



# Experiencias en la implementación de las operaciones morfológicas de erosión y dilatación para imágenes binarias empleando vecindades adaptativas

*Yosbel Morales Olivera, Josué García Parrado, Pablo E. Reyes Fernández y Juan V. Lorenzo Ginori*

## **RESUMEN / ABSTRACT**

La morfología matemática es una herramienta muy poderosa para el procesamiento digital de imágenes. Se plantea que la morfología variable en el espacio muestra mejores resultados que la invariante. Existen varias morfologías variables en el espacio y se diferencian en la forma de asignar el elemento estructurante a cada píxel. Entre estas está la de vecindades adaptativas, que usa los conceptos de función de semejanza y nivel de tolerancia que permiten que el elemento estructurante asignado a cada píxel se ajuste a las características de la vecindad de este y varíen en forma y tamaño. La función de semejanza le asigna a cada píxel un valor real, que puede ser: iluminación, contraste, curvatura, etc. En este trabajo se implementan en MatLab la erosión y dilatación de la morfología de vecindades adaptativas. Las imágenes se procesan con las operaciones estándar de MatLab y con las implementadas en el trabajo, para observar similitudes y diferencias.

Palabras claves: morfología matemática variable en el espacio, vecindades adaptativas, erosión, dilatación.

*Mathematical morphology is a powerful tool in image processing. It is asserted that the spatially-variant morphology produces better results than the invariant one in many cases. There are several spatially-variant morphologies and they differ in the form of assigning the structuring element to each pixel. One of these is the adaptive neighborhood morphology, which uses the concepts of criterion mapping and homogeneity tolerance. These concepts allow that the structuring element assigned to each pixel is adjusted to the characteristics of the pixel neighborhood and vary in form and size. The criterion mapping assigns to each pixel a real value that can represent: luminance, contrast or curvature among others. In this paper the operations of erosion and dilation of the adaptive neighborhood morphology are implemented using MatLab. The images are processed with the MatLab conventional operations and with the operations implemented in this paper to observe similarities and differences.*

**Key words:** *spatially-variant mathematical morphology, adaptive neighborhood, erosion, dilation.*

*Experiences in the implementation of the morphological erosion and dilation operations for binary images using adaptive neighborhoods.*

## INTRODUCCION

Desde su surgimiento en la década de los años 70, la morfología matemática se ha convertido en una herramienta muy poderosa y utilizada para diferentes tareas en el procesamiento digital de imágenes. Las operaciones básicas de la morfología matemática son las de erosión y dilatación, <sup>1 2</sup>. La morfología matemática introduce el concepto de elemento estructurante, el cual es utilizado para procesar las imágenes. En la morfología invariante en el espacio se usa el mismo elemento estructurante para procesar toda la imagen. Se ha mostrado que este tipo de procesamiento no siempre logra los resultados deseados, <sup>3 5</sup>. En el caso, por ejemplo, del filtrado morfológico al usar un elemento estructurante fijo para toda la imagen se pueden eliminar detalles útiles de la misma si el tamaño del elemento estructurante es muy grande, mientras que un elemento estructurante muy pequeño no eliminaría elementos que se desea suprimir en la imagen. Por su parte en la morfología variable en el espacio, a cada píxel de la imagen se le procesa con un elemento estructurante diferente. Existen muchas formas de tomar el elemento estructurante que le corresponde a cada píxel de la imagen, lo que da origen a diferentes morfologías. En <sup>3</sup> se muestran algunos ejemplos de morfologías variables en el espacio, entre otras están: *Morfología Circular*, *Morfología Ameba* y *Morfología de Vecindades Adaptativas*.

En <sup>5</sup> se realiza la discusión teórica acerca del filtrado morfológico usando la morfología de vecindades adaptativas (*Adaptive Neighborhood Morphology*). Se plantea que el elemento estructurante que se usa en este tipo de morfología matemática se ajusta a las características propias de la región de la imagen que se esté procesando. Estas características de la imagen pueden ser contraste, iluminación, curvatura, etc. Se puede ver además que los elementos estructurantes en esta morfología varían tanto en tamaño como en forma, ajustándose a cierto nivel de tolerancia especificado.

En este trabajo se implementan en MatLab las operaciones de erosión y dilatación con el uso de las vecindades adaptativas (*Adaptive Neighborhood*). Se muestra además el uso de estas operaciones con imágenes binarias y se comparan los resultados obtenidos con los que se obtienen al aplicar las operaciones equivalentes convencionales que ya están implementadas en MatLab <sup>4</sup>.

## MATERIALES Y MÉTODOS

En esta sección se muestra la implementación del algoritmo computacional para la dilatación y la erosión usando vecindades adaptativas. En una sección posterior se exponen los resultados obtenidos.

En la morfología de vecindades adaptativas existen dos elementos muy importantes a tener en cuenta; la función de semejanza y el nivel de tolerancia, pues de ellos depende el elemento estructurante a utilizar para cada píxel. La función de semejanza le asigna a cada píxel de la imagen un valor real, o sea:  $S : \mathcal{R}^2 \mapsto \mathcal{R}$  y es una función que representa ciertas características intrínsecas de la vecindad del píxel a procesar. En este trabajo se usan como funciones o matrices de semejanza la componente de saturación S y la componente de Valor V de las imágenes en el espacio de color HSV, pues no se está tratando ninguna aplicación específica de esta morfología y solo se desea mostrar el funcionamiento del algoritmo computacional propuesto.

En el caso del nivel de tolerancia, su importancia radica en el hecho de que al variar su valor varía la componente conexa formada por los píxeles semejantes al píxel que se está analizando, pues dos píxeles x e y son semejantes si  $|S(x) - S(y)| \leq m$ .

### A. Algoritmo computacional

#### A.1 Dilatación

El algoritmo computacional para la operación de dilatación se muestra en el siguiente pseudo código:

#### Datos de entrada

A // Imagen binaria original

S // Matriz de semejanza

m // Nivel de tolerancia

k // contador

#### Resultado del algoritmo

A1 // Imagen dilatada

## Begin

//  $\forall x \in A$  buscar los  $y \in A$  tales que:  $|S(x) - S(y)| \leq m$  y crear una matriz D (con valor uno en las posiciones de los píxeles que cumplen la condición antes mencionada y cero en el resto).

// Se realiza un etiquetado de la matriz D usando la función de MatLab *bwlabel*. El elemento estructurante del píxel que se está analizando será entonces la componente conexas de la matriz D a la cual pertenece dicho píxel.

// Se inicializa la matriz de la imagen de salida ( $A1 = A$ ).

// Si  $A(i, j) = D(i, j) = 1$ , entonces:  $A1(i, j) = 1$ , en caso contrario  $A1(i, j) = 0$ .

// La imagen dilatada es A1.

## End

### A.2 Erosión

En el caso de la erosión solo hay que cambiar el penúltimo paso del algoritmo que en este caso quedaría en tres pasos:

//  $k = 0$ .

//  $\forall D$ , si  $A(i, j) = D(i, j) = 1$  y no se cumple que:  $D(i, j) = 1$  y  $A(i, j) = 0$ , entonces  $k = k + 1$ .

// Si  $k > 1$ , entonces  $A1(i, j) = 1$ , en caso contrario  $A1(i, j) = 0$ .

### B Imágenes de prueba

Para probar el algoritmo primeramente se construyeron imágenes binarias de 5x5 y se construyó para cada una de ellas una matriz de semejanza conveniente para mostrar las operaciones de erosión y dilatación, como se muestra en la figura 1. Luego se escogieron imágenes a color de una base de imágenes de hematología, tomando regiones de interés de estas últimas, véase la figura 2. Estas imágenes fueron transformadas a binarias y se escogieron estas últimas como imágenes a procesar. Para escoger la matriz de semejanza de estas imágenes a color estas se transformaron del modelo RGB al modelo HSV y se escogieron como matriz de semejanza las componentes de saturación S y de valor V de este último modelo. Los resultados se obtuvieron para varios niveles de tolerancia.

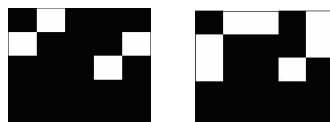


Figura 1

Imágenes binarias de 5x5 construidas para probar el algoritmo computacional.



Figura 2

Imágenes a color escogidas de una base de imágenes hematológicas para probar el algoritmo computacional.

### C Dilatación

Sea la imagen binaria:

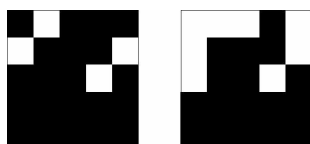
$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

A continuación se muestra la matriz de semejanza que se utiliza en la dilatación (los valores de esta matriz fueron escogidos convenientemente para mostrar el funcionamiento del algoritmo, ellos no representan valores de ningún parámetro relacionado con la imagen):

$$S = \begin{pmatrix} 5 & 8 & 9 & 15 & 18 \\ 6 & 23 & 56 & 60 & 19 \\ 4 & 98 & 67 & 71 & 100 \\ 300 & 320 & 350 & 400 & 500 \\ 600 & 610 & 620 & 630 & 640 \end{pmatrix}$$

La matriz B, obtenida de la dilatación de A, usando el algoritmo mostrado con nivel de tolerancia  $m=2$  es:

$$B = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



**Figura 3**

A la izquierda aparece la imagen A y a la derecha la imagen B, obtenida al dilatar A con la matriz de semejanza S y un nivel de tolerancia  $m=2$ .

A continuación se muestra otro ejemplo de dilatación de una imagen binaria. En este caso se parte de una imagen a color, la cual se convierte a binaria usando la función de MatLab *im2bw* con parámetro 0.6, figura 4. En este ejemplo se toma como matriz de semejanza S, a la componente de saturación del modelo HSV de la imagen a color.



**Figura 4**

A la izquierda aparece la imagen original y a la derecha la imagen obtenida al transformar la primera con la función de MatLab *im2bw* con parámetro 0.6.

En la figura 5 se aprecia en la primera fila, de izquierda a derecha la dilatación producida en la imagen con el algoritmo computacional implementado en este trabajo para los valores de tolerancia  $m = 0.001$ ,  $m = 0.005$ ,  $m = 0.01$  respectivamente. En la segunda fila, de izquierda a derecha, se muestran los resultados de la dilatación con la función *imdilate* del MatLab usando como elemento estructurante un disco de radios 1, 2 y 3 respectivamente.



Figura 5

Primera fila de izquierda a derecha la dilatación con el algoritmo computacional propuesto para los valores de tolerancia  $m=0.001$ ,  $m=0.005$ ,  $m=0.01$  respectivamente. Segunda fila de izquierda a derecha dilatación con la función de MatLab *imdilate*, usando como elemento estructurante un disco de radios 1,2 y 3 respectivamente.

En el siguiente ejemplo de dilatación se usa nuevamente una imagen a color, la cual se convierte a binaria de forma análoga al ejemplo anterior, figura 6. En este caso se usa como matriz de semejanza a la componente de valor V del modelo HSV de la imagen a color.



Figura 6

A la izquierda aparece la imagen original y a la derecha la imagen obtenida al transformar la primera con la función de MatLab *im2bw* con parámetro 0.7.

En la figura 7 se aprecia en la primera fila, de izquierda a derecha la dilatación producida en la imagen con el algoritmo computacional implementado en este trabajo para los valores de tolerancia  $m = 0.01$ ,  $m = 0.02$  y  $m = 0.04$  respectivamente. En la segunda fila, de izquierda a derecha, se muestran los resultados de la erosión con la función *imdilate* del MatLab usando como elemento estructurante un disco de radio 1, 2 y 3 respectivamente.



Figura 7

Primera fila de izquierda a derecha la dilatación con el algoritmo computacional propuesto para los valores de tolerancia  $m=0.01$ ,  $m=0.02$  y  $m=0.04$  respectivamente. Segunda fila de izquierda a derecha dilatación con la función de MatLab *imdilate*, usando como elemento estructurante un disco de radios 1,2 y 3 respectivamente.

Se puede apreciar que a medida que se aumenta el nivel de dilatación con la función *imdilate* del MatLab se tiende a perder la topología original de la imagen (se unen componentes de la imagen que originalmente no estaban unidas), esto ocurre porque se usa el mismo elemento estructurante en todos los píxeles de la imagen. Por su parte, con el

algoritmo implementado en este trabajo se conserva mejor la topología durante la dilatación, hecho que está

motivado por la forma de elegir el elemento estructurante para cada píxel, dependiendo de las características intrínsecas de la región que se va a procesar.

### C Erosión

Sea la imagen binaria:

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

A continuación se muestra la matriz de semejanza que se utiliza en la erosión (los valores de esta matriz fueron escogidos convenientemente para mostrar el funcionamiento del algoritmo, ellos no representan valores de ningún parámetro relacionado con la imagen):

$$S = \begin{pmatrix} 5 & 8 & 9 & 15 & 18 \\ 6 & 23 & 56 & 60 & 19 \\ 4 & 98 & 67 & 71 & 100 \\ 300 & 320 & 350 & 400 & 500 \\ 600 & 610 & 620 & 630 & 640 \end{pmatrix}$$

La matriz B, obtenida de la Erosión de A, usando el algoritmo mostrado con nivel de tolerancia  $m=2$  es:

$$B = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

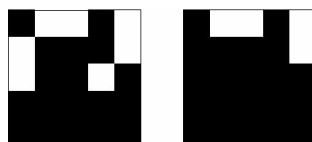


Figura 8

A la izquierda aparece la imagen A y a la derecha la imagen B, obtenida al erosionar A con la matriz de semejanza S y un nivel de tolerancia  $m=2$ .

Seguidamente se ilustra otro ejemplo de erosión de una imagen binaria. En este caso, al igual que en el segundo ejemplo de dilatación se parte de una imagen a color, la cual se convierte a binaria usando la función de MatLab *im2bw* con parámetro 0.6, figura 9. En este ejemplo se toma como matriz de semejanza S, a la componente de saturación del modelo HSV de la imagen a color.

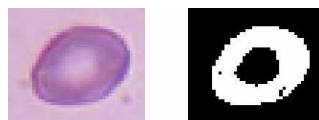


Figura 9

A la izquierda aparece la imagen original y a la derecha la imagen obtenida al transformar la primera con la función de MatLab *im2bw* con parámetro 0.6.

En la figura 10 se aprecia en la primera fila, de izquierda a derecha la erosión producida en la imagen con el algoritmo computacional implementado en este trabajo para los valores de tolerancia  $m=0.015$ ,  $m=0.03$  y

$m = 0.047$  respectivamente. En la segunda fila, de izquierda a derecha, se muestran los resultados de la erosión con la función *imerode* del MatLab usando como elemento estructurante un disco de radio 1, 2 y 3 respectivamente.

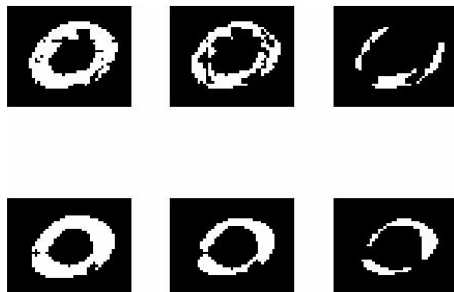


Figura 10

Primera fila de izquierda a derecha la erosión con el algoritmo computacional propuesto para los valores de tolerancia  $m=0.015$ ,  $m=0.03$  y  $m=0.047$  respectivamente. Segunda fila de izquierda a derecha erosión con la función de MatLab *imerode*, usando como elemento estructurante un disco de radios 1,2 y 3 respectivamente.

Se puede apreciar, que la erosión producida por el algoritmo programado en el presente trabajo no es uniforme en todas las partes de la imagen, a diferencia de la erosión producida por la función *imerode* del MatLab. Esta falta de uniformidad en la erosión del algoritmo propuesto está dada por el uso de elementos estructurantes que varían en la imagen tanto en forma como en tamaño, de acuerdo a la función de semejanza y al nivel de tolerancia.

## RESULTADOS Y DISCUSIÓN

Los resultados obtenidos en el procesamiento de imágenes binarias con el uso de la implementación en MatLab de la morfología matemática de las vecindades adaptativas han sido mostrados con imágenes binarias. Esta morfología matemática es variable en el espacio y construye un elemento estructurante para cada píxel de la imagen teniendo en cuenta cierta función de semejanza. En las imágenes binarias que fueron procesadas (dilatadas y erosionadas) se usaron como funciones de semejanza entre los píxeles la componente de saturación S y la componente de valor V del modelo HSV de las imágenes originales a color. Es de notar que la dilatación y la erosión producida por el algoritmo implementado muestran resultados diferentes a los que se obtienen por las operaciones morfológicas convencionales. Estas diferencias están dadas por el hecho de que el elemento estructurante construido por la morfología de las vecindades adaptativas para cada píxel es variable en el espacio y varía tanto en tamaño como en forma, a diferencia de los elementos estructurantes utilizados en las operaciones morfológicas invariantes en el espacio.

Tanto las funciones de semejanza utilizadas, como los valores de tolerancia utilizados han sido escogidos de forma heurística para mostrar el funcionamiento del algoritmo computacional. Es de esperar que un estudio profundo sobre la forma de elegir estos parámetros proporcione mejores resultados que los mostrados en este trabajo para aplicaciones específicas. Esta constituye una línea de trabajo a desarrollar para el futuro, o sea, tratar de inferir para ciertas aplicaciones que función de semejanza utilizar con el fin de lograr algún objetivo con las imágenes a procesar, así como estimar rangos de valores para el nivel de tolerancia que sean los más útiles para cada aplicación.

## CONCLUSIONES

En este artículo se presentó la implementación en MatLab de las operaciones básicas de erosión y dilatación para la morfología matemática de las vecindades adaptativas. Se mostró el funcionamiento de estas operaciones sobre imágenes binarias, comparando los resultados con los obtenidos al usar las operaciones de erosión y dilatación invariantes en el espacio. Las funciones de semejanza usadas fueron las componentes de saturación S y de valor V del modelo HSV de las imágenes originales a color. Un estudio más detallado sobre como escoger esta función de semejanza, así como el valor de tolerancia para aplicaciones más específicas debe brindar mejores resultados en el uso de estas operaciones cuando forman parte de procesos más complejos como pueden ser la apertura, el cierre, aplicaciones de filtrado y de reconstrucción morfológica.

## REFERENCIAS

1. **GONZALEZ, R. C. y WOODS, R. E.:** *Digital Image Processing*. Prentice-Hall, Inc. New Jersey 2002.
2. **SOILLE, P.:** *Morphological Image Analysis*. Springer Verlag. Berlin Heidelberg 2003.
3. **BOUAYNAYA, NIDHAL; CHARIF-CHEFCHAOUNI, MOHAMMED y SCHONFELD, DAN:** “Theoretical Foundations of Spatially-Variant Mathematical Morphology Part I: Binary Images” en *IEEE Transactions on pattern analysis and machine intelligence*, Vol. XXX. No 5, 2008.
4. **GONZALEZ, R. C.; WOODS, R. E. y EDDINS, S. L.:** *Digital Image Processing using MatLab*. Prentice-Hall, Inc. New Jersey 2004.
5. **DEBAYLE, JOHAN y PINOLI, JEAN-CHARLES:** “Spatially Adaptive Morphological Image Filtering Using Intrinsic Structuring Elements” en *Image Anal Stereol*, 2005.

## AUTORES

**Yosbel Morales Olivera**, Licenciado en Matemática, Máster en Ciencias, Universidad de Camagüey “Ignacio Agramante y Loynaz”, Camagüey, Cuba, correo electrónico: [yosbel.morales@reduc.edu.cu](mailto:yosbel.morales@reduc.edu.cu)

**Josué García Parrado**, estudiante de cuarto año de la carrera de Ingeniería Eléctrica, Universidad de Camagüey “Ignacio Agramante y Loynaz”, Camagüey, Cuba, correo electrónico: [Josue.garcia@reduc.edu.cu](mailto:Josue.garcia@reduc.edu.cu)

**Pablo Enrique Reyes Fernández**, estudiante de cuarto año de la carrera de Ingeniería Eléctrica, Universidad de Camagüey “Ignacio Agramante y Loynaz”, Camagüey, Cuba, correo electrónico: [pablo.reyes@reduc.edu.cu](mailto:pablo.reyes@reduc.edu.cu)

**Juan Valentín Lorenzo Ginori**, Ingeniero Eléctricista, Doctor en Ciencias Técnicas, Universidad Central “Marta Abreu”, Villa Clara, Cuba, correo electrónico: [juanl@uclv.edu.cu](mailto:juanl@uclv.edu.cu)