

# Visión Artificial Aplicada al Control de Malezas en Tiempo Real

Marco Miretti<sup>†</sup>, Emanuel Bernardi, Hugo Pipino, Gastón Peretti  
Grupo de Investigación y Desarrollo Electrónico (GIDE)  
Universidad Tecnológica Nacional,  
Facultad Regional San Francisco (UTN-FRSFCO)  
San Francisco, Córdoba, Argentina  
<sup>†</sup>marco.miretti@gmail.com

Marcelo Costamagna  
Departamento de Mecánica  
Universidad Tecnológica Nacional,  
Facultad Regional Villa María (UTN-FRVM)  
Villa María, Córdoba, Argentina

**Resumen**—Hoy en día la aplicación selectiva de agroquímicos resulta indispensable en la preservación del medio ambiente. Es por ello que mediante el presente trabajo, el cual es parte de un proyecto de mayor envergadura, se pretende la detección de malezas en barbecho a través del procesamiento de imágenes en tiempo real. Para luego, en base a la información recopilada, comandar la dosificación selectiva de herbicidas. Las pruebas en laboratorio arrojaron resultados favorables respecto a la técnica de detección empleada, aunque expusieron la necesidad de mejora en los tiempos de procesamiento.

**Keywords**—Procesamiento de imágenes, Linux, OpenCV, Intel Galileo, barbecho, herbicidas.

## I. INTRODUCCIÓN

El período de barbecho, es aquel en el que cierta porción de tierra cultivable se deja sin sembrar con el objetivo de que la misma recupere la materia orgánica y humedad necesarias para una nueva siembra. Durante éste, es de vital importancia erradicar las malezas remanentes, debido a que mejora la predisposición del suelo al posterior cultivo. Desafortunadamente, el proceso de eliminación de malezas implica la dosificación de herbicidas, cuyos impactos ambientales adversos [1] han conducido a los investigadores a desarrollar técnicas para reducir o limitar su uso [2, 3].

Actualmente, las sistemas de detección de malezas emplean un arreglo de sensores que captan longitudes de onda específicas reflejadas por las mismas [4], lo que requiere de sensores especiales para dicha tarea. Otras formas de abordar la detección de malezas utilizan visión artificial (también conocida como visión computarizada) [5, 6], o incluso, un híbrido entre éstas y redes neuronales [7]. La implementación de estas técnicas precisa de un alto poder de cómputo, lo que las transforma en poco atractivas para su aplicación. Es por ello, que en este trabajo se plantea un método alternativo, donde a partir de herramientas básicas del procesamiento de imágenes se identifican formas, texturas y colores, que permiten determinar la presencia o ausencia de maleza en tiempo real sobre una plataforma de desarrollo de complejidad moderada.

Las técnicas de procesamiento de imágenes empleadas en este trabajo se basan en el análisis de histogramas, en estos, se representa la frecuencia con la que cierta característica de la imagen ocurre. Es decir, si analizamos el matiz de una imagen, el eje de las ordenadas nos indicará la proporción de veces que

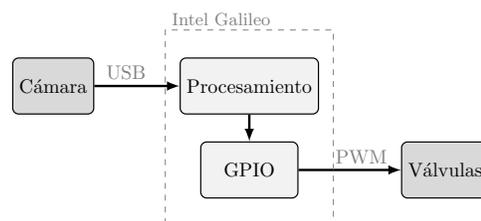


Figura 1. Diagrama general del sistema.

el valor de matiz (situado en el eje de las abscisas) ocurre. A continuación, mediante herramientas de inferencia estadística se modelan los histogramas obtenidos, aproximándolos a distribuciones conocidas y a partir de los parámetros de los mismos se determina la presencia de maleza. Por último, utilizando transformaciones trigonométricas definimos la ubicación de la maleza previamente detectada.

El sistema de dosificación controlada de herbicidas propuesto se bosqueja en la Fig. 1, donde se observa la interconexión entre los distintos periféricos. Esto es, la cámara de adquisición de imágenes se conecta por medio del puerto USB (Universal Serial Bus) y el comando de las válvulas de control se lleva a cabo generando señales PWM (Pulse Width Modulation) a través de las GPIO (General Purpose Input/Output) de la plataforma *Intel Galileo Gen 1*.

## II. MATERIALES Y MÉTODOS

En la implementación del sistema de detección, se consideró necesario el uso de una plataforma que nos permita la captura y procesamiento de imágenes, para la posterior toma de decisiones en consecuencia. Por lo tanto, se optó por la *Intel Galileo Gen 1* que se observa en la Fig. 2. Este dispositivo se encuentra equipado con un CPU *Intel Quark X1000* de 400 MHz, y soporte para conexiones de entrada/salida de gran utilidad en el desarrollo del trabajo. Además, es capaz de ejecutar un sistema operativo *Linux*, específicamente una distribución de *Linux Yocto* [8], que incluye numerosas librerías precargadas para la utilización específica en esta plataforma. Es de destacar, que si bien contiene una gran cantidad de bibliotecas es posible, y fue necesario, instalar otras que nos permitieron implementar el sistema de visión artificial deseado.

Sobre la plataforma se instaló la distribución de *Linux Yocto*, anteriormente mencionada, la cual se descargó desde

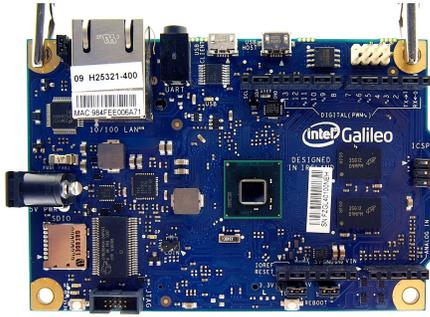


Figura 2. Plataforma *Intel Galileo Gen 1*. De: Wikimedia [9].

el repositorio *Intel Galileo* [10]. Dicha distribución se grabó en un medio físico extraíble Micro SD, donde a través de la interfaz *EFI* (Extensive Firmware Interface), el dispositivo detecta el *GRUB* (GRand Unified Bootloader) de Yocto para iniciar el sistema operativo.

Para la obtención de imágenes se utilizó la cámara *Genius Facecam 2020*, que es soportada por la versión 4.8.17 del controlador *uvcvideo* [11], y utiliza la API (Application Programming Interface) de *Video4Linux 2*, incluida en el sistema Yocto. Ésta se conecta a través del puerto USB y al ser detectada como dispositivo de video, se ubica en el directorio */dev/video0*, que posteriormente invocamos en la captura de imágenes.

En el desarrollo del algoritmo, se utilizó la librería de código abierto para visión computarizada *OpenCV* [12], la cual es de uso gratuito tanto para fines académicos como comerciales (licencia BSD). La misma se encuentra disponible para distintos lenguajes, como ser *C*, *C++*, *Python* y *Java* y tiene soporte en los sistemas operativos *Linux*, *Windows*, *Mac OS*, *IOS* y *Android*. Además, siendo que la librería *OpenCV* fue diseñada para ser computacionalmente eficiente y con una fuerte orientación hacia las aplicaciones en tiempo real, se encontró en ésta una poderosa herramienta para satisfacer las necesidades del proyecto. Por último, en el análisis de los modelos estadísticos propuestos se precisó de la librería *Astropy* [13].

### III. VISIÓN ARTIFICIAL

#### III-A. Adquisición de imágenes

Generalmente, un video se puede definir como una secuencia de imágenes que forman escenas en movimiento. Y la cantidad de cuadros por segundo (FPS, del inglés frames per second) determina la velocidad de la cámara para entregar una nueva imagen. Es de destacar, que el ser humano es capaz de percibir la sensación de movimiento deseada si la tasa de actualización supera los 15 FPS, experimentando una mejor sensación de fluidez por encima de los 48 FPS [14]. En cambio, en este trabajo la tasa de refresco está determinada por la relación de avance del dispositivo de dosificación de herbicidas, considerándose aceptables valores de entre 5 a 20 FPS.

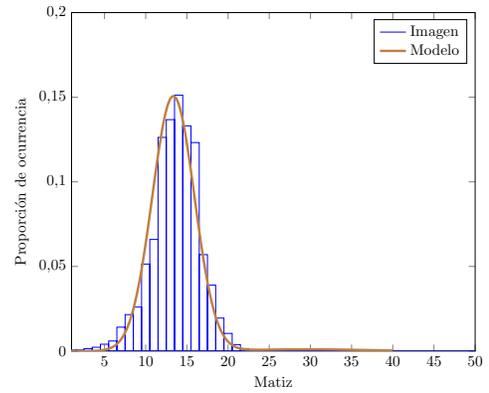


Figura 3. Histograma de matiz de la Fig. 4 y su modelo de ajuste.



Figura 4. Campo durante el período de barbecho libre de malezas.

#### III-B. Transformación del espacio de color

Cada imagen que arriba desde la cámara se encuentra representada en el espacio de color RGB, compuesto por tres canales: rojo, verde y azul (como denotan sus siglas en inglés). En la que cada uno de sus píxeles está compuesto por la intensidad de los tres canales en cuestión, en un rango de 0 a 255 por canal, siendo estos la mínima y máxima intensidad, respectivamente. Investigaciones previas [6], muestran que se pueden obtener mejores resultados en la separación entre el suelo y la maleza si se realiza una transformación hacia el espacio de color HSV, compuesto por tres canales: matiz, saturación e intensidad (como denotan sus siglas en inglés). Donde el rango de valores mínimos y máximos está determinado según el software que los interprete. En el caso de *OpenCV*, el matiz toma valores de  $0^\circ$  a  $180^\circ$ , mientras que la saturación e intensidad toman el rango de 0 a 255.

#### III-C. Análisis estadístico

Una vez realizada la transformación del espacio de color en la imagen, procedimos con la evaluación de los histogramas de la misma. Es por ello, que para introducir al lector en la técnica de clasificación utilizada en la detección, bosquejamos e interpretamos el histograma de matiz, presentado en la Fig. 3, de una imagen en la que se visualiza un campo en período de barbecho libre de malezas, como se muestra en la Fig. 4.

De la observación del histograma de la Fig. 3 se desprende que la distribución de los datos del barbecho se ajusta, con cierta precisión, a una distribución normal o gaussiana desplazada ( $\mu \neq 0$ ). Esta hipótesis se comprobó estadísticamente a partir



Figura 5. Campo afectado por la maleza durante el período de barbecho.

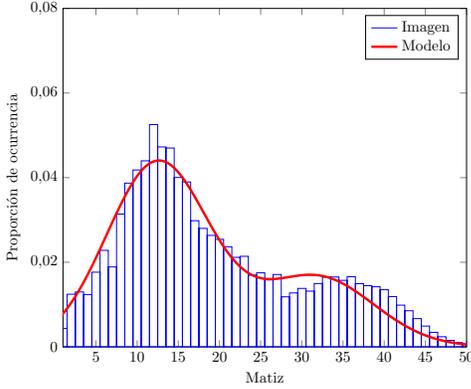


Figura 6. Histograma de matiz de la Fig. 5 y su modelo.

de una prueba de bondad de ajuste, que la confirmó a un nivel de confianza del 99 % ( $p < 0,01$ ). Notar que esta apreciación resulta de utilidad para establecer la precisión y rigurosidad en la selección de los límites de matiz que se emplean en el proceso de detección.

En cambio, cuando en la imagen hay presencia de maleza con matices verdes, como en la Fig. 5, el histograma correspondiente, que vemos en la Fig. 6, se compone de la suma de dos distribuciones normales desplazadas, una correspondiente al tono marrón del suelo y otra al verde de la maleza, siendo la última de menor amplitud. El ajuste de los modelos separados puede apreciarse en la Fig. 7. De esta manera, hemos sido capaces de encontrar una característica de la imagen en la cual hacer hincapié para la correcta separación barbecho-maleza.

A continuación, debemos obtener los parámetros correspondientes a los modelos de ajuste propuestos en el histograma de la Fig. 7, para ello, mediante la técnica de regresión por mínimos cuadrados se computó la amplitud, valor medio y desviación estándar de cada uno de los mismos. Por último, para confirmar los resultados obtenidos en la regresión, se evaluó el coeficiente de determinación  $R^2$  que resultó superior a 0,975 y se mantuvo en ese rango de valores en pruebas posteriores con nuevas imágenes.

### III-D. Segmentación de la imagen

Una vez obtenidos los parámetros del modelo, debemos detectar la presencia de maleza en la imagen, para ello, se exploró el histograma en busca de matices próximos a la media

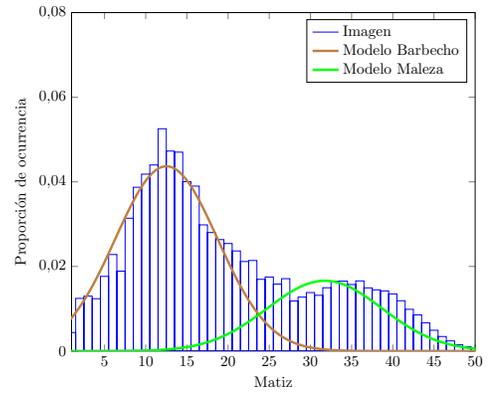


Figura 7. Histograma de matiz de la Fig. 5 y su modelo separado.

de la distribución atribuida a la maleza. Específicamente, establecimos límites de detección superior e inferior. Es importante notar, que hacia la derecha de la distribución de interés se ubican las tonalidades azules, ausentes en nuestro caso, por lo que definimos el límite de detección superior como,

$$L_s := \mu + 3\sigma \quad (1)$$

Es decir, a una distancia de tres desviaciones estándar de la media, pudiendo así abarcar el 99,73 % de la maleza cuyo matiz esperado es mayor a la media (49,865 % del total).

En cambio, la elección del límite inferior conllevó una mayor dificultad, debido a que éste debe distinguir entre los matices que representan la maleza y el resto de las componentes de la imagen con matices a la izquierda de su media. Por lo que se estableció como límite de detección inferior al valor de matiz “ $x$ ” en el que se intersecan las funciones de probabilidad de ambos modelos. Esto es

$$L_i := x \Big|_{f(\text{barbecho})=f(\text{maleza})} \quad (2)$$

De este modo, se buscó una solución de compromiso que ajuste de la mejor forma la probabilidad de error de tipo I (falso positivo) y de tipo II (falso negativo).

Luego de haber definido los límites de detección (1) y (2), utilizamos la regla (3) para crear una máscara de la imagen. Resultando una imagen binaria que contiene los puntos en los que existe maleza.

$$\begin{cases} 1, & L_i \leq x \leq L_s \\ 0, & \text{resto.} \end{cases} \quad (3)$$

Finalmente, debido a la posibilidad de que unos pocos píxeles de la imagen binaria queden aislados, producto de una detección errónea, se utilizó un filtro morfológico de apertura. El cual, elimina este ruido suprimiendo áreas detectadas menores a un par de píxeles cuadrados. A modo de ejemplo, en la Fig. 8 se muestra la imagen binaria resultante del proceso de detección de maleza en la Fig. 5.

### III-E. Transformaciones espaciales

La cámara empleada en la adquisición de imágenes se encuentra montada en un dispositivo, que aparte de absorber vibraciones, la posiciona con un ángulo de 45° por debajo de la horizontal. Esta inclinación a sido adoptada para obtener

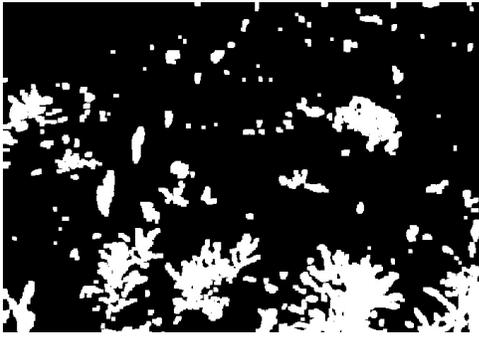


Figura 8. Máscara binaria de la Fig. 5.

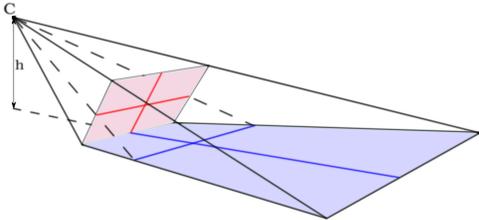


Figura 9. Efecto provocado debido al ángulo de la cámara.

imágenes amplias que nos permitan compensar los tiempos de procesamiento. Desafortunadamente dicha inclinación provoca una deformación en la imagen proyectada, como se muestra en la Fig. 9. Por lo tanto, fue necesario realizar una transformación espacial para corregirla.

Los parámetros a tener en cuenta para realizar esta transformación son: la distancia al suelo, el ángulo respecto de la horizontal y la apertura de la cámara (horizontal y vertical). Es importante destacar, que la apertura del lente se consideró constante y su valor se estimó en laboratorio contrastando mediciones en múltiples imágenes adquiridas con ésta. Además, vale aclarar que la transformación de coordenadas se realiza únicamente sobre la máscara binaria.

Por último, debido a que la cantidad de válvulas a manipular en la dosificación de herbicidas es significativamente menor a la resolución de la imagen, un cambio de resolución en la imagen binaria, mediante una interpolación de área, resulta aceptable. Este cambio, nos permite reducir considerablemente los tiempos de cómputo empleados en la implementación de la transformación espacial. De este modo, y sin pérdida de precisión, cada válvula recibirá una señal de activación conforme a la proporción de maleza en su zona de actuación.

### III-F. Tiempo de procesamiento y tasa de captura

Debido a las acotadas posibilidades de procesamiento con las que cuenta la plataforma utilizada y al objetivo de concebir un método de detección de malezas en tiempo real, el cómputo de los modelos y su posterior análisis para la obtención de los límites de detección no se efectúa en el procesamiento de cada imagen. En cambio, decidimos utilizar límites de detección fijos, los que se determinaron en laboratorio a partir de la evaluación de una muestra de cincuenta imágenes que contemplan distintas densidades de maleza en barbecho. Es de destacar, que en el estudio realizado no se observó una variación significativa de los límites de detección. Por lo



Figura 10. Resultado de la detección de maleza sobre la Fig. 5.

Tabla I. APERTURA POR N° DE VÁLVULA SEGÚN FIG. 10.

1	2	3	4	5	6
0,09	0,15	0,12	0,09	0,18	0,19

tanto, como consecuencia de la simplificación del algoritmo, se obtuvo una reducción en el tiempo de procesamiento promedio.

## IV. RESULTADOS

Como producto de la ejecución del algoritmo propuesto sobre la Fig. 5, se obtuvo la Fig. 10, donde se aprecia la transformación espacial y los sectores de maleza detectados. En la misma, se observa un alto grado de detección y un correcto nivel de separación frente al barbecho. Es de notar, que pese a contemplar una etapa encargada de atenuar el ruido en la imagen, todavía existe un pequeño nivel de éste.

Cabe destacar que la Fig. 10 tiene como único propósito ejemplificar e inspeccionar los resultados obtenidos, debido a que en la práctica el algoritmo de detección no realiza la transformación espacial de la imagen completa, ni la superposición de resultados. En cambio, los datos obtenidos al concluir un ciclo de procesamiento son los de la Tabla I, que se corresponden con la proporción de maleza en cada uno de los canales de aplicación (cada una de las seis válvulas). Dicha proporción indica la apertura normalizada de cada válvula necesarias para la dosificación controlada de herbicidas en ese instante.

Otra característica a evaluar, es el desempeño del algoritmo propuesto en imágenes con ausencia de malezas como la Fig. 4, donde es sencillo identificar el error de tipo I en la detección. El resultado obtenido del procesamiento de la Fig. 4, que se observa en la Fig. 11, solo muestra un par de pequeños puntos detectados. Tal como se esperaba, la proporción de malezas es prácticamente nula y no afecta la proporción de activación.

En cuanto a la capacidad de aplicación en tiempo real del sistema propuesto, los tiempos de detección han mejorado considerablemente luego de aplicar la simplificación propuesta en la sección § III-F, a partir de la cual se logró un tiempo de procesamiento promedio de 0,111 s, calculado en base a la evaluación de múltiples escenarios de detección. Éste se traduce en una tasa de adquisición de imágenes posible de 9 FPS, que se encuentra dentro del intervalo deseado en una primera instancia. Se debe tener en cuenta que los tiempos obtenidos están intrínsecamente ligados a la capacidad de procesamiento de la plataforma utilizada.



Figura 11. Resultado de la detección de maleza sobre la Fig. 4.

## V. CONCLUSIÓN

Mediante este trabajo, hemos presentado una sencilla forma de abordar la detección de malezas en tiempo real en un dispositivo embebido. De esta manera, la implementación sobre un pulverizador autopropulsado estándar resulta factible.

De acuerdo a los resultados, previamente discutidos, podemos afirmar que el grado de detección obtenido es acorde a las necesidades de la aplicación. Si bien, en los ensayos de laboratorio hemos observado ciertos fallos asociados a las condiciones del entorno y al límite impuesto en el análisis estadístico, estos fueron de baja magnitud. Por otro lado, hemos alcanzado una tasa de captura de video que permite el desplazamiento del pulverizador a bajas velocidades. Es por ello que, si bien los logros de este trabajo se muestran alentadores, no son suficientes.

Entonces, en pos de una mejora en el incremento de la capacidad de procesamiento de imágenes, se propone la utilización de un sistema embebido capaz de ejecutar mayor cantidad de instrucciones por segundo y/o el uso de un lenguaje de programación de mayor eficiencia en el manejo de los recursos disponibles.

Por lo tanto, la implementación de las técnicas propuestas en este trabajo, permiten una reducción en la utilización de herbicidas, contribuyendo así al cuidado del medio ambiente.

## AGRADECIMIENTOS

Los autores desean agradecer a **Intel**<sup>®</sup>, por la donación de la plataforma de desarrollo *Intel Galileo Gen 1* en el marco del programa “*Intel Donation University Program*” [15].

## REFERENCIAS

- [1] Amita Chattopadhyay y col. «Influences of environmental factors and antidote addition on glyphosate toxicity to freshwater fish, *Labeo rohita* (Hamilton)». En: *Chemistry and Ecology* 23.4 (2007), págs. 279-287.
- [2] RB Brown y JPGA Steckler. «Prescription maps for spatially variable herbicide application in no-till corn». En: *Transactions of the ASAE* 38.6 (1995), págs. 1659-1666.
- [3] Martina Koller, W Lanini y col. «Site-specific herbicide applications based on weed maps provide effective control». En: *California agriculture* 59.3 (2005), págs. 182-187.
- [4] Rainer H Biller. «Reduced input of herbicides by use of optoelectronic sensors». En: *Journal of Agricultural Engineering Research* 71.4 (1998), págs. 357-362.
- [5] Filip Feyaerts y Luc Van Gool. «Multi-spectral vision system for weed detection». En: *Pattern Recognition Letters* 22.6 (2001), págs. 667-674.
- [6] Jochen Hemming y Thomas Rath. «PA—Precision agriculture: Computer-vision-based weed identification under field conditions using controlled lighting». En: *Journal of agricultural engineering research* 78.3 (2001), págs. 233-243.
- [7] SeI Cho, DS Lee y JY Jeong. «AE—Automation and Emerging Technologies: Weed-plant Discrimination by Machine Vision and Artificial Neural Network». En: *Biosystems Engineering* 83.3 (2002), págs. 275-280.
- [8] A Linux Foundation Collaborative Project. Yocto Project. *About*. 2016. URL: <https://www.yoctoproject.org/about>.
- [9] Thomas Springer. *Intel Galileo Developer Board*. 2014. URL: [https://commons.wikimedia.org/wiki/File:Embedded\\_World\\_2014\\_Intel\\_Galileo\\_01.jpg](https://commons.wikimedia.org/wiki/File:Embedded_World_2014_Intel_Galileo_01.jpg).
- [10] Intel. *Downloads*. 2017. URL: <https://software.intel.com/en-us/iot/hardware/galileo/downloads>.
- [11] Ideasonboard. *Linux UVC driver and tools*. URL: <http://www.ideasonboard.org/uvcl/>.
- [12] OpenCV Team. *Open Source Computer Vision Library*. 2017. URL: <http://opencv.org/>.
- [13] The Astropy Developers. *Compound models*. 2017. URL: <http://docs.astropy.org/en/stable/modeling/#compound-models>.
- [14] Jessie YC Chen y Jennifer E Thropp. «Review of low frame rate effects on human performance». En: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 37.6 (2007), págs. 1063-1076.
- [15] Intel Newsroom. *Intel CEO Announces Collaboration with Arduino to Inspire Creativity, Learning and Invention with Makers and Students*. Inf. téc. Maker Faire Rome, 2013.