



## Revista Internacional de Investigación e Innovación Tecnológica

Página principal: [www.riit.com.mx](http://www.riit.com.mx)

---

### Detección de obstáculos en imágenes de campos de maíz

### Object detection in cornfield images for agricultural applications

**Campos-Silvestre, Y<sup>a</sup>, Sossa-Azuela, H<sup>b</sup>, Pajares-Martinsanz, G<sup>a</sup>.**

<sup>a</sup> Departamento de ingeniería de Software e Inteligencia Artificial, Facultad de Informática, Universidad Complutense de Madrid, Madrid 28040, España: [yeraniac@uc.es](mailto:yeraniac@uc.es), [pajares@ucm.es](mailto:pajares@ucm.es)

<sup>b</sup> Instituto Politécnico Nacional-CIC, Av. Juan de Dios Bátiz, S/N, Col. Nueva Industrial Vallejo, México 07738, México: [hsossa@cic.ipn.mx](mailto:hsossa@cic.ipn.mx)

**Innovación tecnológica:** Método para la detección de obstáculos en imágenes de agricultura.

**Área de aplicación industrial:** Tecnología de alimentos.

Recibido: 25 enero 2016.

Aceptado: 20 junio 2016.

### Resumen

El avance de la tecnología ha permitido el diseño y desarrollo de vehículos móviles que son capaces de ejecutar de manera autónoma tareas agrícolas con gran precisión y eficacia. Datos provistos por cámaras y sensores se analizan y procesan con dispositivos computacionales de altas prestaciones para cumplir con las tareas para las que se diseña cada vehículo. En vehículos autónomos uno de los tópicos de mayor relevancia es la navegación segura, es decir, el sistema debe tener la capacidad de detectar elementos inesperados en el área de cultivo con el fin de garantizar la seguridad y continuidad del proceso.

La detección de obstáculos en imágenes de campos de maíz es nuestro tópico de interés. En este trabajo se presenta una metodología que combina técnicas de visión computacional y algoritmos de aprendizaje supervisado para detectar obstáculos en imágenes agrícolas. Las imágenes fueron capturadas con una cámara montada en un tractor en un campo experimental en la estación de investigación “La poveda”, Arganda del Rey, Madrid. Los resultados experimentales muestran que el método propuesto es suficiente para detectar obstáculos dentro del área de cultivo con una precisión de 93%.

**Palabras clave:** Navegación autónoma, detección de obstáculos, aprendizaje supervisado, aplicaciones agrícolas.

## Abstract

The rapid development of technology has been associated with the design of autonomous vehicles in the agricultural industry. Data provided by cameras and sensors is processed with computational algorithms to carry out agricultural tasks with high accuracy and efficiency. In autonomous vehicles, safety becomes a major challenge where unexpected obstacles in the working area must be conveniently addressed to ensure the integrity, security and continuity of the process.

Detection of unexpected obstacles on the path of an autonomous agricultural vehicle is our main aim. In this paper, we propose a strategy for automatic obstacle detection combining image processing techniques and supervised learning algorithms. Images were captured with a single camera mounted onboard a tractor in an experimental field of maize in La Poveda Research Station, Arganda del Rey, Madrid. Experimental results are favorably compared with the ones presented in the state of the art achieving an average accuracy of 93%.

**KeyWords:** Autonomous navigation, agriculture precision, obstacle detection, classifier model, machine learning.

## 1. Introducción

El avance de la tecnología ha permitido la integración de mecanismos “autónomos” en distintos ámbitos de la vida cotidiana. Hoy en día se diseñan y producen robots que facilitan el trabajo y esfuerzo en las diversas actividades que realiza el ser humano. La industria de la agricultura no ha quedado exenta de estas innovaciones, donde vehículos son equipados con dispositivos, cámaras de vídeo y sensores para el desarrollo de actividades agrícolas (plantación, riego, fertilización, eliminación de malas hierbas, inspección, etcétera) con gran precisión y eficiencia, Figura 1.

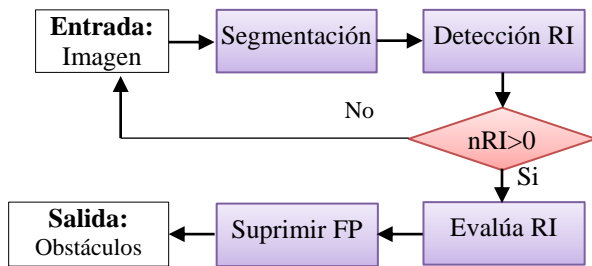


**Figura 1.** Izda. Tractor equipado con cámaras y sensores, el vehículo es parte de la flota del proyecto RHEA [1]. Dcha. Imagen obtenida con la cámara montada en el tractor.

Las imágenes que provén las cámaras, Figura 1. Dcha., se procesan mediante técnicas de visión computacional con el objetivo de discriminar/identificar los diferentes elementos en cada escena.

En imágenes de agricultura hay dos elementos predominantes: plantas verdes y suelo, otros elementos en el área de cultivo (personas, animales, postes, vehículos, etc.) son críticos para la navegación autónoma del vehículo. La detección oportuna de obstáculos permite garantizar la seguridad,

operación y continuidad del proceso, este es nuestro principal tópicico de interés.



**Figura 2.** Proceso para detectar obstáculos en imágenes de agricultura.

En este trabajo se presenta una metodología (esquema de la Figura 2), para detectar obstáculos en imágenes de agricultura. Para alcanzar nuestro objetivo se usaron técnicas de visión computacional en conjunto con modelos de aprendizaje supervisado, la combinación de estas técnicas permite detectar obstáculos en el área próxima al vehículo.

**Base de datos.** Las imágenes procesadas fueron capturadas durante el período de remoción de mala hierba en campos de maíz usando una cámara de alta resolución. La cámara está instalada en la parte superior de un tractor a una altura de 2.20 m y con un ángulo de inclinación de 25° con respecto a la vertical, Figura 1. Izda. La calibración y características del sistema se describen en [2]. Un conjunto de 15000 imágenes componen la base de datos incluyendo imágenes de la demostración final del proyecto RHEA. Las imágenes fueron capturadas con un sensor de tipo SVS4050CFLGEA (SVS-VISTEK, 2015) durante los meses de Abril, Mayo y Junio del 2012 al 2014 en campos experimentales de 1.7 Ha ubicados en La Poveda (Arganda del Rey, Madrid). Las imágenes se tomaron con intervalos de cinco a seis días, es decir, bajo diferentes condiciones de iluminación y

crecimiento de las plantas. La resolución de las imágenes es de  $2336 \times 1752$  píxeles. Cabe destacar que a pesar de que el tractor está equipado con sensores de tipo LIDAR (por sus siglas en inglés, *Light Detection and Ranging*) en esta propuesta se aborda la tarea de detección de obstáculos únicamente con la información provista en las imágenes.

El resto del documento se organiza como sigue. En la Sección 2 se describen trabajos relacionados a la detección de obstáculos en aplicaciones agrícolas. El método propuesto para detectar obstáculos se describe en la Sección 3. Los resultados de la evaluación del sistema se detallan en la Sección 4. El documento concluye en la Sección 5 con una discusión de las lecciones aprendidas y direcciones futuras de investigación.

## 2. Antecedentes

La evasión de obstáculos es un problema clásico en robótica. En la literatura se ha propuesto una amplia variedad de métodos y algoritmos enfocados a resolver esta tarea [3,4]. Hay avances significativos en el tema, sin embargo, mejoras son posibles en distintas aplicaciones. Los métodos actuales presentan deficiencias principalmente en ambientes no controlados, imágenes de exterior, donde la complejidad de las imágenes aumenta significativamente debido a la gran variación en los cambios de iluminación. Este es el caso de las imágenes de agricultura, donde se analizan secuencias de imágenes para detectar y remover malas hierbas, identificar líneas de cultivo, eliminar pestes, y muchas actividades más. En [5,6] se provee una amplia revisión del estado del arte en aplicaciones agrícolas. Cabe mencionar que en la literatura, los autores se enfocan principalmente en el desarrollo y ejecución de las tareas para las que se diseñan los vehículos autónomos,

pocas propuestas incluyen sistemas para detectar y evadir obstáculos dentro del área de cultivo. Enseguida se presenta un resumen de métodos recientemente publicados es este contexto.

Reina y Milella, presentan un método de auto aprendizaje para detectar obstáculos mediante un sistema de visión estéreo trinocular [7]. El algoritmo funciona bajo el supuesto de que al inicio del proceso el área está libre de obstáculos. En nuestra propuesta los obstáculos pueden estar presentes en cualquier momento.

Bibery et al. [8], usan la información provista por sensores LIDAR para evadir obstáculos. Los tractores en la flota del proyecto RHEA están equipados con sensores de este tipo para detectar obstáculos [9]. Los sistemas de detección basados en estos sensores han sido eficaces; sin embargo, resultados en investigaciones recientes muestran que la combinación de sensores LIDAR con técnicas de visión por computadora tienen una mejora significativa [10,11]. En este trabajo, la detección de obstáculos se realiza sobre el análisis de imágenes mediante el uso de técnicas de visión computacional, es decir, no se considera la información provista por los sensores.

Ross et al. [12] presentan un algoritmo para detectar obstáculos estudiando las propiedades de color de imágenes estéreo. El detector presenta fallas cuando el color de los obstáculos es similar al entorno natural. Para abordar este problema aquí se incorpora información de la textura de la imagen. La textura es también considerada en [13] para detectar objetos en un invernadero en datos

provistos por un dispositivo Kinect.

La diferencia de altura y pendiente entre distintos puntos tridimensionales es propuesto en [14] para detectar obstáculos El método presenta buenos resultados a un alto costo computacional. Con la finalidad de incrementar la eficiencia en tiempo real del algoritmo presentado en [14], Mendes et al. [15] presentan una versión del algoritmo con programación paralela. La nube de puntos se obtiene mediante visión estéreo y un sensor RGB-D.

En 2005, Wei et al. [16] presentan un sistema para detectar personas en campos de agricultura mediante el uso de imágenes obtenidas con un sistema de visión estéreo. En 2011 se presenta un algoritmo para detectar troncos de árboles de olivo [17]. La detección se lleva a cabo mediante el uso de sensores y un sistema de visión monocular. El tractor navega en el área de cultivo usando como referencia los troncos detectados. En el mismo contexto, Fleischmann y Berns [18] presentan un método de detección de obstáculos para la construcción de mapas usando imágenes obtenidas con un sistema de visión estéreo. En nuestros resultados, los troncos de árboles se detectan exitosamente, lo cual sugiere que el método aquí propuesto se puede usar para construir mapas de navegación.

A diferencia de los trabajos previamente descritos donde la detección se basa en información provista por sensores y múltiples cámaras, aquí se presenta una estrategia para detectar obstáculos a partir del análisis de imágenes provistas por una cámara.

### 3. Propuesta metodológica

El proceso para detectar obstáculos en una imagen de agricultura se resume en el diagrama de la Figura 2. Primero, la imagen se divide en dos conjuntos complementarios de píxeles, para obtener una imagen binaria, 1- frente y 0- fondo. El frente contiene píxeles que son posibles objetos, mientras que el fondo contiene los elementos predominantes en la escena, suelo y plantas verdes. La imagen binaria se procesa con un método de etiquetado para obtener regiones de interés (*RI*). Si el número de regiones de interés (*nRI*) es igual a cero el proceso concluye, en caso contrario, las *nRI* se evalúan con un modelo de clasificación para remover los falsos positivos (FP), regiones en la imagen que no son obstáculos. Detalles de la implementación se presentan a continuación.

**3.1. Segmentación.** Después de estudiar diversos espacios de color encontramos que el modelo  $L^*a^*b^*$  es el más apropiado para estudiar imágenes de agricultura. Este sistema consta de una componente de luminosidad ( $L^*$ ) y dos de color ( $a^*$  que va de verde a rojo y el  $b^*$  que va de azul a amarillo). Las componentes se obtienen con las ecuaciones 1, 2 y 3, donde;  $X$ ,  $Y$  y  $Z$  son los valores de rojo, verde y azul mientras que  $X_w$ ,  $Y_w$  and  $Z_w$  son las constantes del color blanco de referencia [19].

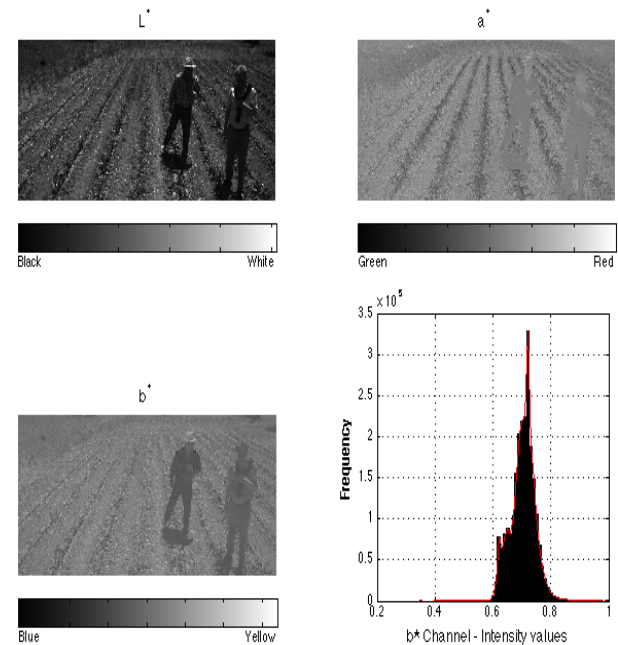
$$L^* = 116 \times h(Y/Y_w) - 16 \quad 1)$$

$$a^* = 500[h(X/X_w) - h(Y/Y_w)] \quad 2)$$

$$b^* = 200[h(Y/Y_w) - h(Z/Z_w)] \quad 3)$$

$$h(q) = \begin{cases} \sqrt[3]{q} & q > 0.008856 \\ 7.787q + \frac{16}{116} & q \leq 0.008856 \end{cases}$$

En la Figura 3 se muestran los canales en espacio  $L^*a^*b^*$  de la imagen a color en la Figura 1.



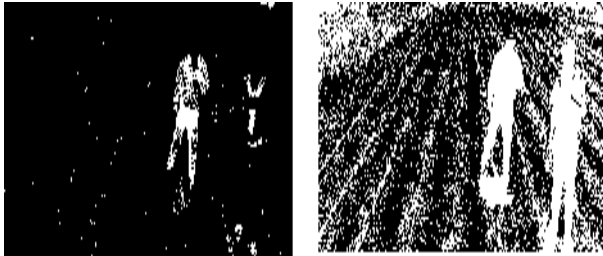
**Figura 3.** Componentes en espacio  $L^*a^*b^*$  de la imagen en la Figura 1. El histograma se obtiene de los valores de intensidad de la componente  $b^*$ .

Del análisis de los histogramas de intensidad en los distintos canales se observó que el contraste entre el fondo y el frente es más notable en la componente  $b^*$ . Más aún, el histograma es unimodal cuando el área está libre de obstáculos, esto no sucede cuando hay elementos ajenos al ambiente, en este caso el histograma presenta perturbaciones como se observa en la Figura 3. Esta característica es la base del algoritmo de segmentación, el proceso se resume en el Algoritmo 1 (ver tabla 1).

**Tabla 1.** Algoritmo 1 (segmentación de imágenes de agricultura).

<b>Entrada:</b>	Imagen a color $I_{rgb}(x, y, z)$ .
<b>Salida:</b>	Imagen binaria $I_{bin}(x, y)$ .
i) $I_{rgb} \rightarrow I_b$ : Evaluar la ecuación 3 para obtener la componente b*.	
ii) De los valores de intensidad en $I_b$ calcular,	
<ul style="list-style-type: none"> <li>Histograma de frecuencias (<math>h</math>) y ajustar una curva a los elementos en <math>h</math>. Obtener máximos y mínimos. Almacenar el mínimo anterior al máximo global (<math>\tau_1</math>).</li> <li>Media (<math>\mu</math>), desviación estándar (<math>\sigma</math>) y asimetría (<math>s</math>) [20].</li> </ul>	
iii) $thl = \min(\tau_1, \mu + 2\sigma)$ . Sí, $s < 0$ entonces, $thl = \mu - 2\sigma$ .	
iv) $I_{bin}(x, y) = \begin{cases} 1, & I_b(x, y) < thl \\ 0, & \text{otro caso} \end{cases}$	

En la Figura 4 se muestra la segmentación de imagen en la Figura 1 mediante el Algoritmo 1, imagen de la izquierda. El resultado de la derecha se obtiene usando Otsu [21], técnica comúnmente usada en aplicaciones agrícolas.



**Figura 4.** Izda. Segmentación obtenida con el Algoritmo 1 ( $I_{bin}$ ). Dcha. Binarización usando Otsu.

Como se observa en la Figura 4, el Algoritmo 1 tiene deficiencias cuando el color de los obstáculos es similar al entorno natural, este problema también es reportado en [12]. Con el fin de mejorar la segmentación incorporamos información extraída de la textura de la imagen como sigue: a) a cada pixel se le asigna la diferencia entre el valor máximo menos el valor mínimo de los valores de intensidad

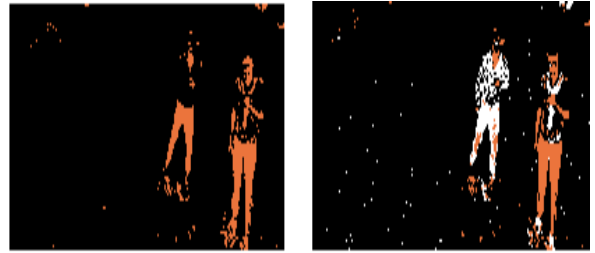
contenidos en una vecindad, dimensión  $5 \times 5$ , centrada en el pixel del interés, el resultado se almacena en  $I_t$ . b) Cálculo de media ( $\mu_t$ ) y desviación estándar ( $\sigma_t$ ) de los elementos en  $I_t$ . c) evaluar la expresión 4 para obtener una imagen binaria  $I_{bin}^*$ .

$$I_{bin}^*(x, y) = \begin{cases} 1, & I_t(x, y) < (\mu_t - 2\sigma_t) \\ 0, & \text{otro caso} \end{cases} \quad 4)$$

La segmentación final se obtiene unificando la información en  $I_{bin}$  con  $I_{bin}^*$  mediante la expresión 5.

$$I(x, y) = \begin{cases} 1, & I_{bin}(x, y) + I_{bin}^*(x, y) \geq 1 \\ 0, & \text{otro caso} \end{cases} \quad 5)$$

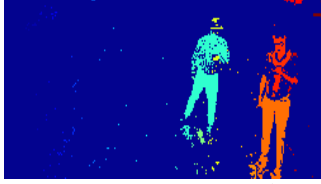
Continuando con el ejemplo de la Figura 1, la imagen binaria de la Figura 5, se obtiene con la expresión 4. La segmentación final se muestra en la misma figura, imagen de la derecha.



**Figura 5.** Izda. Resultado obtenido del análisis de textura,  $I_{bin}^*$ . Dcha. Segmentación final al unir el resultado de la derecha con el de la primer imagen de la Figura 4.

### 3.2. Detección de las regiones de interés.

De la segmentación obtenida en la sección previa, pixeles vecinos se agrupan y forman una componente conectada [22]. Cada componente representa una región de interés (RI). En la Figura 6 se ilustra este proceso, donde cada color representa una RI.



**Figura 6.** Componentes conectadas de imagen binaria en Figura 5 Dcha. Cada color representa una región de interés.

Las  $n$ - regiones de interés ( $nRI$ ) se evalúan en la siguiente etapa para determinar cuál de ellas son objetos y remover las regiones que pertenecen al fondo – falsos positivos (FP).

**3.3. Modelo de clasificación.** Se construye para discriminar entre tres tipos de elementos en una imagen: plantas verdes, suelo y obstáculos. La función de clasificación se modela mediante máquinas con vectores soporte (SVM, *support vector machine*, por sus siglas en inglés) y usando un núcleo de base radial [23]. Las características que se extraen de cada clase y que se usan para entrenar el modelo de clasificación juegan un papel importante. En este artículo probamos distintos patrones y encontramos que para caracterizar imágenes de agricultura los llamados descriptores robustos acelerados [24] (SURF, *Speeded-Up Robust Features*, por sus siglas en inglés) son los más apropiados.

El proceso consta de una etapa de entrenamiento y una etapa de prueba. En la primera etapa se ajustan los parámetros del modelo de clasificación mientras que en la segunda se evalúa el desempeño del modelo con datos nunca antes vistos [25].

**Selección de datos.** De la base de datos descrita en la Sección 1, un conjunto de 50 imágenes fueron seleccionadas. Estas imágenes se usan exclusivamente para diseñar el clasificador. Las imágenes

contienen elementos de las tres clases de interés (plantas verdes, suelo y obstáculos). De cada imagen se hace una selección manual de 15 sub-imágenes/regiones (5 de plantas, 5 de suelo y 5 de objetos), cada región contiene elementos de una clase. En total se tienen 750 regiones, 250 para cada clase. Identificamos a este conjunto de regiones como  $A$ .

Cada elemento en  $A$  se procesa como sigue: 20 píxeles se seleccionan aleatoriamente y se calcula su correspondiente descriptor SURF. Esta información se almacena en pares de la forma  $(v, l)$ , donde  $v \in \mathcal{R}^{128}$  es el descriptor SURF y  $l \in \{p, s, o\}$  la etiqueta asociada a la región a la que pertenece el pixel: plantas verdes ( $p$ ), suelo ( $s$ ) y obstáculo ( $o$ ).

Del proceso anterior se obtiene un conjunto  $\Theta = \{(v_1, l_1), \dots, (v_n, l_n)\}$ , con  $n = 15000$  elementos. El conjunto  $\Theta$  se divide en dos conjuntos complementarios:

$$\Theta_1 = \{(v_1, l_1), \dots, (v_u, l_u)\}, \quad u = 10000$$

$$\Theta_2 = \{(v_1, l_1), \dots, (v_v, l_v)\}, \quad v = 5000$$

El conjunto  $\Theta_1$  se utiliza para estimar los parámetros del modelo mientras que  $\Theta_2$  se usa para medir el rendimiento del mismo. La separación del conjunto de prueba en dos subconjuntos evita sesgos en el desempeño del clasificador. Hay métodos para determinar cuál es la mejor manera de dividir el conjunto de datos, en este trabajo aplicamos el método de validación cruzada [25].

**Etapas de entrenamiento.** Los elementos en  $\Theta_1$  se utilizan para obtener el modelo de

clasificación usando SVM. Hay parámetros que deben estimarse: el parámetro de corrección de los vectores soporte ( $C$ ) y el parámetro de la función de base radial ( $\Gamma$ ), para esta tarea usamos la herramienta provista por Chang y autores, en [26] se describe detalladamente el proceso para obtener un modelo de clasificación ( $M$ ). El lector interesado en SVM puede consultar la referencia citada.

**Etapa de prueba:** Con el conjunto  $\Theta_2$  se evalúa el rendimiento del modelo  $M$ . Los descriptores de  $\Theta_2$  se evalúan con  $M$ , el modelo regresa un conjunto de etiquetas  $L^*$ , una asociada a cada descriptor en  $\Theta_2$ ;

$$M(\{v_1, \dots, v_v\}) = \{l'_1, \dots, l'_v\} = L^*$$

Las etiquetas estimadas se comparan con las etiquetas reales  $\{l_1, \dots, l_v\}$  y se calcula una matriz de confusión, Tabla 2.

**Tabla 2.** Matriz de confusión.

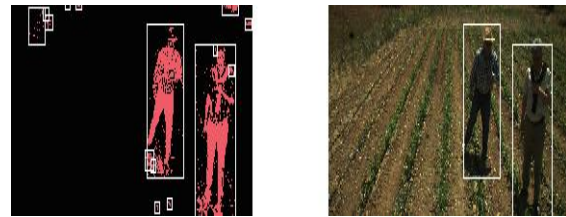
Clases	Estimación		
	Planta	Suelo	Obstáculo
Planta	$P_p$	$P_s$	$P_o$
Suelo	$S_p$	$S_s$	$S_o$
Obstáculo	$O_p$	$O_s$	$O_o$

Los elementos de la matriz representan el porcentaje de patrones correcta e incorrectamente clasificados. Por ejemplo;  $S_p$  representa el porcentaje de elementos clasificados como ‘planta’ siendo que pertenecen a la clase ‘suelo’. Un buen clasificador tiene una matriz de confusión con una diagonal principal dominante, es decir, valores de  $P_p, S_s$  y  $O_o$  cercanos a 100. El porcentaje promedio del clasificador se calcula como:  $(P_p + S_s + O_o)/3$ .

### 3.4 Evaluación de las regiones de interés.

El modelo descrito previamente se usa para eliminar regiones de interés que son plantas, suelo o una mezcla de ellos. La evaluación de una  $RI$  se hace como sigue.

Sea  $P = \{(x_1, y_1), \dots, (x_m, y_m)\}$  el conjunto de píxeles que componen la  $RI$ ,  $m$  es el número de píxeles en la región. Para cada pixel en  $P$  se calcula su descriptor SURF;  $\Phi = \{v_1, \dots, v_m\}$ . El conjunto  $\Phi$  se evalúa en el modelo  $M$  para obtener las etiquetas asociadas a cada elemento,  $M(\Phi) = L'$ . La clase predominante en  $L'$  es la que se asigna a la  $RI$ . Si la clase es  $p$  o  $s$  los píxeles en la  $RI$  se eliminan de la imagen binaria. En la Figura 6 (b) se muestra una imagen binaria y el resultado después de evaluar las regiones de interés con el modelo de clasificación descrito en esta sección.



**Figura 6 (b).** Izda: En rectángulos se enmarca las regiones de interés en la imagen binaria. Dcha: componentes que son objetos.

## 4. Resultados

La detección de obstáculos en la escena depende de la segmentación de la imagen. El Algoritmo 1 es de fácil implementación y los resultados experimentales muestran que es suficiente para detectar regiones de interés con gran potencial de ser obstáculos. En la Figura 7 se muestran algunos resultados de segmentación obtenidos con el proceso descrito en la Sección 3.1. La segmentación se resalta en color rojo. Como se puede apreciar, los objetos ajenos al campo de cultivo (un tractor y dos personas) son



capturados con el algoritmo propuesto. Una buena segmentación reduce considerablemente el número de RI a ser evaluadas en la siguiente etapa, preservando las regiones de interés.



Figura 7. Segmentación de imágenes de agricultura.

Ahora bien, el diseño del clasificador requiere estimar los valores de los parámetros  $C$  y  $\Gamma$ , Sección 3.2. Estos valores se obtuvieron mediante una búsqueda exhaustiva. Se probaron distintas combinaciones de la forma  $(c, \gamma)$ , con  $0 < c, \gamma \leq 20$ , en intervalos de 0.1. Los mejores resultados se obtuvieron con el par  $(c, \gamma) = (1.6, 12.6)$ . En la Tabla 3 se presenta la matriz de confusión. La precisión promedio alcanzada es de 86%, el clasificador presenta problemas para discernir entre plantas y suelo, sin embargo, nuestro interés son los objetos y en este sentido se alcanza un 93.3%. Este valor es suficiente para determinar que regiones son obstáculos y cuáles no.

Tabla 3. Matriz de confusión. Resultados en porcentaje, la precisión promedio es 86%.

Clases		Estimación		
		Planta	Suelo	Obstáculo
Real	Planta	84.7	12.7	2.6
	Suelo	18.7	80	1.3
	Obstáculo	4.7	2	93.3

Una vez que se ha hecho la selección de parámetros, el modelo se usa para detectar obstáculos en las imágenes de la base de datos, en la Figura 8a y Figura 8b se muestran resultados visuales de la detección. Durante el análisis sólo se estudia la región más próxima al tractor, zona limitada con un rectángulo negro. Los obstáculos detectados se encierran en rectángulos de colores. Cada color representa la proximidad del obstáculo al tractor; en azul los más distantes, en amarillo a una distancia intermedia y en rojo los más próximos (figura 8 a y b).

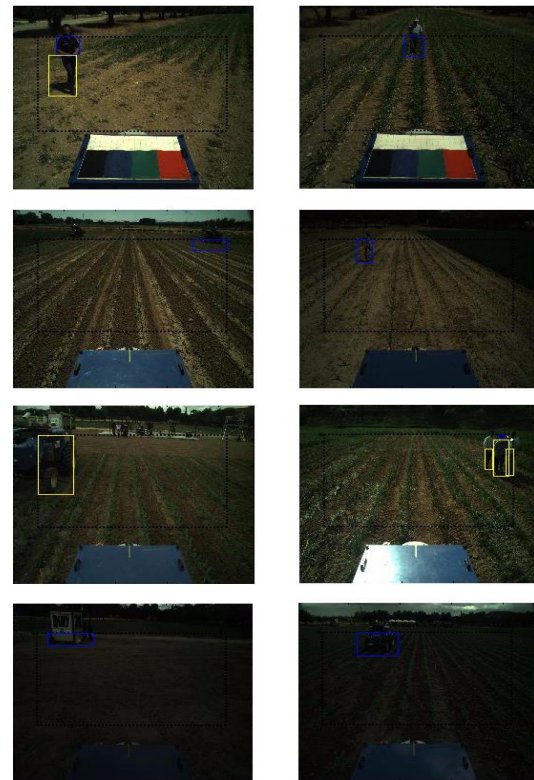
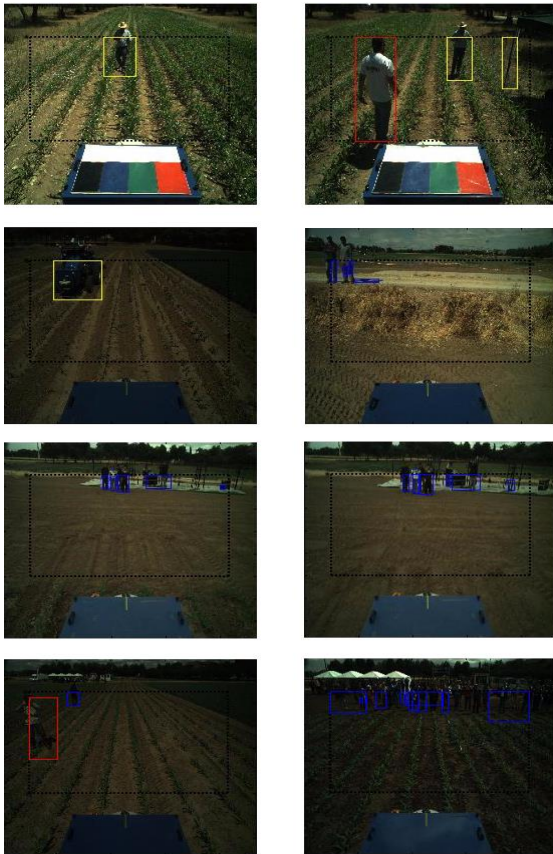


Figura 8a. Detección de obstáculos en imágenes de agricultura. En las imágenes se observan cambios en la iluminación y distintos estados de crecimiento del maíz.



**Figura 8b.** Detección de obstáculos en imágenes de agricultura.

Con el propósito de mejorar el desempeño del modelo obtenido con SVM se realizaron pruebas usando otros modelos de clasificación: *K-vecinos más cercanos* [27], *perceptrón multicapa* [28], y *clasificador de Bayes* [29] (KNN, MLP y NB respectivamente, por sus siglas en inglés). Se entrenaron 3 diferentes modelos: Modelo<sub>1</sub>, Modelo<sub>2</sub> y Modelo<sub>3</sub>. El primero para discriminar entre plantas y obstáculos, el segundo para discriminar entre suelo y obstáculos. Finalmente, el tercero para discriminar entre plantas, suelo y obstáculos. La precisión promedio alcanzada en cada caso se exhibe en la Tabla 4.

**Tabla 4.** Desempeño de 3 clasificadores (columnas) usando diferentes técnicas de clasificación (renglones). Precisión promedio en porcentaje (%).

	Modelo <sub>1</sub>	Modelo <sub>2</sub>	Modelo <sub>3</sub>
<b>KNN</b>	92.0	89.4	71.6
<b>MLP</b>	49.4	49.2	51.7
<b>NB</b>	89.5	89.8	71.9
<b>SVM</b>	94.4	93.5	86.0

De la Tabla 4 se puede ver que el clasificador SVM tiene el mejor desempeño en todos los casos. Note que el valor del quinto renglón y la cuarta columna (86%) corresponde a la matriz de confusión de la Tabla 3. En cuanto a los parámetros para cada modelo, la búsqueda de los valores óptimos se realizó como se describe a continuación.

El valor del parámetro  $K$  (número de vecinos más próximo) del modelo KNN se determinó después de probar distintos valores en el intervalo de 1 a 100. El mejor valor se obtuvo con  $K = 3$  en los tres modelos. De la misma manera, el número de capas  $h$  en el modelo MLP se obtuvo después de probar con valores en el rango del 1 y 10. En este caso, el valor más alto obtenido se alcanzó con  $h = 1$ .

Para concluir esta sección, la implementación del algoritmo se realizó en código Matlab 2013a con un equipo Intel Core 2 CPU, 3 GHz, y 4 GB RAM. Para el diseño del clasificador usamos la librería LIBSVM [27].

## 5. Discusión y conclusiones

Un buen sistema de detección es de especial relevancia para evitar daños derivados del uso de vehículos autónomos, como ha sido demostrado en el proyecto RHEA. El costo de un mal sistema de detección puede generar grandes daños y en algunos casos, daños irreversibles.

En este trabajo se presenta un método para detectar obstáculos en campos de agricultura. El proceso incluye un algoritmo de segmentación y un proceso de clasificación. El método puede aplicarse a distintos escenarios en aplicaciones de agricultura [4].

Mejoras a la propuesta aquí presentada son posibles. Primero, las sombras aún son difíciles de tratar. Se deben implementar algoritmos para tratar con estos elementos. Segundo, como se ha mencionado con anterioridad, el problema se resuelve únicamente del análisis de imágenes. Se puede integrar la información provista por el sensor LIDAR con la finalidad de incrementar la eficacia de nuestra propuesta. Por último, como se sugiere en [24] y [25], la detección de obstáculos se puede usar para construir mapas del ambiente y trazar trayectorias para garantizar la navegación autónoma del tractor.

## Agradecimientos

El presente trabajo pertenece al VII Programa Marco de la Unión Europea de Investigación y Desarrollo tecnológico (FP7/2007-2013). Acuerdo de subvención No. 245986. El primer autor agradece también al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el soporte económico para la continuidad a los estudios de posgrado. H. Sossa agradece a la SIP-IPN

en el marco del proyecto 20161126 y al CONACyT en el marco de los proyectos 155014 (Investigación básica) y 65 (Fronteras de la Ciencia) por el apoyo económico brindado para el desarrollo de la presente investigación.

## Referencias

- [1] RHEA (2015). Robot fleets for highly effective agriculture and forestry management. <http://www.rhea-project.eu/>.
- [2] Romeo, J., Guerrero, J. M., Montalvo, M., Emmi, L., Guijarro, M., Gonzalez-de-Santos, P., & Pajares, G. (2013). Camera sensor arrangement for crop/weed detection accuracy in agronomic images. *Sensors*, 13(4), 4348-4366. DOI: 10.3390/s130404348.
- [3] Panchal, P., Prajapati, G., Patel, S., Shah, H., & Nasriwala, J. A (2015). A Review on Object Detection and Tracking Methods. *International Journal for Research in Emerging Science and Technology*, 2(1), 7-12. DOI: 10.15680/ijrccce.2015.0307078.
- [4] Kinana, B. & Mehul, C. (2015). A comprehensive survey on moving object segmentation methods. *International Journal for Technological Research in Engineering*, 2(8), 1394-1398.
- [5] Mousazadeh, H. (2013). A technical review on navigation systems of agricultural autonomous off-road vehicles. *Journal of Terramechanics*, 50(3), 211-23. DOI: 10.1016/j.jterra.2013.03.004.
- [6] Bochtis, D. D., Sørensen, C. G., & Busato, P. (2014). Advances in agricultural machinery management: A review. *Biosystems Engineering*, 126, 69-81.
- [7] Reina, G., & Milella, A. (2012). Towards Autonomous Agriculture: Automatic Ground Detection Using Trinocular Stereovision. *Sensors (Basel, Switzerland)*, 12(9), 12405–12423. DOI: 10.3390/s120912405.
- [8] Biber, P., Weiss, U., Dorna, M., & Albert, A. (2012). Navigation system of the autonomous agricultural robot Bonirob. In *Workshop on Agricultural*

- Robotics: Enabling Safe, Efficient, and Affordable Robots for Food Production (Collocated with IROS 2012), Vilamoura, Portugal.
- [9] Garrido, M., Pérez-Ruiz, M., Valero, C., Gliever, C.J., Hanson, B.D., & Slaughter, D.C. (2014). Active Optical Sensors for Tree Stem Detection and Classification in Nurseries. *Sensors*, 4, 10783-10803. DOI: 10.3390/s140610783.
- [10] Douillard, B., Fox, D., & Ramos, F. (2007, October). A spatio-temporal probabilistic model for multi-sensor object recognition. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on* (pp. 2402-2408). IEEE. DOI: 10.1109/IROS.2007.4399537.
- [11] Cristiano Premebida, Oswaldo Ludwig, and Urbano Nunes. Lidar and vision-based pedestrian detection system. *Journal of Field Robotics*, 26(9), 2009.
- [12] Ross, P., English, A., Ball, D., Upcroft, B., Wyeth, G., & Corke, P. (2014, May). Novelty-based visual obstacle detection in agriculture. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (pp. 1699-1705). IEEE. DOI: 10.1109/ICRA.2014.6907080.
- [13] Nissimov, S., Goldberger, J., & Alchanatis, V. (2015). Obstacle detection in a greenhouse environment using the Kinect sensor. *Computers and Electronics in Agriculture*, 113, 104-115. DOI: 10.1016/j.compag.2015.02.001.
- [14] Talukder, A., Manduchi, R., Rankin, A., & Matthies, L. (2002). Fast and reliable obstacle detection and segmentation for cross-country navigation. In *Intelligent Vehicle Symposium*, 2, 610-618. DOI: 10.1109/IVS.2002.1188019.
- [15] Mendes, C. C. T., Osório, F. S., & Wolf, D. F. (2013). An efficient obstacle detection approach for organized point clouds. In *Intelligent Vehicles Symposium (IV)*, 1203-1208. DOI: 10.1109/IVS.2013.6629630.
- [16] Wei, J., Rovira-Mas, F., Reid, J.F., Han, S. (2005). *Obstacle detection using stereo vision to enhance safety autonomous machines*. *Transactions of the ASABE*, 48(6): 2389-2397. DOI: 10.13031/2013.20078.
- [17] Cheein, F. A., Steiner, G., Paina, G. P., & Carelli, R. (2011). *Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection*. *Computers and Electronics in Agriculture*, 78(2), 195-207. DOI: 10.1016/j.compag.2011.07.007.
- [18] Fleischmann, P., & Berns, K. *A Stereo Vision Based Obstacle Detection System for Agricultural Applications*. *Proceedings of Field and Service Robotics (FSR)*, 2015. Toronto, Canada. DOI: 10.1007/978-3-319-27702-8\_15.
- [19] Rafael C. Gonzalez and Richard E. Woods. 2006. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- [20] *Estadística Matemática con Aplicaciones*. Dennis D. Wackerly, William Mendenhall, Richard L. Scheaffer. Cengage Learning, 2009.
- [21] Nobuyuki Otsu. *A Threshold Selection Method from Gray-Level Histograms*, in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, Jan. 1979. DOI: 10.1109/TSMC.1979.4310076.
- [22] Dillencourt, M. B., Samet, H., & Tamminen, M. (1992). *A general approach to connected-component labeling for arbitrary image representations*. *Journal of the ACM (JACM)*, 39(2), 253-280. DOI: 10.1145/128749.128750.
- [23] Scholkopf B, Somola A. *Learning wit kernels*. The MIT Press (EUA), 2002.
- [24] Bay, H., Tuytelaars, T., & Van Gool, L. (2006). *Surf: Speeded up robust features*. In *Computer vision–ECCV2006* (pp. 404-417). Springer Berlin Heidelberg. DOI: 10.1016/j.cviu.2007.09.014.
- [25] Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification* (2th ed.). John Wiley and Sons.
- [26] Chang, C. C., & Lin, C. J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), 27. DOI: 10.1145/1961189.1961199.
- [27] García, V., Debreuve, E., & Barlaud, M. (2008). *Fast k-nearest neighbor search using GPU*. *Computer Vision and Pattern Recognition Workshops, CVPRW'08*.pp. 1-6. DOI: 10.1109/CVPRW.2008.4563100.
- [28] Bishop, C. M. (2006). *Pattern recognition and machine learning* (Vol. 4, No. 4, p. 12). New York: springer.
- [29] Mitchell, T. M. (1997). *Machine Learning (1)*. McGraw-Hill, Inc., New York, NY, USA.