

Practica2: Modelado de imagen y video

Arturo Labajo

Rubén Urraca

Grupo: fsi005

Ejercicio 1: Cargue la imagen peppers.png , y desatúrela a un valor entre el 40% y 60%. Para ello, se recomienda trabajar en coordenadas HSV aplicando un factor de ponderación al canal de saturación. Represente la imagen desaturada.

Primero, y como dice el enunciado de la práctica, partimos de la imagen peppers.png, la cual deberemos reducir su nivel de saturación.

Con ese propósito, utilizaremos las siguientes líneas de código:

```
% Cargamos la imagen y la pasamos a HSV
I = im2double(imread('peppers.png'));
I_hsv = rgb2hsv(I);

% Desaturamos la imagen
factor_desaturacion = 0.6; % Variamos entre 0.4 y 0.6
I_hsv(:,:,2) = I_hsv(:,:,2) * factor_desaturacion;
I_desaturada = hsv2rgb(I_hsv);
```

En el código, primero cargamos la imagen en double y luego la pasamos a HSV. Luego, para desaturarla, solo hay que multiplicar el canal de saturación por el factor correspondiente. Como consecuencia, obtendremos las imágenes correspondientes que podemos ver en las figuras 1 y 2.



Figura 1: Imagen desaturada 40%



Figura 2: Imagen desaturada 60%

Ejercicio 2: Aplique a continuación un tono cálido al resultado obtenido. Para ello, debe ponderar con valores superiores a 1 los canales rojo y verde, e inferiores a 1, el canal azul. Se recomienda no variar los valores de los canales más de un 20%. Muestre el resultado obtenido.

Con el mismo procedimiento que el ejercicio anterior, debemos ponderar los 3 canales con un factor cada uno. Así pues, el código quedaría de la siguiente manera:

```
% Aplicamos un tono cálido
factor_R = 1.2; % Incrementamos el rojo 20%
factor_G = 1.1; % Incrementamos el verde 10%
factor_B = 0.8; % Reducimos el azul 20%
I_calida = I_desaturada;
I_calida(:,:,1) = I_calida(:,:,1) * factor_R;
I_calida(:,:,2) = I_calida(:,:,2) * factor_G;
I_calida(:,:,3) = I_calida(:,:,3) * factor_B;
I_calida = min(I_calida, 1); % Limitar
```

De esta manera, conseguiremos un tono más cálido para nuestra imagen, como podremos observar en la figura 3.

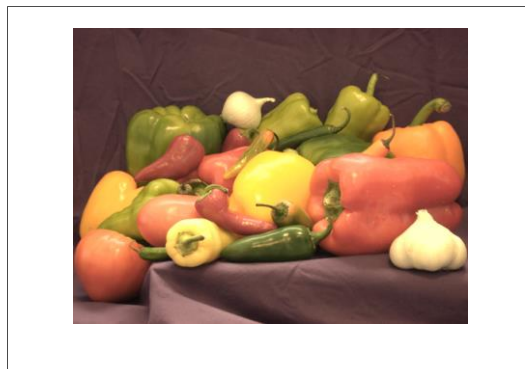


Figura 3: Imagen con tono cálido

Ejercicio 3: Añada ruido a la imagen, justificando la elección del tipo de ruido y los parámetros del mismo sobre la base del ejemplo mostrado.

Para añadir ruido, hemos utilizado la siguiente línea de código:

```
% Añadimos ruido Gaussiano
I_ruido = imnoise(I_calida, 'gaussian', 0.05);
```

Hemos elegido el ruido gaussiano porque da ese efecto “vintage” que queremos, a causa de un 5%, renombrando así la imagen cuyo resultado vemos en la figura 4.

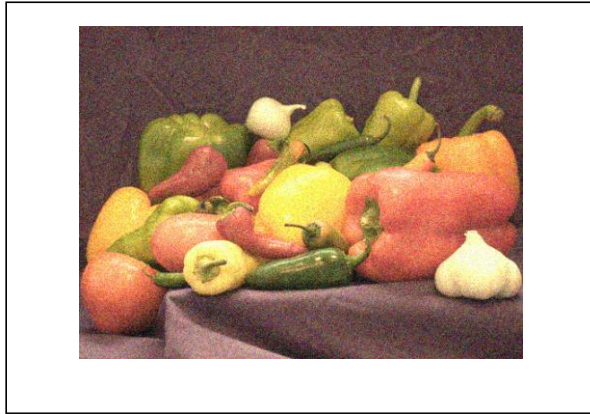


Figura 4: Imagen con ruido

Ejercicio 4: A continuación, vamos a añadir a la imagen un efecto destello. Represente la máscara obtenida.

El siguiente código implica la selección de parámetros, así como la expresión matemática aportada en el enunciado de la práctica:

% Máscara del destello

```
[m, n, ~] = size(I);
Cx = round(n*0.90);
Cy = round(m*0.15);
r = min(m, n) / 4;
[x, y] = meshgrid(1:n, 1:m);
mascara = 0.5 * exp(-((x - Cx).^2 + (y - Cy).^2) / (2 * r^2));
```

Vemos en la figura 5 que la máscara es lo más parecida posible a la indicada en el enunciado.



Figura 5: Máscara para el efecto de destello

Ejercicio 5: Añada la máscara a la imagen procesada. Represente el resultado.

Bastará con sumar la máscara a los 3 canales de nuestra imagen de la siguiente manera:

```
% Añadimos la máscara a la imagen con ruido gaussiano
```

```
I_destello = I_ruido + repmat(mascara, [1, 1, 3]);
```

```
I_destello = min(I_destello, 1); % Limitar a [0, 1]
```

Y como era de esperar, la imagen resultante será la que teníamos con ruido, tono claro y desaturada, pero con un destello correspondiente a la máscara anterior. Podemos comprobarlo en la figura 6.

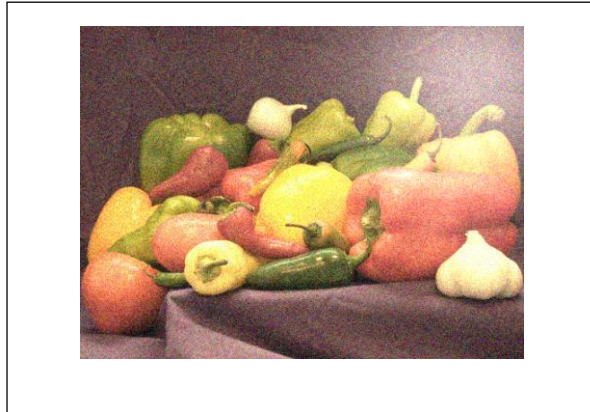


Figura 6: Imagen con efecto destello

Ejercicio 6: Aumente el brillo de la imagen resultante en un factor entre un 10-20%. Represente la imagen realzada.

Multiplicando la imagen por el factor de brillo, en concreto el tercer canal tras pasarla a HSV, seremos capaces de modificarlo. Aumentar el brillo entre 10% y 20% significa que el factor debe estar entre 1.1 y 1.2.

Así pues, el código será el siguiente. Las imágenes con brillo aumentado un 10% y un 20% las vemos en las figuras 6 y 7 respectivamente.

```
% Aumentamos el brillo
```

```
factor_brillo = 1.2; % Modificamos este parámetro ente 1.1 y 1.2
```

```
I_brillo = rgb2hsv(I_destello);
```

```
I_hsv(:, :, 3) = I_hsv(:, :, 3) * factor_brillo;
```

```
I_brillo = hsv2rgb(I_brillo);
```

```
I_brillo = min(I_brillo, 1); % Limitamos
```



Figura 7: Imagen con 10% más de brillo

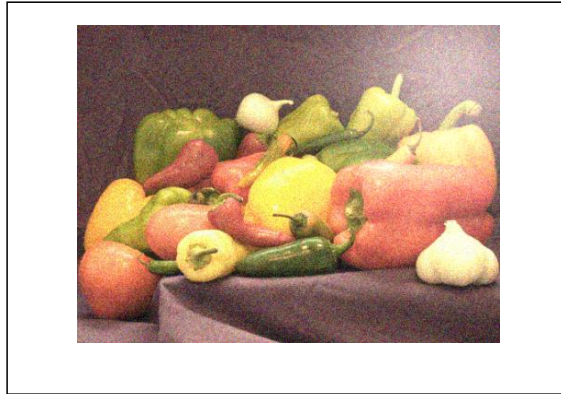


Figura 8: Imagen con 20% más de brillo

Ejercicio 7: Utilice la orden `imadjust` para aplicar una corrección de a la imagen anterior en un factor de +10% y represente la imagen final.

% Corrección de gamma con `imadjust`

`factor_gamma = 1.1;`

`I_gamma = imadjust(I_brillo, [], [], factor_gamma);`

Al aplicar una corrección de gamma, conseguimos compensar la falta de luminosidad proyectada en la pantalla para así obtener un balance de colores en la foto que hace que tenga un mejor parecido a la realidad.

Ese es el código que se nos pide utilizar, y el resultado obtenido es el que aparece en la figura 8.

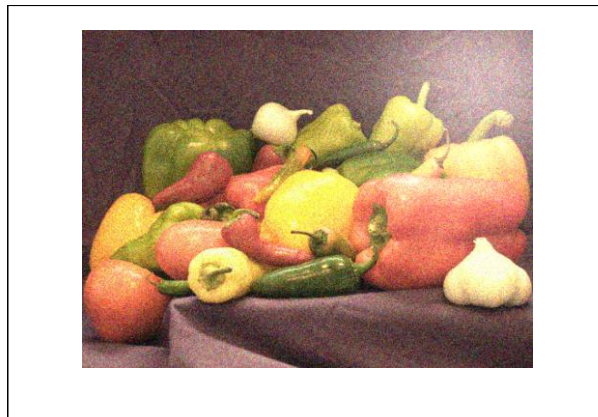


Figura 9: Imagen con corrección de gamma del 10%

Como vemos con `imadjust` hemos podido ajustar los valores de intensidad de nuestra imagen, en este caso aplicando una corrección de gamma del 10%.

Ejercicio 8: Obtenga de nuevo el resultado, pero llevando a cabo un realce de bordes de la imagen antes de corregir su brillo y gamma. Extraiga las oportunas conclusiones sobre las diferencias entre ambos resultados finales.

Realzamos los bordes como se nos pide:

```
% Realzamos los bordes
```

```
I_bordes = imsharpen(I_ruido, 'Radius', 2, 'Amount', 1.5);
```

```
% Aplicamos brillo y gamma
```

```
I_final = imadjust(I_bordes * factor_brillo, [], [], factor_gamma);
```

Ahora vemos el resultado obtenido en la figura 9.

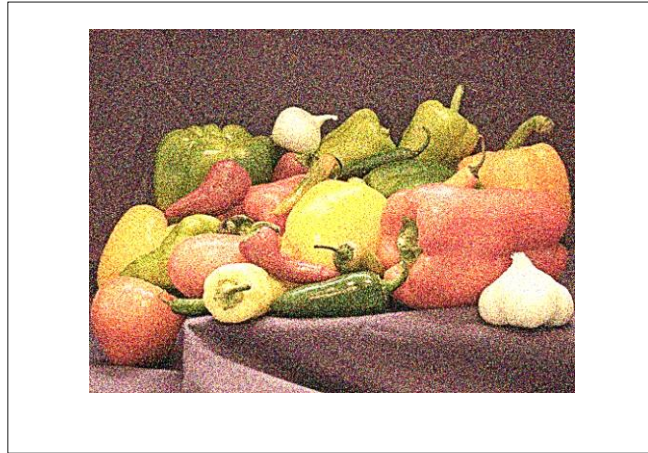


Figura 10: Imagen final con bordes realzados

Como podemos comprobar, al realzar los bordes antes de aplicar una corrección de gamma como en la figura anterior, podemos observar que el destello es apenas apreciable a diferencia del apartado anterior.

Podemos afirmar que para hacer una corrección del brillo y gamma de una imagen es interesante tener en cuenta el realce de bordes que nos permite en gran medida hacer desaparecer el destello aplicado.