

RECONOCIMIENTO DE ROSTROS CON VISIÓN POR COMPUTADORA Y OPTIMIZACIÓN BASADA EN CÚMULOS DE PARTÍCULAS

Mitzel Avilés Quintero

Estudiante de Maestría en Ing. en Manufactura

Universidad Autónoma de Ciudad Juárez

Resumen

En la actualidad la solución a diversos problemas como el reconocimiento de rostros, demandan un sin fin de recursos para el análisis exhaustivo de una población. Una de las ramas que se encargan de resolver tales problemas mediante las llamadas técnicas heurísticas, es el cómputo evolutivo a través de los algoritmos evolutivos. La optimización basada en cúmulo de partículas, es una técnica que responde de manera eficiente en la búsqueda de soluciones óptimas, que está inspirada en el comportamiento de las parvadas de aves. Esta investigación describe el algoritmo PSO aplicado en el reconocimiento de rostros con visión por computadora.

Palabras clave: PSO, algoritmo, optimización, visión, procesamiento.

Introducción

Un problema común en computación es encontrar información dentro de un conjunto de datos siguiendo un criterio determinado. Una de las principales dificultades en dicho problema es que no siempre se puede realizar una búsqueda en todo el espacio para seleccionar la mejor solución (Holland, 1992).

Una forma de encontrar soluciones en el espacio de búsqueda es la optimización combinatoria discreta, que trata de seleccionar una secuencia de números discretos que satisfagan un determinado criterio. La secuencia de

números es de tamaño variable y el criterio depende totalmente del problema que se desea resolver (Olague, 2007)

Recientemente, la computación evolutiva se ha enfocado a la solución de problemas de procesamiento de información digital (imágenes) por medio de una computadora, con el objetivo de ofrecer capacidades de visión cercanas a aquellas encontradas en la naturaleza. El objetivo buscado es resolver tareas que requieren un desempeño de optimización robusto en campos como la visión por computadora, para ofrecer capacidades de percepción muy

cercanas a las que tienen los seres humanos de forma natural (Olague et al, 2006).

La visión por computadora evolutiva es una nueva área de investigación que se centra en el estudio de la visión artificial por medio de enfoques computacionales tanto genético como evolutivo (Agnelli *et al*, 2002).

Un algoritmo que forma parte del cómputo evolutivo es el conocido como optimización basada en cúmulos de partículas, del término inglés Particle Swarm Optimization (PSO). Para el PSO cada solución o partícula es un ave en el espacio de búsqueda que está siempre en continuo movimiento y nunca muere (Engelbrecht, 2005). El cúmulo de partículas puede ser considerado como un sistema multiagente. Las partículas son agentes simples que se mueven por el espacio de búsqueda y que guardan la mejor solución que han encontrado. Cada partícula tiene una aptitud, una posición y un vector de velocidad que dirige su vuelo.

Antecedentes

El algoritmo PSO se basa en una población que evoluciona a través de un algoritmo computacional que resuelve problemas. La optimización por cúmulos de partículas es un tipo de inteligencia basada en la

psicología social y provee un comportamiento social que contribuye a aplicaciones ingenieriles (Marczyk, 2009). El algoritmo de optimización por cúmulo de partículas fue descrito por primera vez por James Kennedy y Ruessell C. Eberhart en el año de 1995 (Engelbrecht, 2005).

El PSO representa conductas artificiales tales como el vuelo de una parvada de aves, el conjunto de una especie de peces, una comunidad de hormigas, un enjambre de abejas, entre otros. De tal forma que para ello se fundamenta en la evolución computacional y los algoritmos genéticos (Muñoz *et al*, 2008).

Entre sus características se destaca que el PSO es un atractivo prototipo donde se tienen partículas en el espacio, las cuales son puestas a volar buscando una solución que las deposite en el mejor lugar de acuerdo a la solución esperada.

Por otro lado, el PSO es un método que busca máximos y mínimos de funciones, basándose en cada uno de los entes que trabajan en conjunto con el fin de encontrar la solución. El método para trabajar con PSO se basa en realizar un espacio multidimensional que depende del número de incógnitas requeridas, donde las partículas se colocan en un tipo de enjambre de manera aleatoria (Lima y Barán, 2006).

En tal método, cada partícula se moverá a través del espacio multidimensional, donde cada movimiento se encontrara evaluado por la función a resolver, para ello es importante ir identificando que posiciones le han favorecido a las partículas en algún segmento del espacio, entonces estos segmentos serán guardados obteniendo así el mayor valor del máximo global de dicha función (Acevedo, 2007).

En el PSO existen dos modelos básicos que dependen de su tamaño (Aragón *et al.*, 2006):

- El algoritmo de optimización por cúmulo de partículas global (gbest) que toma en cuenta al cúmulo de cada partícula a la totalidad.
- El algoritmo de optimización por cúmulo de partículas local (lbest) que tiene diferentes topologías dentro del cúmulo total.

Otra forma de definir al PSO es a través de dos topologías, las cuales son en anillo y en estrella (Carlisle y Dossier, 2001). En la Figura 1 y Figura 2, se presenta un ejemplo de las topologías en anillo y estrella respectivamente.

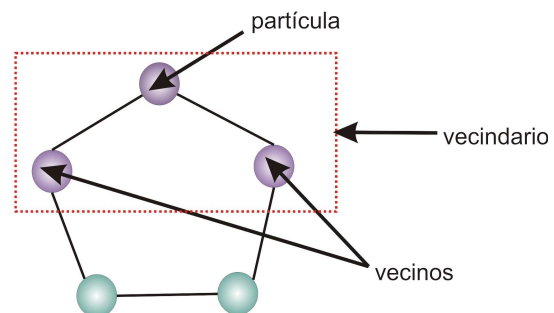


Figura 1. Ejemplo de una topología de anillo [Fuentes, 2008].

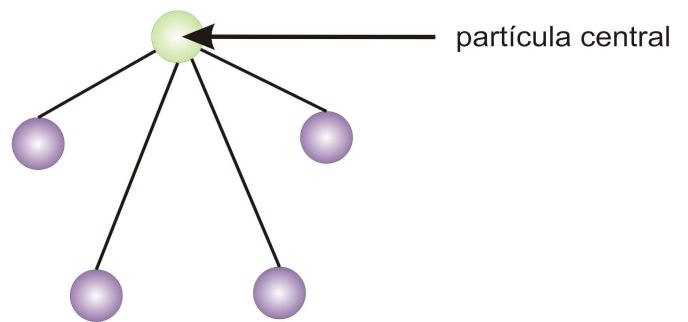


Figura 2. Ejemplo de una topología de estrella [Fuentes, 2008].

En la Tabla 1, se muestran algunos conceptos empleados en el algoritmo básico de optimización por enjambre de partículas. El algoritmo de optimización por cúmulo de partículas tiene como base una población con la cual se realizará un proceso de búsqueda, cada ente es una partícula que está dentro de un enjambre o nube (un ave, dentro de una parvada). Dentro del enjambre cada partícula representa una solución al problema que se requiere optimizar.

Tabla 1. Conceptos del algoritmo

Partícula/ Agente	Un individuo de la nube.
Localización/ Posición	Coordenadas de un agente en un espacio N -dimensional que presenta una solución para el problema.
Nube	Toda una colección de agentes. Una población de individuos.
Fitness (Aptitud)	Un número que indica la calidad de una solución dada (representada por una localización en el espacio de soluciones).
Lbest (Local best)	Es la mejor localización obtenida por un determinado agente a lo largo del proceso.
Gbest (Global best)	Es la mejor localización en toda la nube de partículas.
Vmax	Es la velocidad máxima permitida en una dirección dada.

Objetivos

La tarea de reconocimiento es un problema clásico en visión por computadora, el objetivo es decidir cuándo una imagen contiene algún objeto específico o una característica particular.

La idea principal es utilizar las técnicas de cómputo evolutivo para ayudar a tomar las decisiones cruciales en la solución de problemas de visión artificial.

La investigación propone una metodología para el análisis y diseño de un

algoritmo, que realice el reconocimiento de rostros en imágenes digitales, utilizando optimización basada en cúmulo de partículas. El estudio se llevará a cabo en una población de distintos individuos, con un ambiente diferente para la captura de las imágenes digitales.

Materiales y métodos

Descripción del algoritmo PSO

Los siguientes pasos describen el mecanismo de optimización al utilizar el algoritmo por cúmulo de partículas:

- a) Inicialmente se tiene una población de partículas en un espacio multidimensional, con velocidades y posiciones aleatorias que corresponden al problema de análisis.
- b) Las partículas son evaluadas con la función objetivo.
- c) El valor de la función (fitness) es comparado con el lbest de la partícula. Cuando el valor de fitness es mejor que el lbest, este sustituye al lbest y el que tenía el lbest se intercambia a la posición actual del espacio multidimensional.
- d) Luego el valor de la función fitness se compara con el mejor valor

obtenido de toda la población (gbest).

- e) La velocidad para cada posición de cada partícula se modifica.
- f) Los pasos se vuelven a ejecutar desde el inciso b hasta que el criterio de parada se cumpla y comúnmente se utiliza la variable t_{max} . Donde el criterio de parada se encuentra definido por el número de partículas a evaluar en el espacio multidimensional de la población.

Definición matemática del algoritmo

La parte principal del algoritmo de optimización por cúmulo de partículas son las partículas, las cuales representan un ave en el espacio de búsqueda, matemáticamente dichas partículas se encuentran definidas por:

- Tres vectores
 - El **vector** x que pertenece a la posición actual de la partícula en un estado t .
 - El **vector** $lbest$, que almacena la mejor solución encontrada.
 - El **vector** V que indica la velocidad y dirección con que dicha partícula se

moverá en el espacio de búsqueda.

- Valores de aptitud
 - El valor $x_fitness$ que almacena el valor de la función aptitud del vector x .
 - El valor $l_fitness$ que almacena el valor de la función aptitud del vector $lbest$.

Una vez definidas las partículas en el espacio de búsqueda, se puede expresar el movimiento de cada una de ellas con las siguientes ecuaciones:

$$v_{i(t+1)} = w * v_i(t) + c_1 * rand() * [lbest(t) - x_i(t)] + c_2 * rand() * [gbest(t) - x_i(t)]$$

$$x_{i(t+1)} = x_i(t) + \Delta t * v_i(t + 1)$$

Donde:

- $x_{i(t+1)} = [x_{i1}, x_{i2}, \dots, x_{in}]^T$ guarda la posición de cada partícula.
- $v_{i(t+1)} = [v_{i1}, v_{i2}, \dots, v_{in}]^T$ guarda la velocidad para cada partícula.
- $lbest_{i(t+1)} = [p_{i1}, p_{i2}, \dots, p_{in}]^T$ representa el mejor valor de la función aptitud de cada partícula.
- w ponderación de inercia.

- c_1 , c_2 son las ponderaciones de aprendizaje que controlan los componentes cognitivo y social respectivamente e influyen a cada partícula en dirección de los valores ***gbest*** y ***lbest***.
- $rnd()$ son números aleatorios generados.

Desarrollo del algoritmo

El desarrollo del algoritmo basado en optimización por cúmulo de partículas, se basa en la definición del vector inicial \mathbf{x} que es igual al número de partículas y a la vez es la entrada del algoritmo mediante el cual se busca la solución a un problema.

Una de las características más importantes para el algoritmo es introducir las ecuaciones (1) y (2), ya que mediante el vector velocidad \mathbf{v} , las partículas se mueven a través del vector \mathbf{x} .

El desarrollo del pseudocódigo del PSO es la clave para definir la función de cada una de las variables que definen el algoritmo, con ello se posibilita implementarlo en un problema primeramente trivial, capaz de identificar que su solución sea óptima.

En el algoritmo 1 se muestra el algoritmo original del PSO. Éste requiere como entrada el número de partículas que

constituyen el cúmulo y el número de iteraciones.

El pseudocódigo del PSO muestra en primera instancia que se inicializa la población de partículas, las cuales se encuentran definidas por el problema, posteriormente se crea el vector velocidad de forma aleatoria pues es necesario para empezar con el algoritmo.

Luego el algoritmo del PSO necesita la creación de los vectores ***lbest*** y ***gbest***, haciendo posible la ejecución de las ecuaciones (1) y (2) dentro del pseudocódigo.

Finalmente, se obtiene como salida la mejor solución al ejecutarse el algoritmo con un número finito de iteraciones. Las iteraciones en la mayoría de los casos son una entrada al algoritmo ya que dependen del espacio de búsqueda que es proporcional al número de partículas, es decir, entre mayor sea el número de partículas mayor será la búsqueda de soluciones para una mejor optimización.

Algoritmo 1: Pseudocódigo del PSO

Entrada: Número de partículas, número de generaciones

Salida: Mejor solución

begin

Inicializar una población de partículas con posiciones y velocidades aleatorias.

repeat

for i = 1 to num partículas do

if $G(x_i) > G(lbest_i)$ then

for d = 1 to num dimensiones do

$lbest_i = x_i$; //

pb_{id} es el mejor

hasta el

momento

end

end

$g = i$; // Valor arbitrario

for j = índices de los vecinos do

if $G(gbest_j) >$

$G(gbest_g)$ then

$g = j$; //g es

el índice del

mejor

individuo en

el vecindario

end

end

for d = 1 to num dimensiones do

$v_{id} = w * v_{id} + c_1 * rnd() * [$

$lbest_{id} - x_{id}] + c_2 * rnd() * [gbest_{id} -$

$x_{id}]$;

$x_{id} =$

$x_{id} + v_{id}$;

end

end

until número máximo de generaciones

Reportar la mejor solución.

End

Resultados y discusión

Implementación del algoritmo PSO con fotografías bajo un ambiente no controlado

Con el algoritmo PSO se creó un sistema para el reconocimiento de rostros, donde las pruebas se realizaron en un ambiente poco controlado, para ello se tomaron fotografías de 200x150 píxeles, entre ellas hombres de distintas edades, mujeres y niñas.

La siguiente fase para la implementación del algoritmo fue crear una base de datos con las fotografías obtenidas, en la figura 3 se observan esas fotografías con las cuales se hace la búsqueda en el espacio de población para PSO.

Debido a que dentro de la aplicación para el reconocimiento de rostros mediante PSO, aún no tiene un mecanismo de captura de fotografías para realizar el reconocimiento de rostros con la base de datos ya integrada al sistema, se optó por

realizar una base de fotografías de donde se elegirá una de esas fotos para hacer las pruebas pertinentes.

Esta base de datos contiene a las mismas personas que la base de datos de reconocimiento, con la diferencia de que se les pidió a los participantes hacer una expresión diferente y muy marcada para probar la eficiencia tanto de la función aptitud como la del algoritmo.

La figura 4, muestra las fotografías adquiridas para hacer las diferentes pruebas en el algoritmo PSO de reconocimiento de imágenes, tales fotografías podrán ser elegidas una a una dependiendo del rostro que se desea buscar en la base de datos.

Tanto las fotografías de la Figura 3 como las de la Figura 4, muestran diferencias en el fondo, por ejemplo el color, las sombras, la iluminación, etc.

Tales diferencias pueden afectar el funcionamiento y la eficiencia del algoritmo PSO, ya que se evalúa a las imágenes en su totalidad. Sin embargo, se tiene la ventaja de que los participantes se colocaron en el mismo lugar para la sesión de fotografías en un lapso de tiempo que oscilaba entre 5 y 10 minutos.

Pruebas del algoritmo

Se iniciaron las pruebas eligiendo la fotografía número 2 de la base de datos donde los individuos presentan una emoción de enojados. De la fotografía número 2 de la figura 4 se observa que el individuo tiene mucha similitud al individuo 3 ya que las características del fondo, la camisa y el color de piel son similares.

En la figura 5, se observa el rostro elegido (fotografía 2 de la figura 4) y como resultado se obtuvo en primer lugar al mismo individuo que se encuentra dentro de la base de datos de reconocimiento con la diferencia de que su rostro tiene la emoción de enojado. En segundo plano el sistema es capaz de mostrar al segundo individuo más parecido dentro de la base de datos de reconocimiento, y como anteriormente se había planteado el individuo tres era el probable más parecido, dando como resultado él mismo en el sistema de reconocimiento mediante PSO.

Análisis de resultados y discusión

Las pruebas acerca del algoritmo de reconocimiento de rostros mediante PSO, acertaron de manera eficaz a los individuos requeridos.

El sistema, fue capaz de comparar y mostrar a los sujetos requeridos a pesar de

que las gesticulaciones en el rostro dentro de la base de datos de reconocimiento eran distintas a las fotografías de las pruebas. Por otro lado, el sistema responde muy bien frente a la necesidad de ampliar las soluciones cuando un individuo no es

encontrado dentro de la base de datos de reconocimiento, es decir, entre más amplío sea el rango de diferencias entre las fotografías encontradas más individuos serán presentados como posible solución.



Figura 3. Base de datos para la búsqueda de rostros.



Figura 4. Base de datos para la elección de la fotografía a reconocer.

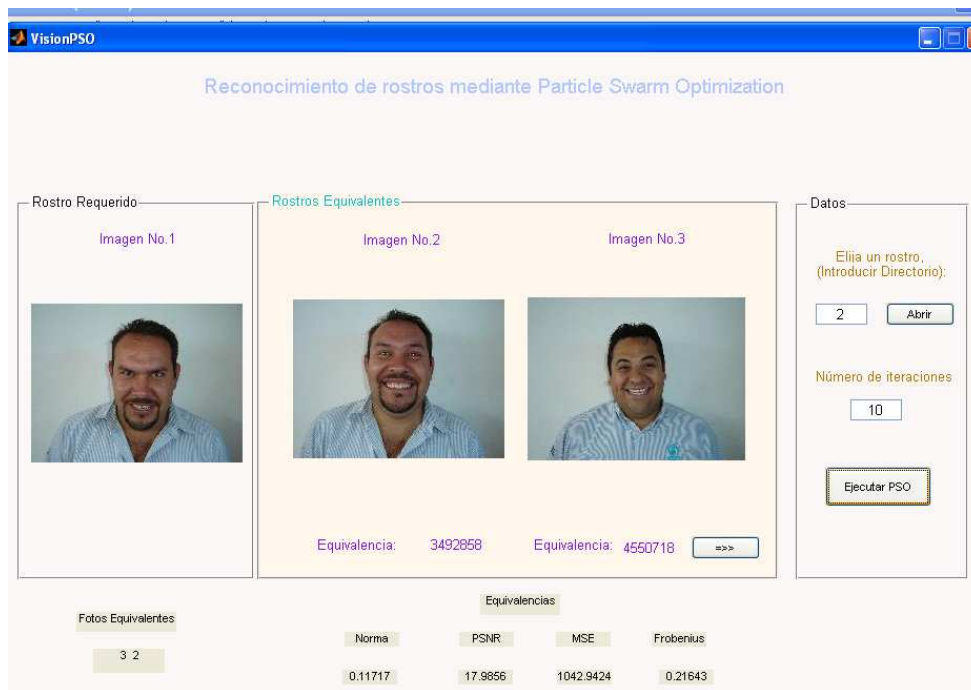


Figura 5. Prueba del reconocimiento de rostros mediante PSO.

Conclusiones

La técnica de PSO, se encuentra diseñada para responder a problemas donde el espacio de búsqueda es muy grande, y el intentar encontrar la solución perfecta es casi imposible demandando muchos recursos computacionales. Tal motivo, impulsó la creación de un algoritmo basado en esta técnica.

Un gran desafío dentro de la visión por computadora, es el reconocimiento de rostros, para llevar ese problema desde su aplicación común a un algoritmo de computación evolutiva, fue indispensable identificar los puntos

exactos y variables donde se debía atacar el problema, para ello se redefinió cada variable dentro de PSO con el reconocimiento de rostros y se creó la aplicación dentro de Matlab 7.5.

El algoritmo PSO fue sometido a un ambiente con fotografías dentro de un marco poco controlado, con lo que se demostró la eficiencia y optimización que se da el crear algoritmos basados en la computación evolutiva, mejorando por mucho las aplicaciones comunes de un sistema no creado para grandes poblaciones.

Referencias

Carlisle, A. and Dossier, G. 2001. An off-the-shelf PSO, Indianapolis, IN, *Proceedings of the Particle Swarm Optimization Workshop*, pp. 1-6.

Marczyk, A. 2009. *Algoritmos genéticos y computación evolutiva*, [En línea] disponible: <http://the-geek.org/docs/algen>, Octubre de 2010.

Engelbrecht, A. P. 2005. *Fundamentals of computational swarm intelligence*, Sudáfrica, John Wiley & Sons, LTD.

Agnelli, D., Bollini, A. y Lombardi, L. 2002. Image classification: an evolutionary approach, Estados Unidos, *Pattern Recognition Letters*, vol. 23, no. 1-3, pp. 303-309.

Acevedo, D. L. 2007. *Aprendizaje y comportamiento social y emergente en sociedades artificiales*, Universidad Nacional de Colombia, Seminario de investigación, Maestría en Ingeniería, pp.1-9.

Olague, G., Cagnoni, S. y Lutton, E. 2006. Introduction to the special issue on evolutionary computer vision and image understanding, Estados Unidos, *Pattern Recognition Letters*, vol. 27, no. 11, pp. 1161-1163.

Olague, G. 2007. *Evolutionary computer vision, survey on the state of the art*, Londres, Inglaterra, Tutorial of the Genetic and Evolutionary Computing Conference (GECCO).

Fuentes Cabrera, J. C. 2008. *Un nuevo algoritmo de optimización basado en optimización mediante cúmulos de partículas utilizando tamaños de población muy pequeños*,

Tesis de Maestría, Centro de Investigación y de Estudios Avanzados (Cinvestav) del Instituto Politécnico Nacional, Unidad Zacatenco, Departamento de Computación, México, D. F.

Holland, J. 1992. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, Cambridge, MA., second ed. MIT Press.

Kennedy, J. y Eberhart, R. 1995. Particle swarm optimization”, Piscataway, NJ, *Proceedings of IEEE International Conference on Neural Networks*, vol. 6, pp. 1942-1948.

Lima, J. Q. M. y Barán, B. 2006. Optimización de enjambre de partículas aplicada al problema del cajero viajante bi-objetivo, *Revista Iberoamericana de Inteligencia Artificial*, vol. 10, no. 32, pp. 67-76.

Muñoz, M., López, J. y Caicedos, E. 2008. Inteligencia de enjambres: sociedades para la solución de problemas (una revisión), *Revista de Ingeniería e Investigación*, vol. 28, no. 2, pp. 119-130.

Aragón, V., Cagnina, L. y Esquivel, S. 2006. Metaheurísticas basadas en inteligencia computacional aplicadas a la resolución de problemas de optimización restringidos, *Anales del VIII Workshop de Investigadores en Ciencias de la Computación (WICC)*, pp. 195-201.

