

HumanHeadDetector

Progetto di elaborazione delle immagini

a.a. 2008/2009

Andrea Mantoni

Contenuti

1. Logica del progetto.....	1
Stage 1: Preprocessing.....	1
Stage 2: Ellipses detection and selection.....	2
Riconoscimento degli ellissi.....	2
Selezione degli ellissi.....	3
2. Struttura dei sorgenti.....	3
3. Descrizione della GUI.....	4
4. Esempi di elaborazione.....	5

1. Logica del progetto

Il riconoscimento delle teste umane avviene in tre stages:

Stage 0: caricamento dell'immagine di input

Stage 1: preprocessing (mediante diversi operatori puntuali e locali)

Stage 2: ellipses detection and selection (mediante RHT e criteri geometrici)

In seguito vengono descritte nel dettaglio la varie fasi.

Stage 1: Preprocessing

1. **to RGB** - Se in un formato differente l'immagine in input viene convertita nello spazio di colore RGB a 32-bit. Questo è necessario per le successive elaborazioni.
2. **Blur Gaussiano** – Viene applicato per ridurre il rumore eventualmente presente nell'immagine.
3. **Filtro di segmentazione della pelle basato su colore** – Diversi studi hanno dimostrato che lo spazio di colore più efficace nel quale eseguire la segmentazione della pelle umana è HSV. Il filtraggio consiste nel preservare solo i colori nei seguenti ranges (modellati sul colore della pelle di razza caucasica):

$$0^\circ < \text{Hue angle} < 50^\circ$$

$$20\% < \text{Saturation} < 85\%$$

$$\text{Value} > 35\%$$

Tipicamente, poiché le teste sono disposte frontalmente rispetto al sensore d'acquisizione questa segmentazione migliora il riconoscimento. Tuttavia in diverse situazioni potrebbe essere necessario disattivarlo (nella versione interattiva del programma il filtro può essere

disattivato mediante un dialog). Ad esempio: se le teste sono disposte di profilo, se indossano una maschera o se hanno un colore della pelle diverso da quello della razza caucasica.

4. **Edge detection** – Eseguito mediante un kernel laplaciano.
5. **Binarizzazione** – Nella versione interattiva il threshold può essere impostato dall'utente.

Stage 2: Ellipses detection and selection

Riconoscimento degli ellissi

L'equazione parametrica dell'ellisse è la seguente:

$$ax^2 + 2bxy + cy^2 + dx + 2ey + f = 0$$

Un'implementazione “classica” della HT richiederebbe un accumulatore a cinque dimensioni e avrebbe una complessità di $O(n^5)$.

La Randomized Hough Transform utilizza la randomizzazione per ridurre lo spazio di ricerca. La complessità finale è di $O(\frac{n^2}{2})$ (“n” è il numero di edge points individuati nello stage precedente).

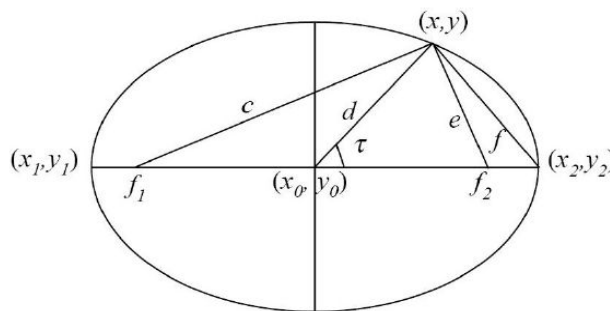
Il suo funzionamento è descritto in seguito.

All'inizio di ogni ciclo l'algoritmo seleziona casualmente una coppia di edge-points (x_1, y_1) (x_2, y_2) e li candida come vertici di un ellisse.

Quindi calcola i seguenti parametri:

- x_0, y_0 – le coordinate del centro dell'ellisse - il punto medio tra i vertici
- a – il semiasse maggiore – la semidistanza tra i vertici

Per calcolare gli altri parametri è necessario un terzo punto (x, y) . L'algoritmo a questo punto cicla tutti gli edge points rimasti e calcola:



- c, e – la distanza del punto (x, y) dai due fuochi
- d – la distanza del punto (x, y) dal centro – deve essere: $a < d < b$
- f – la distanza del punto (x, y) da un vertice
- b – il semiasse minore:

$$b = \sqrt{\frac{a^2 d^2 \sin^2(\tau)}{a^2 - d^2 \cos^2(\tau)}}$$

$$\cos(\tau) = \frac{a^2 + d^2 - f^2}{2ad}$$

Per ogni punto (x,y) analizzato, l'algoritmo effettua una votazione nell'accumulatore (monodimensionale) sul parametro b associato.

Dopo aver ciclato tutti gli edge points, ricerca il massimo nell'accumulatore per determinare b.

Infine inserisce l'ellisse riconosciuto con associato il numero di voti in un albero, per facilitare la seguente selezione.

L'accumulatore viene quindi riassetato e si seleziona una nuova coppia di vertici.

Ulteriori ottimizzazioni implementate:

- L'algoritmo prende in input un range per il semiasse minore. Poiché gli assi devono essere in proporzione tra loro (vedi paragrafo seguente) il range per il semiasse maggiore è calcolato automaticamente.
Se i valori dei semiassi calcolati (i parametri a e b) non rientrano nei range si scarta la coppia di vertici attuale o il punto attuale.
- Il numero minimo di voti che ciascun ellisse deve totalizzare per essere accettato è uguale alla somma delle lunghezze minime dei semiassi.
- E' stato aggiunto un contatore di cicli "idle" che ferma la ricerca se si eseguono un certo numero di passi senza trovare alcun ellisse. (Di default esso è impostato a 10000.)

Selezione degli ellissi

Una volta completato il riconoscimento degli ellissi, si applicano dei criteri geometrici di selezione.

Tutte le teste umane presentano delle proporzioni naturali: il rapporto dei due assi è uguale alla sezione aurea.

$$\phi = \frac{1 + \sqrt{5}}{2} = 1,618...$$

Rilassando un po' la proprietà (considerando che alcune teste potrebbero indossare copricapi, avere la barba, etc.) selezioniamo tutti gli ellissi il cui rapporto rientra nel seguente range:

$$1,25 < \frac{a}{b} < 2,0$$

Infine scartiamo gli ellissi che sono contenuti l'uno nell'altro o che si intersecano (le teste devono essere necessariamente separate).

Quindi, sfruttando l'ordinamento dell'albero in cui sono stati inseriti gli ellissi, si seleziona il numero di ellissi richiesto che hanno totalizzato il numero maggiore di voti e che rispettano le proprietà geometriche descritte.

2. Struttura dei sorgenti

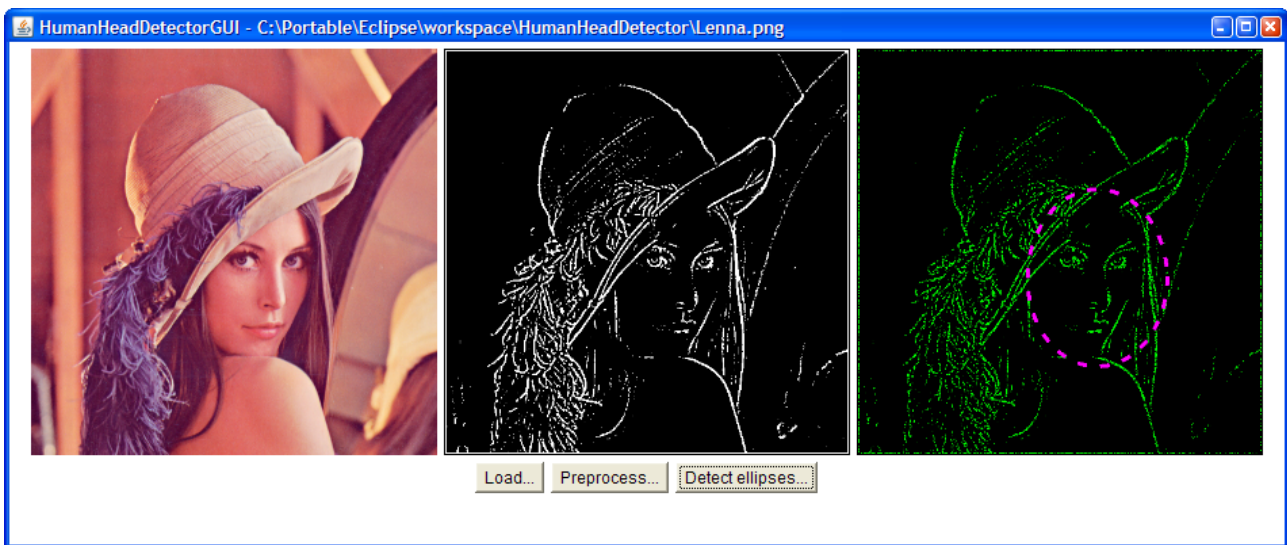
I file sorgenti sono inclusi in un unico package:

- **MImage.java** – wrapper di java.awt.image.BufferedImage, fornisce tutti gli operatori

puntuali e locali usati nel progetto

- **HumanSkinFilter.java** – filtro che esegue la segmentazione basata su colore secondo i parametri della pelle umana
- **RHTEllipse.java** – implementazione della RHT per il riconoscimento degli ellissi. Esegue anche la selezione degli stessi.
- **HumanHeadDetectorGUI.java** – interfaccia grafica per effettuare il riconoscimento delle teste umane (per uso interattivo).
- **HumanHeadDetectorCUI.java** – interfaccia a linea di comando per effettuare il riconoscimento delle teste umane (per uso batch).

3. Descrizione della GUI



La GUI presenta tre aree (canvas) con i quali è possibile visualizzare l'output di ogni stage di elaborazione. Se necessario l'immagine viene ridimensionata mantenendo le proporzioni.

Facendo click su una di queste aree si apre una finestra che mostra l'immagine nella scala originale e ne consente il salvataggio.

In basso vi sono tre pulsanti con i quali è possibile impostare i parametri ed eseguire le elaborazioni:

- **Load...** - carica l'immagine di input
- **Preprocess...** - esegue le elaborazioni dello stage 1
- **Detect ellipses...** - esegue le elaborazioni dello stage 2


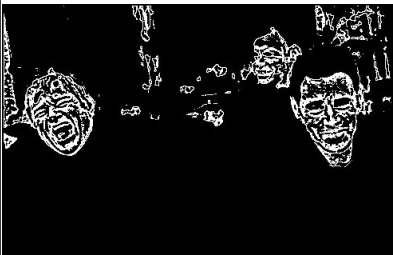
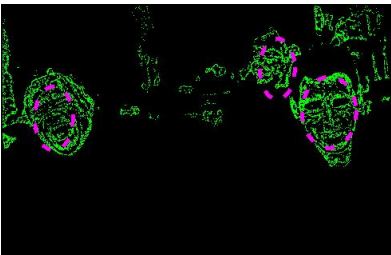


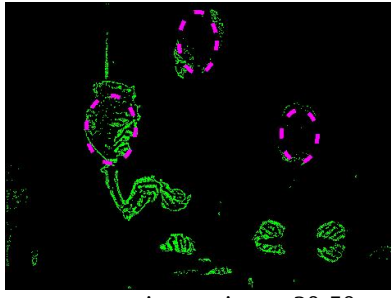


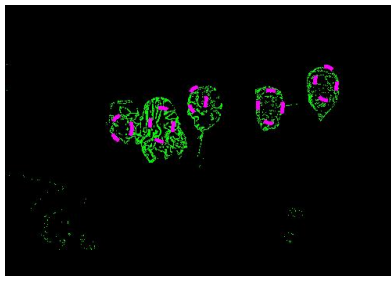


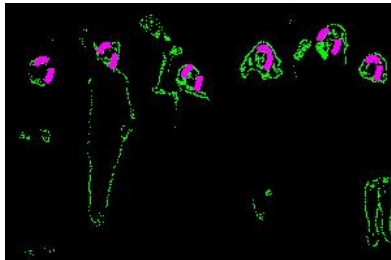
Il primo canvas mostra sempre l'immagine originale non elaborata.

Il secondo canvas mostra l'output dello stage 1.

Al termine dell'ultimo stage d'elaborazione viene mostrata nel terzo canvas l'immagine con gli ellissi tratteggiati con il color magenta e gli edge points analizzati in verde.

4. Esempi di elaborazione

Nella tabella sono mostrati alcuni esempi di elaborazione con i relativi parametri.

File	Stage 0	Stage 1	Stage 2
in-bruges.courtesy-02.jpg		 segmentazione pelle: si binary threshold: 5	 range semiasse minore: 20-50 n° di ellissi: 3
200559768-007.jpg		 segmentazione pelle: si binary threshold: 10	 range semiasse minore: 20-50 n° di ellissi: 3
AU060886.jpg		 segmentazione pelle: si binary threshold: 10	 range semiasse minore: 10-50 n° di ellissi: 5
happy_people_jumpin_g_300w.jpg		 segmentazione pelle: si binary threshold: 10	 range semiasse minore: 5-10 n° di ellissi: 7