Descripción de la Textura con LBP

Objetivos:

- Aprender a calcular y aplicar el descriptor de textura LBP.
- Aprender a describir la textura de una región usando histogramas.

1. Parte mínima de la práctica (hasta 7/10 puntos)

Codifica una programa denominado "`lbpdesc.exe`" que testeará las siguientes funciones que serán implementadas en un módulo LBP (ver en el esqueleto proporcionado los ficheros `lbp.cpp, lbp.h`):

A. (2.5 pts) Dada una imagen con tipo cv::8UC1 calcular el correspondiente código LBP para cada pixel (que se pueda calcular, rellenar con zeros donde no se pueda).
```
void fsiv_lbp(const cv::Mat & imagem, cv::Mat & lbpmat);
```

B. (0.5 pt) Visualiza una imagen LBP.
```
void fsiv_lbp_disp(const cv::Mat & lbpmat, const
std::string & winname);
```

C. (2 pts) Codifica una función para calcular un histograma de LBP de una región a partir de la imagen LBP. El histograma estará normalizado (suma de todos los elementos será 1.0) por defecto. Sugerencia estudia la función cv::calcHist().
```
void fsiv_lbp_hist(const cv::Mat & lbpmat, cv::Mat &
lbphist, bool normalize=true);
//! \param lbphist [out]: row vector with 256 dimensions
//! \param normalize: return a normalized histogram.
Default, true.
```

D. (2 pts) Codifica una función que calcule la distancia Chi^2 entre dos histogramas x, y:

$$d(x,y) = 0.5 \sum_i \frac{(x_i - y_i)^2}{(x_i + y_i)}$$

Aviso: Los histogramas serán matrices con rows=1 y con cols=N. No asumas que N=256 siempre.

Aviso: ten cuidado con las divisiones por cero.
```
float fsiv_chisquared_dist(const cv::Mat & h1, const
cv::Mat & h2);
//! \param h1, h2 Row vectors with the same dimensions.
```

2. Parte opcional de la práctica (hasta 3/7 puntos).

Añade al programa "`lbpdesc.exe`" llamadas para testear las siguiente funcionalidad:

A. (+1.5 pts) Dada una configuración de rejilla (#Filas,#Columnas), dividir la imagen en #Fx#C regiones, calculando el histograma lbp para región y generando finalmente el descriptor de la región completa concatenando los #Fx#C histogramas LBP calculados como una sola fila (sugerencia estudia la función cv::hconcat()). El

descriptor estará normalizados por defecto (sum(desc)==1.0). Utiliza los flag '-r=#F' y '-c=#C' en la CLI del programa de test para indicar que queremos utilizar el descriptor por rejillas.

```
void fsiv_lbp_desc(const cv::Mat & image, cv::Mat &
lbp_desc, const int *ncells, bool normalize=true);
//! \param lbp_desc [out] Row vector containing the
compound LBP descriptor.
//! \param ncells [in] [rows x cols] E.g. {6,4}
```

B. (+1.5 pts) Codifica una función que calcule el LBP uniforme (U-LBP). Usa un flag "-u" para indicar esto en el programa de test. Sugerencia utiliza un parámetro opcional "nbins=256" en las funciones de cálculo de histograma lbp/descriptor en rejilla para reutilizar todo el código posible.

```
void fsiv_ulbp(const cv::Mat & imagem, cv::Mat &
ulbpmat);

void fsiv_lbp_desc(const cv::Mat & image, cv::Mat &
lbp_desc, const int *ncells, bool normalize=true, int
nbins=256);
//! \param lbp_desc [out] Row vector containing the
compound LBP descriptor.
//! \param ncells [in] [rows x cols] E.g. {6,4}
//! \param nbins [in] 256 for LBP, 59 for uLBP.

void fsiv_lbp_hist(const cv::Mat & lbpmat, cv::Mat &
lbphist, bool normalize=true, int nbins=256);
//! \param lbphist [out]: row vector with 256 dimensions
//! \param normalize: return a normalized histogram.
Default, true.
//! \param nbins [in] 256 for LBP, 59 for uLBP.
```

La siguiente tabla permite obtener el código uLBP correspondiente a cada código LBP:

```
static int uniform[256] =
{
0,1,2,3,4,58,5,6,7,58,58,58,8,58,9,10,11,58,58,58,58,58,58,58,12,58,58,58,13,58,
14,15,16,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,17,58,58,58,58,58,58,18,
58,58,58,19,58,20,21,22,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,
58,58,58,58,58,58,58,58,58,58,58,23,58,58,58,58,58,58,58,58,58,58,58,58,58,
58,58,24,58,58,58,58,58,58,58,25,58,58,58,26,58,27,28,29,30,58,31,58,58,58,32,58,
58,58,58,58,58,58,33,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,34,58,58,58,58,
58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,58,
58,35,36,37,58,38,58,58,58,39,58,58,58,58,58,58,58,40,58,58,58,58,58,58,58,58,58,
58,58,58,58,58,58,41,42,43,58,44,58,58,58,45,58,58,58,58,58,58,58,46,47,48,58,49,
```

```
58,58,58,50,51,52,58,53,54,55,56,57
};
```