Intro. to Image Processing HW1

110550093 蔡師睿

Method

Exchange Position

Swap the matrix represented the upper left picture with the matrix represented the upper right picture.

Gray Scale

Change the channel values of every pixel in the lower right picture by using the method : $R = G = B = \frac{(R+G+B)}{3}$. Note that before calculate, we've to do type casting, changing **uint8** to **int**; otherwise, it will cause overflow if $R + G + B > 255$.

Intensity Resolution

Before we do intensity resolution (256 $\rightarrow$ 4), we have to do gray scale again. Then, due to $\frac{256}{4} = 64$, I divide 64 into the new channel values, cast the new channel values from **float** to **int**, and multiply the new channel values by 64 to get the result.

Color Filter – Red

According to the hint, we've to keep the pixel wich is $R > 150$ and $R * 0.6 > B$ and $R * 0.6 > G$; thus, I turn the pixel into gray scale using the formula mentioned above, $R = G = B = \frac{(R+G+B)}{3}$, if $R \leq 150$ or $R * 0.6 \leq B$ or $R * 0.6 \leq G$.

Color Filter – Yellow

According to the hint, we've to keep the pixel which is $(G + R) * 0.3 > B$ and $|G - R| < 50$; thus, I turn the pixel into gray scale using the formula mentioned above, $R = G = B = \frac{(R+G+B)}{3}$, if $(G + R) * 0.3 \leq B$ or

$|G - R| \geq 50$.
Similarly, due to the addtion, we've to do type casting, changing **uint8** to **int**.

Channel Operation

First, I check whether the channel value which index is 1 larger than 127 or not; if so, the value is assigned to 255, and the other values not greater than 127 are multiplied by 2.

Bilinear Interpolation

From the matrix column and row (i, j) , let
$x_f = \frac{(j+0.5)}{2} - 0.5$, $y_f = \frac{(i+0.5)}{2} - 0.5$,
$x_i = min(\lfloor x_f \rfloor, length\ of\ row - 2)$, and
$y_i = min(\lfloor y_f \rfloor, length\ of\ column - 2)$. Because we don't want the x_i and y_i to be out of border.
Second, I let $dx = x_f - x_i$ and $dy = y_f - y_i$, and dx and dy mean to be the ratio from the point to its upper right neighbor point.
Third, assume the neighbor point at upper right is at (x_0, y_0) , and the point we want to calculate is at (x, y) ; then, apply the formula :

$$\begin{gathered}
f(x, y) = (1 - dx) \times (1 - dy) \times f(x_0, y_0) + \\
dx \times (1 - dy) \times f(x_0 + 1, y0) + \\
(1 - dx) \times dy \times f(x_0, y_0 + 1) + \\
dx \times dy \times f(x_0 + 1, y_0 + 1)
\end{gathered}$$

Bicubic Interpolation

The first step is same as we do in the bilinear interpolation, which we get x_f, x_i, dx, y_f, y_i, and dy.
Next, we caculate and get its 16 neighbor points to do interpolation. If the point is out of border, we can view the value of the point as 0.

Then, we can apply the formula :

$$f(p_0, p_1, p_2, p_3, x) = (-\frac{1}{2}p_0 + \frac{3}{2}p_1 - \frac{3}{2}p_2 + \frac{1}{2}p_3)x^3 + \\ (p_0 - \frac{5}{2}p_1 + 2p_2 - \frac{1}{2}p_3)x^2 + \\ (-\frac{1}{2}p_0 + \frac{1}{2}p_2)x + p_1$$

, which p_0, p_1, p_2 and p_3 represent the values at $x = -1$, $x = 0$, $x = 1$, and $x = 2$, respectively.
Because of the 16 points, we can get four cubic functions, and then apply the formula again to get a new cubic function cross the four lines. Finally, we can get the value from the last cubic function.

Result

Feedback

Bicubic is difficult to understand at first, but it's a brilliant method.