



Selección óptima de parámetros para algoritmos de detección de obstáculos con visión monocular

Jorge S. Delgado Morales, Gustavo Viera López, Raúl J. Rodríguez Gómez, Antonio Serrano Muñoz

RESUMEN / ABSTRACT

Una de las tareas más importantes en el campo de la robótica móvil autónoma es la detección de obstáculos. Para la ejecución de esta tarea ha sido empleada con frecuencia la visión computacional, particularmente la visión monocular. Esto se debe a la complejidad inherente a los sistemas de visión estereoscópica y el creciente desarrollo de investigaciones que utilizan una sola cámara para detectar obstáculos. Los algoritmos de procesamiento de imágenes y visión computacional para la detección de obstáculos presentan múltiples parámetros que necesitan ser ajustados para un funcionamiento eficiente según las condiciones del entorno donde opera el robot. En este trabajo se propone un método desarrollado para la selección óptima de los parámetros de este tipo de algoritmos para un ambiente determinado. Para ello se modeló el problema de la detección de obstáculos como un problema de optimización. Además se explica el funcionamiento de dos de estos algoritmos de detección de obstáculos basados en visión monocular que son usados para la validación del método. Para la solución del problema modelado, se incluyen los resultados obtenidos mediante varias metaheurísticas. Finalmente se comparan los resultados del uso de esta técnica en diferentes entornos.

Palabras claves: Detección de obstáculos, Optimización, Metaheurísticas, Procesamiento de Imágenes, Visión Computacional, Robótica Móvil

One of the most important task in the field of mobile robotics is obstacle detection. To solve this task, computer vision has been used often, especially monocular vision. This is due to the inherently complexity of stereo vision systems and the increasing development of the research that uses a single camera to detect obstacles. Computer vision and image processing algorithms for obstacles detection require multiple parameters that need to be adjusted to work efficiently according to the characteristics of the robot and the conditions of the environment in which it operates. In the present work, a method for the optimum selection of the parameters of this kind of algorithm for a certain environment is proposed. To achieve that, the obstacle detection problem was modeled as an optimization problem. Besides, an explanation of two of these algorithms based on monocular vision are given. These are used for the validation of the given method. For the solution of the given problem, obtained results with different metaheuristics are included. Finally, the obtained results from using these techniques in different environments are compared.

Key words: Obstacle Detection, Optimization, Metaheuristics, Images Processing, Computer Vision, Mobile Robotics
Optimal parameters selection for obstacle detection algorithms based on monocular vision

INTRODUCCIÓN

El desarrollo de robots móviles de propósito específico se ha incrementado considerablemente en los últimos años [1]. Esto ha sido condicionado gracias al acceso masivo a la información y la disminución gradual de los costos en los materiales y componentes necesarios. Además, los robots móviles se han insertado y desarrollado en múltiples aplicaciones [2].

Una tarea fundamental para las plataformas robóticas móviles es la navegación. Esta permite al robot conducirse en su entorno para alcanzar un destino dado [2]. Es común en la navegación detectar obstáculos y cambios en el entorno debido a la naturaleza dinámica de éste, lo cual hace posible que el robot pueda planificar sus acciones para alcanzar su meta. Los métodos tradicionales para la detección de obstáculos requerían de sensores de rango, tales como los sensores ultrasónicos o sensores infrarrojos, pero el uso de cámaras y sistemas de visión (junto a técnicas de visión computacional) para esta tarea se ha expandido en los últimos tiempos [3]. Se puede considerar que las cámaras se encuentran entre los sensores más empleados en la robótica, debido al aumento de las capacidades de cómputo de los sistemas, el relativo bajo costo de éstas y a la cantidad de información que pueden proveer [4].

Los sistemas de visión basados en cámaras hacen posible la detección de obstáculos o la localización dentro de un mapa, por poner ejemplos [2], pero son relativamente complejos de desarrollar especialmente si se desea un buen desempeño con el mínimo costo computacional posible. En las últimas dos décadas, estos sistemas se han desarrollado considerablemente y se ha extendido su uso tanto para interiores como exteriores, convirtiéndose en una de las mayores áreas de investigación en la comunidad de robótica móvil [3].

El uso de cámaras para la detección de obstáculos requiere el uso de algoritmos de procesamiento de imágenes y visión computacional. Los sistemas que hacen uso de la visión por computadoras normalmente se dividen en etapas que tratan una o varias imágenes para extraer la información requerida. Los procesamientos a que son sometidas las imágenes presentan parámetros y éstos pueden llegar a ser numerosos en dependencia de la complejidad de los algoritmos y de la cantidad de etapas que formen el sistema.

La selección de los parámetros en los algoritmos de procesamiento de imágenes y visión computacional puede llegar a ser un reto debido a la naturaleza de estos sistemas. Los parámetros de estos algoritmos deben ser seleccionados con cuidado debido a la relación que puedan presentar los algoritmos de una etapa y la siguiente. O sea, un parámetro puede ser seleccionado de forma tal que garantice un buen resultado para los algoritmos en una etapa, sin embargo estos resultados pueden afectar las etapas siguientes y hacer que el resultado del sistema no sea el esperado.

Para la estimación de parámetros se utilizan un amplio rango de técnicas que a menudo se ajustan a la aplicación en que son empleadas, los datos subyacentes y algunas consideraciones viables [5]. En el trabajo presentado en [5] se realiza un análisis general de la estimación de parámetros en algoritmos de procesamiento de imágenes de bajo nivel y en aplicaciones de visión computacional de más alto nivel. Se analizan casos como el filtrado de imágenes utilizando “kernels”, la restauración de imágenes y la segmentación, además se exponen técnicas para la selección de parámetros para la extracción de rasgos en una escena.

La bibliografía consultada no arroja muchos trabajos que aborden el tema de la selección de parámetros para sistemas de detección de obstáculos utilizando visión computacional. La solución que se plantean muchos autores para el reconocimiento de objetos en general, es el uso de técnicas de “machine learning” que entrenan a los sistemas para que éstos “aprendan” a reconocer y clasificar objetos [5]. La detección de obstáculos en la robótica móvil supone una gran diferencia entre los objetos a detectar debido a la naturaleza dinámica de los ambientes en que se puede mover un robot. Esto hace muy difícil la aplicación de esta técnica [5] debido a la variedad que tienen los obstáculos en cuanto a su apariencia o forma.

A menudo la selección de parámetros se realiza de forma arbitraria y experimental a modo de prueba y error en dependencia de la aplicación. En este trabajo se propone un método automatizado para la selección óptima de parámetros en los algoritmos de visión computacional para detectar obstáculos. Es un método simple que incluye el uso de imágenes del ambiente en que va a operar un robot y la clasificación de los obstáculos en las imágenes por parte de un humano. Se pretende que el sistema se comporte de la forma más similar posible a la que el humano lo hace. Esto se lleva a cabo utilizando métodos metaheurísticos para la optimización de la función de error entre el resultado del sistema de detección de obstáculos y el resultado obtenido por un humano.

SISTEMAS DE DETECCIÓN DE OBSTÁCULOS CON VISIÓN MONOCULAR

Existen numerosos métodos desarrollados por distintos investigadores que permiten la detección de obstáculos utilizando información proveniente de sistemas de visión. El método desarrollado en [6] utiliza métodos de segmentación de colores y

homografía estéreo de colores para diferenciar áreas planas del terreno de los obstáculos. En [7] los autores exponen un método que utiliza el flujo óptico para construir un mapa de rango (similar al que proveen los sensores de rango tradicionales) y de ahí encontrar los obstáculos. Otros autores utilizan la información de 3 dimensiones que proveen los sistemas de visión estereoscópica para detectar y localizar obstáculos usando diversas técnicas [8-10].

Para este trabajo serán analizados dos algoritmos de detección de obstáculos usando una sola cámara. Se han realizado trabajos también en sistemas de detección monoculares como se puede apreciar en [11-16] El primer algoritmo que se aborda a continuación fue presentado por Iwan Ulrich [17] en el año 2000. Este método permite la detección de obstáculos basándose en las imágenes adquiridas solamente por una cámara a bordo del robot móvil. La detección de los obstáculos se basa en la apariencia de éstos y sus diferencias visuales con el resto de la imagen. Para garantizar la efectividad del método propuesto se asumen tres condiciones para el ambiente en que se mueve el robot. Estas consideraciones se cumplen en una gran variedad de entornos, tanto interiores como exteriores. Estas condiciones son:

- Los obstáculos difieren en su apariencia con el suelo
- El suelo es relativamente plano
- No hay obstáculos suspendidos en el aire

La clasificación de un pixel como parte de un obstáculo o del suelo se puede basar en varios atributos visuales, tales como la intensidad, el color, bordes y textura. La información que provee el atributo seleccionado debe ser lo suficientemente rica para que el sistema se comporte de manera fiable en diferentes ambientes. Además el atributo seleccionado debe requerir relativamente poca capacidad de procesamiento de manera que el funcionamiento del sistema pueda ser en tiempo real. Esto garantizaría un buen índice de detección de obstáculos por unidad de tiempo y una mayor rapidez de navegación segura para el robot.

Se seleccionó utilizar la información del color como primera pista para la clasificación de un pixel como obstáculo o suelo. El color tiene muchos atributos de apariencia y provee más información que la utilización de la intensidad únicamente. En comparación con la textura, se puede decir que el color es un atributo más local y conlleva cálculos más rápidos. Los sistemas que se basan en la detección de bordes solo pueden ser utilizados en entornos con suelos sin texturas, además que tienen mayor dificultad para diferenciar entre sombras y obstáculos que los sistemas basados en color [17].

El método para clasificar un pixel como parte de un obstáculo o del suelo utiliza la información de una sola imagen. El no contar con el análisis de imágenes adquiridas anteriormente permite una independencia de las condiciones cambiantes del entorno en que opera el robot, por ejemplo, cambios en la iluminación en las escenas. Además permite eliminar la etapa de entrenamiento propuesta por los autores en [17]. A continuación se muestran los cuatro pasos a seguir para la aplicación de este método:

1. Filtrar las componentes de alta frecuencia en la imagen capturada
2. Transformación al espacio de color HSI
3. Realización de un histograma a un área de referencia
4. Comparación de cada pixel de la imagen con el histograma de referencia

Para filtrar las componentes de alta frecuencia de la imagen se aplica un filtro que disminuye el ruido. Esto permite centrar el análisis en las regiones más amplias obviando detalles pequeños que puedan introducir errores. Al filtrar imágenes es posible que se haga más difícil el proceso de detección de bordes. Esto ocurre debido a que los métodos de detección de bordes a menudo usan los gradientes en las imágenes y el efecto de éstos se ve reducido en el proceso de filtrado. Como el método de detección de obstáculos utilizado no hace uso de los bordes, sino del color, esto no representa una desventaja.

Como área de referencia se debe tomar un área que sea libre de obstáculos. Se asume que el área cercana al frente del robot estará libre de obstáculos al menos para la posición inicial del robot. Esa área es una referencia válida a partir de que se garantice en la navegación del robot que siempre estará libre de obstáculos.

En la Figura 1 se puede ver una imagen captada por la cámara a bordo del robot y el área de referencia libre de obstáculos. Las dimensiones del área de referencia son un parámetro de este algoritmo y pueden cambiar debido a las condiciones del entorno en que opera el robot. Debido a la perspectiva de la cámara el área está delimitada por un trapecio que se extiende a lo largo del plano del suelo y que tiene como base el ancho del robot.



Figura 1

Área de referencia para la captura del histograma.

Una vez delimitada la región de referencia, se realiza un histograma para los valores de matiz e intensidad en esa área. Debido a que la información del color es muy ruidosa para bajos valores de intensidad, solo se tendrán en cuenta los componentes de matiz y saturación si el componente de intensidad asociado está sobre un valor mínimo. De igual forma, para bajos valores de saturación, el componente asociado al matiz no posee información alguna. Por eso solo serán tenidos en cuenta los valores de matiz cuya saturación asociada exceda un valor mínimo. A continuación los histogramas se filtran con un filtro de promedios.

Las ventajas fundamentales del empleo de histogramas para esta aplicación es el bajo requerimiento de capacidad de procesamiento. En cuanto a la cantidad de memoria y el tiempo de cómputo necesario para su creación se puede decir que son relativamente bajos. Además es conocido que los histogramas representan naturalmente distribuciones multimodales [17].

Una vez aplicado el filtro gaussiano, obtener la transformación de espacio de color y la construcción del histograma del área de referencia, se debe analizar toda la imagen. Para la determinación de los obstáculos en la escena, se procede a analizar cada elemento de la matriz que describe la foto original. Luego de esta operación se obtiene una imagen binaria en la que cada pixel es clasificado como parte de un obstáculo o no (ver Figura 2 **Erro! Fonte de referência não encontrada.**).

- Un píxel es clasificado como un obstáculo si al menos una de las siguientes condiciones se satisface:
- El valor del histograma correspondiente a la componente de matiz del píxel se encuentra por debajo de un umbral definido
- El valor del histograma correspondiente a la componente intensidad del píxel se encuentra por debajo de un umbral definido

El segundo algoritmo analizado para la detección de obstáculos mediante el empleo de una sola cámara se basa en el flujo óptico. El flujo óptico es el patrón del movimiento aparente de los objetos, superficies y bordes en una escena causado por el movimiento relativo entre un observador (un ojo o una cámara) y la escena. Es el algoritmo más general para extraer el movimiento aparente de una secuencia de imágenes [18]. Este algoritmo calcula un estimado independiente del movimiento de cada pixel minimizando una función que involucra la diferencia de intensidad o color entre los píxeles de las imágenes. Existen múltiples variantes de detección de obstáculos usando flujo óptico como en [7], [19-22].

De forma general, las técnicas de detección usando flujo óptico se basan en comparar dos imágenes tomadas de forma consecutiva de una misma escena, y analizar patrones que puedan brindar información útil sobre los obstáculos. La efectividad de la detección está influida en gran medida por la frecuencia de muestreo de los cuadros. La detección de obstáculos empleando flujo óptico puede ser descrita de forma breve en 3 pasos:

- Localizar la mayor cantidad de píxeles posibles de la imagen anterior en la nueva imagen.

- Crear una matriz de vectores a partir de la diferencia en la posición de cada píxel.
- Seleccionar bajo algún criterio los vectores que describen obstáculos.



Figura 2

Resultado de la detección de obstáculos con el primer algoritmo.

En la figura 3 se puede observar una muestra de los vectores de flujo óptico extraídos de dos imágenes tomadas de forma consecutiva por una plataforma de robótica móvil.



Figura 3

Vectores de flujo óptico.

Una vez obtenidos los vectores que indican el desplazamiento de cada píxel respecto a la imagen anterior, existen diversas variantes de técnicas que se pueden emplear para extraer información útil referente al ambiente. En este caso se empleó un criterio muy simple, basado en la idea intuitiva de que si el observador se encuentra en movimiento y el ambiente se mantiene invariable, la velocidad con que se desplazan los píxeles correspondientes a obstáculos es diferente a la velocidad correspondiente a los píxeles del suelo.

Por esta razón, estableciendo un umbral en la norma de los vectores se pueden filtrar según este criterio y obtener una aproximación de las regiones de la imagen que pueden ser consideradas regiones de suelo o de obstáculos. Los métodos basados en flujo óptico se han empleado en detección de obstáculos para robots móviles pero aún quedan problemas por resolver en esta técnica como se menciona en [23]. De forma general se obtienen mejores resultados cuando el observador mantiene su posición y los obstáculos son dinámicos.

SISTEMA PARA LA SELECCIÓN ÓPTIMA DE PARÁMETROS

Existen diferentes parámetros presentes en los algoritmos de detección de obstáculos descritos anteriormente. Ejemplos de estos parámetros pueden ser los valores umbrales de matiz y saturación para la clasificación final de los píxeles o el tamaño del área de referencia en el algoritmo basado en apariencia. A continuación se propone un método para la selección óptima de estos parámetros que resulta independiente del algoritmo que se emplee para la detección de obstáculos.

El método diseñado se debe realizar “off-line”, es decir se determinan los parámetros óptimos para que luego éstos sean utilizados por el robot. Nuestro método se describe en los siguientes pasos:

1. Tomar un conjunto de imágenes de prueba para evaluar el algoritmo de detección de obstáculos
2. Indicar los obstáculos en cada imagen de forma manual usando un programa de edición de imágenes
3. Encontrar de forma automática los parámetros que hacen que la detección realizada por el algoritmo sea lo más similar posible a la realizada por un humano.

La obtención de los parámetros óptimos del algoritmo de detección de obstáculos se puede considerar como un problema de optimización. Para esto es necesario modelar nuestro problema mediante una función objetivo. De esta manera se contaría con una función que se puede evaluar varias veces para encontrar valores extremos, es decir, óptimos.

La implementación de este método se basó en encontrar un mínimo en la función del error (o diferencia) entre las imágenes correspondientes a los obstáculos detectados por el algoritmo y por un humano. En la Figura 4 se muestra el sistema desarrollado para encontrar los parámetros óptimos para el algoritmo de detección de obstáculos. El esquema general de la figura muestra cómo se obtiene un valor que representa la calidad de la detección de obstáculos. Esta función varía su valor en dependencia de los parámetros que sean fijados para el algoritmo. De esta forma, el problema se reduce a encontrar los parámetros correspondientes al menor valor del error.

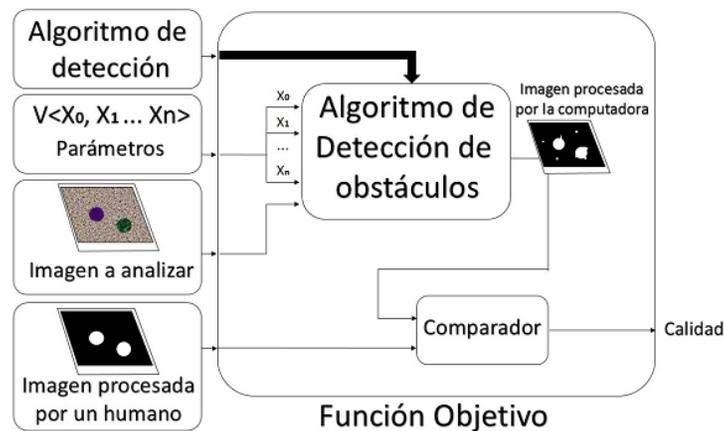


Figura 4

Sistema para la optimización de los parámetros.

Ambas detecciones, por parte del humano y por parte del algoritmo, tienen como resultado imágenes binarias que representan si un píxel forma parte de un obstáculo o no. La calidad de la detección de obstáculos se obtiene como resultado de comparar las imágenes binarias. En el bloque comparador (ver Figura 4) se realiza una comparación píxel a píxel entre las dos imágenes. La relación entre la cantidad de píxeles diferentes y la cantidad total de píxeles en la imagen se define como el error. Este es el que se debe minimizar para obtener una detección de obstáculos con mayor calidad.

El procesamiento de la imagen por parte del humano debe tener en cuenta las particularidades del algoritmo de detección a usar. Esto se debe a que a la hora de evaluar el resultado, se deben considerar las limitaciones de dicho algoritmo. Esto implica que aunque el método de optimización es independiente del algoritmo a optimizar, los datos de entrenamiento no lo son.

Al no contar con una expresión analítica de la función de error no se pueden aplicar los métodos básicos de optimización de funciones. El único recurso con el que se cuenta es con la evaluación de la función de error. La optimización de los parámetros del algoritmo no requiere realizarse en un tiempo determinado, pero se debe tener presente que para evaluar la función de

error se debe ejecutar el algoritmo de detección de obstáculos. Además, el algoritmo tiene varios parámetros y éstos pueden tomar disímiles valores. Entonces se debe tratar de hacer la menor cantidad de evaluaciones posibles de la función para encontrar los valores óptimos de los parámetros.

Los algoritmos metaheurísticos constituyen una buena alternativa para solucionar problemas de optimización, donde la función objetivo es multimodal y una búsqueda exhaustiva no resulta conveniente debido al alto costo computacional de ésta [24]. Una metaheurística es un método heurístico para resolver un tipo de problema computacional general. Este método usa los parámetros dados por un usuario sobre procedimientos genéricos y abstractos.

Para encontrar los parámetros óptimos de nuestro sistema de detección de obstáculos se propone utilizar un algoritmo metaheurístico. Con la utilización de una metaheurística diseñada para optimizar en espacios continuos es posible recorrer el dominio de la función objetivo de forma eficiente [25]. Este dominio está dado por la cantidad de parámetros del algoritmo que queremos optimizar. Debido a la naturaleza mixta de los parámetros, es recomendable seleccionar métodos diseñados para recorrer espacios continuos [24]. Hay que destacar que los resultados que se obtienen al aplicar este método son aproximados.

Métodos metaheurísticos

Existen diversos algoritmos metaheurísticos que se basan en ideas muy diferentes. En nuestro caso, se puede obviar el funcionamiento del algoritmo, ya que lo más importante son los parámetros de entrada y el resultado. De esta forma, se pueden intercambiar diferentes algoritmos y evaluar su desempeño. El resultado que se obtiene es la mejor combinación de variables encontradas que corresponden al menor valor de la función objetivo obtenido durante las evaluaciones. Los parámetros que recibe esta caja negra son:

- Función que se desea optimizar
- Número de dimensión de la función
- Dominio de la función
- Cantidad de evaluaciones posibles

PSO (optimización de enjambres de partículas) es un método metaheurístico estocástico basado en poblaciones. Está basado en estudios sobre inteligencia de enjambres [24]. Se basa en mimetizar el comportamiento social de los organismos naturales como grupos de peces o aves que buscan un lugar para alimentarse. Se ha comprobado que en estos enjambres surgen comportamientos coordinados usando movimientos locales sin ningún centro de control. PSO fue diseñado originalmente para problemas de optimización continuo, pero se puede aplicar exitosamente en problemas discretos con pocas variaciones.

Otro método metaheurístico es el método de búsqueda de población mínima (MPS por sus siglas en inglés) es una metaheurística diseñada para optimizar problemas continuos multimodales con funciones objetivo costosas. La idea de MPS consiste en proveer fuertes mecanismos de exploración y diversificación que le permiten cubrir de manera efectiva el espacio de búsqueda manteniendo un tamaño de población relativamente pequeño [26]. MPS ha demostrado ser una técnica efectiva para solucionar problemas de optimización continuos multimodales razón por la cual se decidió su utilización en este trabajo.

Los autores en [26] recomiendan utilizar como tamaño de población la dimensionalidad del problema (n , cantidad de parámetros). MPS utiliza un mecanismo secundario para escapar el hiperplano dimensional (de $n-1$ dimensiones) definido por los miembros de la población. Para generar nuevas soluciones, se utilizan segmentos de línea para buscar en ese hiperplano y un paso siguiente ortogonal al mismo posibilita abarcar todas las dimensiones del espacio de búsqueda.

RESULTADOS EXPERIMENTALES

Se confeccionaron varios sets de imágenes para poder realizar los experimentos. Cada set contiene un grupo de imágenes representativo de un entorno. En cada experimento, se selecciona una imagen del conjunto y se comprueba el resultado del algoritmo de detección de obstáculos (basado en apariencia o flujo óptico) con parámetros arbitrarios. Posteriormente se utiliza el algoritmo metaheurístico (PSO o MPS) para optimizar los parámetros del algoritmo y se comprueba el resultado.

Tanto los algoritmos metaheurísticos como los de detección de obstáculos se implementaron en el lenguaje Python. Se utilizó OpenCV para el procesamiento de las imágenes. Se realizaron diferentes cantidades de iteraciones a la hora de realizar la optimización. Se encontró que alrededor de 1000 iteraciones para los experimentos realizados ofrecían resultados de error relativamente bajo (los obstáculos se detectaban correctamente). Además estos valores de error son muy cercanos a los obtenidos con un mayor número de iteraciones. En la Tabla 1 se pueden observar los parámetros seleccionados para la optimización con los algoritmos elegidos, así como el rango de posibles valores a tomar. El último parámetro (Norma del vector) es utilizado en el algoritmo basado en flujo óptico.

Tabla 1
Parámetros a optimizar

<i>Parámetro a optimizar</i>	Límite inferior	Límite superior
<i>Tamaño del gaussiano</i>	1	5
<i>Varianza del gaussiano</i>	0.000001	5.0
<i>Valor mínimo del componente saturación</i>	0	50
<i>Valor mínimo del componente intensidad</i>	0	50
<i>Tamaño del histograma de saturación</i>	200	255
<i>Tamaño del histograma de intensidad</i>	200	255
<i>Tamaño del histograma de matiz</i>	200	255
<i>Umbral de matiz</i>	0	200
<i>Umbral de intensidad</i>	0	200
<i>Norma del vector (Flujo óptico)</i>	0.0	5.0

En la Tabla 2 se muestra el error obtenido al ejecutar la detección de obstáculos en una imagen con parámetros seleccionados arbitrariamente por un humano y después de optimizarlos con diferentes métodos metaheurísticos, la Tabla 3 muestra los parámetros seleccionados por un humano y los que se obtuvieron luego de aplicar nuestro método. Se ejecutaron 1000 iteraciones del algoritmo de optimización para calcular dichos parámetros. Como se puede comprobar, se obtienen mejores resultados con los parámetros que resultan de la optimización. Se debe notar que los métodos metaheurísticos logran el mismo error luego de optimizar los parámetros. Ambos algoritmos encuentran los parámetros que arrojan el mismo valor mínimo de error en la función objetivo.

Tabla 2
Error obtenido para una imagen optimizando con 1000 iteraciones.

	<i>Original</i>	<i>MPS</i>	<i>PSO</i>
<i>Basado en apariencia</i>	27.69%	14.16%	14.16%
<i>Basado en flujo óptico</i>	32.65%	15.06%	15.06%

En la Figura 5 se puede ver un ejemplo del resultado antes y después de optimizar para el algoritmo de detección de obstáculos basado en apariencia. En la Figura 6 se aprecia el resultado para el algoritmo basado en flujo óptico.



Figura 5

Resultados para el algoritmo basado en apariencia.



Figura 6

Resultados para el algoritmo basado en flujo óptico.

Tabla 3

Parámetros del algoritmo de detección de obstáculos basado en apariencia.

<i>Parámetro a optimizar</i>	Seleccionado por un humano	Obtenido por la optimización
<i>Tamaño del gaussiano</i>	5	9
<i>Varianza del gaussiano</i>	0.32	0.317
<i>Valor mínimo del componente matiz</i>	50	19
<i>Valor mínimo del componente intensidad</i>	30	48
<i>Tamaño del histograma de saturación</i>	245	241
<i>Tamaño del histograma de intensidad</i>	230	255
<i>Tamaño del histograma de matiz</i>	253	253
<i>Umbral de matiz</i>	195	169
<i>Umbral de intensidad</i>	177	200

El set de parámetros obtenidos después de la optimización es utilizado para aplicar el algoritmo a otras imágenes del mismo set. Esto nos da una medida de lo bien que se comporta el detector de obstáculos en el mismo entorno. De esta forma se comprueba la efectividad de la optimización. En la Figura 7 se puede observar el resultado de aplicar el algoritmo basado en apariencia con los parámetros optimizados a otras imágenes del set.



Figura 7

Detección de obstáculos en otras imágenes del set.

CONCLUSIONES

Como se puede comprobar en los experimentos realizados, el método propuesto es capaz de optimizar los parámetros del algoritmo de detección de obstáculos. De esta forma, se pueden adecuar los parámetros para un entorno de trabajo determinado. De esta manera se logra que la detección de obstáculos se comporte de forma óptima para el ambiente en que se encuentra el robot.

Una propuesta para mejorar el método es aplicar la optimización a todas las imágenes del grupo. El conjunto de parámetros obtenidos de cada imagen puede ser evaluado después en las restantes imágenes del grupo. De esta forma se puede crear un índice de calidad global del set para cada conjunto de parámetros. Esos resultados se pueden someter a un análisis para seleccionar entonces los parámetros de mayor calidad. Idealmente éstos corresponderían a la imagen más representativa del entorno. También se propone comprobar cómo se comporta la evaluación con otros índices de calidad (o medidas de error) diferente al empleado.

REFERENCIAS

1. Siciliano B, Sciavicco L, Villani L, Oriolo G. Robotics. Modelling Planning and Control. Naples and Rome: Springer Verlag; 2009.
2. Siegwart R, Nourbakhsh IR, Scaramuzza D. Introduction to Autonomous Mobile Robotics. 2nd Edition. Massachusetts: The MIT Press; 2011.
3. Gray KW, Baker K. Obstacle detection and avoidance for an autonomus farm tractor. MSc Thesis. Utah State University; 2000. Disponible en: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.449.879&rep=rep1&type=pdf>
4. Chatterjee A, Rakshit A, Sinhg N. Vision Based Autonomous Robot Navigation. Heidelberg: Springer Verlag; 2013.
5. Garbe CS, Ommer B. Parameter Estimation in Image Processing and Computer Vision. In: Bock HG, Carraro T, Jäger W, Körkel S, Rannacher R, Schlöder J. Model Based Parameter Estimation. Berlin: Springer; 2013. p. 311-334.
6. Batavia PH, Singh S. Obstacle detection Using Adaptative Color Segmentation and Color Stereo Homograpy. Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference. Pittsburgh. 2001. p. 705-710.
7. Low T, Wyeth G. Obstacle Detection using Optical Flow. Australasian Conference on Robotics and Automation. Sydney; 2004.
8. Zhu ZX, Guangyou CS, Lin X. Dynamic Obstacle Detection through Cooperation of Purposive Visual Modules of Color , Stereo and Motion. Proceedings of the 1994 IEEE International Conference on Robotics and Automation. San Diego. 1994. p. 1916-1921.
9. Zhang YX, Xiangyang LH, Dai Y. Two-Stage Obstacle Detection Based on Stereo Vision in Unstructured Environment. 2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics. Beijing. 2014. p. 168-172.
10. Zhu Z, Lin X, Shi D, Xu G. A Single Camera Stereo System for Obstacle Detection. 4th Int. Conf. Information Systems Analysis and Synthesis, Proceedings. 1998. p. 230-237.
11. Bhoite A, Beke N, Nanduri S, Duffy T, Torres M. Advanced Situational Awareness And Obstacle Detection Using A Monocular Camera. IEEE (ed.), Image Processing Workshop (WNYIPW). Rochester. 2010. p. 30-33.
12. Einhorn E, Schroter CH, Gross HM. Can'T Take My Eye Off You : Attention-Driven Monocular Obstacle Detection And 3d Mapping. IEEE/RSJ International Conference On Intelligent Robots And Systems (Iros). Taipei. 2010. p. 816-821.
13. Yamaguchi K, Kato T, Ninomiya T. Moving Obstacle Detection Using Monocular Vision. Intelligent Vehicles Symposium. Tokyo. 2006. p. 288-293.
14. Lalonde JR. Monocular Obstacle Detection For Moving Vehicles. MSc Thesis. University of Ottawa; 2011. Disponible en: https://www.researchgate.net/publication/267202590_Monocular_Obstacle_Detection_for_Moving_Vehicles
15. Lalonde J, Laganriere R, Martel L. Single-View Obstacle Detection For Smart Back-Up Camera Systems. 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. Providence. 2012. p. 1-8.

16. Zhang F, Gourban R, Straznicky P. Obstacle Detection For Low Flying UAS Using Monocular Camera. Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International. Graz. 2012. p. 2133-2137.
17. Ulrich I, Nourbakhsh I. Appearance-Based Obstacle Detection with Monocular Color Vision. Proceedings of the AAAI National Conference on Artificial Intelligence. Austin. 2000. p. 866-871.
18. Szeliski R. Computer Vision: Algorithms and Applications. Washington: Springer; 2010.
19. Pantilie CD, Bota S, Haller I, Nedevschi S. Real-time Obstacle Detection Using Dense Stereo Vision and Dense Optical Flow. IEEE International Conference on Intelligent Computer Communication and Processing (ICCP). Cluj-Napoca. 2010;1(1):191-196.
20. Sarcinelli-Filho M, Schneebeli HJA, Cadeira EM, Silva Bm. Optical flow-based reactive navigation of a mobile robot. Revista Controle & Automação. 2007;18(3):267-277.
21. Soria CM, Carelli R, Sarnicelli-Filho M. Using Panoramic Images and Optical Flow to Avoid Obstacles in Mobile Robot Navigation. IEEE International Symposium on Industrial Electronics (ISIE). Montreal. 2006.
22. Grunewald M, Sitte J. A Resource-Efficient Approach to Obstacle Avoidance via Optical Flow. Systems and Circuit Technology. Proceedings of the 5th International Heinz Nixdorf Symposium: Autonomous Minirobots for Research and Edutainment (AMIRE). Heinz. 2001;1(1):205-214.
23. Guzel MS, Serdar BR. Vision Based Obstacle Avoidance Techniques. In: Topalov A. Recent Advances in Mobile Robotics. Rijeka: InTech; 2011. p. 83–108.
24. Talbi EG. Metaheuristics: from design to implementation. New Jersey: John Wiley & Sons; 2009.
25. Doerner KF, Gendreau M, Greistorfer P, Gujahr WJ, Richard F, Reinmann M. Metaheuristics: Progress in Complex Systems Optimization. Vienna, Montreal and Graz: Springer Science + Bussiness Media; 2007.
26. Bolufé-Rholer A, Chen S. Minimum Population Search – A Scalable Metaheuristic for Multi-Modal Problems. Revista Investigación Operacional. 2015;36(1):85-95.

AUTORES

Jorge Silvio Delgado Morales, Ingeniero en Automática, Ing., CUJAE, La Habana, Cuba.
jsmorales@electronica.cujae.edu.cu

Gustavo Viera López, Ingeniero en Telecomunicaciones, Ing., CUJAE, La Habana, Cuba. gvieralopez@gmail.com

Raúl Joaquín Rodríguez Gómez, Ingeniero en Automática, Ing., CUJAE, La Habana, Cuba. rjrg1989@gmail.com

Antonio Serrano Muñoz, Ingeniero en Automática, Ing., CUJAE, La Habana, Cuba. toni.sm91@gmail.com