

Pontificia Universidad Católica de Chile

FUNDAMENTOS DE PROCESAMIENTO DE
IMAGENES

IIC/IEE-3713

Tarea 1

Detección de la piel humana en una imagen digital

Ralf Wilke

ralf@ralfwilke.com

Numero alumno 08410038

Estudiante de intercambio de la RWTH Aachen, Universidad de Aachen, Alemania

Introducción

Durante los últimos años, la computación y manipulación de imágenes con computadores se ha desarrollado mucho. En casi todas las aplicaciones dedicadas a imágenes digitales hay un software para mejorar, manipular o comprimir la imagen.

Detecatar la piel humana es una tarea con muchas aplicaciones. Un uso puede ser el caso de supervisión con una cámara de video. Si la cámara puede detectar si hay personas al frente a la lente, ella puede sacar una foto solemante en este caso. Con esta técnica, la cantidad de datos que se debe transmitir o memorizar puede ser reducidio. Esto significa que se puede reducir los gastos de explotación también. Otra aplicación del sector de las cámaras es la detección de sonrisa en las cámaras fotográficas. El algoritmo, para detectar la sonrisa, tiene que saber primero donde están las caras. Después puede buscar la boca y los labios. Para detectar la cara, la piel humana es un buen indicador, porque normalemte las personas al frente de la cámara llevan ropa y la cara es una parte de la imagen sin ropa. Pero existe también una aplicación de la detección de la piel humana en el sector, donde la gente no lleva ropa. En internet, hay muchas imágenes de pornografía [1]. Cuando los niños navegan por internet, ellos pueden ver estas imágenes fácilmente y esto puede ser dañino para su desarrollo. Sería mejor que, cuando niños navegan por internet, los pardes los observen. Pero en realidad, esto sucede pocas veces. Las imágenes de pornografía contienen normalmente un gran porcentaje de piel humana. Con un detector incluido en el navegador de internet, se puede poteger a los niños de estos contenidos.

Marco Teórico

La tarea es detectar los pixeles en una imagen que pertenezcan a la piel humana. El método de detección sólo puede analizar pixel por pixel. Por lo tanto, el algoritmo no puede usar informacion de los pixeles en el ambiente para decidir si es piel humana o no. Con esta restricción, es más difícil de detectar la piel humana, porque normalmente muchos pixeles forman un área en la imagen que contiene piel humana. Se puede disminuir la cuota de detecciones falsas con una comprobación del tamaño del área.

Muchos de los algoritmos conocidos hasta ahora, usan métodos que no funcionan pixel por pixel. Por ejemplo, en [2] se usan imágenes de muestra de caras y autos para que un algoritmo detecte la forma de objetos 3D con un método estadístico. Un algoritmo muy famoso es [3]. Este usa una cascada de clasificadores. La complejidad de los clasificadores sube con cada uno y los anteriores desechan lugares en que no hay caras. Con este método, los clasificadores complejos no tienen que buscar en toda la imagen, solamete en las áreas donde la probabilidad de la existencia de una cara es alta. Una aplicación usando este algoritmo es la detección y corrección automática de ojos rojos, que es depresentado en [8]. Otros métodos de la detección están descritos en [5], [6] y [7].

Para saber cómo detectar los pixeles en esta tarea, es importante saber los diferentes espacios de color. El espacio muy usado en el mundo de los computadores es el espacio RGB que significa Red-Green-Blue. Estos son los tres colores fundamentales. Al mezclarlos, se pueden construir todos los colores. Los valores de los colores pueden tener 256 niveles, representados con un byte para cada color, con números de 0 a 255. Entonces, cada pixel necesita 3 bytes. (0,0,0) corresponde al negro y (255,255,255) al blanco.

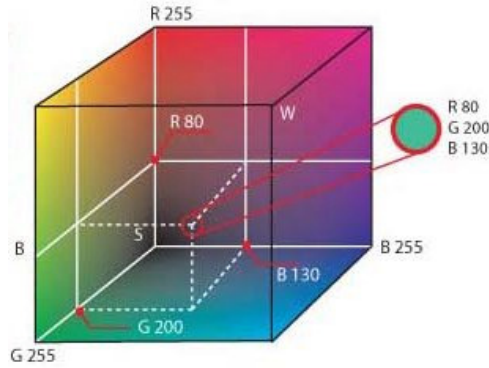


Figura 1: Cubo del espacio RGB. Fuente: Wikipedia

Para detectar la piel humana, buscar por el color de la piel humana es una posibilidad. Pero no es bueno hacerlo en el espacio RGB, porque con una iluminación variante, los lugares en el cubo de RGB cambian también. Una mejor forma es la transformación de la imagen en el espacio HSV, que tiene los componentes Hue, Saturation y Value. El componente Hue contiene la información sobre el color, el componente S contiene la saturación y el V corresponde a la luminosidad.

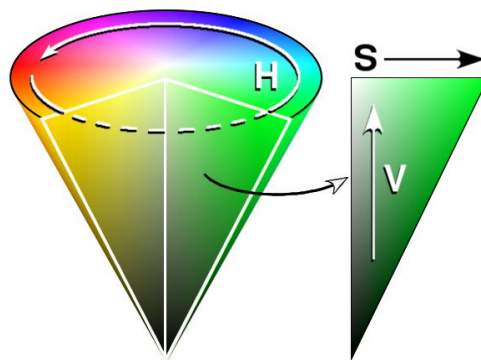


Figura 2: Cono del espacio HSV. Fuente: Wikipedia

La transformación de RGB en HSV está determinada por las siguientes fórmulas:

$$\begin{aligned}
 R, G, B &\in [0,1]; \quad MAX = \max(R, G, B); \quad MIN = \min(R, G, B) \\
 H &:= \begin{cases} 0^\circ & , \text{ si } MAX = MIN \Leftrightarrow R = G = B \\ 60^\circ \cdot \left(0 + \frac{G - B}{MAX - MIN}\right), & \text{ si } MAX = R \\ 60^\circ \cdot \left(2 + \frac{B - R}{MAX - MIN}\right), & \text{ si } MAX = G \\ 60^\circ \cdot \left(4 + \frac{R - G}{MAX - MIN}\right), & \text{ si } MAX = B \end{cases} \\
 \text{Si } H < 0^\circ : H &:= H + 360^\circ \\
 S &:= \begin{cases} 0 & , \text{ si } MAX = 0 \Leftrightarrow R = G = B = 0 \\ \frac{MAX - MIN}{MAX}, & \text{ otros casos} \end{cases} \\
 V &:= MAX
 \end{aligned}$$

En la presentación de Powerpoint [4] se exponen los intervalos del espacio HSV entre los cuales están los colores de la piel humana. Ellos son :

$$H \in ([0,25] \cup [355,360]), \quad S \in [0.4,0.6], \quad V \in [0.4,1]$$

La decisión de los pixeles puede ser la de comprobar si el valor HSV está dentro de este espacio. Si es así, este pixel es de la piel.

Otro método es el de coleccionar los pixeles de piel conocido por una referencia y calcular una distribución de la probabilidad que un pixel con un valor HSV es de la piel.

Diseño

Mi solución constiste en dos métodos. El primero, es una selección de pixeles de piel con los intervalos de [4]. Es rápido y tiene buenos resultados, que están descritos más tarde. El segundo, trabaja con la distribución de la probabilidad que viene de una investigación anterior de imágenes de referencia. Este algoritmo funciona así:

Después la conversión de la imagen original del espacio RGB al espacio HSV, la máscara de la imagen de referencia está aplicado para seleccionar solamente los pixeles que representan la piel humana. Si la máscara tiene solamente los valores 0 y 1, la selección es posible con una multiplicación por cada pixel. La imagen del resultado posterior es negro en todos los pixeles que no son piel. En los otros pixeles, la información HSV sobrevive. Después busco estos pixeles, agregándolos en un vector. Puedo combinar los vectores de múltiples imágenes para considerar un cantidad más grande. Con un histograma puedo ver cual es la distribución que tienen los valores de la piel. Si la cantidad de una investigación es suficientemente grande, la distribución se aproxima a la probabilidad. Después de una normalización de los valores es necesario satisfacer la condición para los tres componentes H, S y V

$$\int_{-\infty}^{\infty} p(x)dx = 1$$

Durante el estadio de la detección para cada pixel se puede clasificar con una probabilidad que es o no piel humana para cada componente. Para combinar estas tres probabilidades es necesario multiplicarlas. Si $P(X)$ es la probabilidad del evento X y $P(Y)$ es la del evento Y, la probabilidad que el evento X e Y ocurran al mismo tiempo $P(X \text{ y } Y)$ es $P(X) * P(Y)$. Entonces, el resultado es una imagen de la probabilidad para cada pixel si es piel. Usando un nivel determinado se puede elegir finalmente cual pixel representa la piel humana.

Experimentos y resultados

Intervalos fijos

Primero, voy a presentar el experimento con los intervalos fijos de [4]. El algoritmo es fácil y selecciona sucesivamente cada componente del espacio HSV. Después sigue la combinación para obtener la imagen binaria, en la cual los píxeles que están detectados como piel tienen un color blanco que corresponde al valor 1. Los otros píxeles tienen el valor 0, que significa negro. La comparación con la imagen de referencia da la información de cuántos píxeles están mal detectados. La tabla lógica para evaluar eso es:

Detección	Referencia	Bien Seleccionado	Falso positivo	Omitido
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

Tabla 1: Tabla lógica para la evaluación del algoritmo

El programa tiene la capacidad para calcular estos números. Los resultados están expuestos en la tabla posterior. La evaluación contiene algunas imprecisiones, porque en las imágenes de referencia están marcadas solamente las caras. Pero esta detección sirve para piel y no para caras. Entonces, las imágenes están cortadas para no tener los brazos.

Número de la imagen	altitud de la imagen cortada	píxeles bien seleccionados	píxeles falsos positivos	píxeles omitidos
1	650	1129512 (93.63%)	40055 (3.32%)	36833 (3.05%)
2	770	1319135 (92.30%)	65319 (4.57%)	44666 (3.13%)
3	770	1361275 (95.25%)	46363 (3.24%)	21484 (1.50%)
4	670	1177762 (94.71%)	35793 (2.88%)	29965 (2.41%)
5	750	1322738 (95.02%)	31128 (2.24%)	38134 (2.74%)
6	850	1408581 (94.87%)	43975 (2.96%)	29832 (2.01%)
7	800	1389367 (93.57%)	57605 (3.88%)	37828 (2.55%)
Promedio		94.20%	3.29%	2.48%

Tabla 2: Resultados del método de intervalos fijos



Figura 3: Resultado del método de intervalos fijos; las detecciones correctas son blancas

Imagen de Probabilidad

El segundo método trabaja con la distribución de la probabilidad. Después de la lectura de los datos del conteo anterior, formo un vector con todos los valores HSV de las imágenes 1 a 6. Uso la imagen con número 7 para evaluar el algoritmo. Se puede elegir una cantidad en la cuantificación del histograma. Pienso que 1000 es un valor bueno. La detección empieza con la lectura de la imagen original y la transformación en el espacio HSV. Los valores HSV de los píxeles de la piel están divididos en 1000 niveles para el histograma. Después hay una estandarización de la distribución para transformarla en una distribución de la probabilidad. Para el procesamiento de la imagen original sigue el método de una Look-Up-Table. Con el valor de cada uno de los píxeles, se calcula el índice que corresponde en el histograma. Para las convenciones de MATLAB, es necesario poner atención a los valores del índice. Pixel por pixel y por cada componente del espacio HSV, comienza la formación de la nueva imagen con valores de probabilidad. Cuando este proceso está terminado, junto la probabilidad de los 3 componentes con la multiplicación. Los valores del resultado son muy pequeños por la multiplicación de tres probabilidades. Al final, uso un nivel determinado para seleccionar los píxeles que están en la piel humana. El mismo algoritmo es usado para evaluar los resultados que sirve también antes.

Experimento 1: A base de la imagen 1

En este experimento uso solamente la información de la imagen 1 para clasificar los valores HSV.

Número de la imagen	altura de la imagen cortada	píxeles bien seleccionados	píxeles falsos positivos	píxeles omitidos
1	650	1070446 (88.73%)	16032 (1.33%)	119922 (9.94%)
2	770	1297724 (90.81%)	23724 (1.66%)	107672 (10.7%)
3	770	1322162 (92.52%)	19726 (1.38%)	87232 (6.10%)
4	670	1148097 (92.33%)	14707 (1.18%)	80716 (6.49%)
5	750	1288800 (92.59%)	11170 (0.80%)	92030 (6.61%)

6	850	1466525 (92.96%)	19782 (1.25%)	91293 (5.79%)
7	800	1338132 (90.12%)	24261 (1.63%)	122407 (8.24%)
Promedio		91.44%	1.18%	7.6957%

Tabla 3: Resultados del método con la probabilidad a base de la imagen 1



Figura 4: Resultado del método con la probabilidad a base de la imagen 1; las detecciones correctas están blancas

Experimento 2: A base de las imágenes de 1 a 6

En este experimento uso toda la información de las imágenes 1 a 6 para clasificar los valores HSV.

Número de la imagen	altitud de la imagen cortada	pixeles bien seleccionados	pixeles falsos positivos	pixeles omitidos
1	650	1070297 (88.72%)	13910 (1.15%)	122193 (10.1%)
2	770	1307594 (91.50%)	26045 (1.82%)	95481 (6.68%)
3	770	1324977 (92.71%)	19856 (1.39%)	84287 (5.90%)
4	670	1155312 (92.91%)	15649 (1.26%)	72559 (5.83%)
5	750	1297691 (93.22%)	16229 (1.17%)	78080 (5.61%)
6	850	1472619 (93.35%)	20860 (1.32%)	84121 (5.33%)
7	800	1352258 (91.07%)	24857 (1.67%)	107685 (7.25%)
Promedio		92.02%	1.40%	6.67%

Tabla 4: Resultados del método con la probabilidad a base de las imágenes de 1 a 6.

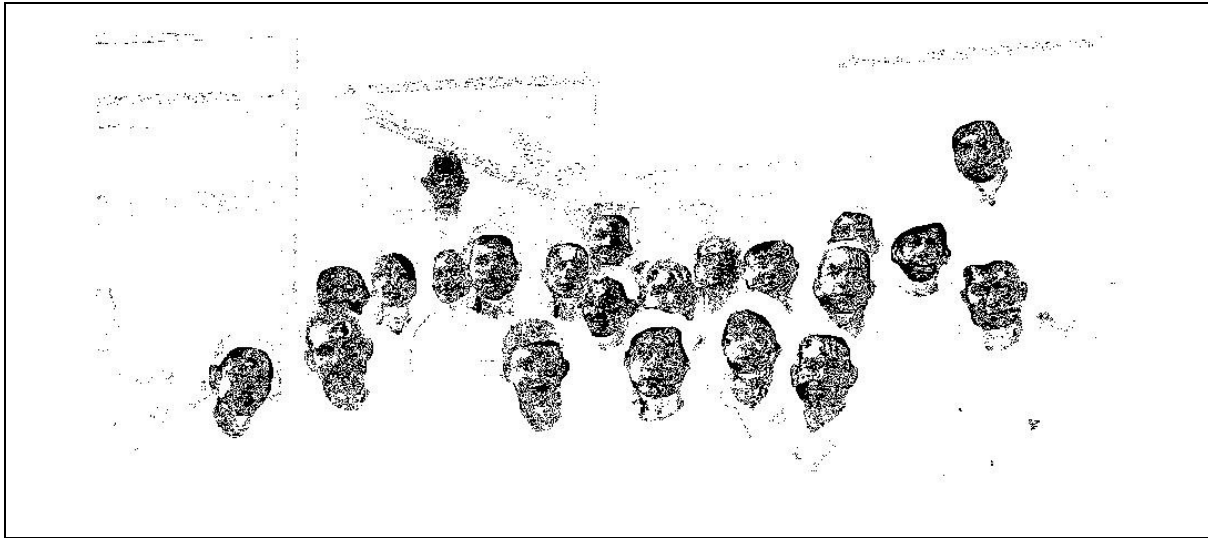


Figura 5: Resultado del método con la probabilidad a base de las imágenes de 1 a 6; las detecciones correctas están blancas

Conclusiones

A través de la percepción y también como lo demuestran los resultados numéricos, se puede apreciar que el método con los intervalos fijos tiene mejores porcentajes en la detección correcta. El promedio es de 94% de píxeles, por lo cual el algoritmo se ha elegido bien. Son solamente 3.29% de falsos positivos y aproximadamente 2.5% de píxeles omitidos.

Los resultados del método con la probabilidad son más o menos los mismos. Un porcentaje de 91% (92%) para la detección correcta, 1.18% (1.40%) para los falsos positivos y 7.7% (6.67%) para los píxeles omitidos.

Los valores entre paréntesis representan a los resultados de la investigación en base a las imágenes de 1 a 6.

Es necesario dar cuenta que las imágenes de referencia contienen solamente las áreas de las caras y no contienen otros píxeles de la piel. Por eso, la evaluación de los píxeles falsos positivos tienen en realidad un valor más pequeño.

Generalmente, los experimentos muestran que la detección con los intervalos fijos tiene buenos resultados y que estos son mejores que los del método con la distribución de la probabilidad. Además, es más fácil de buscar por píxeles dentro de algunos intervalos que usar la distribución de la probabilidad. Más allá de esto, es necesario investigar antes la distribución que es innecesario con el otro algoritmo. Entonces, recomiendo el método de los intervalos fijos.

Bibliografía

- [1] Bärbel Geisendrees: Tesina de la Universidad Paderborn, Alemania, Methoden zur Erkennung von Hautfarbe in Farbbildern, 09/2001
- [2] Henry Schneidermann, Takeo Kanade: A Statistical Method for 3D Object Detection Applied to Faces and Cars, Robotics Institute, Carnegie Mellon University, Pittsburgh
- [3] Paul Viola, Michael Jones: Rapid Object Detection using a Boosted Cascade of Simple Features, Mitsubishi Electric Research Labs, Compaq CRL, Cambridge
- [4] Klaus Broelemann, Florian Daiber, Monika Nientiedt: Automatic detection and correction of (partially occluded) red-eyes, Powerpoint Presentation, <http://cvpr.uni-muenster.de/teaching/ws04/seminarWS04/downloads/RedEyes-Folien.pdf>
- [5] Henry A. Rowley: Neural Network-Based Face Detection, School of Computer Science, Carnegie Mellon University, Pittsburgh, May 1999
- [6] Ming-Hsuan Yang, David J. Kriegman, Narendra Ahuja: Detecting Faces in Images: A Survey, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 1, January 2002
- [7] Henry A. Rowley, Shumeet Baluja, Takeo Kanade: Rotation Invariant Neural Network-Based Face Detection, School of Computer Science, Carnegie Mellon University, Pittsburgh, Justsystem Pittsburgh Research Center, Pittsburgh
- [8] Matthew Gaubatz, Robert Ulichney: Automatic Red-Eye Detection and Correction, Cornell University, Hewlett-Packard Co.

Anexo A: Manual de uso del programa

Hay tres programas en total. Un archivo *interval.m* hace la detección con los intervalos fijos, el archivo *censo.m* hace la investigación de la distribución de la probabilidad y el programa que se llama *probabilidad.m* es el detector de la piel humana a base de la probabilidad.

Los dos detectores tienen 3 parámetros que se puede usar. La imagen para detectar la piel se puede elegir con la variable `inputfile`. El nombre del archivo para la imagen de resultado es `ouputfile`. Para la evaluación con una imagen de referencia sirve la variable `reffile`.

El programa que se llama *censo.m* es solamente para usarlo en el estadio de la creación de la distribución de valores del píxel en el espacio HSV. Los resultados son memorizados en el archivo *piel_todo.mat*. Por eso no es necesario usar el programa *censo.m* para detectar piel humana.

Anexo B: Los Programas

Interval.m

```
% FUNDAMENTOS DE PROCESAMIENTO DE PROCESAMIENTO DE IMAGENES IIC/IEE-3713
%
% Tarea 1
% Ralf Wilke ralf@ralfwilke.com
% No Alumno 08410038
%
% 14.09.2009
%
% Seleccion con intervalos fijas

close all;

inputfile = './caras/img07.jpg';
reffile = './caras/ref07.png';
outputfile = './caras/piel07_interval.jpg';

% lectura del imagen orginial
imrgb = imread(inputfile);
imshow(imrgb);
imhsv=rgb2hsv(imrgb);

% Calculacion del pixeles que son entre el interval para H
piel1 = (imhsv(:,:,1)<=(25/360));
piel2 = (imhsv(:,:,1)>=(335/360));
pielH = or(piel1, piel2);

% Calculacion del pixeles que son entre el interval para S
piel3 = (imhsv(:,:,2))>0.6;
piel4 = (imhsv(:,:,2))<0.2;
pielS = not(or(piel3,piel4));

% Calculacion del pixeles que son entre el interval para V
pielV = (imhsv(:,:,3))>0.4;

% Combinacion
piel = and(pielH,pielS); piel = and(piel,pielV);
figure;imshow(piel);

% Examinacion
```

```

% lectura del imagen referencia
ref = imread(reffile);

% Evaluacion, solamente en la area arriba, los brazos no cuentan
im_bienseleccionado = not(xor(piel(1:800,:),ref(1:800,:)));
im_falsepositive = and(piel(1:800,:),not(ref(1:800,:)));
im_omitido = and(not(piel(1:800,:)),ref(1:800,:));

figure;imshow(im_bienseleccionado);
figure;imshow(im_falsepositive);
figure;imshow(im_omitido);

num_elementos = numel(im_bienseleccionado);

num_bienseleccionado = sum(sum(im_bienseleccionado))
rel_bienseleccionado = num_bienseleccionado / num_elementos

num_falsepositive = sum(sum(im_falsepositive))
rel_falsepositive = num_falsepositive / num_elementos

num_omitido = sum(sum(im_omitido))
rel_omitido = num_omitido / num_elementos

% Output
imwrite (piel,outputfile,'jpg');

```

Censo.m

```

% FUNDAMENTOS DE PROCESAMIENTO DE PROCESAMIENTO DE IMAGENES IIC/IEE-3713
%
% Tarea 1
% Ralf Wilke ralf@ralfwilke.com
% No Alumno 08410038
%
% 14.09.2009
%
% Censo de Data estadistico

inputfile = './caras/img06.jpg';
reffile = './caras/ref06.png';

% lectura del imagen orginial
close all;
imrgb = imread(inputfile);
imhsv=rgb2hsv(imrgb);

% lectura del imagen de referencia
imref = imread(reffile);

% Estandarizacion de los valores en la imagen de referencia
imrefnorm = double(imref >= 1);

% Separación de canales y selección de lugares, que tienen cajas
h_caja = imhsv(:,:,1) .* double(imrefnorm);
s_caja = imhsv(:,:,2) .* double(imrefnorm);
v_caja = imhsv(:,:,3) .* double(imrefnorm);

% Combinacion de canales
imcarahsv = imhsv;
imcarahsv(:,:,1) = h_caja;
imcarahsv(:,:,2) = s_caja;
imcarahsv(:,:,3) = v_caja;

%
figure;imshow (hsv2rgb(imcarahsv));

% Buscar por pixeles que no son blanco, agregacion de ellos en un vector

piel6 = [0,0,0];
for x = 1:1392
    for y = 1:1856

```

```

        if (imcarahsv(x,y,1) > 0)
            piel6 = [piel6; [imcarahsv(x,y,1), imcarahsv(x,y,2),imcarahsv(x,y,3)]];
        end
    end
end
end

```

Probabilidad.m

```

% FUNDAMENTOS DE PROCESAMIENTO DE PROCESAMIENTO DE IMAGENES IIC/IEE-3713
%
% Tarea 1
% Ralf Wilke ralf@ralfwilke.com
% No Alumno 08410038
%
% 14.09.2009
%
% Selecion con la distribucion de la probabilidad

% Quatizacion
quant = 1000;

%Threshold
threshold = 0.9e-7;

close all;

inputfile = './caras/img07.jpg';
reffile = './caras/ref07.png';
outputfile = './caras/piel_probabilidad07.jpg';

% lectura del datos
load ('./piel/piel_todo.mat');

% Juntar los datos en un vector
piel = piel1;
piel = [piel; piel2];
piel = [piel; piel3];
piel = [piel; piel4];
piel = [piel; piel5];
piel = [piel; piel6];

% Quantizar los valores double en 'quant'
piel_quant = piel * quant;

% Hacer un histograma para los componenter H, S y V
hist_H = hist(piel_quant(:,1),quant);
hist_S = hist(piel_quant(:,2),quant);
hist_V = hist(piel_quant(:,3),quant);

% Estandarizacion para hacer una distribucion de la probabilidad
hist_norm_H = hist_H ./ sum(hist_H);
hist_norm_S = hist_S ./ sum(hist_S);
hist_norm_V = hist_V ./ sum(hist_V);

% Estos vectores sirven ahora como una Look-Up-Table
figure;plot (hist_norm_H);
figure;plot (hist_norm_S);
figure;plot (hist_norm_V);

% lectura del imagen original
imrgb = imread(inputfile);
figure;imshow(imrgb);
imhsv=rgb2hsv(imrgb);

% Procesar todo el imagen
for x = 1:1392
    for y = 1:1856

```

```

index_H = round(imhsv(x,y,1)*quant)+1;
if (index_H > quant)
    index_H = quant;
end
improbab(x,y,1) = hist_norm_H(index_H);

index_S = round(imhsv(x,y,2)*quant)+1;
if (index_S > quant)
    index_S = quant;
end
improbab(x,y,2) = hist_norm_S(index_S);

index_V = round(imhsv(x,y,3)*quant)+1;
if (index_V > quant)
    index_V = quant;
end
improbab(x,y,3) = hist_norm_V(index_V);
end
end

% Calculacion de la probabilidad junta
improbab_junto = improbab(:, :, 1) .* improbab(:, :, 2) .* improbab(:, :, 3);

% Aplicacion de nivel detreminado
im_piel_final = improbab_junto > threshold;

% Monstra resultado
figure; imshow(im_piel_final);

% lectura del imagen referencia
ref = imread(reffile);

altitud = 800;

% Evaluacion, solamente en la area arriba, los brazos no cuentan
im_bienseleccionado = not(xor(im_piel_final(1:altitud,:), ref(1:altitud,:)));
im_falsepositive = and(im_piel_final(1:altitud,:), not(ref(1:altitud,:)));
im_omitido = and(not(im_piel_final(1:altitud,:)), ref(1:altitud,:));

figure; imshow(im_bienseleccionado);
figure; imshow(im_falsepositive);
figure; imshow(im_omitido);

num_elementos = numel(im_bienseleccionado);

num_bienseleccionado = sum(sum(im_bienseleccionado))
rel_bienseleccionado = num_bienseleccionado / num_elementos

num_falsepositive = sum(sum(im_falsepositive))
rel_falsepositive = num_falsepositive / num_elementos

num_omitido = sum(sum(im_omitido))
rel_omitido = num_omitido / num_elementos

%Output
imwrite (im_piel_final, outputfile, 'jpg');

```