Visione artificiale: ricerca e didattica in un istituto tecnico industriale

Angelo Monfroglio

ITIS Omar-Novara

Via Beldì 19, 28068 Romentino (NO) angelomonfroglio@mclink.it

Vengono descritte esperienze di uso di vari linguaggi di programmazione, grafici, specifici (simulatori di reti neurali artificiali), visuali e di impiego generale (C e Java) per applicazioni di visione artificiale (computer vision) in ambiente didattico e di ricerca, presso un istituto tecnico industriale.

1. Introduzione

La visione artificiale (Computer Vision) è la trasformazione di dati da una foto o video digitali in una rappresentazione o una decisione (ad esempio, muoversi in avanti, a destra, ecc.). La trasformazione è volta al raggiungimento di uno o più obiettivi. I dati in ingresso possono comprendere anche informazioni contestuali come la distanza del soggetto inquadrato, ottenute per mezzo di sensori, come localizzatori a LED, Laser, ultrasuoni, ecc. Un sistema elementare di visione (si veda [Palmisano, 2010]) è così costituito:

1.1 Cattura delle immagini con un sensore CCD o CMOS, come nelle foto e video camere digitali. Le immagini sono memorizzate in matrici a 2 dimensioni, con le coordinate X e Y, e i valori 0 e 1; oppure un valore per la scala di grigi; oppure un valore a colori (RGB). Un insieme di operatori chiamati filtri: ad esempio, la media sui valori dei pixel; oppure la soglia: 0 tutti i valori al di sotto della soglia, 1 sopra; cambiamento di contrasto e di luminosità

1.2 Algoritmo di rilevazione del contorno o perimetro (edge detection). E' il primo algoritmo fondamentale e, in alcuni semplici casi, l'unico necessario. Si estrae l'oggetto dal fondo, determinando lati, angoli, ecc. L'algoritmo è basato sul calcolo del gradiente (discreto) per ogni pixel lungo le direzioni vicine. Ad esempio: ordina i pixel nella matrice; per ogni pixel, analizza ognuno degli 8 pixel vicini; memorizza il valore più basso (scala dei grigi, o dei colori base RGB) e il più alto e il più basso; se (valore più alto – valore più basso) > soglia riscrive il valore del pixel come 1 altrimenti scrive 0.

1.3 Rilevazione di una forma e riconoscimento di una configurazione (Shape Detection e Pattern Recognition). Un caso comune è l'isolamento di una faccia e il riconoscimento di una identità. Si costruisce un data base di

forme: si esegue il rilevamento dei contorni (passo precedente); si calcola il numero di lati contigui: un cambiamento repentino nella direzione significa linea diversa; se si identificano tre lati allora è un triangolo, se 4 un quadrato, se la linea è continua un cerchio, ecc. Per figure più complesse sono necessari altri approcci: analisi probabilistica, Fuzzy Logic (logica sfumata), reti neurali artificiali (si veda un paragrafo successivo).

1.4 Centro di massa e rilevamento di blob (macchia di colore). L'algoritmo di Blob detection si usa per determinare se un gruppo di pixel sono tra loro correlati. Se c'è solo un blob, il centro di massa è facilmente determinabile. Se ce ne sono di più occorre etichettare ogni singolo blob.

1.5. Correlazione di immagini (Template Matching) e Riconoscimento facciale: si costruisce un data base di caratteristiche (features) e si calcola l'intensità di somiglianza. Un caso molto noto è il riconoscimento di una faccia

1.6. Rivelazione di movimento (Motion Detection): si tratta di un'applicazione fondamentale, tra l'altro, nei sistemi di sorveglianza: vedremo il caso del robot Surveyor. L'algoritmo opera con differenze fra frame successivi: ad esempio, valutando il movimento di un blob di colore. Si vedano anche [Jaehne e Haussecker, 2000], [Szeliski, 2011].

2. Kit educativi in commercio

2.1 POB Robot (Pob Technology) Si tratta di robot mobile su cingoli (4 ruote motrici) della ditta francese POB-TECHNOLOGY. E' interamente progettato attorno al sistema di visione (POB-Eye) e fornito con ambiente di sviluppo grafico proprietario Risbee; inoltre si può programmare in linguaggio C e Java. Vengono forniti programmi dimostrativi per riconoscimento di semplici forme (quadrato, triangolo, croce, ecc.), inseguimento di forme e pattern recognition. Il sistema è decisamente avanzato e ben documentato. Abbiamo sviluppato programmi in C che hanno funzionato molto bene.

2.2 Dagu Mr Tidy (Arduino) E' un robot mobile coreano, basato però sulla piattaforma robotica italiana open source Arduino dell'Olivetti. Il robot monta un semplice sistema in grado di discriminare colori RGB, ed è fornito con dimostrativo che vede il robot impilare bicchieri a seconda del colore. I bicchieri possono essere collocati ovunque senza uno schema fisso. Il software di sviluppo (free) è scaricabile dal sito di Arduino. Abbiamo sviluppato e ampliato il dimostrativo, sperimentandolo con pieno successo.

2.3 Surveyor E' una piattaforma open source fornita senza programmi di corredo, ma sul sito c'è un'abbondante quantità di software, fra cui un

embrione di console java che abbiamo scaricato, completato e provato con ottimi risultati. Il robot è dotato di una telecamera a colori con risoluzione fino a 1280 x 1024. Ha una radio per la comunicazione wireless verso PC. Inoltre c'è un sistema per misurare le distanze con due puntatori laser. Un distanziometro a laser può essere usato in due modi: misurando il ritardo di risposta, oppure operando una triangolazione. Surveyor può operare in telepresenza come robot di sorveglianza, o autonomamente, o infine come sciame di robot interagenti. Può essere gestito da un PC locale come se fosse una rete wireless non protetta, o in remoto con un access point locale, come se fosse un nodo con indirizzo IP, in Internet , in qualunque parte del mondo.

3. Ricerche all'ITIS "OMAR" di Novara

Il laboratorio di robotica industriale all'ITIS Omar di Novara si è costituito in forma sperimentale nel 1984 dopo una serie di lezioni del professor Marco Somalvico del Politecnico di Milano. Nel 1989 il nostro ex allievo Pier Luigi Poy, studente del Poli di Milano, sviluppò, con la nostra collaborazione, una tesi su un sistema in linguaggio C per la visione computerizzata di immagini Bit Map. Nel 2008 gli allievi di ingegneria Napolitano e Trombin svolsero il tirocinio nel nostro laboratorio, sviluppando un sistema in Visual Basic, costituito da un manipolatore a 5 gradi di libertà e una Web Cam collegata con un personal computer, sotto la guida dei professori Alessandro De Rose e Giuseppe Peretti,e del tecnico di laboratorio Sergio Conti. Il sistema era in grado di impilare pedine collocate in posizioni fisse su una tastiera, distinguendole per il colore: rosso, verde, giallo. Questo fu sufficiente per passare il tirocinio. Tuttavia, emersero alcuni problemi: in condizioni particolari di luce un colore non era riconosciuto correttamente. Questo ci spinse a una nuova ricerca per analizzare i colori non nello spazio additivo R G B (tipico delle Web Cam), ma in quello sottrattivo H S V (Hue, Saturation, Brighness) tipico dei TV. Si veda [Brdadski e Kaehler, 2008]. E' facile scrivere un programma per convertire il primo nel secondo formato. I risultati sono stati buoni.

3.1 Interfaccia grafica per la programmazione logica vincolata

Abbiamo proposto in [Monfroglio, 1987] un'interfaccia grafica per la programmazione logica. Il Prolog è stato un linguaggio promettente per la robotica. I suoi due meccanismi di Matching e Unificazione permettono di operare il ragionamento automatico sulla base di una programmazione dichiarativa. Abbiamo così pensato di dotarlo di un'interfaccia grafica intelligente per applicazioni come la visione artificiale. L'idea è di operare sui gruppi di trasformazioni geometriche: similarità, affinità, topologia, ecc. Tuttavia, la realizzazione pratica, anche se non abbandonata, si è rivelata ardua. Più

agevole è stato realizzare l'approccio con le reti neurali artificiali ([Monfroglio, 1998]).

3.2 Reti neurali artificiali per la visione

Abbiamo utilizzato un simulatore software di reti neurali, NeuralWorks Professional II/Plus che fornisce in uscita un listato in linguaggio C. Questo, opportunamente adattato e compilato, viene inserito nel sistema di visione. Le applicazioni che abbiamo realizzato sono il riconoscimento di lettere in stampatello, anche imperfette; il riconoscimento di figure;il riconoscimento di scritti a mano. I paradigmi che abbiamo sperimentato sono stati: Percettroni a 3 strati con Back-Propagation, Radial Basis Function Networks, Self Organizing Maps e Spatio Temporal Pattern Recognition Networks.

4. Contributi didattici e conclusioni

L'esperienza di utilizzo di diversi linguaggi di programmazione per la visione intelligente e la robotica si può così riassumere: in una fase iniziale i linguaggi grafici specifici e proprietari sono utili e facilitano l'apprendimento; tuttavia, riteniamo che non si possa prescindere dai linguaggi generali come il Visual Basic (da noi proposto in terza classe); il C (in quarta classe) e Java (in quinta); Java si è dimostrato il linguaggio più arduo, ma anche il più idoneo per applicazioni distribuite e su Internet. D'altra parte, la possibilità di gestire robot come Surveyor da un'aula all'altra, e addirittura da casa verso la scuola, per mezzo di programmi scritti in Java, si è rivelata molto motivante e gratificante per gli studenti. Molto importante è stato l'uso della libreria in linguaggio C e C++ Open CV, e dell'ambiente di sviluppo Microsoft Robotics Developer Studio. Il contributo didattico che possiamo fornire nasce dall'esperienza di oltre 25 anni nel settore della robotica industriale, e dal continuo contatto, oltre che con le altre scuole medie superiori, con i Politecnici di Milano e Torino.

Bibliografia

Brdadski G., A. Kaehler, Open CV, Computer vision with OpenCV Libray, O'Reilly, Sebastopol, CA, 2008

Jaehne B., H. Haussecker, Computer Vision and Applications, Academic Press, San Diego Ca, 2000

A. Monfroglio, Finite Contraint Satisfaction, in Neural Network Systems Techniques and Applications edited by Cornelius T. Leondes, Academic Press, San Diego Ca, 1998

A. Monfroglio, Graphical Interface for Logic Programming, ACM SIGART Bulletin, n. 101, July 1987

J. Palmisano, Computer Vision Tutorial, www.societyofrobots.com, 2010

R. Szeliski, Computer Vision, Algorithms and Applications, Springer, London, 2011