

Practica 2: Dithering

Aguilera Palacios Luis Ernesto, leapvader.1998@hotmail.com
 Padilla Castillo Aaron Samir, samir.castill@gmail.com,
 Facultad de Ingeniería, Universidad Nacional Autónoma de México

Resumen—En esta practica aprenderemos sobre la tecnica de ‘Dither’ (tramado) en el procesamiento digital de imágenes, implementaremos los algoritmos algunos de los metodos mas comunes y al final haremos una comparacion con cada uno de los resultados.

Abstract—In this school work we’ll learn about Dithering in digital image processing while implementing some basic MATLAB codes. At the end there’s going to be a comparison of each result.

I. INTRODUCCIÓN

El tramado o Dithering es una técnica usada en computación gráfica para crear la ilusión de profundidad de color en imágenes con una paleta de colores limitada (reducción de color). En una imagen tramada, los colores no disponibles en la paleta se aproximan por una difusión de píxeles de color dentro de la gama de colores disponibles. El ojo humano percibe la difusión como una mezcla de los colores dentro de ésta (ver percepción del color). El tramado es análoga a la técnica denominada semitonos utilizada en impresión. Las imágenes tramadas, en particular las que tienen relativamente pocos colores, a menudo se distinguen por un grado de granulosidad característico o por un aspecto moteado. El tramado se usa habitualmente en el procesamiento de datos de audio y video digital, y a menudo es una de las últimas etapas para la masterización del audio en un CD. [1]

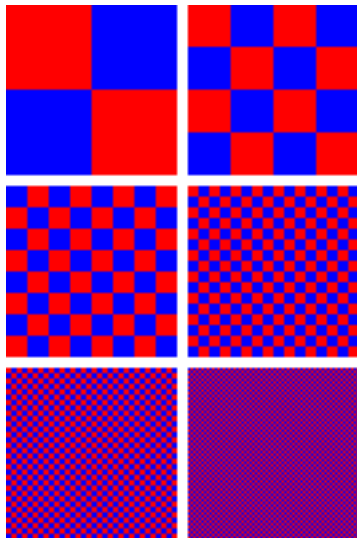


Fig. 1. Ejemplo de dithering.

La reducción de la profundidad de color de una imagen a menudo puede tener importantes efectos secundarios visuales. Si la imagen original es una fotografía, es probable que tenga miles o incluso millones de distintos colores. El proceso de limitar los colores disponibles a una *paleta de colores* específica efectivamente quita una cierta cantidad de información de color.

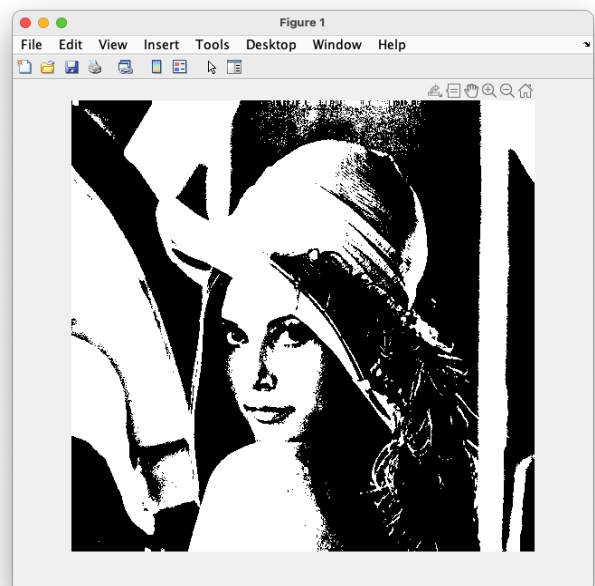
Un gran número de factores puede afectar la calidad resultante de una imagen en color reducido. Tal vez el más significativo es la paleta de colores que se utilizará en la imagen reducida. Por ejemplo, una imagen original podrá reducirse a 216 colores. Si los colores de los píxeles originales son simplemente traducidos al color más próximo disponible en la gama de colores, no se produce tramado. Por lo general, este enfoque produce zonas planas (contornos) y una pérdida de detalle, y puede producir manchas de color que son significativamente diferentes del original.

II. DESARROLLO

Implementación de métodos de dithering.

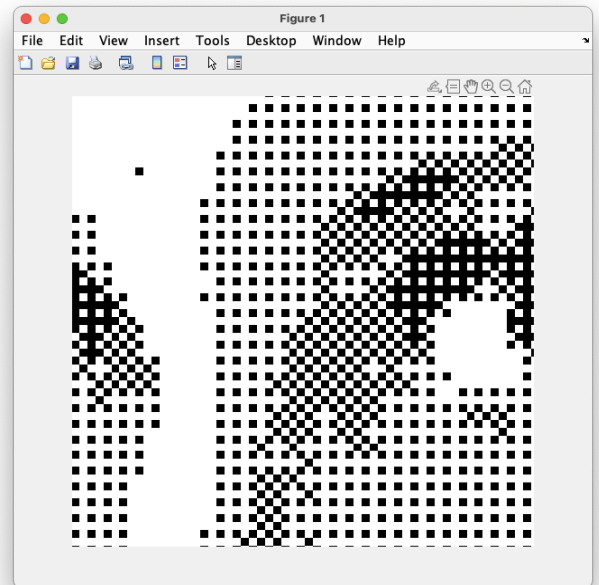
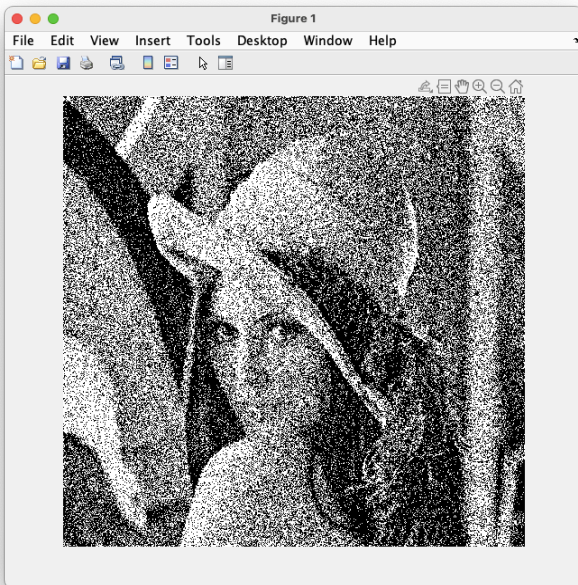
a) Cuantice una imagen con 1 bit de profundidad

```
%un bit de profundidad
im = imread('lenna.jpeg');
[m n] = size(im);
mat = 128*ones(m, n);
unbit = im > mat; %1(blanco) si im > mat
imshow(unbit);
```



b) Implemente el método de dithering aleatorio

```
%un bit de profundidad con dithering aleatorio
im = imread('lenna.jpeg');
[m n] = size(im);
mat = randi([0 255],m,n);
ditheredA = im > mat; %1(blanco) si im > mat
imshow(dithered);
```

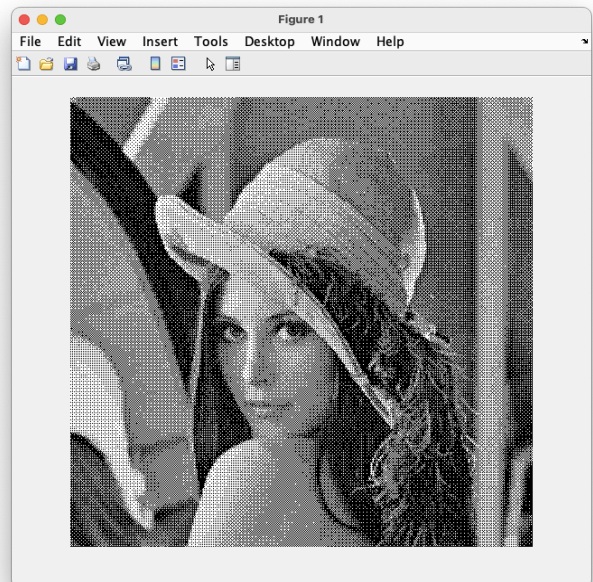
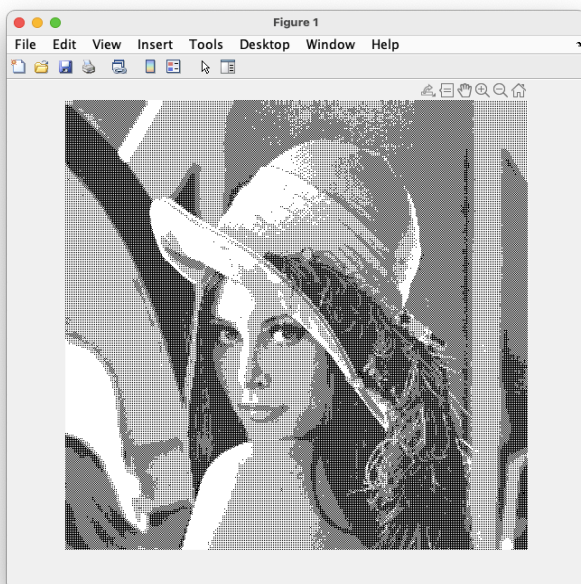


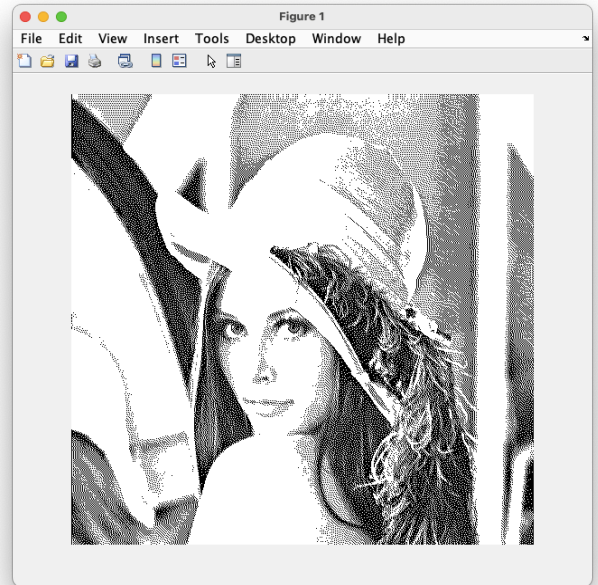
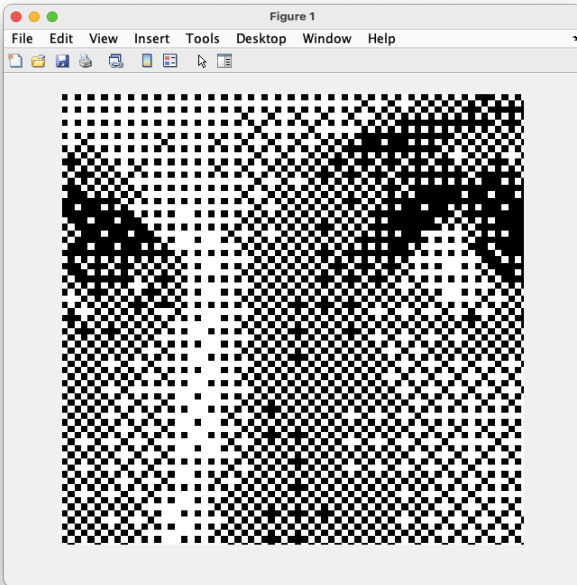
d) Implemente el método con matrices de 4x4

```
% dithering con matrices 4x4
im = imread('lenna.jpeg');
di = [0 8 2 10; 12 4 14 6; 3 11 1 9; 15 7 13 5] %matriz dithering
[m n] = size(im);
mat = repmat(di, m/4, n/4); %repetimos matriz dithering en escala
de la im original
im = im / 17;
dithered4 = im > mat; %decision para pintar uno de los 5 grises
imshow(dithered);
```

c) Implemente el método con matrices de 2x2

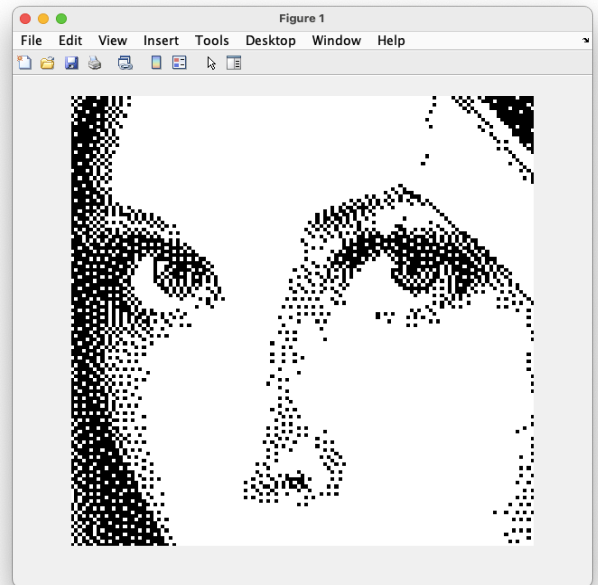
```
%dithering con matrices 2x2
im = imread('lenna.jpeg');
di = [64 128; 192 0] %matriz dithering
[m n] = size(im);
mat = repmat(di, m/2, n/2); %repetimos matriz dithering en escala
de la im original
dithered2 = im > mat; %decision para pintar uno de los 5 grises
imshow(dithered);
```





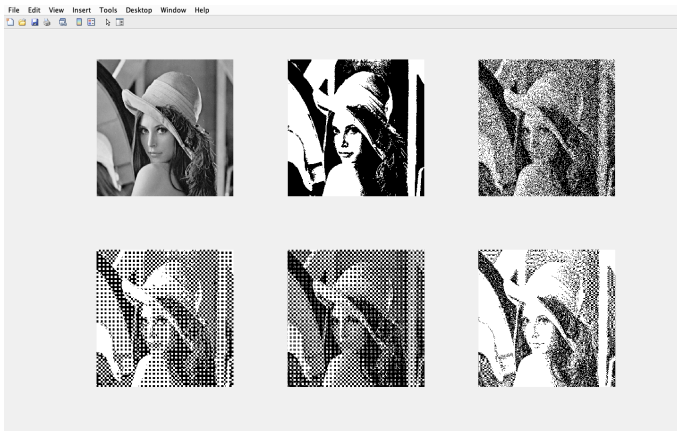
e) Implemente el método de dithering con difusión del error de Floyd-Steinberg [2]

```
%dithering con difusión del error de Floyd-Steinberg
ima = imread('lenna.jpeg');
s=size(ima);
for i = 1 : s(1)
    for j = 1 : s(2)
        if(ima(i,j) < 127)
            imb(i,j)=0;
        else
            imb(i,j) = 255;
        end;
        qerror = ima(i,j) - imb(i,j);
        if(j < s(2))
            ima(i,j+1) = ((7/16 *qerror)+ima(i,j+1));
        end;
        if(i<s(1) && j > 1)
            ima(i+1,j-1) = ima(i+1,j-1) + (3/16 *qerror);
            ima(i+1,j) = ima(i+1,j) + (5/16 *qerror);
        end;
        if(j<s(2) && i<s(1))
            ima(i+1,j+1) = ima(i+1,j+1) + (1/16 *qerror);
        end;
    end;
end;
ditheredFS = imb;
imshow(ditheredFS);
```



f) Compare los resultados de los 5 métodos

```
%todo
im = imread('lenna.jpeg');
if true
    figure;
    subplot(2,3,1),imshow(im);
    subplot(2,3,2),imshow(unbit);
    subplot(2,3,3),imshow(ditheredA);
    subplot(2,3,4),imshow(dithered2);
    subplot(2,3,5),imshow(dithered4);
    subplot(2,3,6),imshow(ditheredFS);
end
```



III. CONCLUSIONES

En esta practica hemos aprendido cinco de los algoritmos basicos para aplicar la tecnica de dither a imágenes en escala de grises, aplicar este tipo de tecnicas nos ayudan a reducir la profundidad de bits de la matriz/imagen a nuestra conveniencia, dependiendo de cual sea nuestro objetivo, nos puede ser de ayuda para mejorar la eficiencia de la imagen para despues procesarla con otras herramientas. Cabe agregar que su funcionamiento es muy parecido a un filtro de otra indole.

REFERENCIAS

- [1] Anónimo. Tramado. [online] Disponible: <https://es.wikipedia.org/wiki/Tramado>
- [2] Praveen Settipalli (2020). Error Diffusion Algorithm (<https://www.mathworks.com/matlabcentral/fileexchange/6953-error-diffusion-algorithm>), MATLAB Central File Exchange. Retrieved October 27, 2020.