

# Implementación del algoritmo de umbralización de Yen

Erick Salvador Alvarez Valencia, CIMAT A.C.

## I. INTRODUCCIÓN

En el presente reporte se mostrará sobre el estudio y la implementación del algoritmo de umbralización no paramétrico de Yen usando el lenguaje de programación python. Se dará un poco de énfasis en las ventajas de este algoritmo basado en el punto de vista de los autores del paper original. De la misma forma se mostrará el pseudocódigo de la implementación y los resultados de aplicar el algoritmo con ciertas imágenes médicas.

mds

Febrero, 2018

## II. DESCRIPCIÓN

La umbralización de imágenes es el proceso por el cual se busca un umbral óptimo que permita distinguir en una imagen los objetos del fondo de los objetos del primer plano. Para comenzar hay que destacar que el método que se mostrará en este reporte es un algoritmo de umbralización basado en histograma y sobre todo, en la entropía, se puede dar una pequeña introducción con respecto a este tipo de métodos. Un método de umbralización basado en histograma busca el punto (o valor) umbral en el cual el histograma de una imagen se divide en dos picos. En la mayoría de las imágenes este valor resulta un poco difícil de encontrar gráficamente debido a la complejidad de estos histogramas. Es por eso que se usan métodos paramétricos y no paramétricos que modelizan el problema y encuentran diferentes maneras de obtener el mismo. Para la siguiente parte es necesario definir algunos conceptos con los que trabajará el algoritmo.

Considere  $f_i$  como la frecuencia del  $i$ -ésimo nivel de gris en una imagen con  $m$  diferentes niveles de gris. Ahora sea  $G_m = \{0, 1, \dots, (m-1)\}$  el conjunto de todos los niveles presentes en la imagen, la probabilidad de que el  $i$ -ésimo nivel de gris esté en la imagen es:

$$p_i = \frac{f_i}{NxN}, p_i \in G_m \quad (1)$$

De lo anterior podemos obtener una distribución, para algún  $s \in [0, m-1]$  se puede obtener un valor de probabilidad acumulada:  $\sum_{i=0}^{s-1} p_i$  el cual es mayor que cero y menor o igual que uno.

De lo anterior podemos obtener los dos siguientes conjuntos de distribuciones de probabilidad:

$$A = \left\{ \frac{p_0}{P(s)}, \frac{p_1}{P(s)}, \dots, \frac{p_{s-1}}{P(s)} \right\} \quad (2)$$

$$B = \left\{ \frac{p_s}{1-P(s)}, \frac{p_{s+1}}{1-P(s)}, \dots, \frac{p_{m-1}}{1-P(s)} \right\} \quad (3)$$

De los cuales denotan si  $p_i$  está en uno u otro conjunto con  $P(s) = \sum_{i=0}^{s-1} p_i$  como la probabilidad total hasta  $s$ .

La idea básica con lo anterior es encontrar el valor de  $s$  tal que maximice el monto de información que provee el objeto de interés y el fondo de la imagen. Como dicho monto es medido por la entropía entonces el monto total de información que provee  $A$  y  $B$  es:

$$E_T(s) = E_A(s) + E_B(s) \quad (4)$$

De lo anterior planteamos el problema de maximización como:

$$E_T(s^*) = \max_{s \in G_m} E_T(s) \quad (5)$$

Los algoritmos más clásicos basados en este enfoque son los propuestos por Kapur<sup>2</sup>, Sahoo y Wong donde, por ejemplo, Kapur considera al objeto de interés y al fondo de la imagen como dos conjuntos de señales distintas las cuales siguen el enfoque descrito anteriormente por lo que propone la siguiente función a maximizar:

$$TC(s) = C_b(s) + C_f(s) \quad (6)$$

$$TC(s) = -\log\left\{\sum_{g=0}^s \left[\frac{p(g)}{P(s)}\right]^2\right\} - \log\left\{\sum_{g=s+1}^G \left[\frac{p(g)}{1-P(s)}\right]^2\right\} \quad (7)$$

Donde la ecuación anterior refiere a la entropía de Shannon para los elementos que están en el conjunto del fondo y los que están en el conjunto del objeto de interés.

Lo que observó Yen es que en el método anterior propuesto por Kapur se generan muchas evaluaciones de la función de lo cual conduce a muchas evaluaciones con logaritmos naturales que cuentan con un costo computacional alto, por lo que Yen propone usar un criterio de correlación el cual se define de la siguiente manera:

$$C(X) = -\log \sum_{i \geq 0} p_i^2 \quad (8)$$

La idea es utilizar el criterio de correlación en lugar de la entropía justificándose en que en la teoría del caos, la correlación y la entropía son usados para representar imágenes

y modelar algunos objetos. Sustituyendo lo anterior en la función objetivo se minimizará el costo computacional ya que la evaluación de logaritmos es mucho menor. Ahora, basado en la definición dada en (5) y la función de correlación definida en (8) se genera la siguiente función a maximizar:

$$TC(s) = C_A(s) + C_B(s)$$

$$TC(s) = -\log \sum_{i=0}^{s-1} \left( \frac{p_i}{P(s)} \right)^2 - \log \sum_{i=s}^{m-1} \left( \frac{p_i}{1-P(s)} \right)^2 \quad (9)$$

$$TC(s) = -\log \{ [G(s)xG'(s)] / [P(s)^2 x(1-P(s))^2] \}$$

$$\text{Donde } G(s) = \sum_{i=0}^{s-1} p_i^2 \text{ y } G'(s) = \sum_{i=s}^{m-1} p_i^2$$

$$= -\log[G(s)xG'(s)] + 2\log[P(s)x(1-P(s))] \quad (10)$$

Por lo que finalmente el algoritmo se define de la siguiente manera:

$$TC(s^*) = \max_{s \in G_m} TC(s) \quad (11)$$

### III. IMPLEMENTACIÓN

Para el presente trabajo se implementó el algoritmo de entropía de Yen usando el lenguaje de programación Python. A continuación se mostrará el pseudocódigo de dicha implementación:

---

#### Algorithm 1 Entropía de Yen.

---

```

1: procedure YEN(data)
2:   tot ← cumsum(data).
3:   normHist ← data / tot.
4:   P ← array(0, 256).
5:   P1 ← array(0, 256).
6:   P2 ← array(0, 256).
7:   P[0] ← normHist[0].
8:   P1[0] ← normHist[0]2.
9:   for i = 1 → 255 do
10:    P[i] ← P[i - 1] + normHist[i].
11:    P1[i] ← P1[i - 1] + normHist[i]2.
12:   for i = 254 → 0 do
13:    P2[i] ← P2[i + 1] + normHist[i]2.
14:   threshold ← -1.
15:   max ← -∞.
16:   for i = 0 → 255 do
17:     if P1[i]*P2[i] > 0 and P[i] * (1 - P[i]) > 0 then
18:       aux ← -log(P1[i] * P2[i]) + 2log(P[i] * (1 -
19:         P[i])).
20:       if aux > max then
21:         max ← aux.
22:         threshold ← i.
23:   return threshold

```

---

Del algoritmo anterior, el arreglo *data* es el histograma de la imagen, *P* es la función de probabilidad acumulada, *P1* es el cuadrado de la FPA y *P2* también representa el cuadrado de

la FPA solamente que la suma es comenzando desde el último elemento del arreglo hasta el primero, esto nos va a permitir fácilmente dar la respuesta en tiempo constante a la  $G'(s)$ , por lo demás podemos ver que para maximizar el umbral se prueban todos los valores contenidos en el conjunto de las escalas de gris en una imagen a la función de correlación y se guarda el máximo, al final este será el umbral buscado.

### IV. EJECUCIÓN Y RESULTADOS

En este trabajo se ejecutó el algoritmo de umbralización de Yen primeramente para un conjunto de dos imágenes estándar y posteriormente con algunas imágenes médicas obtenidas de prueba, de la misma forma se aplicó a estos conjuntos el algoritmo de umbralización de Otsu y a continuación se mostrarán los resultados a modo de comparativa.

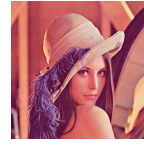


Figura 1:  
Imagen  
original

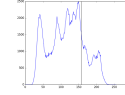


Figura 2:  
Umbral  
encontrado



Figura 3:  
Segmen-  
tación por  
Yen



Figura 4:  
Segmen-  
tación por  
Otsu

Valor del umbral encontrado: 160.



Figura 5:  
Imagen  
original

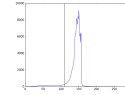


Figura 6:  
Umbral  
encontrado



Figura 7:  
Segmen-  
tación por  
Yen



Figura 8:  
Segmen-  
tación por  
Otsu

Valor del umbral encontrado: 110.

Ahora se mostrarán los resultados del algoritmo con un pequeño conjunto de imágenes médicas.

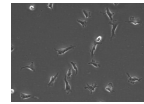


Figura 9:  
Imagen  
original

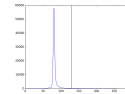


Figura 10:  
Umbral  
encontrado



Figura 11:  
Segmen-  
tación por  
Yen



Figura 12:  
Segmen-  
tación por  
Otsu

Valor del umbral encontrado: 130.

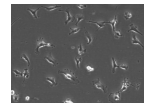


Figura 13:  
Imagen  
original

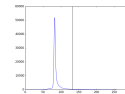


Figura 14:  
Umbral  
encontrado

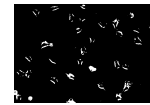


Figura 15:  
Segmen-  
tación por  
Yen



Figura 16:  
Segmen-  
tación por  
Otsu

Valor del umbral encontrado: 133.

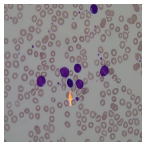


Figura 17:  
Imagen  
original

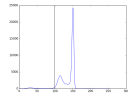


Figura 18:  
Umbral  
encontrado



Figura 19:  
Segmen-  
tación por  
Yen

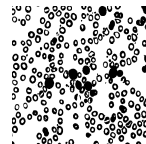


Figura 20:  
Segmen-  
tación por  
Otsu

Valor del umbral encontrado: 98.

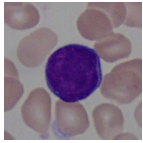


Figura 21:  
Imagen  
original

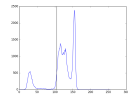


Figura 22:  
Umbral  
encontrado



Figura 23:  
Segmen-  
tación por  
Yen



Figura 24:  
Segmen-  
tación por  
Otsu

Valor del umbral encontrado: 105.

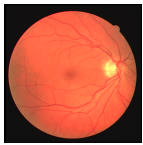


Figura 25:  
Imagen  
original

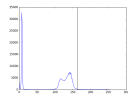


Figura 26:  
Umbral  
encontrado



Figura 27:  
Segmen-  
tación por  
Yen

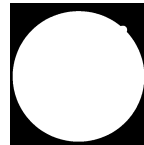


Figura 28:  
Segmen-  
tación por  
Otsu

Valor del umbral encontrado: 160.

## V. CONCLUSIONES

En este reporte se trató el tema de una implementación al algoritmo de umbralización basado en entropía de Yen. Como conclusiones finales cabe mencionar que como menciona el autor, este algoritmo es eficiente hablando en costo computacional, primeramente porque no es iterativo y segundo porque la función de evaluación no es tan pesada computacionalmente hablando, en general las ejecuciones tornaban con una media de 1s incluyendo el precálculo del histograma y las funciones de probabilidad.

Hablando en la parte de resultados se puede ver que funciona muy bien para ciertas imágenes, como en este caso son las imágenes celulares y de leucocitos pero en otras como es el caso de la última mostrada en los ejemplos, la imagen de retina este algoritmo no pudo segmentar correctamente el objeto de interés, esto ya que el algoritmo es muy susceptible a cambios bruscos en las intensidades de los píxeles.

## REFERENCIAS

- [1] J. C. Yen, F. J. Chang, and S. Chang, *A new criterion for automatic multilevel thresholding*, IEEE Trans. Image Process, 1995.
- [2] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, *A new method for gray-level picture thresholding using the entropy of the histogram*, Graph. Models Image Process, 1985.
- [3] M. Sezgin, and B. Sankur, *Survey over image thresholding techniques and quantitative performance evaluation*, Journal of Electronic Imaging 13(1), 146 – 165, 2004.