

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

CORRELATORE:  
Prof.ssa Susanna Spinsante  
Ing. Samuele Gasparrini (?)

TESI DI LAUREA DI:  
Ilario Pierbattista

---

ANNO ACCADEMICO 2014/2015

# Indice

<b>Introduzione</b>	<b>2</b>
<b>1 Panoramica del sistema</b>	<b>3</b>
1.1 Il Kinect	3
1.2 Head and Shoulder Profile	4
1.3 Un Problema di Classificazione	5
1.4 Preprocessing	6
<b>2 Haar-like Features</b>	<b>7</b>
2.1 Definizione	7
2.2 Dalle Wavelet di Haar alle Feature	8
2.3 Immagine Integrale	10
2.4 Decision Stump	11
<b>3 Adaboost</b>	<b>13</b>
3.0.1 Il dataset dall'allenamento	13
3.0.2 Preprocessing	13
3.0.3 Estrazione del classificatore forte	14
3.0.4 Il miglior classificatore debole	16
3.1 Costruzione dei Dataset	19
3.2 Test e valutazione	19
3.3 Rilevazioni su registrazioni reali	19
3.4 Conclusioni	19

# Introduzione

Il monitoraggio delle persone è un problema di visione artificiale di importanza fondamentale. Sistemi di *human sensing* vengono continuamente sviluppati ed utilizzati nei più disparati contesti applicativi. Sistemi di sorveglianza, apparecchiatura di supporto  
5 per missioni di salvataggio, dispositivi designati all'utilizzo in ambienti assistivi automatizzati e persino alcuni sistemi automatici per effettuare indagini di mercato utilizzano tecniche di percezione automatica delle persone dall'elaborazione dei dati acquisiti per mezzo di sensori.

Nel 2013 Zhu e Wong descrivono in [10] un sistema allenabile di rilevamento e conteg-  
10 gio delle persone che attraversano una stanza. Il riconoscimento della persona avviene elaborando i dati catturati dal sensore di profondità di un dispositivo Kinect, il quale è montato sul soffitto della stanza ed è puntato verso il pavimento. La crescente popolarità dei dispositivi Kinect, anche al di fuori degli ambienti videoludici, lo rende un interessante oggetto di studio.

15 Il presente elaborato fornisce una dettagliata analisi riguardante l'implementazione e le caratteristiche di tale sistema di rilevamento. Nel capitolo 1 si parlerà di tizio, Nel capitolo due si parlerà di caio.

# Capitolo 1

## Panoramica del sistema

20 Riproducendo la configurazione proposta in [10] al soffitto del laboratorio è stato fissato un dispositivo Kinect V2 rivolto verso il pavimento: all'ingresso di una persona nella visuale del Kinect, ne saranno ben visibili la testa e le spalle.

### 1.1 Il Kinect

Ad onor del vero, ciò che viene comunemente chiamato *sensore di profondità* (o sensore di distanza) del Kinect, è in realtà uno *scanner 3D a luce strutturata*<sup>1</sup>. Un sorgente di raggi infrarossi proietta una serie di pattern codificati. La deformazione indotta dalle superfici degli oggetti interessati viene acquisita da una o più telecamere ed utilizzata per il calcolo delle coordinate tridimensionali.

Il risultato di un sensore di questo  
30 tipo è un insieme di triplette  $(x, y, z)$ , organizzate in una *immagine di profondità*, una struttura dati che è molto simile ad una semplice immagine in scala dei grigi, dove il valore di  
35 ogni pixel rappresenta la misura in millimetri della distanza della superficie dal sensore. La forte somiglianza con le immagini in scala dei grigi è supportata dal fatto che ogni pixel  
40 è codificato utilizzando 16bit.

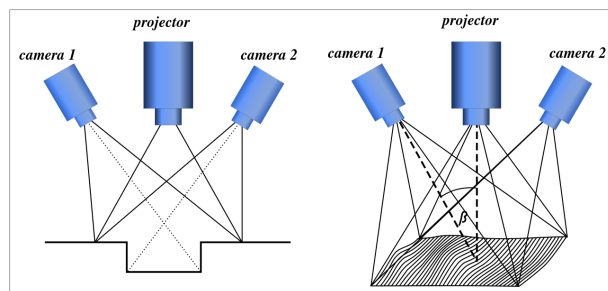


Figura 1.1: Schematizzazione di uno scanner 3D a luce strutturata.

La massima affidabilità del sensore del Kinect si ha per distanza comprese tra 50cm e 4,5m. Il dispositivo è montato al soffitto a 2,8m da terra e ha un campo visivo

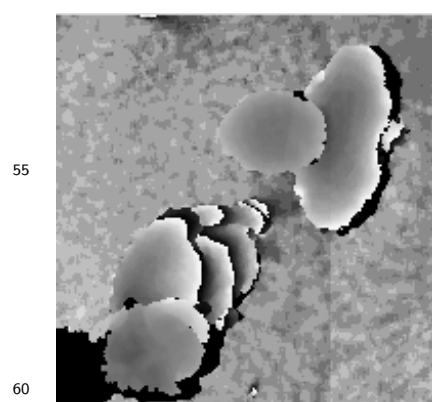
---

<sup>1</sup>Nonostante ciò, lo si continuerà a chiamare *sensore di profondità*, leggermente inesatto, ma decisamente più breve ed intuitivo

di  $70^\circ \times 60^\circ$ , il quale, all'altezza del pavimento, determina un'area di cattura di circa  $4m \times 5m$ .

45 La dimensione di ogni immagine di profondità è di  $512 \times 424$  pixel. Nativamente non vengono codificate in alcun modo particolare, sono delle semplici matrici di interi. Il Kinect V2 è in grado di catturarne fino a 30 al secondo. Utilizzando un apposito software di registrazione è stato possibile mettere insieme dei video di profondità a 30 fps.

## 50 1.2 Head and Shoulder Profile



55  
60  
Figura 1.2: Due persone in un ritaglio proveniente da un frame di profondità. Il frame è stato acquisito con il Kinect V1, a differenza di quello in figura 1.3, acquisito con il Kinect V2.

In condizioni ottimali, la figura umana ripresa dall'alto è composta solamente dalla testa e dalle spalle. Finché si trova quasi in corrispondenza del sensore, nella parte centrale dell'immagine di profondità, la restante parte del corpo rimane quasi del tutto nascosta. Sulla posizione delle braccia non si possono fare delle assunzioni precise.

In figura 1.2, la figura dell'individuo a destra corrisponde a tale descrizione, mentre dell'altro è visibile parte del corpo ed una delle due spalle è nascosta dalla testa. Nelle zone periferiche di un frame di profondità, l'immagine è soggetta alla *distorsione prospettica* così come lo è quella di qualunque telecamere RGB.

Tale distorsione costituisce un disturbo, dal momento che la figura dello stesso soggetto varia a seconda della relativa posizione nell'area di visualizzazione. Si vedrà

più avanti come affrontare tale situazione, per il momento si considerano solamente le immagini provenienti dalle zone centrali dei frame (come quella in figura 1.3).

Un grande vantaggio dell'utilizzo del Kinect in questa configurazione è l'*assenza di*  
70 *occlusione*. Infatti, rispetto ai molteplici sistemi di riconoscimento frontali, i soggetti non possono nascondersi l'uno con l'altro al sensore.

Utilizzare le immagini di profondità significa ragionare con le distanze: piuttosto che cercare di descrivere l'immagine del profilo umano in termini di forma, deve essere descritto in termini di differenze di quota rispetto all'ambiente circostante. Alla luce di  
75 ciò, si possono identificare i seguenti criteri descrittivi:

1. L'immagine di una persona è caratterizzata da uno spazio vuoto di fronte ad essa e dietro di essa. Per *spazio vuoto* si intende una regione la cui distanza dal sensore è circa quella del pavimento.

2. A sinistra della spalla sinistra ed a destra della spalla destra del profilo dall'alto di  
80 una persona, sono presenti degli spazi vuoti.
3. Tra la testa e le spalle vi è una differenza di altezza.

Di qui in avanti, con l'acronimo *HASP* (*Head And Shoulder Profile*), ci si riferirà proprio al profilo della persona ripreso dall'alto che soddisfa i criteri appena elencati.

### 85 1.3 Un Problema di Classificazione

Quello che il sistema di rilevamento deve fare è riconoscere se e dove è presente l'immagine *HASP* di una persona all'interno del frame di profondità. Ciò avviene analizzando sequenzialmente delle porzioni del frame originale in modo da  
90 coprire tutta l'area. Si tornerà più avanti su questo aspetto, per il momento si consideri che il problema di rilevamento è naturalmente riconducibile ad un problema di classificazione.

Gli oggetti da classificare sono immagini di profondità e le possibili classi di appartenenza di tali oggetti sono due: la classe delle immagini che ritraggono il profilo di  
95 una persona è la classe delle immagini che invece ne sono prive. Il fatto che vi siano solamente due classi lo rende un *problema di classificazione binario*.

L'operazione di classificazione avviene mediante la misurazione di alcune *caratteristiche* (*feature*) interessanti dell'immagine di profondità. I criteri di riconoscimento presentati nella sezione 1.2 sono dei validi esempi di caratteristiche. Sono espresse in  
100 linguaggio naturale e sono comprensibili agli esseri umani, ma mancano di rigore e perciò non sono direttamente implementabili come criteri di classificazione delle immagini *HASP*.

Il *classificatore* è il componente che esegue effettivamente la classificazione. La definizione di un classificatore può avvenire *manualmente*, mediante l'applicazione di un  
105 modello atto a descrivere al meglio gli oggetti di una determinata classe, oppure può essere *allenato*. Gli algoritmi per la definizione di classificatori allenati si distinguono a loro volta tra algoritmi di *allenamento supervisionato* e di *allenamento non supervisionato*, a seconda che facciano uso o meno di un insieme di allenamento.

Un insieme di allenamento non è altro che un insieme di oggetti adeguati al problema  
110 di classificazione per cui viene fornita la classificazione *reale*. In questo caso saranno delle immagini di profondità opportunamente marcate come immagini che contengono o meno il profilo di una persona.

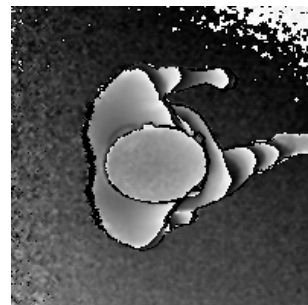


Figura 1.3: Una persona mentre cammina

Il sistema in esame prevede l'utilizzo di classificatori allenati con *Adaboost*. Sono molti i sistemi di riconoscimento di immagini che utilizzano Adaboost per allenare i rispettivi classificatori, primo tra tutti il quello di riconoscimento facciale proposto da Viola e Jones [9], ad oggi considerato il più robusto ed efficiente della sua categoria.

La forte somiglianza con il lavoro di Viola e Jones farà sì che saranno molteplici i riferimenti ad esso nel testo e i dettagli implementativi in comune.

## 1.4 Preprocessing

Ogni pixel di un'immagine di profondità denota la distanza in millimetri della superficie dell'oggetto dal sensore. Tuttavia, prima di poter essere utilizzate da qualsiasi componente del sistema di rilevamento, occorre convertire il valore di ogni pixel in quota della superficie dal pavimento.

Il pavimento è la superficie la cui distanza dal sensore è massima, quindi per individuare il valore sarebbe sufficiente ricercare il valore massimo di ciascuna immagine di profondità. Nella pratica ciò non è vero, in quanto, a causa della distorsione prospettica, nelle aree periferiche dell'immagine la distanza percepita del pavimento è maggiore rispetto a quella misurata al centro dell'immagine.

Quindi, per determinare la reale distanza del pavimento è necessario disporre di un'immagine di riferimento che ritragga solamente il pavimento stesso. A questo punto la distanza di riferimento è quella misurata esattamente al centro del frame.

La trasformazione delle distanze in quote avviene, come è intuibile, sostituendo il valore di ogni pixel  $d$  con il valore  $d'$ .

$$d' = d - d_{\text{pavimento}} \quad (1.1)$$

L'operazione di conversione ha complessità computazionale  $\Theta(n \cdot m)$ , dove  $n$  ed  $m$  sono le dimensioni del frame.

## Capitolo 2

# Haar-like Features

La descrizione di un qualsiasi oggetto avviene attraverso la descrizione dei suoi attributi. La forma geometrica, le dimensioni, il peso, il colore sono solo una manciata dei possibili attributi con i quali descrivere un oggetto. L'essere umano è dotato di 5 sensi attraverso i quali è in grado di *percepire* la realtà circostante e di un'infinità di modi con cui *descriverla*.

Il riconoscimento di un oggetto avviene solo in un secondo momento, attraverso l'analisi della sua descrizione, valutando gli attributi che, in quel contesto, sono più significativi. Nell'ambito della visione artificiale, il riconoscimento degli oggetti segue esattamente lo stesso flusso. Gli attributi di un oggetto vengono descritti mediante l'utilizzo di *feature*, ovvero dei meccanismi per misurare le varie proprietà dell'oggetto stesso.

Le immagini di profondità sono il frutto della percezione della realtà, le feature di Haar costituiscono gli attributi con i quali è possibile descrivere tali immagini.

### 2.1 Definizione

Le feature di Haar riescono a misurare la quantità e il verso delle variazioni di intensità tra due regioni adiacenti di una immagine. La comune rappresentazione grafica di tali feature mette in evidenza le due regioni colorandone una bianco e l'altra di nero. La somma delle intensità dei pixel della regione bianca a cui viene sottratta la somma delle intensità della regione nera fornisce una misura della variazione media di intensità tra le due regioni (equazione 2.1).

Di feature ne esistono di diverse forme: in figura 2.1 sono presenti quelle utilizzate in questo ambito, ma disponibili molte forme di feature utilizzate da altri sistemi di object detection. Non tutte sono formate solamente da *due regioni adiacenti*, ma l'importante è che tutte abbiano una regione *nera* ed una regione *bianca*.



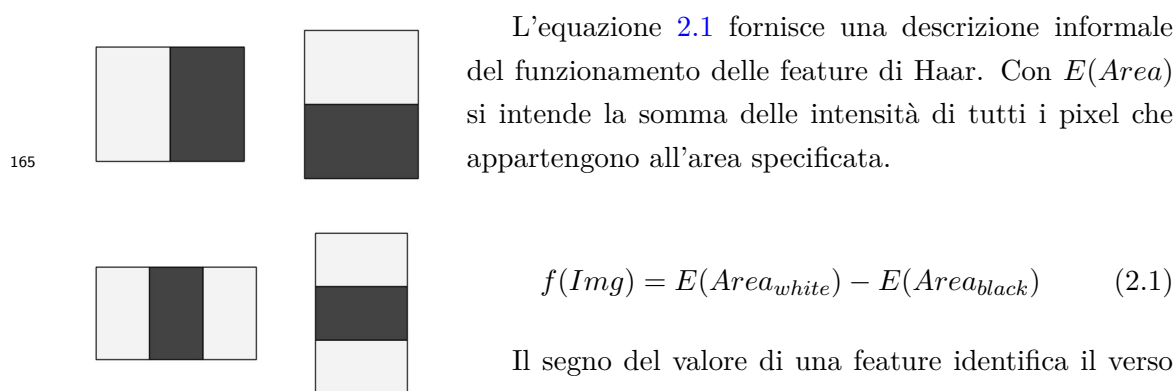


Figura 2.1: Tipi di feature di Haar utilizzati in [10].

170 bianca rispetto alla media della regione nera e viceversa.

Evidenziare le variazioni di intensità delle immagini integrali attraverso l'applicazione delle feature di Haar equivale ad evidenziare le differenze di quota medie delle regioni individuate dalla feature e questo le rende particolarmente adatte a questa applicazione.

In seguito sarà necessario dover calcolare gruppi di feature in diverse scale. Ridimensionare un gruppo di feature è una cosa semplice e verrà trattata in seguito, ma affinché il valore della feature sia il meno possibile sensibile ai ridimensionamenti è necessario rapportare tale valore all'estensione totale dell'area interessata (equazione 2.2).

$$f'(Img) = \frac{E(Area_{white}) - E(Area_{black})}{size(Area_{white}) + size(Area_{black})} \quad (2.2)$$

## 2.2 Dalle Wavelet di Haar alle Feature

180 Le feature di Haar derivano dall'estensione bidimensionale delle wavelet di Haar, che sono il primo esempio di wavelet e vennero sviluppate nel 1909 da Alfrèd Haar [4]. La wavelet madre (equazione 2.3) è una funzione oscillante di lunghezza finita, caratteristica comune a tutti i tipi di wavelet.

$$\psi(x) = \begin{cases} 1 & 0 \leq t < 1/2 \\ -1 & 1/2 \leq t < 1 \\ 0 & \text{altrimenti} \end{cases} \quad (2.3)$$

Ogni segnale può essere rappresentato per mezzo delle wavelet di Haar. Costituiscono un sistema di rappresentazione duale alla serie di Fourier.

185 Alfrèd Haar propose anche la prima DWT (*Discrete Wavelet Transform*) con la quale, le wavelet che compongono un segnale, vengono campionate discretamente. Le applicazioni sono notevoli e ad ampio spettro, prime tra tutte quelle nell'ambito della codica

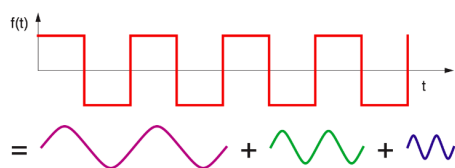


Figura 2.2: Rappresentazione di un segnale con una serie di funzioni armoniche.

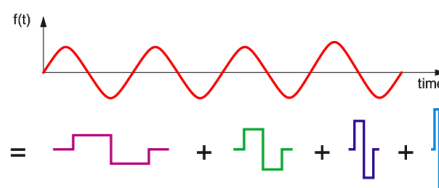


Figura 2.3: Rappresentazione di un segnale con una serie di wavelet di Haar.

dei segnali e nella compressione dei dati. Per fare un esempio, lo standard di compressione JPEG2000 sfrutta la trasformazione DWT [7] per ottenere risultati qualitativamente  
 190 migliori rispetto allo standard precedente.

Una variante della trasformazione DWT con le wavelet di Haar bidimensionali (figura 2.2) è stata utilizzata da Oren et al. [5] in un sistema in grado di riconoscere la presenza di pedoni in delle immagini.

Applicando alle immagini la trasformata DWT in diverse scale,  
 195 si passa da una rappresentazione dell'immagine in scala dei grigi, ad una rappresentazione in termini di coefficienti delle wavelet di Haar. Tali coefficienti denotano le differenze di intensità tra le aree adiacenti dell'immagine e vengono utilizzati per evidenziare analogie strutturali delle immagini che contengono la figura di un pedone.

Nella descrizione di un framework generale per l'object detection, Papageorgiou et al. [6] utilizzano la trasformata DWT a wavelet di Haar bidimensionali non per l'estrazione di un template dell'oggetto espresso in variazioni di intensità, bensì per la selezione delle wavelet più significative al fine del riconoscimento dell'oggetto.

Una wavelet di Haar bidimensionale di una data dimensione, utilizzata per campionare un'immagine in una posizione specifica mette in evidenza le differenze di intensità dell'immagine nella regione descritta dall'area della wavelet. Tale differenza di intensità costituisce una proprietà osservabile e misurabile dell'immagine che ritrae l'oggetto. Queste sono le *Haar-like feature* (o semplicemente feature di Haar).  
 205  
 210



Figura 2.4: Wavelet di Haar bidimensionali utilizzate in [5].

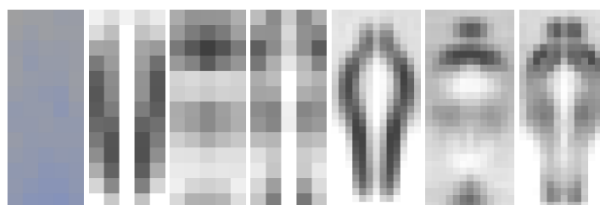


Figura 2.5: Coefficienti delle wavelet di trasformate DWT di diverse scale applicate alla stessa immagine. I coefficienti vengono codificati utilizzando la scala dei grigi [5, Figura 3].

## 2.3 Immagine Integrale

Le feature di Haar vengono utilizzate anche nel riconoscimento dei volti umani di Viola-Jones. Sono molto apprezzate, sebbene siano molto primitive rispetto ad altri tipi più evoluti di feature, per la loro efficienza computazionale.

Il calcolo della somma delle intensità dei pixel di un'area è un'operazione il cui costo aumenta linearmente con la quantità di pixel. Introducendo il concetto di *immagine integrale*, ottenuta rielaborando l'immagine di partenza, tale somma viene eseguita una volta per ogni immagine, permettendo in seguito di calcolare il valore di ogni feature in un tempo costante.

**Definizione 1.** Sia  $I$  un'immagine larga  $w$  pixel ed alta  $h$  pixel. Con la scrittura  $I(x, y)$  si identifica il pixel dell'immagine  $I$  che si trova alla colonna  $x$  e alla riga  $y$ . Si definisce *immagine integrale* (altrimenti detta *Summed Area Table*) una seconda immagine  $SAT$  delle stesse dimensioni per cui vale  $\forall x \in [1, w], y \in [1, h]$ :

$$SAT(x, y) = \sum_{i=1}^x \sum_{j=1}^y I(i, j) \quad (2.4)$$

Calcolare la somma del valore dei pixel contenuti in una regione rettangolare dell'immagine originale, con l'ausilio dell'immagine integrale, è un'operazione velocissima.

Si consideri la figura 2.6: si vuole calcolare la somma del valore dei pixel nella regione  $S$  evidenziata. I vertici del rettangolo saranno i punti  $A : (x_1, y_1)$ ,  $B : (x_2, y_1)$ ,  $C : (x_2, y_2)$ ,  $D : (x_1, y_2)$ , notando che  $1 \leq x_1 \leq x_2 \leq w$  e che  $1 \leq y_1 \leq y_2 \leq h$ <sup>1</sup>. Quindi:

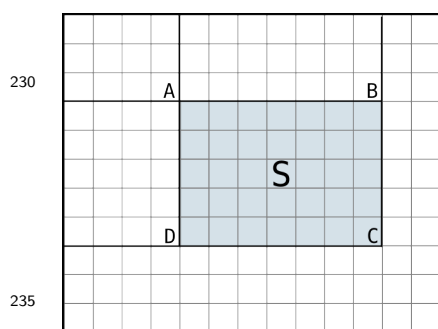


Figura 2.6

$$\begin{aligned} \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} I(i, j) &= \sum_{i=1}^{x_2} \sum_{j=y_1}^{y_2} I(i, j) - \sum_{i=1}^{x_1} \sum_{j=y_1}^{y_2} I(i, j) = \\ &= \left( \sum_{i=1}^{x_2} \sum_{j=1}^{y_2} I(i, j) - \sum_{i=1}^{x_2} \sum_{j=1}^{y_1} I(i, j) \right) - \left( \sum_{i=1}^{x_1} \sum_{j=1}^{y_2} I(i, j) - \sum_{i=1}^{x_1} \sum_{j=1}^{y_1} I(i, j) \right) = \\ &= SAT(C) - SAT(B) - SAT(D) + SAT(A) \end{aligned}$$

Elaborare l'immagine integrale è un'operazione con complessità computazionale  $\Theta(w \cdot h)$ , cioè il cui costo varia linearmente con la dimensione dell'immagine. Una volta ot-

<sup>1</sup>Nelle immagini raster, si iniziano a contare le colonne e le righe dal punto in alto a sinistra.

tenuta però, permette di calcolare qualsiasi somma di pixel in regioni rettangolari con operazioni di complessità  $\Theta(1)$ .

Grazie all'immagine integrale è quindi possibile valutare in tempi brevi, gruppi enormi di feature sulla stessa immagine.

## 2.4 Decision Stump

Si tratta di *alberi decisionali* binari di profondità unitaria. Vi è un unico test alla radice, in base al quale si decide l'appartenenza di un oggetto ad due classi. Combinando tra loro diversi decision stump, è possibile creare alberi decisionali più complessi da poter utilizzare in alberi di classificazione non binari.

I decision stump costituiscono il meccanismo più semplice di classificazione che si può avere a disposizione. Il test da collocare alla radice dell'albero viene eseguito sul valore di una feature di Haar.

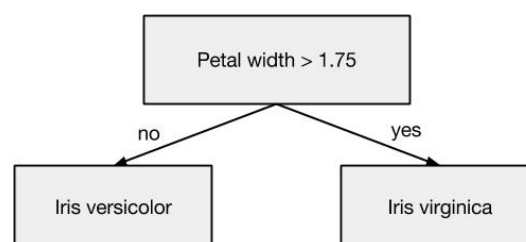


Figura 2.7: Esempio di decision stump usato per dividere gli oggetti in due classi delle tre descritte nel dataset delle caratteristiche dei fiori di Iris [2]

$$test_1(x) = \begin{cases} true & \text{se } f_1(x) < \theta_1 \\ false & \text{altrimenti} \end{cases} \quad (2.5)$$

$$test_2(x) = \begin{cases} true & \text{se } f_2(x) > \theta_2 \\ false & \text{altrimenti} \end{cases} \quad (2.6)$$

Le equazioni 2.5 e 2.6 descrivono due possibili test decisionali che possono essere collocati alla radice di un decision stump. Per standardizzare la forma dei test si utilizzerà l'equazione 2.7.

$$test(x) = \begin{cases} true & \text{se } f(x)p > p\theta \\ false & \text{altrimenti} \end{cases} \quad (2.7)$$

In riferimento all'equazione 2.7,  $x$  è un'immagine di profondità,  $f$  una feature di Haar ed  $f(x)$  il valore della feature misurato sull'immagine  $x$ . Il parametro  $p \in \{1, -1\}$ , che assumerà il nome di *polarità*, serve solamente per invertire il segno della disequazione all'occorrenza. Il parametro  $\theta$  è il valore di *soglia* (*threshold*).

Utilizzare i decision stump con le immagini di profondità equivale a verificare la presenza di variazioni di quota tra regioni contigue dell'immagine al di sopra o al di

265 sotto di un certo valore di soglia. La scelta del valore di soglia non è un problema banale  
e sarà trattato approfonditamente nel prossimo capitolo.

Il successo o l'insuccesso del test deve essere interpretato correttamente ai fini della  
classificazione. Di qui in avanti qualsiasi test decisionale applicato ad immagini di pro-  
fondità che avrà successo, segnerà la presenza di una persona, mentre, viceversa, quelli  
270 che non avranno successo ne segneranno l'assenza.

## Capitolo 3

# Adaboost

Adaboost è un *meta algoritmo* di *machine learning* che viene utilizzato per aumentare le prestazioni di altri algoritmi di apprendimento. Ciò avviene estraendo un *classificatore* 275 *forte* a partire da un insieme di *classificatori deboli*, dove per classificatore si intende una funzione  $f(x)$  che identifichi la *classe* di appartenenza dell'oggetto  $x$  in input.

La realtà d'interesse del problema prevede l'esistenza di due classi di oggetti: *umano* e *non umano*.

I dati in input per la fase di allenamento sono costituiti dal *dataset di allenamento*, 280 ovvero una raccolta di oggetti preclassificati attraverso i quali Adaboost riuscirà a selezionare una combinazione di classificatori deboli che meglio approssimano la classificazione degli elementi del dataset.

### 3.0.1 Il dataset dall'allenamento

La finestra di visualizzazione, come accennato in precedenza è di  $512 \times 424$  pixel. In una 285 registrazione che riprende dall'alto una persona che attraversa la stanza camminando, la figura della persona occupa una porzione della finestra di circa  $160 \times 100$  pixel.

Un dataset di allenamento non è altro che un insieme di porzioni di frame di una certa dimensione (la stessa per tutti) che vegono classificati manualmente dal creatore<sup>1</sup> del dataset.

### 290 3.0.2 Preprocessing

Gli elementi del dataset verranno preprocessati prima di poter essere dei dati di input adeguati per l'algoritmo.

---

<sup>1</sup>Il sistema di apprendimento che si sta trattando ricade nella categoria dei sistemi di *apprendimento supervisionato*.

Da ricordare che stiamo trattando di matrici  $160 \times 100$  (porzione del frame principale) i cui valori corrispondono al risultato della misurazione del sensore di profondità del Kinect ed è proporzionale alla distanza rilevata.

### Conversione delle distanze

La prima operazione di preprocessing consiste nel trasformare tali distanze (sensore-superficie) in distanze dal pavimento della stanza. Come è facilmente intuibile

$$d' = d_{\text{pavimento}} - d$$

dove la distanza del pavimento dal sensore ( $d_{\text{pavimento}}$ ). La distanza del pavimento può essere estratta dall'immagine stessa

Questa prima operazione di preprocessing ha complessità computazionale  $\Theta(n \cdot m)$  con  $n$  ed  $m$  dimensioni della porzione del frame.

### 3.0.3 Estrazione del classificatore forte

Sia  $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$  un insieme di  $n$  coppie costituite da un'immagine ( $x_i$ ) e la relativa classificazione ( $y_i \in \{0, 1\}$ ). Se  $y_i = 1$ , allora  $x_i$  appartiene alla classe *umano* (la coppia  $(x_i, y_i)$  prende il nome di *esempio positivo*), altrimenti appartiene alla classe *non umano* (la coppia  $(x_i, y_i)$  prende il nome di *esempio negativo*).

L'insieme  $D$  può essere partizionato come segue:

$$P = \{(x, y) \in D | y = 1\} \text{ e } N = \{(x, y) \in D | y = 0\}$$

Si tenga a mente che, essendo  $P$  ed  $N$  partizioni di  $D$ , valgono le seguenti<sup>2</sup>:

$$D = P \cup N \tag{3.1}$$

$$P \cap N = \emptyset \tag{3.2}$$

$$\#(D) = \#(P \cup N) = \#(P) + \#(N) \tag{3.3}$$

Si introduce anche il concetto di *classificatore debole*. Si tratta di una funzione che per una data immagine  $x$  in ingresso, assume il valore che simboleggia la presunta classe di appartenenza di quest'ultima. Nel dettaglio:

$$h(x) = \begin{cases} 1 & \text{se } pf(x) < p\theta \\ 0 & \text{altrimenti} \end{cases} \tag{3.4}$$

---

<sup>2</sup>La scrittura  $\#(D)$  denota il numero di elementi dell'insieme  $D$ .

dove  $f(x)$  è il valore di una feature di Haar applicata all'immagine  $x$ ,  $p \in \{-1, 1\}$  è detta *polarità* e  $\theta$  è la *soglia* (*threshold*). Tutti i classificatori deboli sono costruiti usando un'unica feature.

315 L'obiettivo di Adaboost è quello di formare un *classificatore forte* come combinazione lineare dei migliori classificatori deboli estraibili dal set di allenamento, dove il fattore moltiplicativo di ogni classificatore nella combinazione è inversamente proporzionale agli errori di classificazione compiuti da quest'ultimo in fase di allenamento.

320 Il seguente algoritmo descrive la procedura di estrazione e combinazione di  $T$  classificatori deboli.

1. Si associa ad ogni elemento  $(x_i, y_i) \in D$  un peso  $w_i$  tale che  $w_i = \frac{1}{2l}$  se  $(x_i, y_i) \in P$  oppure  $w_i = \frac{1}{2m}$  se  $(x_i, y_i) \in N$ , dove  $l = \#(P)$  ed  $m = \#(N)$  (numero degli esempi positivi e numero degli esempi negativi).

Sia inoltre  $n := \#(D) = \#(P) + \#(N) = l + m$ .

- 325 2. For  $t = [1 : T]$

- (a) Si normalizzano i pesi, in modo che la loro somma sia pari ad 1:

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

- (b) Si estrae il miglior classificatore debole. La procedura viene esposta nel dettaglio nella sezione 3.0.4, ma si tenga presente che il miglior classificatore è quello il cui *errore pesato* è minimo per la corrente iterazione.

$$\epsilon_t = \min_{f,p,\theta} \left\{ \sum_{i=1}^n w_{t,i} \cdot |h(x_i, f, p, \theta) - y_i| \right\}$$

Siano inoltre  $f_t, p_t, \theta_t$  i parametri del classificatore debole che ne minimizzano l'errore pesato:

$$h_t(x) := h(x, f_t, p_t, \theta_t)$$

- (c)  $\beta_t \leftarrow \frac{\epsilon_t}{1-\epsilon_t}$

- (d) Si aggiornano i pesi

$$w_{t+1,i} \leftarrow w_{t,i} \cdot \beta_t^{e_i}$$

dove  $e_i = 1$  se  $(x_i, h_t(x_i)) \in D$  (ovvero se  $x_i$  è classificata correttamente),  $e_i = 0$  altrimenti.

- (e)  $\alpha_t \leftarrow \log\left(\frac{1}{\beta_t}\right)$



330 3. Il classificatore forte è dato da:

$$F(x) = \begin{cases} 1 & \text{se } \sum_{t=1}^T \alpha_t h_t(x) > \theta \sum_{t=1}^T \alpha_t \\ 0 & \text{altrimenti} \end{cases} \quad (3.5)$$

dove  $\theta \in [0, 1]$  è la soglia.

Si noti che, nell'operazione 2b, l'errore pesato non è altro che la somma dei pesi degli esempi non classificati correttamente. Infatti:

$$h(x_i, f, p, \theta) = y_i \Rightarrow |h(x_i, f, p, \theta) - y_i| = 0$$

$$h(x_i, f, p, \theta) \neq y_i \Rightarrow |h(x_i, f, p, \theta) - y_i| = 1$$

Al punto 2c, il valore di  $\beta_t$  non è altro che il rapporto tra l'errore pesato del classificatore debole e la somma dei pesi delle immagini classificate correttamente. Tale valore è chiaramente  $0 < \beta_t < 1$ .

335 In fase di aggiornamento dei pesi (punto 2d), i pesi relativi ad esempi classificati correttamente vengono moltiplicati per  $\beta_t$  ( $\beta_t^1 = \beta_t < 1$ ) e quindi decrementati, mentre gli altri vengono lasciati inalterati ( $\beta_t^0 = 1$ ). Fare in modo che gli esempi non classificati correttamente abbiano un peso maggiore di quelli classificati correttamente è il modo per influenzare la scelta del classificatore debole successivo che andrà a colmare le lacune  
340 del suo predecessore.

La scelta della soglia per il classificatore forte (punto 3) deve minimizzare il numero di esempi classificati in modo errato.

### 3.0.4 Il miglior classificatore debole

Si è detto che un classificatore debole è costruito a partire da una feature di Haar. La  
345 scelta del migliore, quindi, mira ad identificare la feature, la polarità e la soglia che minimizzano l'errore pesato di classificazione.

Si ricordi che le feature di Haar sono degli indicatori di quanto le intensità dei pixel variano da una regione della feature ad un'altra. Il classificatore debole, quindi, classificherà l'immagine a seconda che tale indice sia maggiore o minore di una certa soglia. Il  
350 compito della polarità è quello di stabilire il verso della disuguaglianza.

Il pool di feature da testare è costituito - teoricamente - da tutte quelle individuabili nell'immagini di allenamento. Nell'opera di Viola e Jones vengono utilizzate immagini di allenamento di  $24 \times 24$  pixel e 5 tipologie di feature ([9], sezione 2.2): il numero di possibili features in tale area è maggiore di 160000. In questa applicazione si utilizzano  
355 immagini di  $160 \times 100$  pixel e 4 tipologie di feature: il pool è costituito da un numero di elementi maggiore di almeno 4 ordini di grandezza.

È da tener presente che moltissime di queste feature sono poco significative in questa situazione: non ha senso calcolare la variazione di intensità di due aree molto piccole con delle immagini che hanno una risoluzione tanto alta. Effettuando una prima scrematura,  
 360 si cercherà di avere un pool di feature selezionabili la cui dimensione non superi quella del pool di Viola-Jones per più di un ordine di grandezza.

Sia  $\{f_1, \dots, f_k\}$  l'insieme di tutte le feature selezionabili,  $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$  l'insieme degli esempi di allenamento e  $\{w_1, \dots, w_n\}$  l'insieme dei relativi pesi. La scelta del classificatore debole avviene come descritto dal seguente algoritmo in pseudocodice:

1. Si calcolano  $T^+$  e  $T^-$ , rispettivamente, somma dei pesi degli esempi negativi e di quelli negativi:

$$T^+ \leftarrow \sum_{i=1}^n (w_i y_i), \quad T^- \leftarrow \sum_{i=1}^n [w_i (1 - y_i)]$$

- 365 2. *For*  $f = [f_1, \dots, f_k]$

- (a) Si inizializza una lista di  $n$  elementi per memorizzare i valori della feature  $i$ -esima applicata ad ogni immagine di allenamento:

$$values[i] \leftarrow f(x_i) \quad \forall x_i \in D$$

- (b) Si ordinano gli elementi della lista in ordine crescente. Si tenga in conto che, dopo tale operazione, all' $i$ -esima posizione della lista non corrisponderà più il valore della feature applicata all' $i$ -esima immagine di allenamento.
- (c) Si inizializzano  $S^+$  ed  $S^-$ , con le quali, scorrendo gli elementi della lista con un cursore, indicheremo rispettivamente la somma dei pesi degli esempi positivi e di quelli negativi:  $S^+ \leftarrow 0, S^- \leftarrow 0$   
 370

- (d) *For*  $i = [1 : n]$

- i. A causa del rilocamento degli indici, alla posizione  $i$ -esima della lista corrisponderà il valore della feature dell'elemento  $x_j$  con classificazione  $y_j$  e peso  $w_j$  tale che  $(x_j, y_j) \in D$ :

$$x_j, y_j, w_j \leftarrow values[i]$$

- ii. *If*  $y_i = 1$  *Then*

$$A. \quad S^+ \leftarrow S^+ + w_j$$

- iii. *Else*  
 375

$$A. \quad S^+ \leftarrow S^+ + w_j$$

iv. Si calcola l'errore pesato di classificazione:

$$e_i = \min\{S^+ + (T^- - S^-), S^- + (T^+ - S^+)\}$$

(e) Si determinano la polarità ( $p_f$ ) e la soglia ( $\theta_f$ ) per cui l'errore pesato ( $\epsilon_f$ ) di classificazione per un classificatore che utilizza la feature  $f$  è minimo:

$$p_f, \theta_f | \epsilon_f = \min\{e_1, \dots, e_n\}$$

3. Si scelgono polarità ( $p$ ) e soglia ( $\theta$ ) finali, ovvero quelle del classificatore debole con errore pesato  $\epsilon$  minore tra tutti i classificatori possibili:

$$p, \theta | \epsilon = \min\{\epsilon_1, \dots, \epsilon_k\}$$

Il miglior classificatore debole è quindi:

$$h(x) := \begin{cases} 1 & \text{se } pf(x) < p\theta \\ 0 & \text{altrimenti} \end{cases} \quad (3.6)$$

Il metodo di identificazione della polarità e della soglia non viene riportato nell'algoritmo, essendo un passaggio che merita una trattazione a parte. Selezionare un valore di soglia vuol dire trovare il *punto che partiziona al meglio la lista dei valori della feature*   
 380 *calcolata sulle immagini di allenamento, in modo tale da minimizzare gli errori di classificazione.* La miglior soglia di una buona feature fa in modo che la maggior parte delle immagini appartenenti alla stessa classe assumano un valore minore (o maggiore). Si può notare, come diretta conseguenza, che con una feature pessima per la classificazione   
 385 non sarà possibile trovare un valore di soglia che soddisfi tale criterio.

Al punto 2(d)iv viene calcolato l'errore pesato di classificazione per una feature  $f$  con soglia  $values[i]$ <sup>3</sup>. Per essere più espliciti, bisogna effettuare una serie di osservazioni:

1.  $T^+$  ( $T^-$ ) corrisponde alla somma dei pesi degli esempi positivi (negativi);
2.  $S^+$  ( $S^-$ ) corrisponde alla somma dei pesi degli esempi positivi (negativi) dalla   
 390 prima posizione fino all' $i$ -esima della lista (quella su cui è posizionato il cursore);
3.  $T^+ - S^+$  ( $T^- - S^-$ ) corrisponde alla somma dei pesi degli esempi positivi (negativi) dalla posizione  $i + 1$  della lista fino alla fine;

---

<sup>3</sup>I possibili valori delle soglie corrispondono esattamente ai valori della feature calcolata sugli esempi di allenamento

- 395 4. Per un classificatore della forma (3.6) con  $p = 1$ ,  $S^+$  corrisponde alla *somma dei pesi degli esempi classificati correttamente*, mentre  $S^- + (T^+ - S^+)$  corrisponde alla somma dei pesi degli esempi classificati in modo scorretto;
5. Per l'osservazione 4 un classificatore (3.6) con  $p = 1$ , la quantità  $S^- + (T^+ - S^+)$  è *l'errore pesato*;
6. Analogamente alle osservazioni 4 e 5, un classificatore (3.6) con  $p = -1$  commetterà un errore pesato pari alla quantità  $S^+ + (T^- - S^-)$ .

400 Grazie alle osservazioni 5 e 6, dal semplice calcolo dell'errore di classificazione pesato, si ottiene anche il valore della polarità:

$$p = \begin{cases} 1 & \text{se } S^- + (T^+ - S^+) < S^+ + (T^- - S^-) \\ -1 & \text{altrimenti} \end{cases} \quad (3.7)$$

In definitiva, con uno scorrimento della lista ordinata si ottengono i parametri per la costruzione del classificatore debole. La complessità di tale operazione è fortemente legata all'algoritmo di ordinamento della lista, la quale ha  $\Theta(n \log n)$  come limite teorico inferiore [1, p. 167]. Nell'implementazione è stato scelto proprio un algoritmo che avesse  
405 complessità  $\Omega(n \log n)$  nel caso peggiore. Ripetendo queste operazioni per ognuna delle feature selezionabili, si ottiene che l'algoritmo di selezione del miglior classificatore debole ha complessità  $\Theta(kn \log n)$ .

### 3.1 Costruzione dei Dataset

### 410 3.2 Test e valutazione

### 3.3 Rilevazioni su registrazioni reali

### 3.4 Conclusioni

# Bibliografia

- [1] Thomas H Cormen. Introduction to algorithms. 2009.
- 415 [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.
- 420 [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*, pages 193–199. IEEE, 1997.
- 425 [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.
- 430 [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.
- 435