

UNIVERSITÀ DEGLI STUDI DI PISA



Scuola di Ingegneria

Corso di Laurea specialistica in Ingegneria Informatica

STUDIO E PROGETTAZIONE DI UNA SMART CAMERA PER LA RILEVAZIONE DEL TRAFFICO

RELATORE

Prof. Luca Fanucci

CANDIDATO

Tommaso Innocenti

CORRELATORE

Prof. Marco Avvenuti

SOMMARIO	5
1. INTRODUZIONE	6
2. STATO DELL'ARTE.....	9
2.1. Rilevatori basati su sensori di pressione	10
2.2. Contatori basati su radar	11
2.3. Contatori basati su scanner laser	13
2.4. Contatori basati su smart camera	16
2.4.1. Atalaya	17
2.4.2. TraffiCam	19
3. HARDWARE: MOUSECAMERA	22
3.1. Descrizione del sistema.....	22
3.2. Architettura	25
3.2.1. Sensore ADNS-3060	26
3.2.2. Microprocessore	27
3.2.3. Trasmettitore Radio	28
3.3. Principio di funzionamento	29
4. MIDDLEWARE	31
4.1. Elaborazione dell'immagine	31
4.1.1. Pedestal Correction	31
4.1.2. Gamma Correction	33

4.2.	Protocollo di rete	34
4.3.	Algoritmo di motion estimation.....	41
5.	IMPLEMENTAZIONE DEL SOFTWARE PER LA RILEVAZIONE DEL TRAFFICO	44
5.1.	Calibrazione dell'immagine.....	44
5.2.	Implementazione dell'algoritmo di ME: i 3 motori.....	46
5.3.	Riconoscimento degli oggetti	49
5.3.1.	Erosion	49
5.3.2.	Labeling.....	51
5.3.3.	Snake	54
5.4.	Classificazione dei veicoli	56
5.4.1.	Pedoni.....	57
5.4.2.	Cicli	58
5.4.3.	Automobili	59
6.	ANALISI DEI RISULTATI.....	60
7.	CONCLUSIONI	68
8.	Bibliografia	70
9.	Tabella dei contenuti.....	72

SOMMARIO

Questa tesi presenta uno studio per la progettazione di una tecnologia di rilevazione del traffico stradale basata su smart camera. L'hardware utilizzato è la MouseCamera, una smart camera a bassa risoluzione, basata sul sensore per immagini dei mouse ottici, scelta per il suo costo e le ridotte grandezze. La rilevazione del traffico è effettuata attraverso l'algoritmo di stima 3D-RS, implementato e ottimizzato per l'hardware scelto. Dalla matrice di stima del movimento vengono riconosciuti gli oggetti in movimento attraverso algoritmi implementati ad hoc e classificati a seconda della tipologia di veicolo attraverso i parametri di area, velocità e rapporto altezza/lunghezza. I collaudi effettuati mostrano uno scostamento dal conteggio reale dei veicoli di circa il 10%.

1. INTRODUZIONE

I sistemi di informazione sul traffico mirano a fornire messaggi affidabili e veloci, per consigliare i conducenti sulle condizioni stradali e le congestioni. Per generare informazioni sul traffico, possono essere utilizzate diverse fonti di dati, quali segnalazioni di utenti, sensori di traffico o la raccolta di dati nei sistemi di pedaggio. Oltre a questi metodi, gli operatori stradali sfruttano la videosorveglianza per riconoscere situazioni di interesse.

L'analisi video automatizzata è la metodologia più usata per rilevare situazioni quali ingorghi o condizioni stradali che impediscono il transito. I dati video possono essere analizzati sia centralmente, in un server remoto sia decentrati sui dispositivi locali (smart camera). Molti operatori autostradali possiedono una vasta installazione di semplici telecamere di sorveglianza, che forniscono i flussi video a server remoti, dove può essere applicata l'analisi senza ulteriori installazioni a bordo strada. Tuttavia, soluzioni centralizzate presentano l'inconveniente di avere alti requisiti di comunicazione e una bassa scalabilità. Al contrario, la soluzione decentrata effettua l'analisi dei flussi video in locale, e li trasmette unicamente per verifica.

Questa opera descrive il lavoro effettuato durante un tirocinio di sei mesi presso la "VITO NV" a Mol in Belgio, una organizzazione indipendente e orientata al cliente, che offre soluzioni tecnologiche innovative, consulenza e supporto tecnologico a clienti quale il governo del Belgio. Il gruppo di lavoro interessato si occupa di tecnologie di reti di sensori ottici, volte all'assistenza

agli anziani, alle case intelligenti (smart home), alla videosorveglianza e al controllo dei flussi di traffico stradale.

La problematica studiata nasce dalla necessità, da parte del gruppo di lavoro, di effettuare delle misurazioni di conteggio e classificazione del traffico su strada nell'area metropolitana di Anversa. Per le rilevazioni, venivano utilizzati sensori pneumatici, in quanto portabili, che però hanno un alto costo (circa 5'000 euro). Da qui è nata la necessità di cercare un modo alternativo, a basso costo e portabile, per effettuare queste misurazioni.

Nel secondo capitolo, dedicato allo stato dell'arte dei sistemi di rilevazione del traffico, verranno descritte alcune tecnologie di vario tipo, basate su vari tipi di sensori. Verranno anche illustrati alcune tecnologie di smart camera progettate per questo scopo.

Nel terzo capitolo verrà descritta l'architettura dell'hardware scelto, la MouseCamera, ed il suo principio di funzionamento. Questa smart camera è stata ideata con l'obiettivo di sviluppare una tecnologia di sensore visivo ad alta mobilità e basso costo, che sono proprio due dei punti critici nell'utilizzo di questo tipo di sensori in aree aperte, poiché soggette a incidenti, furti e atti di vandalismo. Verranno anche approfondite le tecnologie dei vari componenti della MouseCamera: sensore ottico, microprocessore, dispositivi di comunicazione.

Nel quarto capitolo si introdurranno le varie implementazioni software utilizzate per il filtraggio del rumore nelle immagini rilevate dal sensore ottico, il protocollo di comunicazione tra più MouseCamera progettato per

possibili comunicazioni in una rete di smart camera e la descrizione del principio di funzionamento dell'algoritmo di *motion estimation 3D-RS*.

Nel quinto capitolo è descritta la progettazione del software per l'utilizzo della MouseCamera come rilevatore del traffico. Verranno descritte le scelte implementative per l'algoritmo di motion estimation applicato alla strada e per il riconoscimento di oggetti. Successivamente verranno espone le tecniche per il conteggio e la classificazione dei veicoli stradali.

Nel sesto capitolo vengono riportati i risultati dell'utilizzo della MouseCamera come rilevatore del traffico, la loro analisi e alcune delle possibili migliorie per sviluppi futuri.

2. STATO DELL'ARTE

I rilevatori di traffico sono dei dispositivi, spesso di natura elettronica, utilizzati per contare, classificare o misurare la velocità di passaggio del traffico veicolare lungo un carreggiata. Il dispositivo è di solito implementato nei pressi della carreggiata e utilizza un mezzo intrusivo, come tubi pneumatici stradali stesi, sensori piezoelettrici incorporati, spire tagliate, o una combinazione di questi sulla carreggiata per rilevare i veicoli che transitano.



FIGURA 2-1 FOTOGRAFIA DI SENSORE INVASIVO

Una delle prime unità di conteggio del traffico, chiamato registratore di traffico, è stato introdotto nel 1937 e si trattava di un nastro steso attraverso la carreggiata, alimentata da una batteria a 6V.

Recentemente, nell'interesse della sicurezza e della facilità di installazione dei lavoratori, sono state sviluppate tecnologie non intrusive. Questi dispositivi utilizzano generalmente una sorta di energia trasmessa come onde radar o raggi infrarossi per rilevare veicoli passando sopra la carreggiata.

2.1. Rilevatori basati su sensori di pressione

I rilevatori basati su sensori di pressione sono costituiti da un tubo sdraiato attraverso la strada e un controllore, che immagazzina le informazioni e le elabora.



FIGURA 2-2 FOTOGRAFIA DI RILEVATORE BASATO SU SENSORI DI PRESSIONE

Questo tipo di dispositivo conta ogni volta che viene rilevata una pressione nel tubo, quindi nel caso di veicoli questo avviene per ogni asse di ruote.

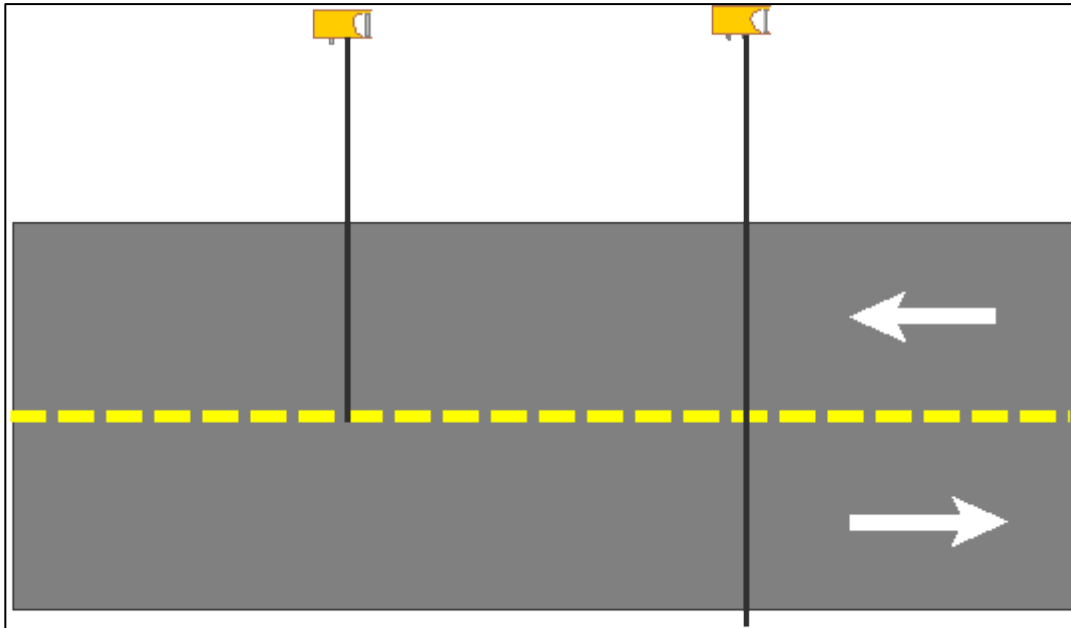


FIGURA 2-3 SCHEMA DI UTILIZZO DEL SENSORE A PRESSIONE

Questo tipo di rilevatore è invasivo e non è raccomandato per l'uso in condizioni di neve o ghiaccio. Può classificare i veicoli in base al peso, il numero di ruote, alla lunghezza e alla velocità dei veicoli. La classificazione dei veicoli non viene fatta on-line ma a posteriori, elaborando i dati immagazzinati durante le rilevazioni.

2.2. Contatori basati su radar

I contatori basati su radar sono un tipo di tecnologia non invasiva per il controllo del traffico. Questi possono essere situati a lato della strada o in una posizione più alta rispetto a questa.



FIGURA 2-4 FOTOGRAFIA DI CONTATORE BASATO SU RADAR

La tecnologia radar raccoglie volume, velocità e lunghezza per la classificazione dei veicoli. Ha un campo di rilevamento di circa 75m con la capacità di rilevare fino a dieci corsie di traffico contemporaneamente.

I rivelatori radar hanno un principio di funzionamento basato su un fenomeno fisico noto come effetto "Doppler": quando le onde radio emesse da una sorgente fissa vengono riflesse da un oggetto in movimento, la frequenza delle onde riflesse è diversa da quella delle onde emesse. Questa deviazione di frequenza è direttamente proporzionale alla velocità del bersaglio in movimento. Inoltre a seconda se l'oggetto si muove verso la sorgente o lontano da esso, questa differenza sarà positiva o negativa.

Un rilevatore radar utilizza onde ad alta frequenza in trasmissione e ricezione (24.125 GHz) che analizzano costantemente la deviazione di frequenza tra le onde emesse e quelle riflesse. Si misura tramite queste la velocità e la direzione dell'oggetto in movimento. Utilizzando sofisticati algoritmi e tecniche di analisi del segnale radar è possibile anche misurare la lunghezza dei veicoli ai fini della classificazione.

2.3. Contatori basati su scanner laser

I contatori basati su scanner laser calcolano la velocità e la lunghezza dei veicoli dal tempo necessario per rompere i due fasci e per quanto tempo i fasci sono perturbati.



FIGURA 2-5 FOTOGRAFIA DI CONTATORE BASATO SU SENSORI LASER

Questi calcoli vengono poi memorizzati in una memoria esterna con ora e data [9].



FIGURA 2-6 SCHEMA DI UTILIZZO DEI CONTATORI CON TECNOLOGIA LASER

I contatori laser sono soggetti a una percentuale di cancellazioni di lettura, nel caso che vengano installati su strade con più corsie.

Questo è dovuto al fatto che più veicoli possono passare davanti al rivelatore allo stesso tempo. Esistono dei rilevatori laser in grado anche di effettuare la classificazione dei veicoli.



FIGURA 2-7 FOTOGRAFIA DI UN CONTATORE LASER IN GRADO DI CLASSIFICARE I VEICOLI

In questo caso lo scanner laser è utilizzato per la scansione parallela della strada lungo una corsia. I veicoli sono esaminati nel loro asse longitudinale [8].

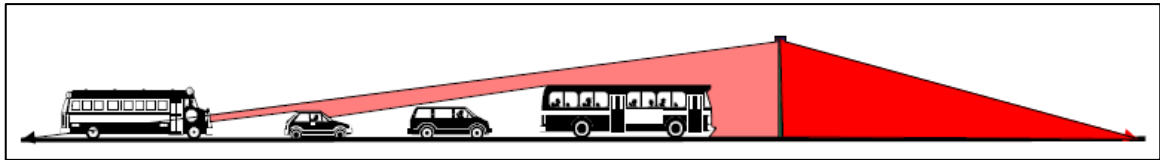


FIGURA 2-8 SCHEMA DI UTILIZZO DI UN CLASSIFICATORE LASER

La lunghezza e l'altezza ottenuta permettono la classificazione in diversi gruppi di veicoli. L'uso aggiuntivo di un secondo scanner permette di descrivere completamente la struttura dei veicoli.

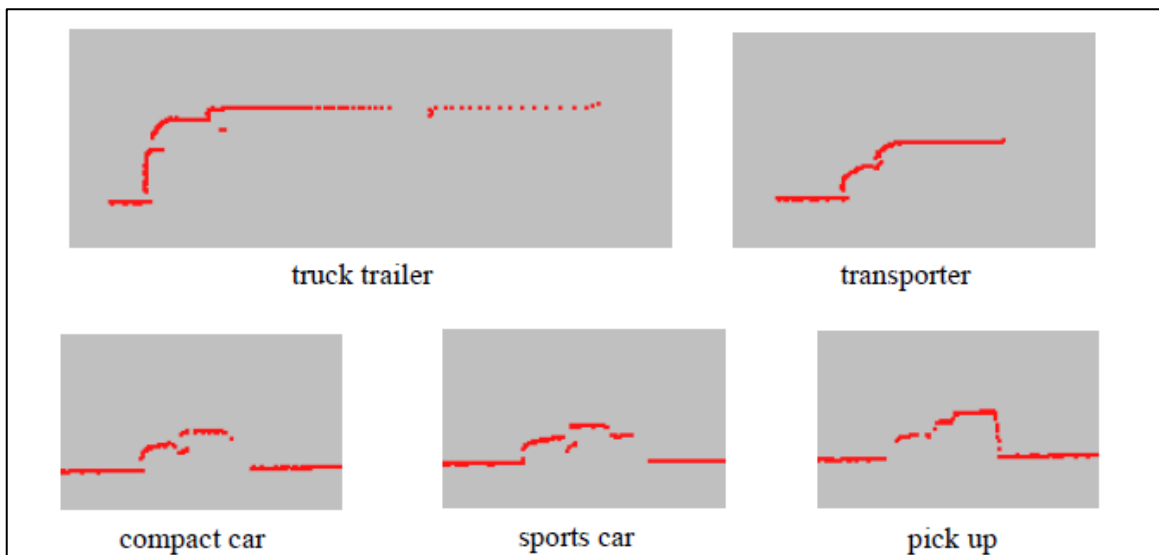


FIGURA 2-9 ESEMPI DI RISULTATI DELLA CLASSIFICAZIONE TRAMITE CONTATORE LASER

Questo permette una classificazione più dettagliata di: tipi di camion, auto, moto, ciclisti, pedoni.

2.4. Contatori basati su smart camera

I sistemi distribuiti di Smart Camera hanno ricevuto l'attenzione della comunità di ricerca nel corso degli ultimi anni. L'utilizzo di telecamere in combinazione con la potenza di calcolo incorporata, interconnesse attraverso collegamenti radio ha aperto una nuova area di interesse.

L'elaborazione di immagini in *realtime* e l'elaborazione distribuita resa possibile dalle reti di Smart Camera non solo possono migliorare le applicazioni esistenti, ma anche darne vita di nuove. I possibili campi di utilizzo vanno dal monitoraggio domiciliare all'assistenza agli anziani, ambienti intelligenti per la sicurezza e sorveglianza in edifici pubblici o aziendali. I punti di forza, che influenzano il successo delle implementazioni delle Smart Camera, comprendono un funzionamento affidabile e robusto con la minor manutenzione possibile.

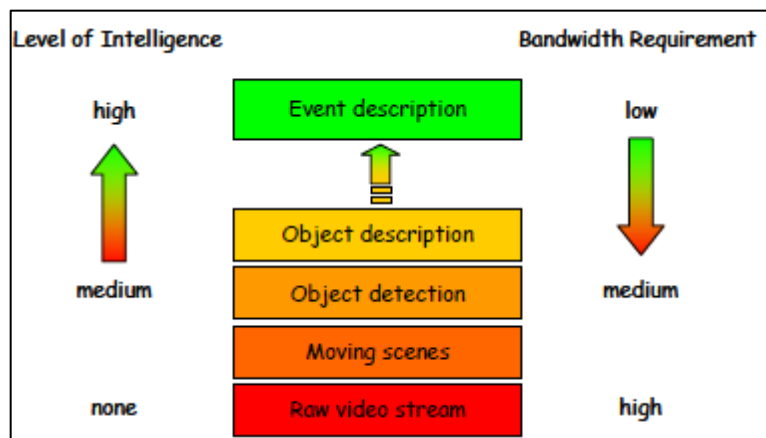


FIGURA 2-REQUISITI DI BANDA PER I LIVELLI DI ELABORAZIONE NELLE RETI
DISTRIBUITE DI SENSORI VISIVI

Rispetto ai sensori scalari, ad esempio di temperatura, pressione, umidità, velocità e accelerazione, i sensori visivi generano dati con una banda molto più elevata, a causa della natura bidimensionale della matrice di pixel dell'immagine. La quantità di dati grezzi generati impedisce l'analisi umana in molte applicazioni. Gli algoritmi dei sistemi distribuiti, supportati dall'elaborazione delle immagini su ogni nodo sono quindi necessarie per lavorare con successo su reti scalabili di Smart Camera senza fili.

2.4.1. Atalaya

Atalaya è un prodotto della Imagsa ed è un sistema di controllo del traffico basato su una smart camera [7].

E' una fotocamera attivata automaticamente per eseguire il riconoscimento automatico delle targhe di ogni veicolo contemporaneamente su 2 corsie e in tempo reale.



FIGURA 2-10 FOTOGRAFIA DELLA SMART CAMERA ATALAYA

Atalaya utilizza sensori di immagine ultrarapidi (250 fps) e processori di elaborazione in parallelo (FPGA), con grandi vantaggi sulle prestazioni, dimensioni, consumo e costo della soluzione.

Nel proprio hardware questo dispositivo integra il sensore di targa ultra-veloce Mercury Traffic, un illuminatore ad alta potenza IR e una CPU programmabile. Tutti questi componenti sono racchiusi in un unico involucro e non è quindi necessaria nessuna attrezzatura stradale supplementare e i costi complessivi di integrazione sono ridotti.

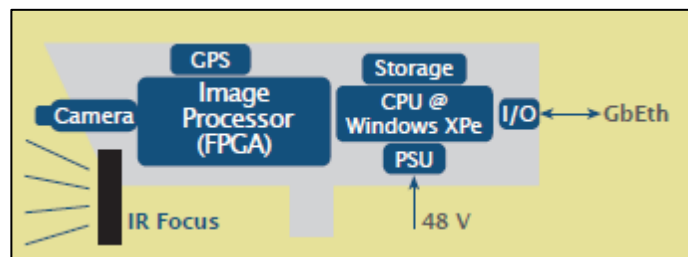


FIGURA 2-11 ARCHITETTURA DELLA SMART CAMERA ATALAYA

Di seguito il principio di funzionamento. Il sensore ad alta velocità cattura più di 250 immagini al secondo, garantendo che la targa del veicolo sia sempre catturata alla stessa distanza dalla fotocamera, senza degradazione della risoluzione dell'immagine (pixel / cm) causata dal movimento del mezzo. Parallelamente all'acquisizione delle immagini, una CPU (ATOM processor @1.6GHz RAM: 1 GByte) le elabora attraverso un motore OCR (Optical Character Recognition) per rilevare e memorizzare la targa.

Per limitare il rumore dovuto alla luminosità, Atalaya integra una tecnica di multi-esposizione, che utilizza, tra le altre variabili, i dati di un GPS. La posizione del sole rispetto all'orientamento della fotocamera viene monitorato e aggiornato ogni secondo in modo da conoscere i dati sulla riflessione delle targhe.

2.4.2. *TraffiCam*

TraffiCam[13] è una smart camera progettata a Sydney per applicazioni di videosorveglianza sul traffico per aree metropolitane. È basata su un algoritmo di rilevamento di eventi a tempo reale, progettato appositamente per essa.

Questa smart camera è stata progettata utilizzando tecnologia FPGA, e si propone di rilevare e monitorare, in tempo reale, veicoli in movimento sulla strada, automaticamente ed autonomamente, fornendo ad alto livello informazioni sul flusso di traffico quale la presenza di congestione, la lunghezza delle code ed il conteggio di veicoli agli incroci di strade principali o su autostrade.

TraffiCam è una smart camera costruita utilizzando un sensore di immagini CMOS da un Mega-Pixel, un singolo FPGA per l'analisi video e una unità di comunicazione in uscita. Gli algoritmi di elaborazione delle immagini, sono progettati su misura e applicati sui video grezzi ad alta qualità catturati dal sensore di immagine. Il costo di questa smart camera si aggira sui 300 \$.

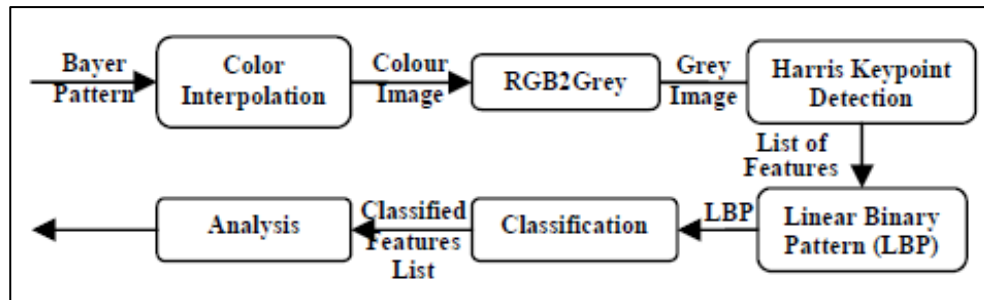


FIGURA 2-12 DIAGRAMMA A BLOCCHI DEL FLUSSO DI PROCESSO DI TRAFFICAM

La figura 2-12 mostra il flusso di processo su cui si basa Trafficcam, dall'aquisizione RGB, all'analisi finale dal punto di vista implementativo dell'FPGA. L'elaborazione dell'immagine e la sua conversione avviene tramite l'interpolazione di colore del modulo RGB2Grey, mentre il rilevamento dei punti contorni dell'immagine, descrizione e classificazione sono eseguite utilizzando l'algoritmo 'Harris Keypoint', utilizzato per la rilevazione di contorni, il modulo LPB e quello di classificazione. La stima della velocità e della densità del traffico viene poi effettuata dal modulo di analisi.

I risultati ottenuti in termini di rilevazione del traffico sono orientati alla densità e alla velocità del traffico. I possibili valori rilevati per la densità sono:

- Empty
- Low
- High

e per la velocità:

- Zero

- Slow
- Smooth

3. HARDWARE: MOUSECAMERA

Il gruppo di lavoro sulle reti distribuite di sensori di VITO, si prefiggeva come obiettivo quello di progettare una tecnologia di sensore visivo ad alta mobilità [1]. Sono stati utilizzati quindi sensori a bassa risoluzione (ADNS-3060), che forniscono una bassa quantità di dati per immagine (675 byte) e quindi consentono di utilizzare microprocessori dsPIC, adatti per sistemi embedded, con buone prestazioni ed un basso consumo di potenza. Per permettere l'utilizzo della MouseCamera in reti distribuite, è stato utilizzato un ricetrasmittitore radio per le comunicazioni wireless.

3.1. Descrizione del sistema

Volendo progettare una smart camera di piccole dimensioni, basso consumo di potenza ed, allo stesso tempo, economica, è stato necessario raggiungere un compromesso con le prestazioni.

Uno dei motivi per la scelta del sensore, l'ADNS-3060, è il suo basso costo (circa 6\$), essendo questo progettato per l'utilizzo nei gaming mouse senza fili.

L'ADNS-3060 rileva fino a 6400 immagini al secondo, con una risoluzione di 30x30 pixel (675 byte), di forma quadrata con un range dinamico di 6 bit in scala di grigi. La bassa risoluzione dei frame rilevati permette la non violabilità della privacy, essendo impossibile il riconoscimento dei volti.

Al sensore è stato affiancato il microprocessore dsPIC33FJ128GP802. Questo processore, particolarmente utilizzato nei sistemi embedded, comunica con il sensore tramite l'interfaccia seriale. Questa interfaccia limita il frame rate del sensore a 80 fps. Il microprocessore ha una memoria in grado di immagazzinare fino a 17 immagini (16 KByte di RAM). Il suo costo si aggira sui 6\$, in linea con quello del sensore. La presenza di un modulo DMA permette di dedicare più cicli di elaborazione al programma utente, lasciando il compito al DMA di comunicare parallelamente con i dispositivi esterni al microprocessore.

Per le comunicazioni, la MouseCamera rende disponibile una periferica di connessione wireless ed un collegamento seriale verso un PC.

Il collegamento wireless viene effettuato tramite il modulo RFM12B, un trasmettitore FSK a bassa potenza. Questo può comunicare su vari canali alla frequenza di 433, 868 e 915 MHz. Il suo frame rate raggiunge valori fino a 115Kbit al secondo (fino a 10 immagini al secondo, considerando anche le comunicazioni di sincronizzazione), ed un range massimo di 30 metri in interni. E' collegato al microprocessore tramite la porta seriale utilizzando il DMA, analogamente al collegamento con il sensore.



FIGURA 3-1 FOTOGRAFIE DELLA MOUSECAMERA

Per visualizzare le immagini o inviare dati ad un'interfaccia utente è invece necessario un collegamento con un PC. Questo avviene tramite l'interfaccia UART, che comunica i risultati all'utente sempre tramite il controllore DMA, con una velocità massima di 115Kbps (massimo 16 immagini al secondo).

L'alimentazione è controllata da un convertitore DC/DC, che permette alla batteria di scendere fino a 0,3 V, mantenendo però 3,3V di alimentazione per il sistema.

Il tutto, compreso una lente che viene montata sul sensore, ha una grandezza di 62x41x20 mm. Il costo totale è sotto i 25 euro, il che permette la costruzione di reti di questi sensori con una spesa limitata. Il basso costo del sistema consente il suo utilizzo anche in ambienti in cui la perdita del sensore a causa di furti, atti vandalici e altro è possibile.

Il consumo di potenza totale del sistema va dai 3 mW ai 240 mW, a seconda dell'utilizzo delle periferiche di connessione (in sleep mode il consumo è di 3 mW).

3.2. Architettura

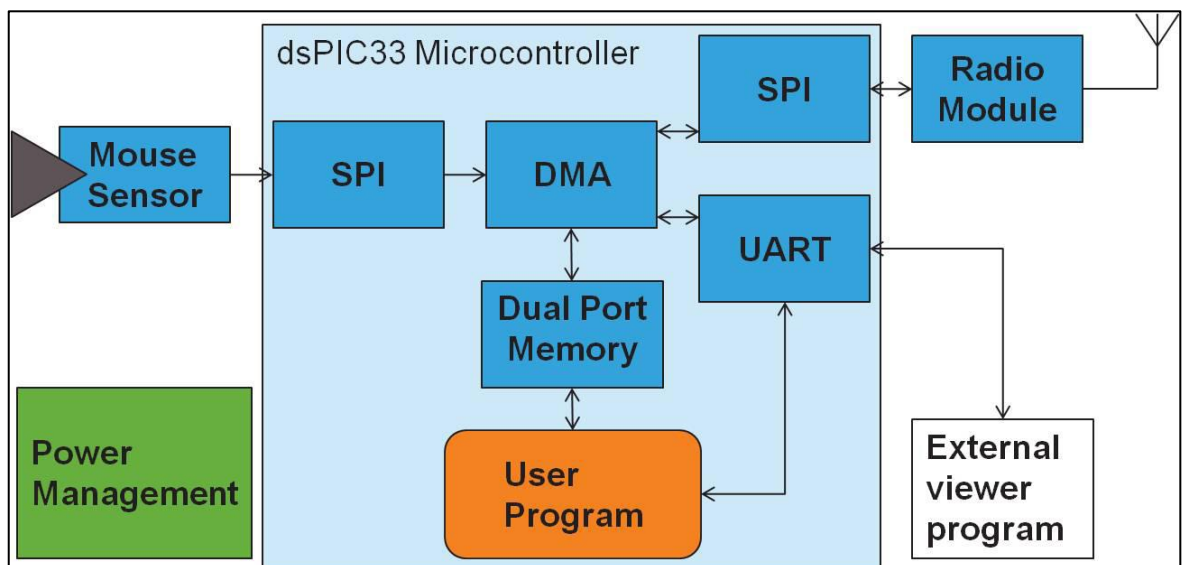


FIGURA 3-2 DISEGNO DELL'ARCHITETTURA DELLA MOUSECAMERA

3.2.1. Sensore ADNS-3060

Il ADNS-3060 è un sensore ottico ad alte prestazioni prodotto da Avago, progettato per l'utilizzo nei mouse, ed in particolare nei gaming-mouse senza fili [4].

E' capace di rilevare movimenti ad alta velocità, fino a 40 pollici al secondo con accelerazioni fino a 15g.

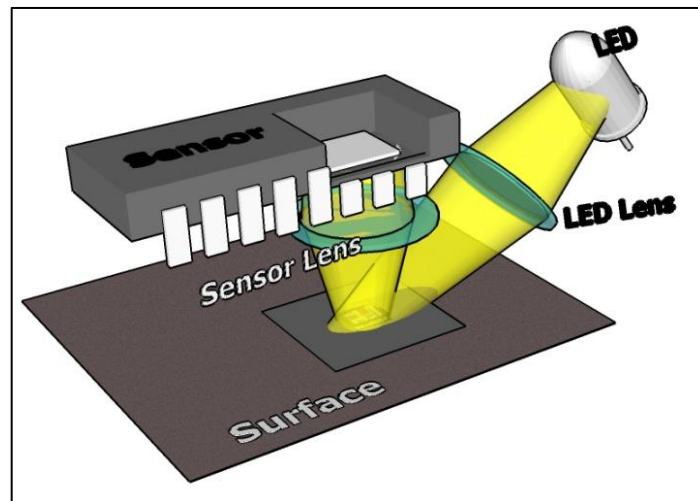


FIGURA 3-3IMMAGINE DEL SENSORE OTTICO

Il suo principio di funzionamento è basato su una tecnologia chiamata “Optical Navigation Technology”, la quale misura gli spostamenti acquisendo una serie di immagini della superficie e determinando matematicamente la direzione e l'intensità del movimento.

Contiene un sistema di acquisizione di immagini (IAS, Image Acquisition System), un DSP (Digital Signal Processor) per calcolare gli spostamenti e una porta four-wire per le comunicazioni.

Oltre al calcolo degli spostamenti, il sensore è in grado di catturare immagini di 30x30 pixel. Ogni pixel è rilevato con 6 bit in scala di grigio. Il frame rate raggiungibile è di 6400 fps con un consumo di potenza di 15 mW in modalità attiva e 1 mW in modalità Idle.

Il vantaggio nell'utilizzo di questo tipo di sensore per mouse è che lavora con una tensione di 3,3 V, in accordo con il resto dei componenti della MouseCamera, invece che con i soliti 5 V dei mouse ottici USB.

L'immagine catturata è letta in "burst-mode" usando l'interfaccia seriale SPI. Un insieme di registri interni danno le informazioni relative a misurazioni quali:

- scostamento Δx e Δy ;
- misure sulle intensità dei pixel (Pixel_Sum, Maximum_Pixel);
- configurazioni dell'otturatore (shutter).

In normali modalità di funzionamento, il sensore effettua un controllo automatico dell'esposizione, garantendo un'immagine nitida configurando automaticamente l'otturatore (shutter).

Nonostante l'alto frame rate del sensore (6400 fps), il protocollo di trasferimento della porta seriale limita la velocità a circa 80 fps.

3.2.2. *Microprocessore*

La MouseCamera è costruita intorno a un DSP prodotto da Microchip, il dsPIC33FJ128GP802 [5].

Questo è strutturato con una doppia architettura Harvard (sia le istruzioni che gli operandi hanno il proprio bus di memoria), in modo da rendere più veloci le istruzioni di image processing, in confronto ad altri dispositivi con architettura Von Neumann, ad esempio l'ARM7.

Il processore DSP lavora a 16 bit, in accordo con gli algoritmi di image processing utilizzati, senza avere l'overhead ed il consumo di potenza che hanno i processori a 32 bit.

Con un clock di 80 MHz, il processore raggiunge le prestazioni di 40 MIPS che lo rende capace di circa 30000 istruzioni per pixel al secondo sulle immagini catturate dallo ADNS-3060, che produce immagini di 30x30 pixel.

Il microprocessore possiede blocchi hardware quali SPI, I2C, UART e DMA che facilitano il trasferimento dei dati con le interfacce esterne, lasciando libero il processore di dedicarsi ai programmi utenti.

3.2.3. *Trasmettitore Radio*

Il trasmettitore radio è costituito dallo RFM12B [6], prodotto da HopeRF. E' un dispositivo di telecomunicazioni a basso costo, di tipo FSK che lavora con bande di 433, 868 e 915 MHz.

Rende possibile la comunicazione su 128 canali a 433 MHz, ed è quindi particolarmente adatto per reti di sensori numerose.

La potenza massima di trasmissione è 1 mW, che limita il range di comunicazione a 30 metri in interni.

La velocità massima di trasmissione raggiungibile dalla MouseCamera con l'RFM12B è di circa 115 Kb/s, con un conseguente frame-rate massimo via radio di 10 fps.

3.3. Principio di funzionamento

La comunicazione tra microprocessore e sensore avviene in modalità Master-Slave. Il programma utente richiede l'inizio dell'acquisizione dell'immagine, che configura la porta SPI per l'utilizzo della DMA in modo da far continuare l'esecuzione per poi, in un secondo momento, recuperare l'immagine.

Il sensore ADNS-3060 lavora in una modalità chiamata *frame capture*. Questa consente di scaricare velocemente la gamma completa di valori dei pixel di un singolo fotogramma. Per attivare questa modalità, viene scritto un qualunque dato su un apposito registro del sensore, chiamato appunto Frame_Capture. Il prossimo frame disponibile sarà memorizzato nella memoria interna del sensore, insieme ai 2/3 del fotogramma successivo (1536 valori). I dati vengono recuperati dalla lettura di un registro chiamato Pixel_Burst in modo cadenzato, guidato dal clock dell'intero sistema. L'intero processo impiega circa 1 ms.

L'alternativa per scaricare un frame dal sensore sarebbe quella di leggere tante volte dal registro Frame_Capture quante sono i pixel, cioè 900. Questa modalità è stata però scartata, in quanto non permetteva il parallelismo tra i due sistemi, e quindi impegnava per troppo tempo il calcolatore.

Al termine del download del frame, l'immagine si trova in una locazione di memoria interna al microprocessore, che può quindi elaborare.

4. MIDDLEWARE

4.1. Elaborazione dell'immagine

Di seguito verranno descritti i metodi utilizzati per rendere le immagini rilevate dal sensore processabili con maggiore precisione.

I sensori, per loro natura, introducono nelle immagini del rumore. Compito della pre-elaborazione è quella di eliminare questi rumori in modo che questi non vadano ad influire nella vera elaborazione delle immagini.

4.1.1. *Pedestal Correction*

Il sensore della MouseCamera introduce una distorsione che varia da sensore a sensore, la quale rende l'immagine più scura avvicinandosi ai bordi. Ciò è causato da una errata rilevazione del valore dei pixel proporzionalmente alla lontananza di questi dal centro dell'immagine.

Per ovviare a questo problema sono state effettuate delle misurazioni su più sensori messi in rilevazione di una immagine bianca, che dovrebbe dare valore massimo su tutti i pixel. Da queste misurazioni, è stata costruita una matrice di coefficienti di correzione.

3,06	3,06	2,88	2,58	2,45	2,33	2,23	2,13	2,04	1,96	1,75	1,81	1,75	1,81	1,81	1,75	1,75	1,75	1,81	1,88	1,81	2,04	2,13	2,23	2,33	2,45	2,72	2,88	3,06	3,5	
2,33	2,23	2,04	1,96	1,81	1,75	1,69	1,63	1,58	1,53	1,48	1,44	1,44	1,44	1,44	1,4	1,44	1,4	1,48	1,48	1,53	1,58	1,69	1,69	1,75	1,88	1,96	2,13	2,23	2,45	
2,13	2,04	1,96	1,81	1,69	1,63	1,58	1,53	1,48	1,44	1,4	1,36	1,32	1,32	1,32	1,32	1,32	1,32	1,36	1,4	1,44	1,48	1,53	1,58	1,63	1,75	1,81	1,96	2,04	2,33	
2,04	1,88	1,81	1,69	1,63	1,53	1,48	1,44	1,4	1,36	1,29	1,29	1,26	1,26	1,26	1,23	1,26	1,26	1,29	1,32	1,36	1,4	1,44	1,53	1,58	1,63	1,75	1,88	2,04	2,23	
1,96	1,81	1,69	1,58	1,53	1,44	1,4	1,36	1,29	1,26	1,23	1,23	1,2	1,17	1,17	1,17	1,17	1,2	1,23	1,26	1,26	1,29	1,36	1,44	1,48	1,58	1,69	1,75	1,88	2,04	
1,88	1,75	1,63	1,53	1,44	1,4	1,32	1,29	1,26	1,23	1,17	1,14	1,14	1,14	1,11	1,11	1,11	1,14	1,17	1,17	1,23	1,26	1,29	1,36	1,4	1,48	1,58	1,69	1,81	1,96	
1,75	1,69	1,58	1,48	1,4	1,32	1,29	1,26	1,23	1,17	1,14	1,11	1,09	1,09	1,09	1,09	1,09	1,09	1,11	1,14	1,17	1,2	1,26	1,29	1,44	1,4	1,53	1,58	1,75	1,88	
1,69	1,63	1,53	1,44	1,36	1,32	1,26	1,23	1,17	1,14	1,09	1,09	1,04	1,04	1,04	1,04	1,04	1,07	1,07	1,11	1,14	1,14	1,2	1,23	1,63	1,36	1,48	1,58	1,69	1,81	
1,69	1,63	1,48	1,44	1,36	1,26	1,2	1,17	1,11	1,11	1,07	1,04	1,02	1,02	1,02	1,02	1,02	1,02	1,07	1,09	1,09	1,11	1,17	1,23	1,29	1,36	1,44	1,53	1,63	1,75	
1,63	1,58	1,48	1,4	1,32	1,26	1,17	1,14	1,11	1,09	1,04	1,02	1,02	1	1	1,02	1	1,02	1,02	1,07	1,09	1,11	1,14	1,2	1,23	1,32	1,4	1,48	1,58	1,75	
1,63	1,53	1,44	1,36	1,29	1,23	1,17	1,14	1,09	1,07	1,04	1,02	1,02	1,02	1,02	1,02	1,02	1,02	1,07	1,07	1,09	1,14	1,2	1,23	1,29	1,36	1,48	1,53	1,69		
1,63	1,53	1,44	1,36	1,29	1,23	1,14	1,11	1,09	1,07	1,04	1,04	1,02	1,04	1,02	1,02	1,04	1,02	1,02	1,04	1,09	1,09	1,14	1,17	1,23	1,29	1,36	1,44	1,53	1,63	
1,63	1,53	1,44	1,36	1,26	1,2	1,14	1,09	1,07	1,07	1,04	1,04	1,02	1,04	1,07	1,04	1,04	1,02	1,07	1,09	1,11	1,17	1,2	1,26	1,29	1,36	1,44	1,53	1,63		
1,63	1,53	1,4	1,32	1,26	1,2	1,17	1,11	1,09	1,07	1,07	1,04	1,04	1,04	1,04	1,04	1,04	1,04	1,04	1,09	1,11	1,14	1,2	1,23	1,29	1,36	1,44	1,53	1,63		
1,63	1,53	1,44	1,32	1,29	1,23	1,17	1,11	1,09	1,09	1,09	1,07	1,04	1,04	1,04	1,04	1,04	1,04	1,07	1,11	1,14	1,2	1,26	1,32	1,36	1,44	1,53	1,63			
1,63	1,53	1,44	1,36	1,29	1,23	1,2	1,14	1,11	1,09	1,07	1,07	1,04	1,04	1,04	1,04	1,02	1,04	1,04	1,07	1,11	1,17	1,2	1,26	1,32	1,4	1,44	1,53	1,69		
1,63	1,58	1,48	1,36	1,32	1,26	1,2	1,17	1,11	1,09	1,07	1,07	1,07	1,04	1,04	1,04	1,04	1,04	1,04	1,07	1,11	1,17	1,23	1,29	1,36	1,4	1,48	1,58	1,75		
1,69	1,58	1,48	1,44	1,32	1,29	1,23	1,2	1,14	1,14	1,09	1,07	1,04	1,04	1,04	1,04	1,02	1,02	1,04	1,07	1,09	1,14	1,2	1,26	1,29	1,36	1,44	1,53	1,63	1,75	
1,75	1,63	1,53	1,44	1,36	1,32	1,26	1,23	1,17	1,11	1,09	1,09	1,07	1,07	1,04	1,02	1,04	1,04	1,04	1,09	1,11	1,17	1,2	1,29	1,32	1,44	1,48	1,58	1,63	1,81	
1,81	1,69	1,58	1,48	1,44	1,32	1,29	1,26	1,2	1,17	1,11	1,09	1,07	1,04	1,04	1,04	1,04	1,07	1,09	1,11	1,14	1,17	1,23	1,29	1,4	1,44	1,53	1,63	1,75	1,88	
1,81	1,75	1,63	1,53	1,44	1,4	1,32	1,29	1,26	1,23	1,14	1,11	1,09	1,09	1,07	1,07	1,09	1,11	1,11	1,17	1,2	1,23	1,29	1,36	1,4	1,48	1,58	1,69	1,81	1,96	
1,88	1,81	1,69	1,58	1,53	1,44	1,32	1,32	1,29	1,26	1,2	1,17	1,11	1,11	1,11	1,11	1,14	1,17	1,17	1,2	1,23	1,26	1,32	1,4	1,48	1,53	1,63	1,75	1,88	1,96	
1,96	1,88	1,75	1,63	1,53	1,48	1,4	1,36	1,32	1,29	1,26	1,23	1,17	1,17	1,17	1,17	1,2	1,2	1,23	1,26	1,26	1,32	1,4	1,48	1,53	1,63	1,69	1,81	1,96	2,13	
2,04	1,96	1,81	1,69	1,63	1,53	1,48	1,4	1,36	1,36	1,32	1,29	1,26	1,23	1,26	1,23	1,26	1,29	1,32	1,36	1,4	1,48	1,53	1,58	1,75	1,81	1,96	2,04	2,23		
2,23	2,04	1,88	1,75	1,69	1,58	1,53	1,48	1,44	1,44	1,4	1,36	1,32	1,32	1,29	1,29	1,32	1,36	1,36	1,4	1,44	1,53	1,58	1,63	1,75	1,81	1,88	2,04	2,23	2,33	
2,33	2,13	1,96	1,88	1,81	1,69	1,58	1,58	1,53	1,48	1,48	1,44	1,44	1,4	1,36	1,4	1,4	1,4	1,44	1,48	1,53	1,58	1,69	1,81	1,81	1,96	2,13	2,23	2,33	2,58	
2,45	2,33	2,13	2,04	1,88	1,81	1,75	1,63	1,58	1,58	1,53	1,53	1,53	1,48	1,48	1,48	1,53	1,53	1,63	1,63	1,69	1,81	1,88	2,04	2,13	2,23	2,45	2,58	2,72		
2,58	2,45	2,33	2,13	1,96	1,88	1,81	1,75	1,69	1,69	1,63	1,63	1,63	1,58	1,58	1,63	1,63	1,69	1,75	1,81	1,88	1,96	2,04	2,13	2,23	2,45	2,58	2,72	3,06		
2,88	2,58	2,45	2,33	2,13	2,04	1,96	1,88	1,81	1,81	1,75	1,75	1,75	1,75	1,75	1,75	1,75	1,75	1,81	1,81	1,88	1,96	2,04	2,13	2,23	2,33	2,45	2,58	2,88	3,06	3,27
3,06	2,88	2,58	2,45	2,33	2,23	2,13	2,04	2,04	1,96	1,96	1,96	1,88	1,96	1,96	1,88	1,96	1,96	2,04	2,04	2,13	2,23	2,33	2,45	2,58	2,72	2,88	3,06	3,27	3,77	

FIGURA 4-1MATRICE DI CORREZIONE DELL'IMMAGINE

Moltiplicando i coefficienti della matrice con i valori rilevati dal sensore dando come risultato una immagine con una luminosità più uniforme.

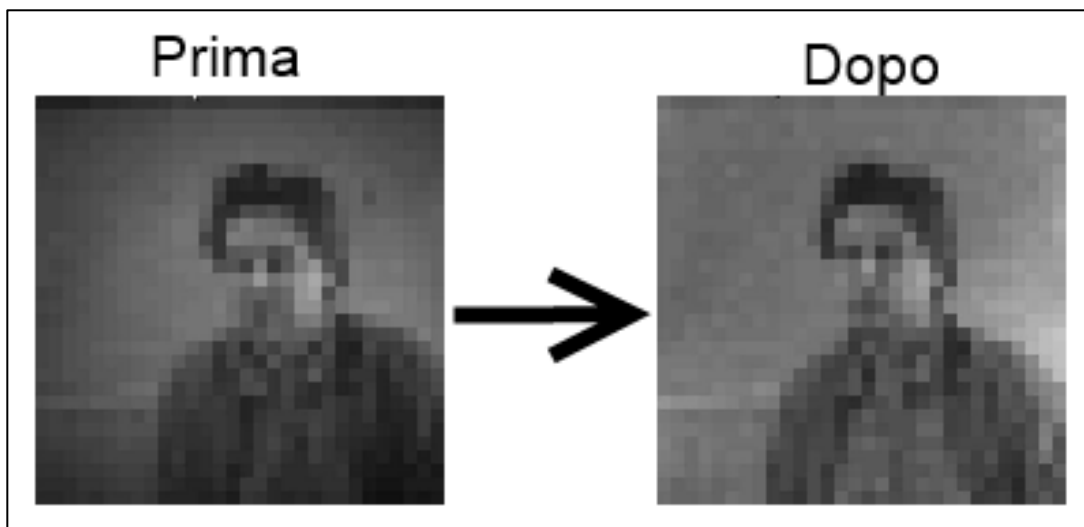


FIGURA 4-2RISULTATO DELLA PEDESTAL CORRECTION

4.1.2. Gamma Correction

La correzione gamma è una operazione non lineare, usata per codificare i valori discreti rilevati dal sensore.

E' richiesta per compensare alcuni limiti dell'occhio umano, massimizzando l'utilizzo di informazione relativamente alla banda di percezione del colore e dell'intensità di luce.

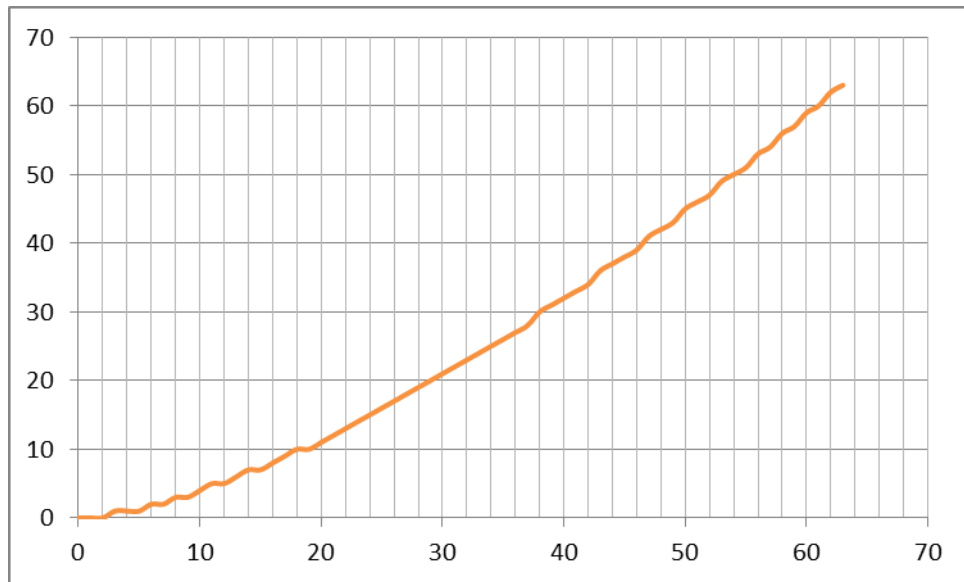


FIGURA 4-3 GRAFICO DELLA CORREZIONE GAMMA. IN ASCISSA IL VALORE DEI PIXEL IN INGRESSO, IN ORDINATA IL VALORE DEI PIXEL IN USCITA DALLA CORREZIONE

Nella MouseCamera viene applicata la seguente trasformazione non lineare:

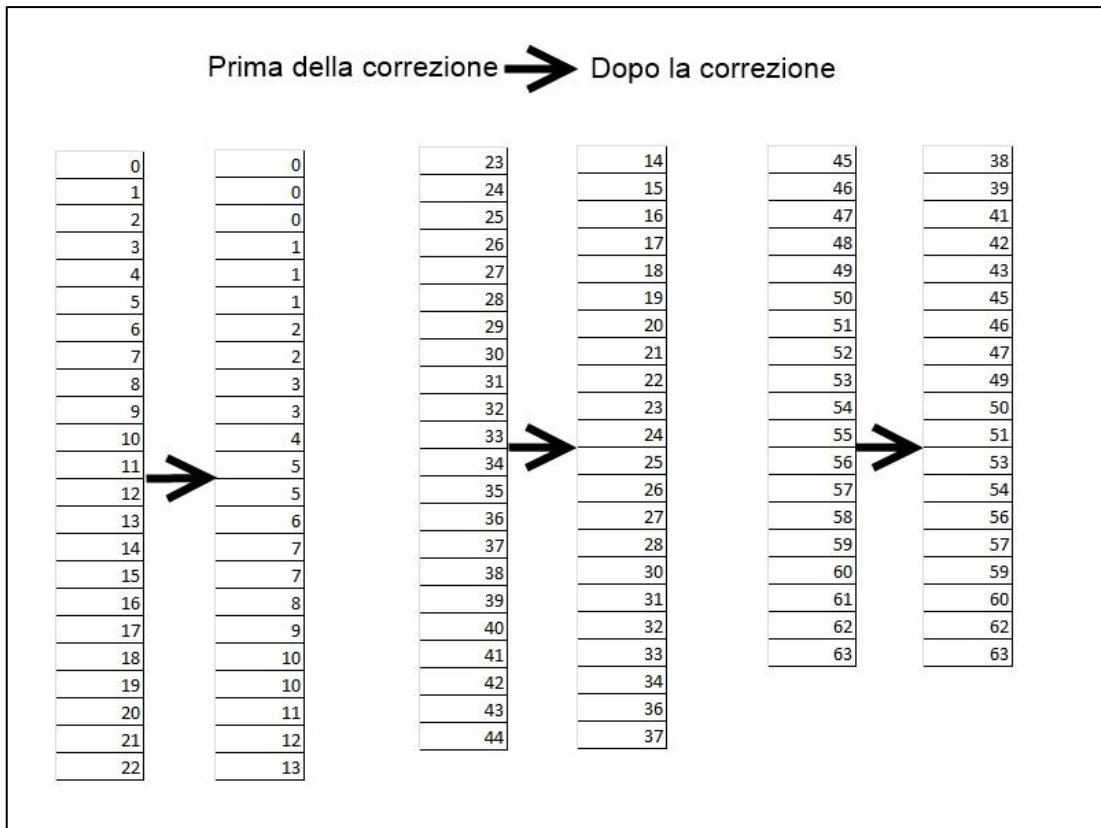


FIGURA 4-4 TRASFORMAZIONE NON LINEARE PER LA CORREZIONE GAMMA

4.2. Protocollo di rete

Di seguito verrà descritta la modalità di interazione pensata per una rete di MouseCamera. Questo protocollo prevede la definizione del linguaggio, costituito dai messaggi scambiati, che devono potersi interpretare correttamente. L'utilizzo di questo protocollo su più sensori permette la comunicazione tra di essi, garantendo allo stesso tempo l'esecuzione indipendente delle proprio mansioni.

Uno dei requisiti fondamentali di una rete è la scalabilità, cioè la capacità di rilevare l’inserimento o la rimozione di nuovi dispositivi, senza la riconfigurazione “offline” di tutta la rete.

La topologia utilizzata è quella di una rete a stella, nella quale uno dei sensori agisce da hub.

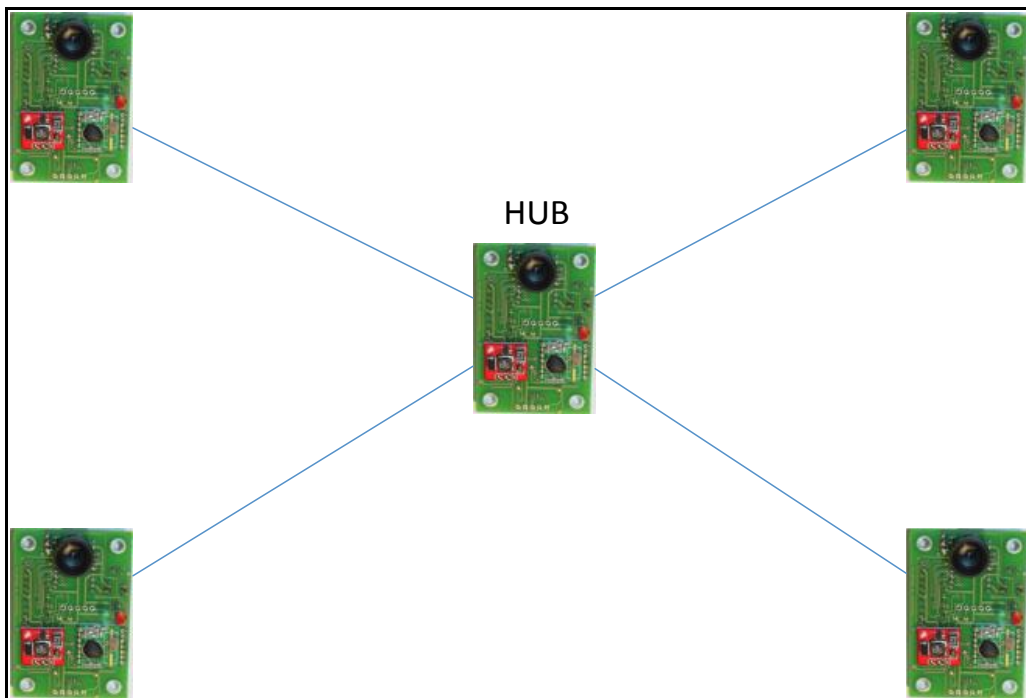


FIGURA 4-5 TOPOLOGIA DELLA RETE DI SENSORI

Lo hub ha i seguenti compiti:

- inizializza il sistema;
- scansiona, a intervalli di tempo regolari, l’area alla ricerca di sensori da registrare alla rete;

- si interfaccia con un dispositivo esterno (es. un PC), comunicando i dati rilevati dalla rete all'esterno
- invia comandi ai sensori

Ogni sensore, all'interno della stessa rete, è configurato con un indirizzo univoco a 8 bit. Sono quindi utilizzabili in una rete fino a 255 MouseCamera. L'indirizzo '00000000' è riservato all'hub.

All'inizializzazione del sistema, lo hub invia su uno dei 128 canali, configurabili dal trasmettitore della MouseCamera, un messaggio (DISCOVERY) a tutti gli indirizzi da 1 a 255. Il canale su cui viene mandato questo messaggio, è chiamato "Discovery Channel". Ogni MouseCamera, alla sua accensione, attende un messaggio dall'host su questo canale.

Alla ricezione del messaggio di discovery, gli host rispondono all'hub con DISCOVERY_RES, grazie al quale il relativo sensore viene aggiunto alla rete.

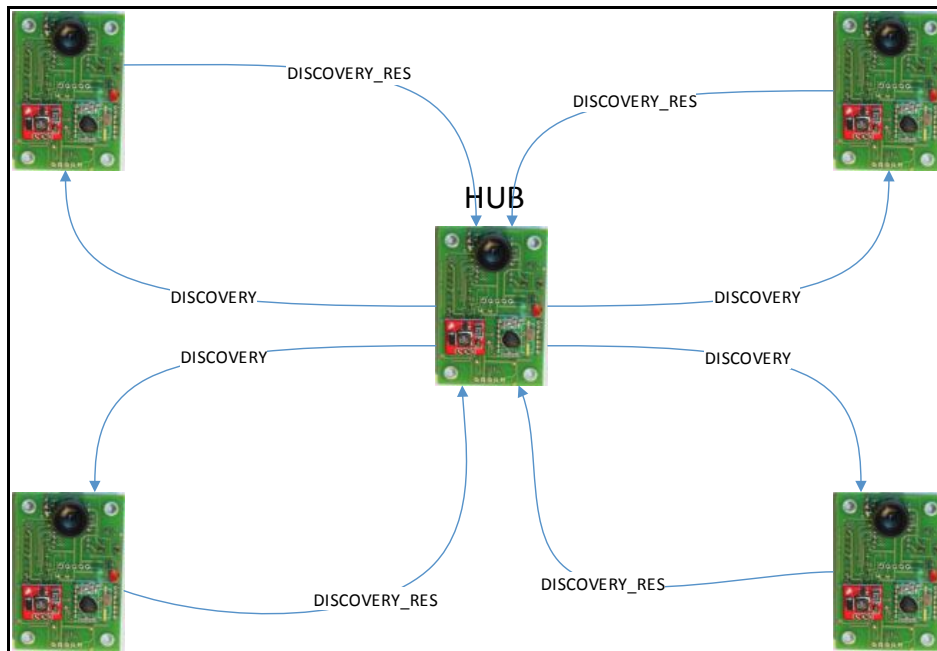


FIGURA 4-6 FLUSSO DI COMUNICAZIONI DURANTE IL MECCANISMO DI DISCOVERY

Finita la fase di discovery, l’hub invia il comando SWITCHCHANNEL a tutti gli host della rete, facendo cambiare a tutti il canale di comunicazione. Questo è chiamato “Operative Channel”, sul quale verranno effettuate tutte le comunicazioni (eccetto quelle di discovery).

Ad intervalli regolari (5 min), l’hub si mette in comunicazione sul Discovery Channel, ed invia un messaggio di discovery (DISCOVERYALL) per aggiungere nuovi sensori alla rete. A differenza del messaggio inviato nella fase di inializzazione, DISCOVERYALL viene inviato in broadcast, mentre DISCOVERY in unicast. Questa scelta è stata fatta valutando il costo, in termini di tempo, della scansione su tutti gli indirizzi effettuata dalla DISCOVERY, dovendo l’hub tornare operativo in poco tempo e considerando

il numero di sensori da aggiungere alla rete relativamente piccolo (dell'ordine dell'unità).

I sensori che ricevono il messaggio DISCOVERYALL, attendono un tempo casuale che va da 0 a 9 intervalli temporali, equivalenti al timeout dell'hub, in attesa di risposta. La casualità del tempo di attesa viene giustificata per evitare collisioni di segnali tra i dispositivi. Nel caso che più MouseCamera scelgano lo stesso intervallo temporale, esse non verranno aggiunte dall'hub alla rete, ma avranno una nuova possibilità nella successiva fase di discovery.

I messaggi sono così strutturati:

- indirizzo di destinazione (0xFF per broadcast): 1 Byte;
- indirizzo del mittente (0xFF per broadcast): 1 Byte;
- identificativo del tipo di messaggio: 1 Byte;
- payload: 0-123 Byte;
- controllo di errore CRC: 1 Byte;

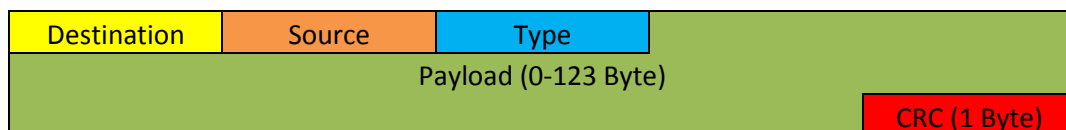


FIGURA 4-7 DEFINIZIONE DEI MESSAGGI DI COMUNICAZIONE

Il payload ha una lunghezza massima di 123 byte. Questo è dovuto alla limitazione imposta dal trasmettitore RFM della MouseCamera (127 Byte). È stato quindi implementato un meccanismo di divisione in pacchetti per

messaggi la cui lunghezza superasse il limite imposto, come ad esempio il messaggio SENDIMAGE (un'immagine compressa è rilevata dalla MouseCamera è grande 675 byte).

Nello sviluppo di un protocollo di rete per la comunicazione fra più Mouse Camera sono stati definiti i seguenti tipi di messaggi:

- **SYNCREQ:** è un messaggio che solo il sensore host può inviare. Può essere sia Multicast che Unicast. Viene utilizzato quando è necessaria la sincronizzazione tra i dispositivi;
- **ACK:** è un messaggio vuoto, che funziona da conferma (acknowledgement) a messaggi precedentemente ricevuti;
- **DISCOVERY:** utilizzato dall'host per comunicare a uno specifico dispositivo client la sua raggiungibilità per inserirlo nella rete;
- **DISCOVERYALL:** utilizzato dall'host per comunicare a tutti i client raggiungibili la sua accensione e la conseguente inizializzazione delle rete;
- **DISCOVERY_RES:** risposta del client verso l'host ad una richiesta di discovery;
- **STARTMOTIONCAPTURE:** messaggio broadcast per comunicare ai client di iniziare a rilevare i movimenti tramite l'apposita funzionalità del sensore mouse;
- **STOPMOTIONCAPTURE:** messaggio broadcast per comunicare ai client di smettere di rilevare i movimenti;

- **GETMOTIONCAPTURE**: messaggio inviato dall'host per ricevere lo scostamento dell'immagine rilevato dal client tramite l'apposita funzionalità del mouse;
- **SENDMOTIONCAPTURE**: messaggio inviato dal client all'host comprendente lo scostamento dell'immagine rilevato;
- **SWITCHCHANNEL**: messaggio inviato dall'host in broadcast, per comunicare il canale sul quale i client dovranno comunicare;
- **GETIMAGE**: messaggio inviato dall'host verso un client per richiedere l'invio dell'immagine rilevata;
- **SENDIMAGE**: messaggio inviato dal client verso l'host, contenente l'immagine rilevata;
- **TIMESYNC**: messaggio inviato dall'host per sincronizzare il timer dei client;
- **STARTMOTIONESTIMATION**: messaggio inviato dall'host verso il client per inizializzare l'algoritmo di stima del movimento;
- **SENDMOTIONESTIMATION**: messaggio inviato dal client per comunicare all'host il risultato della stima del movimento;

4.3. Algoritmo di motion estimation

L' algoritmo utilizzato per la stima del movimento è il 3D-RS (Three Dimensions Recursive Search) [3], ideato dal gruppo di ricerca di teoria dell'informazione dell'università di Delft, Olanda. Questo, oltre a rilevare la presenza di movimento in una serie di immagini, fornisce anche una stima della velocità (in pixel/frame) e della direzione.

L'idea di base è quella di suddividere l'immagine in una matrice di blocchi $N \times N$. Per ogni immagine elaborata dall'algoritmo, vengono stimati 2 vettori di movimento, rappresentanti il movimento in orizzontale e in verticale. Per ogni elaborazione, vengono passati all'algoritmo l'immagine ed i vettori precedentemente calcolati (ricorsivamente).

Di seguito i principali passi dell'algoritmo implementato:

- Viene creato uno spazio di ricerca, espandendo l'immagine oltre i propri margini delle dimensioni di un blocco, mantenendo l'immagine al centro. Questo viene fatto per permettere la ricerca anche sui bordi.

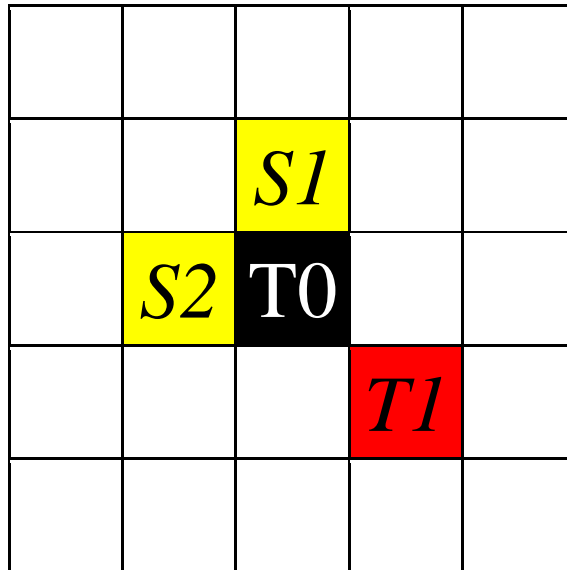


FIGURA 4-8 DETTAGLIO DEI CAMPIONI SU CUI VIENE APPLICATO L'ALGORITMO 3DRS DI
MOTION ESTIMATION

- Per ogni blocco dell'immagine vengono calcolati 4 candidati per il vettore di movimento:
- T0: è una copia del precedente vettore di movimento (candidato temporale);
- S1: è una copia del vettore del blocco superiore (spaziale);
- S2: è una copia del vettore del blocco di fianco (spaziale);
- T1: al vettore presente nel blocco diagonale viene aggiunta una componente casuale (temporale);
- Per ogni candidato viene calcolata la differenza (in termini di somma del valore dei pixel) tra il blocco candidato, cioè indicato dallo scostamento del vettore, e quello in elaborazione.

- Viene scelto il miglior candidato, cioè quello con la differenza minima tra i blocchi.

Nell'implementazione della Mouse Camera viene applicato l'algoritmo 3DRS con blocchi di grandezza 3×3 pixel, si hanno quindi matrici 10×10 di vettori di stima del movimento come risultato dell'elaborazione. La componente casuale, che permette al sistema la ricerca di nuovi movimenti, è forzata solo nella direzione orizzontale, vista la natura del problema di rilevazione del traffico, e quindi l'assenza di movimenti verticali. Questo è stato fatto per non avere ulteriori fonti di rumore che potrebbero disturbare la stima del movimento dei veicoli.

5. IMPLEMENTAZIONE DEL SOFTWARE PER LA RILEVAZIONE DEL TRAFFICO

5.1. Calibrazione dell'immagine

Le immagini rilevate con la MouseCamera sono rappresentate da una matrice 30x30 di pixel, rappresentati con 6 bit e quindi con valori che vanno da 0 a 63. Dovendo lavorare all'aria aperta, il sistema viene sottoposto a variazioni di illuminazione continue, dovute ai cambiamenti meteorologici.

I cambi di illuminazione portano il sensore a lavorare con immagini su cui non riesce a distinguere i pixel, a causa di scarsa o troppa illuminazione.

Risulta quindi necessario un meccanismo per riportare la MouseCamera a lavorare nelle proprie condizioni operative.

Il sensore ADNS-3060, possiede un sistema automatico di rilevazione e configurazione dell'otturatore, che permette alla luce di passare solo per un determinato periodo di tempo per frame.

Il regolatore automatico cambia il livello di otturazione ad ogni rilevazione di immagine. Il risultato è che le immagini rilevate non hanno una illuminazione stabile con il risultato di una alta probabilità di avere false rilevazioni di movimento con l'algoritmo 3DRS.

E' stato scelto di disabilitare questa funzionalità, in favore di una migliore efficacia nella stima del movimento, e di lasciare la rilevazione di cambi di illuminazione al software.

Il software controlla il valore medio dei pixel e il numero di pixel a massimo (o minimo) valore periodicamente una volta al secondo.

Per rendere il sistema più stabile, prima di effettuare un cambio di otturazione, il sensore deve uscire dal range operativo per quattro periodi consecutivi.

Attraverso sperimentazioni, si è visto che il range operativo della MouseCamera, in termini di valore medio dei pixel rilevati è da 16 a 44 (su valori che vanno da 0 a 63).

Sempre sperimentalmente, è stato valutato che, il sensore dava una alta percentuale di false rilevazione nel caso che più del 10% dei pixel rilevati avessero il massimo o il minimo valore (63 o 0).

Dopo aver rilevato un cambio di illuminazione, è necessario riconfigurare l'otturatore del sensore ADNS-3060. Purtroppo nell'implementazione della MouseCamera, per ottenere alte prestazioni di frame rate, il sensore ADNS-3060 viene utilizzato in *burst mode* e questo crea problemi.

Quando il sensore è in *burst mode*, per modificare qualsiasi configurazione di questo, deve essere effettuato un reset. A causa del reset, il sensore smette di rilevare immagini per 80ms, e si ha quindi un fermo dell'applicazione.

Al riavvio, viene lasciato al regolatore automatico il compito di regolare l'otturazione nella migliore configurazione possibile per la luminosità

rilevata, e viene riattivato il *burst mode* con la conseguente ripresa dell'applicazione.

5.2. Implementazione dell'algoritmo di ME: i 3 motori

Nello studio per l'implementazione dell'algoritmo di Motion Estimation 3DRS, è stato misurato sperimentalmente il tempo tra una elaborazione e l'altra. L'algoritmo impiega di motion estimation, comprensivo del tempo per recuperare l'immagine dal sensore e convertirla in un array, impiega circa 15,5 ms.

Il frame rate massimo raggiungibile dalla MouseCamera con l'algoritmo di Motion Estimation è quindi circa 64,5 fps.

Sono state effettuate varie sperimentazioni a diverse velocità dell'algoritmo. In particolare sono stati studiati, nell'applicazione del caso di sistema di controllo del traffico, i seguenti frame rate:

- 64 fps
- 32 fps
- 16 fps
- 8 fps
- 4 fps

Si è visto che gli algoritmi a 64 e 32 fps sono molto sensibili a movimenti veloci, in quanto le immagini elaborate catturano con maggiore definizione spostamenti rapidi. Al contrario con questi due frame rate, i movimenti lenti vengono rilevati con difficoltà.

Discorso opposto invece per gli algoritmi a 16, 8 e 4 fps, che sono forti con movimenti lenti, e deboli con movimenti veloci.

Sperimentalmente si è valutato che, nel caso del sistema di controllo del traffico, i frame rate si adattavano alle seguenti categorie:






- 64 e 32 fps: macchine e mezzi pesanti ad alta velocità;
- 16 fps: macchine e mezzi pesanti a bassa velocità, cicli ad alta velocità;
- 8 fps: cicli a bassa velocità, pedoni.
- 4 fps: pedoni

Avendo il sistema come requisito quello di classificare tutte le classi di veicoli e pedoni, è stato necessario studiare un sistema applicare tutti gli algoritmi ed unirne i risultati. I frame rate a 64 fps e 4 fps possono essere non utilizzati, in quanto ottengono gli stessi risultati rispettivamente dei motore di motion estimation a 32 fps e 8 fps.

Il sistema studiato è quello di dividere il tempo in slot. Ogni slot impiega 15,625 ms (64 slot al secondo).



FIGURA 5-1 SUDDIVISIONE DEL TEMPO DI ELABORAZIONE NELLA MOUSECAMERA

-  Motore ME a 32 fps
-  Motore ME a 16 fps
-  Motore ME a 8 fps
-  Unione, elaborazione e invio risultati
-  Calibrazione dell'illuminazione

I risultati degli algoritmi di Motion Estimation vengono elaborati ed inviati 4 volte al secondo. Ogni risultato quindi comprende:

- 8 stime del movimento a 32 fps
- 4 stime del movimento a 16 fps
- 2 stime del movimento a 8 fps

Ogni stima produce una matrice dei vettori di stima del movimento (matrice 10x10). Ogni vettore contiene valori che possono andare da 0 a 27, con il segno + o – che indica la direzione dello spostamento.

In mezzo ad ogni elaborazione dei risultati, che avviene quattro volte al secondo, per ogni motore di Motion Estimation, vengono sommati tutti i valori delle matrici dei vettori di stima.

I risultati dei tre algoritmi vengono comparati, inserendo nella matrice risultato il valore più grande rilevato.

5.3. Riconoscimento degli oggetti

La matrice di vettori di stima del movimento, prima di essere valutata per dare dei risultati, viene processata attraverso alcuni algoritmi per l'eventuale presenza di rumore e per facilitare la classificazione dei veicoli.

Gli algoritmi utilizzati sono:

- Erosion
- Labeling
- Snake

5.3.1. *Erosion*

Con questo algoritmo, vengono eliminati eventuali vettori di stima del movimento rilevati fuori dalla forma degli oggetti, che si muovono nelle immagini catturate.

E' utile nel caso di presenza di ombre o riflessi, che altrimenti andrebbero a modificare la forma rilevata dell'oggetto in movimento, e quindi ne renderebbero più difficile la classificazione.

Per ogni vettore della matrice di stima del movimento 10x10, viene conteggiato il numero di vettori diversi da zero nel suo intorno. Se il numero dei vettori vicini diverso da zero è minore di tre, il vettore viene cancellato. L'algoritmo viene applicato scorrendo ogni elemento per colonna, e successivamente per riga.

		0							
						1			
							4	3	
					4	5	7	5	
				5	7	8	8	5	
			2	3	5	5	5	4	
									1

FIGURA 5-2EROSION: RISULTATO DEL CONTEGGIO DELLE CELLE VICINE

Nella figura viene data evidenza, per ogni casella, la quantità di vicini non nulli. Le caselle in rosso hanno un valore minore di tre e quindi vengono eliminate, dando come risultato dell'algoritmo la figura.

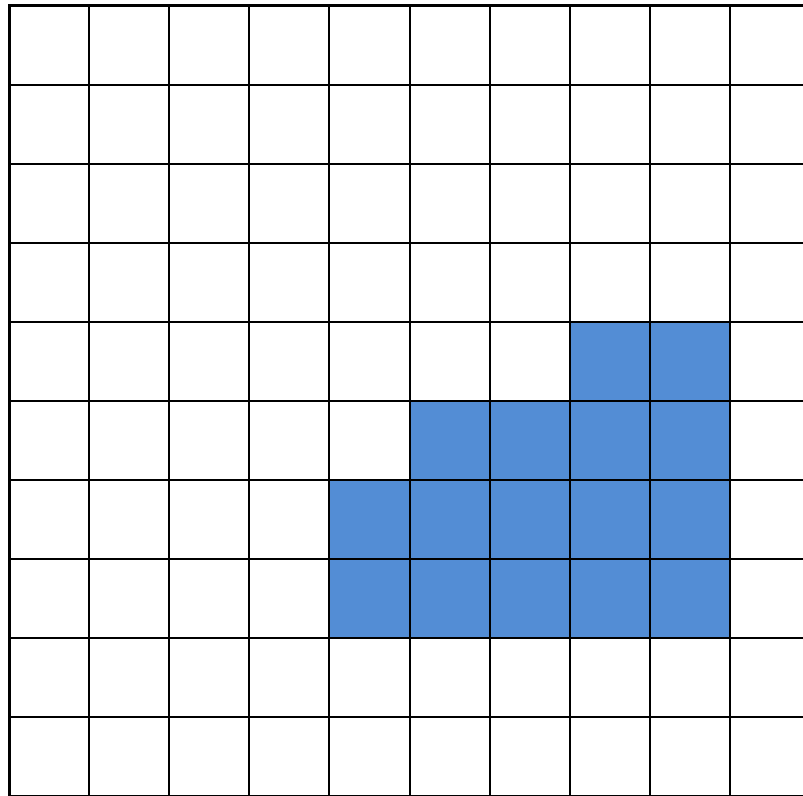


FIGURA 5-3EROSION: RISULTATO FINALE DELL'ALGORITMO

5.3.2. *Labeling*

Il labeling è algoritmico di teoria dei grafi, dove sottoinsiemi di componenti collegati sono etichettati univocamente. Grazie a questo algoritmo si riesce a distinguere oggetti diversi all'interno della matrice di stima del movimento.

I passi dell'algoritmo sono:

- si scorre ogni elemento dei dati per colonna, poi per riga;
- se l'elemento non è lo sfondo (valore non nullo):

- vengono presi tutti gli elementi confinanti con l'elemento corrente;
- se non ci sono vicini, viene etichettato in modo univoco l'elemento corrente e si continua con l'algoritmo;
- in caso contrario, si trova l'elemento successivo con l'etichetta più piccola e vi si assegna l'etichetta corrente;
- si salvano le etichette equivalenti
- Si effettua un secondo passaggio scorrendo ogni elemento dei dati per colonna, poi per riga:
- Se l'elemento non è lo sfondo
- all'elemento viene assegnato l'etichetta equivalente più basso

Si dia la seguente immagine rilevata:

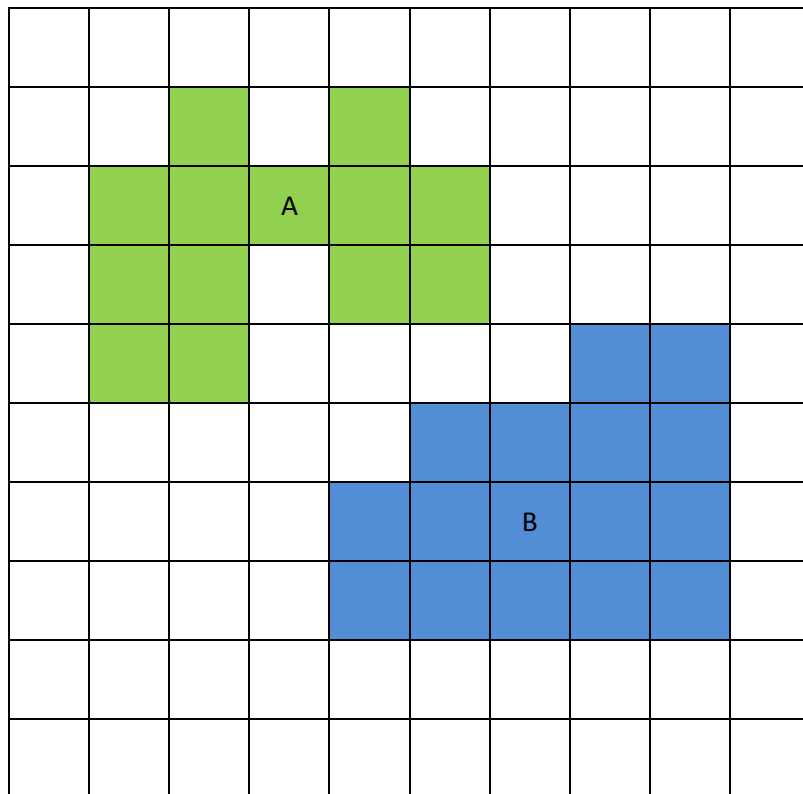


FIGURA 5-4 LABELING: SITUAZIONE INIZIALE

Al primo passaggio la situazione sarebbe la seguente:

0	0	0	0	0	0	0	0	0	0
0	0	1	0	2	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0
0	1	1	0	1	1	0	0	0	0
0	1	1	0	0	0	0	3	3	0
0	0	0	0	0	4	4	4	4	0
0	0	0	0	4	4	4	4	4	0
0	0	0	0	4	4	4	4	4	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

FIGURA 5-5 LABELING: SITUAZIONE AL PRIMO PASSAGGIO

La tabella di equivalenza delle etichette è:

Oggetto A	1,2
Oggetto B	3,4

TABELLA 1 LABELING: EQUIVALENZE

Perciò al secondo passaggio si avrà:

0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0
0	1	1	0	1	1	0	0	0	0
0	1	1	0	0	0	0	3	3	0
0	0	0	0	0	3	3	3	3	0
0	0	0	0	3	3	3	3	3	0
0	0	0	0	3	3	3	3	3	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

FIGURA 5-6 LABELING: RISULTATO FINALE DELL'ALGORITMO

5.3.3. Snake

Per delineare i contorni di ogni oggetto etichettato tramite il labeling, viene utilizzata la tecnica Active Contour Model, detta anche Snake.

0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0
0	1	1	0	1	1	0	0	0	0
0	1	1	0	0	0	0	3	3	0
0	0	0	0	0	3	3	3	3	0
0	0	0	0	3	3	3	3	3	0
0	0	0	0	3	3	3	3	3	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0
0	1	1	0	1	1	0	0	0	0
0	1	1	0	0	0	0	3	3	0
0	0	0	0	0	3	3	3	3	0
0	0	0	0	3	3	3	3	3	0
0	0	0	0	3	3	3	3	3	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

FIGURA 5-22 SNAKE: RICERCA DEI CONTORNI IN VERTICALE

Questa prevede, per ognuna delle due direzioni e per ogni verso, la rilevazione dei primi elementi con l'etichetta ricercata.

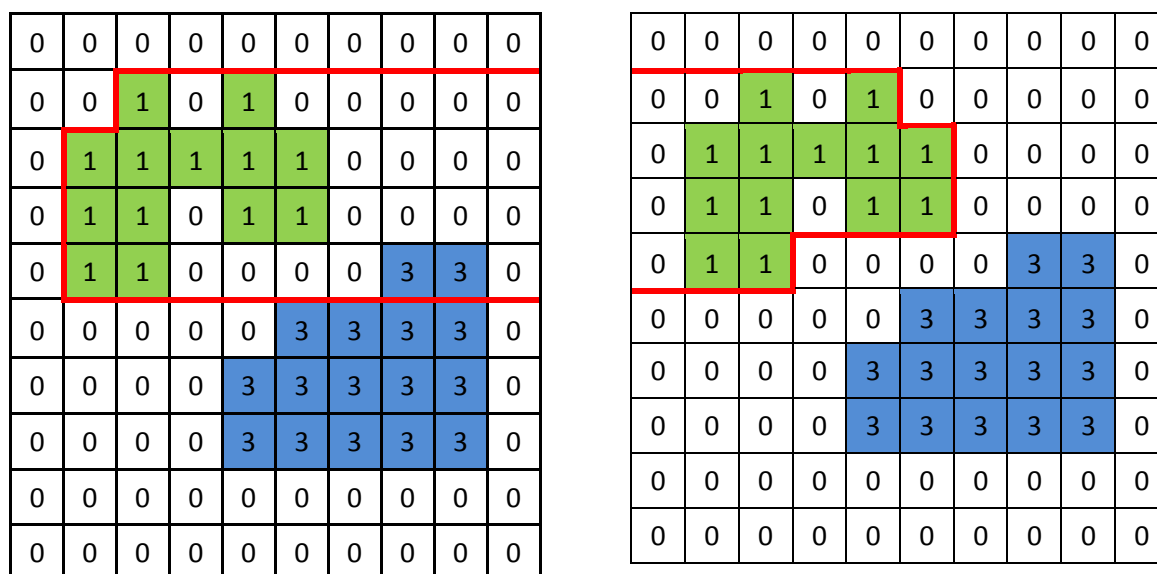


FIGURA 5-23 SNAKE: RICERCA DEI CONTORNI IN ORIZZONTALE

Al termine dell'algoritmo si ha, per ogni oggetto etichettato, il contorno di questo. Con questa conoscenza è possibile calcolare:

- il movimento stimato medio dell'oggetto;
- il verso di movimento dell'oggetto;
- la larghezza;
- l'altezza;
- la posizione all'interno dell'immagine.

0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0
0	1	1	0	1	1	0	0	0	0
0	1	1	0	0	0	0	3	3	0
0	0	0	0	0	3	3	3	3	0
0	0	0	0	3	3	3	3	3	0
0	0	0	0	3	3	3	3	3	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

FIGURA 5-7 SNAKE: RISULTATO FINALE

5.4. Classificazione dei veicoli

Alla fine della elaborazione della matrice di stima del movimento, per ogni oggetto rilevato all'interno delle immagini vengono valutati i seguenti parametri:

- Area;
- Velocità
- Rapporto altezza/larghezza (ratio)

L'area è il numero di caselle occupate dall'immagine, mentre la velocità è la media dei vettori di stima del movimento all'interno dell'oggetto.

Dovendo classificare in base a questi valori, è stata fatta una serie di sperimentazioni, valutando per ogni classe di veicolo gli intervalli di validità.

5.4.1. Pedoni

Al passaggio di pedoni ci si aspetta di rilevare una figura alta e stretta, piccola e non molto veloce.

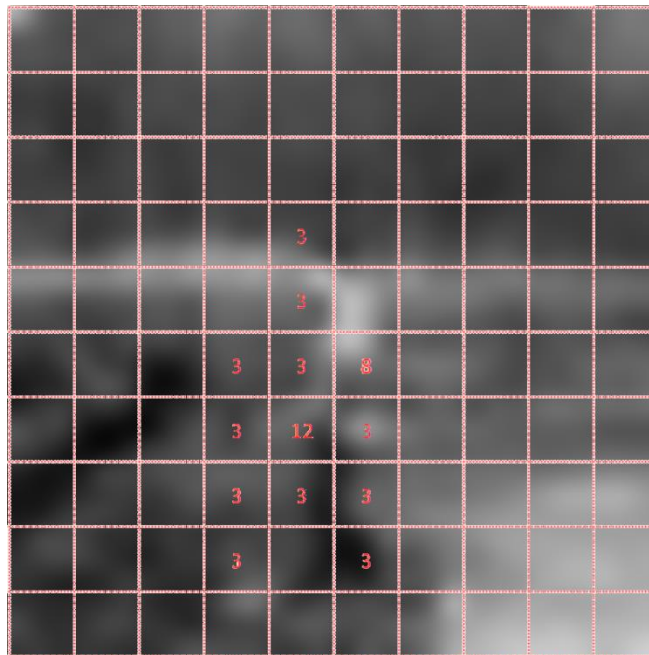


FIGURA 5-8 RILEVAZIONE DI UN PEDONE

E' stato visto che per i pedoni, i parametri di classificazione potevano essere riassunti in:

- Area > 4
- Ratio > 1,2
- Velocità <= 8

5.4.2. Cicli

Al passaggio di cicli ci si aspetta di rilevare una figura con un rapporto altezza/larghezza vicino all'unità, una velocità maggiore di quella dei pedoni oltre a una area di occupazione dell'immagine media.

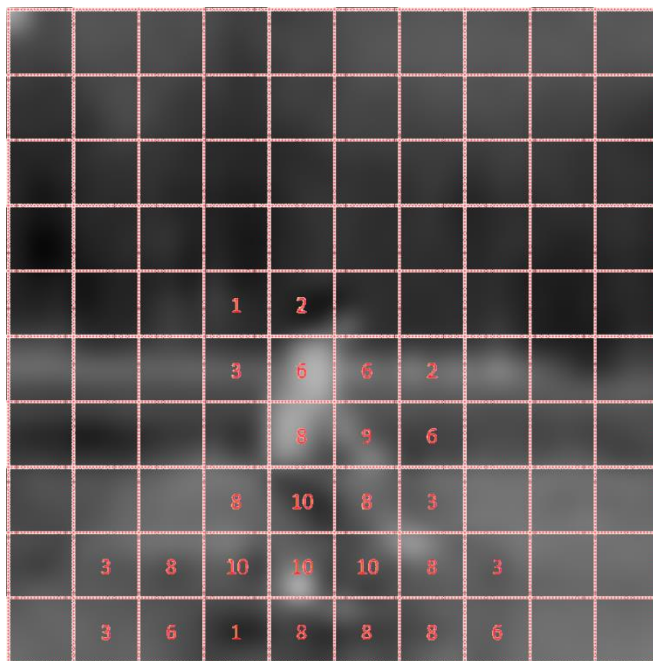


FIGURA 5-9 RILEVAZIONE DI UNA BICICLETTA

E' stato visto che per i cicli, i parametri di classificazione potevano essere riassunti in:

- Area ≥ 10
- $0,5 \leq \text{Ratio} \leq 1,2$
- $4 \leq \text{Velocità} \leq 12$

5.4.3. Automobili

Al passaggio di un automobili ci si aspetta di rilevare una figura con un rapporto altezza/larghezza basso, una alta velocità ed una area molto grande.

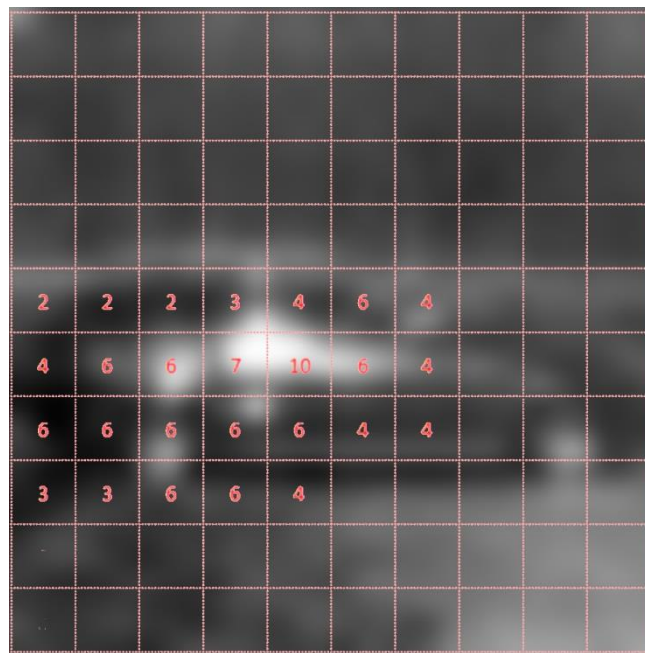


FIGURA 5-10 RILEVAZIONE DI UNA MACCHINA

E' stato visto che per le automobili, i parametri di classificazione potevano essere riassunti in:

- Area ≥ 20
- Ratio ≤ 1

- Velocità ≥ 6

Come si evince dai parametri di classificazione, la distinzione dei pedoni dagli altri veicoli è la più semplice da effettuare, avendo questi una forma e una velocità nettamente diversa dagli altri veicoli.

Dal grafico si intuisce invece come ci sia una certa difficoltà nel distinguere i cicli dalle automobili con gli attuali parametri di classificazione.

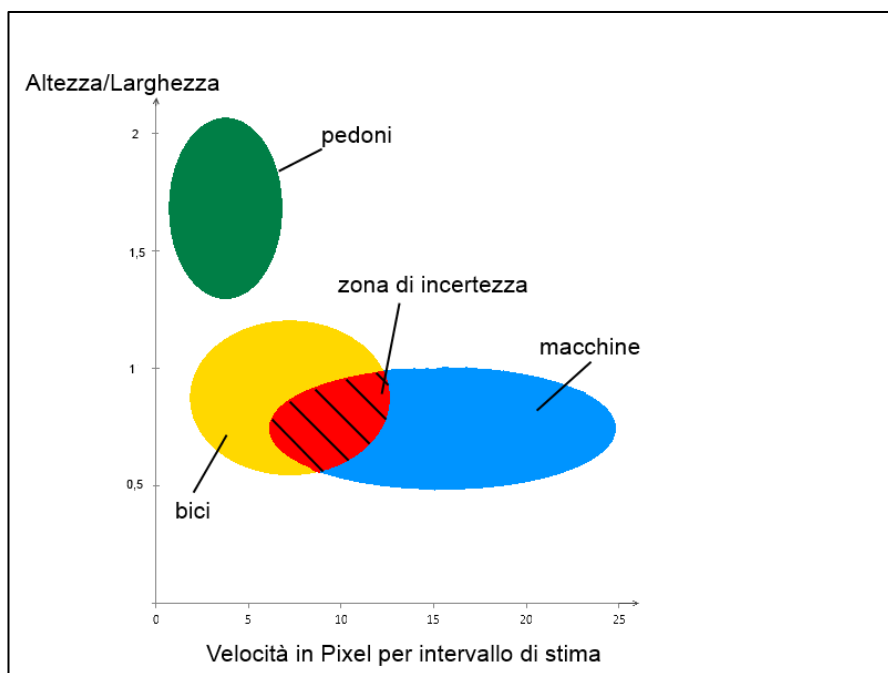


FIGURA 5-11 AREE DI CLASSIFICAZIONE DEI VEICOLI

6. ANALISI DEI RISULTATI

Di seguito vengono riportati i risultati dell'utilizzo della MouseCamera come sistema di gestione del traffico.

I test iniziali sono stati effettuati su una strada di campagna, stretta e poco frequentata (circa 100 veicoli l'ora). Questi i risultati:

	MouseCamera	Manuale
Pedoni	10	8
Cicli	24	42
Automobili	52	50
Cicli-Automobili	76	92
Totale	79	100

TABELLA 2 RISULTATI SU STRADA A BASSA FREQUENTAZIONE

Con i seguenti scostamenti percentuali:

Tipo di Veicolo	Scostamento %
Pedoni	25,0%
Cicli	42,9%
Automobili	4,0%
Cicli-Automobili	17,4%
Totale	14,0%

TABELLA 3 SCOSTAMENTO DALLA REALTÀ DEI RISULTATI OTTENUTI SU STRADA A BASSA FREQUENTAZIONE

Come si evince dagli scostamenti rilevati, il sistema ha una buona tolleranza per quanto riguarda il conteggio delle automobili, mentre si dimostra debole per i cicli.

Il tipo di strada su cui sono stati effettuati i primi test però non dava un campione soddisfacente per quanto riguarda il caso d'uso reale di utilizzo. Si è spostata quindi l'attenzione verso strade più larghe e maggiormente trafficate.

Effettuando alcuni test su strada a due corsie, mediamente trafficata (circa 500 veicoli l'ora), con transito a maggioranza di macchine, si sono avuti i seguenti risultati di conteggio, prendendo come riferimento il valore valutato manualmente:

	MouseCamera	Manuale
Pedoni	13	18
Cicli	127	81
Automobili	323	454
Cicli-Automobili	450	535
Totale	463	553

TABELLA 4 RISULTATI SU STRADA URBANA

Dai risultati si nota come il numero dei cicli conteggiati sia, al contrario delle altre categorie, superiore al numero reale transitato del 56,8%. Questo è appunto dovuto ad una problematica relativa alla non corretta interpretazione dei

parametri di classificazione per mezzi che si trovano su corsie diverse. Automobili che transitano sulla corsia più lontana dal sensore vengono considerati più lenti rispetto a quelli che transitano più vicino, e quindi si ha la possibilità che siano classificati come cicli.

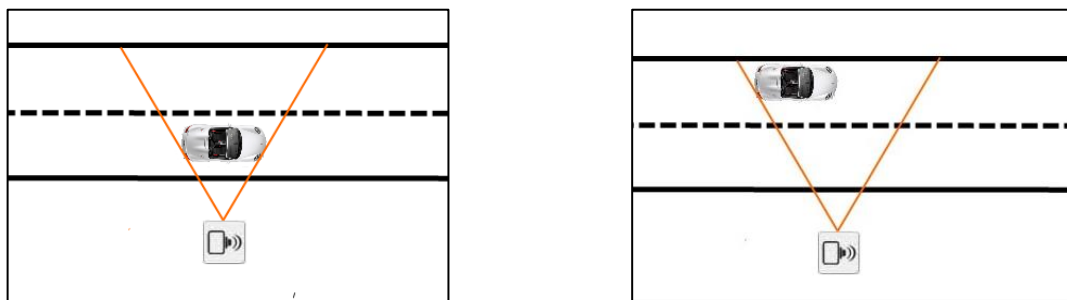


FIGURA 6-1 DIFFERENZA DI RILEVAZIONE TRA CORSIE

Per ogni categoria di veicolo abbiamo i seguenti scostamenti dal risultato reale:

Categoria	Scostamento %
Pedoni	27,8%
Cicli	56,8%
Automobili	28,9%
Cicli-Automobili	15,9%
Totale	16,3%

TABELLA 5 SCOSTAMENTO DALLA REALTÀ DEI RISULTATI OTTENUTI SU STRADA MEDIAMENTE
TRAFFICATA

In un test successivo sono stati esaminati i risultati del conteggio separatamente per direzione degli oggetti in movimento, mantenendo il sensore a distanza ed un'altezza maggiore rispetto all'altro test.

	Verso Destra		Verso Sinistra		Entrambe le direzioni	
	Contati	Reali	Contati	Reali	Contati	Reali
Pedoni	4	3	5	3	9	6
Cicli	35	38	38	25	73	63
Macchine	155	170	161	209	316	379
Macchine +Cicli	190	208	199	234	389	442
Totale	194	211	204	237	398	448

TABELLA 6 ANALISI PER DIREZIONE DEL CONTEGGIO

I risultati mostrano un netto miglioramento rispetto all'esperimento precedente, visto lo scostamento totale del 10,4% contro il 16,3% prima ottenuto. Dai dati si vede la differenza di efficacia nel classificare i veicoli a seconda della direzione.

Essendo una strada dedicata al traffico di veicoli, il conteggio del passaggio di pedoni può essere considerato non rilevante ai fini dell'esperimento, ma comunque ci sono state false rilevazioni di pedoni (9 contro i 2 reali).

Tralasciando quindi i pedoni, la differenza più evidente è quella nella classificazione dei cicli, dove vediamo che, per la direzione destra, cioè quella nella corsia più vicina al sensore, abbiamo uno scostamento del 7,9% contro il 52% della direzione destra.

	Verso Destra	Verso Sinistra	Entrambe le direzioni
Pedoni	25,0%	40,0%	33,3%
Cicli	7,9%	52,0%	15,9%
Macchine	8,8%	23,0%	16,6%
Macchine+Cicli	8,7%	15,0%	12,0%
Totale	8,0%	13,9%	11,2%

TABELLA 7 SCOSTAMENTI PER DIREZIONE

I parametri di classificazione perciò vanno differenziati a seconda della direzione del veicolo da rilevare, essendo quelli che transitano nella corsia più lontana rilevati come oggetti più piccoli e lenti. Un'altra soluzione potrebbe essere quella di utilizzare la MouseCamera per i conteggi in una sola direzione.

Oltre alla distanza del passaggio dal sensore, i problemi trovati nella rilevazione sono anche:

- Sovrapposizione dei passaggi nelle due direzioni
- Contiguità di passaggi consecutivi

Nel caso di sovrapposizione dei passaggi, si hanno due veicoli che, nello stesso intervallo di rilevazione, passano davanti al sensore andando in direzioni diverse.

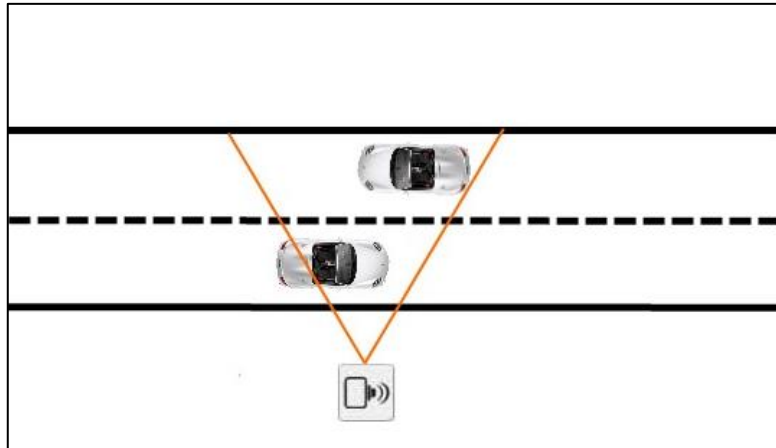


FIGURA 6-1 SOVRAPPOSIZIONE DI PASSAGGI

Il contatore effettua una media di ciò che vede muoversi ed il risultato è imprevedibile.

Altro problema è la contiguità nel passaggio dei veicoli.

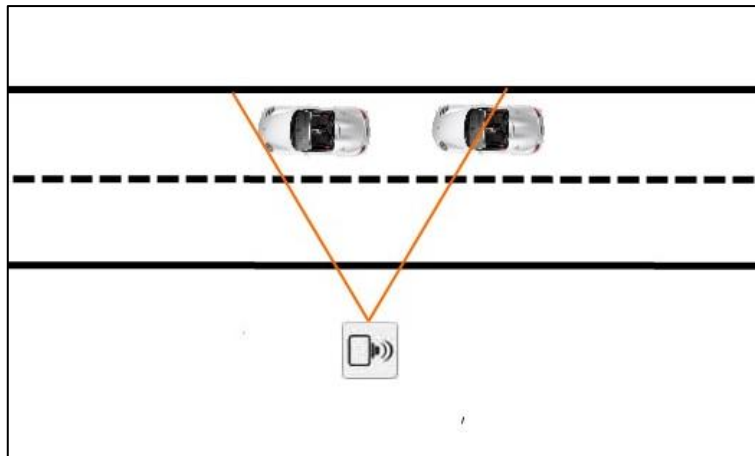


FIGURA 6-2 CONTIGUITÀ NEI PASSAGGI

Il sensore considera i due veicoli come un oggetto solo, mancando la rilevazione di uno (o più) di questi.

Una possibile soluzione per i problemi della sovrapposizione e della contiguità è quella di posizionare la MouseCamera più rialzata rispetto alla strada, in modo da poter dividere la strada in due fasce di rilevazione.

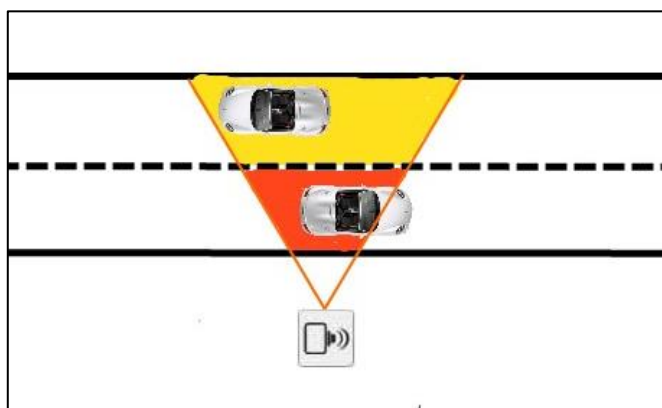


FIGURA 6-3 DIVISIONE DELLA CARREGGIATA IN DUE FASCE DI RILEVAZIONE

7. CONCLUSIONI

Nei precedenti capitoli è stata descritta una particolare tecnologia di smart camera a bassa risoluzione, progettata per la rilevazione del traffico stradale.

Rispetto alle altre tecnologie di rilevatori del traffico, la MouseCamera è un dispositivo a basso costo, non invasiva e portatile, viste le sue ridotte dimensioni. Può essere utilizzata a lato di qualunque tipologia di strada e da una stima del conteggio e della classificazione dei veicoli rilevati.

La particolarità dell'hardware risiede nel sensore scelto, l'ADNS-3060. Questo, essendo progettato per l'utilizzo su mouse, ha una risoluzione di 900 pixel se utilizzato come sensore di immagini, e non di movimento. La bassa risoluzione permette l'utilizzo di altri elementi a basso costo e ridotte dimensioni da affiancare al sensore.

Le immagini rilevate dal sensore ADNS-3060, essendo affette da rumore, hanno bisogno di una pre-elaborazione. Sono state descritte due tecniche che hanno migliorato la qualità delle immagini rilevate quali la Pedestal Correction e la Gamma Correction.

Dovendo la MouseCamera lavorare in spazi aperti, questa sarà soggetta a cambi di illuminazione dovuti alla situazione meteorologica e a riflessi. È stata descritta una tecnica di adattamento alle variazioni di luminosità, basata sul meccanismo interno del sensore ADNS-3060, di controllo dell'otturatore automatico.

Per cercare di rilevare movimenti a differenti velocità, sulla MouseCamera sono stati implementati tre motori di stima del movimento a frame rate differenti. I risultati dei tre motori vengono fusi per poi essere classificati e conteggiati tramite le tecniche di erosion, labeling e snake, implementate ad hoc per il microprocessore utilizzato all'interno della MouseCamera.

Inoltre sono stati analizzati i risultati del collaudo su due tipologie di strade, una stretta, a bassa frequentazione e una a due corsie ad alta frequentazione. I dati mostrano la difficoltà del sistema nella classificazione dei veicoli, soprattutto nel distinguere macchine da cicli. Il posizionamento di questo rilevatore è punto critico, in quanto al variare di questo, le forme dei veicoli analizzate cambiano. Il sistema necessita quindi di uno studio sul posizionamento più approfondito, o della progettazione di un protocollo di calibrazione. Successivamente andranno rivisti i parametri di classificazione degli oggetti, calcolati sperimentalmente mantenendo la posizione scelta.

Si evince dai risultati il rapporto inverso tra distanza di rilevazione e lo scostamento dalla realtà, che porta a pensare al prossimo stadio di progettazione di questo rilevatore, cioè l'utilizzo di due MouseCamera, opposte, una per lato della strada. Attraverso la comunicazione wireless, di cui è stato progettato il protocollo, è possibile scambiare dati fra i dispositivi per condividere le informazioni sui veicoli rilevati, per una maggiore precisione nella rilevazione e nella classificazione.

8. Bibliografia

- [1] Kleihorst R., Camilli M., *Mouse Sensor Networks, the smart camera*, (Mol, Belgium), 2011.
- [2] Hengstler, S Prashanth, D., Sufen Fong, Aghajan, H., “*MeshEye: A Hybrid-Resolution Smart Camera Mote for Applications in Distributed Intelligent Surveillance*”, (Stanford Univ., Stanford), April 2007.
- [3] G. de Haan, P. Biezen, H. Huijgen, and O. Ojo, “*True-motion estimation with 3-d recursive-search block matching*,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 3, pp. 368–379, Oct 1993.
- [4] Agilent Technologies, “*Agilent ADNS-3060 High-performance Optical Mouse Sensor Data Sheet*”, Oct.2004
- [5] Microchip Technology Inc., “*dsPIC33FJ32GP302/304, dsPIC33FJ64GPX02/X04, and dsPIC33FJ128GPX02/X04 Data Sheet: High-Performance, 16-bit Digital Signal Controllers*”, 2011
- [6] Hope Microelectronics Co., “*RFM12B Universal ISM Band FSK Transceiver Datasheet*”, 2006

- [7] Imagsa Inc, "*Atalaya*", (Barcelona, Spain), 2011

- [8] K. Fürstenberg, J. Hipp, A. Liebram, "A laser scanner for detailed traffic data collection and traffic control", (Hamburg, Germany), 2000

- [9] Sodi Scientifica, "KV Laser Traffic Counter: Statistic Vehicle Detector", (Calenzano, Italy), Mar. 1999

- [10] Quaritsch M., Rinner B., "*Dscagents: a lightweight middleware for distributed smart cameras*", (Klagenfurt University, Austria), 2009

- [11] Kleihorst R., Schueler B., Danilin A., "*Architecture and Applications of wireless Smart Cameras (Networks)*," in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2007), (Honolulu, Hawaii, USA), Apr. 2007.

- [12] Rinner B., Jovanovic M., Quaritsch M., "*Embedded middleware on distributed smart cameras*", (Graz University of Technology, Austria), 2007

- [13] Wei B. N., Shi Y., Ye G., Xu J., "*Developing a Smart Camera for Road Traffic Surveillance*", Sydney, Australia, 2008

9. Tabella dei contenuti

Figura 2-1	Fotografia di sensore invasivo	9
Figura 2-2	Fotografia di rilevatore basato su sensori di pressione	10
Figura 2-3	Schema di utilizzo del sensore a pressione	11
Figura 2-4	Fotografia di contatore basato su Radar	12
Figura 2-5	Fotografia di contatore basato su sensori laser	13
Figura 2-6	Schema di utilizzo dei contatori con tecnologia laser	13
Figura 2-7	Fotografia di un contatore Laser in grado di classificare i veicoli	14
Figura 2-8	Schema di utilizzo di un classificatore Laser	15
Figura 2-9	Esempi di risultati della classificazione tramite contatore Laser	15
Figura 2-10	Fotografia della Smart Camera Atalaya	17
Figura 2-11	Architettura della Smart Camera Atalaya.....	18
Figura 2-12	Diagramma a blocchi del flusso di processo di Trafficom	20
Figura 3-1	Fotografie della MouseCamera.....	24
Figura 3-2	Disegno dell'architettura della MouseCamera	25
Figura 3-3	Immagine del sensore ottico	26
Figura 4-1	Matrice di correzione dell'immagine	32
Figura 4-2	Risultato della Pedestal Correction.....	32
Figura 4-3	Grafico della correzione Gamma. In ascissa il valore dei pixel in ingresso, in ordinata il valore dei pixel in uscita dalla correzione	33

Figura 4-4Trasformazione non lineare per la correzione gamma	34
Figura 4-5Topologia della rete di sensori.....	35
Figura 4-6Flusso di comunicazioni durante il meccanismo di discovery ...	37
Figura 4-7Definizione dei messaggi di comunicazione	38
Figura 4-8Dettaglio dei campioni su cui viene applicato l’algoritmo 3DRS di Motion Estimation	42
Figura 5-1Suddivisione del tempo di elaborazione nella MouseCamera....	47
Figura 5-2Erosion: risultato del conteggio delle celle vicine	50
Figura 5-3Erosion: risultato finale dell'algoritmo	51
Figura 5-4Labeling: situazione iniziale	52
Figura 5-5Labeling: situazione al primo passaggio.....	53
Figura 5-6Labeling: risultato finale dell’algoritmo	54
Figura 5-7Snake: risultato finale	56
Figura 5-8Rilevazione di un pedone.....	57
Figura 5-9Rilevazione di una Bicicletta	58
Figura 5-10Rilevazione di una macchina	59
Figura 5-11Aree di classificazione dei veicoli	60
Figura 6-1Sovrapposizione di passaggi.....	66
Figura 6-2Contiguità nei passaggi.....	66
Figura 6-3Divisione della carreggiata in due fasce di rilevazione	67