

UNIVERSIDAD CATÓLICA DEL MAULE
Facultad de Ciencias de la Ingeniería
Magíster en Ciencias de la Computación

PROFESOR GUÍA
Dr. Marco Mora Cofré

**Extensión de un Algoritmo Inspirado en Colonia de
Hormigas para la Detección de Contornos en
Imágenes con Ruido Multiplicativo.**

Sergio Antonio Baltierra Valenzuela

Tesis de grado presentada en conformidad a los requisitos para
obtener el grado de Magíster en Ciencias de la Computación.

TALCA, ABRIL 2014

**UNIVERSIDAD CATÓLICA DEL MAULE
FACULTAD DE CIENCIAS DE LA INGENIERÍA
MAGÍSTER EN CIENCIAS DE LA COMPUTACIÓN**

**TESIS PARA OPTAR AL
GRADO DE MAGÍSTER EN CIENCIAS DE LA COMPUTACIÓN**

**Extensión de un Algoritmo Inspirado en Colonia de Hormigas para la Detección
de Contornos en Imágenes con Ruido Multiplicativo.**

Sergio Antonio Baltierra Valenzuela

COMISIÓN EXAMINADORA

FIRMA

Supervisor

Dr. Marco Mora Cofré

Universidad Católica del Maule

Revisor Interno

M. Sc. Paulo González

Universidad Católica del Maule

Revisor Externo

M. Sc. María Angélica Pinninghoff

Universidad de Concepción

Revisor Externo

M. Sc. Ricardo Contreras

Universidad de Concepción

Revisor Externo

Dra. María Juliana Gambini

Instituto Tecnológico de Buenos Aires
Argentina

NOTA FINAL EXAMEN

TALCA, ABRIL 2014

Agradecimientos

En primer lugar agradecer a Dios por no dejarme caer en los momentos difíciles.

En segundo lugar agradecer a mis padres que me dieron nuevamente su apoyo en esta nueva etapa de mi vida profesional y que sin ellos esta tesis mas que seguro no existiría.

A mi hermano que espero que saque un magíster y se de cuenta que con esfuerzo y perseverancia todo se puede.

A mis abuelas que día a día se preocupan y me dan apoyo incondicional.

A mi profesor guía, Dr. Marco Mora, por sus consejos y conocimientos que me permitieron seguir el camino a pesar del poco tiempo que tuvo. Además de facilitarme su equipo para terminar las pruebas.

A mis amigos Jorge y Sabino que me dieron sus fuerzas y apoyo como buenos amigos que son.

A todas las personas que conocí en algún momento en el transcurso de esta tesis y me dieron su apoyo, sin conocerme, sin saber de mi, sólo dieron sus buenas vibras, suficiente para que uno siga adelante.

Sergio Antonio Baltierra Valenzuela.

Abril 2014

Resumen

En esta tesis se propone una mejora de un algoritmo para la detección de contornos en imágenes con ruido multiplicativo usando Ant Colony System.

Primero se selecciona un algoritmo de la literatura que tiene como característica el uso de dos tipos de feromonas. La primera se denomina feromona de tipo global el cual usa el gradiente como estímulo para detector de contornos. La segunda se denomina feromona de tipo local y es la que deposita cada hormiga al detectar contornos en la imagen.

Se selecciona e implementa un algoritmo que use Ant Colony System para la detección de contornos propuesto en la literatura.

Se realiza una adaptación del algoritmo usando Ant Colony System para la detección de contornos, modificando algunas reglas y parametros.

Se realiza una mejora de un algoritmo que utiliza Ant Colony System para la detección de contornos en imágenes con ruido multiplicativo. La mejora se basa en usar como feromona de tipo global el Coeficiente de Variación.

Para una evaluación objetiva se utiliza funciones de performance permitiendo comparar y analizar el método propuesto versus el de la literatura.

Las pruebas se clasifican en dos partes: En la primera se realizan pruebas usando un ambiente controlado a partir de sintéticas. La segunda consiste en realizas pruebas con imágenes reales.

Finalmente se procede a comparar los resultados entregados por la evaluación objetiva de ambos algoritmos. El análisis de los resultados de la evaluación se realizan mediante gráficas y tablas comparativas.

Índice

1. Introducción	2
1.1. Planteamiento del Problema	2
1.2. Propuesta de Tesis	3
1.3. Hipótesis	3
1.4. Objetivos	3
1.4.1. Objetivo General	3
1.4.2. Objetivos Específicos	4
1.5. Contribución	4
2. Fundamentos del Trabajo	5
2.1. Ant Colony System	5
3. Método de Detección de Contornos basado en ACS y Coeficiente de Variación	8
3.1. Adaptación de ACS para la Detección de Contornos	8
3.2. Adaptación para Detección de Contornos en Imágenes con Ruido Multiplicativo	9
4. Metodología de los Experimentos	11
5. Resultados	15
5.1. Resultados Imagen Sintética	15
5.2. Resultados Imágenes Reales	22
6. Conclusiones y Trabajos Futuros	30

Índice de Figuras

1.	(a) Vecindad de Moore (b) Vecindad de Von Neumann	8
2.	Evaluación de Performance.	14
3.	Imagen Sintética Simple: (a) Imagen Original, (b) Terreno de la verdad, (c) Bajo nivel de ruido ($\sigma = 0,05$), (d) Nivel medio de ruido ($\sigma = 0,1$), (e) Alto nivel de ruido ($\sigma = 0,15$), (f) Histograma de bajo nivel de ruido, (g) Histograma de nivel medio de ruido, y (h) Histograma de alto nivel de ruido.	16
4.	Contornos detectados en un imagen sintética: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente A (LLN-GAC-A), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente A (MLN-GAC-A), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente A (HLN-GAC-A).	17
5.	Contornos detectados en un imagen sintética: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente B (LLN-GAC-B), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente B (MLN-GAC-B), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente B (HLN-GAC-B).	17
6.	Contornos detectados en un imagen sintética: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente C (LLN-GAC-C), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente C (MLN-GAC-C), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente C (HLN-GAC-C).	18
7.	Contornos detectados en un imagen sintética: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente D (LLN-GAC-D), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente D (MLN-GAC-D), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente D (HLN-GAC-D).	18
8.	Contornos detectados en un imagen sintética: (a) Bajo nivel de ruido de Coeficiente de Variación Ant Colony (LLN-CVAC), (b) Nivel medio de ruido de Coeficiente de Variación Ant Colony (MLN-CVAC), y (c) Alto nivel de ruido de Coeficiente de Variación Ant Colony (HLN-CVAC).	19
9.	Función de performance GAC-A, GAC-B, GAC-C, GAC-D y CVAC para tres niveles de ruido: (a) Bajo nivel de ruido, (b) Nivel medio de ruido, (c) Alto nivel de ruido.	20
10.	Función de performance óptimo para niveles continuos de ruido: (a) 20 muestras del mismo nivel de ruido y (b) función de performance promedio.	22
11.	Imagen real: (a) Imagen Original, (b) Terreno de la verdad estimado, (c) Bajo nivel de ruido ($\sigma = 0,05$), (d) Nivel medio de ruido ($\sigma = 0,1$), (e) Nivel alto de ruido ($\sigma = 0,15$), (f) Histograma de bajo nivel de ruido, (g) Histograma de nivel medio de ruido, y (h) Histograma de alto nivel de ruido.	23
12.	Contornos detectados en un imagen real: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente A (LLN-GAC-A), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente A (MLN-GAC-A), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente A (HLN-GAC-A)	24

13.	Contornos detectados en un imagen real: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente B (LLN-GAC-B), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente B (MLN-GAC-B), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente B (HLN-GAC-B)	24
14.	Contornos detectados en un imagen real: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente C (LLN-GAC-C), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente C (MLN-GAC-C), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente C (HLN-GAC-C)	25
15.	Contornos detectados en un imagen real: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente D (LLN-GAC-D), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente D (MLN-GAC-D), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente D (HLN-GAC-D)	25
16.	Contornos detectados en un imagen real: (a) Bajo nivel de ruido de Coeficiente de Variación Ant Colony (LLN-CVAC), (b) Nivel medio de ruido de Coeficiente de Variación Ant Colony (MLN-CVAC), y (c) Alto nivel de ruido de Coeficiente de Variación Ant Colony (HLN-CVAC).	26
17.	Función de performance GAC-A, GAC-B, GAC-C, GAC-D y CVAC para tres niveles de ruido de la imagen real: (a) Bajo nivel de ruido, (b) Nivel medio de ruido y (c) Alto nivel de ruido.	27
18.	Función de performance óptimo para niveles continuos de ruido: (a) 20 muestras del mismo nivel de ruido y (b) función de performance promedio.	29

Índice de Tablas

1. Valores óptimos de la función de performance para el bajo nivel de ruido. 21
2. Valores óptimos de la función de performance para el nivel medio de ruido. 21
3. Valores óptimos de la función de performance para el alto nivel de ruido. 21
4. Valores óptimos de la función de performance para el bajo nivel de ruido. 28
5. Valores óptimos de la función de performance para el nivel medio de ruido. 28
6. Valores óptimos de la función de performance para el alto nivel de ruido. 28

1. Introducción

1.1. Planteamiento del Problema

La extracción de información desde las imágenes permite dar soluciones a problemas complejos en áreas tan dispares como la medicina, geociencias, control industrial, astronomía, entre otros.

El ruido es un factor limitante en el método de extracción de características como son los contornos. Uno de los más agresivos ruidos existentes donde muchos esfuerzos se encuentran en la literatura es el ruido multiplicativo [11, 3, 7, 18, 16].

El método más utilizado para la detección de contornos es el Coeficiente de Variación (CV) [12, 6, 10, 13].

Los algoritmos Bioinspirados han sido adaptados para la extracción de características en imágenes. Sin embargo estos no están adaptados para imágenes ruidosas.

En este trabajo proponemos extender el algoritmo Ant Colony System (ACS) propuesto en [4] para la detección de contornos en imágenes con ruido multiplicativo.

Un trabajo pionero en esta línea se encuentra en [15]. En dicho trabajo, las hormigas son depositadas en forma aleatoria sobre los píxeles de la imagen, y se mueven de forma probabilística hacia el píxel vecino (se considera una vecindad de 8 elementos). La función de probabilidad que rige el desplazamiento de las hormigas tiene relación con la intensidad de los píxeles, y al final del algoritmo se obtiene como resultado una matriz llamada Matriz de Feromona que es la que contiene la feromona depositada por todas las hormigas al detectar contornos. Para obtener los contornos se umbraliza la Matriz de Feromona.

En [17] se implementa ACS con una vecindad adaptada al tipo y forma de la imagen. Para desplazar la hormiga a un píxel se considera una función de visibilidad, la cual se determina aplicando un estadístico sobre una vecindad de la imagen que es denominada “clique”. La función de visibilidad se computa mediante funciones lineales, cuadráticas o trigonométricas. De forma no automática, en el trabajo se determina el tipo de expresión adecuada a la imagen que se está procesando.

En [2] se comparan dos algoritmos de la familia Ant Colony en la detección de contornos: el algoritmo Ant System (AS) y el algoritmo ACS. Concluyendo que ACS debido a su mejor estructura probabilística para determinar el desplazamiento de las hormigas, permitiendo estimar de mejor forma los contornos en la imagen.

En [5] se definen dos tipos de feromonas, una de tipo global y otra de tipo local. La feromona de tipo global se computa mediante el gradiente de la imagen. Se utiliza la feromona global como estímulo para que las hormigas inicien el proceso de búsqueda. A medida que las hormigas recorren la imagen, éstas depositan una segunda feromona que es denominada feromona de tipo local, la cual es independiente de la feromona global. Al final del algoritmo se suman ambos tipos de feromonas y se realiza una umbralización para estimar los contornos.

1.2. Propuesta de Tesis

En esta tesis se propone el cómputo de la feromona de tipo global basado en el Coeficiente de Variación, en vez del gradiente como se propone en [5]. Este estadístico es ampliamente utilizado en el tratamiento de imágenes afectadas por speckle.

1.3. Hipótesis

Es posible adaptar y mejorar un algoritmo para la detección de contornos en imágenes con ruido multiplicativo usando Ant Colony System.

1.4. Objetivos

1.4.1. Objetivo General

Adaptar y mejorar un algoritmo para la detección de contornos en imágenes con ruido multiplicativo usando Ant Colony System.

1.4.2. Objetivos Específicos

Para llevar a cabo el objetivo general, se propone los siguientes objetivos específicos:

- Mejorar el algoritmo Ant Colony System, para la detección de contornos en imágenes con ruido multiplicativo, en base al coeficiente de variación.
- Definir una función de performance para la comparación objetiva de detección de contornos.
- Comparar la performance del algoritmo mejorado y el algoritmo original considerando imágenes sintéticas y reales.

1.5. Contribución

En esta tesis se ha desarrollado un nuevo algoritmo para detectar contornos en imágenes con ruido multiplicativo en base a Ant Colony System y el coeficiente de variación.

La contribución anterior ha permitido el desarrollo y la submisión del siguiente artículo: Sergio Baltierra and Marco Mora, “Edge Detection in Images Multiplicative Noise by using Ant Colony System”, submitted to Applied Soft Computing, January 31, 2014. El texto completo de este artículo se presenta en anexo.

2. Fundamentos del Trabajo

2.1. Ant Colony System

El algoritmo ACS fue propuesto como solución al problema del vendedor viajero (TSP) [4]. Dado un conjunto de ciudades a visitar, el problema consiste en encontrar la ruta más corta de todas las posibles combinaciones de rutas, partiendo y terminando en la misma ciudad. La inspiración del algoritmo ACS proviene del comportamiento de una colonia de hormigas. Las hormigas mientras se desplazan van depositando feromona, y el camino más corto resulta ser el que tiene una mayor cantidad de feromona depositada.

El modelamiento del comportamiento de las hormigas se realiza fundamentalmente en base a 2 expresiones. La primera regla hacia dónde se mueve la hormiga y se denomina Regla de Transición de Estado. La segunda define la cantidad de feromona depositada y se denomina Regla de Actualización de Feromona. En particular, el algoritmo ACS trabaja con 2 reglas de actualización de feromonas, una Regla de Actualización Local y otra Regla de Actualización Global. La Regla de Actualización Local de Feromona se aplica luego de cada paso de cada una de las hormigas. La Regla de Actualización Global de Feromona se aplica finalizando el movimiento de la última hormiga de una iteración. En [4] se define iteración al movimiento de todas las hormigas en un intervalo $(t, t + 1)$, donde t es un paso.

La expresión de la Regla de Transición de Estado es la siguiente:

$$j = \begin{cases} \operatorname{argmax}_{j \in J_k(i)} (\tau(i, j) \cdot (\eta(i, j))^\beta) & \text{si } q \leq q_0 \\ P(i, j) & \text{otherwise} \end{cases}, \quad (1)$$

donde j es la ciudad destino, i es la ciudad de origen, $\tau(i, j)$ es la intensidad de rastro de feromona depositada, $\eta(i, j)$ es la visibilidad (distancia) entre la ciudad i y la ciudad j , β es un parámetro que determina la importancia relativa de la feromona respecto de la distancia, q es un número aleatorio de distribución uniforme en $[0, 1]$, y q_0 es un parámetro ($0 \leq q_0 \leq 1$) que determina el tipo de búsqueda que realizan las hormigas. Si $q \leq q_0$ la búsqueda se denomina por explotación, en caso contrario, la búsqueda se

denomina por exploración. La expresión de $P(i, j)$ viene dada por la siguiente expresión:

$$P(i, j) = \begin{cases} \frac{\tau(i, j) \cdot \eta(i, j)^\beta}{\sum_{j \in J_k(i)} \tau(i, j) \cdot \eta(i, j)^\beta} & \text{si } j \in J_k(i) \\ 0 & \text{otherwise} \end{cases}, \quad (2)$$

donde J_k es el conjunto de ciudades o nodos que van a ser visitados por las k -ésimas hormigas posicionadas en las ciudades i . La Regla de Transición de Estado favorece la transición hacia los nodos conectados por caminos cortos y con una gran cantidad de feromona depositada.

La Regla de Actualización Local de Feromona tiene la siguiente expresión:

$$\tau(i, j) = (1 - \varphi) \cdot \tau(i, j) + \varphi \cdot \tau_{ini} \quad (3)$$

donde $\varphi \in]0, 1]$ es un parámetro.

Una vez que todas las hormigas de la colonia han dado un paso, es decir, una iteración del algoritmo, se aplica la Regla de Actualización Global de Feromona, que tiene la siguiente expresión:

$$\tau(i, j) = (1 - \rho) \cdot \tau(i, j)^{n-1} + \rho \cdot \sum_{k=1}^K \Delta\tau(i, j)^k \quad (4)$$

donde ρ es el coeficiente de evaporación, y $\Delta\tau(i, j)^k$ esta dado por la siguiente expresión:

$$\Delta\tau(i, j)^k = \begin{cases} \frac{1}{L_k} & \text{si } (i, j) \in \text{tour de la hormiga } k \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

donde L_k es el largo de la ruta de la k -ésima hormiga.

Las ecuaciones (4) y (5) de la Regla de Actualización Global de Feromona es aplicada sólo a la mejor hormiga, es decir a la ruta más corta, según [4]. Debido a que las rutas cortas se les deposita gran cantidad de feromona con la Regla de Actualización Global de Feromona, las rutas largas sufren una mayor evaporación de feromona.

Los pasos del algoritmo ACS son los siguientes:

-
1. Determinar la cantidad “m” de hormigas de la colonia, y la cantidad de iteraciones “n” del algoritmo.
 2. Para cada iteración del algoritmo, y para cada una de las hormigas de la colonia, se realizan los siguientes pasos:
 - a) Aplicar la Regla de Transición de Estado.
 - b) Almacenar la ciudad visitada en una lista denominada Lista Tabú. La idea de esta lista es no visitar una ciudad que ha sido visitada. Al final de todas las iteraciones la Lista Tabú tiene todas las ciudades visitadas por las hormigas.
 - c) Aplicar la Regla de Actualización Local de Feromona.
 3. Aplicar la Regla de Actualización Global de Feromona.
 4. Al final de la última iteración se obtendrá la trayectoria más corta.

3. Método de Detección de Contornos basado en ACS y Coeficiente de Variación

3.1. Adaptación de ACS para la Detección de Contornos

El punto de partida de la detección de contornos mediante un algoritmo de la familia Ant Colony corresponde a la representación de la imagen. Una imagen se entiende como una matriz bidimensional, en que cada pixel es un nodo que está conectado con los nodos de su vecindad local. La figura 1 presenta dos tipos de vecindades ampliamente utilizadas en la literatura. Dada la forma de representación, una hormiga puede avanzar de nodo en nodo como si fuese de ciudad en ciudad.

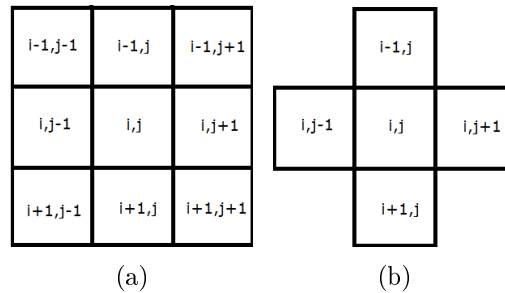


Figura 1: (a) Vecindad de Moore (b) Vecindad de Von Neumann

Para la detección de contornos en imágenes se conserva la Regla de Transición de Estados P , debido a que es de nuestro interés obtener una Matriz de Feromona para estimar los contornos. Por el contrario, en vez de computar la función de visibilidad η en base a la distancia entre los nodos, esta se debe computar en base a la intensidad del pixel de la imagen. Además se conserva la Regla de Actualización Local de Feromona τ . Para el caso de la Regla de Actualización Global de Feromona $\Delta\tau(i, j)^k$, esta se asume como constante [5].

La diferencia fundamental entre el uso ACS para resolver el TSP y detectar contornos es la siguiente: en vez de que la solución sea entregada por el camino recorrido de una hormiga, la solución para la detección de contornos considera el aporte de toda

la colonia de hormigas (a este concepto se le conoce como Swarm Intelligence).

Una vez terminado el algoritmo, se umbraliza la Matriz de Feromona para obtener una estimación de los contornos de la imagen.

3.2. Adaptación para Detección de Contornos en Imágenes con Ruido Multiplicativo

Para la detección de contornos en imágenes, la feromona de tipo global se estima en base al gradiente de la imagen. En particular, es el módulo del gradiente el que se utiliza como detector de contornos. La expresión del detector de contornos es la siguiente:

$$G(I) = \nabla I \quad (6)$$

donde ∇I es el vector pendiente, un vector cuyos componentes corresponden a las derivadas parciales respecto a los vectores linealmente independientes que conforma la dimensión de la imagen. Se usará la notación $\frac{\delta I}{\delta x}$ que expresa la derivada parcial respecto a x y $\frac{\delta I}{\delta y}$ que expresa la derivada parcial respecto a y .

Se define el vector unitario $\vec{u} = [1, 1]$ el cual al operar el producto punto se obtendrá una imagen escalar, mediante la siguiente ecuación:

$$\vec{u} \cdot |\nabla I| = \left| \frac{\delta I}{\delta x} \right| + \left| \frac{\delta I}{\delta y} \right| \quad (7)$$

Como el objetivo de esta tesis es proponer una mejora de [5], se necesita abordar más de un gradiente. Por lo tanto realizamos el cálculo de cuatro tipos de gradientes, los cuales son gradientes numéricos cuyas ecuaciones se muestran a continuación:

- Gradiente A: donde $\frac{\delta I}{\delta x}$ corresponde a la derivada parcial numérica de la imagen con respecto a x , y $\frac{\delta I}{\delta y}$ corresponde a la derivada parcial numérica de la imagen con respecto a y .

- Gradiente B: donde $\frac{\delta I}{\delta x} = \frac{1}{2}(I(i, j + 1) - I(i, j - 1))$ corresponde a la derivada parcial numérica de la imagen con respecto a x , y $\frac{\delta I}{\delta y} = \frac{1}{2}(I(i + 1, j) - I(i - 1, j))$ corresponde a la derivada parcial numérica de la imagen con respecto a y .
- Gradiente C: donde $\frac{\delta I}{\delta x} = (I(i, j) - I(i, j - 1))$ corresponde a la derivada parcial numérica de la imagen con respecto a x , y $\frac{\delta I}{\delta y} = (I(i, j) - I(i - 1, j))$ corresponde a la derivada parcial numérica de la imagen con respecto a y .
- Gradiente D: donde $\frac{\delta I}{\delta x} = (I(i, j + 1) - I(i, j))$ corresponde a la derivada parcial numérica de la imagen con respecto a x , y $\frac{\delta I}{\delta y} = (I(i + 1, j) - I(i, j))$ corresponde a la derivada parcial numérica de la imagen con respecto a y .

Una vez calculado los cuatro gradientes de la imagen I , calculamos el módulo de cada gradiente que corresponde a la siguiente expresión:

$$|G(I)| = \sqrt{\frac{\delta I^2}{\delta x} + \frac{\delta I^2}{\delta y}} \quad (8)$$

Es un hecho conocido que el gradiente no es un detector de contornos robusto al ruido. Debido a la naturaleza del ruido multiplicativo, el gradiente produce una alta tasa de errores en la detección de contornos. Por ello, en esta tesis se propone el cómputo de la feromona de tipo global basado en el Coeficiente de Variación. Este estadístico es ampliamente utilizado en el tratamiento de imágenes afectadas por speckle, tal como se ha explicado en la introducción del trabajo. La expresión del coeficiente de variación es la siguiente:

$$CV(I) = \frac{\sigma}{|\bar{x}|} \quad (9)$$

donde σ es la desviación estándar y $|\bar{x}|$ es el valor absoluto del promedio. Ambos valores se computan sobre la vecindad de tamaño 3×3 , específicamente la vecindad de Von Neumann indicada en la figura 1(b).

4. Metodología de los Experimentos

Para las experimentaciones se adoptan los parámetros de ACS propuestos en [5]. El detalle de dichos parámetros es el siguiente: para la cantidad de hormigas “m” se considera un porcentaje del área de la imagen, y el número de iteraciones “n” como el máximo entre el ancho y largo de la imagen. Los valores de los parámetros restantes son: $\tau_{ini} = 1$, $\alpha = 1$, $\beta = 1$, $\rho = 0,3$ y $\lambda = 0,5$.

Para la evaluación de la performance de la detección de contornos, se propone una mejora de la metodología utilizada en [14]. Se realiza una evaluación objetiva de los resultados de la detección de contornos entre Gradient Ant Colony (GAC) y Coefficient of Variation Ant Colony (CVAC). Para realizar la comparación se debe estimar la distancia entre los contornos obtenidos por cada método y el terreno de la verdad. El terreno de la verdad corresponde a una imagen binaria, en la cual los contornos son pixeles blancos y el resto de los pixeles de la imagen son negros. Para computar la distancia respecto del terreno de la verdad se adopta una función de performance ampliamente utilizada para evaluar la calidad de la detección de contornos [8]. La expresión de la Función de Performance (PF) es:

$$PF = \frac{card(E)}{card(E) + card(FP) + card(FN)} \quad (10)$$

donde $card(X)$ denota el número de elementos del conjunto X , E es el conjunto de pixeles de contornos detectados correctamente, FP es el conjunto de falsos positivos (contornos falsos), y FN es el conjunto de falsos negativos (contornos del terreno de la verdad perdidos por el detector de contornos). El valor de PF es cercano a 1 cuando hay pocos pixeles mal clasificados, y cercano a 0 cuando hay muchos pixeles mal clasificados.

Para contaminar la imagen con ruido multiplicativo se adopta un modelo ampliamente aceptado en la literatura [1, 9]. El modelo usado para contaminar una imagen es el siguiente:

$$I_n = I_o \times n \quad (11)$$

donde I_n es el ruido de la imagen, I_o es la imagen original, y n es el ruido. El ruido corresponde un conjunto de números aleatorios, cuyo histograma sigue una determinada distribución de probabilidad. Típicamente la distribución de probabilidad corresponde a una distribución gaussiana, de media μ y desviación estándar σ . Dado que el ruido multiplica a la imagen se debe considerar una distribución gaussiana de media $\mu = 1$. Para obtener imágenes con diferentes grados de contaminación, se deben generar distribuciones de números aleatorios con distintos valores de σ .

Al aplicar a una imagen en escala de grises un método de detección de contornos, se produce una imagen continua en la cual los contornos corresponden a píxeles claros, y las zonas homogéneas a píxeles oscuros. Debido a que la imagen es continua, para determinar cuáles son los contornos detectados por un método se requiere establecer un umbral. Al aplicar el umbral se obtiene una imagen binaria, la cual puede ser comparada con el terreno de la verdad. Lo anterior significa que el valor de la función de performance depende del umbral escogido, por ello se debe buscar el umbral óptimo (th_{op}) que produce un máximo en la función de performance (PF_{op}). Para comparar dos métodos de detección de contorno es necesario calcular el umbral óptimo para ambos métodos. El mejor método es aquel que tiene la mayor función de performance en su umbral óptimo. La figura 2(a) muestra las curvas de la función de performance para dos métodos de detección de contorno evaluada en distintos umbrales. Los círculos negros indican el valor máximo de la función de performance para cada uno de los métodos.

El método de umbralización se calcula mediante la siguiente ecuación:

$$threshold = k \cdot max(MF) \quad (12)$$

donde $k \in [0, 1]$ y $max(MF)$ es el valor máximo de la Matriz de Feromona.

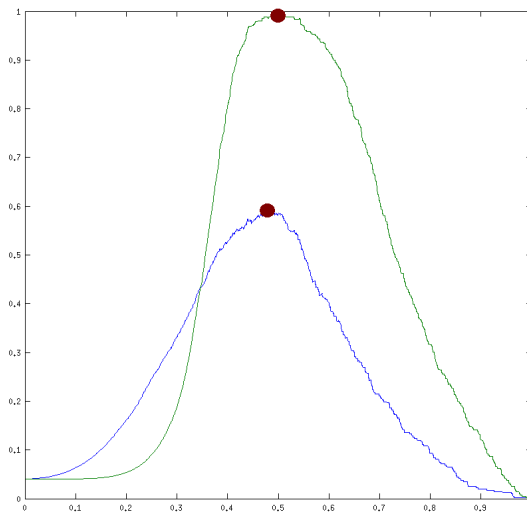
Obteniendo el valor del umbral se procede a calcular los contornos de la imagen mediante la siguiente ecuación:

$$I_{th} = MF > threshold \quad (13)$$

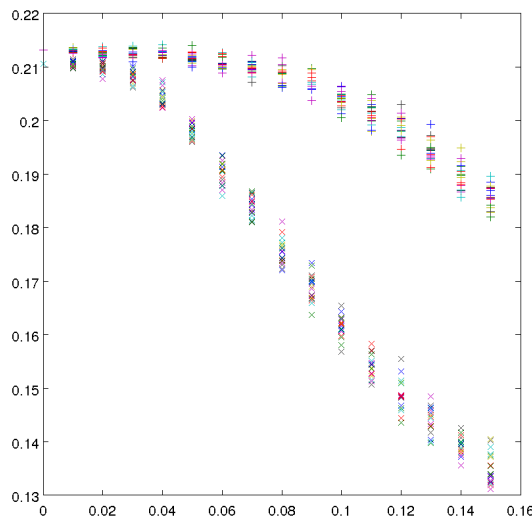
donde I_{th} es una imagen umbralizada. Los valores de MF que son mayores al umbral de la Matriz de Feromona calculados en la ecuación (12) toman el valor 1 para blanco (contorno), de lo contrario toman el valor 0 para negro (no contorno).

Debido a que se está proponiendo un detector de contornos adaptado al ruido, no es suficiente obtener el umbral óptimo para un caso de ruido. Es necesario estudiar el comportamiento de la función de performance con diferentes niveles de ruido. Partiendo con $\sigma = 0$ para una imagen sin ruido, se debe aumentar gradualmente el valor de σ hasta que la función de performance sea aproximadamente igual a 0 (para esta tesis $0 \leq \sigma \leq 0,15$).

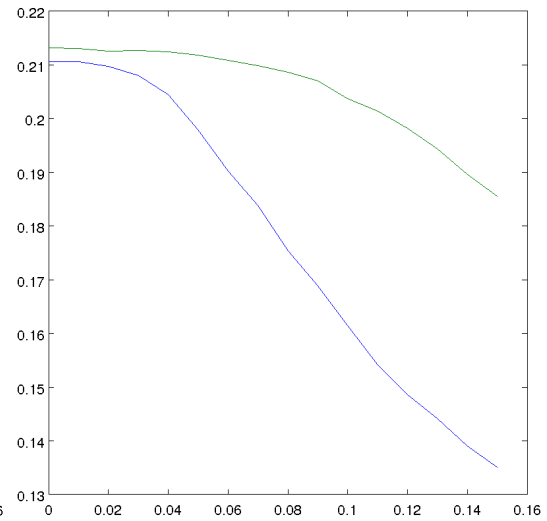
Otra consideración que se debe realizar respecto del ruido, es que se pueden obtener una enorme cantidad de matrices de ruido que tengan el mismo valor de σ . Cada una de estas muestras de ruido de igual σ , van a producir distintos valores óptimos para la función de performance. Para concluir objetivamente si un detector de contornos es mejor que otro, se debe realizar un gráfico que muestre los valores óptimos de la función de performance considerando muchas muestras de ruido por cada σ , tal como muestra la figura 2(b). Luego de lo anterior, se computa el promedio de los valores óptimos de la función de performance por cada σ , tal como muestra la figura 2(c). Por lo tanto teniendo las figuras 2(b) y 2(c) podemos concluir que, “en promedio un detector de contornos es mejor que otro”.



(a)



(b)



(c)

Figura 2: Evaluación de Performance.

5. Resultados

En esta sección se muestran los resultados del método propuesto, utilizando la metodología de evaluación objetiva presentada en la sección anterior.

Se muestran resultados con dos imágenes. La primera imagen corresponde a una imagen sintética simple en la cual se conoce el terreno de la verdad, es decir los contornos del objeto. La segunda imagen es una imagen real compleja, con objetos pequeños y delgados, y con zonas de bajo contraste. Para evaluar el método en la imagen real, no obstante no existe terreno de la verdad, se realizó una estimación de los contornos. Esta estimación se construyó filtrando la imagen real mediante un algoritmo que preserva los contornos, y luego se detectaron los bordes mediante una técnica iterativa que origina contornos de un pixel.

5.1. Resultados Imagen Sintética

En la figura 3 se muestra la imagen sintética simple contaminada con tres niveles de ruido. La figura 3(a) corresponde a una imagen sintética con dos niveles de grises, y la figura 3(b) muestra el terreno de la verdad. Las figuras 3(c), (d), y (e) muestran la imágenes contaminadas con ruido bajo ($\sigma = 0,05$), medio ($\sigma = 0,1$) y alto ($\sigma = 0,15$), respectivamente. Finalmente las figuras 3(f), (g), y (h) muestran los histogramas de los tres niveles de ruido. Como se observa en los histogramas, la distribución Gaussiana del ruido está centrada en 1, con desviación estándar muy pequeña para no degradar de forma importante la imagen.

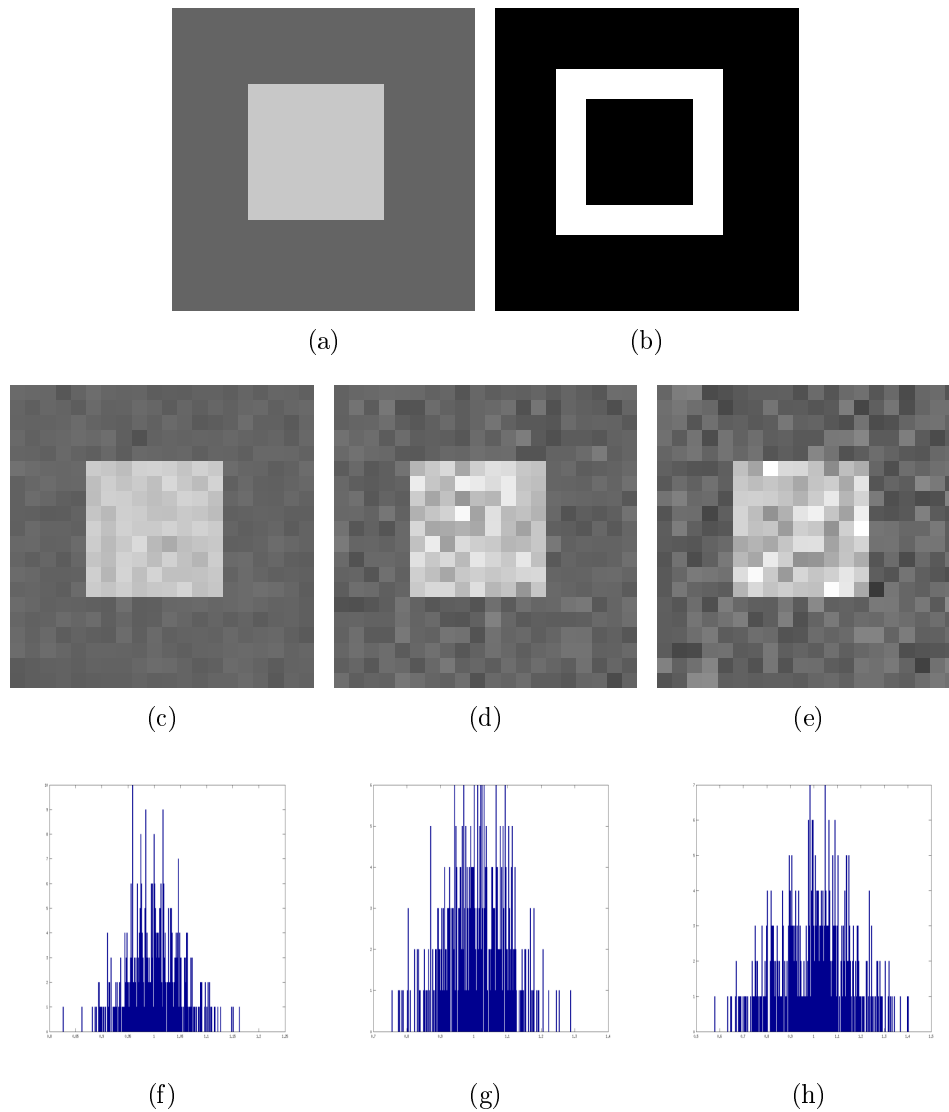


Figura 3: Imagen Sintética Simple: (a) Imagen Original, (b) Terreno de la verdad, (c) Bajo nivel de ruido ($\sigma = 0,05$), (d) Nivel medio de ruido ($\sigma = 0,1$), (e) Alto nivel de ruido ($\sigma = 0,15$), (f) Histograma de bajo nivel de ruido, (g) Histograma de nivel medio de ruido, y (h) Histograma de alto nivel de ruido.

En las figuras 4, 5, 6, 7 y 8 se muestran los contornos detectados por los Gradients Ant Colony (GAC) y Coefficient of Variation Ant Colony (CVAC) con los tres niveles de ruido. Donde la figura 4 corresponde a los contornos detectados por el GAC del gradiente A (GAC-A). En la figura 5 corresponde a los contornos detectados por el GAC del gradiente B (GAC-B). En la figura 6 corresponde a los contornos detectados por el

GAC del gradiente C (GAC-C). En la figura 7 corresponde a los contornos detectados por el GAC del gradiente D (GAC-D). Y en la figura 8 corresponde a los contornos detectados por el CVAC. Los casos con ruido bajo, medio y alto se muestran en la primera, segunda y tercera columna de cada figura respectivamente. Como es evidente, a medida que aumenta el ruido existen mayor cantidad de errores en la detección de contornos. No obstante lo anterior, con el método CVAC se obtienen notoriamente una menor cantidad de errores (lo que indica una mejor detección de contornos del CVAC).

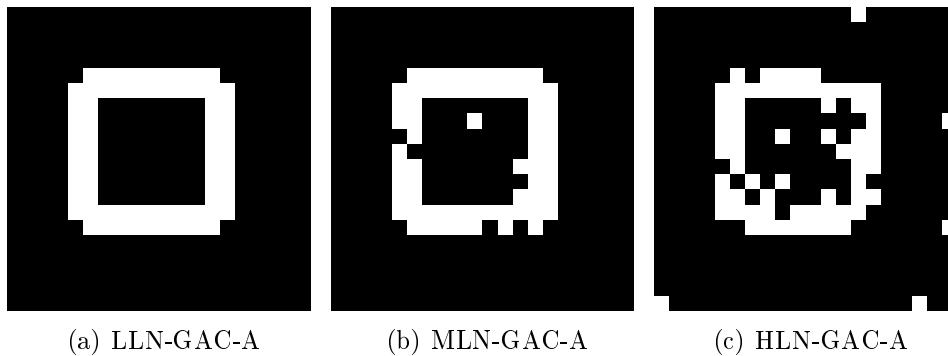


Figura 4: Contornos detectados en un imagen sintética: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente A (LLN-GAC-A), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente A (MLN-GAC-A), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente A (HLN-GAC-A).

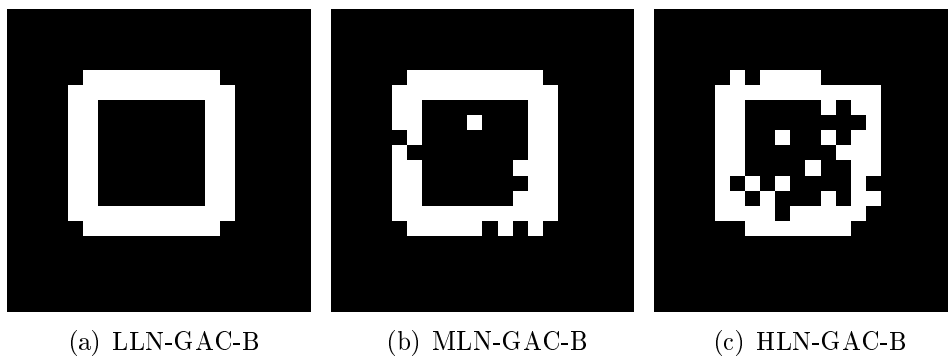


Figura 5: Contornos detectados en un imagen sintética: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente B (LLN-GAC-B), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente B (MLN-GAC-B), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente B (HLN-GAC-B).

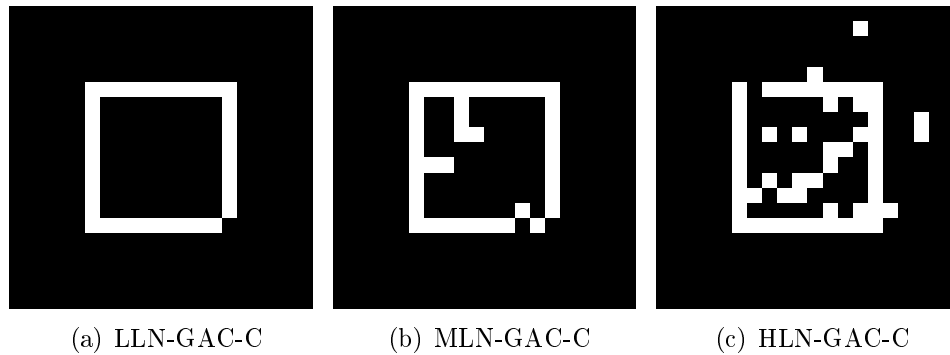


Figura 6: Contornos detectados en un imagen sintética: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente C (LLN-GAC-C), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente C (MLN-GAC-C), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente C (HLN-GAC-C).

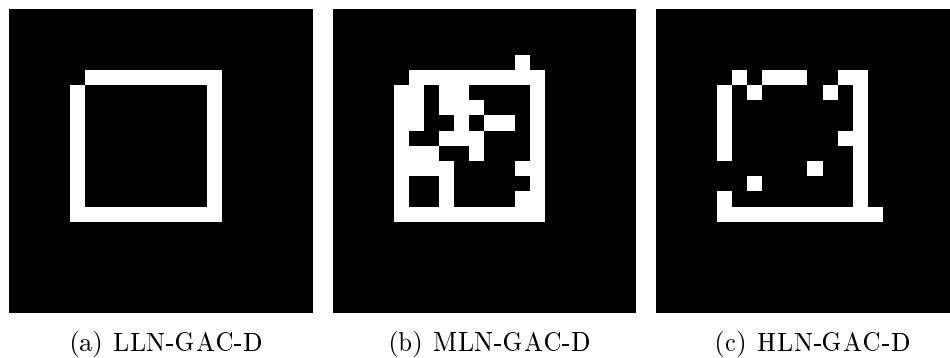


Figura 7: Contornos detectados en un imagen sintética: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente D (LLN-GAC-D), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente D (MLN-GAC-D), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente D (HLN-GAC-D).

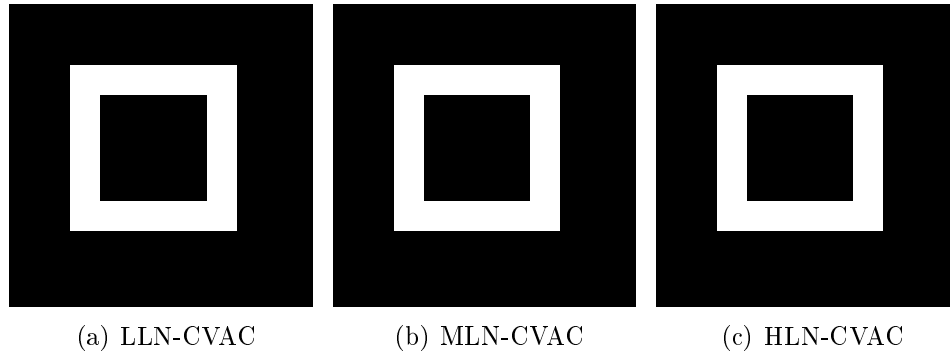
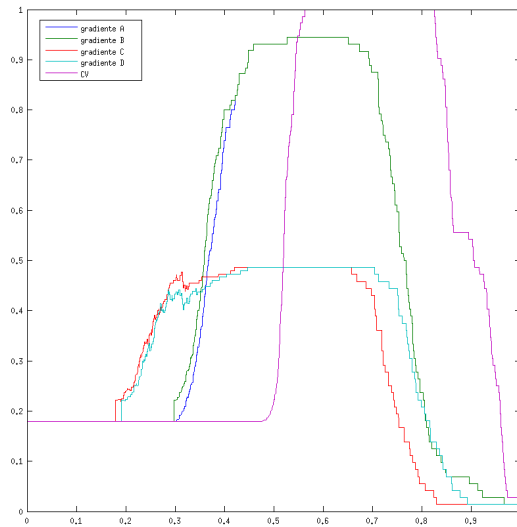
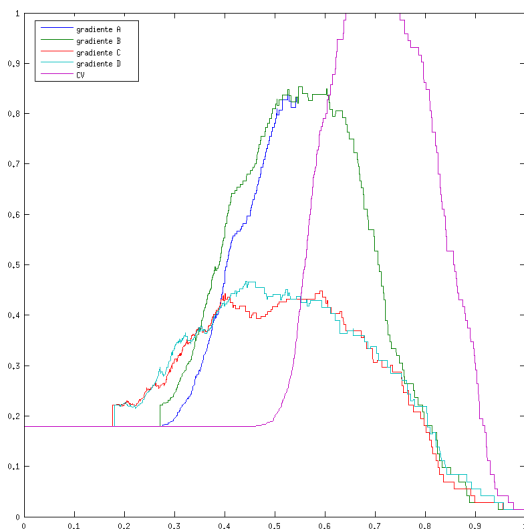


Figura 8: Contornos detectados en un imagen sintética: (a) Bajo nivel de ruido de Coeficiente de Variación Ant Colony (LLN-CVAC), (b) Nivel medio de ruido de Coeficiente de Variación Ant Colony (MLN-CVAC), y (c) Alto nivel de ruido de Coeficiente de Variación Ant Colony (HLN-CVAC).

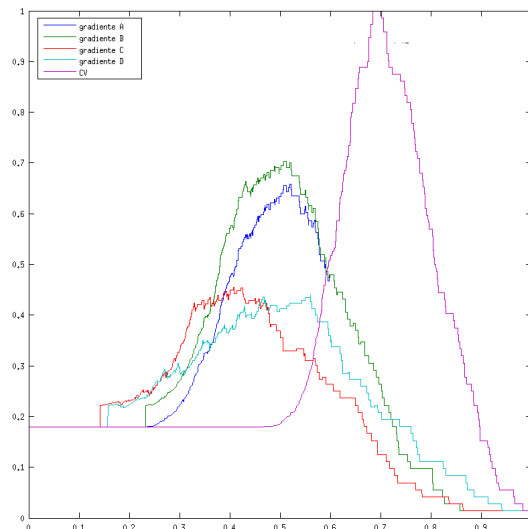
La figura 9 muestra los gráficos de la función de performance para distintos niveles del umbral considerando las tres situaciones de ruido. Las figuras 9(a), (b), y (c) corresponden a la función de performance para ruido bajo, medio, y alto respectivamente. En las figuras se observa que, no importando el nivel de ruido, la función de performance en el umbral óptimo es siempre mayor para el CVAC. Lo anterior corrobora lo que se observa visualmente en las figuras 4, 5, 6, 7 y 8, es decir, el CVAC detecta mejores contornos que el GAC-A, GAC-B, GAC-C y GAC-D. La tabla 1, 2 y 3 muestran los valores de la función de performance y el umbral óptimo para los 3 niveles de ruido, para ambos métodos de detección de contornos. Se observa que el valor de la función de performance baja a medida que el ruido aumenta, y que este valor es siempre mayor para el CVAC.



(a) LLN-PF



(b) MLN-PF



(c) HLN-PF

Figura 9: Função de performance GAC-A, GAC-B, GAC-C, GAC-D e CVAC para três níveis de ruído: (a) Baixo nível de ruído, (b) Nível médio de ruído, (c) Alto nível de ruído.

	Bajo Nivel de Ruido				
	GAC-A	GAC-B	GAC-C	GAC-D	CVAC
Performance óptimo	0.95	0.95	0.48	0.48	1
Umbral óptimo	0.53	0.53	0.42	0.46	0.56

Tabla 1: Valores óptimos de la función de performance para el bajo nivel de ruido.

	Nivel Medio de Ruido				
	GAC-A	GAC-B	GAC-C	GAC-D	CVAC
Performance óptimo	0.83	0.85	0.44	0.47	1
Umbral óptimo	0.52	0.54	0.58	0.44	0.64

Tabla 2: Valores óptimos de la función de performance para el nivel medio de ruido.

	Alto Nivel de Ruido				
	GAC-A	GAC-B	GAC-C	GAC-D	CVAC
Performance óptimo	0.66	0.7	0.45	0.44	1
Umbral óptimo	0.51	0.5	0.41	0.56	0.68

Tabla 3: Valores óptimos de la función de performance para el alto nivel de ruido.

La figura 10 presenta los valores óptimos de la función de performance para un conjunto continuo de niveles de ruido ($0 \leq \sigma \leq 0,15$). Por cada nivel de σ se computan 20 muestras de ruido tal como se muestra en la figura 10(a). La figura 10(b) muestra el gráfico del promedio de las 20 muestras de ruido. Desde el último gráfico se observa que la curva promedio del CVAC está siempre por sobre el GAC-A, GAC-B, GAC-C y GAC-D. Lo anterior significa que, en promedio el CVAC obtiene mejores contornos que el GAC-A, GAC-B, GAC-C y GAC-D independientemente del ruido. Es importante hacer notar que la varianza de las muestras para un mismo nivel de ruido es muy pequeña, lo que significa que por cada nivel de ruido, el CVAC va a tener siempre mejores resultados que el GAC-A, GAC-B, GAC-C y GAC-D.

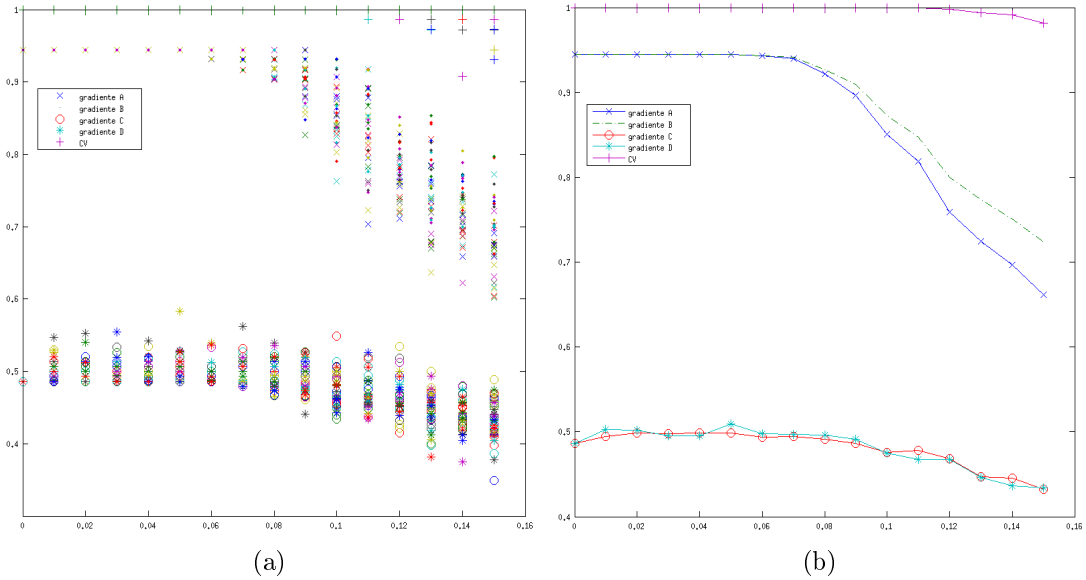


Figura 10: Función de performance óptimo para niveles continuos de ruido: (a) 20 muestras del mismo nivel de ruido y (b) función de performance promedio.

5.2. Resultados Imágenes Reales

Como forma de generalizar los resultados se presentan experimentaciones en una imagen real. Esta imagen tiene los problemas como por ejemplo: objetos pequeños y delgados, y zonas de bajo contraste. Los problemas anteriores adicionando el ruido multiplicativo presentan un desafío importante para el método propuesto.

Los resultados se presentan de forma análoga a la imagen sintética. La figura 11(a) presenta la imagen real, 11(b) el terreno de la verdad estimado, las figuras 11(c), (d), (e) las imágenes ruidosas (ruido bajo, ruido medio y ruido alto). Y las figuras 11(f), (g) y (h) se muestra el histograma de los tres niveles de ruido.

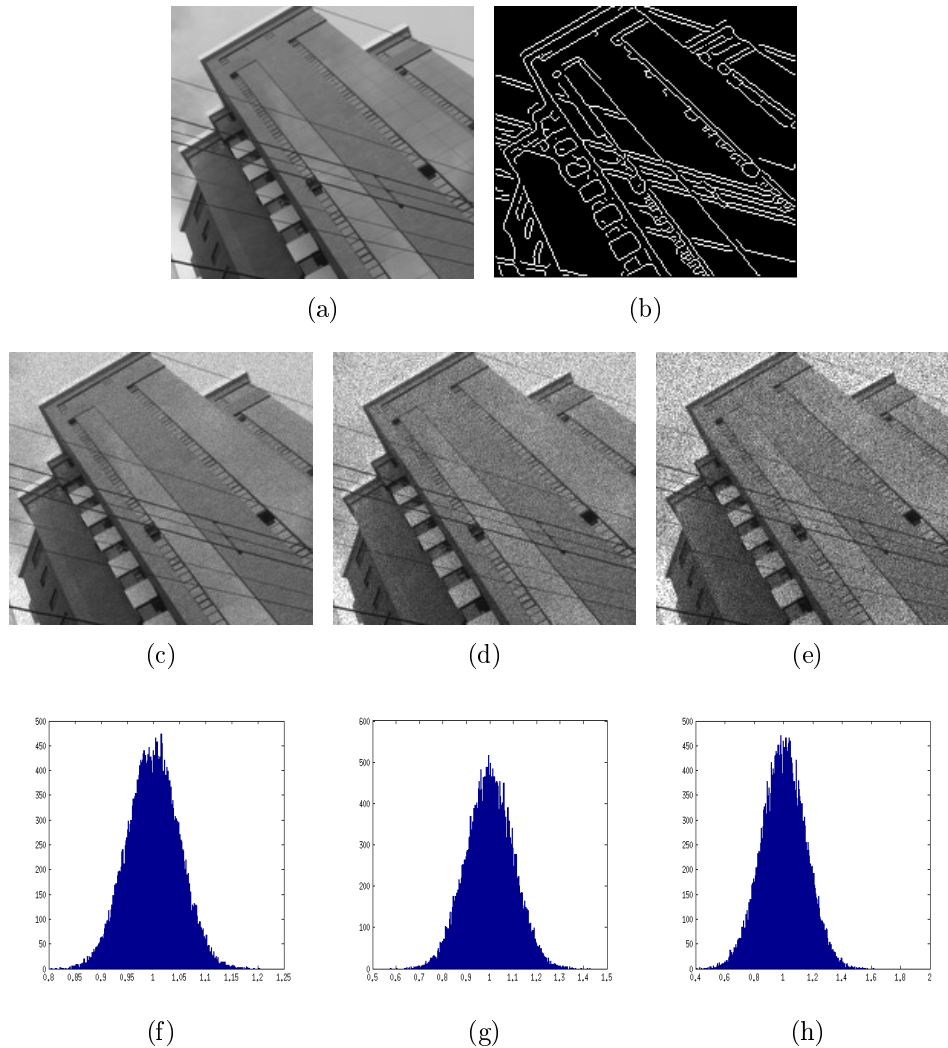


Figura 11: Imagen real: (a) Imagen Original, (b) Terreno de la verdad estimado, (c) Bajo nivel de ruido ($\sigma = 0,05$), (d) Nivel medio de ruido ($\sigma = 0,1$), (e) Nivel alto de ruido ($\sigma = 0,15$), (f) Histograma de bajo nivel de ruido, (g) Histograma de nivel medio de ruido, y (h) Histograma de alto nivel de ruido.

En las figuras 12, 13, 14, 15 y 16 se muestran los contornos detectados por los cinco métodos, para los tres casos de ruido. Donde la figura 12 muestra los contornos detectados por el GAC del gradiente A (GAC-A). La figura 13 muestra los contornos detectados por el GAC del gradiente B (GAC-B). La figura 14 muestra los contornos detectados por el GAC del gradiente C (GAC-C). La figura 15 muestra los contornos detectados por el GAC del gradiente D (GAC-D). Y la figura 16 muestra los contornos

detectados por el CVAC. Se observa que al igual que en los resultados en la imagen sintética, el método basado en el GAC-A, GAC-B, GAC-C y GAC-D genera una mayor cantidad de falsos positivos, que el método basado en CVAC. Esta cantidad de falsos positivos crece a medida que aumentamos el nivel de ruido. Por lo que podemos deducir que el algoritmo propuesto CVAC tiene mejores resultados que el algoritmo propuesto en la literatura.

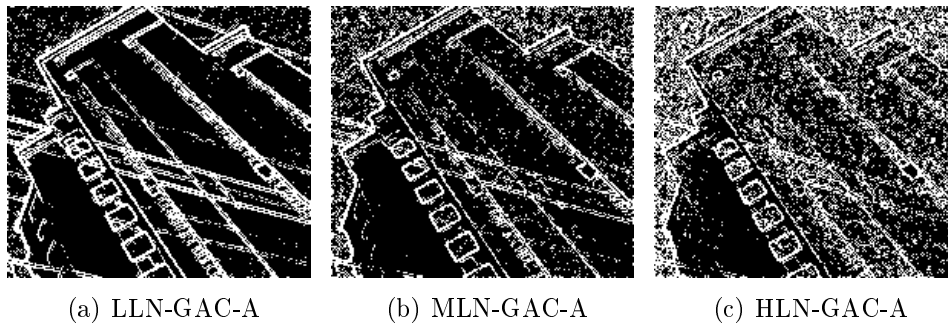


Figura 12: Contornos detectados en un imagen real: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente A (LLN-GAC-A), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente A (MLN-GAC-A), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente A (HLN-GAC-A)

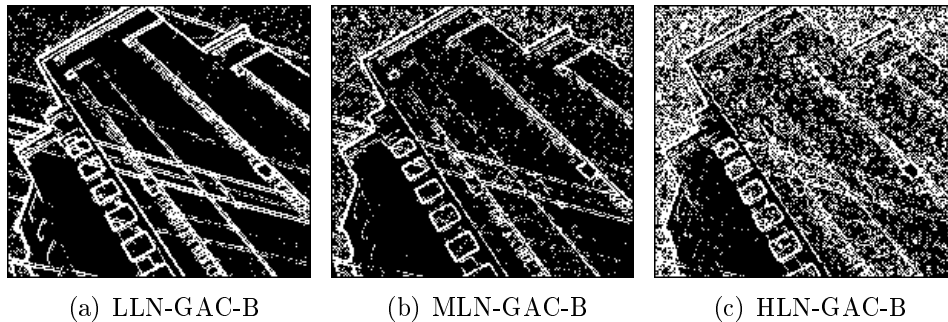


Figura 13: Contornos detectados en un imagen real: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente B (LLN-GAC-B), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente B (MLN-GAC-B), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente B (HLN-GAC-B)

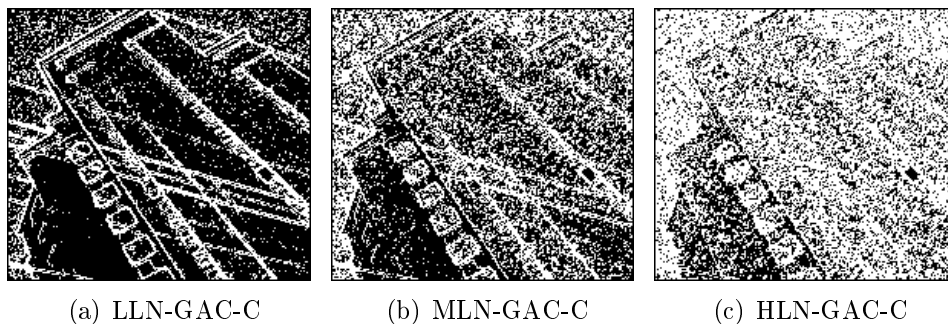


Figura 14: Contornos detectados en un imagen real: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente C (LLN-GAC-C), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente C (MLN-GAC-C), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente C (HLN-GAC-C)

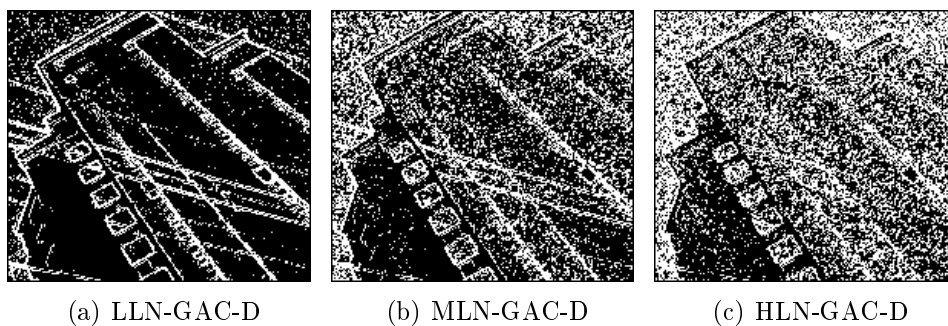


Figura 15: Contornos detectados en un imagen real: (a) Bajo nivel de ruido de Gradient Ant Colony del gradiente D (LLN-GAC-D), (b) Nivel medio de ruido de Gradient Ant Colony del gradiente D (MLN-GAC-D), (c) Alto nivel de ruido de Gradient Ant Colony del gradiente D (HLN-GAC-D)

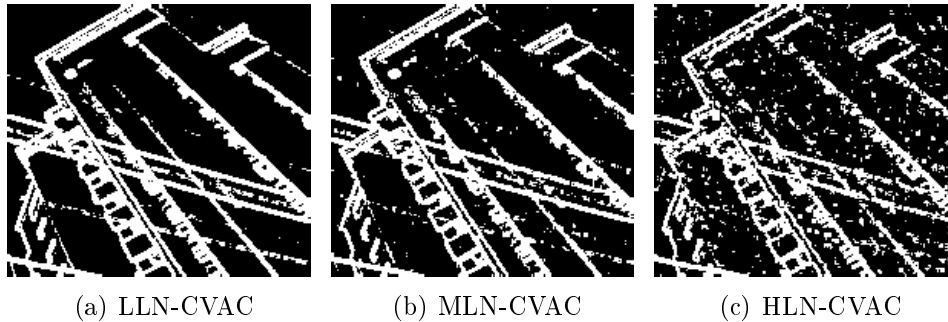


Figura 16: Contornos detectados en un imagen real: (a) Bajo nivel de ruido de Coeficiente de Variación Ant Colony (LLN-CVAC), (b) Nivel medio de ruido de Coeficiente de Variación Ant Colony (MLN-CVAC), y (c) Alto nivel de ruido de Coeficiente de Variación Ant Colony (HLN-CVAC).

La figura 17 se muestra la búsqueda de los umbrales óptimos para los tres casos de ruido. Los valores numéricos del umbral y función de performance óptimos de las figuras 17(a), (b), y (c) se presentan en la tabla 4, 5 y 6 respectivamente. Como se observa, para cada nivel de ruido, CVAC siempre obtiene el mayor valor para la función de performance.

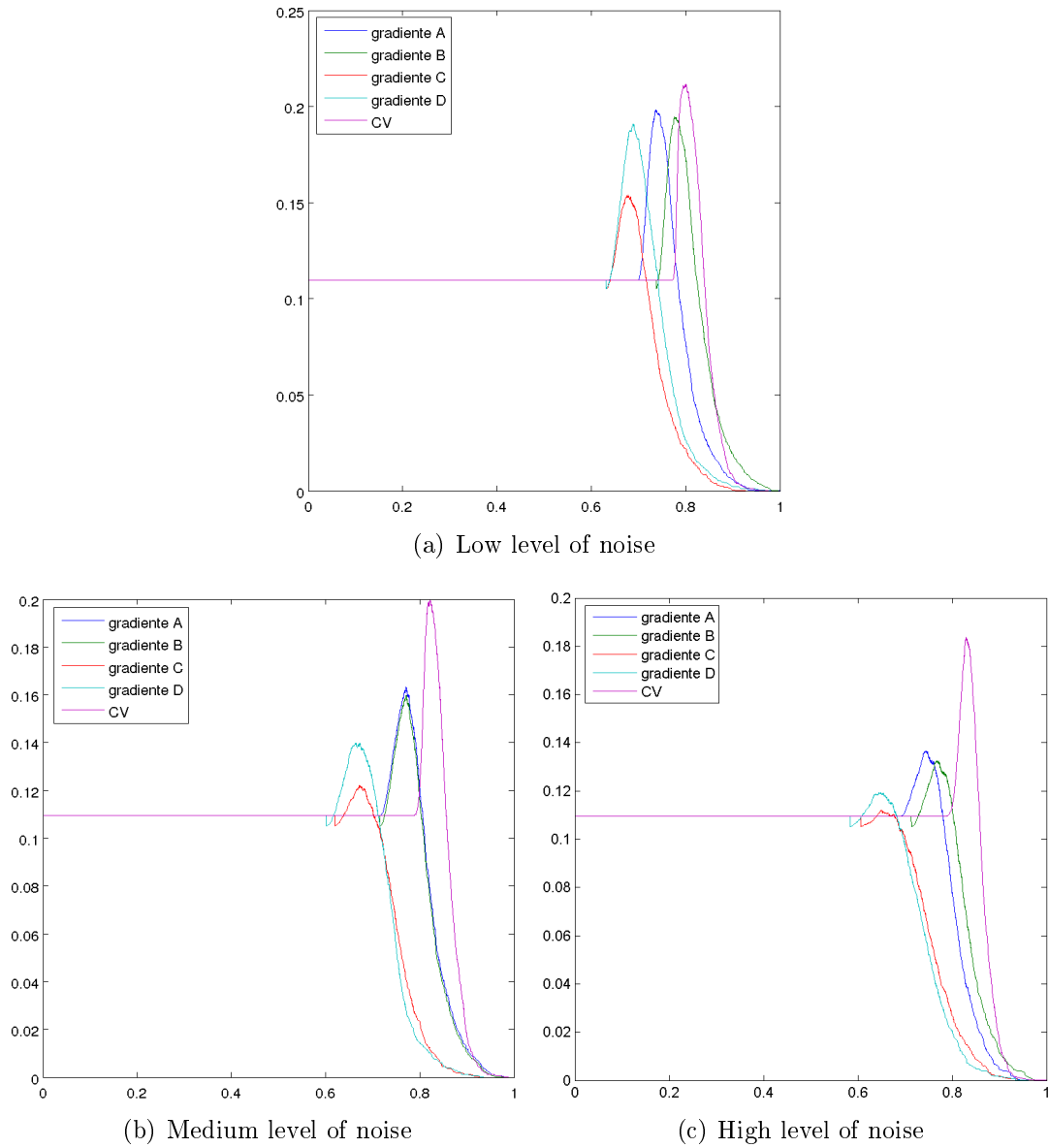


Figura 17: Función de performance GAC-A, GAC-B, GAC-C, GAC-D y CVAC para tres niveles de ruido de la imagen real: (a) Bajo nivel de ruido, (b) Nivel medio de ruido y (c) Alto nivel de ruido.

	Bajo Nivel de Ruido				
	GAC-A	GAC-B	GAC-C	GAC-D	CVAC
Performance óptimo	0.199	0.197	0.152	0.195	0.211
Umbral óptimo	0.73	0.77	0.67	0.68	0.80

Tabla 4: Valores óptimos de la función de performance para el bajo nivel de ruido.

	Nivel Medio de Ruido				
	GAC-A	GAC-B	GAC-C	GAC-D	CVAC
Performance óptimo	0.163	0.16	0.122	0.139	0.20
Umbral óptimo	0.77	0.77	0.67	0.66	0.82

Tabla 5: Valores óptimos de la función de performance para el nivel medio de ruido.

	Alto Nivel de Ruido				
	GAC-A	GAC-B	GAC-C	GAC-D	CVAC
Performance óptimo	0.138	0.142	0.112	0.119	0.183
Umbral óptimo	0.74	0.76	0.65	0.64	0.82

Tabla 6: Valores óptimos de la función de performance para el alto nivel de ruido.

En la figura 18 muestra el valor máximo de la función para los distintos niveles de ruido (de orden σ). La figura 18(a) presenta 20 muestras para el mismo nivel de ruido, y la 18(b) presenta un gráfico del promedio de las muestras de ruido. Se puede observar que la varianza por cada nivel de ruido es pequeña para ambos métodos. Además, a medida que aumenta el ruido la curva de performance asociada al GAC-A, GAC-B, GAC-C y GAC-D decae fuertemente, estando siempre la curva del CVAC por sobre de las curvas del GAC-A, GAC-B, GAC-C y GAC-D. Lo anterior indica que, en promedio el método basado en el CV tiene ventaja respecto del método basado en el Gradiente.

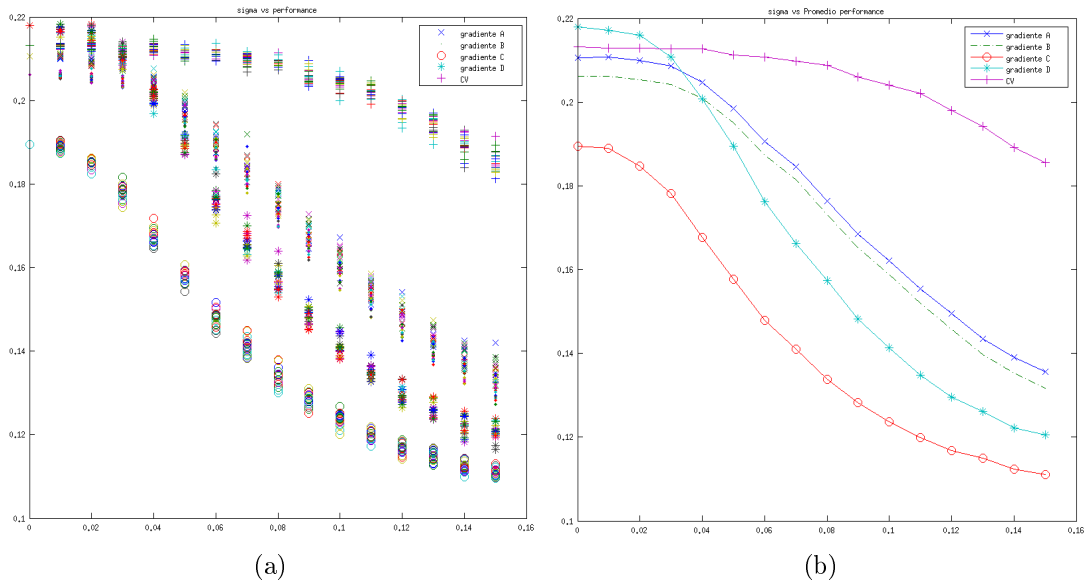


Figura 18: Función de performance óptimo para niveles continuos de ruido: (a) 20 muestras del mismo nivel de ruido y (b) función de performance promedio.

6. Conclusiones y Trabajos Futuros

En esta tesis se ha propuesto un método de detección de contornos en imágenes con ruido multiplicativo usando el algoritmo Ant Colony System.

En los trabajos de la literatura que han sido revisados, la detección de contornos basado en Ant Colony System considera la estimación de la Matriz de Feromona Global en base al gradiente de la imagen. Para abordar el ruido multiplicativo, el método propuesto estima la Matriz de Feromona Global basándose en el Coeficiente de Variación.

Las figuras de merito permiten comparar de forma objetiva la detección de contornos usando Gradient Ant Colony System y Coefficient of Variation Ant Colony System. El análisis de los resultados permite concluir que el método basado en el CV se adapta mejor a la presencia de ruido multiplicativo en las imágenes.

Se propone los siguientes trabajos futuros:

- Extender el algoritmo para imágenes con otros tipos de ruido.
- Proponer otro detector de contornos, que no utilice el Coeficiente de Variación como feromona de tipo global.
- Extender el algoritmo para imágenes en color con presencia de ruido.
- Como Ant Colony System adaptado para la detección de contornos es un algoritmo donde la solución entregada por cada hormiga es independiente. Entonces se puede paralelizar Ant Colony System para que las hormigas hagan una búsqueda de contornos en paralela, así podemos disminuir el tiempo de cómputo de Ant Colony System para la detección de contornos.

Referencias

- [1] A. Achim, A. Bezerianos, and P. Tsakalides. Novel bayesian multiscale method for speckle removal in medical ultrasound images. *IEEE Transactions on Medical Imaging - TMI*, 20(8):772–783, 2001.
- [2] A. V. Batera and C. Oppus. Image edge detection using ant colony optimization. *WSEAS Trans. Sig. Proc.*, 6(2):58–67, 2010.
- [3] K. Caballero, J. Barajas, O. Pujol, J. Mauri, and P. Radeva. Reconstructing ivus images for an accurate tissue classification. In *Proceedings of 2nd International Conference Computer Vision Theory and Applications (VISAPP)*, page 113–119, 2007.
- [4] M. Dorigo and L. M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.
- [5] S. A. Etemad and T. White. An ant-inspired algorithm for detection of image edge features. *Applied Soft Computing*, 11(8):4883–4893, 2011.
- [6] V. Frost, J. Stiles, K. Shanmugan, and J. Holtzman. A model for radar images and its application to adaptive digital filtering of multiplicative noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(2):157–166, 1982.
- [7] C. Gatta, O. Pujol, O. Rodriguez-Leor, J. Mauri, and P. Radeva. Fast rigid registration of vascular structures in ivus sequences. *IEEE Transactions on Information Technology in Biomedicine - TITB*, 13(6):1006–1011, 2009.
- [8] C. Grigorescu, N. Petkov, and M. Westenberg. Contour detection based on nonclassical receptive field inhibition. *IEEE Transactions on Image Processing*, 12(7):729–739, 2003.
- [9] K. Krissian, C.-F. Westin, R. Kikinis, and K. Vosburgh. Oriented speckle reducing anisotropic diffusion. *IEEE Transactions on Image Processing*, 16(5):1412–1424, 2007.
- [10] D. Kuan, A. S. an T. Strand, and P. Chavel. Adaptive restoration of images with speckle. *IEEE Transactions on Acoustics Speech and Signal Processing*, 35(3):373–383, 1987.
- [11] M. J. Ledesma-Carbayo, J. Kybic, M. Desco, A. Santos, S. Member, M. Sühling, S. Member, P. Hunziker, and M. Unser. Spatio-temporal nonrigid registration for ultrasound cardiac motion estimation. *IEEE Transactions on Medical Imaging*, 24(9):1113–1126, 2005.

-
- [12] J.-S. Lee. Digital image enhancement and noise filtering by use of local statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(2):165–168, 1980.
 - [13] A. Lopes, R. Touzi, and E. Nezry. Adaptive speckle filters and scene heterogeneity. *IEEE Transactions on Geoscience and Remote Sensing*, 28(6):992–1000, 1990.
 - [14] M. Mora, F. Córdova-Lepe, and R. Del-Valle. A non-newtonian gradient for contour detection in images with multiplicative noise. *Pattern Recognition Letters*, 33(10):1245–1256, 2012.
 - [15] H. Nezamabadi-pour, S. Saryazdi, and E. Rashedi. Edge detection using ant algorithms. *Soft Computing*, 10(7):623–628, 2006.
 - [16] M. Schickert. Progress in ultrasonic imaging of concrete. *Materials and Structures*, 38(9):807–815, 2005.
 - [17] J. Tian, W. Yu, and S. Xie. An ant colony optimization algorithm for image edge detection. In *In 2008 IEEE World Congress on Evolutionary Computation*, pages 751–756, 2008.
 - [18] R. Touzi. A review of speckle filtering in the context of estimation theory. *IEEE Transactions on Geoscience and Remote Sensing*, 40(11):2392–2404, 2002.