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

Capitolo 1

Introduzione

1.1 Human Sensing

1.1.1 Human Sensing

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 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.

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

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'obbiettivo di tale riproduzione non si limita alla semplice acquisizione di una rappresetazione bidimensionale di una regione di spazio, ma mira all'interpretazione del relativo 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.

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

Gli ambiti d'utilizzo del dispositivo Kinect, nota periferica legata a sistemi videoludici, si stanno espandendo constantemente. 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

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.

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

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 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 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 delle lenti correttive per aumentare il range di affidabilità del sensore.

Molto 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 può nascondere dietro di sè un'altra persona alla vista del sensore (se non in scomode posizioni), cosa frequentessima 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 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 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 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 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 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.

1. L'immagine di una persona è caratterizzata da uno *spazio vuoto*[1] 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

Definizione dei moduli funzionali

Un modulo software sarà dedicato all'allenamento.

Un modulo software sarà dedicato al rilevamento.

Allenamento

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, 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 successvi 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, 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 classificazione dell'oggetto stesso.

Rilevamento

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

Il classificatore ottenuto al termine dell'esecuzione di Adaboost, sarà in grado di 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 un 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, consterà

[1] 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.

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 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)

Cosa sono le caratteristiche di un oggetto

Le caratteristiche dipendono da cosa si vuole evidenziare

Le caratteristiche dipendono da cosa si ha a disposizione

2.1.2 Wavelet di Haar

Le feature di Haar derivano dalle wavelet di Haar

Definizione informale delle wavelet di Haar

Chi le ha sviluppate

Cosa sono (base ortonormale spazio funzionale)

Rappresentazione dei segnali (fourier duale)

Wavelet di Haar e DWT

Utilizzi commerciali DWT (JPEG2000)

Utilizzo delle dwt per il pattern recognition

2.1.3 Formula di calcolo standard

Rappresentazione visuale

Formula generale

Altri tipi di feature (OpenCv)

Tipi di feature utilizzate

2.1.4 Cosa mette in evidenza la feature di Haar

Immagini normali (Viola Jones)

Immagini di profondità (Zhu Wong)

2.1.5 Formula di calcolo invariante ai resize

Anticipazione del problema del ridimensionamento

Formula: Normalizzazione sull'area

2.1.6 Vantaggi

Differenze di intensità vs Valutazione dei singoli pixel

Differenze di intensità vs Estrazione dei contorni

Estrema efficienza computazionale

2.2 Immagine Integrale

2.2.1 Definizione rigorosa dell'immagine integrale

Problema: efficienza nel calcolo di somme di pixel

Complessità computazionale del calcolo *ignorante*

Soluzione: rendere queste somme subito disponibili

Definizione immagine integrale

Formula di calcolo della somma dei pixel in un'area

Enunciazione della formula

Dimostrazione della formula

Complessità computazionale del calcolo della feature

2.2.2 Complessità computazionale generale

Complessità del calcolo dell'immagine integrale

Convenienza del calcolo dell'immagine integrale

2.3 Decision Stump

2.3.1 Problema: utilizzare le feature

È necessario un meccanismo primitivo per utilizzare le feature

Bisogna discriminare in base al valore

2.3.2 Definizione di albero decisionale

2.3.3 Definizione di decision stump

Radice: Test, funzione booleana

Foglie: risultati possibili

Formule di calcolo binaria

Formula di calcolo unica: polarità

Capitolo 3

L'Algoritmo di Allenamento: Adaboost

3.1 Apprendimento Supervisionato *Ensamble*

3.1.1 Apprendimento Supervisionato

Definizione

Obiettivo

Spazio delle Ipotesi

Esempi di Supervised learning

Algoritmi di Sup.Learning

Maggiori campi applicativi

Concetti di base

Overfitting

Ensamble Learning

3.1.2 *Adaptive Boosting*

Algoritmi di Boosting

Aptive: adattabilità

Strong learner e Weak learner

3.2 Dataset di Allenamento

3.2.1 Categorie di Classificatori

Variabilità della forma HASP

Variazione dell'orientazione

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

Acquisizioni

Soggetti, percorsi

Acquisizione delle registrazioni

Ritaglio dei samples

Trainset Creator

3.2.3 Preprocessing

Resize

Nearest Neighbour

Altri algoritmi di resize

Conversione delle distanze

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

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

Soglia del classificatore

4.3.2 Algoritmo di ricerca della soglia e del NWL ottimi

4.4 Analisi dei Risultati

Capitolo 5

Rilevamento

5.1 Tecnica di Rilevamento

5.1.1 Detection Window

5.1.2 Rilevazione su frame

Resize detection window

Slide detection window

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

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

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.