

Control Visual Mediante Optimización Basada en Enjambres Artificiales de Abejas

Marco Perez-Cisneros, Jose Luis Muñoz-Velasco, Erik Valdemar Cuevas Jimenez,
Daniel Zaldivar Navarro

{marco.perez, erik.cuevas, daniel.zaldivar}@cucei.udg.mx, jl_csyc@live.com*

(Paper received on August 10, 2012, accepted on August 24, 2012)

Resumen Este artículo presenta un nuevo esquema para utilizar un algoritmo de optimización heurística en la implementación de esquemas de control visual basado en imagen. Tomando como base la optimización heurística, la tarea de control visual se plantea como un problema de optimización. El error entre las características de la imagen de referencia y las características instantáneas en la imagen permiten plantear una función objetivo a minimizar. Las diversas pruebas realizadas destacan lo útil que puede llegar a ser la optimización aplicada a este tipo de problemas, sobre todo cuando el método clásico presenta problemas de inestabilidad y convergencia.

Palabras Clave: IBVS, control visual, optimización heurística, ABC abejas.

1 Introducción

Desde los años ochenta, el control visual ha desarrollado los postulados teóricos que sustentan la implementación de distintos esquemas de regulación visual. Un número importante de implementaciones se han reportado en la literatura con aplicaciones en la aeronáutica, en naves acuáticas, robots médicos, entre otras [1],[2],[3].

Los esquemas de control visual pueden clasificarse en tres enfoques: control basado en imagen (2D), control basado en posición (3D) y un enfoque híbrido (2 1/2 D) [4],[5].

Este proyecto de investigación concentra el interés sobre el Control Visual Basado en Imagen (IBVS por sus siglas en inglés). En este contexto, la tarea de control consiste en determinar la entrada que se aplica al sistema robótico de acuerdo con el error que se genera al comparar la imagen de referencia y la imagen actual de la cámara.

El esquema se implementa bajo la arquitectura “eye-in-hand” que refiere al montaje de una cámara sobre el eslabón final de una cadena robótica de seis grados de libertad (6DOF). La relación entre la velocidad de la cámara τ y la variación del tiempo de las características visuales \dot{s} es dada por:

$$\dot{S}(t) = L_s \tau(t) \quad (0.1)$$

Donde L_s es la matriz de interacción asociada a s . El objetivo del control visual es minimizar el error $e(t)$ entre las características de referencia s^* , suponiendo que son constantes y las características medidas $s(t)$. Se define de la siguiente manera:

$$e(t) = s(t) - s^* \quad (0.2)$$

Con el fin de satisfacer un decaimiento exponencial del error (1.2), matemáticamente expresado por:

$$\dot{e}(t) = -\lambda e(t) \quad \text{Con } \lambda > 0 \quad (0.3)$$

Y considerando el modelo de lazo abierto (0.1), obtenemos la ley de control de retroalimentación clásica escrita como:

$$\tau(t) = -\lambda L_s^+ e(t) \quad (0.4)$$

Donde L_s^+ es la pseudo-inversa de la matriz de interacción aproximada.

Otras formas de tratar con las tareas del IBVS es usando leyes de control avanzadas como control óptimo [6], desigualdades lineales matriciales [7], modelo de control predictivo (MPC) [8], [9], [3], [10] y usando algoritmos de optimización heurística.

Una parte fundamental de esta investigación se basa en la aplicación de esquemas de optimización heurística dentro de la estructura de un controlador visual basado en imagen 2D. El planteamiento supone que dicha inclusión permitirá mejorar el rendimiento del sistema y de forma importante suprimir algunos estados de indefinición en la operación de esquemas de control visual que se han documentado ampliamente en la literatura [11].

La implementación del algoritmo de optimización heurística nos permite hacer simulaciones haciendo uso de un modelo robótico en combinación con el modelo de la cámara y obtener el resultado que minimice de manera óptima el error entre las características de referencia y las características actuales. Evadiendo de esta manera problemas de estabilidad que se presentan de manera inherente en los esquemas de control visual clásico, como la oclusión de alguno de los puntos de la imagen, o cuando la matriz jacobiana es una matriz singular.

Existe una amplia gama de algoritmos de optimización heurística que pueden utilizarse en este caso. Algunos de estos se encuentran disponibles en la literatura. Atendiendo criterios sobre la facilidad de implementación y adaptación al algoritmo de control visual, se decidió experimentar en primera instancia con el algoritmo de “Artificial Bee Colony” (ABC).

Este artículo está organizado de la siguiente manera: En la sección 2 se reafirma el contexto del estudio y se menciona brevemente el algoritmo “Artificial Bee Colony”. Después, en la sección 3, la estrategia propuesta es desarrollada. En la sección 4, los experimentos de simulación remarcan la efectividad del método propuesto. Finalmente, en la última sección, se detallan las conclusiones y trabajo futuro.

2 IBVS planteado como un problema de optimización

2.1 Algoritmo de colonia artificial de abejas

El algoritmo de la colonia artificial de abejas (Artificial Bee Colony ABC) asume la existencia de un conjunto de operaciones que asemejan algunas características del comportamiento de las abejas melíferas. Por ejemplo, cada solución dentro del espacio de búsqueda incluye un conjunto de parámetros representando posiciones de fuentes de alimento. El valor de afinidad (fitness) hace referencia a la calidad de la fuente de comida, la cual está fuertemente relacionada con la posición de la misma. El proceso imita la búsqueda de las abejas por fuentes de alimento valiosas dando como resultado un proceso análogo para encontrar soluciones óptimas [12].

2.1.1. Perfil biológico de las abejas

El modelo mínimo para una colonia de abejas melíferas consiste de tres clases de abejas: abejas obreras, abejas espectadoras y abejas exploradoras [13]. Las abejas obreras serán responsables de investigar las fuentes de alimento y compartir la información con abejas espectadoras reclutadas. Éstas, a su vez, tomarán una decisión al escoger fuentes de comida considerando dicha información. Las fuentes de alimento que tengan una mayor calidad tendrán más posibilidades de ser seleccionadas por abejas espectadoras que aquellas que presenten una calidad más baja. Una abeja obrera cuya fuente de comida es rechazada por abejas obreras y espectadoras debido a su baja calidad cambiará a ser una abeja exploradora para buscar aleatoriamente por nuevas fuentes de alimento. Por lo tanto, la explotación es efectuada por abejas obreras o espectadoras mientras que la exploración es llevada a cabo por abejas exploradoras.

Los detalles de implementación de tales operaciones tipo abeja dentro del algoritmo ABC son descritas en la siguiente sección.

2.1.2. Descripción del algoritmo

Semejante a otros enfoques basados en enjambre, el algoritmo ABC es un proceso iterativo. Comienza con una población de soluciones o fuentes de comida generadas aleatoriamente. Las siguientes tres operaciones son aplicadas hasta que un criterio de paro es alcanzado [14]:

1. Enviar abejas obreras.
2. Selección de fuentes de comida por parte de abejas espectadoras.
3. Determinar las abejas exploradoras.

2.1.2.1. Inicialización de la población

El algoritmo comienza definiendo el tamaño N_p de la colonia, posteriormente se inicializan $N_f = N_p/2$ fuentes de alimento para las abejas obreras, la mitad restante de abejas serán observadoras; cada fuente de comida es un vector D -dimensional que contiene los valores de los parámetros a ser optimizados, los cuales son aleatoria y uniformemente distribuidos entre los límites inferiores x_j^{\inf} y superiores x_j^{\sup} previamente definidos.

$$x_{j,i} = x_j^{\inf} + \text{rand}(0,1) \cdot (x_j^{\sup} - x_j^{\inf}); \quad (1.1)$$

$$j = 1, 2, \dots, D; i = 1, 2, \dots, N_f$$

Con j e i siendo los índices de parámetro y población respectivamente. Por lo tanto $x_{j,i}$ es el j -ésimo parámetro del i -ésimo individuo (posición de fuente de alimento).

2.1.2.2. Enviar abejas obreras

La cantidad de abejas obreras es igual al número de fuentes de alimento. En esta etapa cada abeja obrera genera una nueva fuente de comida en la vecindad de su posición actual de la siguiente manera:

$$v_{j,i} = x_{j,i} + \phi_{j,i} (x_{j,k} - x_{j,i}); \quad (1.2)$$

$$k \in \{1, 2, \dots, N_f\}; j \in \{1, 2, \dots, D\}$$

$x_{j,i}$ Es un parámetro j seleccionado aleatoriamente del i -ésimo individuo y k es una de las N_f fuentes de alimento, satisfaciendo la condición $i \neq k$. Si un parámetro dado de la solución candidata v_i excede sus límites predeterminados, ese parámetro debe ser ajustado de manera tal que se encuentre en el rango apropiado. El factor de escalamiento $\phi_{j,i}$ es un número aleatorio entre $[-1, 1]$. Una vez que una nueva solución ha sido generada, se calcula un valor de afinidad asociado con una solución particular el cual representa su viabilidad.

El valor de afinidad para un problema de minimización puede ser asignado a cada solución v_i a través de la siguiente expresión:

$$\text{fit}_i = \begin{cases} \frac{1}{1+J_i} & \text{si } J_i \geq 0 \\ 1+abs(J_i) & \text{si } J_i < 0 \end{cases} \quad (1.3)$$

Donde J_i es la función objetivo a ser minimizada. Posteriormente se aplica un proceso de selección rigurosa entre v_i y x_i . Si la cantidad de néctar (afinidad) de v_i es mayor, entonces la solución x_i es reemplazada por v_i ; en otro caso x_i permanece.

2.1.2.3. Selección de fuentes de comida por abejas espectadoras

Cada abeja espectadora (La cantidad de abejas espectadoras corresponde al número de fuentes de alimento) selecciona una de las fuentes de comida propuestas dependiendo en su valor de afinidad, el cual ha sido previamente definido por las abejas obreras. La probabilidad de que una fuente de comida sea seleccionada es obtenida a partir de la siguiente ecuación:

$$Prob_i = \frac{\text{fit}_i}{\sum_{j=1}^{N_f} \text{fit}_j} \quad (1.4)$$

Donde fit_i es el valor de afinidad de la fuente de comida i el cual está relacionado al valor de la función objetivo J_i correspondiente a la fuente de comida i . La probabilidad de que una fuente de comida sea seleccionada por una abeja espectadora incrementa con un aumento en el valor de afinidad de la fuente de comida. Después de que la fuente de alimento es seleccionada, las abejas espectadoras irán a dicha fuente de comida y escogerán una nueva fuente de alimento candidata dentro de la vecindad de la fuente de comida seleccionada inicialmente. La nueva fuente de alimento candidata es expresada y calculada por medio de eq. (2.2). En caso de que la cantidad de néctar *i.e.* afinidad de la nueva solución sea mejor que antes, dicha posición es mantenida; en otro caso la última solución permanece.

2.1.2.4. Determinar abejas exploradoras

Si la posición de una fuente de comida i (solución candidata) no puede ser mejorada a lo largo de un número predeterminado de intentos conocido como “límite”, la fuente de alimento se abandona y la abeja empleada u observadora correspondiente se convierte en una exploradora. Una abeja exploradora investiga el espacio de búsqueda sin información previa *i.e.* la nueva solución es generada aleatoriamente tal como se indica en eq. (2.1). Para verificar si una solución candidata ha alcanzado el límite predeterminado un contador A_i es asignado a cada fuente de

comida i . Dicho contador es incrementado como consecuencia de que una operación de tipo abeja falle en mejorar la afinidad de la fuente de comida.

El algoritmo muestra de manera general los pasos efectuados por el algoritmo ABC. Los pasos 6–11 del algoritmo representan las acciones de las abejas obreras mientras que los pasos 12–22 corresponden a las abejas espectadoras. La figura 1a muestra el diagrama de flujo del algoritmo de optimización ABC.

3 Control visual 2D por medio del algoritmo ABC

El IBVS es formulado como la minimización del error en la imagen, de manera que el algoritmo ABC en combinación con el esquema de control visual proporcionen el vector de velocidad que minimice de manera óptima el error entre las características de la imagen actual y las características de la imagen de referencia, en cada iteración.

Para resumir, los pasos que se aplican en el proceso IBVS basado en el algoritmo de optimización ABC son los siguientes:

Paso 1: Se configuran los parámetros de control del algoritmo: tamaño de la colonia, número de fuentes de comida, el límite hasta el cual se abandona una fuente de comida que no puede ser mejorada y el número de ciclos para búsqueda de alimento.

Paso 2: Se asignan las variables específicas del problema: la función de costo a ser optimizada, el número de parámetros del problema a ser optimizados, los límites inferior y superior de los parámetros, y el número de veces que debe ejecutarse el algoritmo para alcanzar la posición deseada.

Paso 3: Inicialización de la cámara (parámetros y la posición inicial en la que se encuentra).

Paso 4: Se generan las fuentes de comida, se prueba cada una en un esquema de control visual simulado y se memoriza la que obtiene mejor resultado.

Paso 5: Se envían a las abejas obreras a las fuentes de comida (cada abeja modifica una posición en la fuente de comida y se evalúa la nueva solución en el esquema de control visual simulado).

Paso 6: Cálculo de probabilidades (la fuente de comida es elegida en base a su probabilidad, la cual es proporcional a su calidad).

Paso 7: Envío de las abejas observadoras a buscar los lugares de fuentes de comida (la abeja espectador decide qué fuente de comida explotar de manera probabilística, se evalúan las nuevas fuentes de comida en el esquema de control visual y se memoriza la mejor solución).

Paso 8: Aplica la mejor solución guardada en memoria al esquema de control visual real.

Paso 9: Regresa al paso 4 y se repite este mismo proceso hasta que se alcance el número de veces que debe ejecutarse el algoritmo para llegar a la posición deseada.

4 Experimentos de simulación

Diversas pruebas muestran lo que puede aportar el algoritmo ABC al esquema de control visual basado en imagen, los resultados son comparados con el esquema de control clásico introducidos en la sección 1, eq (1.1) (1.2) (1.3)

La tarea de control consiste en posicionar la perspectiva de una cámara de vuelo libre con respecto a la proyección en perspectiva de 4 puntos. Estos cuatro puntos forman un cuadrado de 400 pixeles de lado, en el espacio cartesiano

4.1. Rotación alrededor del eje de la cámara

En el IBVS clásico una rotación de π radianes alrededor del eje óptico, se sabe que conduce a una falla en la ley de control [11].

Como el IBVS elige la ruta más corta en el **plano de la imagen**, la cual sería una línea recta de un punto a otro en diagonal, la cámara realiza un retroceso infinito, cayendo en una singularidad. En contraste un controlador basado en el algoritmo ABC genera un control independiente: la rotación de π radianes sobre el eje óptico es realizada con un rango aceptable de traslación en z .

En la figura 1b se muestran los resultados obtenidos al aplicar el algoritmo ABC al esquema de control visual, en ella puede observarse que el error siempre tiende a disminuir en cada uno de los puntos.

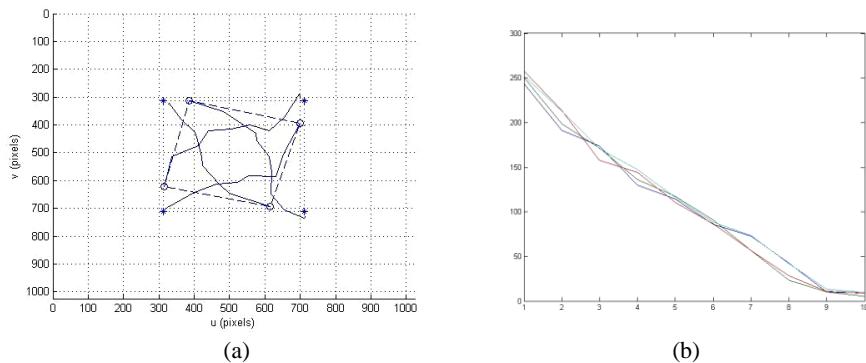


Fig. 1 Desempeño del algoritmo ABC aplicándolo al esquema de control visual. (a) Trayectoria de los puntos en la imagen, (b)

5 Conclusiones

En este artículo, se ha propuesto un método basado en algoritmos de optimización heurística para resolver el problema de control visual basado en imagen. El objetivo es plantear la tarea de control visual como un problema de optimización en el plano de la imagen. La diferencia entre los puntos de referencia en la imagen y los puntos actuales es lo que se pretende minimizar en base al vector de velocidad de la cámara que se ingresa al sistema de control. Se han mostrado resultados de simulaciones y se ha comprobado al resolver varios problemas que al esquema de control visual clásico le es imposible solucionar, que este método puede ser de gran interés para evitar problemas de estabilidad y convergencia que inherentemente se presentan en los esquemas clásicos de control visual basado en imagen.

Esta tarea de control visual basado en imagen formulada como un problema de optimización es adecuada para el manejo de restricciones, por ejemplo las restricciones de visibilidad donde esta propuesta asegura de que los puntos de la imagen se mantengan siempre en el campo de vista de la cámara.

6 Referencias

1. O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, L. Eck, *Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle*, IEEE Trans. on Robotics, to appear, 2009.
2. M. Dunbabin, P. Corke, G. Buskey, *Low-cost vision-based AUV guidance system for reef navigation*, IEEE Int. Conf. on Robotics and Automation (ICRA), pp.7-12, New Orleans, USA, April 2004.
3. J. Gangloff, M. De Mathelin, *Hight speed visual servoing of a 6 dof manipulator using MIMO predictive control*, IEEE Int. Conf. on Robotics and Automation, (ICRA), San Francisco, USA, April 2000.
4. F. Chaumette, S. Hutchinson, *Visual Servo Control, Part I: Basic Approaches*, IEEE Robotics and Automation Magazine, Vol. 14, pp 82-90, December 2006.
5. F. Chaumette, S. Hutchinson, *Visual Servo Control, Part II: Advanced Approaches*, IEEE Robotics and Automation Magazine, Vol. 14, pp 109-118, March 2007.
6. K. Hashimoto, H. Kimura, *LQ optimal and nonlinear approaches to visual servoing*, in *Visual Servoing*, K. Hashimoto, Ed. (Robotics and Automated Systems). Singapore: World Scientific, 1993, vol. 7, pp. 165-198.
7. P. Dan`es, D. Bellot, *Towards an LMI approach to multicriteria visual servoing in robotics*, European Journal of Control, 12(1):86-110, 2006.
8. G. Allibert, E. Courtial, Y. Touré, *Visual Predictive Control for Manipulators with Catadioptric Camera*, IEEE Int. Conf. on Robotics and Automation (ICRA), pp. 510-515, Pasadena, USA, May 2008.
9. G. Allibert, E. Courtial, Y. Touré, *Real-time visual predictive controller for image-based trajectory tracking of mobile robot*, 17th IFAC World Congress, pp. 11244-11249 Seoul, Korea, July 2008.
10. M. Sauvée, P. Poignet, E. Dombre, E. Courtial, *Image Based Visual Servoing through Nonlinear Model Predictive Control*, 45th IEEE CDC, San Diego, USA, December 2006.
11. F. Chaumette, *Potential problems of stability and convergence in image-based and position-based visual servoing*, The Confluence of Vision and Control, Lecture Note in Control and Informations Systems, Vol 237, pp. 66-78, Springer-Verlag, 1998.
12. D. Karaboga, B. Basturk. *On the performance of artificial bee colony (ABC) algorithm*. Applied soft computing, Volume 8, Issue 1, January 2008, Pages 687-697
13. N. Karaboga. *A new design method based on artificial bee colony algorithm for digital IIR filters*. Journal of the Franklin Institute 346 (2009) 328–348.
14. D. Karaboga, B. Akay. *A comparative study of Artificial Bee Colony algorithm*. Applied Mathematics and Computation 214 (2009) 108-132.