

Desarrollo de una plataforma astroinformática para la administración y análisis inteligente de datos a gran escala

Búsqueda Semántica

Mauricio Solar, Marcelo Mendoza, Gabriel Candia,
Jorge Ibsen, Lars Nyman, Eduardo Vera, Diego Mardones, Guillermo Cabrera,
Paola Arellano, Karim Pichara, Paulina Troncoso, Ricardo Contreras,
Victor Parada.

Santiago, 25 de septiembre de 2014

Resumen

En este documento se informa el estado de avance del entregable "Buscador semántico", lo que corresponde a la documentación para el hito 3 de este producto. Se informa sobre el estado del indexador semántico para catalogación de objetos astronómicos en ChiVO. Se incluyen seudocódigos y pruebas de correctitud.

Palabras Claves: Thresholding, binarización, catalogación de objetos, buscador semántico

1. Background

Sea f un kernel de convolución definido como:

$$f = \text{Matrix}_{\text{filas}, \text{columnas}}(1),$$

donde cada entrada de la matriz es 1. Un kernel uniforme se puede obtener realizando la siguiente operación sobre f :

$$f = \frac{f}{\sum(f)}.$$

Sea X la imagen original en escala de grises y sea Y una representación binaria de X . Un algoritmo de *adaptive thresholding* crea Y desde X como sigue:

$$Y(i, j) = \begin{cases} 1, & X(i, j) > f * X \\ 0, & \text{e.t.o.c.} \end{cases}$$

Notemos que el símbolo $*$ representa el operador convolución. Luego, un pixel de Y es 1 si la convolución entre f y X (centrada en (i, j)) es menor que la intensidad de $X(i, j)$. Por tanto, un 1 en Y representa un pixel de X cuya intensidad es mayor que la intensidad relativa de su vecindario.

Los operadores morfológicos trabajan usualmente sobre imágenes binarias. En específico, consideraremos técnicas morfológicas que buscan la estructura geométrica de los objetos de la imagen usando formas denominadas elementos estructurantes. Algunas operaciones verifican cuando estos elementos ajustan a los objetos de la imagen mientras otras operaciones verifican cuando intersectan un determinado objeto.

Dos operadores morfológicos básicos son considerados en este trabajo: erosión y dilatación. La erosión de una imagen binaria Y por un elemento estructurante S , denotada por $Y \ominus S$, produce una nueva imagen binaria G con 1s en todos los pixels de S en los cuales S **ajusta** a Y :

$$G(i, j) = \begin{cases} 1, & \text{si } S \text{ ajusta } Y(i, j) \\ 0, & \text{e.t.o.c.} \end{cases}$$

La dilatación de una imagen Y por un elemento estructurante S , denotado por $Y \oplus S$, produce una nueva imagen binaria G con 1s en todos los pixels de S en los cuales S **intersecta** a Y :

$$G(i, j) = \begin{cases} 1, & \text{si } S \text{ intersecta } Y(i, j) \\ 0, & \text{e.t.o.c.} \end{cases}$$

La erosión contrae una imagen a sus elementos estructurantes mientras que la dilatación agrega capas de pixels a los objetos estructurantes de la imagen.

Es posible componer estas operaciones. La apertura de una imagen, denotada por $Y \diamond S$, es una erosión seguida de una dilatación:

$$Y \diamond S = (Y \ominus S) \oplus S$$

La apertura puede separar objetos conectados por delgados arcos de pixels luminicentes. Luego, los elementos que sobreviven a la erosión son restaurados a sus tamaños originales por una dilatación. En consecuencia, la apertura puede ayudar en la detección/caracterización de la geometría de los objetos, removiendo artefactos introducidos por *thresholding* o por ruido.

El *thresholding* adaptativo y las operaciones erosión/dilatación son las componentes fundamentales de nuestros métodos. En relación a los elementos estructurantes, usamos dos elementos en nuestros experimentos: discos y Gaussianas. Esto nos ayuda a revelar el impacto del uso de estructuras con soporte compacto en la caracterización de la geometría de los objetos.

2. El método

El método propuesto en este documento hace uso de imágenes binarizadas para caracterizar la geometría de los objetos. Comenzamos nuestro análisis verificando propiedades de una composición de operadores convencionales *thresholding/opening*. Este es el primer paso de nuestro método.

2.1. Thresholding/opening: análisis exploratorio

Exploramos la sensibilidad a los parámetros al componer *thresholding/opening* para una imagen de referencia con un número conocido de objetos. Sobre la imagen binarizada aplicamos un algoritmo basado en grafos denominado conteo de componentes conexas, verificando el número de objetos contenidos en la imagen.

Como imagen de referencia usamos una imagen del Sloan Digital Sky Survey ^{*}, recuperada desde el catálogo DAS (parámetro run 3630). La composición *thresholding/opening* esta parametrizada por el tamaño del kernel usado en el algoritmo de *thresholding* adaptativo, y por el radio del disco usado como elemento estructurante en la operación de apertura. Medimos hits entre la imagen binarizada y la imagen de referencia original que se encuentra etiquetada con el número real de objetos contenidos en la imagen. Esto nos permite estimar el error absoluto de la medición del número de objetos de la imagen. Describimos al logaritmo del error absoluto en términos del tamaño del kernel (window width), y del radio del disco, ambos parámetros medidos en número de pixels. Los resultados de este análisis exploratorio son mostrados en la Figura 1.

La Figura 1 muestra que la combinación óptima de parámetros está descrita por una superficie convexa con múltiples óptimos locales. Este hallazgo ilustra que la presencia de un número de objetos con diferentes tamaños en la imagen original implica la presencia de varios valores para los parámetros, con buenas propiedades en la detección de objetos. Además, podemos observar que la composición *thresholding/opening* es muy sensible a la elección de los valores de los parámetros. Desde este punto de vista, podemos observar que la superficie convexa con óptimos locales satisface una desigualdad estricta entre

^{*}<http://das.sdss.org>

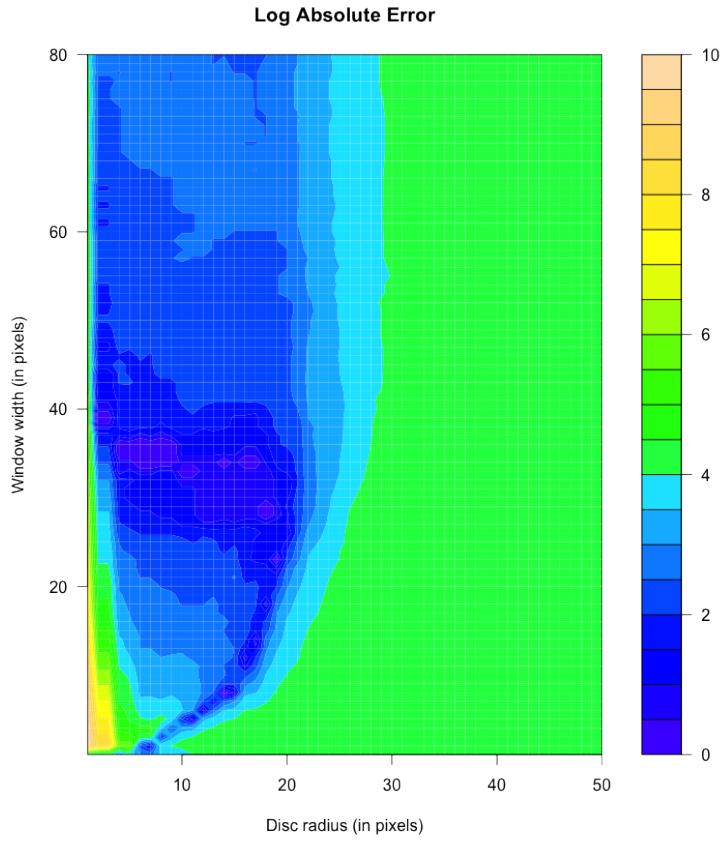


Figura 1: Sensibilidad paramétrica de la secuencia *thresholding/opening*.

los valores de los parámetros, donde el tamaño del kernel de thresholding es siempre menor que el radio del elemento estructurante (disco en este ejemplo).

Concluimos que la presencia de objetos con diferentes tamaños en la imagen original requiere del uso de una representación multiescala para caracterizar la geometría de los objetos de la imagen. Además, como la secuencia *thresholding/opening* es muy sensible a la elección de los parámetros, es necesario definir un procedimiento de sintonización. Exploraremos en la siguiente sección un método simple de estimación de error adecuado para sintonizar automáticamente los parámetros.

2.2. Estimación de error *background/foreground*

Sea bg_p un representante de intensidad del *background* de la imagen, estimado como el valor medio del p -percentil de la distribución de intensidad en una imagen en escala de grises X . Análogamente, definimos fg_p como un representante de intensidad del *foreground* de la imagen, estimada como el valor medio del $100 - p$ percentil de intensidad de X . Exploramos el uso de un método simple de estimación de error en el proceso de binarización de X , considerando dos diferencias relativas, como se define en el siguiente pseudocódigo:

```

1: function BG-FG( $X, Y, fg_p, bg_p$ )
2:   if  $Y(i, j) == 1$  then
3:     if  $|X(i, j) - fg_p| > |X(i, j) - bg_p|$  then
4:       FP++
5:     end if
6:   else
7:     if  $|X(i, j) - fg_p| < |X(i, j) - bg_p|$  then
8:       FN++
9:     end if
10:   end if
11: end function

```

La primera comparación (ver línea 3 del seudocódigo) permite estimar falsos positivos. Como el pixel $X(i, j)$ de la imagen original es más cercano al *background* que al *foreground*, y la composición *thresholding/opening* colocó el pixel en 1, luego contabilizamos el pixel de la imagen binarizada Y como un falso positivo. Análogamente, los falsos negativos pueden ser contabilizados por pixels que se encuentran cercanos al *foreground* pero que son consignados como 0 (ver línea 7 del seudocódigo).

Dos temas surgen en relación a nuestro método de estimación de error background/foreground (método BG-FG, de aquí en adelante): la sensibilidad de BG-FG a la elección de p (el percentil de la distribución de intensidad usada para seleccionar el representante de intensidad), y la adecuación de BG-FG como método de estimación de error para sintonizar el tamaño del kernel.

Comenzamos explorando la sensibilidad de BG/FG al parámetro p . Ilustramos la sensibilidad BG/FG al valor de p tomando valores para p en el rango $\{1, 5\}$ (porcentaje). Dos imágenes de referencia fueron usadas en este análisis exploratorio, cada una de ellas con objetos de geometrías distintas. La primera, que corresponde a la galaxia NGC 6769, y la segunda a la galaxia NGC 1097. La razón de error (falsos positivos + falsos negativos sobre el tamaño de la imagen, en pixels), fue contabilizada para cada valor de p . Los resultados de estas pruebas son mostradas en las figuras 2 y 3.

Las pruebas nos ayudan a identificar algunos hallazgos interesantes. Primero, las curvas BG/FG describen funciones convexas, con un mínimo denotado por \hat{w} . Este hecho es muy relevante, ya que sugiere que este simple método puede ayudar en la estimación de un óptimo para el el tamaño del kernel. Segundo, podemos observar que valores altos de p implican valores altos de \hat{w} . Además, es posible observar una baja varianza en \hat{w} para $p = 3, 4$ y 5 .

Intuición: Como la curva BG/FG es una función optimizada por el conteo de pixels (falsos positivos y falsos negativos), el objeto más grande de X es el que contabiliza más error hits. Luego, \hat{w} ajusta a la componente más grande de X .

Como consecuencia, podemos usar BG-FG como un método de búsqueda para \hat{w} . La lógica es la siguiente. Para un valor de p , contabilizamos la curva BG-FG para distintos tamaños del kernel. Luego, el tamaño del kernel que logra un error más pequeño es una estimación para \hat{w} .

Exploramos en la siguiente sección un método para construir una representación multiescala que comienza usando \hat{w} como valor inicial del elemento estructurante. Luego, aplicando descenso diádico

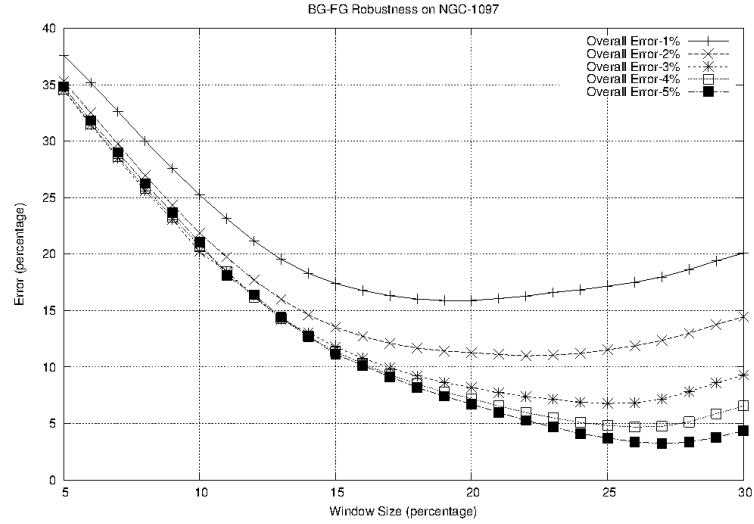


Figura 2: Sensibilidad paramétrica de BG-FG en NGC 1097.

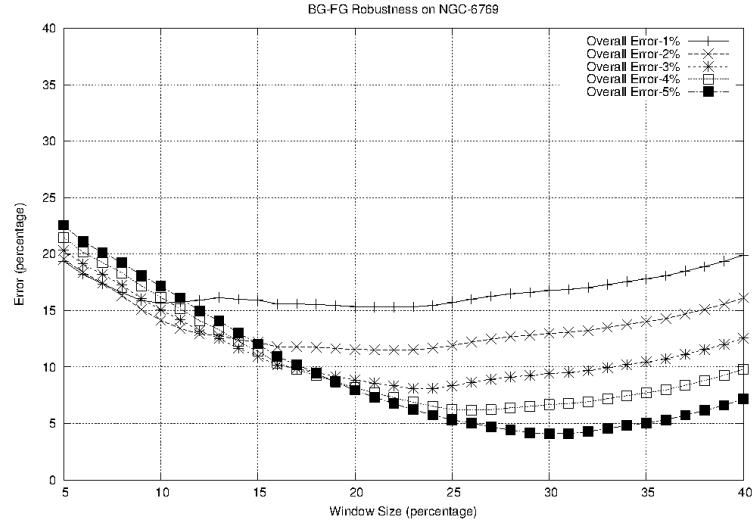


Figura 3: Sensibilidad paramétrica de BG-FG en NGC 6769.

sobre el tamaño del elemento estructurante, conseguimos una secuencia de imágenes. La secuencia es una representación multiescala de X .

2.3. Binarización de la componente gigante

Comenzamos esta sección explorando como \hat{w} se comporta para *adaptive thresholding*. La primera imagen a usar en esta prueba (NGC 1097) y su representación binarizada usando \hat{w} como tamaño del kernel es mostrada en la Figura 4.

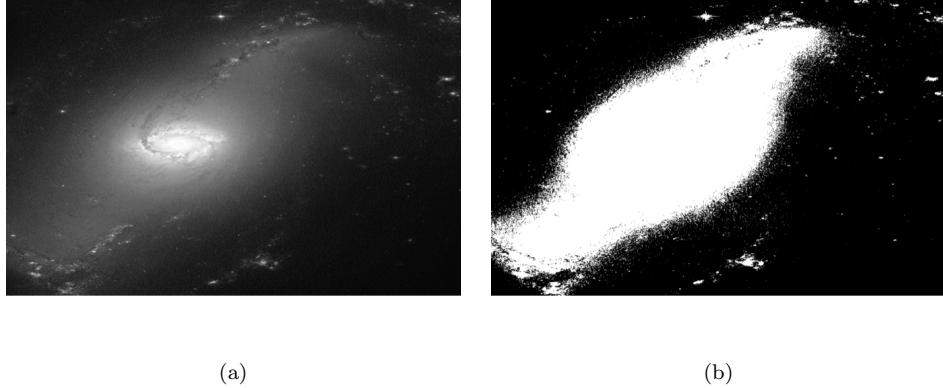


Figura 4: Binarización de la componente gigante de NGC 1097. a) Imagen original en escala de grises, y b) Representación binarizada de NGC 1097.

La Figura 4 muestra que la galaxia es ajustada por la componente binarizada más grande. Un número de pequeñas componentes binarias son también detectadas en esta representación. El tamaño del kernel (window width) usado comprende una matriz cuadrada de 325 pixels, valor que corresponde al valor óptimo \hat{w} , de acuerdo a los resultados obtenidos por el método BG-FG.

Exploramos también la binarización de la componente gigante en una segunda imagen (NGC 6769). Estos resultados son mostrados en la Figura 5.

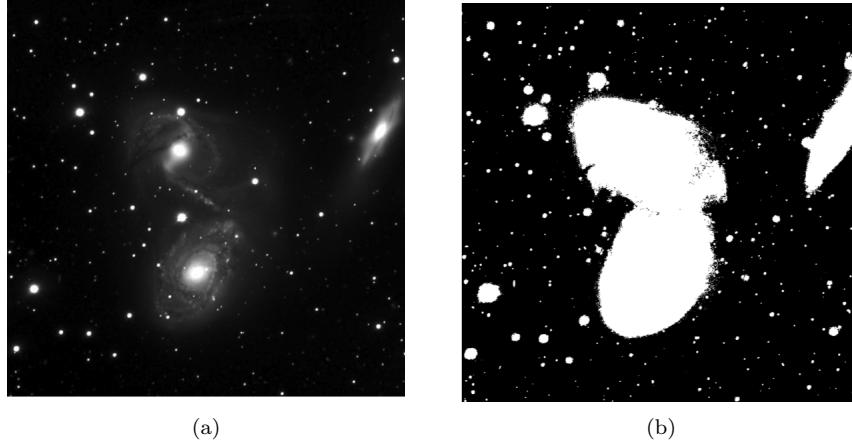


Figura 5: Binarización de la componente gigante de NGC 6769. a) Imagen original en escala de grises, y b) Representación binarizada de NGC 6769.

La Figura 5 muestra que las tres componentes más grandes de la imagen son ajustadas por nuestra estrategia. En este caso, cada componente es aproximada por una envoltura (hull), y las dos componentes centrales son representadas por dos máscaras binarias traslapadas, cubriendo algunas partes de la imagen

que exhiben luminancia.

Proponemos usar \hat{w} como valor máximo para r (el radio del disco). Exploraremos como se comporta la apertura de la imagen usando búsqueda diádica para r usando \hat{w} como valor inicial ($\hat{w}, \frac{\hat{w}}{2}, \frac{\hat{w}}{4}, \dots$), produciendo una lista de representaciones escaladas de la imagen original.

2.4. Representación multiescala de la imagen en escala de grises

En esta sección exploramos como se comporta la apertura de la imagen usando búsqueda diádica sobre r (el radio del disco) como \hat{w} ($\hat{w}, \frac{\hat{w}}{2}, \frac{\hat{w}}{4}, \dots$). Esta operación produce una secuencia de versiones escaladas de la iagen original. Comenzamos nuestro análisis descomponiendo NGC 1097. Seis versiones escaladas de la imagen son mostradas para radios de disco en apertura en valores $\hat{w}, \frac{\hat{w}}{2}, \frac{\hat{w}}{4}, \frac{\hat{w}}{8}, \frac{\hat{w}}{16}$, y $\frac{\hat{w}}{32}$. Estos resultados están en la Figura 6.

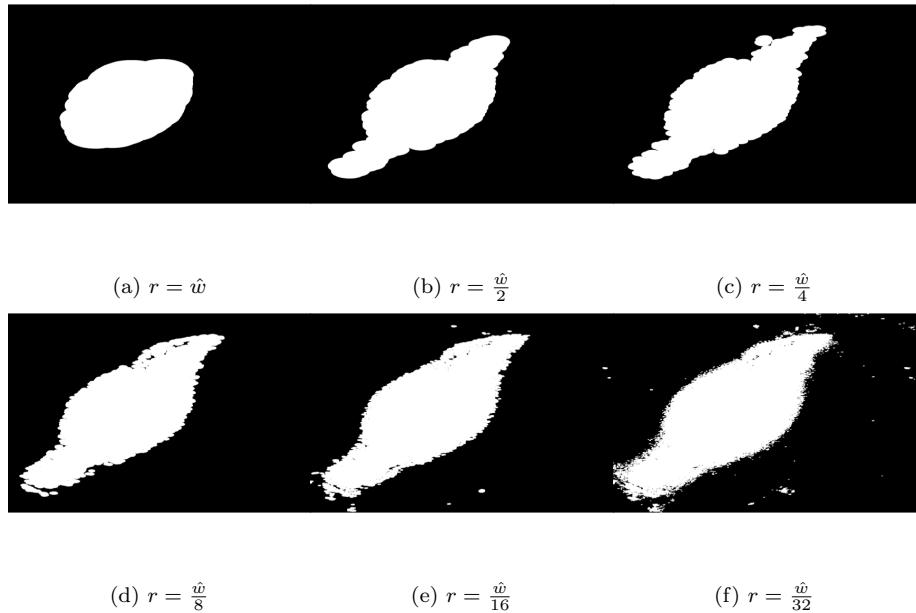


Figura 6: Representación multiescala de NGC 1097.

Como esta secuencia de imágenes muestra, el uso de búsqueda diádica en apertura permite obtener una representación multiescala binarizada de la imagen original. Podemos observar también que la secuencia permite obtener más detalles para valores pequeños del radio. En general, podemos establecer que a medida que el radio es más pequeño, los detalles de la representación son mayores. Sin embargo, esta representación tiene una limitante. La componente gigante es traslapada por sus componentes más pequeñas. Luego, la estructura interna de la componente gigante se pierde. Podemos observar que la componente gigante está conformada por la superposición de elementos estructurantes. Abordaremos esta limitación en la siguiente sección explorando una variante del método propuesto.

2.5. Segmentación multiescala de imágenes en escala de grises

En esta sección introducimos un método capaz de descomponer una imagen binaria en una secuencia de componentes no traslapadas. Hacemos esto aproximando cada componente detectada por una mezcla de Gaussianas, sustrayendo las Gaussianas de la imagen y ejecutando una nueva apertura sobre la imagen transformada. Esta iteración permite descomponer la imagen original en una secuencia de imágenes con elementos estructurantes no traslapados.

Sea g una Gaussiana centrada en el centroide de la componente gigante de una representación binarizada G . Consideramos que el radio del disco, denotado por r , y usado en la apertura, es la desviación 3σ de la Gaussiana. Notemos que G puede tener muchas componentes. Consecuentemente, centrando las gaussianas en cada una de las componentes detectadas de G , creamos una versión aproximada de G .

Sea $Diff$ una iagen en escala de grises calculada desde la diferencia entre la imagen original X y la mezcla de Gaussianas, que denotaremos por Mix_f , la cual aproxima a G . Proponemos usar $Diff$ como imagen de entrada para la composición *thresholding/opening*. Luego, usando $Diff$ logramos una nueva representación binarizada de X , con componentes a una escala definida por el tamaño del kernel y de los elementos estructurantes. Notemos que podemos repetir este proceso dando paso a un algoritmo iterativo. Este algoritmo está definido por el siguiente pseudocódigo:

```

1: function MULTISCALE( $X, p$ )
2:    $r_{min} \leftarrow \text{SIZE}(X)/100$ 
3:    $w_{max} \leftarrow \text{OPTIMAL}(X, p)$ 
4:    $Diff \leftarrow X$ 
5:    $Y \leftarrow \text{THRESHOLDING}(Diff, w_{max})$ 
6:    $r \leftarrow \frac{w_{max}}{2}$ 
7:   while  $r > r_{min}$  do
8:      $G \leftarrow Y \diamond S_r$ 
9:      $\text{objects} \leftarrow \text{BWLABEL}(G)$ 
10:    for  $\text{obj} \in \text{objects}$  do
11:       $\text{kernel} \leftarrow \text{KERNELSMOOTH}(\text{obj.radius})$ 
12:       $Mix_f.append(\text{kernel}, \text{obj.centroid})$ 
13:    end for
14:     $Diff \leftarrow Diff - Mix_f$ 
15:     $Y \leftarrow \text{THRESHOLDING}(Diff, r)$ 
16:     $r \leftarrow \frac{r}{2}$ 
17:  end while
18: end function
```

```

1: function OPTIMAL( $X, p$ )
2:    $r_{min} \leftarrow \text{SIZE}(X)/100$ 
3:    $r_{max} \leftarrow \text{SIZE}(X)/2$ 
4:    $bg_p \leftarrow \text{QUANTILE}(X, p)$ 
5:    $fg_p \leftarrow \text{QUANTILE}(X, 1 - p)$ 
```

```

6:    $r \leftarrow r_{min}$ 
7:   while  $r < r_{max}$  do
8:      $Y \leftarrow \text{THRESHOLDING}(X, r)$ 
9:      $ov \leftarrow \text{BG-FG}(X, Y, fg_p, bg_p)$ 
10:     $list.append(ov)$ 
11:   end while
12:   return  $list.max()$ 
13: end function

```

El algoritmo está codificado en la función denominada `Multiscale`. Comenzamos asignando como valor mínimo para el radio de cada kernel, denominado r_{min} , definido como un porcentaje del tamaño de la imagen. Luego, el radio del kernel que intersecta la componente gigante es determinado llamando a nuestro método `BG-FG`, codificado en la función denominada `Optimal`. Usando w_{max} como tamaño del kernel, ejecutamos el algoritmo de thresholding adaptativo sobre X . Luego, usando búsqueda diádica sobre el radio del disco, un ciclo de operaciones de apertura comienza. Este ciclo finaliza cuando el radio es menor o igual que r_{min} . Luego de cada apertura, la imagen binarizada, denotada por G , es procesada para recuperar los objetos, usando el algoritmo de componentes conexas (codificado en `Bwlabel`), y para cada objeto, un kernel Gaussiano es creado, considerando su radio y centroide. La mezcla de Gaussianas (Mix_f) es restada generando $Diff$. Luego, una nueva iteración comienza cuando una nueva versión de X es recuperada binarizando $Diff$ y reduciendo el radio a la mitad.

Los resultados de nuestro método son mostrados en la Figura 7. En este experimento, `Multiscale` fue ejecutado para NGC 1097, con $p = 0,05$.

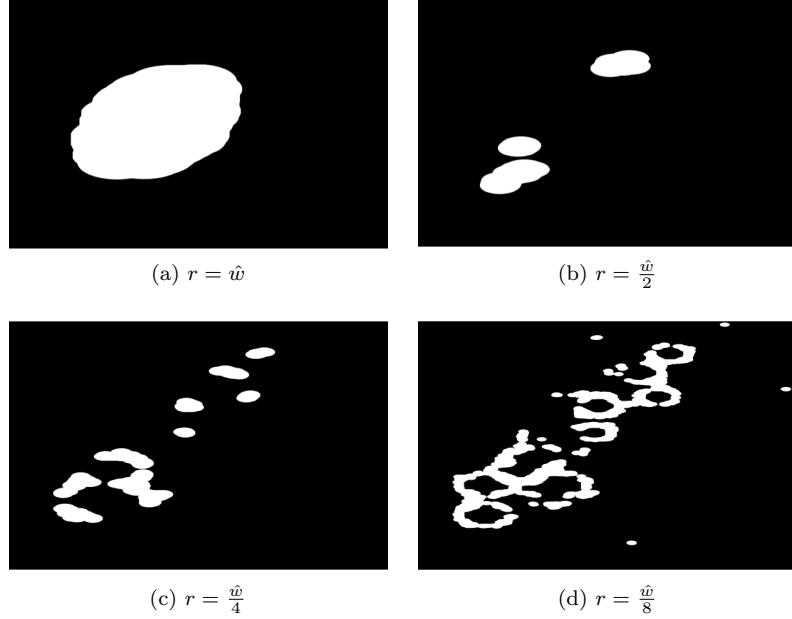


Figura 7: Segmentación multiescala de NGC 1097.

Como la Figura 7 muestra, nuestro método se coporta bien. La primera representación intersecta

la componente gigante de NGC 1097. Además, las siguientes imágenes intersectan componentes más pequeñas de la imagen original, con una baja presencia de traslape. Notemos que para $r = \frac{\hat{w}}{4}$ and $r = \frac{\hat{w}}{8}$, las componentes detectadas pertenecen a la componente gigante, mostrando el hecho que el método propuesto es capaz de segmentar la componente gigante de la iagen con componentes no traslapadas.

La Figura 8 muestra la segmentación multiescala de NGC 6769 con $p = 0,05$.

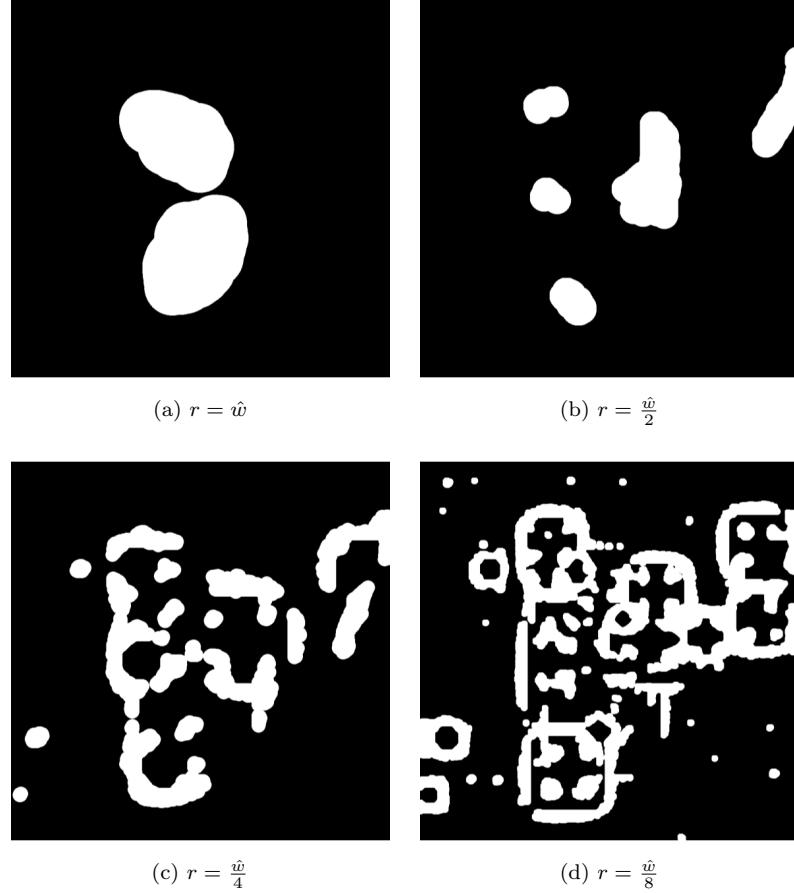


Figura 8: Segmentación multiescala deNGC 6769.

Referencias

- [1] Mukhopadhyay, S., Chanda, B. (2003), *Multiscale morphological segmentation of gray-scale images*. IEEE Transactions on Image Processing, 12(5):533-549.
- [2] Urbach, E., Wilkinson, M. (2008), *Efficient 2D grayscale morphological transformation with arbitrary flat structuring elements*. IEEE Transactions on Image Processing, 17(1):1-8.