ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

TESI DI LAUREA

in
Sistemi Digitali M

Monitoraggio del distanziamento sociale mediante singola telecamera

CANDIDATO
Alessio Mingozzi

RELATORE:
Prof. Stefano Mattoccia

CORRELATORE/CORRELATORI
Dott. Filippo Aleotti
Dott. Matteo Poggi
Dott. Fabio Tosi

Anno Accademico 2019/20

Sessione III

A causa della pandemia da COVID-19, si è reso sempre più necessario attuare nuove misure volte a preservare il distanziamento sociale. Per questo scopo la richiesta di sistemi tecnologici che permettano di monitorare tale norma in maniera automatica è sempre in aumento. Il presente progetto di tesi ha come scopo quello di studiare, realizzare e valutare un sistema per il monitoraggio del distanziamento sociale. Tale sistema verrà impiegato per misurare in tempo reale la distanza tra le persone all'interno della scena inquadrata e allertare in caso di possibile situazione di rischio. L'utilizzo di reti neurali che permettono di ottenere informazioni di profondità da una immagine monoculare è di particolare interesse. Grazie a tali informazioni è possibile ottenere le coordinate nello spazio 3D di qualsiasi punto nella scena, così da permetterne la misurazione della distanza. Verrà studiata l'applicabilità di queste reti in scenari di uso reale, in quanto oggetto di ricerca e interesse all'interno della comunità scientifica.

Indice

Capitolo 1 - Introduzione **5**

Capitolo 2 - Lavori Correlati **7**

Capitolo 3 - Strumenti di sviluppo **10**

3.1 OpenCV 10
3.2 Tensorflow 10
3.3 PyTorch 10
3.4 KITTI Dataset 10

Capitolo 4 - Descrizione del Sistema **11**

4.1 Individuazione delle persone 11
4.2 Posizionamento delle persone 13

Capitolo 5 - Calibrazione telecamera **15**

5.1 Modello Pinhole 15
5.2 Calibrazione 18
5.3 Calcolo delle coordinate X e Y 19

Capitolo 6 - Monodepth Neural Networks **21**

6.1 PyD-Net 22
6.2 MiDaS 23
6.3 Analisi e struttura delle mappe 23
6.3.1 Soluzioni proposte 28

Capitolo 7 - Messa in scala mappe monodepth **29**

7.1 Scala da singolo punto 29
7.1.1 Acquisizione distanza nota 30
7.1.2 Primi Risultati 31
7.2 Scala a punti multipli 32
7.2.1 Acquisizione distanze note 33
7.2.2 Primi Risultati 34

Capitolo 8 - Misurazione della distanza 36

Capitolo 9 - Un sistema alternativo 38

9.1 Omografia 38

9.2 Stima della trasformazione 39

9.3 Segmentazione del piano 41

9.4 Bird-Eye View 42

9.5 Proiezione e distanza tra le persone 42

Capitolo 10 - Risultati Sperimentali 44

Capitolo 11 - Conclusioni e sviluppi futuri 52

Bibliografia 53

Capitolo 1

Introduzione

La pandemia da Covid-19 ha generato la necessità di tenere comportamenti di tipo sociale diametralmente opposti rispetto a quelli a cui eravamo abituati fino ad oggi. Per ridurre la diffusione del virus è di fondamentale importanza indossare dispositivi di protezione individuale, come le mascherine, e seguire le norme di "distanziamento sociale" nelle situazioni più a rischio di assembramento. Seguire queste regole rende possibile spezzare la catena di contagi, la quale diventerebbe incontrollabile in maniera esponenziale. Ciò metterebbe in difficoltà le strutture socioeconomiche del paese in poco tempo. L'apporto che tali comportamenti hanno nella lotta contro la diffusione del virus è fondamentale. Nasce perciò il bisogno di sviluppare nuovi sistemi che permettano di controllare e verificare che questi siano rispettati. Risulta difficile, infatti, far rispettare queste regole nei luoghi della vita sociale impiegando un controllo preciso e capillare. Strumenti tecnologici che permettano di monitorare in maniera accurata e automatica se le distanze interpersonali vengano rispettate, risultano perciò essere di fondamentale importanza. Il progetto di questa tesi si inserisce all'interno dello sviluppo di un sistema di monitoraggio del distanziamento sociale. Nello specifico, ci si è occupati della progettazione del modulo per il calcolo della distanza tra le persone e l'integrazione dei vari moduli in un primo prototipo. Questo sistema, una volta installato e configurato, sarà in grado di monitorare la distanza relativa tra le persone individuate all'interno della scena inquadrata. Nel caso in cui il distanziamento sociale non venga rispettato, la situazione verrà notificata ad un addetto responsabile. Per poter misurare le distanze è necessario collocare le persone in un sistema di riferimento, il quale permette di conoscere le loro coordinate nello spazio. Solitamente, sistemi di questo tipo ottengono la coordinata legata alla profondità per mezzo di telecamere stereo o sensori attivi di profondità (LiDAR, ToF, ecc.). In alternativa ai sistemi visuali, molto diffusi ad oggi sono sistemi che si basano su di una combinazione tra dispositivi mobili e tecnologie GPS o Bluetooth. Uno dei più diffusi è l'applicazione IMMUNI, o altre simili a questa, la quale permette di tenere traccia dei contatti interpersonali per mezzo dell'applicazione installata sullo smartphone e della tecnologia Bluetooth. Una delle criticità di questi sistemi è che risulta necessaria la collaborazione da parte degli utenti perché possa essere efficace. Se nessuno installasse l'applicazione, o non venisse mantenuto attivo il Bluetooth, il sistema non funzionerebbe. Inoltre, in caso si sfruttasse il GPS, esso non funziona bene negli spazi chiusi. I sistemi basati sulla visione devono comunque affrontare problematiche comuni a queste ultime (e.g. garantire

la privacy delle persone), ma permettono di svincolarsi dalla necessità della collaborazione da parte delle persone. In particolare, il sistema qui presentato adotta una soluzione che tenta di ridurre la complessità del dispositivo hardware, cercando comunque di mantenere l'efficacia nel monitoraggio del distanziamento sociale: la profondità viene ottenuta per mezzo di reti neurali specificatamente addestrate, chiamate reti neurali *monodepth*, le quali permettono di stimare la profondità dei punti della scena inquadrata basandosi su una singola immagine, potendo pertanto essere utilizzata in innumerevoli contesti applicativi già operativi e basati su una singola camera, e non due immagini stereo raramente adottate in applicazioni di sorveglianza. Poiché la pandemia ha inficiato in maniera forte sulle capacità di spesa delle aziende, un sistema di questo genere permetterebbe una maggior facilità di installazione e un costo ridotto, poiché potrebbe essere facilmente integrato all'interno di sistemi di videosorveglianza già presenti nei luoghi più a rischio di assembramento, come supermercati, piazze, stazioni, ecc. Le reti neurali monodepth sono una tecnologia di grande interesse all'interno della comunità scientifica e in fase di studio, ma raramente vengono utilizzate all'interno di sistemi reali o commerciali. Nel corso dell'elaborato verranno analizzate nel dettaglio due diverse reti e verrà discussa la loro efficacia una volta applicate all'interno di uno scenario reale.

Il progetto, svolto nel contesto di un progetto attualmente in corso con l'azienda Cloudif.ai (https://cloudif.ai/) di Bologna, è stato realizzato in collaborazione con altri quattro studenti, Enzo Famà, Francesco Olivo, Lorenzo Righi e Niccolò Rosadi, i quali si sono occupati rispettivamente del modulo di estrazione features [1], visualizzazione delle mappe 3D, filtraggio di traiettorie [2] e individuazione delle persone [3].

Figura 1.1 – In uno scenario come quello mostrato in figura, l'obiettivo è individuare situazioni di pericolo assembramento andando a monitorare in tempo reale la posizione e la distanza tra le persone all'interno della scena.

Capitolo 2

Lavori Correlati

Monitorare il distanziamento sociale per contrastare la pandemia è una necessità ad oggi globale. Questo ha portato ad una crescita esponenziale di sistemi di questo tipo, sia all'interno del mondo accademico, nel quale diversi articoli sono stati recentemente pubblicati, sia all'interno del mondo aziendale, in cui alcuni hanno iniziato a proporre soluzioni di tipo commerciale. Il sistema proposto si va ad inserire nella categoria dei sistemi di tipo visuale, i quali sfruttano telecamere o altri strumenti di tipo visivo per ottenere informazioni relative alla scena. Focalizzandoci sul mondo accademico, diverse proposte sono state realizzate, le quali condividono alcuni aspetti chiave con la soluzione proposta. Uno è la necessità di individuare le persone e/o la posa del corpo. Un altro è sfruttare informazioni geometriche della scena per ottenere un riferimento metrico con cui poter calcolare la distanza interpersonale. Una prima trattazione teorica del problema del distanziamento sociale in forma visuale è stata proposta da Cristiani et al [4], i quali descrivono, dal punto di vista teorico, quali caratteristiche e principi debba avere un sistema di questo tipo. Viene inoltre analizzato il concetto di "distanza", dal punto di vista della *prossemica*, visto non come misura dello spazio tra le persone ma come dimensione di uno spazio "interpersonale" che circonda ogni singola persona. Si vanno così a definire una serie di parametri chiave che vanno a comporre un sistema per il monitoraggio del distanziamento sociale. Su di questo fonda le basi il sistema sviluppato da Aghaei et al [5], il quale utilizza i concetti chiave descritti da [4] per costruire un sistema al fine di monitorare il distanziamento sociale. Anch'essi, partendo dal presupposto che integrare il sistema all'interno dell'infrastruttura di videosorveglianza già presente sia più conveniente, rispetto ad installarne di nuovi, vanno a realizzare un sistema che mira a ottenere una misura della distanza tra le persone a partire da una immagine monoculare. L'approccio utilizzato si basa sul calcolo di un'omografia, ovvero una trasformazione prospettica tra due piani. In questo caso tra il piano della scena e quella immagine. Si ottiene così una vista dall'alto, detta *bird-eye view*, e un sistema di riferimento in cui posizionare le persone all'interno della scena. L'omografia viene stimata sfruttando le caratteristiche geometriche della scena, a partire dalla inclinazione della telecamera rispetto al terreno. In questo modo è possibile ottenere tale trasformazione per qualsiasi scenario nel quale non sia possibile effettuare una calibrazione della telecamera in uso. Per ottenere una misura metrica della distanza, l'approccio scelto si basa su di una stima delle dimensioni medie del corpo umano. Utilizzando una rete di stima della posa (OpenPose) si

ottengono le coordinate delle giunture del corpo delle persone all'interno della scena. Associando alle misure in pixel le misure metriche medie delle varie parti del corpo, come torso, spalle, braccia o gambe, è possibile ottenere un riferimento metrico per poi andare a misurare la distanza tra due persone. Così facendo il problema del distanziamento assoluto viene ridotto ad un problema locale. Nonostante sia sicuramente vantaggioso evitare la necessità di una calibrazione della telecamera, tale scelta introduce una serie di limitazioni, ovvero persone con dimensioni del corpo molto diverse dalla media vengono distanziate in modo errato (ad esempio i bambini) e la maggior parte delle parti del corpo devono essere visibili, quindi qualsiasi tipo di occlusione, anche parziale, impedisce alla persona coperta di essere misurata.

Un altro sistema simile a quest'ultimo è Inter-Homines, realizzato da Fabbri et al [6]. Anch'esso sfrutta la stima di una omografia per poter ottenere un sistema di riferimento in cui posizionare le persone e calcolare le distanze. A differenza di [5], l'omografia viene stimata direttamente in scala metrica. Calcolare un'omografia in generale necessita conoscere le distanze tra almeno quattro punti presenti nella scena. Per fare ciò vengono utilizzati nove marker, facilmente identificabili, disposti a griglia. Questo pattern viene posizionato sul piano principale della scena in modo da poter calcolare la matrice che rappresenta la trasformazione. Si ottiene in questo modo un sistema di riferimento metrico, il quale permette di misurare la distanza tra le persone. Poiché, in questa situazione, la distanza può essere calcolata soltanto tra punti che giacciono sul piano, è necessario che il punto di appoggio, ovvero i piedi delle persone, si trovi su di esso. Il problema dell'occlusione si ripresenta anche in questo caso. Per ovviare a ciò è stata realizzata una rete neurale che permette di inferire quale sia la posizione dei piedi a partire da una persona individuata, anche parzialmente occlusa. Insieme al sistema di misurazione della distanza, Inter-Homines mette a disposizione una serie di indicatori e modelli che valutano il rischio di assembramento nell'area e ne assegnano un valore percentuale, oltre che un sistema di individuazione e oscuramento dei volti per motivi di privacy. Aggiungendo il costo di una fase di calibrazione iniziale, il sistema così proposto ha una maggiore robustezza alle occlusioni e non richiede l'individuazione della posa. Ciò nonostante, limite intrinseco di questo approccio, il piano deve essere ben visibile e completamente planare per poter restituire misure robuste.

Diverse altre proposte sono state sviluppate basate sulla stima di un'omografia per il calcolo delle distanze. Reazaei et al [7] si sono concentrati soprattutto sull'operazione di identificazione delle persone, andando a ottimizzare e riaddestrare la rete YOLO [8] in modo da renderla più robusta ad occlusioni e specializzata nella identificazione delle persone. Anche in questo caso la distanza tra le persone viene ottenuta per mezzo del calcolo di un'omografia. In aggiunta è

stato implementato un filtro di Kalman che permette di ottenere una stima delle traiettorie delle persone tracciate, così da avere a disposizione, anche in caso di occlusione, un punto di tracciamento con cui stimare la distanza. Saponara et al [9] introducono inoltre l'utilizzo di una telecamera termica, così da abbinare un controllo della temperatura corporea a quello del distanziamento sociale.

Sistemi visuali alternativi sono stati proposti: Sathyamoorthy et al [10] hanno realizzato un robot a guida autonoma con sensori attivi di profondità, come camere RGB-D e LIDAR, il quale riesce a identificare, per mezzo di tali, gruppi di persone. Se disponibile, può essere integrato da un sistema di telecamere di sorveglianza che, sfruttando un'omografia come visto in precedenza, permettono di ampliare l'area di controllo. Ahmed et al [11] invece utilizzano una prospettiva alternativa rispetto a quelle viste negli altri casi. Invece utilizzare immagini con prospettiva frontale, hanno utilizzato telecamere poste al di sopra della scena inquadrata. In questo modo non è necessario utilizzare informazioni della scena o calibrazioni per ottenere un'omografia. La distanza tra le persone individuate viene calcolata nell'immagine, senza alcuna informazione di tipo metrico. In questo caso la rete YOLO è stata riaddestrata adeguatamente per poter riconoscere le persone da immagini con prospettiva aerea.

Oltre a quelli analizzati, è giusto citare altri sistemi per il monitoraggio del distanziamento sociale, i quali non utilizzano sistemi di visione. Esempi sono quello realizzato da Rusli et al [12] o altri sistemi commerciali, i quali sfruttano applicazioni mobili insieme a tecnologie Bluetooth e GPS.

L'analisi di lavori simili a questo ha mostrato che la soluzione proposta è, ad oggi, unica nel suo genere, il che la rende una sfida complessa ma allo stesso tempo di grande interesse. Nei prossimi capitoli andremo ad analizzare la struttura del sistema proposto e le relative metodologie applicate.

Capitolo 3

Strumenti di sviluppo

In questo capitolo verranno descritti gli strumenti di sviluppo utilizzati. La descrizione non sarà esaustiva, poiché l'obiettivo è quello di dare una conoscenza sufficiente a comprendere le operazioni svolte. Per ulteriori approfondimenti si rimanda alla bibliografia. Il modulo è stato realizzato in linguaggio Python, testato nella versione 3.7 e 3.8. Sono state utilizzate due versioni differenti poiché la più recente non risultava compatibile con la versione 1.15 di Tensorflow, necessaria per il funzionamento della rete neurale monodepth PyD-Net [13].

3.1 OpenCV

OpenCV [14] è una libreria software open source per la computer vision e il machine learning. Essa ha più di 2500 algoritmi ottimizzati, che include un set completo di algoritmi classici e allo stato dell'arte. La versione utilizzata è la 4.2.0 per Python.

3.2 Tensorflow

TensorFlow [15] è una libreria software gratuita e open source per il machine learning. Può essere utilizzata in una serie di attività, ma ha una particolare attenzione al training a all'inferenza di deep neural network. Viene utilizzata nella rete PyD-Net, nella versione 1.15.

3.3 PyTorch

PyTorch [16] è una libreria open source di machine learning basata sulla libreria Torch, utilizzata per applicazioni come la visione artificiale e l'elaborazione del linguaggio naturale. La versione 1.7 è necessaria per il funzionamento di MiDaS [17], un'altra rete monodepth considerata nel corso di questo progetto.

3.4 KITTI Dataset

KITTI Vision Benchmark Suite [18] è un dataset, realizzato dal Karlsruhe Institute of Technology e il Toyota Technological Institute, contenente dati utilizzabili per applicazioni di visione stereo, optical flow, visual odometry, 3D object detection and 3D tracking. Per il presente caso di studio, dato il blocco degli spostamenti, risulta difficile realizzare delle immagini con annessi dati di profondità ottenuti da un sensore attivo. Avere perciò a disposizione immagini con annessi dati LIDAR permette di verificare con precisione l'affidabilità del sistema realizzato. Tra tutti quelli disponibili sono stati utilizzati due dataset in particolare: stereo 2015 [19] e il set 2011_09_28_drive_0039 presente nei raw data [20].

Capitolo 4

Descrizione del Sistema

Come anticipato in precedenza, l'obiettivo finale di questo progetto è quello di realizzare un sistema che permetta di misurare la distanza tra le persone in tempo reale utilizzando una singola telecamera, sfruttando le reti neurali monodepth per la stima della profondità. In Figura 4.1 possiamo vedere uno schema generale del funzionamento a runtime del sistema. Le immagini acquisite dalla telecamera vengono inviate al modulo per l'individuazione delle persone e al modulo per il calcolo della profondità. Successivamente, conoscendo la posizione delle persone individuate all'interno della scena inquadrata, è possibile andare a misurare la distanza tra di esse.

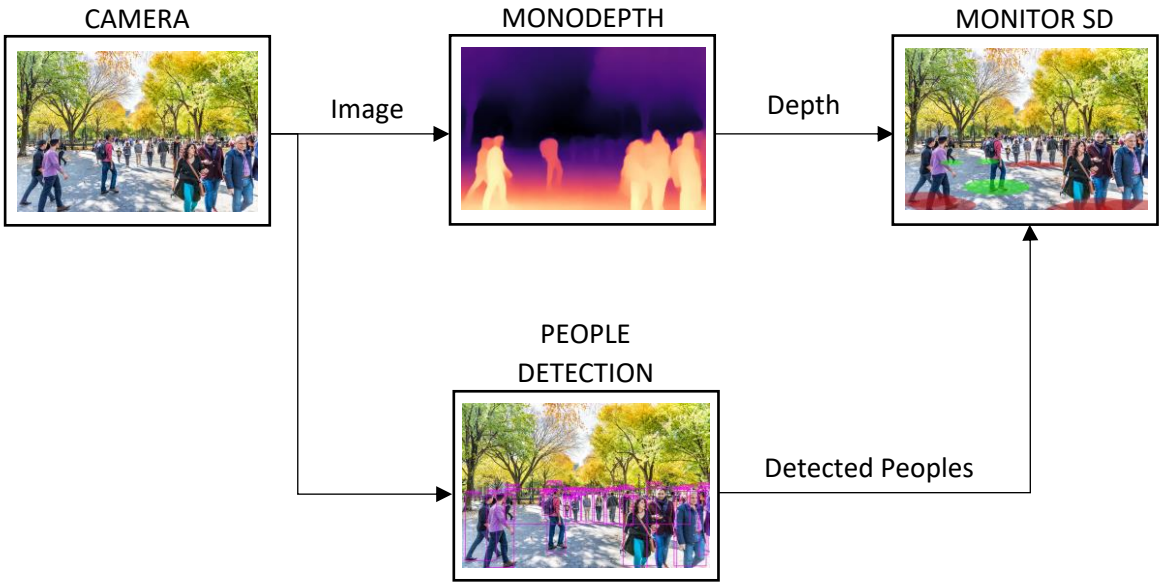

Figura 4.1 – Schema generale del sistema a runtime.

Analizzando il problema nel dettaglio, si può evincere che i punti chiave che vanno a delinearsi sono due:

- L'individuazione delle persone.
- Il posizionamento delle persone.

4.1 Individuazione delle persone

Il modulo *people_detection,* che implementa l'identificazione delle persone all'interno della scena, è stato realizzato da Niccolò Rosadi [3].

Esso sfrutta una rete di real-time object detection allo stato dell'arte, chiamata YOLO. Si tratta di una rete che permette di individuare diverse classi di oggetti all'interno delle immagini analizzate. La rete è stata modificata in modo da considerare unicamente la classe di oggetti "persone".

Il modulo permette, data un'immagine, di ottenere la posizione delle persone che la rete individua all'interno della scena. Questa posizione viene descritta da una *bounding box,* ovvero un rettangolo che racchiude la persona individuata, composta dalle coordinate dell'angolo in alto a sinistra del rettangolo e la lunghezza dei due lati. In Figura 4.2 possiamo vedere un esempio dell'output di questo modulo.

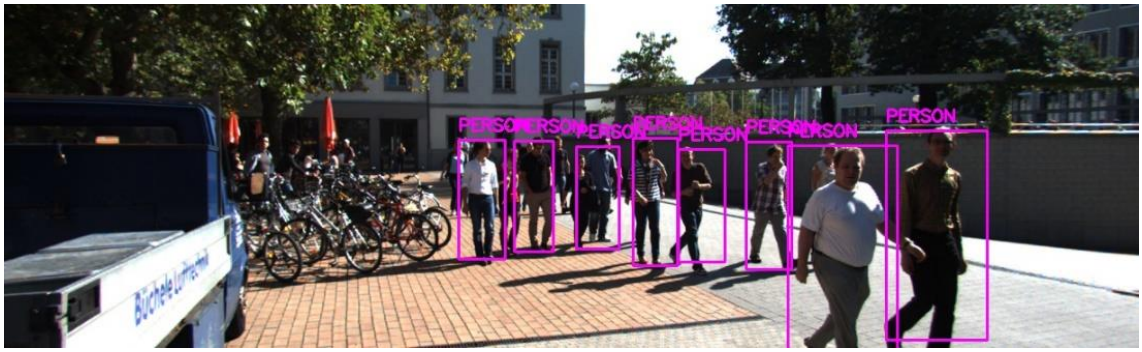

Figura 4.2 – Esempio di output della rete YoloV3 adattata per la sola people detection.

In aggiunta a YOLO, il modulo integra PoseNet [21], una rete che permette di individuare la posa delle persone, ovvero la posizione delle articolazioni di braccia, gambe e corpo, oltre che di occhi, naso e bocca. Essa restituisce una serie di vettori contenenti, per ogni persona, l'insieme delle coordinate dei singoli elementi che compongono la posa. In Figura 4.3 possiamo vedere un esempio dell'output della rete. Queste informazioni possono essere utilizzate per identificare un'ancora di riferimento per ogni persona. Così facendo, la distanza tra due persone è riconducibile alla distanza tra due ancore.

Figura 4.3 – Esempio di output della rete PoseNet.

Rispetto alla bounding box, la quale tende ad avere una oscillazione notevole della dimensione durante una sequenza video, diventando più grande o più piccola della persona racchiusa, la posa risulta essere più stabile come punto di tracciamento poiché ancorata alla struttura fisica della persona. Essa però tende ad avere una maggior variabilità delle giunture disponibili, poiché tende ad avere difficoltà a predire la posizione delle giunture per persone che non sono posizionate di fronte la telecamera. Inoltre, come si può notare in Figura 4.3, a volte le pose di persone molto vicine tra loro si vanno a sovrapporre ed unire. Va valutato perciò quale delle due metodologie risulti più appropriata per lo scenario di utilizzo, verificando anche quanto incidono rispettivamente sulle prestazioni del sistema finale.

4.2 Posizionamento delle persone

Una volta individuate le persone nella scena, il passaggio successivo è quello di andare a misurare la distanza tra di esse. Per poter fare ciò è necessario avere un sistema di riferimento, in scala metrica, in cui poter posizionare le persone. Questo sistema permetterebbe di ottenere un vettore di coordinate $[X, Y, Z]$ per ogni punto dell'immagine. È così possibile posizionare in maniera assoluta, rispetto al sistema di coordinate 3D scelto, le persone individuate, e successivamente calcolare la distanza tra di esse. Possiamo vedere una schematizzazione grafica del sistema in Figura 4.4. Come vedremo in seguito, le coordinate vengono ottenute rispetto alla posizione della telecamera. Di conseguenza il sistema di riferimento avrà tale origine come mostrato nella figura seguente.

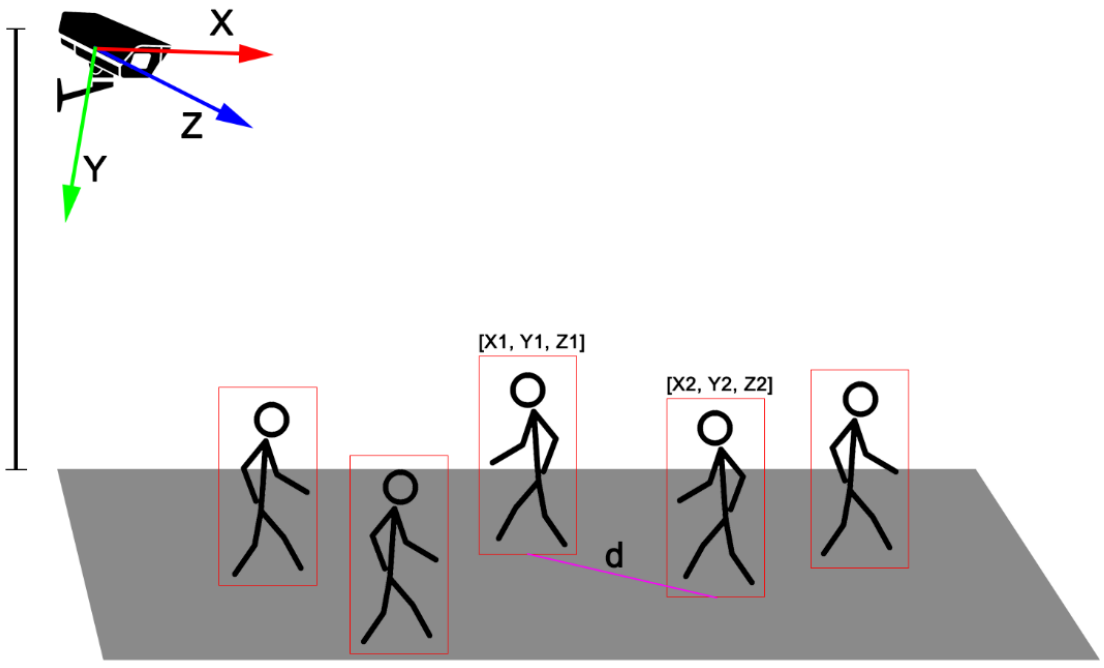

Figura 4.4 – Schematizzazione del problema con relativo sistema di riferimento 3D.

Per ottenere le coordinate X, Y e Z delle diverse persone, viene utilizzata una combinazione tra mappe di profondità, generate da reti neurali monodepth, e una telecamera accuratamente calibrata. Nello specifico Z, ovvero la distanza tra la telecamera e il punto della scena, viene ottenuta per mezzo delle reti neurali monodepth. Grazie invece ad una telecamera calibrata, per la quale è disponibile la matrice degli intrinseci, la quale contiene parametri che ne descrivono il modello matematico, è possibile ottenere le coordinate X e Y.

Proprio a questo punto ci si imbatte nella prima difficoltà d'uso delle reti neurali monodepth: l'output di queste reti non è la profondità reale metrica, ma un valore adimensionale relativo, il quale ci consente di conoscere se un punto dell'immagine si trova più o meno lontano rispetto ad un altro punto, ma non ci permette di sapere esattamente di quanto in unità di misura metriche. Proprio per questo motivo si è cercato di individuare un metodo per *mettere in scala* la mappa di profondità, ovvero trasformare i valori adimensionali in informazioni di profondità metrica.

Nei prossimi capitoli verranno descritte nel dettaglio le metodologie utilizzate, insieme ad un'analisi delle prestazioni ottenute.

Capitolo 5

Calibrazione telecamera

Risulta fondamentale conoscere le caratteristiche intrinseche della telecamera, le quali ci permettono di effettuare la trasformazione dal sistema immagine a quello reale tridimensionale. Nello specifico i parametri intrinseci, i quali definiscono il modello geometrico del sistema telecamera. Telecamere specializzate per la computer vision vengono talvolta vendute già calibrate. Tuttavia, negli scenari di utilizzo del sistema le telecamere difficilmente lo saranno, perciò è necessario effettuare questo procedimento durante l'installazione del sistema. Il processo tramite il quale si vanno ad ottenere i parametri che modellano la telecamera si chiama *calibrazione*. OpenCV fornisce una serie di funzioni che permettono di effettuare il processo di calibrazione su qualsiasi telecamera, permettendo di ottenere una stima dei relativi parametri.

5.1 Modello Pinhole

Le funzioni di calibrazione si basano sul modello chiamato *pinhole*, il quale rappresenta geometricamente in maniera accurata qualsiasi tipo di telecamera.

L'immagine di una qualsiasi scena può essere ottenuta proiettando ogni punto 3D P_w sul piano immagine tramite una trasformazione prospettica, la quale genera il corrispondente pixel p. Dato qualsiasi punto della scena 3D, è possibile ottenere il punto nel quale esso viene proiettato. Una rappresentazione grafica del modello è visibile in Figura 5.1.

La trasformazione può essere scritta come:

$$s\ p = A[R|t]P_w \tag{1}$$

Dove A è la matrice dei parametri intrinseci della telecamera, R e t descrivono la rotazione e traslazione che rappresentano il cambio di coordinate dal sistema di rifermento camera a quello reale, s un fattore di scala arbitrario. La matrice degli intrinseci è formata da quattro elementi: f_x e f_y, che rappresentano la lunghezza focale sull'asse x e y espressa in pixels, e il centro ottico (c_x, c_y). Con questi parametri possiamo scrivere la matrice A come:

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

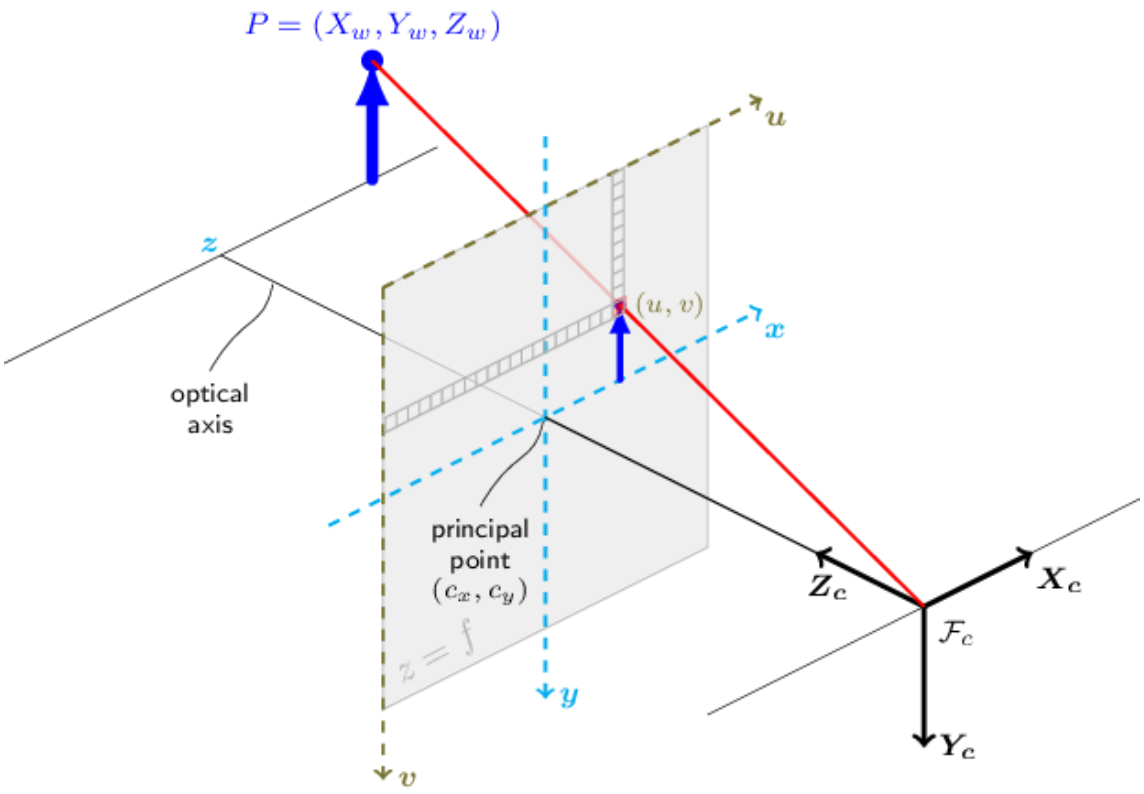

Figura 5.1 – il modello pinhole della telecamera [22].

R è composta da nove parametri di rotazione, tre per ogni asse, mentre t è composta da tre parametri di traslazione, uno per ogni asse. La matrice $[R|t]$ è così composta:

$$[R \quad t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \tag{3}$$

Andando a inserire le due matrici ottenute in (1), l'equazione che descrive la trasformazione diventa:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{4}$$

Questa rappresenta la trasformazione prospettica che proietta un qualsiasi punto 3D della scena in un punto dell'immagine.

Le telecamere reali ottengono immagini per mezzo di lenti, che permettono di focalizzare la luce sul sensore più facilmente. Queste lenti non sono perfette e provocano un effetto di

distorsione. Essa è composta principalmente da due tipi: radiale e tangenziale. La distorsione modifica le coordinate immagine rispetto a quelle reali. Il modello precedente viene perciò esteso per comprendere la distorsione generata dalla lente:

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} x' + \dfrac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 x' y' + p_2 (r^2 + 2x'^2) \\ y' + \dfrac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_2 x' y' + p_1 (r^2 + 2y'^2) \end{bmatrix} \tag{5}$$

Con

$$r^2 = x'^2 + y'^2 \tag{6}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \dfrac{X_C}{Z_c} \\ \dfrac{Y_C}{Z_C} \end{bmatrix} \tag{7}$$

Se $Z_c \neq 0$

I parametri di distorsione radiale sono k_1, k_2, k_3, k_4, k_5 e k_6, mentre p_1 e p_2 sono i coefficienti di distorsione tangenziale.

La distorsione più evidente è quella radiale, rappresentata in Figura 5.2.

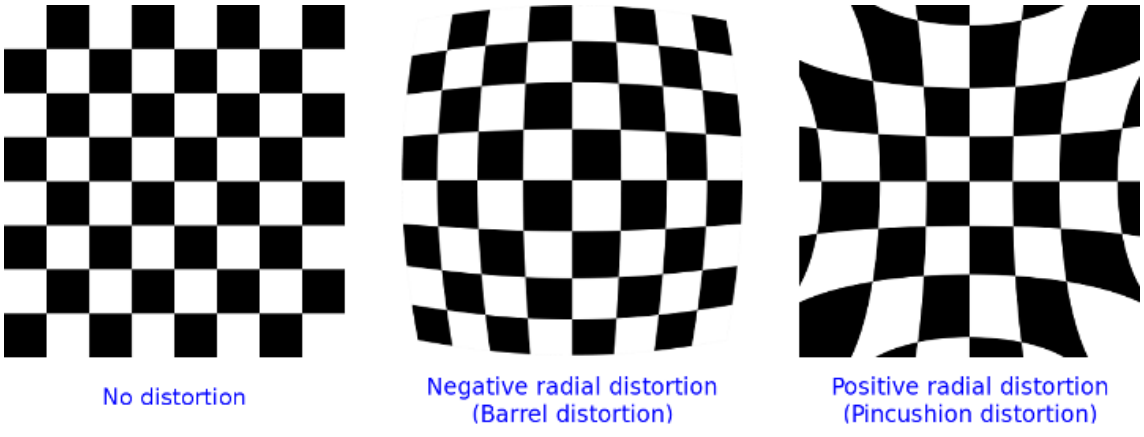

Figura 5.2 – Esempio di distorsione radiale [22].

Si è così ottenuto un modello geometrico che descrive in maniera accurata un qualsiasi sistema telecamera. Vediamo ora come ottenere tale modello tramite il processo di calibrazione.

5.2 Calibrazione

Il processo di calibrazione permette di ottenere, il più accuratamente possibile, una stima dei parametri che compongono il modello geometrico della telecamera in esame. OpenCV [22] mette a disposizione un insieme di funzioni che realizzano tale scopo.

La stima dei parametri è possibile tramite l'utilizzo di un particolare strumento, chiamato *calibration target*. Si tratta di un pattern ben definito, solitamente planare, di cui si conoscono le dimensioni delle sue caratteristiche. I più comuni pattern di calibrazione sono delle scacchiere o delle matrici di cerchi simmetrici o asimmetrici. Questi pattern possiedono caratteristiche ben definite e rilevabili, il che ci permette di sfruttarle per ottenere informazioni sui parametri della telecamera in esame.

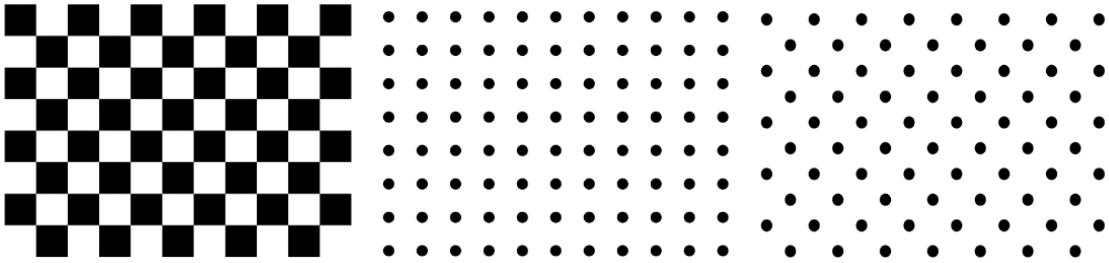

Figura 5.3 – Esempi di pattern di calibrazione.

Per effettuare la calibrazione è prima di tutto necessario acquisire una serie di immagini del pattern di calibrazione a diverse angolazioni. Da queste immagini sarà poi possibile ottenere un sistema di equazioni che, mediante un'operazione di minimizzazione, permettono di stimare i parametri della telecamera. Poiché non è scopo principale di questa tesi, si rimanda ad una descrizione più dettagliata del processo di calibrazione tramite OpenCV nella relativa documentazione [22].

All'interno del codice sviluppato è disponibile un modulo il quale, utilizzando le funzioni messe a disposizione da OpenCV, permette di calibrare la telecamera.

5.3 Calcolo delle coordinate X e Y

Una volta ottenuta la matrice dei parametri intrinseci relativa alla telecamera in uso, il calcolo delle coordinate X e Y nel sistema metrico si ottiene nel seguente modo: dato il modello geometrico della telecamera, la proiezione delle coordinate X, Y dell'oggetto nella scena sull'immagine risulta essere:

$$u = x\frac{f}{z} + c_x \tag{8}$$

$$v = y\frac{f}{z} + c_y \tag{9}$$

Le coordinate immagine corrispondono alle coordinate reali scalate per il rapporto tra la focale e la distanza reale. c_x e c_y corrispondono alle coordinate del centro ottico (principal point in Figura 5.1), ovvero il punto in cui l'asse ottico interseca il piano dell'immagine. Andando a sommare alle rispettive coordinate, si va a compensare la traslazione generata dalla posizione del sistema di coordinate immagine, il quale ha come origine l'angolo alto a sinistra.

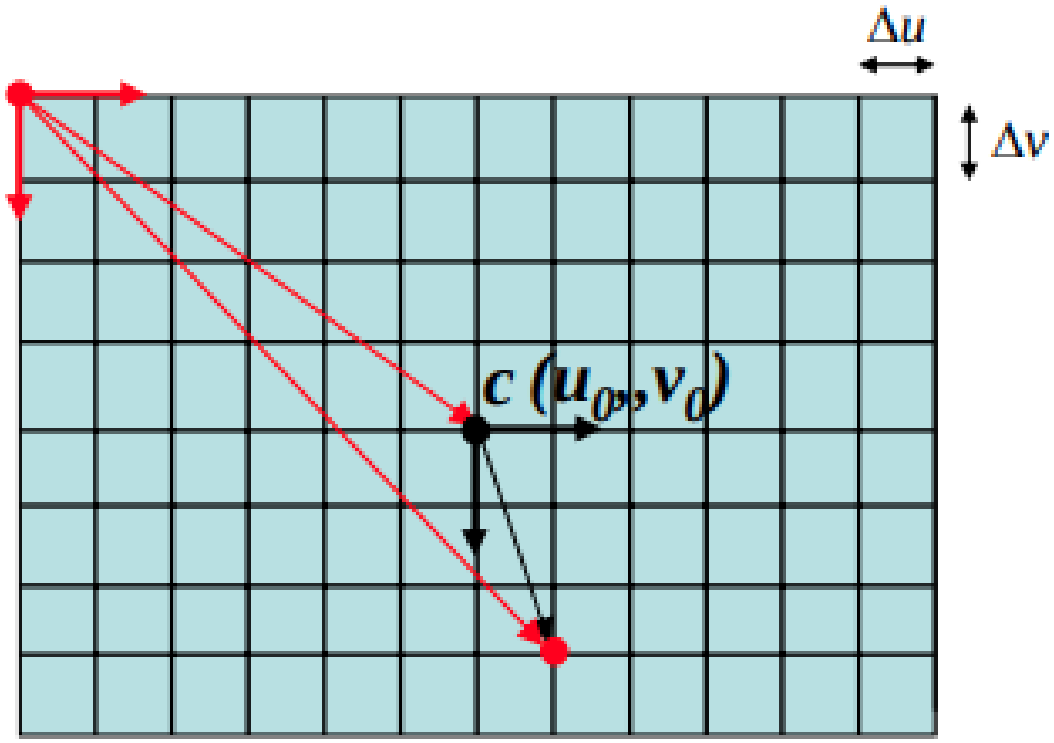

Figura 5.4 – Traslazione tra il sistema di coordinate immagine e il centro ottico.

Effettuando l’operazione inversa è possibile calcolare le coordinate X e Y:

$$x = (u - c_x)\frac{z}{f} \tag{10}$$

$$y = (v - c_y)\frac{z}{f} \tag{11}$$

Se la distanza Z in metri è conosciuta, le coordinate X e Y così ottenute risultano in metri. Si può notare come risulti necessario conoscere la coordinata Z, corrispondente alla profondità, per ottenere le coordinate X e Y. Come detto in precedenza, essa verrà ottenuta per mezzo di reti neurali monodepth.

Nei prossimi capitoli analizzeremo queste reti e vedremo come utilizzarle per ottenere le informazioni di profondità della scena.

Capitolo 6

Monodepth Neural Networks

In tutti i campi applicativi nei quali è necessario avere una misura della profondità in tempo reale (robotica, veicoli autonomi, realtà aumentata, ecc.), i sensori utilizzati per ottenere tale informazione sono principalmente di due tipi:

- Attivi: LiDAR, Structured Light, Time-of-Flight sensors
- Passivi: sistemi di visione stereo

I sensori attivi utilizzano una sorgente di energia controllata, ad esempio un laser o una proiezione di un particolare pattern luminoso, insieme ad un elemento che ne rileva tale emissione. D'altro canto, quelli passivi recuperano informazioni 3D utilizzando il principio della triangolazione. La distanza viene calcolata andando ad analizzare il triangolo che si forma tra l'oggetto e due distinti sensori, ad esempio due telecamere, posti tra loro ad una distanza nota.

Sono innovativi e di grande interesse sistemi di tipo monoculare, ovvero in grado di inferire informazioni sulla profondità a partire da una singola immagine. Questo perché permetterebbe potenzialmente di creare sistemi più piccoli ed economici rispetto alle alternative ad oggi disponibili.

Negli ultimi anni sono nate nuove e innovative tecnologie per l'inferenza della profondità da immagini monoculari, le quali si basano su Reti Neurali Convoluzionali (CNN). Si tratta di Reti Neurali Monoculari (Monodepth Neural Networks). Queste reti, addestrate appositamente, hanno la capacità di restituire informazioni sulla profondità della scena inquadrata senza la necessità di avere a disposizione due immagini stereo.

In particolare, durante questo progetto sono state testate e analizzate due reti: PyD-Net e MiDaS. La necessità di valutare più reti neurali monodepth deriva da problematiche implementative. L'obiettivo è trovare un modello accurato, il quale permetta di monitorare il distanziamento sociale correttamente, che non richieda una quantità elevata di risorse hardware. Questo perché quelle a disposizione non sono particolarmente prestanti.

Per quanto simili, i modelli analizzati presentano prestazioni diverse per all'interno del particolare scenario di utilizzo del sistema. Si è resa perciò necessaria un'analisi della struttura delle rispettive mappe ottenute e delle relative prestazioni.

6.1 PyD-Net

PyD-Net è una rete innovativa proposta dal Computer Vision Lab dell'Università di Bologna, il quale permette di ottenere una stima della profondità da singola camera [13]. Nonostante reti di questo tipo necessitino di potenti GPU per funzionare, la sua particolare struttura ne permette l'utilizzo in maniera efficiente anche su dispositivi embedded.

Essa si ispira al successo dell'architettura piramidale per realizzare una rete con un numero di parametri e un utilizzo di risorse computazionali ridotto rispetto ad altre soluzioni allo stato dell'arte. Da qui il nome PyD-Net (Pyramidal Depth Network).

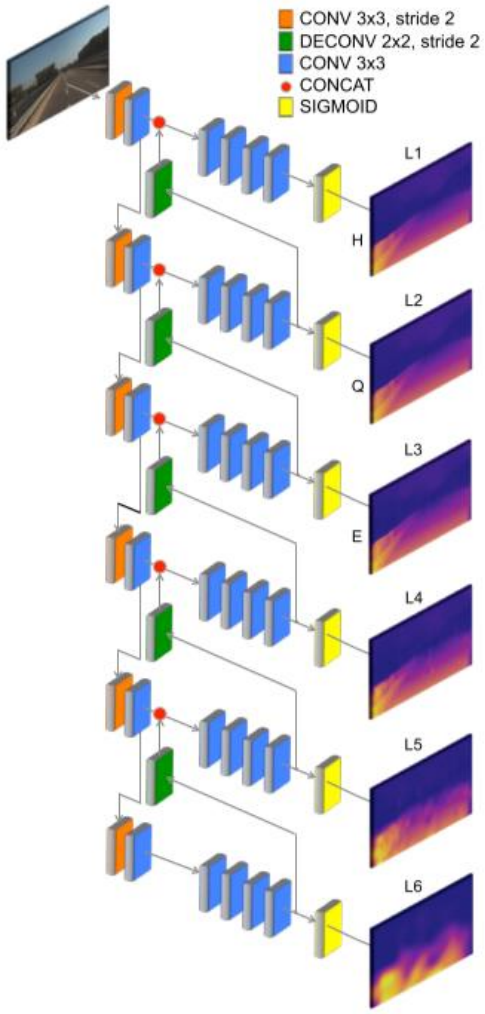

Figura 6.1 – Architettura di PyD-Net.

Tale piramide si compone di sei livelli concatenati, ognuno dei quali analizza l'immagine in ingresso a risoluzione ridotta, da L1 a L6 partendo da metà risoluzione fino a $\frac{1}{64}$.

La profondità viene stimata ad ogni livello per poi essere inviata a quello superiore, dove la stima viene raffinata. Infatti, il risultato ottenuto dal livello corrente viene ogni volta

confrontato con quello ricevuto dal livello più basso. Questo design permette alla rete di predire sempre la profondità alla massima risoluzione.

Messa a confronto con reti allo stato dell'arte, dimostra di avere prestazioni comparabili, permettendone, al costo di una piccola perdita di precisione, l'utilizzo in tempo reale su sistemi con CPU standard o embedded.

La rete PyD-Net utilizzata in questo progetto è stata addestrata con circa 450 mila immagini, utilizzando MiDaS come supervisore.

6.2 MiDaS

MiDaS [17] è una rete neurale monodepth molto accurata sviluppata da Intelligent Systems Lab di Intel. Essa, nella sua versione base, è molto più pesante, dal punto di vista computazionale e di risorse utilizzate, rispetto a PyD-Net. Al tempo di questo progetto la rete è disponibile in versione 2.0 e 2.1. Rispetto alla versione precedente, il modello della versione 2.1 è stato migliorato per avere una maggiore accuratezza di circa il 10%. Di quest'ultima è disponibile anche un modello *small*, il quale è stato alleggerito per permetterne l'utilizzo in tempo reale su piattaforme mobili e dispositivi embedded.

La novità introdotta da MiDaS è la metodologia con cui la rete è stata addestrata. Spesso per questo compito viene utilizzato uno dei dataset già esistenti e disponibili, il quale però può non rappresentare un modello accurato di scenari reali. Al fine di migliorare questo aspetto, sono stati utilizzati per l'addestramento e valutazione del modello un insieme eterogeneo di dataset stereo provenienti da diverse fonti. Tra questi è stato aggiunto un dataset composto da film 3D, i quali forniscono un ampio insieme di dati con relative informazioni stereo.

Questa scelta ha generato una serie di sfide, tra cui il fatto che ci fossero, tra i vari dataset, diverse rappresentazioni della profondità, che essa sia disponibile a meno di un fattore di scala e che per alcuni dataset, ad esempio i film 3D, non è conosciuta la distanza tra le due camere del sistema stereo. Per superare questi ostacoli sono state realizzate una serie di funzioni che permettono di gestire queste ambiguità in fase di allenamento della rete.

6.3 Analisi e struttura delle mappe

Le mappe di profondità ottenute per mezzo di queste reti neurali monodepth non presentano valori assoluti. Questi indicano una dimensione di profondità relativa alla scena, senza alcun valore metrico. Sostanzialmente informano che un certo punto (i, j) si trova davanti o dietro rispetto ad un altro punto nella scena, ma non ci dicono effettivamente di quanto.

Nelle reti analizzate, i valori sono profondità inverse. Definiamo d_{ij} come l'ipotetico valore della profondità al punto (i, j). Possiamo scrivere ogni punto della mappa di profondità $D(i,j)$ come:

$$D(i,j) = \frac{1}{d_{ij}} \tag{12}$$

Questa caratteristica va tenuta in considerazione ogniqualvolta si andranno ad eseguire operazioni sulle mappe, come ad esempio la messa in scala che vedremo nei capitoli successivi.

Per verificare l'accuratezza di queste mappe si può utilizzare in maniera qualitativa una mappa di profondità ottenuta da una telecamera stereo. Un esempio di mappa stereo è mostrato in Figura 6.2.

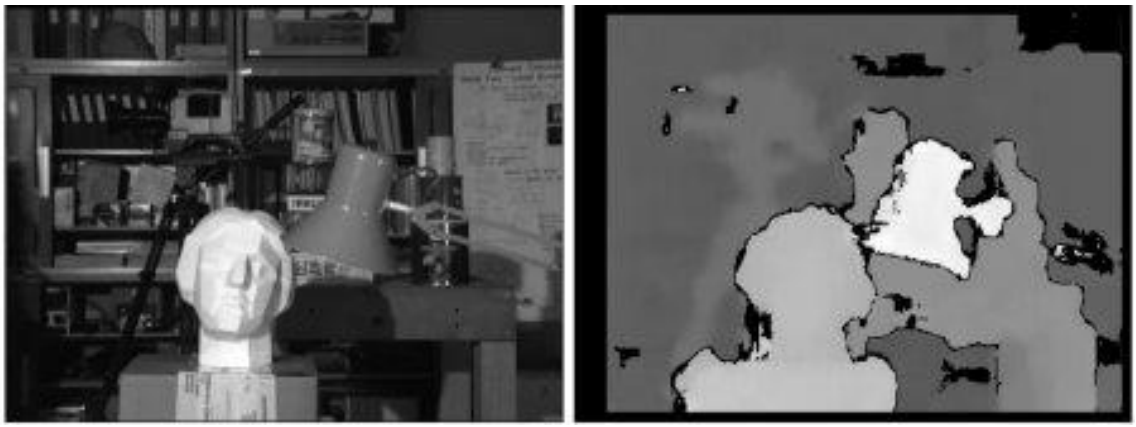

Figura 6.2 – Esempio di mappa di profondità stereo. A sinistra possiamo vedere l'immagine originale, a destra la relativa mappa delle disparità ottenuta per mezzo di una telecamera stereo.

Dopo aver effettuato alcune prove e aver messo a confronto l'uscita di queste reti con mappe stereo, si può notare come i risultati sembrino a prima vista estremamente accurati, dove è visibile un'ottima individuazione delle differenti profondità degli oggetti presenti nella scena analizzata. In Figura 6.3 se ne può vedere un esempio.

Questo risultato è valido sia per una rete come MiDaS, sia per una rete più leggera come PyDNet. Entrambe sono realizzate come reti *general purpose*, pensate per essere utilizzate negli scenari più svariati, e perciò il loro addestramento è stato realizzato con dataset contenenti diversi scenari di utilizzo. A prima vista, queste reti sembrano essere molto promettenti, il che renderebbe possibili sistemi leggeri e accurati per la stima della profondità utilizzando una sola telecamera.

Purtroppo, un'analisi approfondita rivela una serie di problematiche che affliggono queste reti. Lo scenario d'uso in cui si andrà ad inserire il sistema che stiamo considerando, presenta caratteristiche ben definite: uno spazio, aperto o chiuso, in cui sono presenti diverse persone in movimento. Per quanto possa non sembrare un fattore discriminante, nel momento in cui la scena presenta queste caratteristiche notiamo una prima differenza tra le due reti allo stato attuale: il dettaglio con cui vengono rappresentate le profondità delle persone nella scena, visibile in Figura 6.3. In certi scenari, la rete MiDaS non individua correttamente la testa delle persone, assegnando una profondità che più si avvicina a quella degli elementi presenti dietro ad esse. Si può anche notare come le persone più lontane e negli angoli dell'immagine vengano difficilmente individuate.

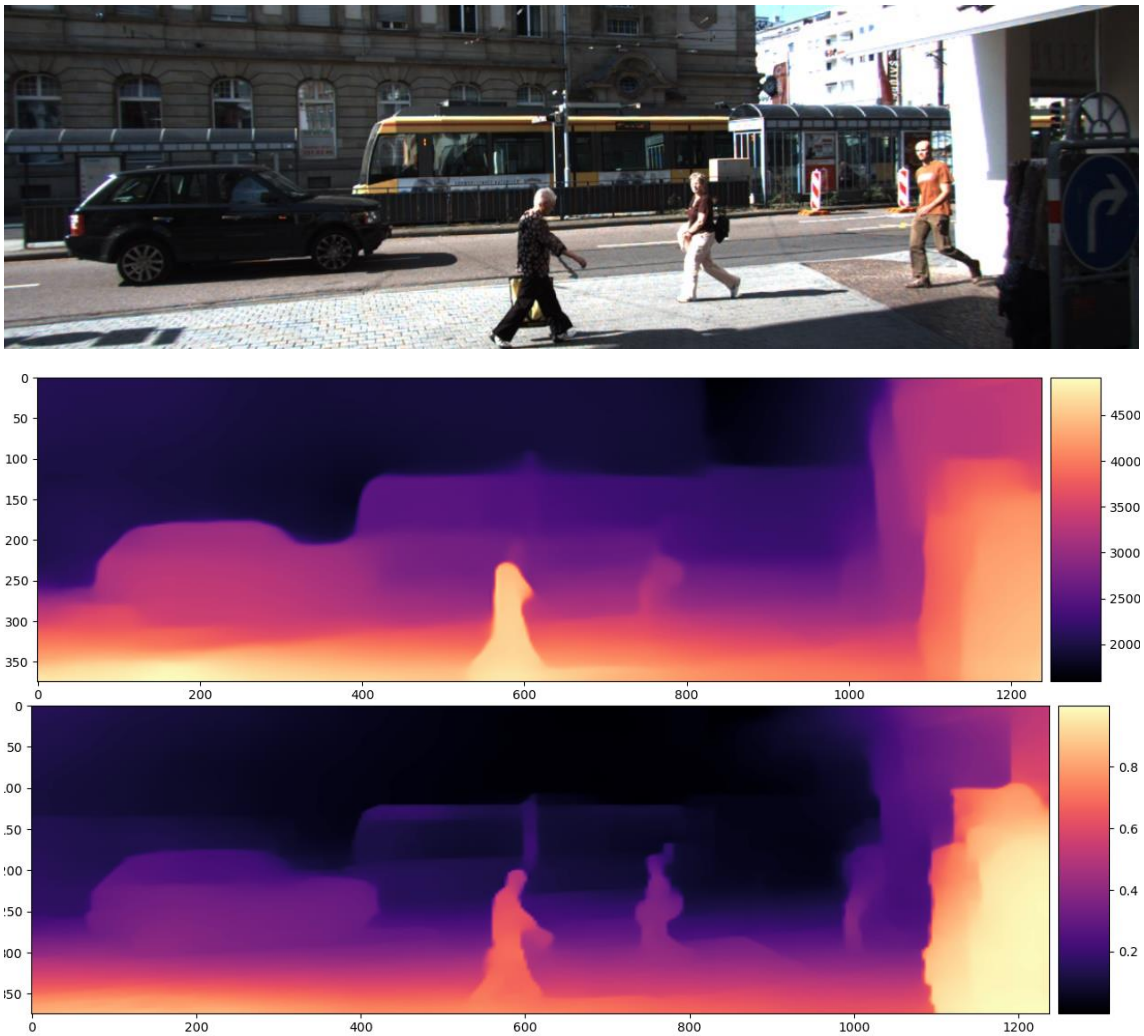

Figura 6.3 – Esempio di inferenza della profondità: in alto l'immagine originale del dataset KITTI. In mezzo la mappa di MiDaS 2.0. In basso la mappa di PyD-Net.

Ciò può essere causa del modo in cui le due reti sono state addestrate. Inoltre, poiché queste reti riducono la dimensione dell'immagine prima di analizzarla, il formato immagine del dataset KITTI potrebbe causare un eccessivo ridimensionamento della larghezza, il che potrebbe causare una distorsione dell'immagine, generando una perdita dei dettagli e di conseguenza l'errata identificazione delle teste. Al contrario PyD-Net sembra non mostrare questo comportamento, ma in diverse situazioni, come quella mostrata in Figura 6.4, la rete non riconosce la presenza di un corridoio. Quest'ultima problematica riguarda uno scenario alquanto particolare, il che non preoccupa. L'errata individuazione delle persone, potrebbe rappresentare un primo grave problema, in quanto il sistema che si vuole costruire si focalizza proprio su questo scenario.

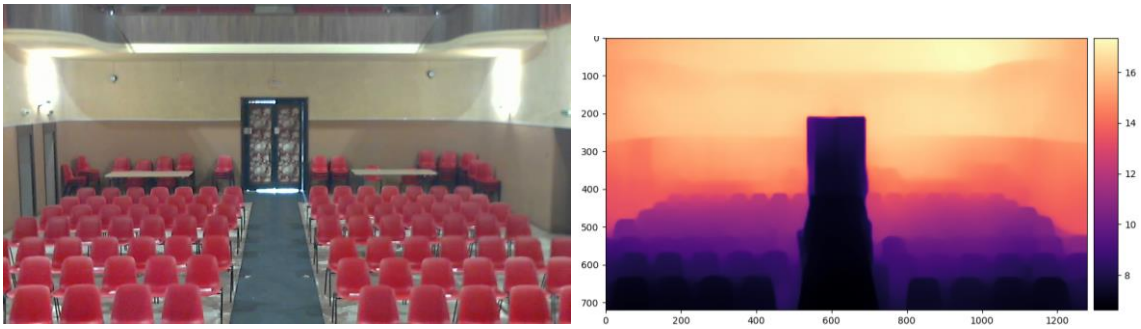

Figura 6.4 – Inferenza di profondità utilizzando PyD-Net. In questo caso la mappa di colore inizia dal nero e passa a colori sempre più caldi man mano che ci si allontana come profondità. Possiamo vedere come il corridoio venga rappresentato tutto alla stessa distanza.

Oltre a quelle appena descritte, entrambe le reti in analisi mostrano un comportamento problematico comune che sembra essere caratterizzante: la predizione della profondità della scena tende ad avere una sovrastima sempre maggiore man mano che si sale in altezza all'interno dell'immagine. Sostanzialmente la parte superiore dell'immagine tende verso il fondo della scena, nonostante nella realtà dell'immagine questo non accada. L'atteggiamento appena descritto si può notare in Figura 6.5. L'immagine mostra la visualizzazione tridimensionale dell'immagine per mezzo di Open3D [23]. Applicando la mappa di profondità ottenuta dall'immagine sorgente, si può notare come il furgone presente nella scena, e anche tutti gli altri oggetti, risulta comprimersi sempre più verso lo sfondo man mano che si sale in altezza. Nelle immagini e scenari in cui sono presenti persone, la visualizzazione 3D mostra le persone inclinate verso lo sfondo, invece che verticali rispetto al piano. Si può dire che la rete abbia un *bias* (pregiudizio), dove tende a credere che la distanza aumenti sempre in maniera costante man mano che si sale nell'immagine. Questo comportamento potrebbe essere causa dei dataset utilizzati per addestrare queste reti. L'ipotesi è che solitamente i dati di profondità sono sempre più sparsi man mano che si sale in una scena, poiché oggetti molto lontani o il cielo non possono

essere misurati. Frequentemente i punti più lontani della scena vengono rappresentati nella parte alta dell'immagine. La rete probabilmente impara questa caratteristica e fatica a discriminare quei punti che, a seconda della prospettiva, si trovano più in alto ma fanno parte di un oggetto in primo piano. Questo bias crea una situazione alquanto problematica, poiché a prima vista rende inutilizzabile questo tipo di reti neurali nelle situazioni in cui precisione e accuratezza sono di fondamentale importanza.

Figura 6.5 – Sopra: immagine sorgente. Sotto: Visualizzazione tridimensionale dell'output della rete monodepth MiDaS. Al centro dell'immagine si può notare come il retro del furgone bianco venga deformato e sempre più schiacciato verso il fondo man mano che si sale in altezza. Questo effetto è causa del bias intrinseco di queste reti.

L'analisi effettuata mostra come queste mappe, che a prima vista sembrano molto precise, in realtà nascondono diverse problematiche interne che richiedono appropriate strategie per renderle utilizzabili efficacemente.

6.3.1 Soluzioni proposte

Tra le diverse soluzioni possibili, quella che potrebbe portare il maggior miglioramento per questa serie di problematiche sarebbe un riaddestramento delle reti utilizzando solo immagini di scene simili a quelle in cui il sistema andrà ad inserirsi. In questo modo si realizzerebbe una rete ad-hoc per l'identificazione della profondità. Se questa fosse l'unica soluzione, poiché lo scenario di utilizzo non è unico, sarebbe necessario addestrare la rete utilizzando immagini relative al sito di installazione. Purtroppo, sarebbe un lavoro troppo lungo che richiederebbe molto tempo sia per la raccolta di un dataset sufficientemente adeguato che per un riaddestramento della rete, tempo improponibile per uno strumento che ha tra gli obiettivi la facilità di installazione.

Inoltre, l'interesse della sperimentazione effettuata era quello di vedere se le reti possono essere utilizzate come *black box*, ovvero senza apportare alcuna modifica.

Come soluzione alternativa, la quale si basa sul fatto che l'elemento di interesse all'interno della scena sono le persone, si è ipotizzato di considerare come distanza di riferimento delle singole persone la media delle distanze delle loro giunture ottenute tramite PoseNet. In questo modo potrebbe essere possibile alleviare in parte le imprecisioni di individuazione delle persone e la distorsione verso il fondo della scena, ottenendo comunque una precisione accettabile nella misura della distanza interpersonale. Vedremo inoltre se la messa in scala metrica delle mappe di profondità riesca a mitigare questo bias.

Nel capitolo successivo verrà descritto il processo di messa in scala e verranno mostrati i primi risultati ottenuti.

Capitolo 7

Messa in scala mappe monodepth

Come anticipato nel capitolo 4, le mappe di profondità ottenute per mezzo di reti neurali monodepth necessitano di una messa in scala per ottenere una misura metrica della distanza. Questo perché esse contengono informazioni sulla posizione relativa dei punti nella scena, senza alcuna informazione di tipo metrico.

L'idea generale alla base del procedimento è quella di ottenere un fattore di scala associando alle mappe di profondità delle distanze note all'interno della scena. Calcolando il rapporto tra la distanza conosciuta e il valore in quel punto della mappa, si ottiene un fattore di scala numerico che, moltiplicato per la mappa, porterebbe in scala metrica tutti i valori contenuti in essa. Questo ragionamento è generalmente valido, ma nel nostro caso viene rinforzato dalla condizione per cui la scena inquadrata rimane costante nel tempo. Infatti, la telecamera, una volta installata, andrà ad inquadrare sempre lo stesso scenario, in cui, con buona probabilità, vi saranno una serie di punti a distanza costante nel tempo (mobili, muri, edifici, ecc.).

I valori di profondità ottenuti dalla rete per uno stesso punto sono differenti se l'immagine in analisi contiene una scena leggermente diversa rispetto a quella precedente. Ad esempio, nel momento in cui le persone si muovono, il valore per ogni punto cambia poiché ottenuto in maniera relativa al resto della scena. Se la scena è diversa, il valore risulta diverso. Questo rende necessario ricalcolare il fattore di scala ad ogni immagine della sequenza e, di conseguenza, avere a disposizione tali distanze note in ogni nuova immagine. La condizione per cui la scena è statica, a meno delle persone, ci consente di individuare punti fissi all'interno della scena che non variano la loro distanza rispetto alla telecamera nel tempo. Questi possono essere sfruttati per ricalcolare il fattore di scala ad ogni immagine.

Ipotizzando inizialmente di avere un metodo per ottenere queste distanze conosciute, ci si è interrogati se fosse sufficiente un singolo punto per rimettere in scala tutta la mappa di profondità oppure se fossero necessari più punti.

7.1 Scala da singolo punto

Conoscendo la distanza di un singolo punto all'interno della scena, si vuole calcolare il fattore che, moltiplicato per la mappa di profondità relativa, porti in scala metrica tutti i valori contenuti in essa.

Data k_{ij} distanza conosciuta al punto (i, j) e $D(i,j)$ il valore della mappa di profondità ottenuta dalla rete neurale monodepth in tale posizione, il fattore di scala s si ottiene come:

$$s = \frac{k_{ij}}{D(i,j)} \tag{13}$$

Una volta calcolato il fattore s è possibile mettere in scala l'intera mappa di profondità per mezzo di una semplice moltiplicazione:

$$D_s(i,j) = s\, D(i,j) \tag{14}$$

In questo modo sarebbe sufficiente conoscere una singola distanza, costante nel tempo e facilmente rintracciabile ad ogni esecuzione, all'interno della scena per rimettere in scala l'intera mappa.

7.1.1 Acquisizione distanza nota

L'acquisizione di una distanza metrica nota è possibile effettuarla in maniera semi-automatica. Per fare ciò è sufficiente un pattern conosciuto, come una scacchiera, il quale abbia delle caratteristiche facilmente identificabili. Il procedimento è simile a quello utilizzato per la calibrazione della telecamera. Andando a posizionare il pattern in un punto della scena corrispondete al punto fisso scelto (Figura 7.1), è possibile selezionarlo in maniera automatica dal software in una fase preliminare, per poi associare a tale punto una distanza metrica.

Per ottenere tale distanza viene eseguita l'operazione inversa della proiezione prospettica rispetto al modello geometrico pinhole.

Sappiamo che un punto della scena 3D viene proiettato sull'immagine tramite la trasformazione:

$$s\, p = A[R|t]P_w \tag{15}$$

È possibile ricavare il punto 3D operando l'operazione inversa

$$P_w = R^{-1}(A^{-1}p - t) \tag{16}$$

Figura 7.1 – La scacchiera viene posizionata in un punto della scena per ottenere la distanza nota.

Avendo a disposizione diversi punti immagine corrispondenti agli angoli della scacchiera, è possibile, per mezzo della Perspective-n-Point, ottenere la matrice di rototraslazione relativa all'angolo in altro a sinistra. Andiamo così ad ottenere le coordinate $[X, Y, Z]$ dell'angolo di riferimento.

La scelta del pattern e della sua grandezza deve tenere in considerazione la risoluzione della telecamera. Se la risoluzione fosse troppo bassa, un pattern piccolo renderebbe imprecisa la localizzazione del corner.

7.1.2 Primi Risultati

I primi risultati di questa metodologia hanno evidenziato una forte dilatazione delle distanze a partire dal punto utilizzato per la messa in scala. I valori relativi ai punti della scena che si trovano alla stessa profondità del punto conosciuto, o a profondità immediatamente vicine, vengono messi in scala in maniera corretta, mentre allontanandosi sia verso il fondo che verso la telecamera, si ottiene una dilatazione sostanziale delle misure, in maniera quasi esponenziale, con aumenti di anche 100m in più rispetto alla misura reale. Questo comportamento è causato probabilmente dalla distorsione intrinseca delle mappe discussa nel capitolo 6. Un singolo punto non permette di ridurre tale effetto, riportando tale distorsione anche nella mappa in scala.

Possiamo inoltre notare come si generino zone dell'immagine, mostrate in Figura 7.2, in cui le distanze in scala raggiungono valori elevatissimi. Questa situazione può essere facilmente risolta filtrando i valori di profondità che superano una certa soglia, ma comunque è un comportamento da tenere in considerazione.

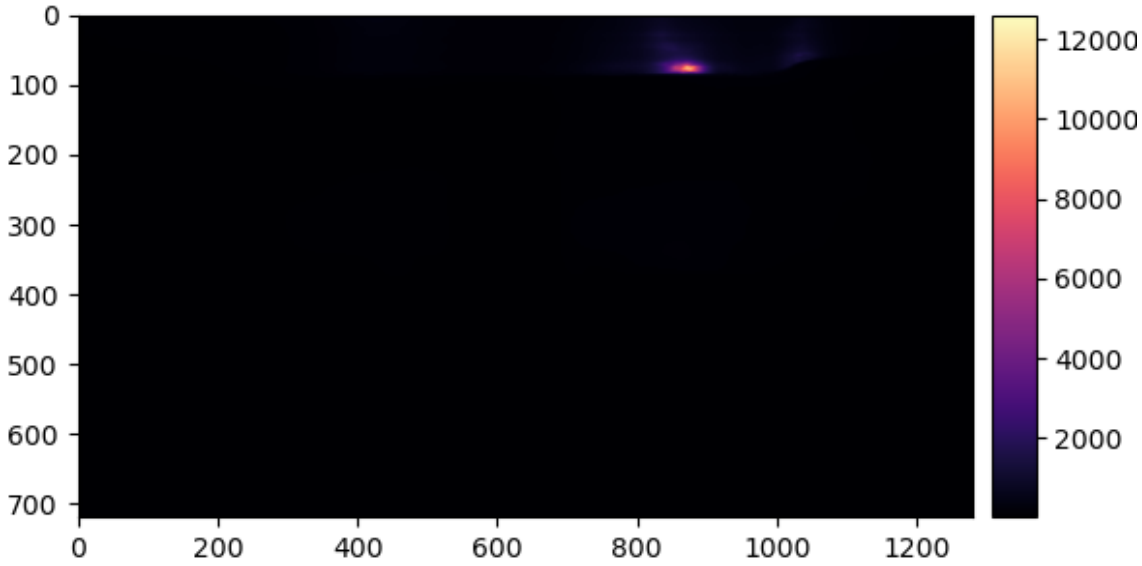

Figura 7.2 – Risultato della messa in scala della mappa di profondità con singolo punto. Si può notare come sia presente un punto concentrato in cui le distanze rimesse in scala raggiugono valori al di sopra dei 1000m.

Dai risultati ottenuti possiamo evincere che l'utilizzo di un singolo punto non sia sufficiente ad ottenere una messa in scala precisa delle mappe di profondità.

7.2 Scala a punti multipli

Abbiamo visto come una singola distanza nota non basti a mettere in scala la mappa di profondità. Risulta perciò necessario usare più di un singolo punto. Questo porta a pensare che probabilmente non sia sufficiente solo il fattore di scala. Questo ulteriore valore si chiama *shift* e, insieme al fattore di scala moltiplicativo, permette di ottenere una mappa di profondità in scala metrica conoscendo una serie di distanze note nella scena.

Il metodo *scale and shift* ha come scopo quello di trovare, a partire da una matrice di distanze note, due variabili x_1(scale) e x_2(shift) tali che:

$$D_r \cong x_1 P + x_2 \tag{17}$$

Dove D_r è la distanza reale e P la predizione della rete neurale monodepth. L'obiettivo è calcolare la retta interpolatrice che meglio approssima i valori relativi dell'inferenza della rete alle distanze conosciute negli stessi punti della scena.

A partire da una matrice sparsa T, tale che:

$$T_{ij} = \begin{cases} D_{rij} \\ 0 \end{cases} \tag{18}$$

dove l'elemento T_{ij} contiene la distanza reale metrica, se conosciuta, del punto (i, j), e M matrice sparsa, tale che:

$$M_{ij} = \begin{cases} 1, & se\ T_{ij} \neq 0 \\ 0, & se\ T_{ij} = 0 \end{cases} \tag{19}$$

è possibile ottenere un sistema di equazioni in due incognite nella forma:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{20}$$

Risolvendo questo sistema ai minimi quadrati, è possibile ottenere i due valori di scala e shift, i quali permettono di mettere in scala metrica la predizione della rete, rispetto ai valori di distanze noti all'interno della scena inquadrata.

Poiché si tratta di un'operazione di interpolazione, almeno due punti conosciuti sono necessari per poter calcolare i due fattori. Teoricamente più punti sono disponibili, maggiore sarà la precisione della messa in scala. Vedremo che nella pratica non è così. Questo metodo è lo stesso che viene utilizzato da MiDaS per la messa in scala.

Ora che sappiamo come ottenere le mappe in scala, risulta necessario capire come avere a disposizione dei valori di profondità noti della scena.

7.2.1 Acquisizione distanze note

Acquisire più punti noti necessità l'impiego di tecniche meno semplici e immediate rispetto all'acquisizione di un singolo punto, il che rende la configurazione iniziale del sistema più laboriosa di quanto si sperasse. Ciò nonostante, questo procedimento risulta necessario per garantire un funzionamento corretto e non è tanto più invasivo di altre operazioni di calibrazione impiegate in altri sistemi.

Diverse soluzioni sono possibili per ottenere i punti necessari. Per esempio:

- Ampliare il numero di punti disponibili utilizzando tutti i punti della scacchiera: Se si considera di utilizzare sempre una scacchiera, è possibile recuperare in maniera automatica la posizione di tutti i suoi punti grazie alla conoscenza dell'angolo principale e le caratteristiche fisiche (dimensione scacchi, n° colonne, n° righe). Questo permetterebbe di ampliare il numero di punti disponibili posizionando la scacchiera in più posizioni nella scena.
- SLAM (Simultaneous Localization And Mapping): Realizzando un Sistema che permetta di eseguire SLAM su un dispositivo portatile (Smartphone o Tablet), si potrebbe ottenere una mappatura della scena inquadrata dalla telecamera, dalla quale estrapolare successivamente una serie di punti da utilizzare per la messa in scala delle mappe di profondità. In questo caso, poiché la telecamera usata per eseguire lo SLAM è diversa da quella usata dal sistema, sarebbe poi necessario calcolare la rototraslazione tra le due, così da poter ottenere una trasformazione tra i due sistemi di riferimento.
- Sensori attivi o passivi di misurazione della profondità: Utilizzare questi sensori solo in fase di installazione del sistema, per poi rimuoverli una volta ottenuta la nuvola di punti. Anche in questo caso andrebbe calcolata la rototraslazione se la posizione delle due telecamere, quella usata per acquisire la mappa e quella usata dal sistema finale, è diversa.

All'interno di questo progetto di tesi non è stato scelto il metodo che verrà utilizzato nel sistema finale, poiché ancora ci si trova in fase di testing.

Per verificare che il metodo fosse funzionante erano necessari dataset contenenti valori di profondità veri della scena. A questo scopo sono stati utilizzati in un primo momento dati ottenuti in maniera empirica per mezzo di un telemetro laser. Vista la complessità richiesta per effettuare misurazioni precise con una simile configurazione, è stato deciso di impiegare alcune sequenze particolari del dataset KITTI, in cui sono disponibili dati di profondità ottenuti mediante LiDAR per una telecamera fissa che inquadra una situazione di affollamento. Dai dati relativi alle profondità vengono poi estrapolati un certo numero punti in maniera randomica che verranno utilizzati per mettere in scala le predizioni delle reti neurali monodepth.

7.2.2 Primi Risultati

I risultati iniziali ottenuti da parte di questo metodo hanno mostrato dati promettenti sulla sua efficacia nel mettere in scale le mappe di profondità. Grazie ai dati di profondità contenuti nel dataset KITTI, è stato possibile creare la nuvola di punti sparsi noti. Un subset di valori viene

utilizzato per la messa in scala, dopodiché la mappa ottenuta viene confrontata con i dati LIDAR per verificarne la precisione.

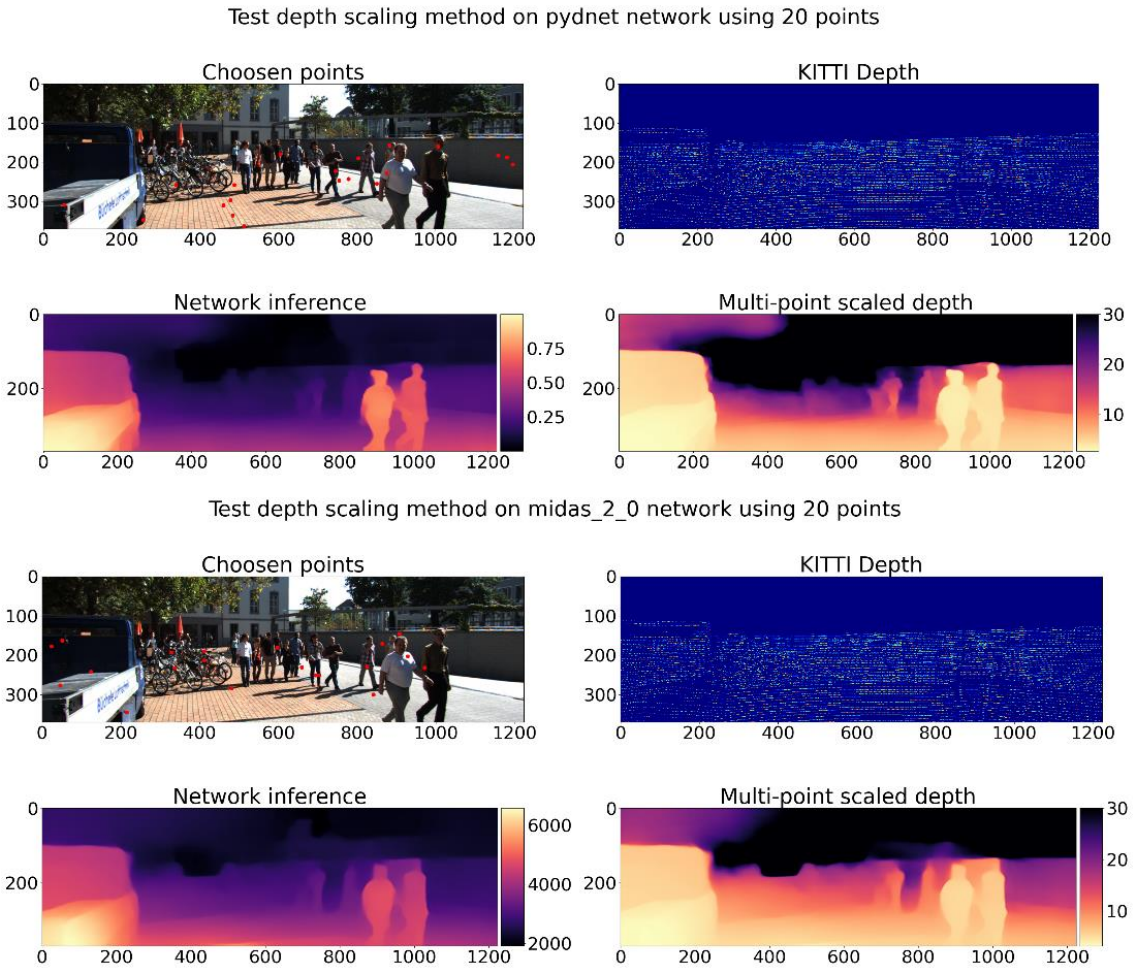

Figura 7.3 – Test di messa in scala di una mappa ottenuta tramite PyD-Net (sopra) e MiDaS 2.0 (sotto). Nell'immagine in basso a destra troviamo la mappa di profondità scalata.

Come possiamo vedere in Figura 7.3, le mappe scalate, confrontate con i dati disponibili da KITTI, sembrano avere una buona precisione paragonate con i punti in cui le informazioni LiDAR sono disponibili. Nella figura si può notare come il range di valori sia 0-30m, il che corrisponde con quello dei dati LiDAR. Poiché si tratta di dati sparsi messi a confronto con una mappa densa, siamo limitati nella misura delle prestazioni. Ciononostante, questi risultati sono incoraggianti e dimostrano che questa è la metodologia da seguire. Nel capitolo 10 verranno mostrati dati sperimentali ottenuti grazie ad un dataset acquisito per mezzo di una telecamera stereo. In Figura 7.4 è visibile l'effetto della messa in scala per la rete MiDaS 2.0.

Capitolo 8

Misurazione della distanza

A questo punto si ha a disposizione un sistema di riferimento, in cui le coordinate in metri di ogni punto della scena sono conosciute.

L'obiettivo ultimo di questo sistema è quello di monitorare in tempo reale il rispetto delle norme di distanziamento sociale. Per fare ciò, come anticipato nel capitolo 4, il primo passo è individuare le persone presenti nella scena per mezzo del modulo di people detection. Successivamente si ottiene, per mezzo della rete neurale monodepth scelta, la mappa di profondità e la si riporta in scala metrica utilizzando il metodo ai minimi quadrati descritto nel capitolo 7.2. A questo punto è sufficiente individuare un riferimento per ogni persona, in questo caso un punto della bounding box che la racchiude oppure una giuntura individuata tramite PoseNet, e calcolare le rispettive coordinate $[X, Y, Z]$ nel sistema di riferimento metrico, utilizzando i metodi precedentemente discussi. In entrambi i casi, perché la misura della distanza sia valida, l'elemento di riferimento scelto deve essere uguale per ogni coppia di persone analizzata.

Poiché il sistema di riferimento 3D in cui posizioniamo le persone è Euclideo, la distanza tra due punti qualsiasi può essere calcolata nel seguente modo:

$$D(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{21}$$

Ovvero la norma L2 della differenza tra i due punti.

Così facendo possiamo conoscere in qualsiasi istante la distanza tra tutte le persone individuate all'interno della scena, come mostrato in Figura 8.1.

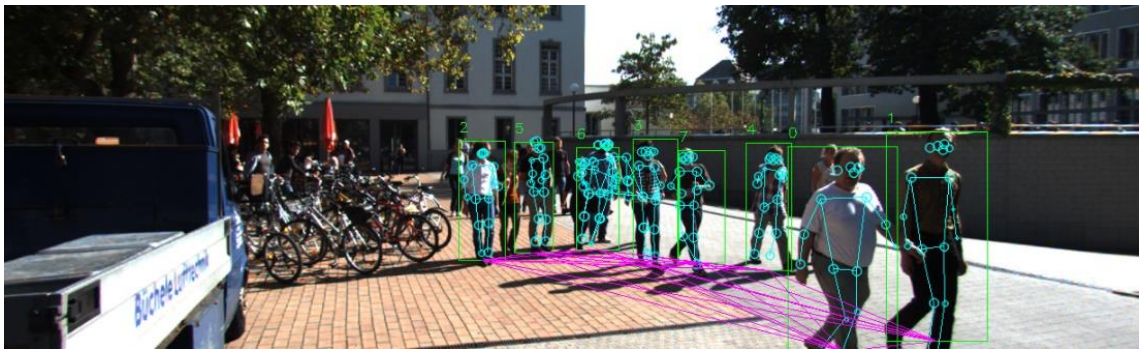

Figura 8.1 – Persone individuate nella scena con le relative distanze tracciate in magenta.

Poiché le mappe di profondità sono dense, non siamo limitati nello scegliere il punto di riferimento ai piedi delle persone, ma qualsiasi punto è utilizzabile. Di fondamentale importanza scegliere correttamente il punto di riferimento delle persone, in modo sia da evitare possibili problemi di occlusione, che per limitare l'imprecisione della misura causata dalla distorsione intrinseca delle mappe generate da reti neurali monodepth.

Alternativamente si può optare, come altri sistemi di questo tipo, di verificare che il volume cilindrico attorno alla persona, rappresentante il suo "spazio personale", non entri in contatto con gli spazi di altre persone. Poiché conosciamo le profondità, questo cilindro può essere proporzionale alla distanza e alle dimensioni della persona che racchiude.

Capitolo 9

Un sistema alternativo

In parallelo al sistema precedente, è stato sviluppato un sistema alternativo. Questo si basa sull'utilizzo di un'omografia [24] per risolvere il problema del distanziamento, in modo analogo ai sistemi alternativi discussi nel capitolo 2.

Lo sviluppo di un sistema alternativo, più simile agli altri proposti, è stato effettuato così da avere la possibilità di confrontarlo il metodo basato su reti neurali monodepth. In questo modo è possibile ricavare informazioni riguardo l'efficacia dei due metodi nelle diverse situazioni e confrontare pregi e difetti di uno e dell'altro sistema.

Per posizionare le persone all'interno della scena si cerca di sfruttare il vincolo per cui esse si trovano su di un piano. Per questo motivo, andando ad associare un sistema di riferimento metrico rispetto al piano è possibile conoscere la posizione di tutti gli elementi presenti su di esso. Questo tipo di soluzione è adatta per scenari nei quali la scena da monitorare si compone di ampi spazi piani, nei quali il piano è sempre visibile, come ad esempio, strade, piazze o ampi spazi all'interno di luoghi chiusi.

Il vantaggio di utilizzare questo metodo è che il sistema di riferimento ancorato sul piano trasforma il problema della localizzazione da 3D a 2D, poiché tutti i punti di interesse giacciono sullo stesso piano.

Per poter ottenere questo sistema di riferimento è necessario calcolare una trasformazione che permetta di proiettare i punti della scena che giacciono sul piano in un nuovo sistema di riferimento, ovvero un'omografia.

9.1 Omografia

Come abbiamo visto in precedenza nel capitolo 5, il modello pinhole ci permette di proiettare qualsiasi punto della scena 3D su di un'immagine 2D.

L'equazione (4) rappresenta tale trasformazione. Considerando il caso più semplice, in cui i punti proiettati si trovino su di uno stesso piano, l'equazione può essere semplificata considerando $Z = 0$. Otteniamo così:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{bmatrix}\begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \tag{22}$$

La proiezione prospettica è composta da una trasformazione dal 2D al 2D. Essa è formata da una matrice 3x3 tale che:

$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{23}$$

Con H:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{24}$$

In questo caso si vuole trovare quella trasformazione che va a mappare i punti del piano immagine nel piano della scena. Per fare ciò è necessario calcolare la matrice H tale che:

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = H^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{25}$$

La trasformazione così ottenuta permette di calcolare in maniera precisa le coordinate dei punti che giacciono sul piano all'interno di un sistema di riferimento metrico.

9.2 Stima della trasformazione

Per ottenere la trasformazione che permette di proiettare sul piano 3D i punti del piano immagine, è necessario capire come calcolare tale matrice H nel caso specifico dello scenario che si sta considerando.

Per poter calcolare una stima, è necessario conoscere dei punti del piano reale. Andando ad associare i punti immagine con i corrispondenti punti della scena reale, è possibile creare un sistema di equazioni che permette di ottenere una stima dei parametri di H. Sono necessari almeno quattro punti per ottenere una stima. Questi possono essere legati a caratteristiche naturali dell'immagine, come ad esempio dimensione di piastrelle del pavimento o distanze note tra oggetti statici della scena presenti sul piano, oppure, come in questo caso, utilizzando marker inseriti artificialmente all'interno dell'immagine.

Posizionando una scacchiera, di dimensione e caratteristiche note, sul piano della scena è possibile acquisire almeno quattro punti (gli angoli della scacchiera), a distanza nota tra loro.

Considerando come origine del sistema di riferimento metrico uno degli angoli della scacchiera, andando ad associare i punti immagine degli angoli con le relative coordinate metriche corrispondenti, a partire dall'angolo scelto come origine, si ottiene una trasformazione che proietta tutti i punti che giacciono sul piano in un sistema di riferimento metrico. Grazie a questa trasformazione è possibile proiettare, per mezzo dell'equazione (25), i punti di appoggio delle persone sul piano in questo sistema di riferimento e posizionarle in maniera assoluta rispetto al piano, permettendo di misurarne la distanza tra loro.

La stima di una trasformazione prospettica tra due piani viene calcolata a partire da due set di punti conosciuti. Usando un set di coordinate immagine corrispondenti agli angoli della scacchiera, e associando il set di coordinate metriche corrispondenti, è possibile ottenere un sistema di equazioni con cui stimare la matrice H che rappresenta l'omografia tra i due piani considerati.

Figura 9.1 – Scacchiera usata come rifermento del piano.

Conoscendo la dimensione metrica degli scacchi, in Figura 9.1 80mm, e il numero di righe e colonne, in figura rispettivamente sei e otto, otteniamo le coordinate di questi quattro punti a partire dal corner in basso a sinistra, considerato origine del sistema di riferimento. Nell'esempio in Figura 9.1, queste coordinate saranno perciò (0,0), (640,0), (0,480), (640,480).

La soluzione viene calcolata utilizzando il metodo robusto basato su RANSAC e, successivamente, viene utilizzato il metodo Levenberg-Marquardt per minimizzare l'errore di riproiezione.

Questo metodo basato su omografia permette in maniera semplice e affidabile di ottenere un sistema di riferimento metrico rispetto alla scena inquadrata. Perché sia preciso è fondamentale che sia la superficie su cui giace il pattern sia il pattern stesso siano planari. È sufficiente, infatti, un errore di pochi centimetri per ottenere una trasformazione completamente errata.

9.3 Segmentazione del piano

Poiché con questo metodo si è limitati a conoscere le coordinate dei punti che giacciono sul piano, è necessario sapere quali tra i punti dell'immagine facciano effettivamente parte del piano della scena, in modo da andare a discriminare se le persone stiano o meno sul piano principale. Per fare ciò, il piano è stato segmentato manualmente, creando una maschera binaria della scena, nella quale il bianco rappresenta il piano e il nero il resto della scena, come mostrato in Figura 9.2. Se non fosse disponibile discriminare se i punti stiano o meno sul piano, certi elementi, come persone riflesse o che si trovano in punti sopraelevati, verrebbero comunque considerate dal sistema di monitoraggio.

Dato che la scena, nello scenario di utilizzo, è statica e il piano non varia nel tempo, la soluzione di segmentare in maniera manuale rimane semplice ma comunque efficace. Tale maschera rimarrà sempre valida a meno di cambiamenti sostanziali della scena o nel caso di uno spostamento della telecamera. Per questo stesso motivo la scacchiera è sufficiente posizionarla sul piano in un solo frame iniziale, necessario per la stima della matrice omografica. Una volta ottenuta tale immagine, può essere rimossa dalla scena, sapendo che essa non cambierà. Nel caso in cui la telecamera possa subire dei movimenti, basarsi su caratteristiche della scena o marcatori permanenti può essere una scelta più robusta.

Inoltre, in alternativa alla segmentazione manuale si potrebbero impiegare metodi di segmentazione automatica, ma porterebbero ad un aumento di complessità non necessario.

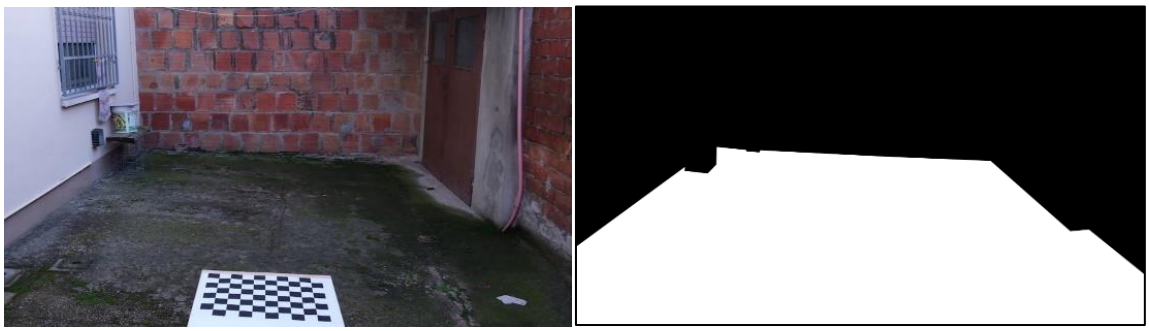

Figura 9.2 – Maschera di segmentazione del piano nella scena.

9.4 Bird-Eye View

Visualmente la proiezione produce una vista dall'alto del piano principale della scena, chiamata comunemente *Bird-Eye View* (vista a volo d'uccello). Un esempio è visibile in Figura 9.3.

I punti che fanno parte del piano della scena vengono riproiettati come se la telecamera si trovasse al di sopra del piano, mentre i punti che non ne fanno parte vengono naturalmente distorti poiché la trasformazione è valida solo per quei punti della scena appartenenti ad esso. Andando a proiettare i punti di riferimento delle persone noteremo il punto che si sposta sulla vista dall'alto, ottenendo una rappresentazione grafica della sua posizione.

Figura 9.3 – Bird-Eye view della scena e della maschera.

9.5 Proiezione e distanza tra le persone

A questo punto sono disponibili tutti gli elementi per andare a posizionare le persone sul piano e monitorare la distanza tra di esse.

Per individuare le persone viene utilizzato lo stesso modulo di people detection usato precedentemente. Una volta individuate, si sceglie il punto di riferimento della persona, un punto della bounding box o una giuntura ottenuta tramite PoseNet, e si proietta tale punto nel sistema di riferimento metrico ottenuto. In questo caso il punto utilizzato come riferimento deve giacere sul piano. In caso contrario non sarebbe possibile calcolarne la posizione. Tramite la maschera si verifica se tale punto si trova effettivamente sul piano e, in caso affermativo, si va a calcolare la distanza con le altre persone presenti nella scena.

Quest'ultima viene sempre calcolata come la norma L2 della differenza delle due coordinate utilizzando l'equazione (21). In questo caso però, dato che i punti si trovano tutti sul piano, la coordinata Z può essere omessa, poiché è fissa e uguale per tutti i punti. Le coordinate effettive delle persone sono composte soltanto da $[X, Y]$ rispetto all'origine del sistema di riferimento.

In Figura 9.4 viene mostrato un esempio del sistema in funzione.

Figura 9.4 – Individuazione delle persone nel sistema di riferimento calcolato e misura distanze.

Nel capitolo seguente verranno analizzate sperimentalmente entrambe le metodologie discusse fino ad ora per mezzo di un dataset di immagini acquisito tramite una telecamera stereo.

Capitolo 10

Risultati Sperimentali

In modo da validare i risultati ottenuti da entrambi i metodi, è stato realizzato internamente un dataset di immagini per mezzo di una telecamera stereo (Zedcam 2 [25]). In Figura 10.1 possiamo vedere un esempio dei dati disponibili in questo dataset.

Figura 10.1 – Esempio dell'immagine del dataset e relativa mappa di profondità stereo.

Si ha così a disposizione una serie di immagini con relative mappe dense di profondità da confrontare con le mappe in scala delle reti neurali monodepth. I test sono stati effettuati mettendo utilizzando le quattro reti neurali monodepth: PyD-Net, MiDaS 2.0, MiDaS 2.1 e MiDaS 2.1_small. Per prima cosa sono stati realizzati una serie di insiemi di punti fissi della scena che verranno utilizzati per la messa in scala delle mappe. A tempo di esecuzione viene tenuto conto della possibile occlusione di alcuni dei punti nel caso una persona vi sia davanti nel frame in analisi. In tal caso il punto non viene considerato per la messa in scala dell'immagine corrente. Ad ogni immagine vengono realizzate maschera e target, descritti al capitolo 7.2, utilizzando l'insieme di punti noti scelto. Successivamente si ottiene la predizione della rete neurale monodepth in analisi, la quale viene messa in scala per mezzo del metodo ai minimi quadrati. Infine, si va a calcolare l'errore medio per immagine come la differenza in valore assoluto tra la mappa monodepth scalata e la mappa di profondità stereo del relativo frame. Questa differenza viene calcolata solo nei punti in cui è disponibile il dato di profondità della telecamera stereo. Infatti, ci sono punti in cui, a causa delle condizioni di luce, poca texture o perché un oggetto è visibile solo in una delle due telecamere, tale informazione non è disponibile e perciò non è possibile andare a confrontare il risultato ottenuto con quello stereo in quelle parti dell'immagine. Un esempio dei risultati ottenuti è visibile in Figura 10.2. Nel riquadro in alto a destra viene mostrata una mappa di calore, in cui i punti che tendono al giallo rappresentano zone in cui l'errore

tra la predizione scalata della rete e la mappa stereo è superiore agli 80cm. Inoltre, si può notare come i punti che presentano l'errore maggiore si trovino per la maggior parte nella sezione alta dell'immagine. Questo succede a causa del bias intrinseco presente nelle reti neurali monodepth analizzate e discusso nel capitolo 6. Come vedremo successivamente, la messa in scala della mappa tende a mitigare questo atteggiamento, ma rimane comunque evidente.

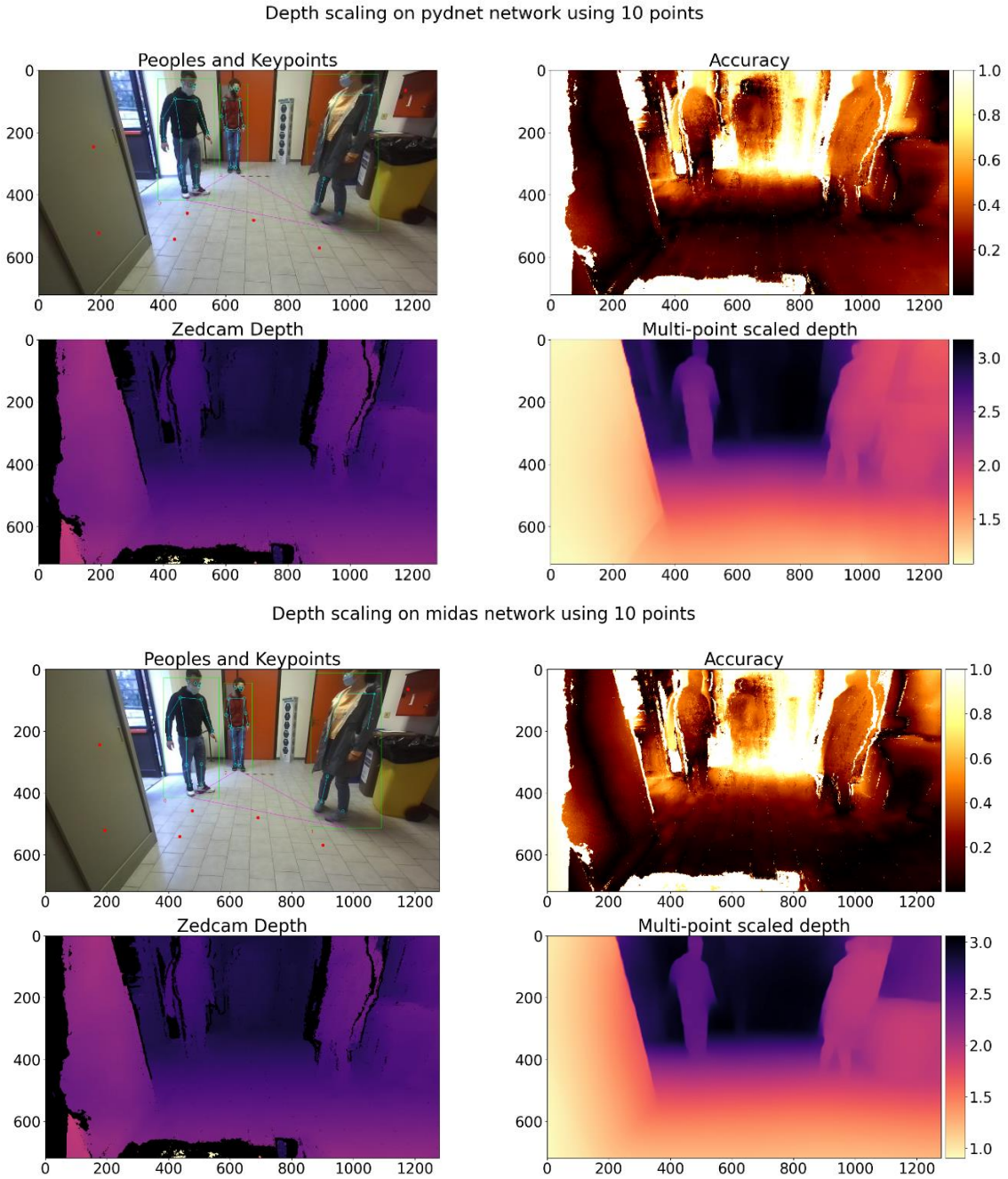

Figura 10.2 – Esempio di uscita del test eseguito sul nuovo dataset usando la rete PyD-Net e MiDaS2.1. In ogni immagine, nella colonna di sinistra troviamo: Immagine con evidenziati punti scelti e uscita rete people detection (sopra), Mappa stereo (sotto). Nella colonna di destra: Mappa di errore (sopra), Predizione in scala (sotto).

Per ottenere dati quantitativi sull'errore, è stato calcolato l'errore medio su di un totale di quattrocento immagini disponibili. Effettuando diversi test in cui veniva variato il numero di punti fissi scelti, è possibile analizzare come cambi l'errore di messa in scala al variare del numero di punti disponibili, in modo da poter valutare se ci sia una soglia oltre il quale il miglioramento, rispetto al numero di punti precedente, non è più significativo. Sono stati creati cinque set di punti, ognuno dei quali contenente gli stessi punti del precedente unito ad altrettanti punti. In Figura 10.3 sono mostrati i risultati ottenuti da questo esperimento. Come possiamo vedere nel grafico, PyD-Net e MiDaS2.1_small sorprendentemente sembrano essere le reti più stabili, con un errore medio tra i 25 e 30 cm. Insolito invece risulta il comportamento per cui all'aumentare del numero di punti fissi l'errore medio ha un andamento oscillatorio. Questo può essere causato dal fatto che in questo dataset il confronto viene fatto rispetto ad una mappa di profondità stereo, in cui potrebbero esserci delle zone particolarmente instabili. Se uno o alcuni dei punti utilizzati per la messa in scala si trovano in queste zone, questi potrebbero causare un aumento dell'imprecisione. Nonostante ciò, da questi primi test si può evincere che aumentando anche notevolmente il numero di punti per la messa in scala, ciò comporta un piccolo aumento della precisione. Sembrerebbe perciò che siano già sufficienti cinque punti stabili ben distribuiti per garantire una buona messa in scala.

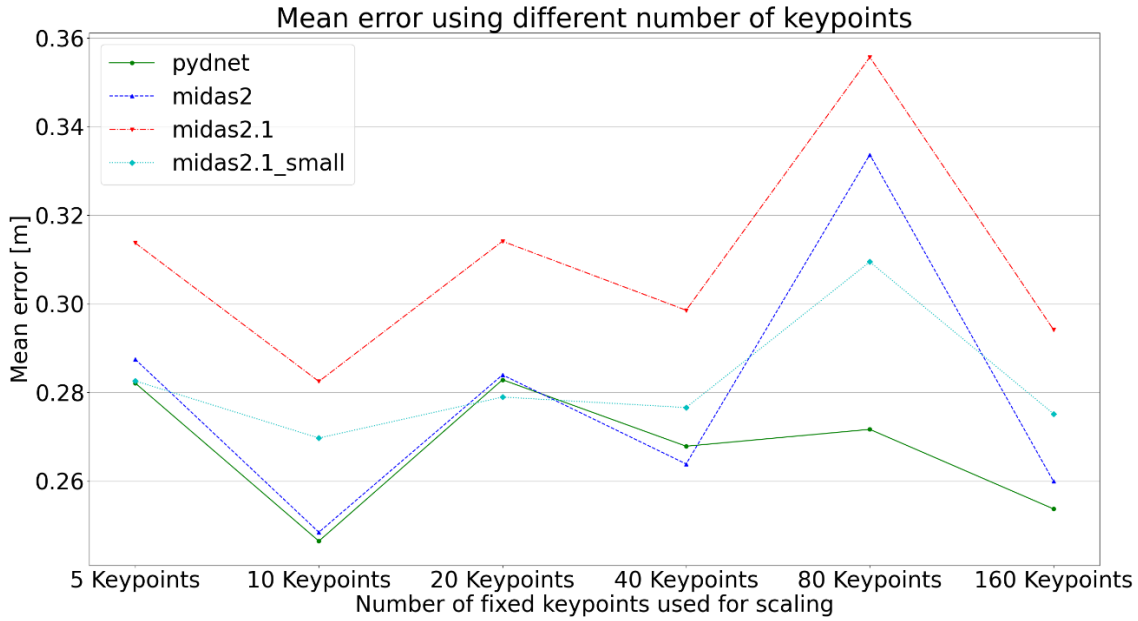

Figura 10.3 – Errore medio calcolato per ogni rete sui differenti insiemi di punti fissi. I valori rappresentati sono puntuali e non interpolati.

Oltre ad analizzare la precisione su tutta la mappa a seconda del numero di punti utilizzati, si è andati a verificare se l'errore medio variasse all'aumentare della distanza dalla telecamera. Per verificare ciò sono stati analizzati i dati ottenuti mantenendo fisso il numero di punti e

considerando solo i valori della mappa messa in scala corrispondenti a punti della mappa stereo entro certe distanze. In Figura 10.4 è visibile il grafico ottenuto analizzando tali risultati. Si può vedere come effettivamente l'errore medio aumenti considerevolmente con l'aumentare della distanza, con una media di quasi 80cm alla massima distanza, nella scena corrente, dalla telecamera. Probabilmente questo comportamento è sempre causa della distorsione intrinseca delle mappe generate. Analizzando la scena ripresa si può notare come i punti più lontani si trovino tutti nella parte alta dell'immagine, la quale è maggiormente soggetta dal bias della rete.

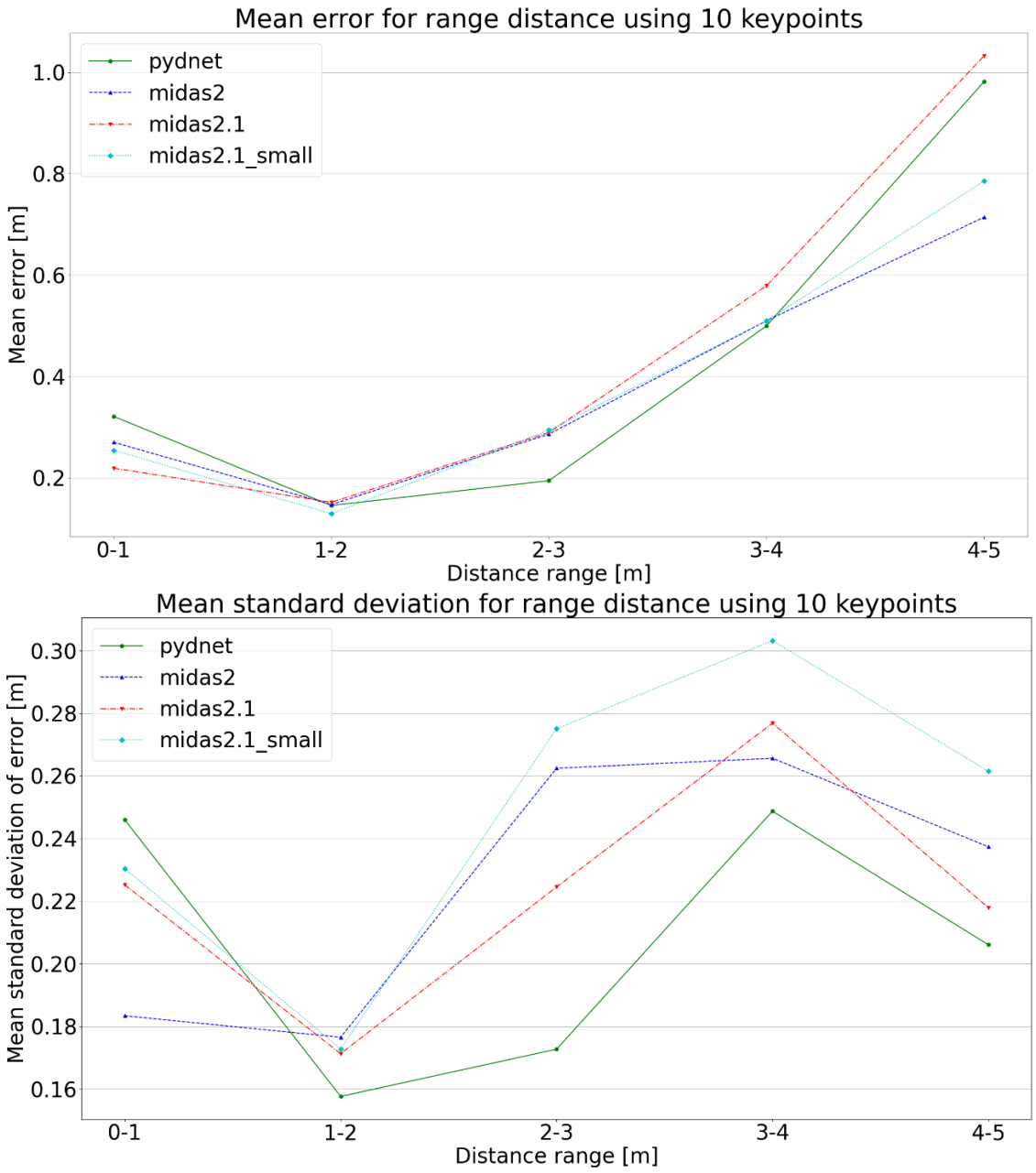

Figura 10.4 – Errore medio (alto) alle varie distanze dalla telecamera. Anche in questo caso i valori sono puntuali. La fascia massima è 4-5 metri poiché la scena ripresa ha una profondità massima di 5m. In basso è mostrata la variazione standard alle varie distanze.

Si può vedere però che la deviazione standard, nelle varie fasce considerate, sia abbastanza bassa. Questo ci porta a pensare che l'errore nelle varie fasce sia uniforme.

I risultati sperimentali ottenuti sono incoraggianti. Nonostante le mappe di profondità portino con sé una serie di problematiche, grazie alla messa in scala tramite il metodo ai minimi quadrati è possibile ottenere in media 30cm di errore rispetto ad una telecamera stereo. Seppur risulti essere di fondamentale importanza la scelta dei punti fissi da utilizzare per la messa in scala e la loro distribuzione nella scena, sapere che dieci punti sono sufficienti per ottenere una buona messa in scala è rassicurante. Ciò permetterebbe di semplificare di molto il metodo di selezione delle distanze conosciute.

È stato inoltre analizzato come l'operazione di messa in scala interagisca con il bias intrinseco delle reti. Per mezzo di Open3D sono state ottenute le viste tridimensionali dell'immagine sia utilizzando direttamente l'inferenza della rete, sia la mappa di profondità scalata.

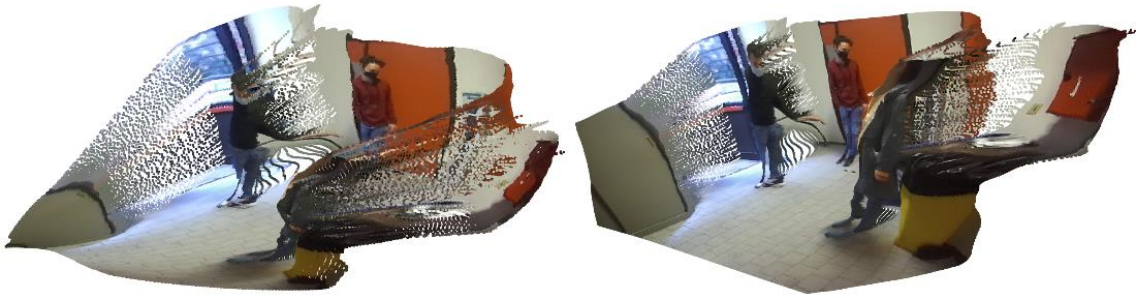

Figura 10.5 – A sinistra la visualizzazione 3D della predizione della rete MiDaS 2.1, a destra la visualizzazione della mappa in scala.

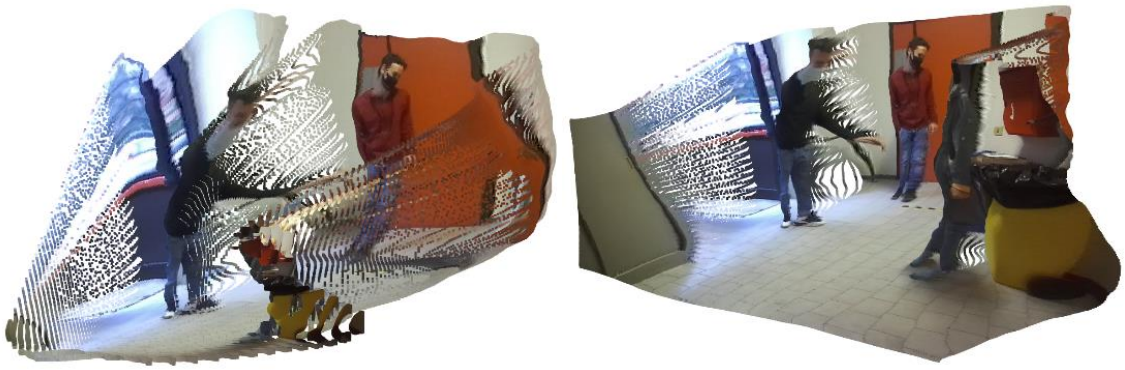

Figura 10.6 – A sinistra la visualizzazione 3D della predizione della rete PyD-Net, a destra la visualizzazione della mappa in scala.

Come possiamo vedere in Figura 10.5 e 10.6, l'operazione di messa in scala aiuta ad alleviare notevolmente la distorsione generata dal bias della rete. Ciononostante, alcune zone rimangono distorte, come si può notare nella testa della persona in nero e nella zona di destra, relativa alla

persona in grigio e ai bidoni della spazzatura. Il piano della scena sembra essere riprodotto quasi perfettamente.

Infine, sono state analizzate le performance ottenute per ogni combinazione di rete neurale monodepth e rete di individuazione delle persone. Nella seguente tabella sono mostrate le prestazioni ottenute su di una CPU Intel i5-4670K. Non si tratta di una piattaforma ideale per questo tipo di compito ma permette comunque di valutare come queste reti si comportino in combinazione tra loro.

	Tempo medio di predizione e messa in scala	**FPS Medi**	
		YOLOv3	PoseNet
PyD-Net	0,11s	2,06	0,67
MiDaS 2.0	0.71s	0,92	0,49
MiDaS 2.1	0.63s	1,20	0.63
MiDaS 2.1_small	0.13s	2,94	0,87

Come ci si poteva aspettare, dato la loro compattezza, PyD-Net e MiDaS 2.1_small risultano essere le più veloci, permettendo di avere oltre i due frame al secondo in combinazione con YOLOv3. Per quanto riguarda le reti di individuazione delle persone, si può notare come PoseNet porti una riduzione delle prestazioni di oltre il 50% rispetto a YOLOv3. Questo la renderebbe inadatta per un utilizzo in un sistema integrato. Alternative ipoteticamente più leggere, come OpenPose, potrebbero essere valutate. Questi risultati sulle prestazioni non sono sicuramente esaustivi, poiché ulteriori ottimizzazioni possono essere impiegate all'interno di questo primo prototipo. Ciononostante, permettono di avere un quadro generale sulle prestazioni delle varie reti e come interagiscono tra di loro.

In questo dataset sono state acquisite anche una serie di immagini in cui una scacchiera è stata posizionata sul pavimento (Figura 10.7). Questo ha consentito di valutare sperimentalmente la metodologia basata su omografia. Come descritto nel capitolo 9, poiché la scacchiera ha dimensioni note, viene calcolata una trasformazione omografica che permette di posizionare con coordinate metriche tutti i punti che giacciono sul piano. Andando a proiettare le ancore relative alle persone sul piano è così possibile calcolare la distanza tra di esse in maniera estremamente precisa. Come possiamo vedere in Figura 10.8, dalla sola valutazione qualitativa del risultato, usando come riferimento le piastrelle, si può già notare come la posizione proiettata sia estremamente corretta.

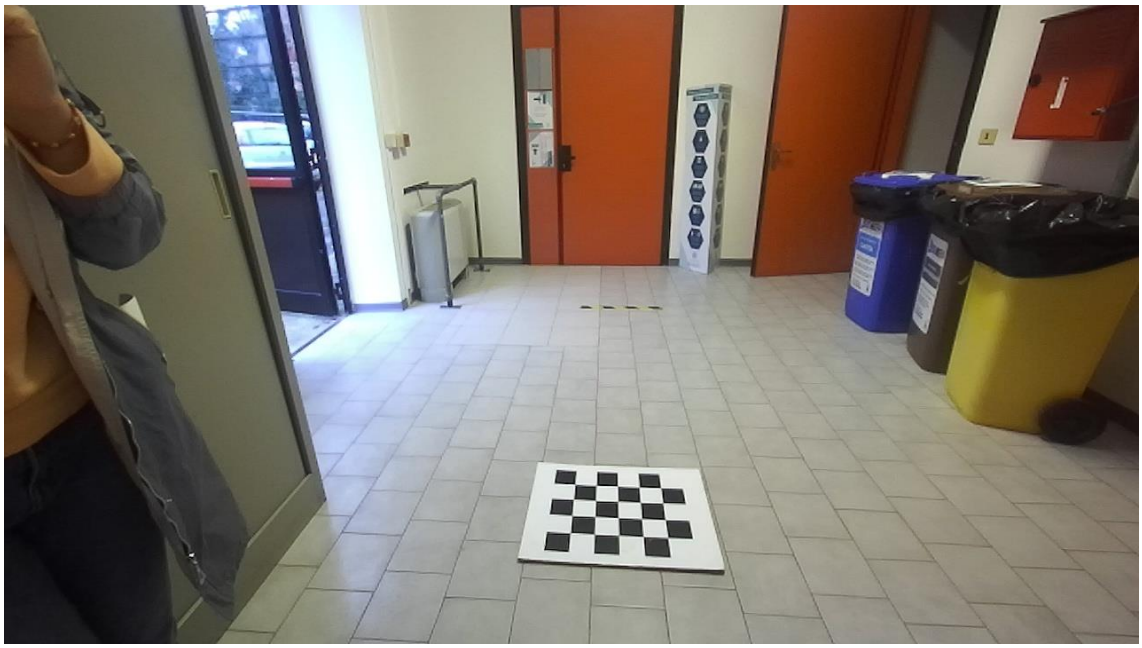

Figura 10.7 – Scacchiera posizionata sul piano.

Avendo a disposizione le mappe stereo delle varie immagini, sono state confrontate le distanze ottenute utilizzando tali profondità e quelle ottenute per mezzo di questo metodo. In media si è ottenuta una differenza tra i due valori di pochi centimetri. Un esempio è mostrato sempre in Figura 10.8. Nell'immagine di destra sono proiettati sul piano sia le posizioni ottenute per mezzo dell'omografia (blu) sia quelle ottenute tramite la profondità stereo (rosso). Possiamo vedere come siano praticamente coincidenti. Verificando su più frame si nota come non sia sempre così, ma qualitativamente la posizione più corretta sembra essere quella omografica. Questo può essere causa di una piccola incertezza della telecamera stereo, soprattutto nei punti più lontani della scena.

Figura 10.8 – A sinistra viene mostrata l'immagine originale, a destra la vista dall'alto con la texture applicata, in modo da poter vedere la posizione dei punti rispetto alle piastrelle del pavimento. Nell'immagine di destra i punti blu sono le proiezioni tramite il metodo omografico, quelle in rosso le posizioni ottenute tramite la profondità stero. In entrambe le immagini è disegnato il sistema di riferimento rispetto alla scacchiera.

I test eseguiti hanno confermato sia i pregi che le problematiche di questo metodo, evidenziate già nei capitoli precedenti.

Infine, si è confrontata la correttezza del posizionamento tra il metodo omografico e quello delle reti neurali monodepth. In Figura 10.9 è mostrata la bird-eye view della scena con texture, in cui sono stati proiettate le ancore ai piedi delle bounding box tramite omografia (blu), profondità stereo (rosso) e mappe monodepth scalate (verde).

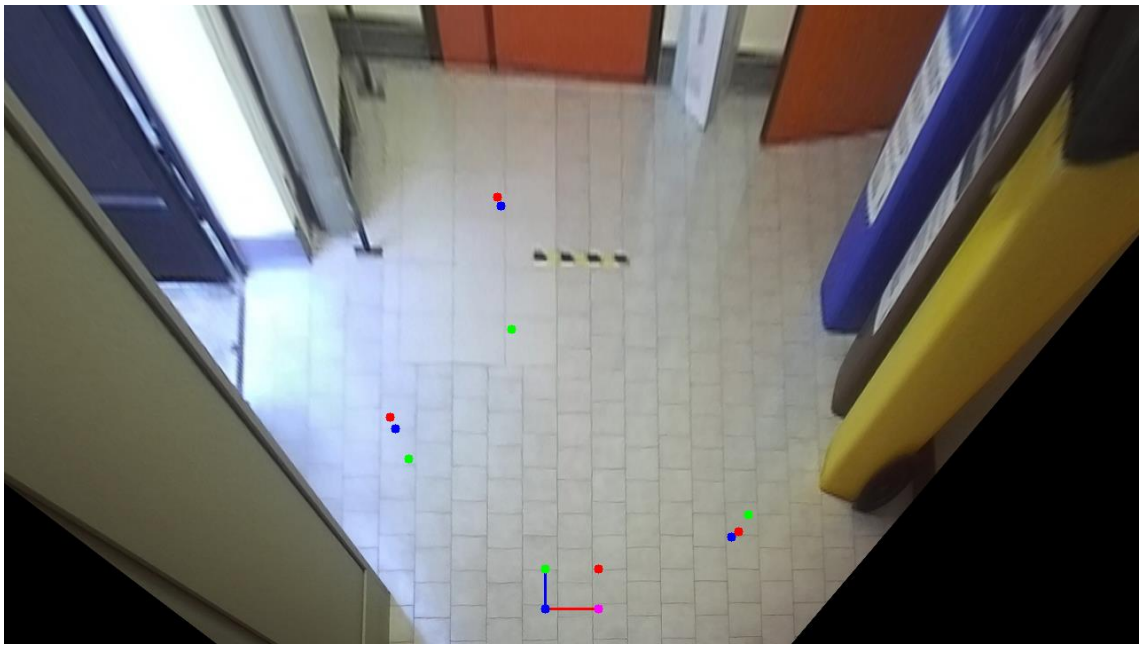

Figura 10.9 – Bird-Eye View della scena con texture, in cui sono proiettate le ancore ai piedi delle persone per mezzo di omografia (blu), profondità stereo (rosso) e mappe monodepth scalate (verde).

Questo confronto evidenzia in maniera visuale i risultati sperimentali ottenuti. Le mappe di profondità metriche acquisite per mezzo della messa in scala consentono di ottenere buoni risultati per quanto riguarda i punti relativamente vicini alla vista della telecamera, mentre l'errore aumenta rispetto agli altri due metodi man mano che ci si allontana.

Capitolo 11

Conclusioni e sviluppi futuri

Sfruttando una singola telecamera e una tecnologia innovativa come le reti neurali monodepth, si è riusciti a costruire un prototipo di sistema di monitoraggio del distanziamento sociale che possa essere installato facilmente nei luoghi in cui è già presente una infrastruttura di videosorveglianza.

Si tratta comunque di uno dei primi utilizzi reali di queste reti e del primo sistema di monitoraggio del distanziamento sociale che utilizza tale tecnologia. Le reti neurali monodepth hanno mostrato di avere una serie di problematiche non trascurabili rispetto ad altre tecnologie per la misura della profondità. Questo progetto ha però dimostrato che è possibile alleviare queste problematiche intervenendo con metodi mirati sulle mappe di profondità da loro generate. Ulteriori test rimangono necessari per valutarne il comportamento in scenari più ampi, come piazze o strade, in cui l'intervallo di distanze da analizzare è nettamente maggiore rispetto a quelli sperimentati sin ora. Il metodo omografico rimane comunque una valida alternativa nei casi in cui sia richiesta una maggiore precisione delle misure o dove le reti neurali monodepth non riescano a produrre mappe di profondità affidabili.

Sviluppi futuri di questo prototipo sono comunque possibili. Rimane tuttora da definire in che modo selezionare i punti stabili della scena da utilizzare in fase di messa in scala. Un'ipotesi potrebbe essere quella di sfruttare il sistema di riferimento ottenuto per mezzo di una omografia per ottenere distanze note sul piano e, da quelle, mettere in scala le mappe. Questo metodo potrebbe unire pregi e difetti di entrambe le soluzioni finora proposte, permettendo un monitoraggio accurato della distanza anche negli scenari in cui il piano è anche solo parzialmente visibile. Uno sviluppo futuro ulteriore consiste nel cercare di limitare il bias della parte alta della persona sfruttando il vincolo per il quale normalmente essa cammina verticalmente rispetto al piano e che la profondità della parte bassa del corpo sembra essere generalmente più precisa rispetto alla parte alta.

Se futuri test confermassero i dati ottenuti in questo progetto, si tratterebbe di un primo passo verso un utilizzo delle reti neurali monodepth all'interno di scenari di uso comune.

Bibliografia

[1] E. Famà, *Analisi della ripetibilità nella corrispondenza tra punti in immagini,* Università di Bologna, AA 2019/2020.

[2] L. Righi, *Filtro di Kalman applicato al tracciamento di persone in immagini,* Università di Bologna, AA 2019/2020.

[3] N. Rosadi, *"Detection di persone in immagini per una applicazione volta a verificare il distanziamento sociale,* Università di Bologna, AA 2019/2020.

[4] M. Cristiani, A. Del Bue, V. Murino, F. Setti e A. Vinciarelli, «The Visual Social Distancing Problem,» *IEEE Access,* vol. 8, pp. 126876 - 126886, 10 Luglio 2020.

[5] M. Aghaei, M. Bustreo, Y. Wang, G. Bailo, P. Morerio e A. Del Bue, «Single Image Human Proxemics Estimation for Visual Social Distancing,» *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV),* pp. 2785-2795, 2021.

[6] M. Fabbri, F. Lanzi, R. Gasparini, S. Calderara, L. Baraldi e R. Cucchiara, «Inter-Homines: Distance-Based Risk Estimation for Human Safety,» *arXiv:2007.10243,* 20 Luglio 2020.

[7] M. Rezaei e M. Azarmi, «DeepSOCIAL: Social Distancing Monitoring and Infection Risk Assessment in COVID-19 Pandemic,» *Applied Sciences,* vol. 10, n. 21, p. 7514, Ottobre 2020.

[8] J. Redmon, S. Divvala, R. Girshick e A. Farhadi, «You Only Look Once: Unified, Real-Time Object Detection,» *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 779-788, 2016.

[9] S. Saponara, A. Elhanashi e A. Gagliardini, «Implementing a real-time. AI-based, people detection and social distancing measuring system for Covid-19,» *Journal of Real-Time Image Processing,* 2021.

[10] A. J. Sathyamoorthy, U. Patel, Y. A. Savlee, M. Paul e D. Manocha, «COVID-Robot: Monitoring Social Distancing Contraints in Crowded Scenarios,» *arXiv:2008.06585,* 2020.

[11] I. Ahmed, M. Ahmad, J. J. Rodrigues, G. Jeon e S. Din, «A deep learning-based social distance monitoring framework for COVID-19,» *Sustainable Cities and Society,* vol. 65, p. 102571, Febbraio 2021.

[12] M. E. Rusli, M. Ali, S. Yussof e A. A. A. Hassan, «MySD: A Smart Social Distancing Monitoring System,» *2020 8th International Conference on Information Technology and Multimedia (ICIMU),* pp. 399-403, 2020.

[13] M. Poggi, F. Aleotti, F. Tosi e S. Mattoccia, «Towards real-time unsupervised monocular depth estimation on CPU,» *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* 2018.

[14] «OpenCV,» [Online]. Available: https://opencv.org/.

[15] «Tensorflow,» [Online]. Available: https://www.tensorflow.org/.

[16] «PyTorch,» [Online]. Available: https://pytorch.org/.

[17] R. Ranftl, K. Lasinger, D. Hafner, K. Shindler e V. Koltun, «Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer,» *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI),* 2020.

[18] «The KITTI Vision Benchmark Suite,» [Online]. Available: http://www.cvlibs.net/datasets/kitti/.

[19] M. Menze e A. Geiger, «Object Scene Flow for Autonomous Vehicles,» *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[20] A. Geiger, P. Lenz, C. Stiller e R. Urtasun, «Vision meets Robotics: The KITTI Dataset,» *International Journal of Robotics Research (IJRR)*, 2013.

[21] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler e K. Murphy, «Towards Accurate Multi-person Pose Estimation in the Wild,» *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4903-4911, 2017.

[22] «Documentazione OpenCV - Calibrazione,» [Online]. Available: https://docs.opencv.org/master/d9/d0c/group__calib3d.html#ga3207604e4b1a1758aa66acb6ed5aa65d.

[23] «Open3D - A Modern Library for 3D Data Processing,» [Online]. Available: http://www.open3d.org/.

[24] R. Hartley e A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2000.

[25] «Zed Camera by Stereolabs,» [Online]. Available: https://www.stereolabs.com/zed/.