

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 este primer ejercicio 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: Fíjate en los comandos *imagesc* y *colorbar*. ¿Qué hacen?

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

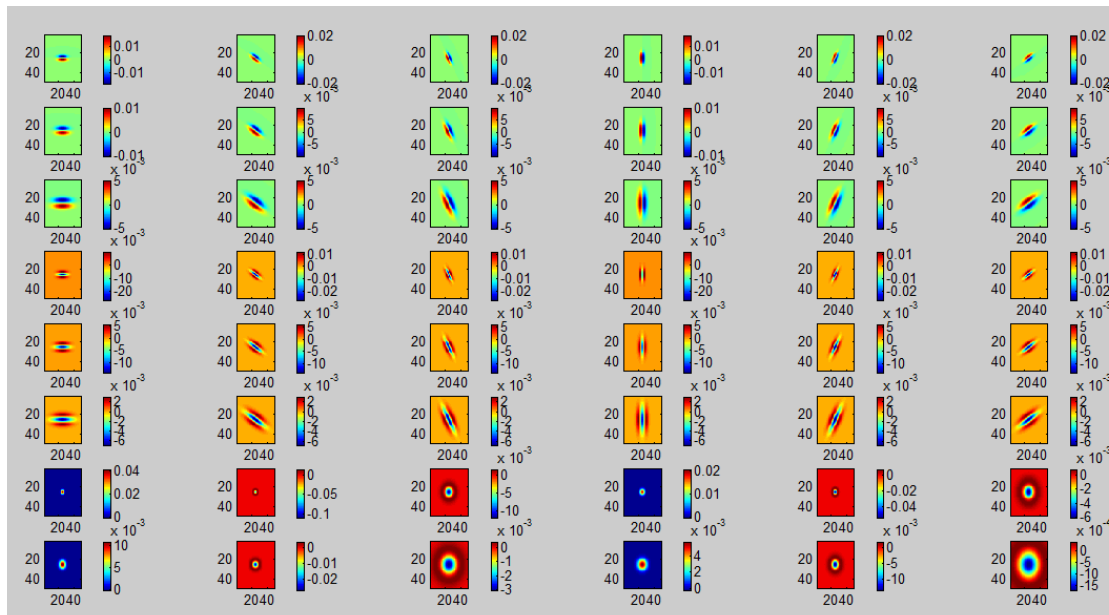


Figura 1. Banco de filtros de la Gaussiana

4.1.2 Implementar una función *getFeatures* que dada una imagen, construye un descriptor de texturas, donde cada elemento del descriptor es el **promedio** del resultado de la convolución de la imagen con los filtros. ¿Qué dimensión tiene el descriptor?

Nota 1: Al buscar la textura de la imagen, no nos importa el color. Por lo tanto, tenéis que convertir la imagen en color a escala de grises.

Nota 2: Utiliza `conv2()` en lugar de `imfilter()` de Matlab para obtener respuestas positivas y negativas.

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.

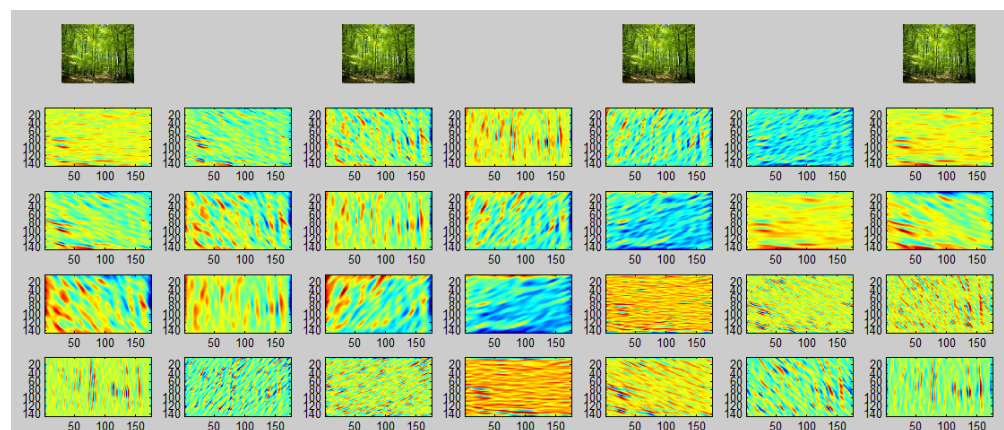


Figura 2. Resultado de la convolución de la imagen arriba (está repetida) con diferentes filtros.

4.1.3 Escribir una función *getClassFeatures* que dado un directorio y una extensión, lee todas las imágenes del directorio con la extensión dada, calcula sus descriptores de textura (usando *getFeatures* del 4.1.2) y los guarda en una matriz donde cada fila es una imagen y cada columna una característica. Aplicar esta función para construir tres matrices de descriptores de textura de las 3 clases de imágenes del directorio 'texturesimages' facilitado con el enunciado.

Nota: Dado un directorio en la variable *directory* (por ejemplo: 'texturesimages/sunset/'), el siguiente comando lee los ficheros que hay en dicho 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 Escribir una función *visualizeFeatures* que visualiza en un plot una lista de una o varias características (por ejemplo: [25, 41]) para las imágenes de los tres directorios con un color diferente para cada directorio (*help plot*).

(Opcional): Escribir otra función *visualizeFeatures3D* que visualiza en 3D los tres grupos de imágenes utilizando tres características determinadas (por ejemplo: [25, 38, 41]).

¿A qué se corresponde cada eje en las visualizaciones 2D y 3D?

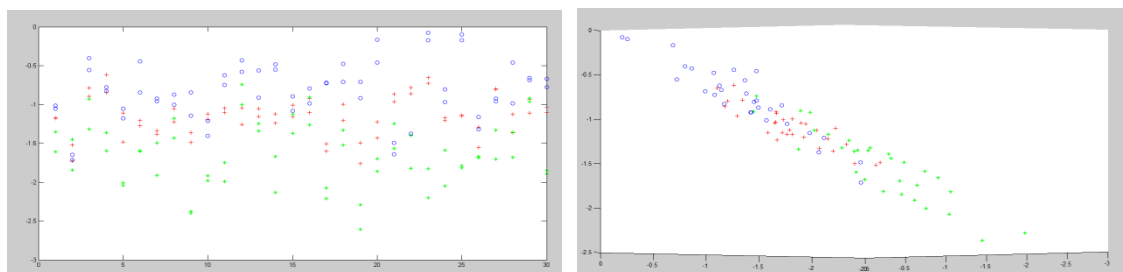


Figura 3. Separación de las 3 clases de imágenes según su textura: características 41 y 25, en el caso 2D (izquierda), y características 41, 25 y 38 en el caso 3D (derecha)).

4.1.5 Para cada imagen de las tres clases, escribir una función *retrieveKImages* que recupera y visualiza las *k* (*k*=9) imágenes más parecidas según sus descriptores de textura (ver Fig. 4). Nota: utilizar el comando *knnsearch*.

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.

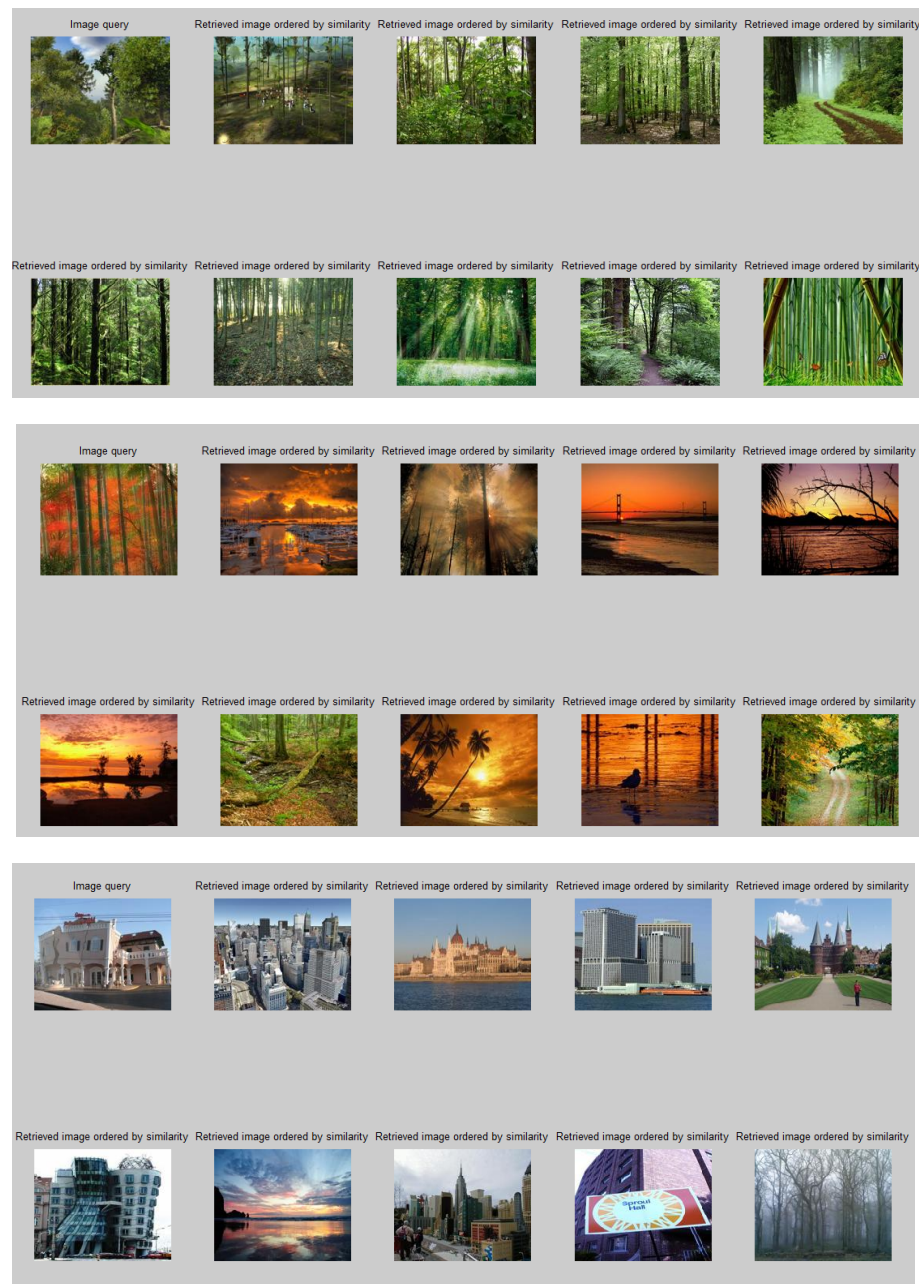


Figura 4. Resultados de la búsqueda por semejanza usando descriptores basados en textura y color.

4.2 Local binary patterns (LBP)

Para realizar este ejercicio necesitamos la librería utilizada en la práctica 3, que también implementa LBP: <http://www.vlfeat.org/index.html>.

Para instalarla en Matlab, necesitamos ejecutar:

```
% Install the VLFeat toolbox
```

```
srcPath=cd('toolbox');  
vl_setup;  
cd(srcPath);  
close all;
```

Revisar el uso del método LBP en las transparencias de teoría, y consultar:

http://www.vlfeat.org/matlab/vl_lbp.html

4.2.1 ¿Qué parámetros de entrada recibe el método *vl_lbp*? ¿A qué corresponde el valor devuelto y qué dimensión tiene?

4.2.2 Implementar una función *getLBPfeatures* que dada una imagen, construye un descriptor de texturas definido como el histograma promedio de los histogramas calculados por la función *vl_lbp* para cada región analizada. ¿Qué dimensión tiene el descriptor?

Utiliza la función *retrieveKImages* implementada en el ejercicio 4.1.5 utilizando como características los descriptores de textura obtenidos con LBP. Compara los resultados obtenidos en ambos casos (filtros Gaussianos vs. LBP), y explica las ventajas de cada uno.

Entrega de la práctica

Tiempo máximo de entrega: martes 13 de Diciembre **a las 23:00h.**

Material para entregar: archivo “nombres_apellidos_P4.zip” que contenga:

- Archivos .m de cada ejercicio con las funciones creadas. **El código debe estar comentado en inglés.**
- Fichero .pdf que incluya las respuestas planteadas en los distintos ejercicios.