

UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

---

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

E DELL'AUTOMAZIONE



**Implementazione di un algoritmo di  
identificazione della persona  
utilizzando frame di profondità**

---

***Implementation of a depth-based human  
identification algorithm***

RELATORE:

**Prof. Ennio Gambi**

CORRELATORI:

**Prof.ssa Susanna Spinsante**

**Ing. Samuele Gasparrini**

TESI DI LAUREA DI:

**Ilario Pierbattista**

---

ANNO ACCADEMICO 2014/2015

# Indice

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduzione</b>                              | <b>3</b>  |
| 1.1      | Human Sensing                                    | 3         |
| 1.1.1    | Human Sensing                                    | 3         |
| 1.1.2    | Stato dell'arte                                  | 4         |
| 1.2      | Panoramica Generale                              | 4         |
| 1.2.1    | Introduzione al lavoro di Zhu Wong               | 4         |
| 1.2.2    | Configurazione dell'Hardware                     | 4         |
| 1.2.3    | <i>Head and Shoulders Profile</i>                | 5         |
| 1.2.4    | Flusso di Lavoro                                 | 6         |
| <b>2</b> | <b>Haar-Like Features</b>                        | <b>8</b>  |
| 2.1      | Definizione                                      | 8         |
| 2.1.1    | Richiamo: cosa è una feature (caratteristica)    | 8         |
| 2.1.2    | Wavelet di Haar                                  | 8         |
| 2.1.3    | Formula di calcolo standard                      | 9         |
| 2.1.4    | Cosa mette in evidenza la feature di Haar        | 9         |
| 2.1.5    | Formula di calcolo invariante ai resize          | 10        |
| 2.1.6    | Vantaggi   | 10        |
| 2.2      | Immagine Integrale                               | 11        |
| 2.2.1    | Definizione rigorosa dell'immagine integrale     | 11        |
| 2.2.2    | Complessità computazionale generale              | 11        |
| 2.3      | Decision Stump                                   | 12        |
| 2.3.1    | Problema: utilizzare le feature                  | 12        |
| 2.3.2    | Definizione di albero decisionale                | 12        |
| 2.3.3    | Definizione di decision stump                    | 12        |
| <b>3</b> | <b>L'Algoritmo di Allenamento: Adaboost</b>      | <b>13</b> |
| 3.1      | Apprendimento Supervisionato <i>Ensamble</i>     | 13        |
| 3.1.1    | Apprendimento Supervisionato                     | 13        |
| 3.1.2    | <i>Adaptive Boosting</i>                         | 14        |
| 3.2      | Dataset di Allenamento                           | 14        |
| 3.2.1    | Categorie di Classificatori                      | 14        |
| 3.2.2    | Preparazione dei Dataset                         | 14        |
| 3.2.3    | Preprocessing                                    | 14        |
| 3.3      | <i>Strong Learner</i>                            | 15        |
| 3.3.1    | Procedura di estrazione del classificatore forte | 15        |

|          |   |           |
|----------|---|-----------|
| 3.4      | <i>Weak Learner</i>                                 | 15        |
| 3.4.1    | Procedura di estrazione del classificatore debole   | 15        |
| 3.4.2    | Valutazione della complessità computazionale        | 15        |
| <b>4</b> | <b>Validazione e Regolazione dei Classificatori</b> | <b>16</b> |
| 4.1      | Criteri di Valutazione                              | 16        |
| 4.2      | Dataset di Validazione                              | 16        |
| 4.2.1    | Criteri di creazione delle registrazioni            | 16        |
| 4.2.2    | Altre caratteristiche                               | 16        |
| 4.3      | Massimizzazione all' <i>Accuracy</i>                | 16        |
| 4.3.1    | Parametri liberi del classificatore                 | 16        |
| 4.3.2    | Algoritmo di ricerca della soglia e del NWL ottimi  | 16        |
| 4.4      | Analisi dei Risultati                               | 16        |
| <b>5</b> | <b>Rilevamento</b>                                  | <b>17</b> |
| 5.1      | Tecnica di Rilevamento                              | 17        |
| 5.1.1    | Detection Window                                    | 17        |
| 5.1.2    | Rilevazione su frame                                | 17        |
| 5.2      | Selezione della Finestra Migliore                   | 17        |
| 5.2.1    | Introduzione al problema                            | 17        |
| 5.2.2    | Algoritmo di selezione                              | 17        |
| 5.3      | Confronto con l'Algoritmo G-C                       | 17        |
| <b>6</b> | <b>Conclusioni</b>                                  | <b>18</b> |
|          | <b>Appendici</b>                                    | <b>19</b> |
| <b>A</b> | <b>Software Sviluppato</b>                          | <b>20</b> |
| A.1      | Componenti  | 20        |
| A.1.1    | Creatore dei Dataset                                | 20        |
| A.1.2    | Allenamento   | 20        |
| A.1.3    | Tuning, Testing, Rilevamento                        | 20        |
| A.2      | Tecnologie utilizzate                               | 20        |
| A.2.1    | C++ e Matlab  | 20        |
| A.2.2    | Git e Github [Opzionale]                            | 20        |
| A.3      | Proposte di miglioramento                           | 20        |
| <b>B</b> | <b>Cenni del funzionamento del sensore Kinect</b>   | <b>21</b> |

# Capitolo 1

## Introduzione

### 1.1 Human Sensing

#### 1.1.1 Human Sensing

##### 5 Definizione

L'insieme di tecniche di riconoscimento della presenza di una persona nello spazio prendono il nome di tecniche di *human sensing*.

Sensori di vario tipo vengono utilizzati nelle tecniche di riconoscimento.

Una volta acquisite le informazioni dai sensori, vanno elaborate da un apposito  
10 algoritmo per rilevare la presenza e la posizione della persona nell'ambiente.

#### Contesti Applicativi

La possibilità di rilevare la presenza di persone, rende le applicazioni di *human sensing* perfette per le applicazioni di sorveglianza.

Sistemi di *people counting* sono utili per la conduzione di indagini di mercato.

15 Dispositivi in grado di rilevare la presenza di corpi umani in contesti di crisi sono utilizzati nelle attività di *search rescue*.

Le applicazioni di *human tracking* sono utili anche negli ambienti assistivi automatizzati al fine di monitorare le attività dell'assistito.

#### HS e Computer Vision

20 Le applicazioni di human sensing che utilizzano sensori di acquisizione *visiva* risolvono problemi di computer vision.

Lo scopo della *computer vision* è quello di riprodurre la vista umana. L'obiettivo di tale riproduzione non si limita alla semplice acquisizione di una rappresentazione bidimensionale di una regione di spazio, ma mira all'interpretazione del relativo  
25 contenuto.

### 1.1.2 Stato dell'arte

#### Pedestrian Detection and Counting

Papageorgiou et Al [6] hanno sviluppato un sistema di riconoscimento e conteggio di pedoni a partire da immagini RGB.

#### 30 Face Recognition

Viola e Jones [9] hanno proposto un framework per il riconoscimento dei volti all'interno di immagini RGB. Al momento è il sistema più solido nel suo contesto.

#### Kinect: a serious game

35 Gli ambiti d'utilizzo del dispositivo Kinect, nota periferica legata a sistemi videoludici, si stanno espandendo costantemente. La quantità e la qualità dei sensori di cui è equipaggiato, il costo relativamente contenuto e l'evoluzione di framework e toolkit di sviluppo, lo rendono un dispositivo particolarmente versatile e adatto allo studio di problemi di computer vision.

## 1.2 Panoramica Generale

#### 40 1.2.1 Introduzione al lavoro di Zhu Wong

##### Elenco delle tecnologie coinvolte

La sorgente di informazione è il sensore di profondità del Kinect.

Il sistema descritto prevede l'utilizzo di un algoritmo di allenamento per sviluppare i criteri di riconoscimento della persona.

45 Ciò che viene presentato da Zhu e Wong è un sistema di rilevamento che prende notevolmente in considerazione le soluzioni proposte da Viola e Jones, eccezion fatta, naturalmente, per il dispositivo di acquisizione.

### 1.2.2 Configurazione dell'Hardware

#### Sensore utilizzato

50 Il sensore di profondità del Kinect V2 fornisce una rappresentazione bidimensionale dello spazio sotto forma di immagini. In tali immagini ogni pixel corrisponde il valore in millimetri della distanza dal sensore della superficie dell'oggetto interessato.

Ci si riferirà a tali immagini chiamandole *immagini di profondità*. Il sensore del Kinect, di cui è disponibile una piccola descrizione più dettagliata all'appendice B, fornisce 55 uno stream di tali immagini ad una frequenza di 30 frame al secondo: è possibile quindi registrare dei *video di profondità*.

#### Configurazione Top-Down

Il dispositivo Kinect viene montato al soffitto di una stanza e l'ambiente viene ripreso da tale prospettiva. Solitamente l'altezza a cui viene montato è di poco inferiore alla 60 distanza del soffitto dal pavimento (poco meno di 2m). La linea focale del sensore

dovrebbe essere quanto più possibile ortogonale al pavimento della stanza, in modo da ridurre ai minimi termini la presenza di asimmetrie nelle riprese. Tali distanze sono perfettamente compatibili con le specifiche tecniche del dispositivo stesso. Nel caso in cui vi sia la necessità di montare il Kinect a soffitti più alti di 4m, si possono utilizzare  
65 delle lenti correttive per aumentare il range di affidabilità del sensore.

Molti sistemi di riconoscimento utilizzano il Kinect in posizione frontale ai soggetti da riconoscere. Di fatti, il dispositivo, concepito per applicazioni videoludiche, è progettato per operare in tale posizione. Tuttavia, la configurazione descritta precedentemente, ha l'enorme vantaggio di eliminare l'occlusione del soggetto: normalmente una persona non  
70 può nascondere dietro di sé un'altra persona alla vista del sensore (se non in scomode posizioni), cosa frequentissima invece con i sistemi di rilevamento frontali.

### 1.2.3 *Head and Shoulders Profile*

#### **L'attività di riconoscimento è un'attività di classificazione**

Ciò che bisogna riconoscere è, all'interno di un'immagine di profondità, la figura della  
75 persona.

Considerando un altro punto di vista, l'attività di riconoscimento consiste nel discriminare gli oggetti che sono figure di persone, da oggetti che non lo sono.

Si definiscono quindi due classi. Il concetto di classe è molto simile alle *classi di equivalenza* dell'algebra astratta e costituiscono degli insiemi di oggetti che condividono  
80 determinate proprietà. Distinguere gli oggetti che rappresentano persone da quelli che non le rappresentano, significa classificare tali oggetti in due classi: le persone e le non persone. Il processo di rilevamento, quindi, si basa sulla determinazione della classe di appartenenza dei vari oggetti: *classificazione*.

#### **La classificazione si basa sulla misurazione di alcune caratteristiche**

Gli oggetti di una stessa classe hanno alcune proprietà in comune, ma differiscono per  
85 altre. Individuare le caratteristiche - ovvero proprietà osservabili e misurabili di un oggetto - in base alle quali discriminarli nelle due classi non è un problema banale. Si può intuire quanto sia vasto l'insieme delle caratteristiche valutabili nella rappresentazione di un oggetto. Ovviamente la natura della rappresentazione influisce nella scelta delle  
90 caratteristiche più rilevanti. Nei capitoli successivi verrà presentato un algoritmo che automatizzerà la selezione delle caratteristiche più rilevanti dell'immagine.

#### **Caratteristiche del profilo HASP in linguaggio naturale**

Un'immagine di profondità, per sua natura, rappresenta la realtà attraverso il valore della distanza misurata in ogni punto dello spazio osservato. È naturale, quindi, considerare  
95 tali distanze come caratteristiche misurabili dell'oggetto rappresentato.

È utile quindi fornire una descrizione, se non altro in linguaggio naturale, della forma del profilo umano ripreso dall'alto, obiettivo del riconoscimento. Tale descrizione è informale.

- 100 1. L'immagine di una persona è caratterizzata da uno *spazio vuoto*<sup>1</sup> di fronte ad essa e dietro di essa.
2. A sinistra della spalla sinistra ed a destra della spalla destra del profilo dall'alto di una persona, sono presenti degli spazi vuoti.
3. Tra la testa e le spalle vi è una differenza di altezza.

### 1.2.4 Flusso di Lavoro

#### 105 Definizione dei moduli funzionali

Un modulo software sarà dedicato all'allenamento.

Un modulo software sarà dedicato al rilevamento.

#### Allenamento

110 Per allenare il sistema è necessario creare un insieme di allenamento, ovvero un insieme i cui elementi sono delle immagini che ritraggono persone e non. In fase di creazione, ogni elemento viene dotato di un'etichetta che identifica la classe di appartenenza reale dell'oggetto.

La componente software che si occupa dell'allenamento del sistema implementa l'algoritmo Adaboost. Quest'ultimo riceve in input l'insieme di allenamento, i cui elementi,  
115 dotati della rispettiva classificazione reale, sono alla base della scelta delle caratteristiche migliori per la descrizione delle classi di oggetti.

Alla fine della sua esecuzione, Adaboost restituisce come output un classificatore. Nei capitoli successivi si darà una definizione più formale di quello che è un classificatore. Per il momento è sufficiente una definizione intuitiva: un classificatore *classifica* i vari oggetti,  
120 ovvero fornisce una *previsione* della relativa classe di appartenenza. La classificazione effettuata da questa componente approssima solamente la classificazione reale. La bontà di tale approssimazione sarà il parametro di valutazione della bontà generale del sistema. Nel caso di Adaboost il classificatore risultante sarà simile ad una collezione di test: il risultato di tali test, eseguiti su di un qualsiasi oggetto, fornirà la previsione della  
125 classificazione dell'oggetto stesso.

#### Rilevamento

In questa fase il sistema analizza i frame di profondità delle acquisizioni in ordine sequenziale, alla ricerca di persone al suo interno.

Il classificatore ottenuto al termine dell'esecuzione di Adaboost, sarà in grado di  
130 classificare porzioni di immagini di profondità, ma non è in grado di predire direttamente, a partire da un intero frame, la presenza e la posizione di una persona al suo interno. Le porzioni analizzabili dal classificatore hanno dei vincoli dimensionali da rispettare. In prima approssimazione si può pensare a tali porzioni come a dei quadrati di dimensione costante. L'attività di rilevamento della persona all'interno del frame, quindi, conterà

---

<sup>1</sup>Per *spazio vuoto* si intende una regione di spazio il cui valore della distanza, percepita dal sensore, è molto vicino al quello della distanza del pavimento della stanza.

135 della sequenziale analisi di tutte le porzioni di frame che rispettano tali vincoli, al fine  
di coprire l'intera area.

Si vedrà in seguito che nei pressi di una persona nell'immagine di profondità, saranno molteplici le porzioni di frame per le quali il rilevamento darà esito positivo. Si pone quindi l'ulteriore problema di selezionare, delle tante porzioni che hanno dato esito  
140 positivo, quella che meglio approssima la reale posizione della persona.



## Capitolo 2

# Haar-Like Features

### 2.1 Definizione

#### 2.1.1 Richiamo: cosa è una feature (caratteristica)

145 Le caratteristiche di un oggetto sono quelle proprietà elementari osservabili e misurabili.  
È stato già detto che la scelta delle caratteristiche è fondamentale e dipende da cosa si vuole mettere in evidenza dell'oggetto in questione.

Ovviamente la scelta delle caratteristiche è sempre subordinata a ciò che si ha disposizione.

#### 150 2.1.2 Wavelet di Haar

**Le feature di Haar derivano dalle wavelet di Haar**

Le feature di Haar si adattano molto bene alle proprietà che si vogliono misurare degli oggetti appartenenti all'applicazione d'interesse. Sono un costrutto derivante dalle *wavelet di Haar*.

#### 155 Definizione informale delle wavelet di Haar

Alfréd Haar sviluppò il primo tipo di wavelet.

Furono sviluppate come un esempio di funzioni ortonormali di base per uno spazio funzionale.

160 In quanto base di uno spazio ortornomale, con le wavelet di Haar è possibile esprimere un qualsiasi segnale limitato. Esse costituiscono, sotto particolari ipotesi, un sistema di rappresentazione dei segnali duale all'analisi spettrale di Fourier. Hanno anche il vantaggio, rispetto a quest'ultimo, di mantenere l'informazione del tempo (approfondire).

#### **Wavelet di Haar e DWT**

Le wavelet di Haar sono state utilizzate nelle *Discrete Wavelet Transform*, in breve DWT.

165 Un'importante applicazione delle DWT è quella definita dal nuovo standard di compressione delle immagini JPEG2000.

Nelle applicazioni di machine learning e pattern recognition, le trasformazioni DWT furono utilizzate nei primi lavori di riconoscimento a partire da immagini RGB (riconoscimento dei pedoni). È da quest'ultima applicazione che hanno origine le feature di Haar come verranno trattate.

### 2.1.3 Formula di calcolo standard

#### Rappresentazione visuale

Le feature di Haar sono rappresentabili come due aree adiacenti, una chiara ed una scura. La somma delle intensità (il valore numerico) di tutti i pixel dell'area scura, viene sottratta alla somma delle intensità di tutti i pixel nell'area chiara. Ciò permette di evidenziare le differenze di intensità medie tra i valori dei pixel nelle due aree. [Formula a due aree] È da notare il fatto che, con le feature di Haar, non si vanno a valutare i singoli pixel al fine di individuarne un pattern, ma si ragiona procedendo per aree adiacenti.

#### Formula generale

La definizione delle feature di Haar può essere estesa all'utilizzo di più di due aree adiacenti. Il principio resta lo stesso: si hanno due gruppi di aree, uno chiaro ed uno scuro. La formula di calcolo si generalizza con estrema semplicità. [Formula generale con più aree]

#### Altri tipi di feature (OpenCv)

È possibile formulare feature di moltissime formule. La libreria di computer vision OpenCV mette a disposizione una grande varietà di feature, introducendo anche quelle la cui forma è inclinata di 45°. (Citare articolo in cui vi è la definizione delle feature a 45°).

#### Tipi di feature utilizzate

In questa applicazione vengono utilizzati solo due 4 tipi di feature, contro i 5 utilizzati nel sistema di riconoscimento di Viola-Jones.

### 2.1.4 Cosa mette in evidenza la feature di Haar

#### Immagini normali (Viola Jones)

Nella framework di riconoscimento dei volti di Viola-Jones, le immagini RGB sono la fonte di informazioni del sistema. Vengono applicate le feature di Haar a tali immagini (ovviamente dopo che queste ultime sono state trattate da opportune operazioni di preprocessing) e ciò che viene evidenziato sono le differenze di intensità tra le regioni della foto. A far variare l'intensità di un pixel in una foto concorrono l'illuminazione, il colore e molti altri fattori. Tuttavia, le feature di Haar divengono il mezzo con il quale si può riconoscere un volto umano. Ad esempio, è stato osservato che nella foto di un volto, l'area che racchiude entrambi gli occhi e l'inizio del naso è caratterizzato da una

particolare variazione di luminosità, misurabile e utilizzabile per discriminare le i volti da i non volti.

## 205 Immagini di profondità (Zhu Wong)

Nelle immagini di profondità, dove il valore di ogni pixel corrisponde alla distanza in millimetri della superficie dal sensore, applicare le feature di Haar ad un'area dell'immagine equivale a misurare le differenze di quota tra due aree adiacenti tra di loro. Il sistema di allenamento deciderà quali sono le feature migliori per il riconoscimento della  
210 persona, ma tutto si basa sul concetto che le differenze di quota osservabili dal profilo ripreso dall'alto di una persona sono caratteristiche della persona stessa e costituiscono il parametro di riconoscimento rispetto ad un qualsiasi altro oggetto. Una sedia, un tavolo o qualsiasi altro elemento presenterà delle differenze di quota differenti dal profilo della persona.

## 215 2.1.5 Formula di calcolo invariante ai resize

### Anticipazione del problema del ridimensionamento

In seguito sarà necessario ridimensionare una feature in modo da coprire un'area più grande, in quanto, dovendo misurare le caratteristiche degli oggetti di interesse, questi ultimi sono variabili in dimensione. Il valore calcolato con la feature in questione, tuttavia,  
220 non dovrebbe essere troppo sensibile ai ridimensionamenti (che saranno frequenti).

### Formula: Normalizzazione sull'area

Al fine di ottenere la massima invarianza ai ridimensionamenti dell'area della feature, il valore di essa viene normalizzato con l'estensione dell'area totale valutata. [Formula normalizzata]

## 225 2.1.6 Vantaggi

### Differenze di intensità vs Valutazione dei singoli pixel

Il primo indiscutibile vantaggio delle feature di Haar sta nel fatto che la caratterizzazione dell'oggetto viene effettuata sulla base di osservazioni d'insieme su intere aree dell'immagine e non su ossevizioni globali effettuate su singoli pixel. Oltre alla grandissima  
230 complessità che una valutazione su singoli pixel introdurrebbe, bisogna prendere atto che, con dati soggetti a disturbi e alla presenza di rumore, delle caratteristiche misurate sui singoli pixel non sarebbero molto significative.

### Differenze di intensità vs Estrazione dei contorni

L'estrazione dei contorni potrebbe essere più significativo della valutazione sui singoli  
235 pixel, ma continuano ad essere caratteristiche abbastanza complesse da calcolare ed ottenere. L'estrazione dei contorni, inoltre, è un concetto fortemente legato alle immagini RGB, usarlo con le immagini di profondità è un forzatura.

### Estrema efficienza computazionale

Il principale vantaggio delle feature di Haar rispetto ad altri tipi più elaborati di feature resta la loro efficienza computazionale. Si vedrà che, con una particolare struttura dati di supporto, calcolare il valore di un feature di Haar per un'immagine è un'operazione eseguibile in un tempo costante.

## 2.2 Immagine Integrale

### 2.2.1 Definizione rigorosa dell'immagine integrale

**Problema: efficienza nel calcolo di somme di pixel**

Il calcolo della somma delle intensità di ciascun pixel appartenente ad un'area, necessario al fine di calcolare il valore delle feature di Haar, è un'operazione il cui costo varia all'aumentare della dimensione complessiva dell'area. Calcolare tali somme infatti ha complessità computazionale  $\Theta(m \cdot n)$  con  $m$  ed  $n$  dimensione dell'area.

La soluzione a tale problema consiste nell'utilizzo dell'immagine integrale, una struttura dati che mette a permette di calcolare la somma dei pixel di qualsiasi area all'interno di essa in un tempo costante.

#### Definizione immagine integrale

Un'immagine integrale è un matrice delle stesse dimensioni dell'immagine di partenza. Ogni elemento di tale matrice contiene il valore della somma dei pixel che si trovano al di sopra e a destra (estremi inclusi) del pixel relativo alla posizione dell'elemento. [Formula]

#### Formula di calcolo della somma dei pixel in un'area

Attraverso l'immagine integrale è possibile calcolare l'area necessaria per una feature sommando i valori dei due vertici dell'area sulla diagonale principale e sottrandovi quelli dei vertici sulla diagonale secondaria. [Formula]

[Dimostrazione formula]

Qualsiasi calcolo di questo tipo richiederà un tempo costante, non più legato alle dimensioni dell'input. La complessità computazionale è quindi  $\Theta(1)$ .

### 2.2.2 Complessità computazionale generale

#### Complessità del calcolo dell'immagine integrale

La complessità computazione totale del calcolo dell'immagine integrale continua ad essere legata alla dimensione dell'input. Se l'immagine è larga  $w$  pixel ed alta  $h$  pixel, la complessità computazionale per la generazione dell'immagine integrale è pari a  $\Theta(w \cdot h)$ .

#### Convenienza del calcolo dell'immagine integrale

L'utilizzo delle immagini integrale è molto vantaggiosa nel momento in cui è necessario calcolare molte feature sulla stessa immagine. Volendo essere più espliciti, se la

complessità computazionale totale del calcolo di  $n$  feature senza l'utilizzo dell'immagine integrale è maggiore di quella per la generazione dell'immagine integrale stessa, allora è  
275 vantaggioso utilizzare tale struttura dati.

## 2.3 Decision Stump

### 2.3.1 Problema: utilizzare le feature

Una volta misurata una caratteristica di un oggetto, bisogna essere in grado di utilizzarla ai fini di classificarlo.

280 È necessario, quindi, un meccanismo per effettuare una previsione della classe di appartenenza dell'oggetto che si basa esclusivamente sul valore della feature misurata.

### 2.3.2 Definizione di albero decisionale

Un albero decisionale è un modello predittivo. Ogni nodo dell'albero rappresenta una variabile o una proprietà osservabile. Ogni arco dal nodo padre verso un nodo figlio  
285 rappresenta un possibile valore per la proprietà del nodo padre. Le foglie dell'albero rappresentano i valori che si predice possa assumere una variabile obiettivo. Il percorso dalla radice ad una foglia rappresenta la previsione del valore di tale variabile obiettivo a fronte delle osservazioni effettuate sui valori delle variabili rappresentate da ciascun nodo attraversato.

### 290 2.3.3 Definizione di decision stump

#### Radice: Test, funzione booleana

Il più semplice albero decisionale è il *decision stump*. Ha profondità unitaria, cioè si basa sull'osservazione di una sola variabile, in base al valore della quale può effettuare due diverse previsioni della variabile obiettivo (ha due foglie).

#### 295 Foglie: risultati possibili

#### Formule di calcolo binaria

#### Formula di calcolo unica: polarità

## Capitolo 3

# L'Algoritmo di Allenamento: Adaboost

300

### 3.1 Apprendimento Supervisionato *Ensamble*

#### 3.1.1 Apprendimento Supervisionato

Definizione

Obiettivo

305 Spazio delle Ipotesi

Esempi di Supervised learning

Algoritmi di Sup.Learning

Maggiori campi applicativi

Concetti di base

310 Overfitting

Ensamble Learning

### 3.1.2 *Adaptive Boosting*

Algoritmi di Boosting

Aptive: adattabilità

315 Strong learner e Weak learner

## 3.2 Dataset di Allenamento

### 3.2.1 Categorie di Classificatori

Variabilità della forma HASP

Variazione dell'orientazione

320 Variazione derivata dalla distorsione prospettica

Definizione delle categorie di classificatori

Categorie: Verticale e Orizzontale

Categorie alternative: Obliquo, a zone

### 3.2.2 Preparazione dei Dataset

325 Acquisizioni

Soggetti, percorsi

Acquisizione delle registrazioni

Ritaglio dei samples

Trainset Creator

330 3.2.3 Preprocessing

Resize

Nearest Neighbour

Altri algoritmi di resize

Conversione delle distanze

### 335 3.3 *Strong Learner*

3.3.1 Procedura di estrazione del classificatore forte

### 3.4 *Weak Learner*

3.4.1 Procedura di estrazione del classificatore debole

3.4.2 Valutazione della complessità computazionale



## Capitolo 4

# Validazione e Regolazione dei Classificatori

### 4.1 Criteri di Valutazione

### 4.2 Dataset di Validazione

#### 345 4.2.1 Criteri di creazione delle registrazioni

#### 4.2.2 Altre caratteristiche

### 4.3 Massimizzazione all'*Accuracy*

#### 4.3.1 Parametri liberi del classificatore

Numero di weak learner

#### 350 Soglia del classificatore

#### 4.3.2 Algoritmo di ricerca della soglia e del NWL ottimi

### 4.4 Analisi dei Risultati

## Capitolo 5

# Rilevamento

### 355 5.1 Tecnica di Rilevamento

#### 5.1.1 Detection Window

#### 5.1.2 Rilevazione su frame

Resize detection window

Slide detection window

### 360 5.2 Selezione della Finestra Migliore

#### 5.2.1 Introduzione al problema

#### 5.2.2 Algoritmo di selezione

### 5.3 Confronto con l'Algoritmo G-C

## Capitolo 6

# Conclusioni

# Appendici

# Appendice A

## Software Sviluppato

### A.1 Componenti

370 A.1.1 Creatore dei Dataset

A.1.2 Allenamento

A.1.3 Tuning, Testing, Rilevamento

### A.2 Tecnologie utilizzate

A.2.1 C++ e Matlab

375 A.2.2 Git e Github [Opzionale]

### A.3 Proposte di miglioramento

## Appendice B

### Cenni del funzionamento del sensore Kinect

# Bibliografia

- [1] Thomas H Cormen. Introduction to algorithms. 2009.
- [2] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.
- [3] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.
- [4] Alfred Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910.
- [5] Michael Oren, Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, and Tomaso Poggio. Pedestrian detection using wavelet templates. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 193–199. IEEE, 1997.
- [6] Constantine P Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *Computer vision, 1998. sixth international conference on*, pages 555–562. IEEE, 1998.
- [7] ITUT Rec. T. 800— iso/iec 15444-1,“. *Information technology—JPEG*, 2000.
- [8] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 1995.
- [9] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.
- [10] Lei Zhu and Kin-Hong Wong. Human tracking and counting using the kinect range sensor based on adaboost and kalman filter. *Advances in Visual Computing*, pages 582–591, 2013.