Realce usando la máscara de desenfoque

Objetivos de la práctica.

- Aprender a mejorar una imagen aplicando un realce por máscara de desenfoque (unsharp mask).
- Aprender a aplicar un cambio de espacio de color.
- Aprender a aplicar una convolución a la imagen.
- Aprender a utilizar la GUI de openCV usando controles con barra de deslizamiento.

Descripción básica.

Recuerda que la operación a aplicar es: $O = (g+1)\cdot I - g\cdot I_L$, siendo I la imagen original, I_L la versión paso baja de la misma y g la ganancia del realce.

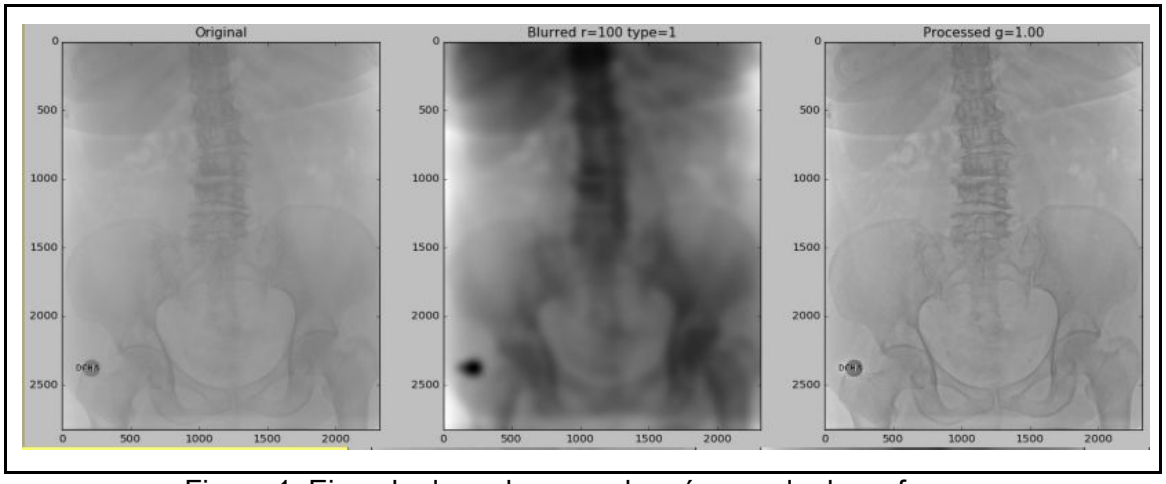

Figura 1. Ejemplo de realce usando máscara de desenfoque.

La operación de realce se realiza sobre el canal de iluminación (“luma”). Si la imagen de entrada está en color, se cambia su espacio de color a otro que separe el “luma”, por ejemplo HSV, y se deshace el cambio a RGB con el canal “luma” realzado (ver cv::cvtColor).

El código debe contener dos funciones:

```
/**
 * @brief Return a box filter.
 * @arg[in] r is the filter's radius.
 * @pre r>0;
 * @post retV.rows==retV.cols==2*r+1
```

```
 * @post cv::sum(retV)==1.0
 */
cv::Mat fsiv_create_box_filter(const int r);

/**
 * @brief Expand an image with zero padding.
 * @arg[in] in is the input image.
 * @arg[in] r is the window's radius to expand.
 * @return the expanded image.
 * @pre !in.empty()
 * @pre r>0
 * @post retV.type()==in.type()
 * @post retV.rows==in.rows+2*r
 * @post retV.cols==in.cols+2*r
 */
cv::Mat fsiv_fill_extension(cv::Mat const& in, const int r);

/**
 * @brief Compute the digital convolution.
 * By default the input image is expanded with zero padding.
 * Note: Code from scracth. Use cv::filter2D is not allowed.
 * @arg[in] in is the input image.
 * @arg[in] filter is the filter to be applied.
 * @arg[in] circular if it is true, the input image is expanded
following a circular fashion.
 * @pre !in.empty() && !filter.empty()
 * @pre (filter.rows % 2) >0
 * @pre (filter.cols % 2) >0
 * @pre in.type()==CV_32FC1 && filter.type()==CV_32FC1.
 * @post ret.type()==CV_32FC1 && ret.size == in.size.
 */
cv::Mat fsiv_convolve(cv::Mat const& in, cv::Mat const& filter, const
bool circular=false);

/**
 * @brief Apply an unsharp mask enhanced to the input image.
 * @arg[in] in is the input image.
 * @arg[in] g is the enhanced's gain.
 * @arg[in] r is the window's radius.
 * @arg[in] filter_type specifies which filter to use. 0->Box,
1->Gaussian.
 * @arg[in] circular specifies if it is true, it be used circular
expansion to do the convolution, else it is used zero padding.
 * @pre !in.empty()
 * @pre in.type()==CV_32FC1
```

```
 * @pre g>=0.0
 * @pre r>0
 * @pre filter_type is {0, 1}
 */
cv::Mat fsiv_usm_enhance(cv::Mat  const& in, float g=1.0, int r=1,
                         int filter_type=0, bool circular=false);
```

La función fsiv_convolve() usará fsiv_fill_extension().
La interfaz de comandos del programa será la siguiente:

```
usm_enhance [-r=1] [-g=1.0] <input img> <output img>
```

- Mediante la opción 'r' controlamos el tamaño del filtro 0<r<`mínimo{W,H}/2`. El valor por defecto será 1.
- Mediante el parámetro flotante 'g', con valores [0.0, 10.0] (por defecto 1.0) es proporcionada la ganancia del realce.

Descripción opcional.

- Añadir la posibilidad de usar un filtro gaussiano. Se utilizará unl parámetro flag 'f': 0->box, 1->gaussian para indicar el tipo de filtro. Para calcular el filtro gaussiano se espera que se codifique la función

```
/**
 * @brief Return a Gaussian filter.
 * @arg[in] r is the filter's radius.
 * @pre r>0;
 * @post retV.rows==retV.cols==2*r+1
 * @post cv::sum(retV)==1.0
 */
cv::Mat fsiv_create_gaussian_filter(const int r);
```

- Implementar la opción de extensión circular para `fsiv_convolve` de forma la extensión de la imagen se haga de forma circular. En la interfaz de comandos se utilizará el flag ‘-c’ para indicar esto. Además se espera que se implemente y use la función siguiente.

```
//Extiende una imagen rellenando de forma circular según un radio de
ventana r.
```

```
//postc: retV.type()==in.type()
//postc: retV.rows==in.rows+2*r
//postc: retV.cols==in.cols+2*r
cv::Mat fsiv_circular_extension(cv::Mat& in, int r);
```

- Añadir soporte de imágenes en color. La operación de realce se realiza sobre el canal de iluminación (“luma”). Si la imagen de entrada está en color, se cambia su espacio de color a otro que separe el “luma”, por ejemplo HSV, y se deshace el cambio a RGB con el canal “luma” realzado (ver cv::cvtColor).
- Añadir modo interactivo. El flag [-i] indica modo interactivo. Se muestran la imagen original, la imagen con la máscara de desenfoque y la imagen procesada. Añadir deslizadores para los parámetros 'radio' y 'ganancia' y mostrar de forma dinámica los cambios. Si el usuario pulsa la tecla “ESC” se sale sin guardar los resultados. Si pulsa la tecla “Intro” se sale guardando el resultado.

Evaluación.

El código está bien escrito, estructurado y compilado.	10%
El código realiza el funcionamiento básico con imágenes monocromas.	50%
El código permite utilizar un filtro gaussiano.	10%
El código implementa la convolución circular.	10%
El código permite procesar las imágenes en color.	10%
El código ofrece una GUI que permite trabajar de forma interactiva todas las opciones.	10%

(* La entrega fuera de plazo supondrá una penalización en la nota obtenida).

Recursos.

Acceso a una ventana de una imagen: cv::Mat::operator().