

Restauración de imágenes digitales usando método de Metrópolis de Montecarlo Márkov Chain.

Oscar De la Cruz Echeveste

División de ciencias e ingenierías
Universidad de Guanajuato

Resumen Se ha implementado la técnica de Montecarlo Markóv Chain para llevar a cabo la restauración de imágenes digitales en escalas de grises. Para tales fines se ha seguido a K. Murthy en su texto sobre Restauración bayesiana de imágenes digitales [1] realizando un código en Python que se a implementado para un par de imágenes con diferentes escalas de grises corrompiéndolas con 5%.

1. Introduction

La restauración de imágenes digitales es un proceso de obtención de una imagen limpia a partir de una generalmente borrosa y ruidosa. La restauración de imágenes es un problema fundamental en el procesamiento de imágenes. El objetivo de las técnicas de restauración de imagen es reducir el ruido y recuperar la pérdida de resolución. La técnica más sencilla y convencional para la restauración de imágenes es la deconvolución, que se realiza en el dominio de la frecuencia para después calcular la transformación de Fourier tanto de la imagen como de la función de dispersión de puntos (PSF) y así, deshace la pérdida de resolución causada por los factores de desenfoque. Esta técnica de deconvolución, debido a su inversión directa del PSF, que generalmente tiene un número de condición de matriz pobre, amplifica el ruido y crea una imagen borrosa imperfecta. Por lo que se han desarrollado técnicas más sofisticadas para estos fines.

En el presente trabajo se ha seguido el artículo de K. Murthy sobre Restauración bayesiana de imágenes digitales para emplear las técnicas de Montecarlo Markov Chain. Se construyó un código en Python con las indicaciones específicas de Murthy a lo largo del artículo. A continuación se presenta un resume de dicho artículo con los puntos más importantes de este y con los cuales se llevó a cabo el código. Además, se muestran los resultados al ser implementado con una imagen sencilla.

2. Marco teórico

2.1. Descripción de una imagen

Una imagen podemos definirla como la región de un plano la cual está compuesta de pequeños cuadros llamados Pixel (picture element) cuya variaciones

en su escala de colores forman patrones y figuras. Para una representación matemática de una imagen definiremos la región espacial de un pixel en plano imagen como i . La colección $\Theta = \{\theta_i : i \in S\}$ la llamaremos imagen donde S es el conjunto finito de pixeles que forman la imagen. θ_i denotará la intensidad de los pixeles (escala de grises). Además, denotamos $\hat{\Theta} = \{\hat{\theta} : i \in S\}$ como la imagen verdadera, es decir, la imagen sin ser dañada con ruido. En un caso real, $\hat{\Theta}$ será desconocida por nosotros. Por otro lado tendremos, la imagen con ruido $X = \{x_i : i \in S\}$. El objetivo del método que se implementará será obtener una aproximación de la imagen verdadera $\hat{\Theta}$ dada una imagen ruidosa X .

Al trabajar sólo con imágenes en escala de grises definiremos 0 como el nivel más bajo (blanco) y Q el nivel más alto (Negro). De esta forma, una imagen de N pixeles será construida con Q niveles de grises discretos $\{0, 1, 2, \dots, Q-1\}$.

Se construirá un modelo estocástico de una imagen X para definir una probabilidad condicional de tener X dada la imagen verdadera $\hat{\Theta}$. Esta probabilidad será el likelihood el cual puede ser usado para describir el proceso estocástico de degradación para producir X desde $\hat{\Theta}$. Consideraremos un modelo general de degradación de imagen en estadística de Poisson.

2.2. Distribución de Poisson como Likelihood

Tenemos un arreglo aleatorio X . Tomaremos un elemento $x_i \in X$ como una variable aleatoria de Poisson. Esto debido a que la adquisición de la imagen es un proceso de contado (contando los fotones incidentes en una fotografía). Tomando como promedio $\langle x_i \rangle = \hat{\theta}_i$ la distribución será:

$$\mathcal{L}(x_i|\hat{\theta}_i) = \frac{1}{x_i!} \left(\hat{\theta}_i\right)^{x_i} \exp(-\hat{\theta}_i) \quad (1)$$

De esta forma, la distribución de todo el conjunto X dado $\hat{\Theta}$ será el producto de todas las distribuciones de las variables aleatorias individuales. Es decir:

$$\mathcal{L}(X|\hat{\Theta}) = \prod_{i \in S} \mathcal{L}(x_i|\hat{\theta}_i) = \prod_{i \in S} \frac{1}{x_i!} \left(\hat{\theta}_i\right)^{x_i} \exp(-\hat{\theta}_i) \quad (2)$$

Por conveniencia podemos escribir la expresión anterior como:

$$\mathcal{L}(X|\hat{\Theta}) = e^{F(\hat{\Theta}, X)} \prod_{i \in S} \frac{1}{x_i!}$$

Donde

$$F(\hat{\Theta}, X) \equiv \sum_{i \in S} \left(\hat{\theta}_i - x_i \ln \hat{\theta}_i\right) \quad (3)$$

Esta función $F(\Theta, X)$ cuantifica que tan lejos está la imagen Θ de otra imagen X . Esta función podemos extenderla para que encaje a la definición de distancia métrica.

2.3. Distancia Kullback-Leibler

De la definición 3, vemos que $F(\Theta, X) \neq 0$ cuando $\Theta = X$, además que no es simétrica en sus argumentos. Rescribiremos la función F como:

$$F(\hat{\Theta}, X) = \sum_{i \in S} f(\theta_i, x_i) \quad (4)$$

De esta manera estudiaremos el comportamiento de $f(\theta_i; x_i) = \theta_i - x_i \ln \theta_i$ como función de θ_i con el valor x_i conocido. Así, tenemos que:

$$\frac{df}{d\theta_i} = 1 - \frac{x_i}{\theta_i} = 0$$

Por lo que $f(\theta_i, x_i)$ es mínimo en $\theta_i = x_i$. Definimos $f_m = x_i - x_i \ln(x_i)$ como el mínimo valores de $f(\theta_i, x_i)$. De esta manera, definimos una nueva función:

$$\begin{aligned} g(\theta_i, x_i) &= f(\theta_i, x_i) - f_m \\ \Rightarrow g(\theta_i, x_i) &= (\theta_i - x_i) + (\theta_i - x_i) \ln(\theta_i/x_i) \end{aligned} \quad (5)$$

Esta función no es simétrica en sus argumentos. Entonces definimos:

$$\begin{aligned} F(\theta_i, x_i) &= g(\theta_i, x_i) + g(\theta_i \rightarrow x_i, x_i \rightarrow \theta_i) \\ \Rightarrow F(\theta_i, x_i) &= (\theta_i - x_i) \ln(\theta_i/x_i) \end{aligned} \quad (6)$$

Ahora podemos definir una distancia métrica entre Θ y X como:

$$F(\Theta, X) = \sum_{i \in S} (\theta_i - x_i) \ln(\theta_i/x_i) \quad (7)$$

A esta función de distancia se le conoce por el nombre de distancia de Kullback-Leibler.

2.4. Distancia de Hamming

Una alternativa para cuantificar la distancia entre dos imágenes Θ y X es la llamada distancia de Hamming, la cual se define como:

$$F(\hat{\Theta}, X) = \sum_{i \in S} \mathcal{I}(\theta_i \neq x_i) \quad (8)$$

Donde:

$$\mathcal{I}(\eta) = \begin{cases} 1 & \text{si } \eta \text{ Verdad} \\ 0 & \text{si } \eta \text{ Falso} \end{cases} \quad (9)$$

Esta función de distancia cuenta el número de píxeles en Θ que tienen un nivel de gris diferente al píxel de X correspondiente. Vemos, además, que cumple con todas las características de una distancia métrica.

2.5. Likelihood para imágenes binarias

Dada una imagen binaria (imagen con sólo dos niveles de grises, blanco y negro) $\hat{\theta}$ la corromperemos con ruido de la siguiente forma. Tomamos el primer pixel $i \in S$ del plano imagen y generamos un número aleatorio ϵ de una distribución uniforme entre cero y uno. Si $\epsilon \leq \hat{p}$ cambiamos el nivel de gris del pixel seleccionado. De otra manera este no sufrirá cambios. Este proceso se ejecuta para cada uno de los pixeles de la imagen. Lo anterior es descrito por el likelihood dado por:

$$\mathcal{I}(x_i|\hat{\theta}_i) = \hat{p}\mathcal{I}(x_i \neq \hat{\theta}_i) + (1 - \hat{p})\mathcal{I}(x_i = \hat{\theta}_i) \quad (10)$$

Resulta conveniente reescribirlo de la siguiente manera:

$$\mathcal{L}(x_i|\hat{\theta}_i) = \frac{\exp[-\beta_L \mathcal{I}(x_i \neq \hat{\theta}_i)]}{1 + \exp(-\beta_L)} \quad (11)$$

En este caso β_L esta relacionado con \hat{p} como:

$$\beta_L = \ln \left[\frac{1}{\hat{p}} - 1 \right] \quad (12)$$

Entonces, el likelihood de $X|\hat{\theta}$ es el producto de todos los likelihood de proveniente de cada pixel. Así, tenemos:

$$\mathcal{L}(X|\hat{\theta}) = \frac{\exp[-\beta_L F(X, \hat{\theta})]}{\sum_x \exp(-\beta_L F(X, \hat{\theta}))} \quad (13)$$

Siendo $F(X, \hat{\theta})$ la distancia de Hamming entre X y $\hat{\theta}$. El parámetro $\beta_L = 1/T_L$ lo llamaremos la temperatura de degradación. Cuando $T_L = 0 \Rightarrow \hat{p} = 0$, es decir, la imagen no presenta ningún ruido y $X = \hat{\theta}$. Mientras incrementa la temperatura incrementa el ruido en la imagen.

2.6. Likelihood para imágenes de niveles de grises multiples

e forma contraria al caso anterior, tomaremos una imagen $\hat{\theta}$ con un nivel de grises $0, 1, 2, \dots, Q - 1$. Donde 0 es blanco y $Q-1$ es negro. El proceso de degeneración de la imagen es igual al caso anterior. En este caso, si $\epsilon \leq \hat{p}$ se cambiara el nivel de gris del i-esimo pixel seleccionado por otro de entre los $Q-1$ niveles. Esto de manera aleatoria y con la misma probabilidad para cada nivel de grises. Este proceso es descrito por el likelihood:

$$\mathcal{L}(x_i|\hat{\theta}_i) = \frac{\exp[-\beta_L \mathcal{I}(x_i \neq \hat{\theta}_i)]}{1 + (Q - 1) \exp(-\beta_L)} \quad (14)$$

donde:

$$\beta_L = \ln \left[(Q - 1) \left(\frac{1}{\hat{p}} - 1 \right) \right] \quad (15)$$

Entonces, el likelihood de $X|\hat{\Theta}$ es el producto de todos los likelihood de proveniente de cada pixel. Así, tenemos:

$$\mathcal{L}(X|\hat{\Theta}) = \frac{\exp \left[-\beta_L F(X, \hat{\Theta}) \right]}{\sum_x \exp \left(-\beta_L F(X, \hat{\Theta}) \right)} \quad (16)$$

Siendo $F(X, \hat{\Theta})$ la distancia de Hamming entre X y $\hat{\Theta}$. El parámetro $\beta_L = 1/T_L$ lo llamaremos la temperatura de degradación. Cuando $T_L = 0 \Rightarrow \hat{p} = 0$, es decir, la imagen no presenta ningún ruido y $X = \hat{\Theta}$. Mientras incrementa la temperatura incrementa el ruido en la imagen. Notece que cuando $Q = 2$ recuperamos la degradación de la imagen binaria.

2.7. Prior

Escoger un prior en la metodología bayesiana es algo subjetivo. Este refleja lo que nosotros creemos que debería lucir una imagen clara y suave. Cada uno de los pixeles debe ser suavemente conectado con su vecindario.

Para llevar a cabo esto, introduciremos la idea de una energía de interacción entre los estados de dos pixeles que se encuentran uno junto al otro. Dichos estados i, j interactúan con una energía $E_{i,j}$. Tomamos esta energía para ser aditiva a pares y así definimos una energía de imagen como:

$$E(\Theta) = \sum_{\langle i,j \rangle} E_{i,j}(\Theta) \quad (17)$$

El símbolo $\langle i, j \rangle$ denota que los píxeles i y j son los vecinos más cercanos y la suma se toma sobre todos los pares distintos de píxeles vecinos más cercanos en el plano de la imagen. De esta manera tenemos que modelar a $E_{i,j}$ siendo muy pequeño cuando los niveles de grises son muy parecidos y grande cuando los niveles de grises difieren demasiado.

Definimos el prior como:

$$p(\theta) = \frac{\exp \left[-\beta_P E(\Theta) \right]}{Z(\beta_P)} \quad (18)$$

En este caso, $\beta_P = 1/T_P$ es el parámetro de suavizamiento y $Z(\beta_P)$ es la constante de normalización.

2.8. El modelo de Ising

El modelo de Ising es un modelo matemático de ferromagnetismo en mecánica estadística y es quisa el modelo más estudiado en esta área. Usaremos este modelo para describir nuestro piror. Por ahora sólo consideraremos una imagen binaria

cuyos niveles de grises son cero y uno. La energía de interacción entre los pixeles más cercanos será:

$$E_{i,j} = \mathcal{I}(\theta_i \neq \theta_j) \quad (19)$$

Si los dos pixeles tienen el mismo nivel de gris, la energía de interacción es cero y de lo contrario será uno. La energía de Ising de una imagen Θ es:

$$E(\Theta) = \sum_{\langle i,j \rangle} E_{i,j}(\Theta) = \sum_{\langle i,j \rangle} \mathcal{I}(\theta_i \neq \theta_j) \quad (20)$$

Así, el prior de Ising es:

$$p(\theta) = \frac{\exp \left[-\beta_P \sum_{\langle i,j \rangle} \mathcal{I}(\theta_i \neq \theta_j) \right]}{Z(\beta_P)} \quad (21)$$

2.9. Modelos de Potts

El modelo de Potts es una generalización del modelo de Ising sobre spins interactivos en una red cristalina. Consideremos una imagen con múltiples escalas de grises $Q - 1$. Usaremos la energía de interacción de Geman y McClure [??] dada por:

$$E_{i,j}(\Theta) = -\frac{1}{1 + C(\theta_i - \theta_j)^2} \quad (22)$$

Donde C determina el ancho de la distribución (ver figura 1). Para C muy grandes, la interacción Geman-McClure se reduce a la interacción de Ising con un nivel de energía base y editado de -1 y 0 respectivamente. De esta forma, el prior de Geman-McClure será:

$$p(\theta) = \frac{1}{Z(\beta_P)} \exp \left[-\beta_P \sum_{\langle i,j \rangle} \frac{-1}{1 + C(\theta_i - \theta_j)^2} \right] \quad (23)$$

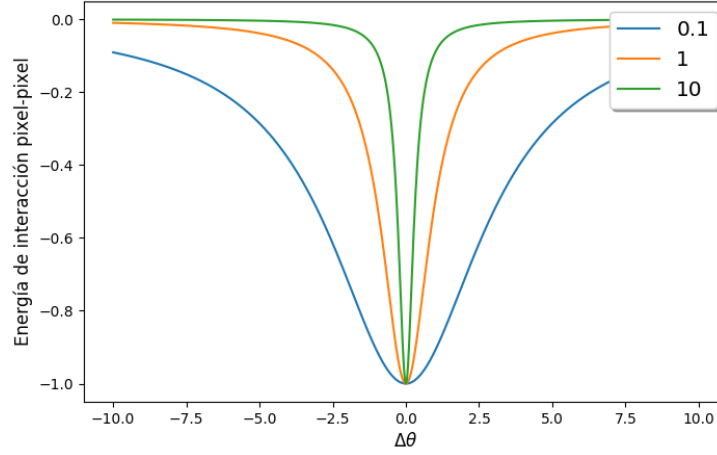


Figura 1: Energía de Gemen-McClure $E_{i,j}$ contra $\theta_i - \theta_j$ para $C = 0,1, 1, 10$.

2.10. Distribucion a Porteriori

La distribución a posteriori está dada por el teorema de Bayes siendo el producto del likelihood y el prior.

$$p(\Theta|X) = \frac{\mathcal{L}(X|\theta)p(\theta)}{\sum_{\Theta} \mathcal{L}(X|\theta)p(\theta)} \quad (24)$$

Este podemos obtenerlo por:

$$p(\Theta|X) = \frac{\exp[-\{\beta_L F(\Theta, X) + \beta_P E(\Theta)\}]}{\sum_{\Theta} \exp[-\{\beta_L F(\Theta, X) + \beta_P E(\Theta)\}]} \quad (25)$$

Ahora, procederemos a dividir E por el número de vecinos más cercanos de los pixeles en la imagen denotado por N_E . Para una imagen de $L \times L$ pixeles, tenemos $N_E = 2L(L-1) \approx 2L^2$ para L muy grandes. Dividimos, además, $F(\Theta, X)$ por L^2 . De esta forma obtenemos:

$$p(\Theta|X) = \frac{\exp[-L^2(1+\beta)^{-1}\{F(\Theta, X) + 2^{-1}\beta E(\Theta)\}]}{\sum_{\Theta} \exp[-L^2(1+\beta)^{-1}\{F(\Theta, X) + 2^{-1}\beta E(\Theta)\}]} \quad (26)$$

Donde β es valor de β_P en unidades e β_L , es decir, $\beta_L = 1$. Además, se ha normalizado los pesos β_L y β_P dividiéndolos sobre $\beta_L + \beta_P$. $T = 1/\beta$ le llamaremos sólo temperatura.

De esta manera, para una temperatura dada, el posteriori incrementa cuando $F(\Theta, X)$ se reduce. Es decir, mientras más cercana este la imagen Θ a la imagen X , más grande será el valor del posteriori $p(\Theta, X)$. Por otro lado el posteriori incrementa cuando la energía de interacción E decrece dada una temperatura T .

En otras palabras, entre más suave sea la imagen Θ , más grande es el valor de $p(\Theta, X)$. Así, tenemos una pelea entre el prior y el likelihood, el prior trata de suavizar la imagen y el likelihood trata de que la imagen Θ permanezca cercana a X como sea posible. De esta manera tenemos como resultado una imagen que tiene características de X (debido al likelihood) y es suave, retirando el ruido (debido al prior).

La temperatura determinará la competición del prior con el likelihood. Para una temperatura dada: el peso para F es $T/(1+T)$ y el peso para E es $1/(1+T)$. Cuando T es grande, se da una mayor importancia a la retención de las características de la imagen X , es decir, al likelihood. El costo de T muy grandes es que el ruido permanecerá y la imagen no cambiará en grandes aspectos. Por otro lado, si T es pequeño, el suavizamiento (el prior) es más importante. En este caso, para T muy pequeños la imagen perderá sus características e incluso puede llegar a terminar siendo irreconocible.

3. Metodología

Montecarlo Markov Chain (MCMC) provee de técnicas para realizar algoritmos para generar un conjunto de muestras aleatorias a partir de una distribución de probabilidad. Los algoritmos crean una cadena de muestras donde cada muestra depende de las anteriores creadas. Estos métodos MCMC son atractivos porque proporcionan una forma directa e intuitiva de simular valores de una distribución desconocida y de utilizar esos valores simulados para realizar análisis posteriores. Esto les permite ser aplicables en una amplia variedad de dominios. Por ejemplo, técnicas numéricas para el procesamiento de imágenes. En este caso nos enfocaremos en implementar el algoritmo de Montecarlo para la restauración de imágenes digitales.

Recordemos que el problema a resolver es: encontrar la imagen Θ que se aproxime a la imagen verdadera $\hat{\Theta}$ a partir de una imagen con ruido X dada. De esta manera, comenzamos fijando una temperatura T y creando una imagen aleatoria Θ_0 que por conveniencia escogeremos $\Theta_0 = X$ ya que, por obvias razones, es la que más se aproxima a $\hat{\Theta}$ por el momento. A esta imagen le llamaremos corriente $\Theta_c = \Theta_0$.

1. Calculamos $p_c = p(\Theta_c|X)$
2. Seleccionamos un pixel de manera aleatoria de la imagen corriente
3. Cambiamos el nivel de gris de este pixel por otro de forma aleatoria a otro nivel de gris $Q - 1$ que compone a la imagen, teniendo estos la misma probabilidad de ser escogidos.
4. Esta imagen con el pixel cambiado le llamaremos imagen trial Θ_t y calculamos $p_t = p(\Theta_t|X)$
5. Si $p_t > p_c$, entonces se acepta a la imagen trial en la siguiente muestra de la cadena $\Theta_1 = \Theta_t$; de otra forma $\Theta_1 = \Theta_0$
6. Ahora, tomamos Θ_i como la imagen corriente Θ_c y comenzamos de nuevo el proceso.

De esta manera tendremos la cadena De Markov:

$$\Theta_1 \rightarrow \Theta_2 \rightarrow \Theta_3 \rightarrow \cdots \rightarrow \Theta_n \rightarrow \cdots$$

Donde la imagen Θ_{k+1} depende solamente de Θ_k .

3.1. Proceso de degradación de la imagen

Para obtener una imagen con ruido X se obtendrá la imagen verdadera $\hat{\Theta}$ y se corromperá con ruido. Para esto, se seleccionará un pixel i de la imagen $\hat{\Theta}$ para después generar un número aleatorio ϵ . Si $\epsilon < \hat{p}$ entonces el nivel de gris del pixel seleccionado se cambiara a otro nivel que contenga la imagen. Aquí $0 < \hat{p} < 1$ es el porcentaje de ruido que se elija ponerle a la imagen. Este proceso se realizara para cada uno de los pixeles que componen la imagen $\hat{\Theta}$.

4. Resultados

4.1. Imagen binaria

Se obtuvo la imagen de un robot con sólo dos niveles de grises $Q = 0, 1$ reduciéndola a un tamaño de 35×35 (Figura 2 izquierda) adecuado para que el tiempo de ejecución del código no sea demasiado. A dicha imagen se le agregó $\approx 5\%$ de ruido dando como resultado la figura 2 central. Usando el Prior de Ising (ecuación 21) y Hamming Likelihood para una maximización del posteriori con $T = 0.5$ y 10000 iteraciones se obtuvo la imagen de la figura 2 derecha. Como se puede apreciar, se mantuvo la estructura del robot original perdiendo el ruido generado. El costo final fue al perder algunos detalles de la figura en el proceso.

4.2. Imagen con 6 niveles de grises

Al igual que en el caso anterior se obtuvo la misma imagen de robot reduciéndola a un tamaño de 35×35 cambiando los niveles de grises en diferentes regiones. Estos niveles van de 0, que corresponde al color blanco, al máximo 5, que corresponde al color negro (ver figura 3 izquierda). Se corrompió la imagen con $\approx 5\%$ de ruido y se obtuvo la imagen X figura 3 central. Para este caso se empleó el prior con la energía de interacción de Geman-McClure y el likelihood de Hamming en el método de metrópolis con una temperatura de $T = 0.5$. De esta forma se obtuvo la imagen de la figura 3 derecha. Además, se capturo el máximo posteriori de las iteraciones y se muestra en la figura 4.

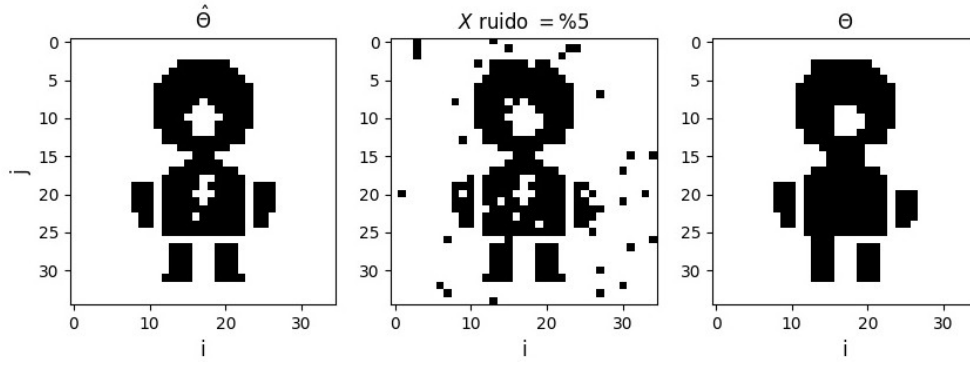


Figura 2: Imagen binaria de tamaño $L = 35$. (Izquierda) imagen verdadera $\hat{\Theta}$, (Centro) imagen con ruido agregado X ($\hat{p} = 0,05$) y (Derecha) imagen restaurada usando el método de metrópolis a una temperatura de $T = 0,5$.

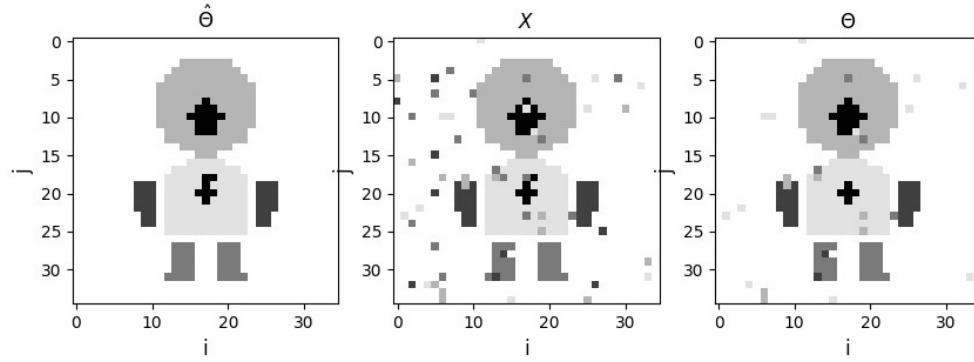


Figura 3: Imagen con 6 niveles de grises de tamaño $L = 35$. (Izquierda) imagen verdadera $\hat{\Theta}$, (Centro) imagen con ruido agregado X ($\hat{p} = 0,05$) y (Derecha) imagen restaurada usando el método de metrópolis a una temperatura de $T = 0,5$.

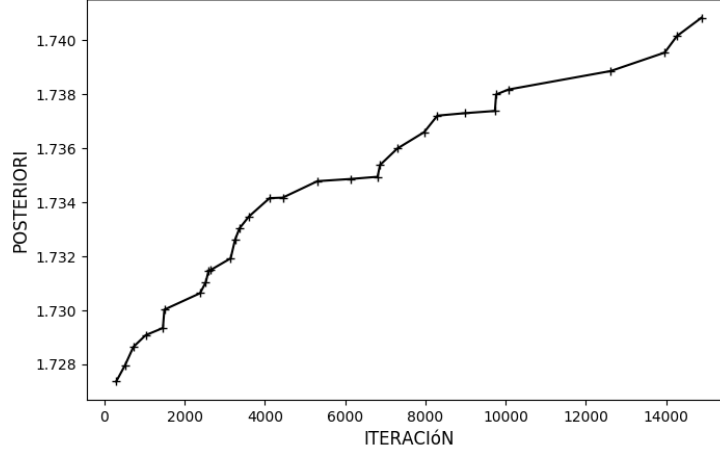


Figura 4: Posteriori máximo contra índice de iteración

Para esta misma imagen se procedió a llevar a cabo el mismo método, pero con temperaturas diferentes $T = 0,01$, $T = 0,99$ y $T = 1,7$. Los resultados se muestra en la figura 5. Vemos que para temperaturas altas la imagen no sufre ningún tipo de cambio. Por otro lado, para valore de T pequeños la imagen comienza a perder más detalles de sí mismas con una buena limpieza de ruido.

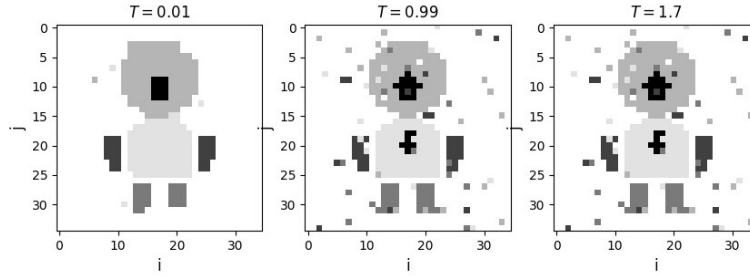


Figura 5: Imagen con 6 niveles de grises de tamaño $L = 35$. (Izquierda) imagen procesada con $T = 0,01$, (Centro) imagen pocesada con $T = 0,99$ y (Derecha) imagen procesada con $T = 0,17$.

5. Conclusión

Se implemento el método de metrópolis para el procesamiento de limpieza de ruido de un par de imágenes digitales sencillas. Los resultados fueron en

principio exitosos. En las primeras iteraciones se logró disminuir el ruido de forma considerable si perder las características de las imágenes verdaderas. El código realizado en Python conlleva un tiempo de cómputo ≈ 6 minutos para 10000 iteraciones. Como se pudo observar en las imágenes obtenidas, el proceso aún necesita más iteraciones para lograr eliminar la mayor parte del ruido de las imágenes. Se optó por no llevar a cabo esta acción por seguridad de la máquina que se usó para esta tarea. Se espera que al realizar un código más eficiente se puede llevar a cabo un número mayor de iteraciones y, de esta manera, arrojar mejores resultados en las imágenes. Por el momento los resultados nos indican que podemos confiar en el método de metrópolis para la limpieza de imágenes.

Bibliografía

- [1] MURTHY, K.; JANANI, M.; SHENBAGA B. *Bayesian Restoration of Digital Images Employing Markov Chain Monte Carlo - a Review* School of Physics, University of Hyderabad.
- [2] SPEAGLE, J.S. *A Conceptual Introduction to Markov Chain Monte Carlo Methods* Center for Astrophysics, Harvard and Smithsonian.
- [3] WIKIPEDIA CONTRIBUTORS. (2019, July 14). *Image restoration*. In Wikipedia, The Free Encyclopedia. Retrieved 23:10, December 13, 2019, from [https : //en.wikipedia.org/w/index.php?title = Image_restoration&oldid = 906240514](https://en.wikipedia.org/w/index.php?title=Image_restoration&oldid=906240514)
- [4] GEMEN, S.; MCCLURE, D. E. *Statistical methods in tomographic image reconstruction*, Proc. 46-th session of the International statistical institute, Bulletin of ISI, 52, 5 (1987)
- [5] FLASH BOT CIRCLE PNG CLIPART <https://www.uihere.com/free-cliparts/flash-bot-circle-7778022>