



# UNIVERSITÀ DEGLI STUDI DI PADOVA

Facoltà di Ingegneria  
Corso di Laurea in Ingegneria dell'Informazione

## Tesi di Laurea Triennale

### Analisi del colore per il riconoscimento della mano

Skin color detection for hand recognition

**Laureando:**  
Marco Fraccaro

**Relatore:**  
Prof. Pietro Zanuttigh  
**Correlatore:**  
Ing. Carlo Dal Mutto



Anno Accademico 2010-2011



# Indice

<b>1</b>	<b>Introduzione</b>	<b>5</b>
<b>2</b>	<b>Definizione del problema e prime scelte implementative</b>	<b>7</b>
2.1	Spazi di colore per lo skin detection . . . . .	8
2.2	Acquisizione del training set e del test set . . . . .	9
2.3	Scelta dello spazio di colore . . . . .	9
<b>3</b>	<b>Sviluppo dell’algoritmo per lo skin detection</b>	<b>11</b>
3.1	L’algoritmo . . . . .	11
3.2	Riduzione del numero di pixel nello Specific Set of Values . . . . .	13
3.3	Rimozione dei pixel sparsi dello ssv . . . . .	14
3.3.1	Rimozione di piccoli gruppi connessi di pixel dello ssv . . . . .	14
3.3.2	Filtro gaussiano per la rimozione dei pixel sparsi dello ssv . . . . .	15
3.3.3	Confronto tra i due metodi proposti . . . . .	16
3.4	Uso della distanza tra colori . . . . .	16
3.5	Scelta del pixel da cui partire . . . . .	19
3.6	Edge Detection . . . . .	19
<b>4</b>	<b>Analisi dei risultati ottenuti</b>	<b>23</b>
4.1	Tempo di esecuzione . . . . .	23
4.2	Esempi di applicazione dell’algoritmo . . . . .	23
<b>5</b>	<b>Conclusioni</b>	<b>27</b>
<b>A</b>	<b>Possibili estensioni del metodo proposto</b>	<b>29</b>
A.1	Limite al numero di espansioni da ogni pixel dello ssv . . . . .	29
A.2	Gestione delle immagini scure . . . . .	30
	<b>Bibliografia</b>	<b>33</b>



# Capitolo 1

## Introduzione

Questa tesi si colloca all'interno di un ampio progetto il cui scopo finale è quello di riuscire, possibilmente soltanto con l'ausilio di normali videocamere, a controllare il computer e a navigare in ambienti 3d con i movimenti della mano. Uno dei primi passi necessari per poterlo realizzare è quello di individuare la posizione della mano nei singoli frame di un video per essere poi in grado di interpretarne il movimento. In questo elaborato verrà spiegato come è possibile, dato un singolo frame di un video (data quindi una singola immagine), individuare al meglio tutte le regioni corrispondenti al colore della pelle. Verrà sviluppato un algoritmo che sfrutta principalmente informazioni di colore e riconoscimento dei contorni (edge detection).

In letteratura sono già presenti diverse soluzioni a questo problema (noto come skin detection) ma non ve ne è una ottimale dal momento che i fattori che influenzano il colore della pelle in una immagine, e che degradano le prestazioni degli algoritmi di riconoscimento, sono molti. Tra i più importanti vi sono [1]:

- **Illuminazione:** un cambiamento nella sorgente luminosa e nell'intensità dell'illuminazione può portare a grandi cambiamenti nel colore della pelle nell'immagine. Ogni ambiente ha una sua propria illuminazione, che dipende dal fatto che si sia al chiuso o all'aperto, dall'ora del giorno, dalle condizioni climatiche, dal tipo di luce usata e dalle ombre.
- **Etnia:** ogni gruppo etnico ha un suo caratteristico colore della pelle: ad esempio il colore di un europeo, di un asiatico e di un africano sono di norma molto differenti. Inoltre, anche tra le persone della stessa etnia vi sono diversi tipi di carnagione.
- **Caratteristiche della fotocamera utilizzata:** anche sotto le stesse condizioni di illuminazione diverse fotocamere portano a risultati diversi tra loro, a seconda della qualità della stessa, del sensore utilizzato e delle diverse tecniche di adattamento luminoso utilizzate.
- **Sfondo:** a seconda dello sfondo dell'immagine lo skin detection può essere più o meno complesso: il caso ideale è quello di dover individuare le porzioni di pelle su sfondi uniformi con colori molto diversi da quello della pelle. Purtroppo però molti oggetti che ci circondano hanno colori abbastanza simili a quello della pelle; in particolar modo il legno è noto per essere molto difficile da distinguere (vedi Figura 4.1d).

Trovare un algoritmo capace di superare tutti questi problemi è impossibile, è dunque necessario introdurre da subito delle ipotesi di lavoro per semplificare il problema e progettare un algoritmo che si adatti il meglio possibile ad ogni specifica applicazione. È essenziale nel nostro caso rimanere più generali possibile: le uniche ipotesi di lavoro saranno utenti non di colore (cambiando i parametri dell'algoritmo sarebbe comunque possibile usarlo anche per quest'ultimi) e ambienti sufficientemente illuminati.

È importante notare che molti dei problemi sopra descritti sarebbero superabili concentrandosi non più sullo spettro del visibile bensì usando metodi che sfruttano tutto lo spettro. Questi sono tuttavia metodi che richiedono hardware molto costoso e dunque non sono adatti a molte applicazioni.

## Capitolo 2

# Definizione del problema e prime scelte implementative

L'algoritmo deve classificare ogni singolo pixel dell'immagine come skin pixel o come non-skin pixel. Ovviamente si vorrebbe che tutti e soli i pixel identificati come skin pixel dall'algoritmo (detected skin pixel) fossero tutti e soli i pixel reali della pelle nell'immagine (real skin pixel). Tuttavia, come si può notare dalla tabella seguente, nella pratica si possono verificare 4 diverse situazioni:

1. Un real skin pixel viene correttamente identificato dall'algoritmo come skin pixel;
2. Un real non-skin pixel viene erroneamente identificato dall'algoritmo come pixel della pelle, causando quindi un falso positivo;
3. Un real skin pixel viene erroneamente identificato dall'algoritmo come pixel non della pelle, causando un falso negativo;
4. Un real non-skin pixel è correttamente identificato come pixel non della pelle.

<b>Real pixel</b> <b>Detected pixel</b>	<b>Skin</b>	<b>Non-skin</b>
<b>Skin</b>	Ok	False positive
<b>Non-skin</b>	False negative	Ok

**Figura 2.1**

Nel nostro caso è importante ridurre al massimo i falsi negativi, cercando, al tempo stesso, di tenere il più possibile basso il numero di falsi positivi: questo è infatti solo il primo passo da effettuare per l'hand recognition, ed è importante per non compromettere il funzionamento dei passi successivi essere certi di avere un alto TPR (true positive rate) anche al costo di un FPR (false positive rate) non troppo basso.

## 2.1 Spazi di colore per lo skin detection

La scelta dello spazio di colore da utilizzare per rappresentare le immagini nel computer è di fondamentale importanza, e influisce notevolmente nelle prestazioni dell'algoritmo. Una sfida importante per lo skin detection è infatti quella di rappresentare il colore in un modo il più possibile insensibile alle variazioni di illuminazione. In letteratura sono stati usati diversi spazi di colore per cercare di superare questo problema; in seguito ne verranno descritti i principali concentrandosi in particolare modo su come viene influenzato lo skin detection [1, 2].

### RGB

Nello spazio di colore RGB i colori sono codificati come combinazione additiva dei tre colori primari: rosso (Red), verde (Green), blu (Blue). Uno dei maggiori vantaggi e il motivo per cui è abbastanza impiegato è la sua semplicità: lo spazio di colore RGB è infatti il più comunemente usato nelle immagini digitali. Non è tuttavia percettivamente uniforme dal momento che la distanza tra due colori nello spazio RGB non corrisponde linearmente alla percezione umana. Inoltre le componenti della luminanza e della cromaticità non sono separate e dunque una piccola variazione della luminosità dell'immagine può influenzare anche abbastanza tutte e tre le componenti.

### XYZ

Lo spazio di colore CIE (Commission Internationale de l'Éclairage) XYZ deriva da una trasformazione lineare delle componenti RGB sopra descritte. Il vantaggio derivante dall'uso di questo spazio di colore rispetto all'RGB è il fatto che a differenza di quest'ultimo, nel quale alcuni colori richiedono coordinate negative, con lo spazio XYZ è rappresentabile tutta la gamma del visibile con coordinate positive. Anche in questo caso comunque le differenze tra colori non sono percepite allo stesso modo in regioni differenti dello spazio di colore.

### CIELAB

Lo spazio di colore CIELAB separa la variabile di luminosità  $L$  (che dipende soltanto dal valore di luminanza  $Y$ ) da due variabili di cromaticità ( $a$  e  $b$ ). Le tre variabili sono percettivamente uniformi. Nonostante i grandi vantaggi dati dall'uniformità questo spazio non è però sempre usato per lo skin detection a causa delle trasformazioni non lineari a cui bisogna ricorrere per arrivarci a partire dalle componenti RGB che aumentano il tempo di esecuzione dell'algoritmo.

### CIELUV

Lo spazio di colore CIELUV è simile come caratteristiche al sopra presentato CIELAB: la componente  $L$  è esattamente la stessa, mentre le componenti  $u$  e  $v$  sono ottenute attraverso delle trasformazioni (ancora non lineari) un po' diverse ma che mantengono comunque l'uniformità percettiva.



## HSV

Nello spazio HSV, le componenti, Hue, Saturation e Value, derivano da trasformazioni non lineari dello spazio RGB. Ogni componente si avvicina molto alla percezione dell'occhio umano di tonalità, saturazione e valore del colore, rendendo lo spazio HSV un'ottima alternativa per lo skin detection, nonostante non sia percettivamente uniforme.

## YCbCr

Lo spazio di colore YCbCr appartiene alla classe degli spazi di colore ortogonali: il colore è rappresentato con componenti statisticamente indipendenti; viene dunque ridotta la ridondanza presente nello spazio RGB. Dal momento che le componenti di luminanza e cromaticità sono separate questo spazio è adatto allo skin detection. Uno dei maggiori vantaggi che derivano dall'uso dell'YCbCr è il fatto che questo è lo spazio di colore principalmente usato nelle videocamere: non sono dunque richieste trasformazioni tra spazi di colore prima di iniziare la fase di elaborazione dell'algoritmo. Purtroppo però non è percettivamente uniforme.

## 2.2 Acquisizione del training set e del test set

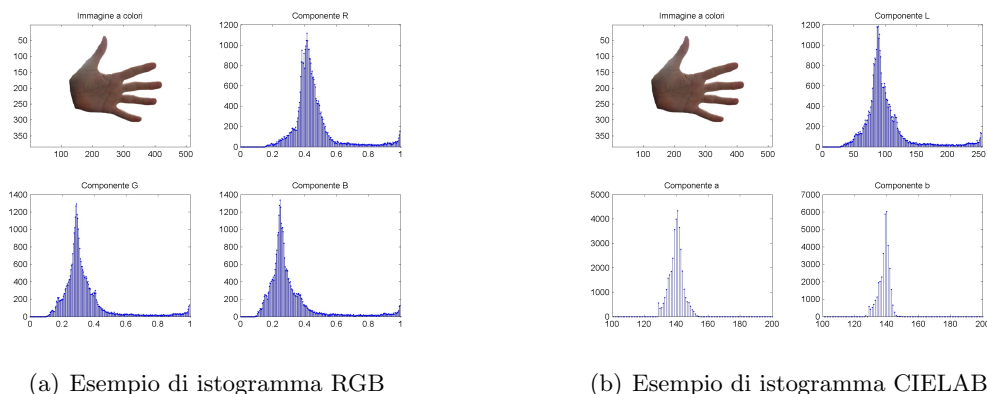
Per riuscire a capire quale fosse lo spazio di colore adatto al nostro scopo e quali fossero i valori delle componenti dello spazio di colore che meglio discriminavano skin e non-skin pixel, sono state necessarie una serie di foto di prova; si sono infatti analizzate circa una trentina di foto di mani di persone diverse, in diversi ambienti e con diverse illuminazioni (lampada alogena, luce solare, neon, lampada da tavolo, luce della stanza). Ove possibile è anche stata inserita un'ombra sulla mano, interponendo un cartoncino nero tra la sorgente luminosa e la scena. In alcuni casi tuttavia l'intensità luminosa era talmente bassa o distribuita uniformemente da rendere trascurabile l'effetto dell'ombra. Sono inoltre state acquisite circa 200 foto in situazioni tipiche di applicazione adatte a testare l'algoritmo.

## 2.3 Scelta dello spazio di colore

Acquisite le foto, si sono svolte delle analisi della distribuzione dei pixel della pelle con i diversi spazi di colore. Per farlo sono stati creati in Matlab degli istogrammi che mostrano in ascissa i vari valori possibili delle componenti dello spazio di colore e in ordinata il numero di pixel dell'immagine con quella componente (sono state usate le definizioni degli spazi di colore presenti nel Image Processing Toolbox di Matlab).

Dopo un'analisi degli istogrammi appare opportuno concentrarsi su spazi di colore uniformi, principalmente per tre motivi:

1. Le componenti relative alla luminanza e alla crominanza sono separate, è quindi possibile trascurare la prima e potersi quindi adattare al maggior numero di illuminazioni possibili. È comunque opportuno sottolineare che in realtà queste due componenti non sono totalmente indipendenti, come si può notare in caso di immagini troppo scure o troppo chiare (che sono perciò molto difficili



**Figura 2.2:** Esempi di istogrammi

da analizzare); tuttavia nella maggior parte dei casi i due spazi di colore CIE le separano abbastanza bene.

2. Le componenti dei pixel della pelle sono concentrate in insiemi relativamente stretti di valori, se paragonati ad esempio allo spazio RGB. È perciò possibile riuscire a isolare abbastanza bene le porzioni di pelle senza includere nell'immagine finale troppi oggetti indesiderati.
3. Usando uno spazio di colore uniforme sarà possibile utilizzare nell'algoritmo il concetto di distanza euclidea tra colori, non includendo quindi pixel con un colore troppo diverso dal colore della pelle.

Tra CIELUV e CIELAB si è scelto di utilizzare nella versione definitiva dell'algoritmo quest'ultimo, ma l'utilizzo del CIELUV avrebbe portato sostanzialmente agli stessi risultati. I test hanno infatti dimostrato come alcune immagini venissero elaborate meglio da uno, altre meglio dall'altro, senza nessuna predominanza di uno dei due. L'utilizzo di uno spazio di colore uniforme porta d'altra parte a un maggiore costo computazionale, ma i vantaggi che si possono ottenere lo compensano.

## Capitolo 3

# Sviluppo dell'algoritmo per lo skin detection

Nel seguito verrà innanzitutto descritto brevemente l'algoritmo proposto per lo skin detection, che è stato implementato in Matlab. Scopo di questa prima parte non è soffermarsi sui dettagli e su tutte le motivazioni che hanno portato a determinate scelte implementative, bensì dare un'idea generale del funzionamento dell'algoritmo, in modo da rendere più semplice la lettura del resto dell'elaborato. Nei paragrafi successivi verranno poi approfondite e meglio spiegate le principali scelte implementative prese in fase di progettazione dell'algoritmo, sia quelle incluse nella versione finale che quelle scartate (Appendice A).

### 3.1 L'algoritmo

L'algoritmo per l'identificazione dei pixel della pelle è basato su hysteresis thresholding (sogliatura a isteresi). Scelto uno spazio di colore (nel nostro caso il CIELAB) si definiscono quindi due insiemi di pixel:

- ***Specific Set of Values (ssv)***: in questo insieme stanno i pixel che sono quasi certamente pixel che appartengono alla pelle. I valori per ogni componente dello spazio di colore di questi pixel vanno determinati attraverso un'analisi delle immagini di prova. L'ideale sarebbe stato trovare dei valori comuni a tutte le immagini; essendo tuttavia le immagini usate talvolta molto diverse tra loro a causa delle diverse illuminazioni non è stato possibile farlo. Le soglie di questo insieme sono state quindi cercate empiricamente osservando gli istogrammi. Per adattarsi al maggior numero di illuminazioni possibili e tenere contemporaneamente limitato il numero di pixel appartenenti allo ssv, le soglie dello ssv sono state prese in due piccoli insiemi disgiunti di valori.
- ***Extended Set of Values (esv)***: a questo insieme, che deve contenere lo ssv, appartengono i pixel che con probabilità abbastanza alta sono skin pixel. L'ideale sarebbe dunque usare l'unione di tutti i valori presenti in tutti gli istogrammi delle immagini di prova. Nella determinazione delle soglie è tuttavia opportuno non essere così rigorosi: sono dunque stati esclusi alcuni valori poco frequenti e quasi anomali che aumentavano la cardinalità dell'insieme peggiorando le prestazioni dell'algoritmo. Le soglie sono state prese in un insieme

continuo di valori (come ci si può aspettare dall'uso di uno spazio di colore uniforme).

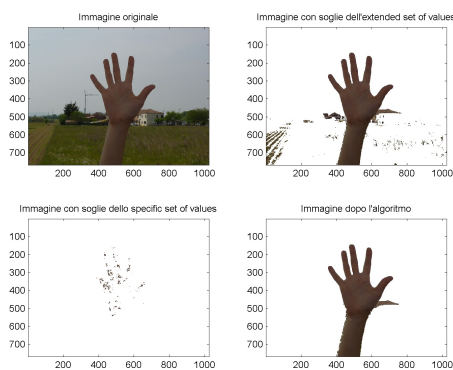
Come risulterà chiaro in seguito, è importante che il numero di pixel dello specific set of values sia abbastanza basso. Oltre a limitare il range dei valori che definiscono questo insieme, viene quindi applicato all'immagine contenente solo questi pixel un filtraggio spaziale lineare, ottenuto attraverso l'uso di un filtro gaussiano: vengono così lasciati nello ssv soltanto pixel appartenenti a regioni dell'immagine in cui sono presenti molti pixel contenuti in questo insieme.

Segue poi la fase di espansione dell'algoritmo: a partire da ogni pixel dello specific set of values si controllano i pixel che lo circondano che, se appartenenti all'extended set of values, vengono classificati come skin pixel soltanto se:

1. Il colore di questo pixel non è troppo diverso dal colore medio di tutti quelli dello specific set of values (il realtà, come verrà meglio spiegato in seguito, sarà una media pesata che verrà continuamente aggiornata durante tutta la fase di espansione).
2. Il pixel non appartiene ad un edge: in questo modo, a patto di riuscire a individuare correttamente tutti i contorni presenti nell'immagine, si riesce ad impedire l'espansione oltre i bordi delle porzioni di pelle dell'immagine.

Questa operazione di espansione è poi ripetuta a partire da ogni nuovo pixel classificato come skin pixel.

Come si può notare dalla figura 3.1 attraverso l'uso di questo algoritmo vengono presi molti skin pixel non appartenenti allo ssv ma appartenenti all'esv ed esclusi molti non-skin pixel appartenenti all'extended set of values.



**Figura 3.1:** Esempio di applicazione dell'algoritmo con buoni risultati

Va comunque precisato che non sempre questo algoritmo porta a risultati che possono giustificare l'elevato numero di operazioni da svolgere. In figura 3.2 vi è un esempio di quanto appena detto: la differenza tra l'immagine con soglie dell'esv e quella dopo l'algoritmo è quasi trascurabile.

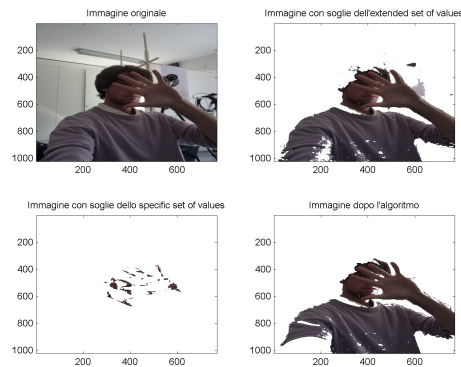
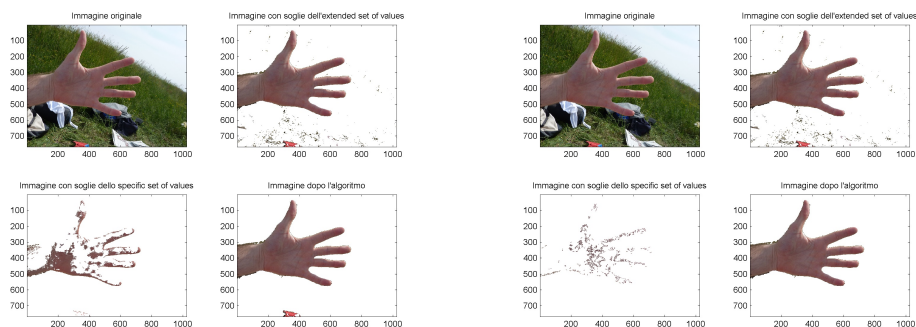


Figura 3.2: Esempio di applicazione dell'algoritmo con scarsi risultati

## 3.2 Riduzione del numero di pixel nello Specific Set of Values

Avere un numero elevato di pixel nello ssv non porta ad alcun vantaggio: per come è strutturato l'algoritmo, infatti, per far includere nell'immagine finale una porzione di pelle ne è sufficiente anche uno solo. È quindi lecito pensare di poter senza problemi ridurre questo numero limitando la dimensione degli intervalli di ogni componente dello spazio di colore che definiscono lo ssv. È in questo modo possibile riuscire a eliminare alcuni oggetti in cui vi erano pochi pixel nello ssv (come ad esempio quello rosso in basso nell'immagine in figura 3.3a).



(a) Prima della riduzione del numero di pixel      (b) Dopo la riduzione del numero di pixel

Figura 3.3: Esempio di rimozione di un oggetto grazie alla riduzione del numero di pixel nello ssv

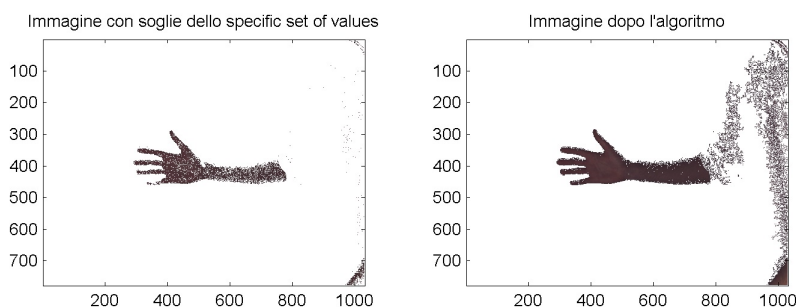
In ogni caso la riduzione del numero di pixel dello ssv non compromette il funzionamento dell'algoritmo in quanto in una porzione di pelle ci sono moltissimi tipi di rosa e certamente alcuni ne verranno presi.

### 3.3 Rimozione dei pixel sparsi dello ssv

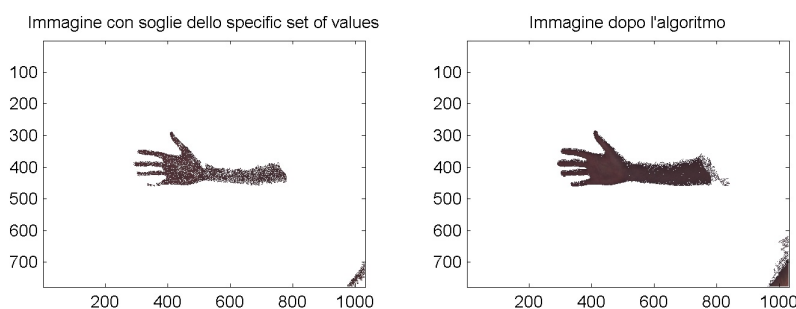
#### 3.3.1 Rimozione di piccoli gruppi connessi di pixel dello ssv

La maggior parte dei pixel dello specific set of values è concentrata, per lo meno in teoria, nelle porzioni di pelle dell'immagine. In queste zone quindi il numero di pixel isolati, cioè non circondati da altri skin pixel, è basso. È quindi ragionevole pensare di poter eliminare dallo specific set of values tutti i pixel isolati, in modo da eliminare anche tutti i non-skin pixel rilevati dall'algoritmo come skin pixel. Se siano o no più i real skin pixel eliminati che i real non-skin pixel non ha importanza in quanto i singoli real skin pixel eliminati, che sono comunque presumibilmente vicini a pixel dello specific set of values, verranno con alta probabilità inclusi nuovamente nell'immagine dall'algoritmo (fanno sempre parte dell'extended set of values). Questo ragionamento può essere esteso anche a gruppi non troppo grandi (composti ad esempio da una decina di elementi) di pixel connessi tra di loro, in cui per pixel connessi si intende 8-connected pixel: i pixel sono connessi sia se hanno un lato in comune, sia se hanno un vertice in comune.

Eliminando questi gruppi di pixel dallo specific set of values, come si può vedere dalle immagini seguenti, i risultati migliorano: le regioni non di pelle che prima venivano incluse a causa dei gruppi sparsi di pixel (ora rimossi) vengono scartate.



(a) Prima della rimozione dei pixel sparsi



(b) Dopo la rimozione dei pixel sparsi

**Figura 3.4:** Eliminando gruppi di pixel con meno di 10 elementi si ottiene un netto miglioramento.

In termine di prestazioni l'algoritmo rimane circa invariato in quanto l'aumento in numero di operazioni necessario per rimuovere i gruppi di pixel sparsi viene compensato dalla diminuzione di pixel dello specific set of values e quindi dal minor

numero di operazioni da eseguire nella fase di espansione dell'algoritmo.

*Osservazione:* Si deve prestare molta attenzione nella scelta della dimensione dei gruppi di pixel connessi che verranno eliminati. I 10 elementi presi in considerazione negli esempi precedenti (ovviamente il numero deve rimanere proporzionale alla risoluzione delle immagini - le immagini usate sono 1032x778) sono un buon numero nel caso di immagini con un numero abbastanza alto di skin pixel in cui si è abbastanza sicuri che i pixel di interesse saranno abbastanza raggruppati (come le precedenti). Nel caso di immagini con pochi skin pixel è opportuno arrivare a qualche unità o addirittura a nessuno nel caso di piccolissime porzioni di pelle dell'immagine. Analogamente, per immagini con ampie porzioni di pelle questo numero può venire aumentato. Un buon compromesso per rimanere il più generali possibili potrebbe essere 2 o 3 pixel.

### 3.3.2 Filtro gaussiano per la rimozione dei pixel sparsi dello ssv

Un metodo per la rimozione dei pixel sparsi dello specific set of values alternativo a quello appena presentato si può basare sulla densità degli stessi: questi saranno infatti molto densi nelle zone in cui sono presenti porzioni di pelle e presumibilmente meno concentrati, almeno nella maggior parte dei casi, nelle altre zone. È quindi logico, basandosi sulle stesse considerazioni fatte per la rimozione dei gruppi connessi di pixel sparsi, pensare di poter eliminare i pixel poco densi dello specific set of values.

È comodo per un'analisi del metodo pensare alla matrice contenente i pixel dello specific set of values come ad una immagine in bianco e nero, in cui ogni punto assume il valore 0 se non appartenente allo ssv, 1 altrimenti (l'implementazione in Matlab dell'algoritmo rappresenta proprio in questo modo i pixel dello ssv). Si può procedere quindi attraverso un filtraggio spaziale lineare, cambiando cioè il valore di ogni pixel con una funzione lineare dei pixel che lo circondano. Innanzitutto si deve definire un'opportuna matrice quadrata  $M$  di coefficienti, chiamata filtro o maschera, per comodità presa di dimensione dispari in modo da poterne individuare univocamente il centro. Il processo consiste ora nel muovere il centro di  $M$  in tutti i punti dell'immagine; ad ogni punto  $(x, y)$  la risposta del filtro in quel punto è data dalla somma dei prodotti dei coefficienti del filtro e del valore dei corrispondenti pixel dell'immagine vicini a  $(x, y)$ . Per continuare l'analogia con l'immagine in bianco e nero, l'immagine risultante dopo la filtratura sarà un'immagine in scala di grigi.

Il filtro utilizzato è un filtro passabasso gaussiano: i coefficienti sono maggiori verso il centro della matrice e decrescono allontanandosi da questo. In questo modo si può dare quindi un'importanza maggiore ai pixel vicini al pixel filtrato. La deviazione standard del filtro è stata scelta abbastanza alta: 1.75. I pixel distanti fino a 4 pixel da quello considerato hanno dunque un peso non trascurabile. La dimensione della maschera deve essere scelta in base alla deviazione standard: la matrice non deve essere così grande da portare a conti inutili dati da pixel pesati con coefficienti troppo piccoli. Con deviazione standard 1.75 la maschera è stata scelta 9x9, ogni pixel è dunque funzione di 81 pixel: se stesso e gli 80 pixel che lo circondano. Una volta filtrata l'immagine dei pixel dello ssv si deve fare in modo che i pixel in regioni poco dense vengano esclusi dallo ssv. Questi sono i punti il cui valore filtrato è basso: vengono quindi tenuti soltanto i pixel il cui valore supera una certa soglia (chiamata  $w_{min}$  in figura 4.3), presa nell'algoritmo in modo tale da essere molto selettivi. È però anche possibile scegliere la deviazione standard molto

alta (50) in modo che i coefficienti della matrice siano tutti uguali e che tutti i pixel abbiano la stessa importanza. Così facendo la fase di rimozione dei pixel poco densi si riduce all'eliminazione dei pixel il cui numero dei pixel dello ssv vicini al pixel in analisi è basso.

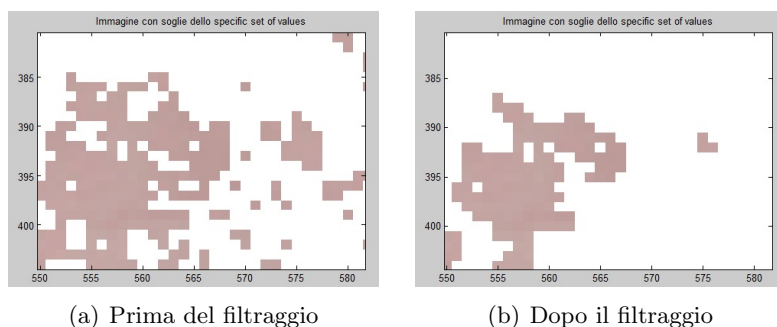


Figura 3.5: Esempio con deviazione standard 1.75

### 3.3.3 Confronto tra i due metodi proposti

Da un'analisi delle immagini di test fatta con i due metodi risulta che la differenza non è molto rilevante (ce se lo poteva aspettare per come è strutturato l'algoritmo); entrambi portano ad un buon risultato. Certamente però dal punto di vista teorico è più corretto il metodo basato sul filtraggio, ed è per questo che è stato scelto quest'ultimo nella versione definitiva dell'algoritmo.

## 3.4 Uso della distanza tra colori

L'utilizzo di uno spazio di colore uniforme suggerisce di usare anche le informazioni sulla distanza tra i colori. Tra le varie definizioni di distanza possibili si è scelto di usare per il momento la semplice distanza euclidea, ma si potrebbe provare con altri tipi di distanze (come ad esempio la CIEDE2000 [3]) che, sebbene più dispendiose in termini di operazioni, correggono alcuni errori della distanza euclidea.

Con queste informazioni aggiuntive si spera di escludere dall'immagine non-skin pixel appartenenti all'esv i cui valori delle singole componenti sono comunque vicini a quelli degli skin pixel (come ad esempio i pixel del legno, vedi figura 3.6).

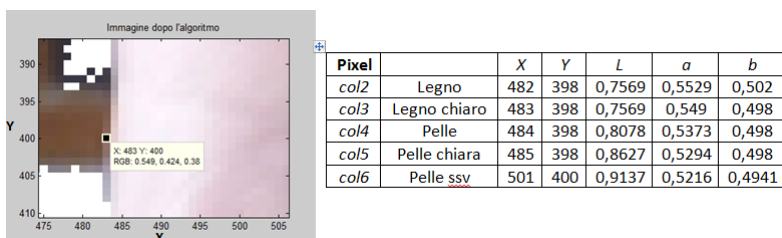


Figura 3.6: Ingrandimento dell'immagine in figura 3.9a; vi è una mano e sullo sfondo del legno. Si notino in particolare nella tabella i valori delle componenti dei pixel (spazio di colore CIELAB).



Distanza	col3	col4	col5	col6
col2	0,0055	0,0535	0,1085	0,1602
col3		0,0523	0,1077	0,1593
col4			0,0555	0,1071
col5				0,0517

**Figura 3.7:** Distanza (in termini di colore) tra i pixel della tabella in figura 3.6

Come si può intuire dalla tabella in figura 3.7, la distanza tra i pixel del legno e i pixel della pelle appartenenti allo ssv è maggiore della distanza tra due pixel della pelle o di quella tra due pixel del legno; è quindi possibile trovare una soglia che permetta, a partire da una misura media delle componenti degli pixel dello ssv di discriminare skin pixel e non-skin pixel.

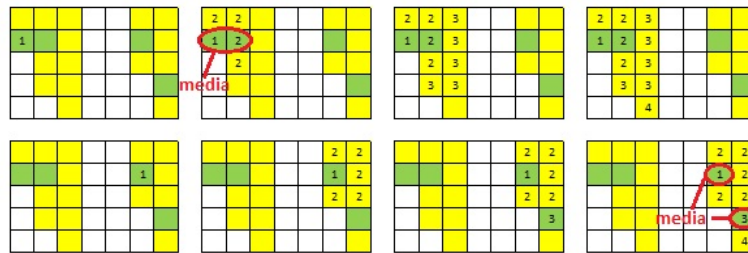
La fase di espansione dell'algoritmo, che include l'utilizzo dell'informazione sulla distanza, si basa sui seguenti passi.

1. Si parte da un singolo pixel dello ssv (nel paragrafo successivo è spiegato come viene scelto), che viene preso come colore di riferimento per i confronti, e che sarà incluso nell'immagine finale.
2. Ci si espande ai pixel che lo circondano
  - Se il pixel appartiene allo ssv (e quindi anche all'esv) si fa la media pesata tra il suo colore e il colore di riferimento e la si usa come nuovo colore di riferimento. Il pixel viene incluso nell'immagine finale e viene tolto dallo specific set of values in modo da evitare un'inutile analisi successiva da parte dell'algoritmo.
  - Se il pixel appartiene all'extended set of values, se la distanza tra il colore del pixel considerato e quello di riferimento è minore di una soglia fissata (negli esempi seguenti 0.2) il pixel viene incluso nell'immagine finale, altrimenti non viene incluso ma ne viene tenuta in memoria la posizione. In questo caso il colore di riferimento resta invariato.
3. Si ripete il procedimento a partire da tutti i nuovi pixel appartenenti all'esv trovati, indipendentemente dal fatto che siano stati inclusi o no nell'immagine finale.
4. La fase di espansione termina quando in un passo dell'algoritmo non viene più analizzato alcun pixel dell'extended set of values (sarà dunque stata analizzata una regione di pixel connessi dell'esv).

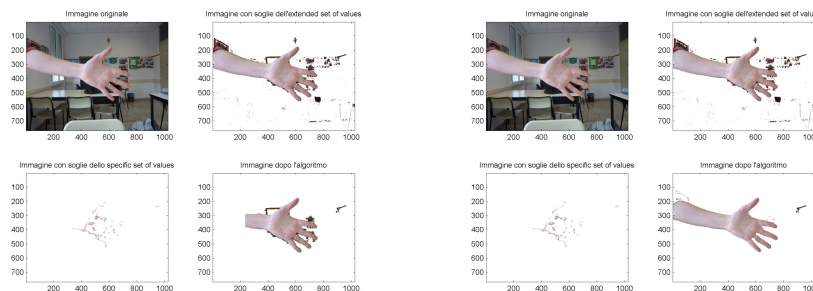
Si fa ora un nuovo controllo sui pixel scartati perché distanti, per eventualmente includerli nell'immagine finale. Questo controllo è reso necessario da come viene modificato in fase di esecuzione dell'algoritmo il colore di riferimento. Infatti, per certe configurazioni dello specific set of values, alcuni pixel, specialmente quelli visitati per primi, potrebbero essere esclusi anche se vicini (in termine di colore) alla media del colore di tutti i pixel dello specific set of values: i primi pixel visitati dello ssv potrebbero essere ad esempio tutti più scuri rispetto alla media finale.

5. Si cerca, se presente, un altro pixel dello specific set of values mai visitato e si ripetono i punti 1, 2, 3 e 4. Per il colore di riferimento si è scelto di non considerare le informazioni sulla media dei colori trovate in precedenza. Due zone disgiunte di pelle potrebbero avere infatti un colore medio diverso (si pensi ad esempio a due mani di due persone diverse nella stessa scena).

In figura 3.8 è mostrato un esempio di espansione; in figura 3.9 sono mostrati due esempi di come migliora l'algoritmo attraverso l'uso della distanza tra colori.

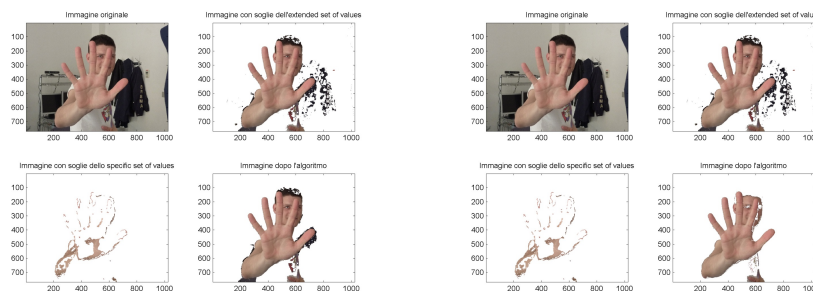


**Figura 3.8:** Esempio di espansione. Le caselle verdi rappresentano pixel dello ssv, quelle gialle pixel dell'esv.



(a) Prima dell'uso della distanza tra colori

(b) Dopo l'uso della distanza tra colori



(c) Prima dell'uso della distanza tra colori

(d) Dopo l'uso della distanza tra colori

**Figura 3.9:** Esempi di applicazione

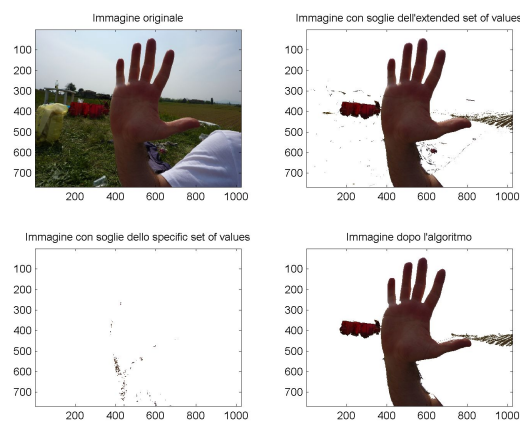
### 3.5 Scelta del pixel da cui partire

Per come viene modificato il colore di riferimento man mano che si incontra un pixel dello ssv, questa procedura dipende dal pixel da cui si è partiti. È quindi opportuno non partire da un pixel qualsiasi che potrebbe ad esempio essere sul bordo o in una zona della mano in cui è presente dell'ombra e quindi essere di un colore molto diverso dalla media, bensì è meglio partire da un pixel appartenente alla più grande regione connessa di pixel dello ssv, e quindi probabilmente un pixel molto più simile alla maggioranza degli skin pixel.

### 3.6 Edge Detection

Un grande miglioramento per l'algoritmo consiste nell'utilizzare informazione sugli edge.

Come si può vedere dall'immagine in figura 3.10, trovando il bordo della mano e impedendo l'espansione oltre a questo, verrebbero escluse due ampie zone di non-skin pixel.



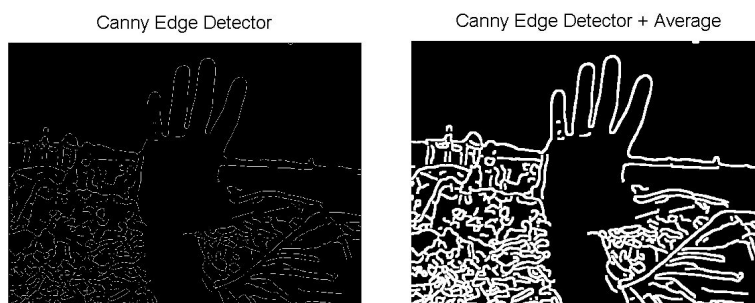
**Figura 3.10:** Risultato dell'algoritmo prima dell'uso dell'edge detection

L'edge detection, ottenuta attraverso l'uso dell'algoritmo di Canny [4], viene effettuata sulla componente dell'immagine relativa alla luminanza principalmente per due motivi:

- I contorni vengono trovati meglio rispetto all'edge detection sulle componenti a e b.
- Per utilizzare un'informazione nuova, che non era stata tenuta in considerazione per la definizione dello specific ed extended set of values (la luminanza veniva usata solo nella distanza tra i colori).

L'edge detection tuttavia spesso fornisce dei bordi non continui; questo deve essere evitato, poiché un solo buco in un edge permetterebbe all'algoritmo di espandersi oltre a questo, rendendone inutile la ricerca. È perciò opportuno fare in modo

di renderli delle linee continue, anche a costo di una minor precisione nella localizzazione dei contorni (precisione che comunque non è necessaria per individuare la mano). Questo è stato fatto tramite l'applicazione all'immagine che rappresenta gli edge di un filtraggio lineare realizzato attraverso un filtro media (averaging filter) di grandezza  $7 \times 7$ , prendendo poi tutti i pixel di valore strettamente positivo. In questo modo molti edge vengono resi continui, al costo però di un maggiore spessore dei bordi (come mostrato in figura 3.11): un filtro  $7 \times 7$  crea bordi spessi 7 pixel. Eventuali bordi all'interno della mano, esclusi nella fase di espansione dell'algoritmo, verranno tuttavia, come descritto in dettaglio più avanti, inclusi nell'immagine finale: la larghezza dei bordi, pur dovendo comunque rimanere più piccola possibile, non è dunque un parametro di fondamentale importanza per una prima analisi.

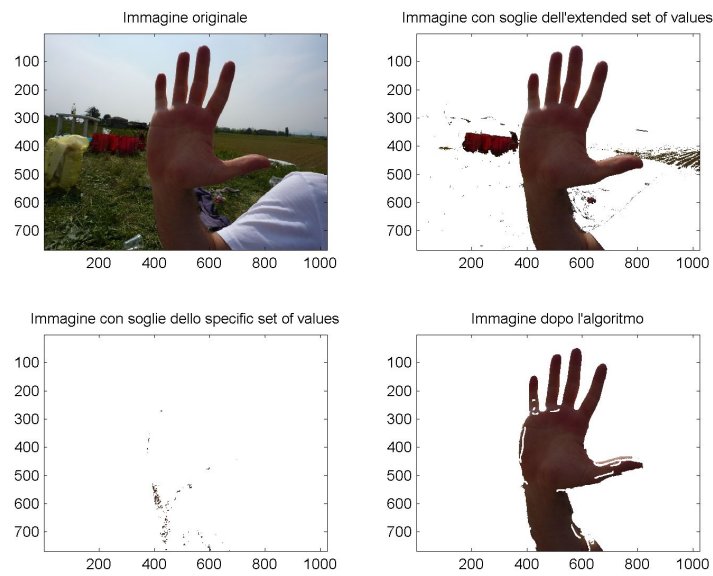


**Figura 3.11:** Esempio di segmentazione e inspessimento dei bordi dell'immagine in figura 3.10

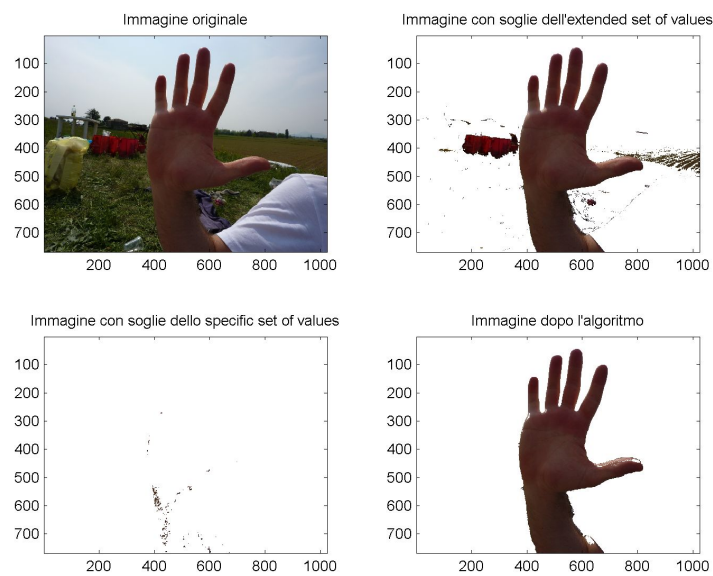
Eventuali pixel dello specific set of values appartenenti ad un edge vengono esclusi dallo ssv, sia per semplicità in fase di implementazione dell'algoritmo, sia per il fatto che, se veramente appartenenti ad un edge, sono probabilmente di colore un po' diverso rispetto alla media di tutti gli skin pixel: escludendoli viene migliorata dunque la media pesata che viene fatta per calcolare la distanza tra i colori.

La fase di espansione dell'algoritmo, descritta nel paragrafo 3.4, rimane praticamente inalterata, ad eccezione del fatto che in presenza di un edge questa termina (tenendo in memoria il pixel in cui è stato trovato). Come si può notare dall'immagine in figura 3.12, i risultati sono molto buoni: le due porzioni di non-skin pixel sono state rimosse. Rimane tuttavia da riempire gli edge all'interno delle porzioni di pelle (in alcuni casi vi è una presenza maggiore di edge all'interno della mano rispetto a questa immagine).

Questo viene fatto sfruttando il fatto che l'algoritmo memorizza i pixel appartenenti ad un edge in cui si è fermata l'espansione. Partendo solamente da questi (e non dunque da tutti gli edge dell'immagine) si procede ad un'ulteriore fase di espansione (i risultati sono mostrati nell'immagine in figura 3.13). Il numero di espansioni consentite da ognuno di questi pixel dipende dalla dimensione del filtro media usato per rendere continui i contorni. Come già detto un filtro  $7 \times 7$  crea bordi spessi 7 pixel: permettendo un numero di espansioni pari a  $\text{ceiling}(7)=4$  si ha che bordi circondati interamente da skin pixel vengono inclusi nell'immagine finale (si espande a partire da tutti i lati del bordo), mentre per i bordi esterni delle regioni di pelle vengono al massimo presi 4 pixel (solamente tuttavia se appartenenti all'extended set of values).



**Figura 3.12:** Passo intermedio dell'algoritmo: vi sono ancora edge all'interno della mano



**Figura 3.13:** Risultato dell'algoritmo dopo l'uso dell'edge detection: sono state eliminate le due regioni di non-skin pixel



## Capitolo 4

# Analisi dei risultati ottenuti

Per un'analisi esatta dei risultati sarebbe necessario un database di immagini in cui ciascun pixel è etichettato come skin pixel o non-skin pixel. In svariati articoli ne vengono citati principalmente due: l' ECU Face Detection Database e il Compaq Database. Purtroppo non è stato possibile riuscire a procurarseli e dunque un'analisi esatta dei risultati in termini di TPR e FPR non è disponibile.

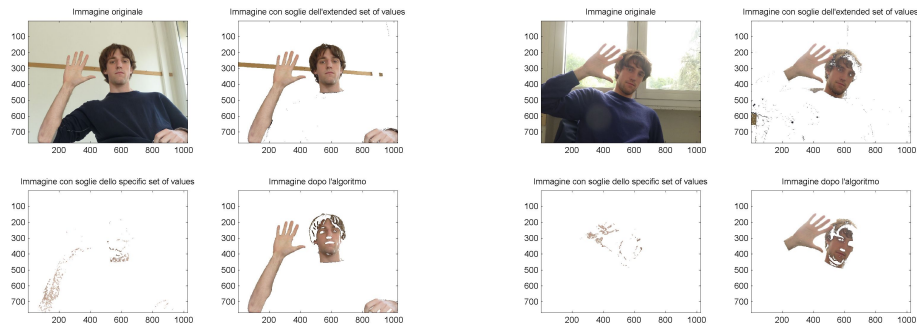
Riuscire perciò a dare una valutazione obiettiva dell'efficacia dell'algoritmo non è semplice. Tuttavia, un metodo spesso utilizzato per lo skin detection (è il metodo più semplice) è quello dell'explicit skin-color space thresholding [1], che consiste nel considerare come skin pixel soltanto i pixel contenuti in certe ampie regioni dello spazio di colore: è dunque la stessa cosa che è stata fatta nello sviluppo dell'algoritmo quando si è definito l'extended set of values. Un confronto possibile è dunque tra il risultato dell'algoritmo e l'immagine con i valori dell'extended set of values. Anche in questo caso tuttavia la valutazione dell'efficacia dell'algoritmo è soggettiva. I risultati comunque sembrano promettenti: molto spesso vengono escluse regioni abbastanza ampie di non-skin pixel appartenenti all'extended set of values. Quanto appena detto è stato possibile notarlo soprattutto in fase di progettazione dell'algoritmo: i nuovi metodi apportati non hanno praticamente mai peggiorato il risultato se confrontato all'immagine con le soglie dell'extended set of values, ma sempre portato a dei buoni miglioramenti.

### 4.1 Tempo di esecuzione

Per quanto riguarda il tempo di esecuzione dell'algoritmo l'implementazione in Matlab è per ora ancora lenta: ogni immagine viene processata in più o meno 15 secondi (l'uso real time dell'algoritmo è perciò ancora impossibile). C'è da dire tuttavia che l'implementazione potrebbe essere molto ottimizzata passando ad un linguaggio di programmazione come C++ (usando librerie con operazioni ottimizzate) o utilizzando ad esempio una GPU.

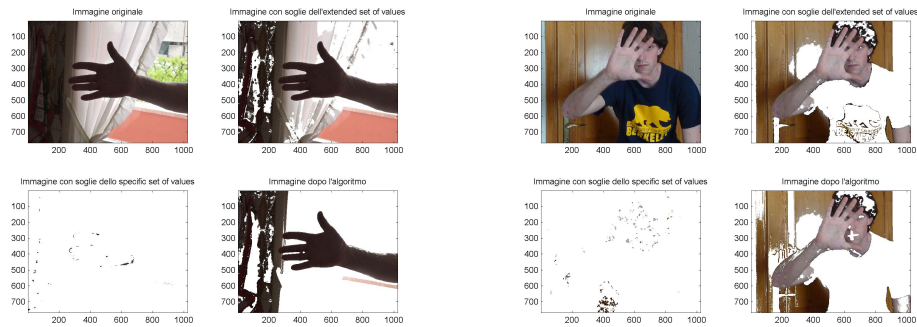
### 4.2 Esempi di applicazione dell'algoritmo

Vengono ora presentati alcuni esempi di applicazione dell'algoritmo (oltre a quelli già mostrati nei capitoli precedenti) in diverse scene e con diversi tipi di illuminazione.



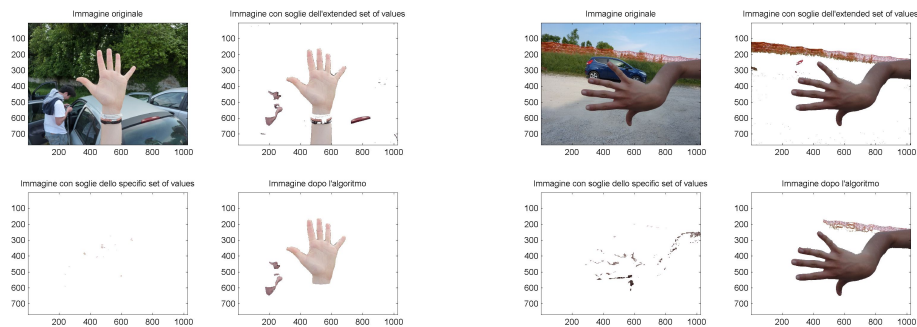
(a) In questo caso lo skin detection è molto buono: la distanza tra colori permette di eliminare la parte di manica del maglione vicina alla mano, mentre la segmentazione consente di rimuovere l'asta di legno orizzontale.

(b) L'algoritmo rimuove quasi tutti i non-skin pixel tranne una porzione di capelli.



(c) L'immagine finale ha ancora un'ampia zona di non-skin pixel; sono stati tuttavia eliminati, rispetto all'immagine con soglie dell'extended set of values, la tenda e gran parte del muro della terrazza (che tra l'altro è rosa).

(d) Questa immagine dimostra come sia difficile distinguere il legno e la pelle.

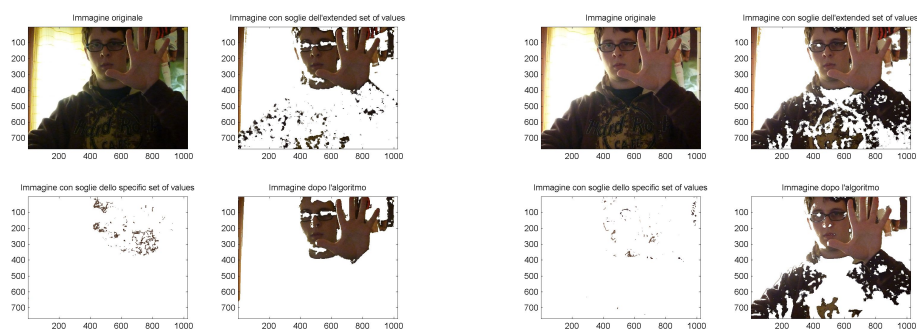


(e) L'assenza di pixel nello ssv nella parte bassa dell'immagine causa l'esclusione della porzione di pelle al di sotto dei braccialetti.

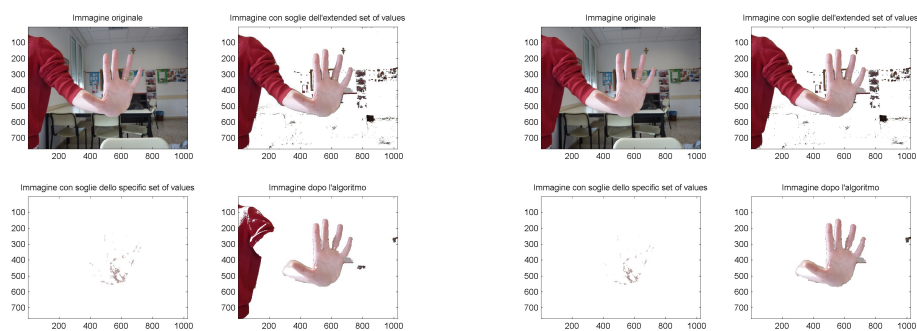
(f) L'uso della distanza tra colori permette di rimuovere gran parte della recinzione.

**Figura 4.1:** Esempi di applicazione





**Figura 4.2:** In questi due esempi si può notare come nella stessa scena ma con tipi di illuminazione differenti i risultati dell'algoritmo possono essere molto diversi.



(a) Filtro gaussiano 9x9, con deviazione standard 1.75 e  $w_{min}=0.512$

(b) Filtro gaussiano 5x5, con deviazione standard 1.75 e  $w_{min}=0.669$

**Figura 4.3:** Usando la stessa deviazione standard, filtri di diverse dimensioni e  $w_{min}$  differenti portano a risultati molto diversi: in questo caso scegliendo un filtro più piccolo e aumentando la soglia viene completamente eliminato il maglione. In entrambe le immagini la distanza tra colori permette di rimuovere tutti gli oggetti attorno alla mano.



## Capitolo 5

# Conclusioni

I risultati forniti dall'algoritmo sono buoni, ma la loro validità potrà essere verificata definitivamente solo in futuro, quando lo skin detection sarà integrato con le altre parti del progetto introdotto brevemente nel capitolo 1 di questo elaborato.

I metodi principali sfruttati dall'algoritmo, che ne costituiscono il punto di forza, sono la sogliatura ad isteresi, l'utilizzo della distanza tra colori e l'edge segmentation. Richiedono tutti alcuni parametri: se il tipo di applicazione consentisse di imporre ulteriori ipotesi di lavoro sarebbe quindi possibile, agendo su quest'ultimi, migliorare notevolmente lo skin detection. Non vi è ancora tuttavia un metodo automatico per la stima di questi parametri: per il momento sono solo stati trovati empiricamente (la ricerca di quest'ultimo potrebbe essere un interessante lavoro futuro).

Se oltre alla normale videocamera RGB si avessero anche a disposizione altre risorse hardware, l'algoritmo spiegato in questo elaborato potrebbe essere il punto di partenza per altre ricerche con l'obiettivo di rendere sempre più preciso lo skin detection. Avendo ad esempio a disposizione i dati di un sensore Time Of Flight si potrebbe rendere più precisa l'edge segmentation sfruttata dall'algoritmo e al tempo stesso usare la distanza tra colori per migliorare le informazioni di profondità (intuitivamente, ci sarà presumibilmente una variazione di colore sull'edge di due oggetti sovrapposti).



## Appendice A

# Possibili estensioni del metodo proposto

Vengono ora presentate alcune soluzioni che sono state escluse dalla versione finale dell'algoritmo in quanto richiedono ipotesi di lavoro molto restrittive.

### A.1 Limite al numero di espansioni da ogni pixel dello SSV

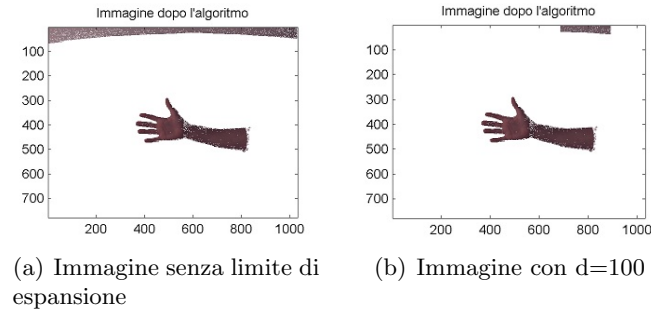
Un metodo preso in considerazione in fase di progettazione dell'algoritmo consisteva nel limitare il numero di espansioni consentite a partire dai pixel dello specific set of values.

Partendo da questi pixel l'algoritmo prima agisce sui pixel distanti 1, poi su quelli distanti 2 e così via (non si deve intendere come distanza euclidea, bensì distanza in numero di espansioni necessarie per arrivare da un pixel all'altro: i pixel a distanza 1, ad esempio, sono i pixel 8-connected). Se si limita la distanza fino a cui si può spingere la procedura, evitando così che un solo pixel nello specific set of values possa far includere nell'immagine finale ampie regioni dell'extended set of values, i risultati che si ottengono possono talvolta essere buoni, altre volte meno.

Ecco ad esempio cosa succede nell'immagine in figura A.1 mettendo come distanza massima  $d=100$ .

Da un ingrandimento dell'immagine dei valori dello specific set of values (qui non presente) si nota che vi è un gruppo di pixel (abbastanza denso da non essere stato eliminato), centrato attorno all'ascissa 800, che è l'unico responsabile nella figura a sinistra dell'inclusione dell'intera fascia superiore. In figura A.1b si è invece parzialmente riusciti a limitare il numero di pixel inclusi.

Questo metodo tuttavia non è stato incluso nella versione finale dell'algoritmo in quanto la scelta del parametro  $d$  dipende dal tipo di immagine che si sta analizzando. Trovare una regola generale che non porti ad un peggioramento dei risultati è impossibile, in quanto dipende molto da come sono concentrati gli skin pixel dello specific set of values, cosa non prevedibile. In ogni caso per immagini come ad esempio precedenti (immagini 1032x778) in cui ci sono porzioni di pelle abbastanza ampie sarebbe opportuno per  $d$  un valore non inferiore al centinaio di pixel (sebbene

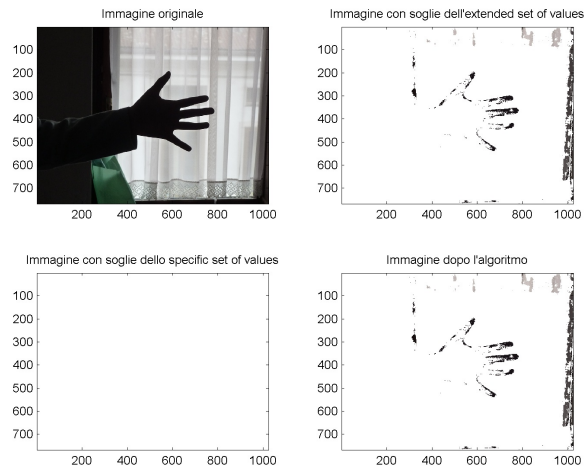


**Figura A.1:** Esempio di applicazione del limite al numero di espansioni

in molti casi ne basterebbero qualche decina), in quanto può accadere che i pixel dello specific set of values siano concentrati ad esempio solo sul bordo superiore della mano, rischiando di non riuscire a completarla tutta. Certamente comunque  $d$  deve crescere proporzionalmente all'estensione delle porzioni di pelle.

## A.2 Gestione delle immagini scure

Per alcune (e fortunatamente poche) immagini molto scure, per cui nemmeno l'occhio umano riesce a distinguere il rosa della mano, i valori delle tre componenti dello spazio di colore dei pixel differiscono molto dai valori comunemente assunti dagli skin pixel, senza seguire nessuna regola che possa permettere di applicare l'algoritmo. Lo specific set of values risulta quindi vuoto o con pochissimi elementi che potrebbero più o meno con la stessa probabilità essere o non essere skin pixel, e anche l'extended set non dà alcun riferimento certo.



**Figura A.2:** Esempio di applicazione dell'algoritmo ad un'immagine scura

Tutto questo porta ad un sicuro insuccesso dell'algoritmo (cioè non verrebbe preso nemmeno un pixel) in quanto non vi sono pixel da cui partire per fare la fase

di espansione o, se ci sono, sono pochi, forse nemmeno skin pixel, e con un'extended set of values che non da alcuna informazione sicura. Per gestire al meglio questa evenienza, nel caso in cui il numero di pixel dello ssv sia al di sotto di una certa soglia, ad esempio 15, la fase di espansione potrebbe non iniziare, e si potrebbe fare in modo di prendere tutti i pixel in un certo intervallo di valori molto grande.

Questo tipo di approccio può però ritenersi valido soltanto nel caso in cui si sia sicuri della presenza della pelle nell'immagine, ed è per questo che è stato escluso dall'algoritmo, in quanto la pelle potrebbe non esserci e quindi lo ssv essere giustamente vuoto: nel caso di immagini senza porzioni di pelle è giusto che l'algoritmo non restituisca alcun pixel.

Nell'immagine in figura A.2 sono stati inclusi nell'immagine finale i valori dell'extended set of values. Soltanto qualche volta questo intervallo funziona abbastanza bene con le immagini scure, ma comunque un'analisi dettagliata non è stata fatta in quanto si è deciso di non usare questo metodo nell'algoritmo.





# Bibliografia

- [1] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, *A survey of skin-color modeling and detection methods*, Pattern Recognition, 2007, pp. 1106-1122.
- [2] A. Elgammal, C. Muang, D. Hu, *Skin Detection-A Short Tutorial*, Encyclopedia of Biometrics R. U. Department of Computer Science, 2009.
- [3] G. Sharma, W. Wu, E. N. Dalal *The CIEDE2000 Color-Difference Formula: Implementation Notes, Supplementary Test Data, and Mathematical Observations*, Color Research and Application, vol. 30. No. 1, February 2005.
- [4] J. Canny, *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679-698, 1986.