risolvere con modifiche sufficientemente rapide e pratiche una delle problematiche principali di questa tipologia di architetture. Riuscire a risolvere, infatti, anche parzialmente la problematica di densità volumetrica della point cloud a livello di architettura, potrebbe contribuire molto a risolvere altri problemi a cascata, soprattutto quelli riguardo la qualità delle mesh generabili da NeRF.

Infine, un approccio molto più controllato delle acquisizioni orientate al Cultural Heritage, potrebbe fornire la base pratica per un test oggettivo di confronto tra i metodi di Structure from Motion. Riuscire a definire in maniera più puntuale di quanto svolto in questo lavoro quanto effettivamente sia necessario a livello di accorgimenti per superare un approccio modulare che include anche un algoritmo di ricerca delle pose potrebbe aiutare a definire una migliore pipeline e quindi uno standard per questo tipo di obiettivi. Inoltre, una maggior attenzione a questa tipologia di esperimenti potrebbe portare alla definizione di un insieme di parametri ideali per spingere ancor di più sui risultati qualitativi della scena.

Ringraziamenti

Ringrazio infinitamente il Chia.mo Relatore Professor Primo Zingaretti, che oltre ad aver creduto in questo progetto ancor prima che avessi modo di pensarlo o definirlo, ha saputo offrire tantissimo in termini di spinta metodologica, formativa e di crescita personale sia all'interno che all'esterno di questo lavoro di tesi. Inoltre, è stato un esperto supporto e un grande ascoltatore in ognuna delle più complesse fasi sperimentali.

Ringrazio altrettanto il Dottor Emanuele Balloni, che non soltanto è ispiratore del tema e stretto collaboratore nella fase di design della quasi totalità dei passi svolti, ma anche metafora di un profondo e attento consiglio per il futuro.

Ringrazio inoltre il Professor Adriano Mancini e con lui il gruppo di ricerca Vision, Robotics and Artificial Intelligence (VRAI) per avermi messo a disposizione un gran numero di consigli, i dataset e le macchine su cui lavorare.

Rigrazio i colleghi David e Davide, dopo tanti anni ancora valido supporto e forti valvole di sfogo.

Ringrazio la mia famiglia e i miei amici, costanti di pace nelle avversità.

Ringrazio infine Elena per essere musa e speranza, sempre forte ed inesorabile come l'alba di un nuovo giorno.

Fabriano, Giugno 2023

David Caprari

Bibliografia

[1] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Yifan Wang, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, Tomas Simon, Christian Theobalt, Matthias Niessner, Jonathan T. Barron, Gordon Wetzstein, Michael Zollhoefer, and Vladislav Golyanik. Advances in neural rendering, 2022.

[2] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[3] Kevin Taylor-Sakyi. Big data: Understanding big data, 2016.

[4] Algorithm - Wikipedia — en.wikipedia.org. `https://en.wikipedia.org/wiki/Algorithm`. [Accessed 11-Jun-2023].

[5] Machine learning - Wikipedia — en.wikipedia.org. `https://en.wikipedia.org/wiki/Machine_learning`. [Accessed 11-Jun-2023].

[6] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[7] Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[8] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks, 2017.

[9] Diganta Misra. Mish: A self regularized non-monotonic activation function, 2020.

[10] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network, 2015.

[11] Computer vision - Wikipedia — en.wikipedia.org. `https://en.wikipedia.org/wiki/Computer_vision`. [Accessed 11-Jun-2023].

[12] Deepanshu Tyagi. Introduction To Feature Detection And Matching — medium.com. `https://medium.com/data-breach/introduction-to-feature-detection-and-matching-65e27179885d`. [Accessed 11-Jun-2023].

[13] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[15] Arthur Appel. Some techniques for shading machine renderings of solids. In *Proceedings of the April 30–May 2, 1968, spring joint computer conference*, pages 37–45, 1968.

[16] Cindy M Goral, Kenneth E Torrance, Donald P Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. *ACM SIGGRAPH computer graphics*, 18(3):213–222, 1984.

[17] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[19] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984.

[20] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022.

[21] Matthias Teschner, Bruno Heidelberger, Matthias Müller, Danat Pomerantes, and Markus H Gross. Optimized spatial hashing for collision detection of

deformable objects. In *Vmv*, volume 3, pages 47–54, 2003.

[22] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[23] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022.

[24] Kyle Gao, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review, 2023.

[25] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021.

[26] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. *arXiv preprint arXiv:1707.05776*, 2017.

[27] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.

[28] Jonas Kulhanek and Torsten Sattler. Tetra-NeRF: Representing neural radiance fields using tetrahedra. *arXiv preprint arXiv:2304.09987*, 2023.

[29] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. 2023.

[30] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 2023.

[31] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. *arXiv preprint*

arXiv:2303.09553, 2023.

[32] Jiaxiang Tang. Torch-ngp: a pytorch implementation of instant-ngp, 2022. https://github.com/ashawkey/torch-ngp.

[33] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.

[34] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, 2023.

[35] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021.

[36] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

[37] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019.

[38] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019.

[39] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[40] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[42] Clement Fuji Tsang, Maria Shugrina, Jean Francois Lafleche, Towaki Takikawa, Jiehan Wang, Charles Loop, Wenzheng Chen, Krishna Murthy Jatavallabhula, Edward Smith, Artem Rozantsev, Or Perel, Tianchang Shen, Jun Gao, Sanja Fidler, Gavriel State, Jason Gorski, Tommy Xiang, Jianing Li, Michael Li, and Rev Lebaredian. Kaolin: A pytorch library for accelerating 3d deep learning research. `https://github.com/NVIDIAGameWorks/kaolin`, 2022.