

Ajuste por histogramas

Tomas Lopez Perez

April 21, 2024

1 Introducción

El procesamiento de imágenes digitales se utiliza para mejorar la calidad de las imágenes y extraer información valiosa de ellas. Esto puede implicar técnicas como la mejora de la imagen, la segmentación de la imagen y la transformación de la imagen. Estas técnicas se utilizan en una variedad de campos [HST71], incluyendo el reconocimiento de patrones y la codificación eficiente de imágenes.

2 Fundamento teórico

2.1 Procesamiento de imágenes

El Procesamiento de imágenes es un campo de investigación muy extenso que involucra diversas áreas del conocimiento, ya que involucra diversos procesos, tales como: la adquisición, transmisión, representación y procesamiento. En términos generales el procesamiento de imágenes se utiliza para modificar una imagen con el fin de mejorar la apariencia visual para un observador y para resaltar convenientemente el contenido de la imagen de cara a la percepción por parte de máquinas, es decir, se hace una manipulación de imágenes con objeto de producir nuevas imágenes que son mejores, en algún sentido. El procesamiento de imágenes comprende distintas técnicas que se utilizan para mejorar, restaurar e inclusive comprimir imágenes ya existentes; la implementación y desarrollo de dichas técnicas agregadas a sistemas ya existentes nos permiten editar imágenes de forma más sencilla y precisa [Alo18]. La Figura 1 muestra el flujo de las etapas del procesamiento digital de imágenes propuestas por Gonzalez en donde

se establece como la etapa inicial la adquisición de la imagen digital, lo que se puede hacer mediante algún sistema de sensores que permitan digitalizar la imagen u obtener las imágenes de algún repositorio o colección. El conjunto de imágenes adquiridas conforma inicialmente la base de conocimiento, ya que para algunas aplicaciones basta con tener la información pura, sin embargo en la mayoría de las aplicaciones del procesamiento de imágenes, es necesario enriquecer la base del conocimiento con al menos las etapas de procesamiento y segmentación, lo cual requiere aplicar una serie de operadores que transformen las imágenes para resaltar características relevantes, en el caso del procesamiento, o separar los diferentes elementos que componen la imagen, en el caso de la segmentación, de modo que los resultados de estas dos etapas permiten tener una base del conocimiento más amplia y robusta. Respecto a la etapa de representación y descripción, las imágenes procesadas y/o segmentadas son usadas para extraer características más específicas sobre la imagen, tales como cantidad, dimensión, forma, posición, etc. de los objetos de interés en las imágenes, esta información también puede formar parte de la base del conocimiento, de modo que en la última etapa, reconocimiento e interpretación, se puedan obtener respuestas a partir de las imágenes y la información generada por estas.

2.2 Histograma

El histograma [FDD05] es un diagrama de barras cuyas abscisas representan los niveles de gris de una imagen, y las ordenadas, las frecuencias relativas de los distintos niveles de gris, es decir, la cantidad de puntos asociados a cada nivel de gris, dividido por

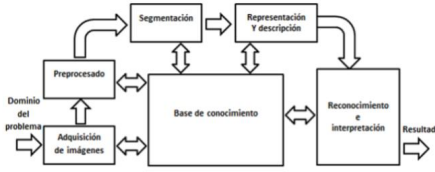


Figure 1: Etapas fundamentales del procesamiento digital de imágenes.

las cantidad total de puntos de la imagen.

$$P_k = \frac{n_k}{n}$$

$$n_k =$$

cantidad de pixeles de nivel gris k

$$n =$$

cantidad total de pixels

En la figura 2 se observa un histograma estrecho, de bajo rango dinámico, en concordancia con el poco contraste visual de la imagen correspondiente.

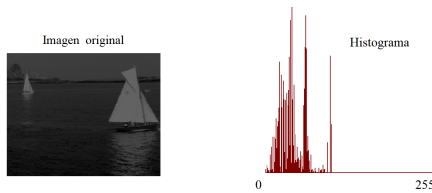


Figure 2: Histogram

2.3 Ecualización del histograma

La ecualización de histograma es una forma de mejorar una imagen determinada. El enfoque consiste en diseñar una transformación $T(.)$ tal que los valores de gris en la salida se distribuyen uniformemente en $[0, 1]$. Necesitamos diseñar una transformación de valor de gris $s = T(r)$, basado en el histograma de la imagen de entrada, que mejorará la imagen. La ecualización del histograma produce una imagen cuyos píxeles están (en teoría) distribuidos uniformemente entre todos los niveles de gris.[PS19]

2.4 Especificación de histograma

La ecualización del histograma produce una imagen cuyos píxeles están (en teoría) distribuidos uniformemente entre todos los niveles de gris. A veces, esto puede no ser deseable. En cambio, es posible que deseemos una transformación que produzca una imagen de salida con un histograma preespecificado. Esta técnica se llama especificación de histograma. La información dada es, Imagen de entrada a partir de la cual podemos calcular su histograma e histograma deseado. Con la ayuda proporcionada, la información deriva una operación puntual que asigna la imagen de entrada a una Imagen de salida que tiene el histograma especificado por el usuario.[PS19]

3 Metodología

En esta sección se presenta una metodología que abarca tanto la ecualización como la especificación de histograma, dos técnicas fundamentales en el procesamiento de imágenes. Ambas técnicas permiten ajustar la distribución de intensidades de una imagen para mejorar su contraste y calidad visual. La ecualización de histograma mejora el contraste redistribuyendo las intensidades de los píxeles en una imagen. Por otro lado, la especificación de histograma ajusta la distribución de intensidades según una referencia deseada, como una función matemática, una función por tramos o incluso otra imagen. Se describen los pasos clave para llevar a cabo estas implementaciones, desde la preparación de los datos hasta la validación de los resultados obtenidos. Esta metodología proporcionará una guía clara y detallada para aquellos que busquen mejorar la calidad visual de sus imágenes de manera efectiva y controlada en el campo del procesamiento de imágenes.

1. **Preparación de datos:** Recolectar datos de entrada, incluyendo imágenes de trabajo y referencias para la especificación de histograma.
2. **Preprocesamiento de imágenes:** Ajustar las imágenes según sea necesario, como la conversión a escala de grises.

3. **Cálculo de histograma:** Determinar la distribución de intensidades de la imagen de entrada mediante el cálculo del histograma.
4. **Selección de la referencia de histograma:** Definir la referencia deseada para el histograma, como una función matemática, una función por tramos u otra imagen.
5. **Mapeo de histograma:** Desarrollar un algoritmo para mapear los niveles de intensidad de la imagen de entrada a los niveles de intensidad de la referencia.
6. **Evaluación y ajuste:** Comparar visualmente la imagen original y la transformada para evaluar los cambios en el contraste, y ajustar parámetros si es necesario para mejorar los resultados.

4 Resultados

Para la implementación de la ecualización de Histogramas se utiliza la implementación de las tablas Lookup Table figura 14, y aplicando un factor multiplicando cada una de las intensidades.

La figura 3 es la imagen en donde aplicaremos los cambios. Para la especificación aplicando \cos , x^2 y \log se puede observar los cambios en la figura 7 5 10. Igual que lo anterior se realiza el uso de Lookup Table y aplicando \cos , x^2 y \log nos da diferentes resultados.

Para implementar el algoritmo Piecewise, se utilizó el siguiente vector, como se muestra en la figura 16. Este vector ayuda al algoritmo a determinar en qué fragmento del vector se asignarán los valores para crear la nueva imagen.

x_change = 123456

y_values = 0.70.80.840.880.91

Para implementar el algoritmo "match", se utiliza la imagen original 3 junto con la imagen de referencia 11. Este algoritmo combina estas dos imágenes para generar una nueva imagen, como se observa en la figura ???. La función "match" toma como entrada la imagen original y la imagen de referencia y produce diferentes resultados dependiendo de la imagen

de referencia utilizada. El código de implementación del algoritmo se encuentra en la figura 17.

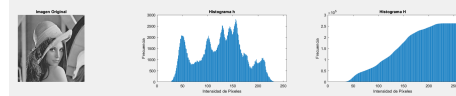


Figure 3: Original

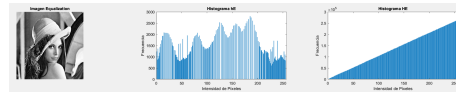


Figure 4: Ecualizacion

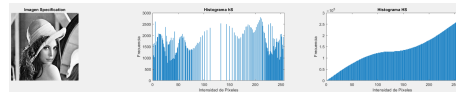


Figure 5: Especificacion cos

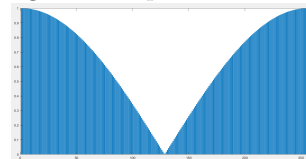


Figure 6: Cos

5 Conclusiones

En conclusión, hemos examinado una variedad de técnicas de procesamiento de imágenes que amplían nuestras capacidades para mejorar la calidad visual y la comprensión de la información contenida en ellas. Desde la potente ecualización de histograma, que resalta detalles mediante el ajuste del contraste y la distribución de intensidades, hasta la especificación de histograma, que ofrece un control preciso a través de funciones matemáticas, incluyendo aquellas por tramos que permiten adaptar transformaciones a diferentes rangos de intensidades. Además, el histogram matching emerge como

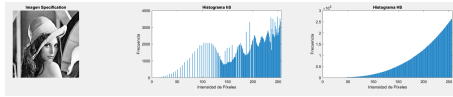


Figure 7: Especificacion x^2

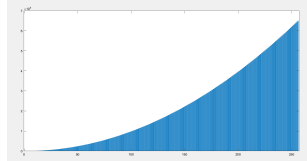


Figure 8: Especificacion x^2

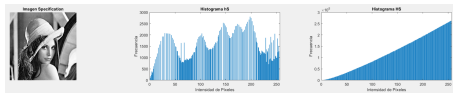


Figure 9: Especificacion Log

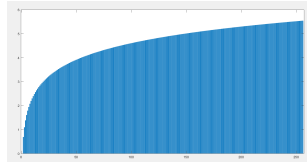


Figure 10: Especificacion Log

una herramienta versátil que facilita la transferencia de distribuciones de intensidades entre imágenes, promoviendo la consistencia visual o ajustando la apariencia según nuestras necesidades. Estas técnicas no solo mejoran la calidad visual de las imágenes, sino que también enriquecen su interpretación, posicionándose como recursos fundamentales en el procesamiento de imágenes con impacto significativo.

References

- [HST71] T.S. Huang, W.F. Schreiber, and O.J. Tretiak. “Image processing”. In: *Proceedings of the IEEE* 59.11 (1971), pp. 1586–1609. DOI: 10.1109/PROC.1971.8491.
- [FDD05] Luis A Fernández, Daniel Diaz, and Roberto Depaoli. “Optimización de la ecualización del histograma en el proce-

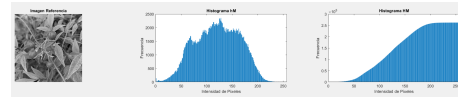


Figure 11: Imagen referencia

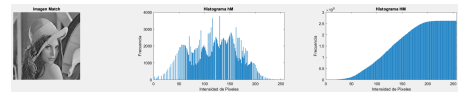


Figure 12: Imagen matching

samiento de imágenes digitales”. In: *VII Workshop de Investigadores en Ciencias de la Computación*. 2005.

- [Alo18] Adan Antonio Alonso-Ramírez. *Implementación de la transformada de Hough en tecnología GPU*. 2018.
- [PS19] Dr Tarun Gupta PriyankaTyagi and Abhishek Singh. “Histogram specification: A Review”. In: *M. Tech, Computer Science & Engineering, AKTU Lucknow RGG. I Meerut, UP, India, IRJET* 6 (2019).

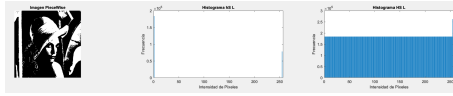


Figure 13: Imagen piece wise

```
function imgE= HEqualization(H, M, N, k, img)
    factor = k/ (M * N);
    LUT = zeros(256,1);
    for intensidad=1:256
        LUT(intensidad) = floor( H(intensidad) * factor);
    end
    imgE = uint8(zeros(size(img)));
    for r=1: size(img,1)
        for c =1: size(img, 2)
            imgE(r,c) = uint8(LUT(img(r,c)));
        end
    end
end
```

Figure 14:Codigo ecualizacion

```
function img2 = HSpecification(H,ref ,img1)
    refC = zeros(256,1);
    for bin=2:256
        refC(bin) = refC(bin-1) + ref(bin);
    end
    %Normalizamos datos en un rango de 0,1
    refC = refC / refC(end);
    H = H / H(end);
    tabla = zeros(256,1);

    for intensidad=0:255
        %Generamos las busquedas de los indices a los que le pertenece la
        %intensidad
        res = find(H(intensidad + 1 ) >= refC);
        tabla(intensidad + 1) = res(end) -1;
    end

    img2 = uint8(zeros(size(img1)));

    for r = 1:size(img1,1)
        for c = 1:size(img1, 2)
            img2(r,c) = uint8(tabla(img1(r, c) + 1));
        end
    end
end
```

Figure 15:Codigo especificacion

```
function img2 = HSpecificacion(HA, HR , img1)
    K = numel(HA);
    PA = HA / HA(end);
    PR = HR / HR(end);
    fhs = zeros(1, K);

    for a = 0:(K-1)
        b = PA(a + 1);

        if b <= LR(1,2)
            a_prime = 0;
        elseif b >= 1
            a_prime = 255;
        else
            %RANGEA DESDE 1
            n = numel(LR(1, 2)) - 1;
            while n >= 0 && LR(n + 1, 2) > b
                n = n - 1;
            end
            a_prime = LR(n+1, 1) + (b - LR(n+1, 2)) * (LR(n + 2, 1) - LR(n+1, 1)) / (LR(n + 2, 2) - LR(n+1, 2));
        end
        disp(a);
        fhs(a + 1) = uint8(a_prime * 255);
    end
    img2 = uint8(zeros(size(img1)));

    for r = 1:size(img1,1)
        for c = 1:size(img1, 2)
            img2(r,c) = uint8(fhs(img1(r, c)));
        end
    end
end
```

Figure 16:Codigo PieceWise

```
function img2 = match(HA, HR , img1)
    % HA: histograma de la imagen objetivo
    % HR: histograma de referencia (del mismo tamaño que HA)

    K = numel(HA);
    PA = HA;
    PR = HR;
    fhs = zeros(K, 1);
    for a = 1:K
        j = K;
        while (j >= 1) && (PA(a) <= PR(j))
            fhs(a) = j;
            j = j - 1;
        end
    end
    img2 = uint8(zeros(size(img1)));
    for r = 1:size(img1,1)
        for c = 1:size(img1, 2)
            img2(r,c) = uint8(fhs(img1(r, c)));
        end
    end
end
```

Figure 17:Codigo Match