

# Prácticas de Visión Artificial

---

## Práctica 4: Búsqueda y Recuperación de imágenes con texturas

---

En esta práctica vamos a ver cómo podemos utilizar las texturas para búsqueda de imágenes.

Los temas principales que necesitaremos son:

- 1) Texturas - filtros de Gaussianas.
- 2) Descriptores de imágenes basados en textura y color

Para completar la práctica es necesario conocer y aplicar los conceptos básicos de las texturas. Para esto, consultar la parte del material de teoría.

### 4.1 Extracción de descriptores de textura

Las texturas que vamos a utilizar en el primer apartado serán los filtros de la Gaussiana (mirar transparencias de clase). Estas se pueden bajar de <http://www.robots.ox.ac.uk/~vgg/research/texclass/code/makeLMfilters.m>.

Una pequeña explicación se puede encontrar en: <http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html>.

4.1.1. Descargar el código para generar un banco de filtros de la Gaussiana (dicho de otra manera: el banco de filtros de Leung-Malik (LM) y visualizarlos en una figura:

```
function [ ] = testFiltros()
% Esta función pretende ilustrar el banco de filtros de la gaussiana

    F=makeLMfilters(); % genera los filtros
    visualizeFilters(F);
end

function [ ] = visualizeFilters(F)
% This function receives a bank of filters and visualize them by
% pseudocolors

    figure, % visualiza todos los filtros
    for k=1:size(F,3);
        subplot(8,6,k);
        imagesc(F(:,:,k)); colorbar;
    end

end
```

Nota: fíjense en los comandos `imagesc()` y `colorbar`. ¿Qué hacen?

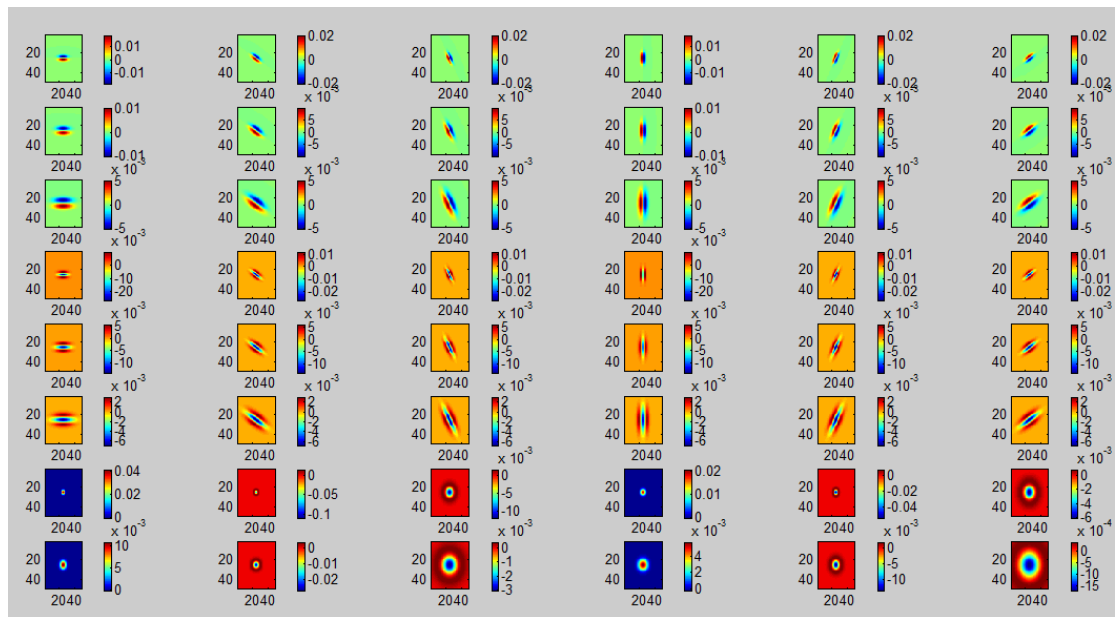


Fig.1. Banco de filtros de la Gaussiana

¿A qué corresponden los diferentes filtros? ¿Qué valores tienen?

4.1.2 Implementa una función *getFeatures* en Matlab que dada una imagen, construye un descriptor de texturas donde cada elemento del descriptor es la **promedia** del resultado de la convolución de la imagen con alguno de los filtros. Nota: al buscar la textura de la imagen, no nos importa el color. Por lo tanto, tenéis que pasar la imagen de color en nivel de gris. Visualiza el resultado de la convolución con algunos filtros. Observa qué filtros tienen una mejor respuesta sobre la imagen que has escogido y comenta el porqué. ¿Qué dimensión tiene el descriptor?

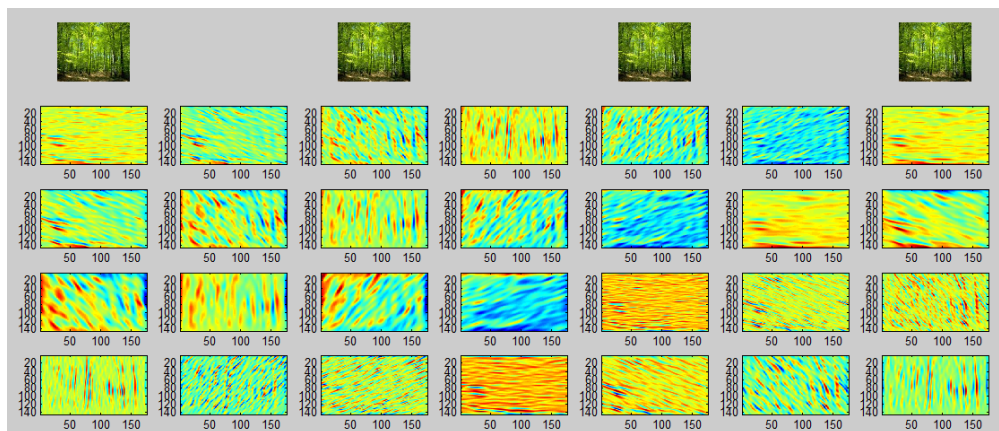


Fig.2 Resultado de la convolución de la imagen arriba (está repetida) con diferentes filtros

4.1.3 Escribir una función *getClassFeatures* en Matlab que dado un directorio y una extensión, lee todas las imágenes del directorio con la extensión concreta, calcula sus descriptores de textura (usando 4.1.1) y los guarda en una matriz donde cada fila corresponde a una imagen y cada columna a una característica. Aplicar esta función para construir tres matrices de descriptores de textura de las 3 clases de imágenes en el directorio textureimages proporcionados en el Campus Virtual. Ayuda: dado un directorio en la variable *directory* (p.e. 'images/sunset/'), el siguiente comando lee los ficheros que hay en este directorio:

```
files=dir(fullfile(directory, '*.jpg'));
im=imread(fullfile(directory, files(i).name));
```

Nota: La forma de acceder al i-ésimo fichero del directorio será a través de files(i).name (comprueba qué hace fullfile).

4.1.4 Escribe una función visualizeFeatures que visualiza en un plot una lista de características (p.e. [25, 41]) para las imágenes de los tres directorios con un color diferente para cada directorio (ayuda: plot). ¿A qué se corresponde cada eje en la visualización?

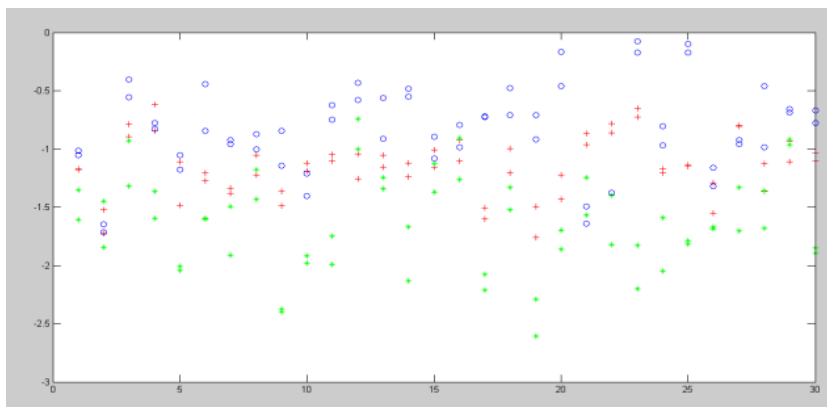


Fig.3 Separación de las 3 clases de imágenes según su textura (características 41 y 25).

4.1.5 Para cada imagen de las tres clases, escribe una función retrieveKImages en Matlab que recupera y visualiza las k (k=9) imágenes más parecidas según sus descriptores de textura (ver Fig. 4). Puedes utilizar el comando knnsearch() en Matlab. (Opcional): observar cómo mejora el resultado si aparte de los filtros de textura añadimos el color (r,g,b) como tres características más por cada imagen. ¿Qué dimensión tendrá el espacio de características si añadimos el color? Observa cuáles son las características más discriminativas para cada conjunto de imágenes primero sin utilizar color y luego utilizando color. Comenta tus observaciones sobre el funcionamiento del algoritmo y posibles mejoras.

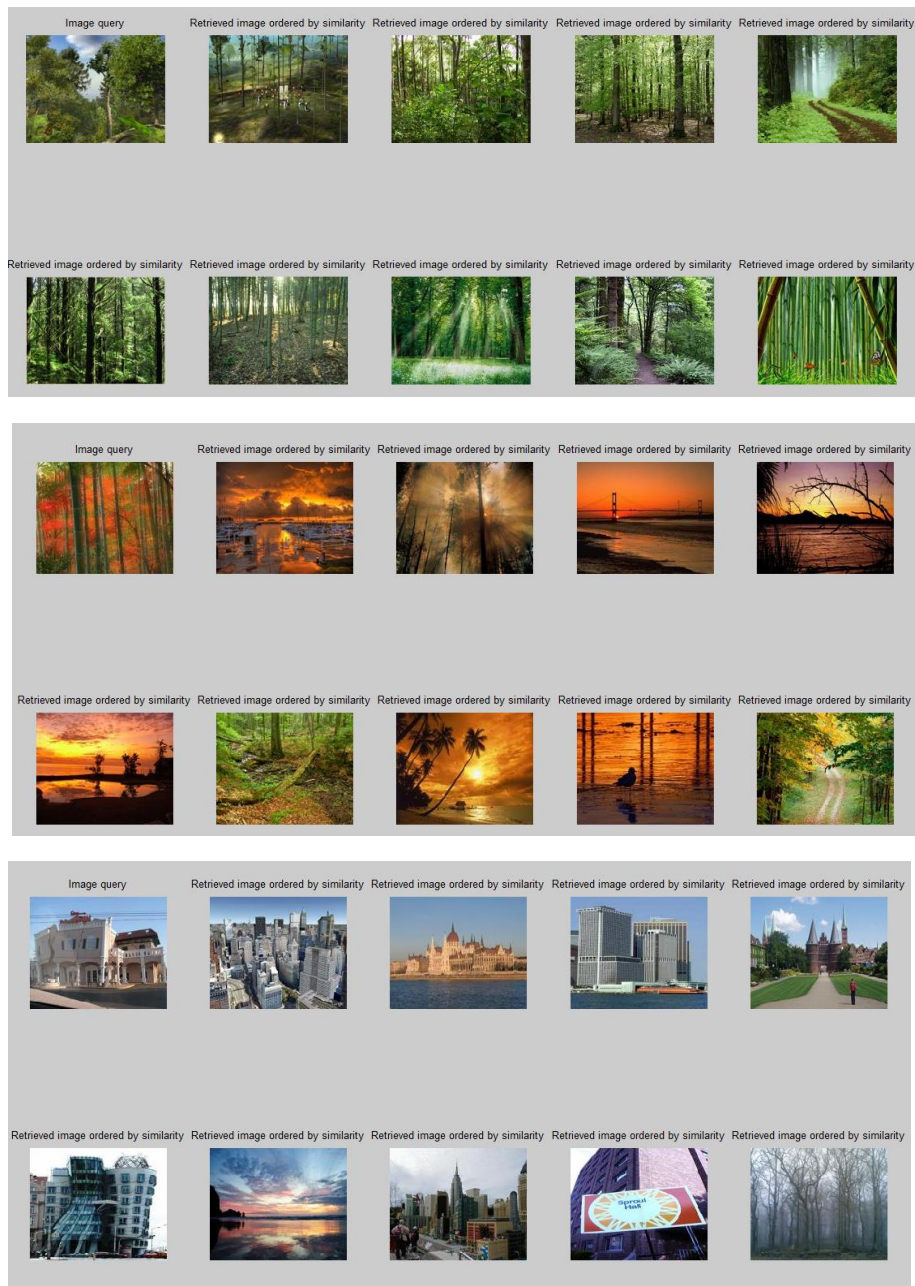


Fig.4 Resultados de la búsqueda por semejanza usando descriptores basados en textura y color