

Pontificia Universidad Católica de Chile

FUNDAMENTOS DE PROCESAMIENTO DE
IMAGENES

IIC/IEE-3713

Tarea 2

Detección de los distintos tipos de superficie en
imágenes satelitales de la tierra

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.

La exploración de la tierra con satélites artificiales fue el mecanismo utilizado tradicionalmente durante los últimos 50 años y hasta el día de hoy sigue en constante desarrollo. Por ejemplo, los satélites para informaciones sobre el tiempo sacan imágenes que todo el mundo puede ver en la noche durante los noticias en la televisión. Otro tipo importante de satélites son los de espionaje, que contienen cámaras ópticas. Estos eran importantes durante la Guerra Fría y siguen teniendo influencia en la política de hoy. Pero hay también muchas aplicaciones en el mundo civil. Todas las ciencias que se dedican a la tierra, la geología, la naturaleza o el crecimiento de ciudades necesitan una fuente de datos para hacer sus investigaciones. Por ejemplo, la examinación de un terreno forestal dañado después de un temporal con mucho viento [6], [7]. La creación de mapas es otra aplicación que permite, junto con la geodesia, tener un conocimiento de la superficie de la tierra con una precisión que nunca se había visto antes.

La empresa Google Inc. con domicilio social en California, compró en 2004 la empresa Keyhole Corp. Después de renombrarla como *Google Earth*, este sistema permite la disponibilidad de descarga gratis del programa en la versión básica. Con este programa cada persona con conexión a internet puede ver imágenes satelitales de la tierra en el espectro de luz visible. Una gran parte de las regiones colonizadas está cubierta con una resolución de metros hasta algunos decímetros.

Marco Teórico

La tarea consiste en detectar los distintos tipos de superficie de la tierra en imágenes de Google Earth. Antes del procesamiento de las imágenes con un tamaño de 512 x 512 píxeles, deberán ser convertidas a blanco y negro. Esto excluye usar la información del color para discriminar las categorías. Por eso, el uso de la Transformada Fourier es el tema especial de esta tarea. Ella permite convertir una imagen del dominio del espacio al dominio de las frecuencias de los cambios de la luminosidad espacial. La fórmula para hacerlo en forma discreta es la siguiente:

$$F(m, n) = \sum_{i=0}^{M-1} \sum_{k=0}^{N-1} f_{i,k} \cdot \exp(-j \cdot 2\pi(\frac{m}{M}i + \frac{n}{N}k)) \in C$$

El resultado es una matriz con la misma dimensión que tiene la imagen original, pero los valores están complejos [1], [2]. En esta tarea voy a usar solamente el valor absoluto. El tiene la información de las frecuencias que están dentro de la imagen y su orientación. La detección de las distintas superficies de la tierra puede ser realizada con la clasificación de las frecuencias que tienen las características para ellos. Por ejemplo, en el desierto, la distribución de la luminosidad es casi la misma en áreas pequeñas. La arena no produce cambios rápidos. Por eso, espero para este tipo de superficie muchas frecuencias bajas y menos altas. El contrario, la ciudad, con sus calles y edificios, contiene muchos cambios rápidos que resultan en un espectro plano, tanto con frecuencias bajas como altas. Para un río espero un peak en la frecuencia característica del cambio campo-agua-campo. Examinando algunos ejemplos de cada superficie espero ser capaz

de investigar las características en el dominio de la frecuencia y después distinguir la base de ellas.

Uso para la clasificación dos aproximaciones. Lo que las dos tienen en común es la idea de usar primero una máscara para seleccionar un área pequeña de la imagen. Todo el procesamiento siguiente será hecho con esta parte de la imagen para sacar información sobre el pixel al medio. Cuando el procesamiento está listo, muevo la máscara por un pixel y repito el algoritmo. Con este método, examino paso a paso toda la imagen.

Los métodos se diferencian en la manera de la clasificación. La primera, calcula el coeficiente 2D de la correlación entre espectros de imágenes referenciales y la imagen para examinar. Se calcula el coeficiente 2D de la correlación de dos imágenes A y B con esta fórmula [7]:

$$c = \frac{\sum_{i=1}^m \sum_{j=1}^n [(A_{ij} - \bar{A}) \cdot (B_{ij} - \bar{B})]}{\sqrt{\left[\sum_{i=1}^m \sum_{j=1}^n (A_{ij} - \bar{A})^2 \right] \cdot \left[\sum_{i=1}^m \sum_{j=1}^n (B_{ij} - \bar{B})^2 \right]}}$$

El coeficiente puede tener un valor de -1 a $+1$. Un valor de 0 significa que las imágenes A y B son totalmente distintas. Si sale un valor de -1 o $+1$, las matrices A y B están correlacionadas perfectamente. Examinando con cual espectro muestral existe la correlación más alta, uno puede determinar cual tipo de superficie es.

La otra opción para clasificar es primero pasar la transformada 2D en una forma 1D, en que hay un promedio de las cantidades de los valores de las frecuencias, independiente de la orientación espacial. Después sigue una examinación de las características especiales de los tipos de superficie. Sobre la base de ellas es posible crear un clasificador que puede ser capaz de distinguir los patrones del grapho para seleccionar.

Diseño

Al inicio, se investigarán los distintos espectros de las superficies. Por eso he seleccionado 3 ejemplos para cada superficie y he calculado la Transformada Fourier de ellos. Los resultados son mostrados en los siguientes 6 gráficos:

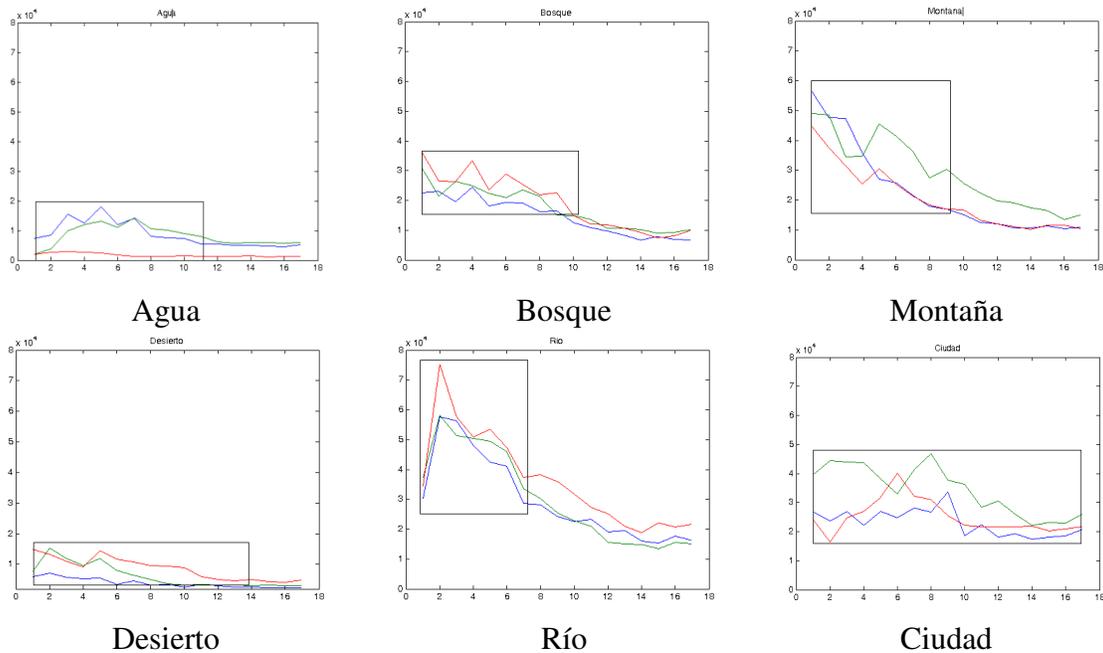


Figura 1: Los espectros de los tipos de la superficie de la tierra

Se puede ver que obviamente hay graphos distintos. Los rectángulos marcan las áreas características siguientes de esta investigación. En la tabla que se presenta a continuación se pueden apreciar los límites que componen los rectángulos.

Tipo	Frecuencia min	Frecuencia max	Valor min/(10 ⁴)	Valor max/(10 ⁴)
Agua	1	11	0	2
Bosque	1	10	1.5	3.8
Montaña	1	17	1.6	4.5
Desierto	1	14	0.2	1.8
Río	1	9	2.5	7.5
Cuidad	1	8	1.5	5.5

Tabla 1: Límites de las áreas en que se encuentran los valores característicos

Los algoritmos siguientes usan estas diferencias para distinguir.

Correlación en 2D con independencia de la orientación

Primero, el algoritmo determina para cada pixel un área en su medioambiente. Después hace la Transformada Fourier y calcula el valor absoluto del resultado complejo. Ahora lo que es más fácil es calcular el coeficiente de la correlación de la transformada con las muestras de la transformada de los distintos tipos de superficie. Pero es probable, que las muestras tengan otra orientación que la pequeña imagen para examinar. Para dar cuenta de ese caso, se rota el área de examinación en 360 pasos. Para cada orientación determino el coeficiente de la correlación 2D del espectro con cada muestra de superficie. Voy a mostrar un ejemplo usando las categorías ciudad y pacífico, porque ellas son bastante distintas.

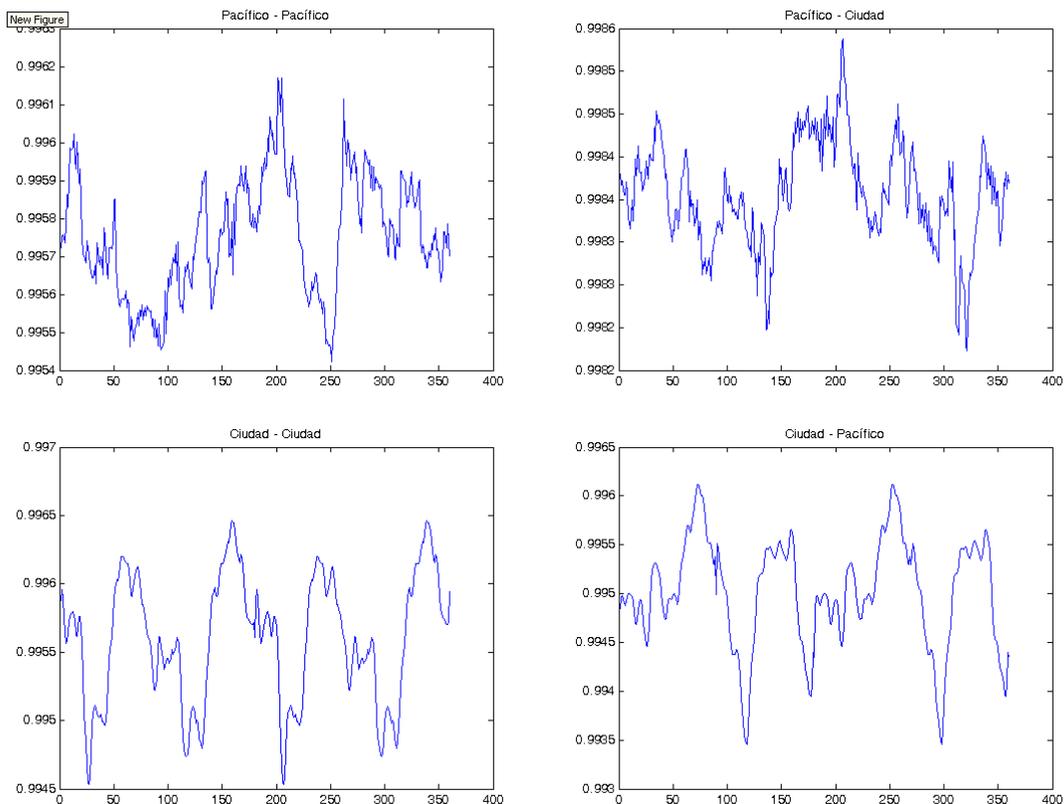


Figura 2: Graphos del coeficiente de la correlación 2D de la combinación con espectros del la ciudad y del Océano Pacífico

Uno puede ver que con todas las combinaciones el valor de la correlación es muy grande. El promedio de los valores cuadrados de cada grapho está en la siguiente tabla:

Patron\Imágen	Pacífico	Ciudad
Pacífico	0.9915	0.9901
Ciudad	0.9968	0.9912

Tabla 2: Promedio de los valores cuadrados del coeficiente de la correlación para cada ángulo

Lo que se observa es que los valores son muy similares y además, que el valor de Pacífico – Pacífico es menos que el valor de Pacífico – Ciudad que va a seguir en una detección falsa.

Pero existe un factor visible. Por la estructura regular de la ciudad, con sus calles cuadráticas, aparece una forma periódica con repetición cada 90° . En el caso del océano, esta periodicidad no es tan alta, pero también está presente. Por eso, mi idea es sacar el promedio y luego hacer con este grapho otra transformada de fourier. Si los patrones de los espectros comparados tienen una correlación grande, la transformada va a librar algunas frecuencias bajas con motivo de la periodicidad de 45° , 90° o 180° y pocos valores en las frecuencias altas. Si los patrones son distintos, el grapho del coeficiente por los ángulos tiene valores casi casuales. Por eso, la transformada de este ruido blanco va a dar un espectro plano o en otras palabras, también valores significantes en las frecuencias altas. Con un filtro de pasa alta voy a calcular la parte de las frecuencias altas y si hay menos, los patrones tienen correlación. Para visualizar voy a mostrar los graphos de las frecuencias.

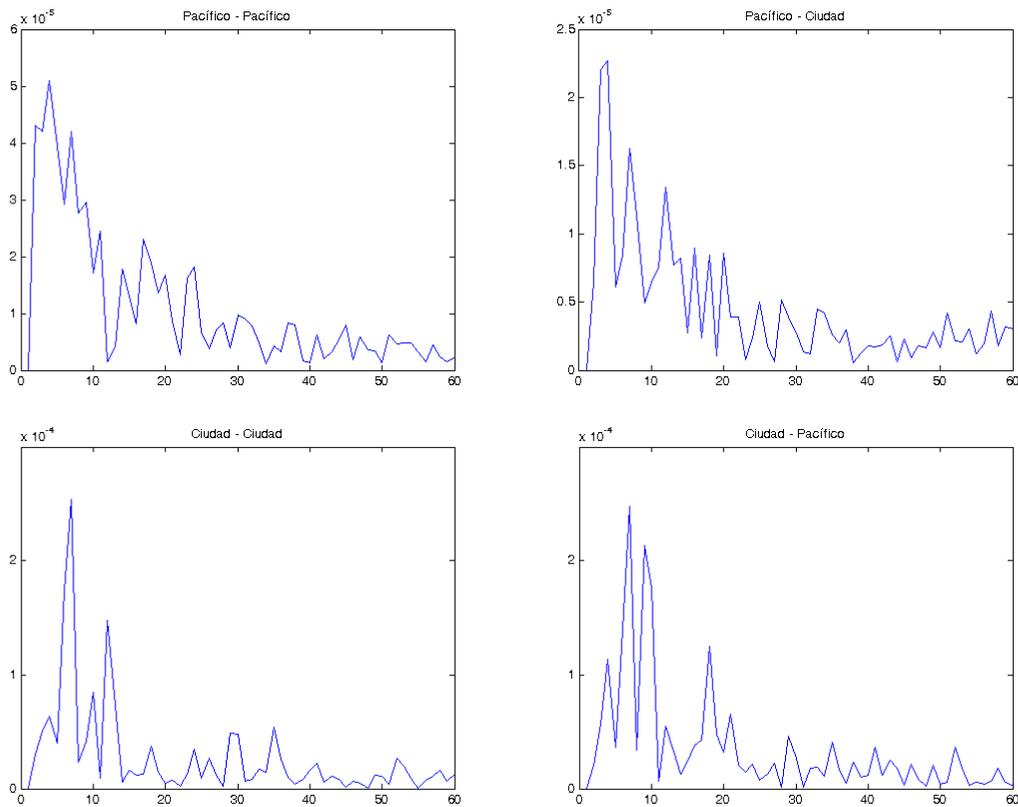


Figura 3: Graphos del espectro de la distribución del coeficiente de la correlación 2D

Excluyendo las primeras 8 frecuencias, la suma normalizada de las frecuencias que permanecen está mostrada en la tabla siguiente.

Patrón\Imágen	Pacífico	Ciudad
Pacífico	0.4760	0.4678
Ciudad	0.5126	0.4390

Tabla 3: Promedio de los valores después del filtro pasa alta normalizados

Entonces ahora tenemos un mínimo para patrones iguales, pero la diferencia no está tan acusada. Además, con un computador moderno el algoritmo se demora demasiado en procesar los datos

requeridos para la identificación de la áreas de estudio. Para una imagen de 512 por 512 pixeles este algoritmo va a necesitar 182 días. Entonces, este método puede servir, pero el gasto es grande.

Análisis del espectro en 1D y clasificación a base de espectros referenciales

Para tener otra manera de hacer la clasificación, escribí un programa que produjo las imágenes en el párrafo *Diseño*. Además, este programa es capaz de calcular el promedio de los graphos para cada tipo de superficie. Después, la distribución de la frecuencia es memorizada en un archivo de MATLAB. Esa es la base de conocimiento para el detector. La transición del espectro 2D a una dimensión está calculado por la suma de los valores, que están en rectángulos concéntricos en la plana aplazada de la frecuencia. El foco de los rectángulos está también en centro de la plana frecuencia. Con ese método, estoy sumando los valores de los distintas frecuencias, independiente de su orientación en cuanto al espacio.

El detector está usando la base de la distribución de las frecuencias para calcular la diferencia entre el espectro a examinar y los espectros característicos de los tipos de superficie. La categoría detectada tiene un mínimo de la diferencia entre los dos graphos. Esta examinación es rápida y con un computador moderno necesita menos de 2 milisegundos.

Los datos de las características contienen también la información de la frecuencia 0, que es el promedio de la luminosidad. Con esta información la diferenciación entre los tipos *Agua* y *Desierto* es posible, porque el desierto está más claro que el Océano Pacífico.

Experimentos y resultados

Correlación en 2D con independencia de la orientación

Para examinar el algoritmo con la correlación, escribí un programa que sirve como un clasificador para imágenes de tamaño pequeño junto con la idea de la máscara, moviéndola sobre toda la imagen a detectar. También en este estado se puede ver, si la detección de las superficies funciona o no. Para los experimentos, usé algunas imágenes preseleccionadas para la investigación principal de la distribución en el dominio de la frecuencia.

Patrón\Imágen	Pacífico	Bosque	Montaña	Desierto	Río	Ciudad
Pacífico	0.5581	0.5574	0.5780	0.5865	0.5195	0.5352
Bosque	0.5802	0.5500	0.5829	0.5504	0.5179	0.5266
Montaña	0.5789	0.5880	0.5528	0.5663	0.5441	0.5380
Desierto	0.5861	0.5742	0.5798	0.5953	0.5170	0.5225
Río	0.5846	0.5866	0.5782	0.5623	0.5109	0.5276
Ciudad	0.5826	0.5585	0.5789	0.5769	0.5193	0.5207
Detección correcto	Sí	Sí	Sí	No (Bosque)	Sí	Sí

Tabla 4: Resultados de la clasificación con el método de la correlación

Para cada examinación el computador necesitó aproximadamente un minuto. Aunque las detecciones están bastante bien no es utilizable para examinar una imagen de tamaño 512 x 512.

Análisis del espectro en 1D y clasificación a base de espectros referenciales

Los resultados con este algoritmo están bastante bien. Para mostrar siguen algunos ejemplos. Cada color usado en estas imágenes significa una superficie. Lo anterior está mostrado en la tabla siguiente. Los bordes negros son del tamaño de la máscara para examinar el pixel y su medioambiente.

Superficie	Color
Pacífico	Amarelo
Bosque	Verde
Montaña	Ciano
Desierto	Azul
Río	Magenta
Ciudad	Verde
Borde de la imagen	Negro

Tabla 5: Asignación de los colores usado en la imagen de los resultados

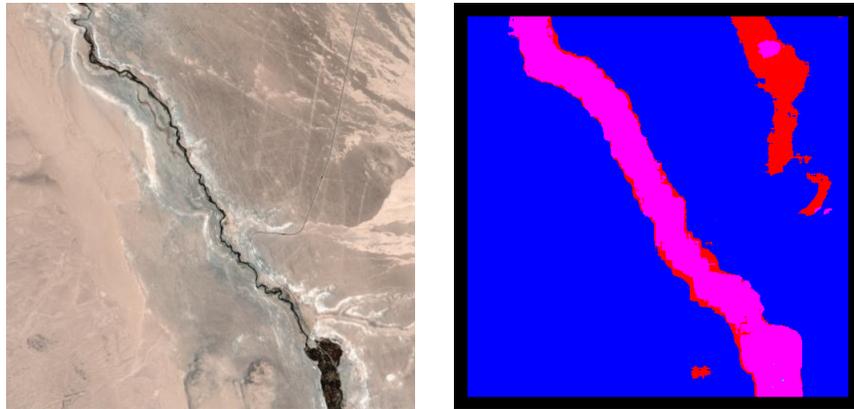


Figura 4: Desierto y Río

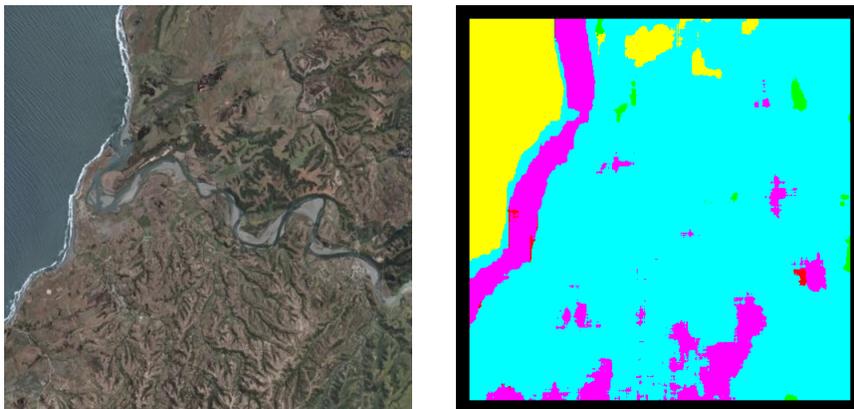


Figura 5: Océano, Río y Montaña

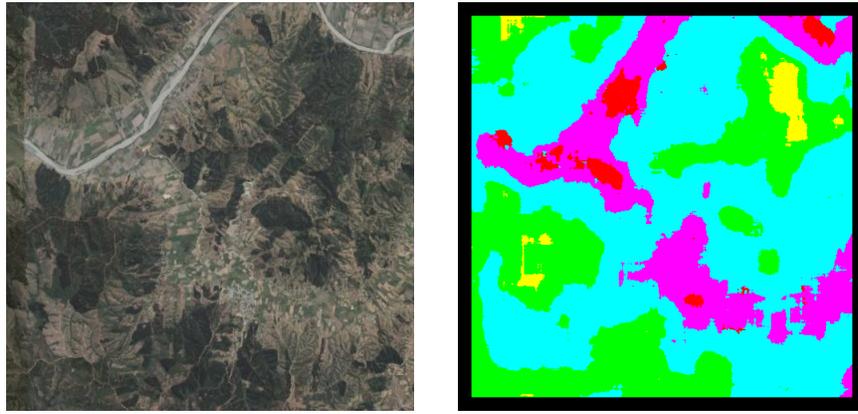


Figura 6: Montaña, Río y Bosque

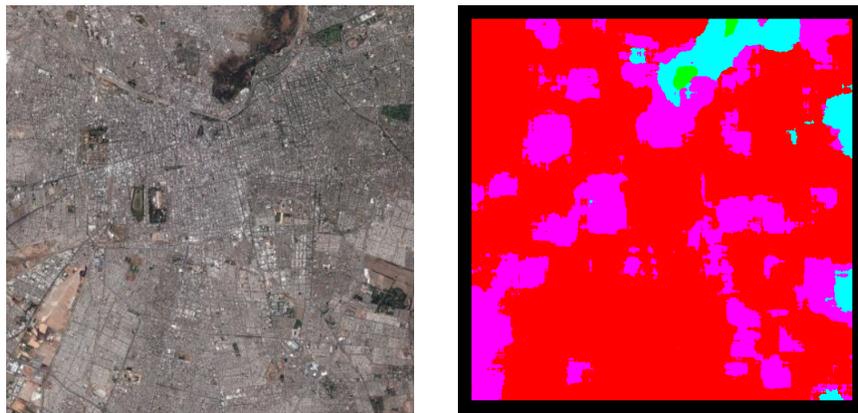


Figura 7: Santiago de Chile, Cerro San Cristóbal

Como se puede ver hay algunas detecciones falsas, pero en la mayoría de los pixeles, la detección es correcta. A veces es también difícil para un humano de elegir cual pixel representa cual tipo de superficie.

Conclusiones

La clasificación del tipo de la superficie de la tierra a base de las frecuencias espaciales es posible con las dos formas que describí. La calculación de los coeficientes para 360 ángulos necesita muchos recursos y junto a ellos, mucho tiempo. Sin embargo, los resultados están bien. La idea de la pasada por segunda vez con la transformada fourier funciona bien con patrones que contienen una estructura que causa una periodicidad. En el caso de las olas del pacífico, este efecto no es tan grande.

La detección a base de la distribución de las frecuencias sacado de la investigación de muestras es mucho más rápido. Los resultados están bien también. Procesar una imagen de 512 x 512 pixeles necesita aproximadamente 10 minutos. Con el otro algoritmo, se necesitaría más de medio año.

Bibliografía

- [1] Kristian Weyerhäuser: Faltung und Korrelation kontinuierlicher Signale, seminario “Digitale Signalverarbeitung”, Universität Koblenz, Institut für integrierte Naturwissenschaften, Abteilung Physik, 10.6.2005
- [2] Yerin Yoo: Tutorial on Fourier Theory, March 2001,
http://dmery.ing.puc.cl/dmery/dmery/docencia/imagenes/material-adicional/fourier_analysis.pdf
- [3] Carsten Mönnig: Urban land use detection and change using an enhanced monitoring system with remote sensing tools and GIS in the EUREGIO Meuse-Rhine area, tesis doctoral, Fakultät für Georessourcen und Materialtechnik der Rheinisch-Westfälischen Technischen Hochschule Aachen, University Aachen, 2007
- [4] Dan Klang: Automatic Detecion of changes in Road Databases using satellite imagery, Department od Geodesy and Photogrammetry, Stockholm, Sweden, IAPRS, Vol. 32, Part 4 “GIS-Between Visions and Applications”, tesis doctoral, Stuttgart, Germany, 1998
- [5] Anke Bogner, Karl-Heinz Thiemann: Machbarkeitsstudie zur Nutzung von Google Earth in der ländlichen Bodenordnung, 1/2007
- [6] Naceur Saidani: Erkennung von Sturmschäden im Wald auf der Basis kleinmaßstäbiger Luftbilder und Entwicklung eines Modells zur Abschätzung der Sturmgefährdung der Wälder auf der Basis von räumlichen Daten, tesis doctoral, Freiburg, Alemania 04/2004
- [7] Hans-Jörg Fuchs: Methodische Ansätze zur Erfassung von Waldbäumen mittels digitaler Luftbildauswertung, tesis doctoral, universidad Göttingen, Alemania 11/2003

Anexo A: Manual de uso del programa

Estoy mostrando en este informe todos los programas que usé, tanto para la investigación de los algoritmos como el detector final. Este está realizado en el archivo *detector.m* y tiene 3 parámetros. El primero es el nombre del archivo, que contiene la imagen para procesar. El segundo es el nombre del archivo en que el programa va a memorizar la imagen del resultado. El último parámetro es el tamaño por el cual el procesamiento está hecho.

El programa tiene la emisión de la columna, que está justo en este momento listo con la examinación. Eso sirve para saber si el programa está todavía corriendo y cuantas columnas faltan.

Para funcionar, también son necesarios los archivos *transformada_fourier.m*, *clasificador_frecuencia.m* y *caracteristica_promedio.mat*. El último contiene la base de las características de la distribución de las frecuencias.

Anexo B: Los Programas

deteccion.m

```
% FUNDAMENTOS DE PROCESAMIENTO DE PROCESAMIENTO DE IMAGENES IIC/IEE-3713
%
% Tarea 2
% Ralf Wilke ralf@ralfwilke.com
% No Alumno 08410038
%
% 14.10.2008
%
% Deteccion de 10 imagenes

tic
N = 512;
detector('./Deteccion/ejemplo1.bmp', './Deteccion/ejemplo1_det.png', N);
detector('./Deteccion/ejemplo2.bmp', './Deteccion/ejemplo2_det.png', N);
detector('./Deteccion/ejemplo3.bmp', './Deteccion/ejemplo3_det.png', N);
detector('./Deteccion/ejemplo4.bmp', './Deteccion/ejemplo4_det.png', N);
detector('./Deteccion/ejemplo5.bmp', './Deteccion/ejemplo5_det.png', N);
detector('./Deteccion/ejemplo6.bmp', './Deteccion/ejemplo6_det.png', N);
detector('./Deteccion/ejemplo7.bmp', './Deteccion/ejemplo7_det.png', N);
detector('./Deteccion/ejemplo8.bmp', './Deteccion/ejemplo8_det.png', N);
detector('./Deteccion/ejemplo9.bmp', './Deteccion/ejemplo9_det.png', N);
detector('./Deteccion/ejemplo10.bmp', './Deteccion/ejemplo10_det.png', N);
toc
```

detector.m

```
% FUNDAMENTOS DE PROCESAMIENTO DE PROCESAMIENTO DE IMAGENES IIC/IEE-3713
%
% Tarea 2
% Ralf Wilke ralf@ralfwilke.com
% No Alumno 08410038
%
% 14.10.2008
%
% Detector de los superficies de la tierra de imagenes de Google Earth

function clase = detector (inputfile,outputfile,N_Source)
%inputfilename = './source/ejemplo1.bmp';
%inputfile = 'santiago.bmp';
%outputfile = 'santiago_detecado.bmp';
%N_Source = 512;
```

```

N_Matrix = 35;

% Acorde con la tarea, primero pasar al valores de gris
I = rgb2gray(imread(inputfile));

% Cortar la imagen, si es necesario
I = imcrop(I,[1 1 N_Source N_Source]);

% Preparar la nueva imagen con resultados de la deteccion
clase = zeros(N_Source, N_Source);

% Debemos tener una aera del examen de N_Matrix x N_Matrix
% Por eso, no podemos procesar el margen de la imagen
% Empezamos con la primera aera posible para contener la matriz completa
for i = ((N_Matrix+1)/2):(N_Source-((N_Matrix+1)/2)+1)
    for j = ((N_Matrix+1)/2):(N_Source-((N_Matrix+1)/2)+1)
        % Cortar con una mascara moviendo pequenas imagens del tamano
        % N_Matrix x N_Matrix
        AreaDelExamen = imcrop(I,[(i-((N_Matrix+1)/2)+1) (j-((N_Matrix+1)/2)+1) (N_Matrix-1)
(N_Matrix-1)]);
        % Clasificar
        clase(j,i) = clasificador_frecuencia(AreaDelExamen,N_Matrix);
    end
    % Para saber, si el programa funcionar todavia ;- )
    i
end

clase_hsv = 1 * ones(N_Source, N_Source,3);
clase_hsv(:,:,1) = (1/6) * clase;
clase_hsv(:,:,3) = clase>0;
clase_rgb = hsv2rgb(clase_hsv);

figure;imshow(imcrop(imread(inputfile),[1 1 N_Source N_Source]));
figure;imshow(clase_rgb);

imwrite (clase_rgb,outputfile,'png');

```

clasificador_frecuencia.m

```

% FUNDAMENTOS DE PROCESAMIENTO DE PROCESAMIENTO DE IMAGENES IIC/IEE-3713
%
% Tarea 2
% Ralf Wilke ralf@ralfwilke.com
% No Alumno 08410038
%
% 14.10.2008
%
% Clasificacion con la frecuencia

function class = clasificador_frecuencia (I_gray, N_Matrix)

% class 1 = agua
% class 2 = vegetacion
% class 3 = montana
% class 4 = desierto
% class 5 = rio
% class 6 = ciudad

% Cargar los resultados de la investigacion
%load ('caracteristica.mat'); caracteristica_promedio = caracteristica;

% Resultados mejores con el promedio
load ('caracteristica_promedio.mat');

% Hacer la transformacion de Fourier
f = abs(transformada_fourier(I_gray,N_Matrix));

% Juntar los puntos con la misma frecuencia
f_sum = zeros((N_Matrix+1)/2,1);
medio = (N_Matrix+1)/2;
for i = 1:((N_Matrix+1)/2)
    start = medio-i+1;
    stop = medio+i-1;

```

```

        f_sum(i) = sum(f(start,start:stop)) + sum(f(stop,start:stop)) + ...
            sum(f(start:stop,start)) + sum(f(start:stop,stop));
end

%f_sum tiene ahora la cantidad de los valores para cada frecuencia

% Hacer la clasificacion con la diferencia entre las muestras y f_sum
diferencia = zeros(18,6);
diferencia_hit = zeros(6,1);
% Para cada tipo de muestra
for tipo = 1:6
    % Calcular la diferencia
    diferencia(:,tipo) = abs(caracteristica_promedio(:,tipo)-f_sum);
    % Sumar sin la frecuencia 0, que es el promedio de los valores. Por eso
    % empezar por indice 2
    diferencia_hit(tipo) = sum(diferencia(1:18,tipo));
end;
% Donde esta el minimo de la diferencia, los curvas se parecen mas
[C,class] = min(diferencia_hit); clear C;

% -----
% La clasificacion con areas no ha funcionado. Los diferencias no estan
% suficiente grande.

% limites [ f_min f_max valor_min valor_max ]
%lim_agua      = [2,12,0.0e4,2.0e4];
%lim_bosque    = [2,11,1.5e4,3.8e4];
%lim_montana   = [2,18,1.6e4,4.5e4];
%lim_desierto  = [2,15,0.2e4,1.8e4];
%lim_rio       = [2,10,2.5e4,7.8e4];
%lim_ciudad    = [2, 9,1.5e4,5.5e4];

% Juntar la informacion para procesamiento mas facil
%lim_all = [lim_agua; lim_bosque; lim_montana; lim_desierto; lim_rio; lim_ciudad];
%figure;plot(f_sum(2:18));axis([0 19 0 80000]);
%hit_sum = zeros(6,2);
%for tipo = 1:6
%    seleccion = (f_sum(lim_all(tipo,1):lim_all(tipo,2)));
%    figure;plot(seleccion(tipo,:));
%    seleccion_hit = not(or((seleccion<lim_all(tipo,3)),(seleccion>lim_all(tipo,4))));
%    figure;plot(seleccion_hit);
%    hit_sum(tipo,:) = [sum(seleccion_hit), (lim_all(tipo,2)-lim_all(tipo,1)+1)];
%end

```

transformada_fourier.m

```

% FUNDAMENTOS DE PROCESAMIENTO DE PROCESAMIENTO DE IMAGENES IIC/IEE-3713
%
% Tarea 2
% Ralf Wilke ralf@ralfwilke.com
% No Alumno 08410038
%
% 14.10.2009
%
% Transformada Fourier

function F_Shift = transformada_fourier(I,N_Matrix)
% Probé con eso, pero no funcionó
%pl_adj = imadjust(pl_gray,stretchlim(pl_gray),[0.1 0.9]);

% Cortar la imagen del tamaño N_Matrix x N_Matrix
pl_crop = imcrop(I,[1 1 (N_Matrix-1) (N_Matrix-1)]);
% Hacer la transformada
P1 = fft2(pl_crop);
F_Shift = fftshift(P1);

```

investigacion_frecuencia.m

```

% FUNDAMENTOS DE PROCESAMIENTO DE PROCESAMIENTO DE IMAGENES IIC/IEE-3713
%
% Tarea 2
% Ralf Wilke ralf@ralfwilke.com

```

```

% No Alumno 08410038
%
% 14.10.2009
%
% Investigacion del las muestras en el espacio de la frecuencia
% para saber, como clasificar

function f_sum_promedio = investigacion_frecuencia

N_Matrix = 35;
% Tenemos 3 de cada una categoria
f_muster = zeros(N_Matrix, N_Matrix,3,6);

Muster1 = rgb2gray(imread('./pacifico/pacificol.bmp'));
Muster2 = rgb2gray(imread('./bosque/bosque1.bmp'));
Muster3 = rgb2gray(imread('./montana/montanal.bmp'));
Muster4 = rgb2gray(imread('./desierto/desierto1.bmp'));
Muster5 = rgb2gray(imread('./rio/rio1.bmp'));
Muster6 = rgb2gray(imread('./ciudad/ciudad3.bmp'));

f_muster(:,:,1,1) = abs(transformada_fourier(Muster1,N_Matrix));
f_muster(:,:,1,2) = abs(transformada_fourier(Muster2,N_Matrix));
f_muster(:,:,1,3) = abs(transformada_fourier(Muster3,N_Matrix));
f_muster(:,:,1,4) = abs(transformada_fourier(Muster4,N_Matrix));
f_muster(:,:,1,5) = abs(transformada_fourier(Muster5,N_Matrix));
f_muster(:,:,1,6) = abs(transformada_fourier(Muster6,N_Matrix));

% -----
Muster1 = rgb2gray(imread('./pacifico/pacifico2.bmp'));
Muster2 = rgb2gray(imread('./bosque/bosque2.bmp'));
Muster3 = rgb2gray(imread('./montana/montana2.bmp'));
Muster4 = rgb2gray(imread('./desierto/desierto2.bmp'));
Muster5 = rgb2gray(imread('./rio/rio2.bmp'));
Muster6 = rgb2gray(imread('./ciudad/ciudad4.bmp'));

f_muster(:,:,2,1) = abs(transformada_fourier(Muster1,N_Matrix));
f_muster(:,:,2,2) = abs(transformada_fourier(Muster2,N_Matrix));
f_muster(:,:,2,3) = abs(transformada_fourier(Muster3,N_Matrix));
f_muster(:,:,2,4) = abs(transformada_fourier(Muster4,N_Matrix));
f_muster(:,:,2,5) = abs(transformada_fourier(Muster5,N_Matrix));
f_muster(:,:,2,6) = abs(transformada_fourier(Muster6,N_Matrix));

% -----
Muster1 = rgb2gray(imread('./pacifico/pacifico3.bmp'));
Muster2 = rgb2gray(imread('./bosque/bosque3.bmp'));
Muster3 = rgb2gray(imread('./montana/montana3.bmp'));
Muster4 = rgb2gray(imread('./desierto/desierto3.bmp'));
Muster5 = rgb2gray(imread('./rio/rio3.bmp'));
Muster6 = rgb2gray(imread('./ciudad/ciudad2.bmp'));

f_muster(:,:,3,1) = abs(transformada_fourier(Muster1,N_Matrix));
f_muster(:,:,3,2) = abs(transformada_fourier(Muster2,N_Matrix));
f_muster(:,:,3,3) = abs(transformada_fourier(Muster3,N_Matrix));
f_muster(:,:,3,4) = abs(transformada_fourier(Muster4,N_Matrix));
f_muster(:,:,3,5) = abs(transformada_fourier(Muster5,N_Matrix));
f_muster(:,:,3,6) = abs(transformada_fourier(Muster6,N_Matrix));

f_sum = zeros(N_Matrix,3,6);
medio = (N_Matrix+1)/2;

% Cada categoria
for k = 1:6
    % Cada paquete de muestras
    for j = 1:3
        % Cada frecuencia
        for i = 1:(N_Matrix+1)/2
            start = medio-i+1;
            stop = medio+i-1;
            f_sum(i,j,k) = sum(f_muster(start,start:stop,j,k)) + ...
                sum(f_muster(stop,start:stop,j,k)) + ...
                sum(f_muster(start:stop,start,j,k)) + ...
                sum(f_muster(start:stop,stop,j,k));
        end
    end
end

```

```

    end
    figure; plot(f_sum(2:19,:,k));axis([0 19 0 80000]);
end

% Para tener el promedio para cada categoria
f_sum_promedio= zeros(18,6);
for a = 1:18
    for b = 1:6
        %
            f_sum_promedio(a,b) = f_sum(a,1,b);
            f_sum_promedio(a,b) = (f_sum(a,1,b)+f_sum(a,2,b)+f_sum(a,3,b))/3;
        end
    end
end

figure; plot(f_sum_promedio(2:18,:));axis([0 19 0 80000]);
caracteristica_promedio = f_sum_promedio;

% Memorizar los resultados tambien en un archivo
save('caracteristica_promedio.mat', 'caracteristica_promedio');

```

Clasificador_correlacion.m

```

% FUNDAMENTOS DE PROCESAMIENTO DE PROCESAMIENTO DE IMAGENES IIC/IEE-3713
%
% Tarea 2
% Ralf Wilke ralf@ralfwilke.com
% No Alumno 08410038
%
% 14.10.2009
%

% Clasificador al base de la correlacion
% class 1 = agua
% class 2 = vegetacion
% class 3 = montana
% class 4 = desierto
% class 5 = rio
% class 6 = ciudad

% Tamano de la area que estoy examinando
N_Matrix = 25;

% Hacer las muestras con ejemplos de los categorias
M_agua = investigacion_general(imread('./pacifico/pacifico1.bmp'),N_Matrix,0);
M_vegetacion = investigacion_general(imread('./bosque/bosque1.bmp'),N_Matrix,0);
M_montana = investigacion_general(imread('./montana/montana1.bmp'),N_Matrix,0);
M_desierto = investigacion_general(imread('./desierto/desierto1.bmp'),N_Matrix,0);
M_rio = investigacion_general(imread('./rio/rio1.bmp'),N_Matrix,0);
M_ciudad = investigacion_general(imread('./ciudad/ciudad3.bmp'),N_Matrix,0);

% La imagen para examinar
I = imread('./ciudad/ciudad4.bmp');

% Calcular el coeficiente de la correlacion para cada angulo con cada
% muestra
for angle = 1:360
    J = investigacion_general(I,N_Matrix,angle);
    corr(angle,1) = corr2(J,M_agua);
    corr(angle,2) = corr2(J,M_vegetacion);
    corr(angle,3) = corr2(J,M_montana);
    corr(angle,4) = corr2(J,M_desierto);
    corr(angle,5) = corr2(J,M_rio);
    corr(angle,6) = corr2(J,M_ciudad);
end;

% Monstra
figure;plot(corr);

% Idea: Si la imagen para examinar esta similar con la muestra, la
% distribucion del coeficiente con el angulo esta periodico con 180°
% Entonces la transformada de esa va a tener principalmene frecuencias bacos y pocos
% frecuencias altas. Si uso un filtro pasa alto y sumo los valores que
% pasan, voy a sumar ruido blanco, si no hay conformidad con la muestra.

```

```

% Entonces la suma mas baja coresondes a la categoria detectada.
% Por lo que la idea ....

FFT = 2^nextpow2(360);
% Para cada categoria
for tipo = 1:6
    f = corr(:,tipo) - (ones(360,1) * sum(corr(:,tipo))/360);
    % Restar el promedio
    % Hacer la transformada Fourier y valor absoluto
    f_corr(:,tipo) = fft(f,NFFT)/360;
    abs_f_corr = abs(f_corr);

    % Monstrar
    figure;plot(f);
    figure;plot(abs_f_corr(1:60,tipo));
    % Hacer un filtro pasa alto, sumar y normalizar
    valor_f_corr(tipo) = sum(abs_f_corr(3:360,tipo))/sum(abs_f_corr(:,tipo));
end

valor_f_corr
[a clase] = min(valor_f_corr); clear a;
% Class tiene la categoria detectada
clase

```

Analisis_corr.m

```

% FUNDAMENTOS DE PROCESAMIENTO DE PROCESAMIENTO DE IMAGENES IIC/IEE-3713
%
% Tarea 2
% Ralf Wilke ralf@ralfwilke.com
% No Alumno 08410038
%
% 14.10.2009
%
% Investigacion, si la idea con correlacion de la transformada fourier
% y SU transformada fourier puedo funcionar

close all;

N_Matrix = 25;

aguaM = investigacion_general(imread('./pacifico/pacifico1.bmp'),N_Matrix,0);
ciudadM = investigacion_general(imread('./ciudad/ciudad3.bmp'),N_Matrix,0);

for angle = 1:360
    agua = investigacion_general(imread('./pacifico/pacifico3.bmp'),N_Matrix,angle);
    ciudad = investigacion_general(imread('./ciudad/ciudad4.bmp'),N_Matrix,angle);
    corr_aa(angle) = corr2(agua,aguaM);
    corr_ac(angle) = corr2(agua,ciudadM);
    corr_cc(angle) = corr2(ciudad,ciudadM);
    corr_ca(angle) = corr2(ciudad,aguaM);
end;

figure;plot(corr_aa);max(corr_aa);sum_aa = sum(corr_aa.^2)/360
figure;plot(corr_ac);max(corr_ac);sum_ac = sum(corr_ac.^2)/360
figure;plot(corr_cc);max(corr_cc);sum_cc = sum(corr_cc.^2)/360
figure;plot(corr_ca);max(corr_ca);sum_ca = sum(corr_ca.^2)/360

NFFT = 2^nextpow2(360);
f_aa = fft(corr_aa-(sum(corr_aa)/360),NFFT)/360;
f_ac = fft(corr_ac-(sum(corr_ac)/360),NFFT)/360;
f_cc = fft(corr_cc-(sum(corr_cc)/360),NFFT)/360;
f_ca = fft(corr_ca-(sum(corr_ca)/360),NFFT)/360;

k = 60;
figure;plot(abs(f_aa(1:k)));
figure;plot(abs(f_ac(1:k)));
figure;plot(abs(f_cc(1:k)));

```

```

figure;plot(abs(f_ca(1:k)));

abs_aa = abs(f_aa);
abs_ac = abs(f_ac);
abs_cc = abs(f_cc);
abs_ca = abs(f_ca);

v_aa = sum(abs_aa(7:360))/sum(abs_aa)
v_ac = sum(abs_ac(7:360))/sum(abs_ac)
v_cc = sum(abs_cc(7:360))/sum(abs_cc)
v_ca = sum(abs_ca(7:360))/sum(abs_ca)

```

Investigacion_general.m

```

% FUNDAMENTOS DE PROCESAMIENTO DE PROCESAMIENTO DE IMAGENES IIC/IEE-3713
%
% Tarea 2
% Ralf Wilke ralf@ralfwilke.com
% No Alumno 08410038
%
% 14.10.2009
%
% Para apender algo sobre la distribucion de las frecuencias

function f = investigacion_general(I,N_Matrix,angle)
% Investigacion de los caracerisicas in la frecuencia
p1_gray = rgb2gray(I);
%figure, imshow(p1);
%p1_adj = imadjust(p1_gray,stretchlim(p1_gray),[0.1 0.9]);
p1_adj = p1_gray;
%figure, imshow(p1_adj);
p1_rot = imrotate(p1_adj,angle,'bicubic');
%figure;imshow(p1_rot);
p1_rot_size = size(p1_rot);
x_start = round((p1_rot_size(1)/2)-((N_Matrix-1)/2));
y_start = round((p1_rot_size(2)/2)-((N_Matrix-1)/2));
p1_crop = imcrop(p1_rot,[x_start y_start (N_Matrix-1) (N_Matrix-1)]);
%figure, imshow(p1_crop);
P1 = fft2(p1_crop);
P1_Sh = fftshift(P1);
%figure;
%mesh(log(abs(P1_Sh)+1));
%mesh(abs(P1_Sh));
%figure, imshow(log(abs(P1_Sh)+1),[0 10]);
%P1_FPA = FPA(abs(P1_Sh),3,N_Matrix);
%Minimum = min(min(P1_Sh))
%Maximum = max(max(P1_Sh))
%figure;
%imshow(P1_FPA,[0 500]);
%mesh(P1_FPA);
%f = sqrt(abs(P1_Sh));
f = abs(P1_Sh);
%figure, imshow(f,[0 50]);

```

color_clase.m

```

% FUNDAMENTOS DE PROCESAMIENTO DE PROCESAMIENTO DE IMAGENES IIC/IEE-3713
%
% Tarea 2
% Ralf Wilke ralf@ralfwilke.com
% No Alumno 08410038
%
% 14.10.2008
%
% Para saber la asignacion de los colores

i = (1/6) * [0:6;0:6;0:6;0:6;0:6;0:6;0:6];
clase_hsv = 1 * ones(7, 7,3);
clase_hsv(:,:,1) = i;
clase_hsv(:,:,3) = i>0;
clase_rgb =imresize(hsv2rgb(clase_hsv),10);
figure;imshow(clase_rgb);

```