

Proyecto Contornos Activos SNAKES

Angel Manrique Pozos Flores
Francisco Fernando Zuñiga Alvarez
Tecnológico Nacional de México,
Blvd. Industrial, Mesa de Otay, 22430 Tijuana, B.C., México.

30 de mayo de 2016

Implementar el algoritmo de snake para el reconocimiento de contornos en nubes de partículas y posteriormente el reconocimiento del centroide de estas distribuciones. Implementar el algoritmo de snake para el reconocimiento y clasificación de tornillos.

1. Introducción

La segmentación de imágenes es la tarea de encontrar grupos de píxeles que van “*juntos*”, en estadística este problema es conocido como el análisis de cluster y a sido ampliamente estudiado mediante una gran variedad de algoritmos (Jain & Dubes, 1998 [3]; Kauffman & Rousseeuw, 1990 [7]; Jain & Duin, 2000 [4]; Jain & Topchy, 2004 [5]). En la visión por computadora, la segmentación de imágenes es uno de los problemas mas viejos y estudiados (Brice & Fennema, 1977; Riesenman & Arbib, 1977; Price & Reddy, 1978; Rosenfeld & Davis, 1979; Haralick & Shapiro, 1985) [8], [1].

Las técnicas clásicas de segmentación resultan útiles cuando las imágenes son sencillas, el objeto y el fondo son uniformes y la presencia de ruido y otros elementos no son muy importantes. Pero, en situaciones mas complejas es necesario realizar un procesamiento posterior a un nivel mas alto, para así, poder interpretar, reconocer y clasificar los objetos segmentados de bajo nivel y de esta forma dar conectividad o eliminar falsos positivos.

2. Contornos activos

Una solución reciente es utilizar técnicas de contornos que son mucho mas robustas pero requieren un tiempo de computo mucho mayor, estas técnicas las podemos agrupar con el nombre de *contornos activos*.

Estos contornos modelan las fronteras entre un objeto, el fondo y el resto de los objetos de la imagen, permiten extraer los contornos de los objetos de interés basándonos en modelos que utilizan información a priori de la forma de los objetos. Estas técnicas son mucho mas robustas frente a la presencia de ruido y demás elementos, permitiendo segmentar imágenes mucho mas complejas que las imágenes para las que eran aplicables los métodos de segmentación de bajo nivel.

La solución proporcionada por estas técnicas en general no requiere procesamiento anterior y son directamente interpretables, puesto que se basan en un modelo establecido *a priori*, si el modelo utilizado es el adecuado, la presencia de falsos positivos o falsos negativos sera mínima.

Los contornos activos se pueden clasificar en *snakes*, *patrones deformables* y *contornos dinámicos*, en este trabajo estudiaremos exclusivamente los primeros.

3. Snakes

La primera mención de las *snakes* fue hecha por sus inventores (Kass, Witkin & Terzopoulos, 1988 [6]) que la definen como una energía-minimizando (energy-minimizing) una curvatura spline de dos dimensiones que se mueve hacia las características de la imagen como los bordes muy gruesos.

Los *snakes* son mecanismos para dar cierto grado de conocimiento a priori a la interpretación de la imagen de bajo nivel, donde, en lugar de esperar que propiedades deseables de los contornos como lo son la continuidad y la suavidad provengan de los datos de la imagen, estas propiedades son impuestas desde el principio.

Donde la curva que es continua y flexible busca ajustarse a los datos de interés de la imagen o sea que este método busca el contorno de una imagen, a diferencia de otras técnicas que buscan contornos prominentes en la imagen, este modelo es activo lo cual quiere decir que siempre esta minimizando su funcional energético y por ello puede considerarse que tiene un comportamiento dinámico, se le dice *snake* porque este se va deslizando conforme va siendo minimizada su energía.

Por ello un *snake* no esta pensado en un principio, para resolver problemas de búsqueda de contornos prominentes de la imagen, si no para refinar la solución propuesta por otra serie de mecanismos, es decir, si partimos de un contorno dibujado manualmente o obtenido mediante alguna función, el contorno va a evolucionar hasta alcanzar el mínimo local mas cercano o el contorno buscado.

Los *snakes* pueden considerarse como un ejemplo de un caso mas general de ajuste de modelos deformables a la imagen mediante un proceso de minimización energética. En resumen, una *snake* se puede definir como una curva *spline* minimizadora de energía, guiada por fuerzas restrictivas externas e influenciada por fuerzas de la imagen, que tiende a localizarse en características de esta como líneas y bordes, es por tanto un contorno activo que evoluciona de forma dinámica hacia los contornos relevantes de la imagen. El snake también posee una serie de fuerzas internas que sirven para imponer restricciones de suavidad, es decir, una regularización de la solución [2].

Donde las fuerzas de la imagen empujan a la *snake* hacia características de la imagen como líneas, bordes y contornos subjetivos, mientras que las fuerzas restrictivas externas añaden información de alto nivel para hacer que la *snake* se vaya hacia el mínimo local deseado.

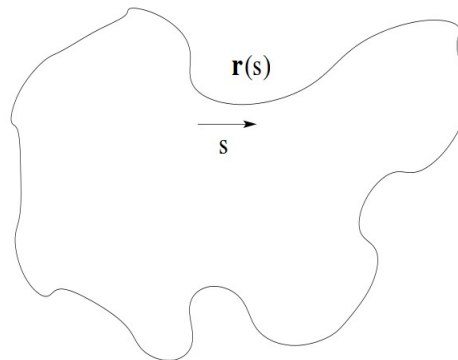


Figura 1: Ejemplo clásico de un contorno.

El *snake* se representa como una curva paramétrica mediante $\mathbf{r}(s) = (x(s), y(s))$, con $s \in (0, 1)$. Como se observa en la figura 1 si variamos s entre 0 y 1 vamos a ir recorriendo el contorno, donde su funcional de energía lo podemos representar como.

$$E_{snake}^*(\mathbf{r}) = \int_0^1 E_{int}(\mathbf{r}(s))ds + \int_0^1 E_{img}(\mathbf{r}(s))ds + \int_0^1 E_{res}(\mathbf{r}(s))ds \quad (1)$$

Donde E_{int} representa la energía interna del snake, E_{img} las fuerzas de la imagen y E_{res} las fuerzas restrictivas externas.

La energía interna del snake se puede escribir como.

$$E_{int}(\mathbf{r}(s)) = \frac{\vartheta_1(s) \|\mathbf{r}'(s)\|^2 + \vartheta_2(s) \|\mathbf{r}''(s)\|^2}{2} \quad (2)$$

Donde la energía interna esta formada por un termino de primer orden controlado por $\vartheta_1(s)$ y otro segundo orden controlado por $\vartheta_2(s)$. Donde el termino de primer orden hace que el *snake* se comporte como una membrana, mientras que el de segundo orden hace que se comporte como una lamina plana delgada. En otras palabras, la primera derivada significa que el contorno evoluciona lentamente según varia el parámetro s , mientras que la segunda derivada significa que el contorno se aproxima a una linea recta.

Ajustando los parámetros $\vartheta_1(s)$ y $\vartheta_2(s)$ se puede controlar la importancia relativa de estos dos términos a lo largo del contorno. Si fijamos $\vartheta_2(s)$ igualándolo a 0 para un cierto valor de s , se estará permitiendo que el contorno tenga una discontinuidad de primer orden y ,por ello, pueda presentar un vértice. Estos dos términos se pueden considerar como los términos regularizadores de la *snake*.

Las fuerzas restrictivas externas que típicamente han sido introducidas por el usuario son dos:

- Las llamadas tipo *Muelle*
- Las llamadas tipo *Volcan*

Tenemos entonces que:

$$E_{res} = E_{mue} + E_{vol} \quad (3)$$

Donde E_{mue} es del primer tipo y E_{vol} del segundo. La fuerza tipo muelle es equivalente a colocar un muelle desde un punto cualquiera de la *snake* hasta cualquier punto de la imagen o otro punto de la *snake*. Esta fuerza esta controlada por el parámetro de restauración de muelle, donde, si los extremos del muelle son \mathbf{p}_1 y \mathbf{p}_2 la fuerza viene dada por:

$$E_{mue} = -k(\mathbf{p}_1 - \mathbf{p}_2)^2 \quad (4)$$

El termino tipo *volcán* crea una fuerza de repulsión del tipo $1/r^2$ que esta truncada cerca de $r = 0$ siendo r la distancia hasta el centro del volcán, este se puede situar en cualquier punto de la imagen y repelerá al *snake* de su centro. Para que los *snake* sean de utilidad es necesario definir funcionales energéticos que atraigan al *snake* hacia las características relevantes de la imagen, que puede considerarse que son, líneas, bordes y terminaciones. La energía de la imagen se puede expresar como una combinación ponderada de estos tres términos.

$$E_{img} = E_{lin}\vartheta_{lin} + E_{bor}\vartheta_{bor} + E_{ter}\vartheta_{ter}. \quad (5)$$

siendo ϑ_{lin} , ϑ_{bor} y ϑ_{ter} los pesos correspondientes a la energía de linea E_{lin} , con borde E_{bor} y de terminación E_{ter} . Según se ajusten estos términos, se puede modificar fácilmente el comportamiento de la *snake* con respecto a la imagen. El funcional energético de imagen mas sencillo es la intensidad de la imagen por si misma, en caso de que la imagen tenga líneas se vera atraído a ellas.

$$E_{lin}(\mathbf{r}(s)) = I(x(s), y(s)) \quad (6)$$

Entonces dependiendo del signo ϑ_{lin} , el *snake* se vera atraído hacia las líneas claras o oscuras. El *snake* sujeto al resto de las restricciones se vera atraído a los contornos cercanos a regiones claras u oscuras de la imagen. Para poder encontrar los bordes de la imagen utilizamos un funcional.

$$E_{bor}(\mathbf{r}(s)) = - \| \nabla I(x(s), y(s)) \|^2 \quad (7)$$

De esta forma el *snake* se vera atraído a las zonas con valor elevado del gradiente. Podemos evitar que el *snake* se quede atrapado en mínimos locales dados por estructuras pequeñas como por ejemplo el ruido. Para ello, elegida una escala σ , se desenfocara la imagen para eliminar estructuras a menor escala antes de determinar el gradiente de la imagen. Este funcional es:

$$E_{bor}(\mathbf{r}(s)) = - |(\nabla^2 h_\sigma(x(s), y(s))) * I(x(s), y(s))|^2 \quad (8)$$

Donde h_σ es una función gaussiana de desviación estándar σ , los mínimos locales de E_{bor} corresponden con los cruces por cero de $(\nabla^2 h_\sigma) * I$. Añadiendo este termino y dada una escala σ , el *snake* se vera atraído a los cruces por cero, pero sujeto al resto de las restricciones, si la escala es muy grande, el *snake* se vera atraído a los bordes de una forma muy aproximada, pero lo podrá hacer desde posiciones bastante alejadas, a escalas pequeñas el *snake* se vera atraído a las micro estructuras o ruido cercanos, pero no podrá alcanzar los bordes reales de las regiones a escalas mayores. Por ello, comúnmente se empieza a escalas grandes

para obtener una primera solución aproximada y después se disminuye la escala para refinar la solución.

Para definir el funcional de terminación, se parte de una version ligeramente suavizada de la imagen $C(x(s), y(s)) = h_\sigma(x(s), y(s)) * I(x(s), y(s))$ y se define a partir del grado de curvatura de las curvas de nivel de la imagen suavizada, el angulo del gradiente viene dado por:

$$\alpha(s) = \tan^{-1} \left(\frac{\frac{\partial}{\partial x} C(x(s), y(s))}{\frac{\partial}{\partial y} C(x(s), y(s))} \right) \quad (9)$$

Los vectores unitarios en la dirección paralela \mathbf{n}_1 y perpendicular \mathbf{n}_2 a la dirección del gradiente vienen dados por:

$$\mathbf{n}_1(s) = (\cos \alpha(s), \sin \alpha(s)) \quad (10)$$

$$\mathbf{n}_2(s) = (-\sin \alpha(s), \cos \alpha(s)) \quad (11)$$

A partir de estos parámetros, el termino de energía de terminación viene dado por:

$$E_{ter}(\mathbf{r}(s)) = \mathbf{D}_{n_2}(s) C(x(s), y(s)) = \frac{\mathbf{D}_{n_2}^2(s) C(x(s), y(s))}{\mathbf{D}_{n_1}(s) C(x(s), y(s))} = \frac{\frac{\partial^2 C}{\partial y^2} \left(\frac{\partial C}{\partial x} \right)^2 - 2 \frac{\partial C}{\partial x} \frac{\partial^2 C}{\partial x \partial y} \frac{\partial C}{\partial y} + \frac{\partial^2 C}{\partial x^2} \left(\frac{\partial C}{\partial y} \right)^2}{\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} \right)^{3/2}} \quad (12)$$

Siendo \mathbf{D}_v la primera derivada direccional según el vector unitario \mathbf{v} y \mathbf{D}_v^2 la segunda derivada direccional. Se puede comprobar experimentalmente la capacidad de la *snake* para segmentar contornos subjetivos, ademas, el *snake* tiene la propiedad de histeresis. Dichas propiedades no existían en los algoritmos clásicos de segmentación y por lo tanto, esto es algo característicos de los contornos activos.

Una vez que el *snake* encuentra un mínimo local de la energía, se queda enganchado a el, si el objeto en cuestión comienza a moverse lentamente, el *snake* simplemente seguirá el movimiento de ese mínimo local. Si el movimiento es demasiado rápido puede dar lugar a que el *snake* pierda ese mínimo local y se quede atrapado en otro.

3.1. Snake GVF

El método de *snake* del flujo del vector gradiente o GVF (C. Xu & L. Prince, 1997 [9]) busca incrementar el rango de captura y dirigir al *snake* dentro de las

concavidades donde este rango de captura se incrementa mediante el proceso de difusión que no desenfoca los bordes, de esta manera la detección de los mismos no pierde precisión, este proceso de difusión permite que aparezca la componente de la fuerza que apunta a las concavidades, además que la inicialización puede ser dentro, fuera o a través del objeto sin necesidad de una información *a priori* que le indique en qué dirección debe dirigirse, la nueva fuerza externa se obtiene a partir de una formulación general.

Esta fuerza externa es la principal diferencia entre el método snake tradicional el cual usa fuerzas potenciales, es decir, fuerzas que pueden ser escritas como el negativo del gradiente de una función potencial, mientras que el método de *snake* GVF utiliza una fuerza externa más general que no puede ser escrita como el negativo del gradiente, por esta razón es especificado directamente a partir de una condición de balance de fuerzas. Esta fuerza es calculada como la difusión de los vectores gradientes del mapa de bordes binario o en escalas de grises, derivado de la imagen, dicha fuerza además de incrementar la magnitud del campo externo, cambia su dirección permitiendo tener componentes que apunten hacia dentro de las concavidades y no solo a los lados como sucedía con el *snake* tradicional.

Esas componentes se obtienen al generarse una interacción de vectores en la vecindad de los bordes y se deben a la intensidad del campo. En la figura 2 se muestra el campo dentro de la concavidad.

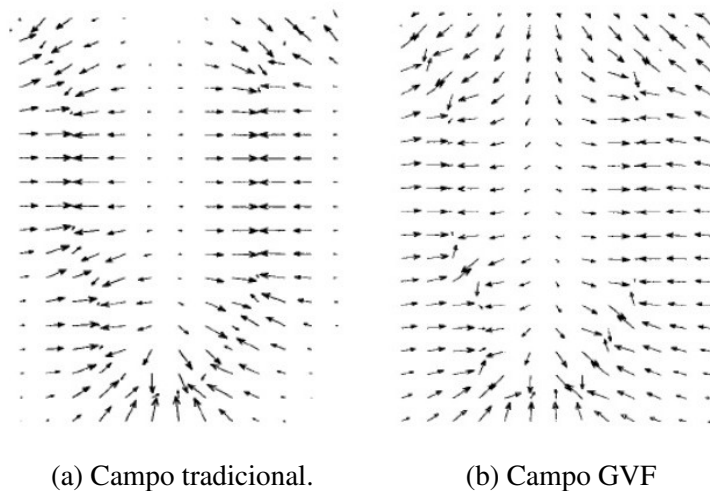


Figura 2: Campos de fuerzas externas, donde se observa como el campo GVF apunta hacia dentro de la concavidad.

Entonces la fuerza externa la podemos definir como:

$$F_{externa} = \mathbf{V}_{GVF}(x, y) \quad (13)$$

Donde.

$$V_{GVF}(x, y) = [u(x, y), v(x, y)] \quad (14)$$

Siendo entonces

- $V_{GVF}(x, y)$ el campo de fuerzas generales GVF.
- $u(x, y)$ y $v(x, y)$ las componentes del nuevo campo vectorial.

Este campo \mathbf{V}_{GVF} es un campo vectorial denso estático, es decir, se obtiene de los datos de la imagen y no cambia con el tiempo ni con la posición de *snake*, para establecer una fuerza general, se utiliza la teoría de los campos vectoriales, la cual indica que un vector es definido por su divergencia y su rotacional. El teorema de Helmholtz establece que un vector puede expresarse como la suma de dos vectores, uno irrotacional y otro solenoidal. Las fuerzas externas del *snake* únicamente cumplen con el primero, por lo tanto al permitir que además los vectores sean solenoidales, se tiene un cambio en el comportamiento del entorno.

La ecuación dinámica que permite obtener la posición final del contorno activo, se obtiene al sustituir la fuerza externa por el campo GVF, este campo debe minimizar el siguiente funcional de energía.

$$\varepsilon_{GVF} = \int \int \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |\mathbf{v}_{GVF} - \nabla f|^2 dx dy \quad (15)$$

Donde:

- f es el mapa de bordes.
- u_x, u_y, v_x, v_y son las derivadas parciales de las componentes del \mathbf{V}_{GVF}

El parámetro μ se encarga de compensar al primer y segundo termino de la integral, su magnitud estará en función de la cantidad de ruido en la imagen, entre mas ruidosa sea la imagen este parámetro va a incrementarse y viceversa.

Para el análisis del funcionamiento es importante tener en cuenta las siguientes propiedades del mapa de bordes.

- El gradiente de un mapa de bordes ∇f esta formado por un campo de vectores que apuntan hacia los bordes y además son perpendiculares a los mismos.

- Estos vectores generalmente tienen magnitudes grandes solo en los alrededores inmediatos a los bordes.
- En las regiones homogéneas, donde la intensidad de grises es casi constante, el gradiente del mapa de bordes es prácticamente nulo $\nabla f \cong 0$

La segunda y tercera propiedades son responsables del problema de inicialización de la *snake* tradicional, sin embargo, buscamos conservar la primera propiedad y mantener magnitudes grandes, no solamente en la vecindad de los bordes, por lo tanto se requiere un campo que varíe paulatinamente a partir de los bordes y que se extienda a las regiones homogéneas de la imagen, en el GVF el mapa de bordes se define como:

$$f(x, y) = -E_{ext}^{(i)}(x, y) \quad (16)$$

A partir de la ecuación 15 se pueden hacer dos observaciones importantes que describen el comportamiento del campo GVF.

- Si $|\nabla f| \cong 0$, la suma de los cuadrados de las derivadas parciales del campo vectorial dominan el comportamiento de energía del GVF ya que:

$$|\nabla f|^2 |V_{GVF} - \nabla f|^2 \approx 0 \quad (17)$$

Esto provoca la aparición de un campo de variación lenta, por lo que las regiones cuasi-homogéneas que están mas allá de los bordes tendrán asignado un campo vectorial que empujara a la *snake* GVF hacia los bordes, esta particularidad hace que el contorno activo tenga una cierta insensibilidad a la inicialización.

- Si $|\nabla f|$ es grande, el elemento dominante es el segundo termino de la integral y la única posibilidad de lograr la minimización del funcional de energía es:

$$|V_{GVF} - \nabla f|^2 = 0 \quad (18)$$

La cual se satisface si

$$V_{GVF} = \nabla f \quad (19)$$

Que nos indica que las regiones inmediatas a los bordes, el campo GVF conservara todas las características del gradiente de un mapa de bordes.

Con esto obtendremos el comportamiento deseado, mantener la magnitud y la dirección del gradiente del mapa de bordes cuando este es grande, y en caso contrario, se obliga al campo V_{GVF} a tener variaciones lentas del gradiente del mapa de bordes, con esto se satisface el inconveniente del rango de captura pequeño.

3.2. Ventajas y desventajas de GVF

Las principales ventajas que presenta el método *snake* GVF respecto al método tradicional de *snake* son:

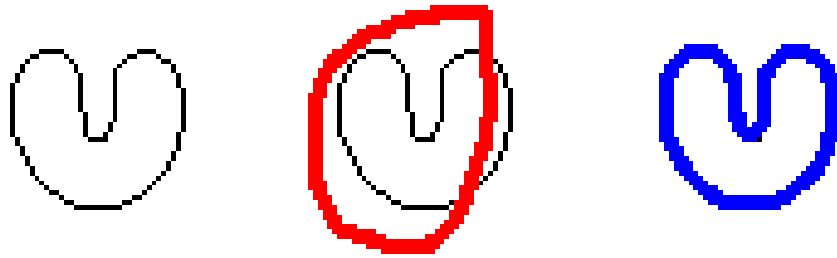
- Independencia respecto a la inicialización del contorno, ya que se puede converger a una solución correcta.
- Capacidad de hacer la detección de bordes cóncavos.
- No necesita utilizar información *a priori* para determinar si contraerse o expandirse.
- Presenta una mayor robustez frente al ruido.
- Presenta un rango de captura amplio que no desenfoca los bordes.

Las principales desventajas del método GVF.

- Ineficiencia en la preservación de bordes débiles y detecciones de concavidades profundas y estrechas, esto es resultado de tratar a las dos componentes del campo GVF como componentes desacopladas durante el proceso de difusión.
- Se requiere de una gran cantidad de operaciones para encontrar su solución, por lo cual es método un poco lento y pesado computacionalmente.

Ya que el código esta un poco extenso se adjunta en el archivo “*snake.cpp*” donde solo es necesario cargar una imagen en cualquier formato, de preferencia una imagen de baja resolución.

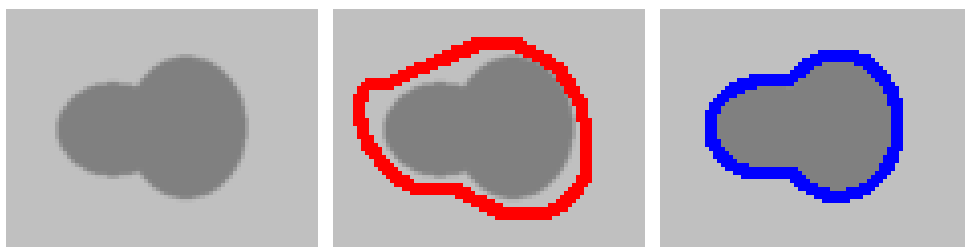
Se realizaron algunas pruebas con contornos para observar el funcionamiento del algoritmo las cuales se muestran en la figura 3.



(a) Imagen original para las pruebas. (b) Imagen con contorno dibujado. (c) Imagen obtenida de GVFsnake.

Figura 3: Imágenes del proceso de aplicación del algoritmo de GVF *snake*, sobre una imagen con concavidad.

Como observamos el GVF *snake* se optimiza buscando la concavidad en la imagen, comparándolo con el método de *snake* tradicional esta concavidad hubiese sido descartada, donde no es necesario que el contorno inicial este dentro o fuera de la imagen para que este converga hacia el contorno de la figura, hice otras pruebas con otra imagen estas se muestran en la figura 4.



(a) Imagen original para las pruebas. (b) Imagen con contorno dibujado. (c) Imagen obtenida de GVFsnake.

Figura 4: Imágenes del proceso de aplicación del algoritmo de GVF *snake* sobre una imagen cerrada.

4. Distribucion de nube de particulas

El propósito es aplicarlo a distribuciones de partículas con el fin de hacer la búsqueda del centroide, esta distribución de partículas representa la luz capturada por un telescopio espacial donde la información se obtiene por medio de un dispositivo electrónico llamado placa micro-canal o MCP (Micro Channel Plate). La imagen utilizada se muestra en la figura 5.

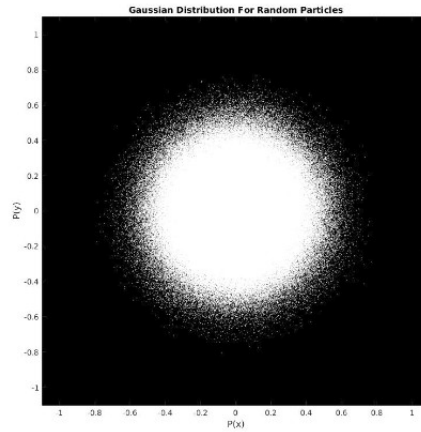
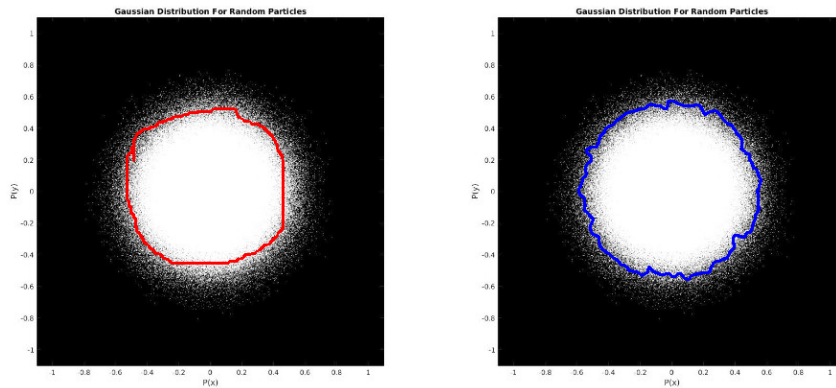


Figura 5: Distribución de partículas.

Aplicando GVF snake obtuvimos.

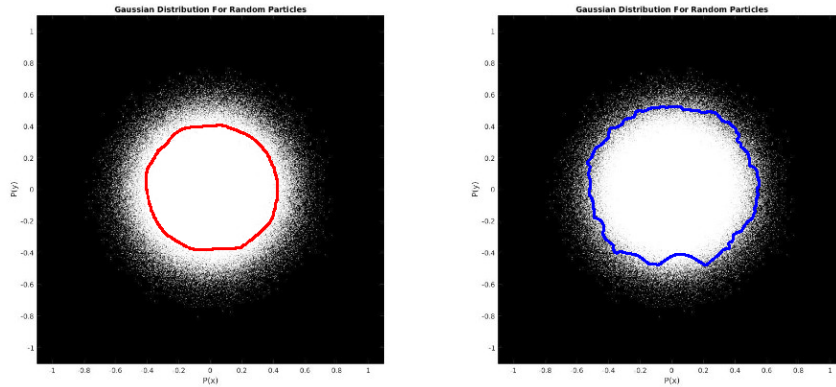


(a) Imagen con contorno dibujado.

(b) Imagen obtenida de GVF snake.

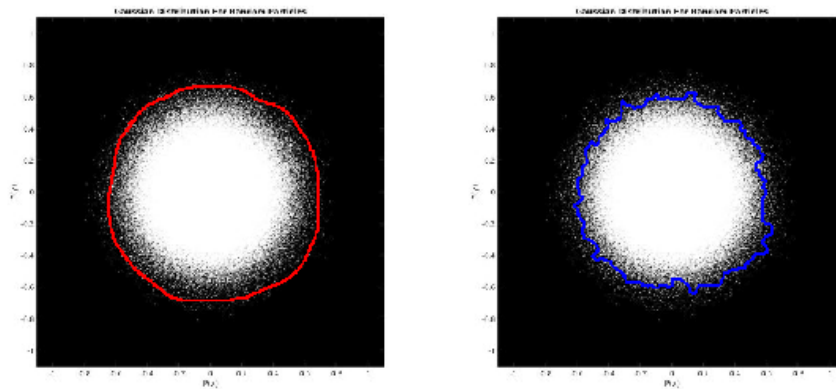
Figura 6: Aplicación del algoritmo de GVF *snake* a una nube de partículas.

Tome mas contornos para los casos dentro y fuera de la imagen para observar convergencia, donde obtuve los siguientes resultados.



(a) Imagen con contorno dibujado. (b) Imagen obtenida de GVF snake.

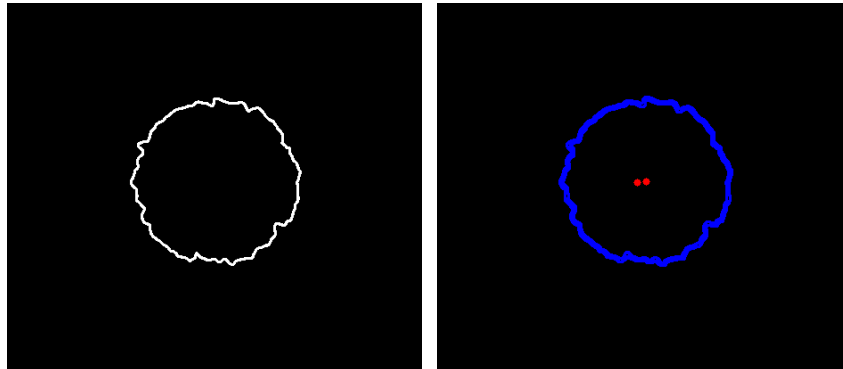
Figura 7: Aplicación del algoritmo de GVF *snake* a una nube de partículas, iniciando el contorno dentro de la imagen.



(a) Imagen con contorno dibujado. (b) Imagen obtenida de GVF snake.

Figura 8: Aplicación del algoritmo de GVF *snake* a una nube de partículas, iniciando el contorno fuera de la imagen.

Una vez obtenido el contorno de la imagen, hacemos la búsqueda de su centroide mediante el programa “*centroid.cpp*” que hace la búsqueda del centro de masa del contorno obtenido (tome el primer contorno que fue dibujado en la periferia de la imagen), en las imágenes de la figura 7 se muestran los resultados obtenidos.



(a) Contorno obtenido de GVFSna- (b) Centroides obtenidos de la distribución de partículas.
 ke para la búsqueda del centroide.

Figura 9: Aplicación del algoritmo de centroide a la nube de partículas.

Observamos que se obtienen dos centroides de nuestro contorno, esto se debe a que al momento de calcular el contorno obtenido lo hace por la línea de adentro del contorno y por la línea de afuera que del contorno.

También agregue unas pruebas con una imagen real para observar que tan robusto es el algoritmo, para ello utilice la imagen de la figura 8.



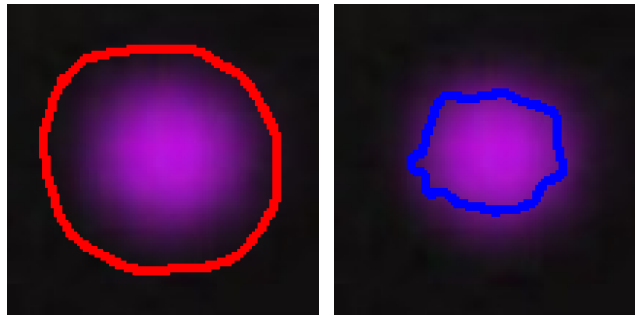
Figura 10: Colisión de galaxias, imagen en rayos x y uv obtenida por el telescopio espacial GALEX.

Tome una de estas distribuciones de partículas para hacer pruebas, la cual se muestra en la imagen de la figura 9.



Figura 11: Distribución de partículas.

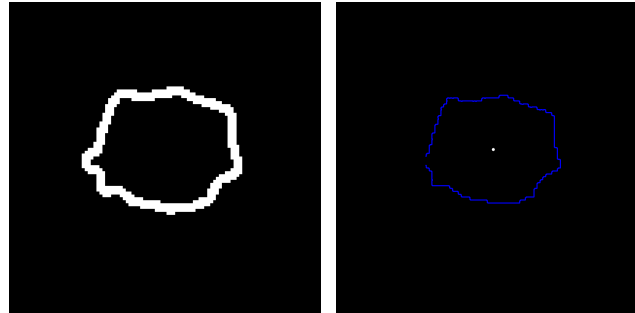
Aplicando GVF *snake*.



(a) Imagen con contorno dibujado. (b) Imagen obtenida de GVF *snake*.

Figura 12: Aplicación del algoritmo de GVF *snake* a una distribución real de partículas.

Aplicando el algoritmo de centroide obtuvimos.



(a) Contorno obtenido de GVF *snake* utilizado para la búsqueda del centroide. (b) Centroide obtenido de la distribución de partículas.

Figura 13: Aplicación del algoritmo de centroide a la nube real de partículas.

4.1. Conclusiones

Aplicando el algoritmo de *snake* podemos ser capaces de obtener el contorno de la imagen que estemos analizando, la ventaja que presenta GVF respecto al *snake* clásico es que este es mas robusto respecto al ruido y converge mejor hacia los contornos, el algoritmo tiende a converger muy bien cuando los contornos de las imágenes están bien definidos pero en nuestro caso no sucede así, ya que nuestra distribución no presenta una forma geométrica uniforme (como la que mostraria un circulo, cuadrado, rectangulo, etc). Por ello el algoritmo trata de encontrar la mejor representación del contorno para esta.

Una vez que obtuvimos nuestra distribución de partículas es necesario realizar la búsqueda del centroide, a sugerencia del profesor me base en la búsqueda del centro de masa, donde, en base al contorno obtenido se calcula el centroide de la distribución y se muestra mediante un punto que le representa, tanto para datos simulados como para datos reales obtuvimos los resultados esperados.

Como sugerencia, para evitar que el algoritmo sea tardado y no converga de forma adecuada, es mejor utilizar imágenes de baja resolución tanto para la búsqueda del contorno como para la búsqueda del centroide, esto es aplicable para nuestro caso solamente, desconozco si el desempeño del algoritmo se vea afectado utilizando imágenes de alta resolución con imágenes bien definidas.

5. Clasificador de tornillos

En la Figura 14 mostramos los tornillos utilizados para las pruebas, donde mediante la aplicación del algoritmo de contornos GVF snake obtenemos una salida calculada de acuerdo a la forma en que el snake converge hacia el objeto como se observa en las Figuras 15 y 16.

Posteriormente obtenemos las ROI obtenidas de nuestro algoritmo de contorno con el fin de realizar la clasificación, estas regiones se muestran en la Figura 17.

Utilizando MOP (multi-objective parameterized interested point detector) desarrollado anteriormente se obtuvieron las imagenes mostradas en la Figura 18.

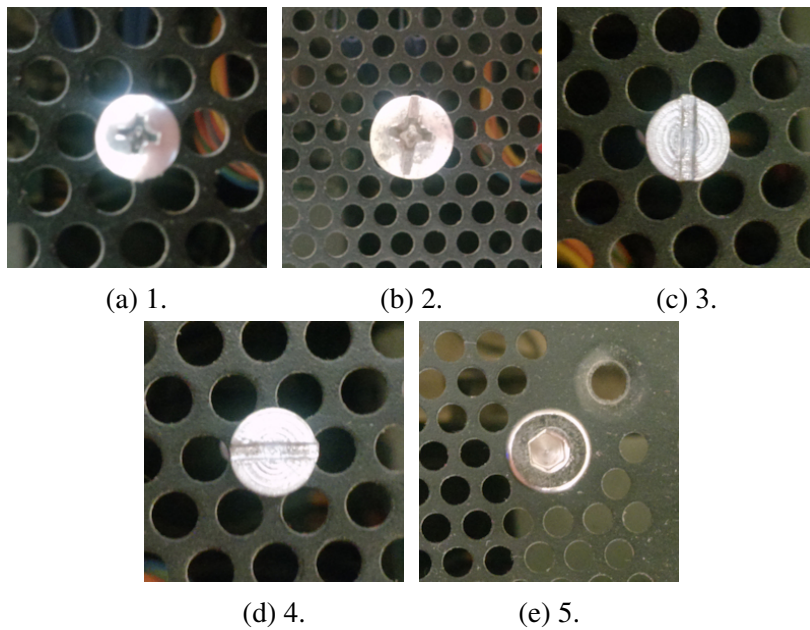


Figura 14: Tornillos utilizados para la búsqueda del contorno mediante el algoritmo de GVF snake.

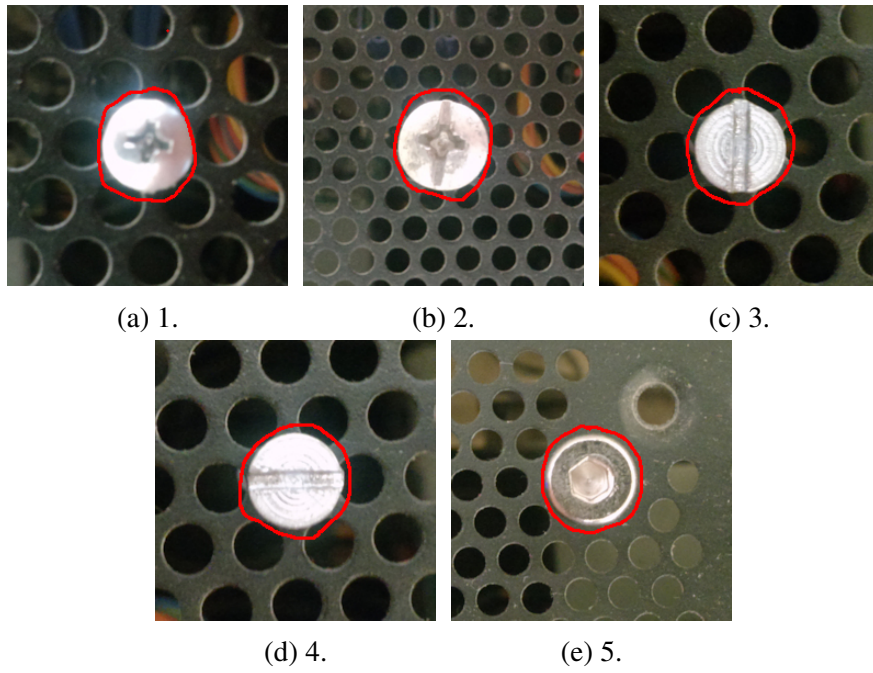


Figura 15: Contornos iniciales propuestos por el usuario.

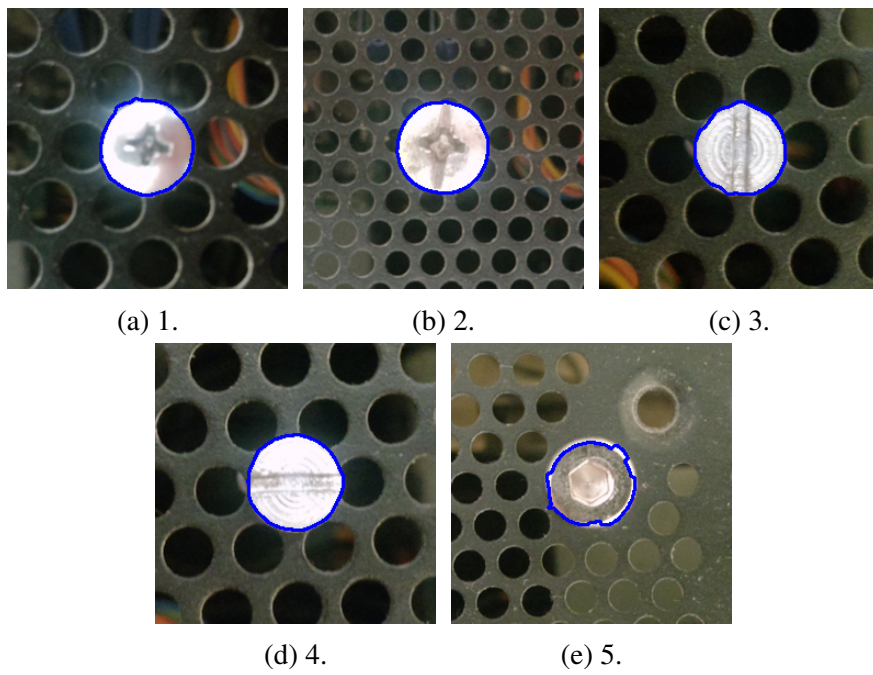


Figura 16: Contornos obtenidos mediante el GVF snake.

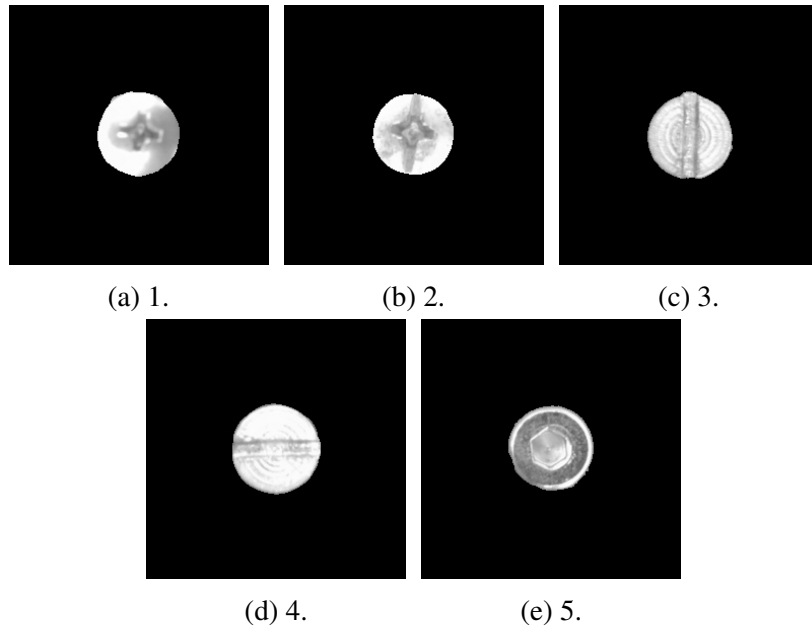


Figura 17: ROI obtenidas del algoritmo de GVF snake, donde se aplica un valor de threshold para su obtencion.

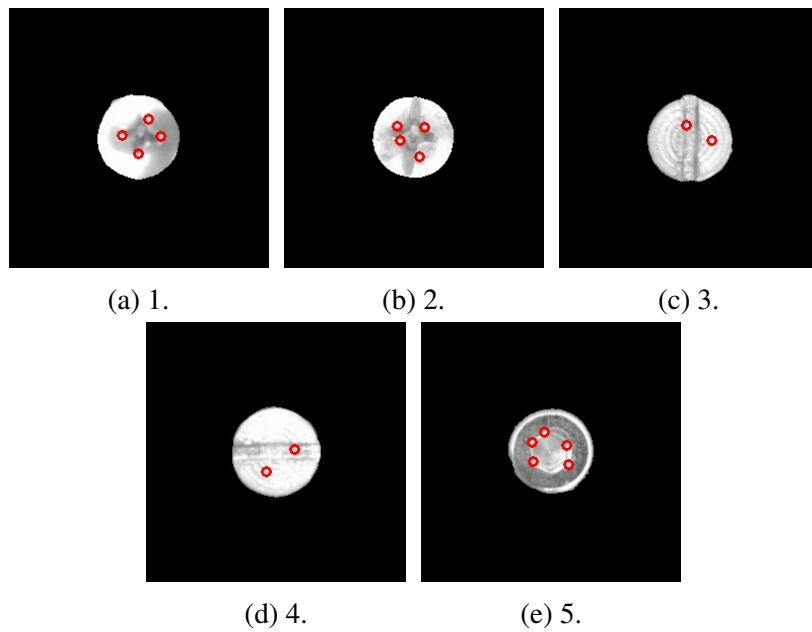


Figura 18: Puntos de interes obtenidos mediante MOP (multi-objective parameterized interested point detector) .

5.1. Conclusiones

Primeramente, se identifica el contorno de los tornillos por medio del algoritmo de contornos activos, una vez que se tiene la imagen generada de este proceso, el siguiente paso consiste en realizar la transformación morfológica de apertura, y al resultado aplicar un threshold para crear una máscara que permita mantener únicamente la región de interés.

Posteriormente en nuestra región de interés aplicamos el detector de puntos de interés MOP, en el que dependiendo de la cantidad de puntos de interés detectados se realiza la identificación del tipo de tornillo, en base al diseño de su cabeza, como plano (1 a 2 puntos), de cruz (3 a 4 puntos), o hexagonal (5 a 6 puntos).

Una de las dificultades que se presentaron en la realización del proyecto fue el fondo en el que se encontraba el tornillo, puesto que, al trabajarlo con un fondo claro, el contorno detectado no era el correcto, de la misma manera con la detección de puntos de interés, si cambia la inclinación con la que es tomada la fotografía la cantidad de puntos detectados para los tornillos con diseño hexagonal en ocasiones se ve reducido a 4 lo cual causa que sea identificado como uno de cruz.

Referencias

- [1] R. M. Haralick and L. G. Shapiro. Image segmentation techniques. *Computer vision, graphics, and image processing*, 29(1):100–132, 1985.
- [2] J. Ivins and J. Porrill. Everything you always wanted to know about snakes (but were afraid to ask). 1995.
- [3] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.
- [4] A. K. Jain, R. P. Duin, and J. Mao. Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37, 2000.
- [5] A. K. Jain, A. Topchy, M. H. Law, and J. M. Buhmann. Landscape of clustering algorithms. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 1, pages 260–263. IEEE, 2004.
- [6] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.
- [7] L. Kaufman and P. J. Rousseeuw. *Introduction*, pages 1–67. John Wiley and Sons, Inc., 2008.
- [8] A. Rosenfeld. Image processing in remote sensing data analysis: The state of the art. *Proceedings of the Indian Academy of Sciences Section C: Engineering Sciences*, 6(2):145–152, 1983.
- [9] C. Xu and J. L. Prince. Gradient vector flow: A new external force for snakes. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 66–71. IEEE, 1997.