

Inverted tracking algorithm for the field survey through artificial vision and robotics

Eduardo Álvarez¹, Sandra Serafino¹, Benjamin Cicerchia^{1 3}, Agustín Balmer¹, Claudia Russo^{1 2}, Hugo Ramón^{1 2}

¹ Instituto de Investigación y Transferencia en Tecnología (ITT), Escuela de Tecnología, Universidad Nacional del Noroeste de la Provincia de Buenos Aires (UNNOBA) – Centro Asociado de la Comisión de Investigaciones Científicas de la Pcia. De Bs. As. (CIC)

² Investigador Asociado de la Comisión de Investigaciones Científicas de la Pcia. De Bs. As. (CIC)

³ Becario de la Comisión de Investigaciones Científicas de la Pcia. De Bs. As. (CIC)

Sarmiento y Newbery (CP 6000), Junín, Buenos Aires, Argentina. Teléfonos (0236) 4636945/44

eduardo.alvarez,sandra.serafino,lucas.cicerchia,claudia.russo,hugo.ramon}@itt.unnoba.edu.ar;
balmerags@gmail.com

Abstract. The area of artificial vision and robotics has very important advances in the recognition and tracking of objects, not only in indoor scenes but also in outdoor ones. These methods and algorithms have given rise to very important technological advances in different areas of knowledge. In the area of Precision Agriculture, the main problem of its use lies in its application in field surveys, whereas in the case of cultivation, we will have fixed objects (seedlings) in established spaces (furrows and plots), but in uncontrolled environments. The determination of the density of these crops and their distance between furrows among other data is in many cases, relevant to their performance. It is the purpose of this paper to solve the automated sensing of this data through the use of cameras and artificial vision techniques. In this work, an inverted tracking algorithm is defined in order to automatically determine the necessary shot-points by means of which the cameras involved as sensors on a robotic platform capture scene images. This will help to survey the density and distance of the crop to be analyzed.

Key Words: Precision Agriculture, Tracking, Data Relief, Artificial Vision.

I. INTRODUCCION

El presente trabajo se inserta en el marco de un proyecto de investigación en el cual se busca realizar el relevamiento de datos acerca del cultivo de maíz a campo. Gracias a los avances tecnológicos en áreas como robótica, visión artificial, sensores, comunicación y conectividad, es que se ha desarrollado una plataforma robótica que contiene un sistema de navegación con sensado de diferentes tipos de imágenes, las cuales mediante procesamiento digital, facilitan no sólo la recolección de datos biofísicos de un cultivo, a lo largo de la evolución de sus estados fenológicos, sino también el monitoreo y evaluación de los mismos [1][2]. Para obtener dichos datos la plataforma debe sensar el cultivo mediante la captura de imágenes, a partir de un recorrido establecido en el ensayo, para lo

cual debe determinar en qué momento del recorrido se deberán sensar las parcelas. Para determinar dicho instante el sistema debe ser capaz de detectar los diferentes plantines del cultivo, teniendo en cuenta además cuales de ellos ya fueron sensados. A partir de esto es que se dispone anexar a la plataforma la utilización de una cámara de video que permita realizar un tracking de las diferentes parcelas.

El video tracking es el proceso de localización de un objeto móvil en un período de tiempo obtenido mediante una videocámara. El tracking que se resuelve en este caso posee la particularidad que no son los objetos de la escena los que se mueven delante de una cámara fija sino que es la cámara la que está en movimiento ante un escenario con objetos que se mantienen estáticos (de allí se determina el invertido), ello sumado a que se tiene que detectar más de un objeto (planta). En la literatura se encuentran diferentes trabajos que contienen algunas de las particularidades del caso. Donde hemos destacado sistemas de tracking en ambientes outdoor para secuencias de imágenes que no varían demasiado en el tiempo [3], tracking de objetos mediante la utilización de robots [4][5], vehículos aéreos no tripulados (UAV) [6] o tracking mediante la utilización de plataformas en movimiento [6] [7], aplicados por ejemplo, como asistencia al manejo de automóviles [8]. Por último, se encuentran trabajos donde se realiza el tracking de objetos en movimiento con cámaras en movimiento [8] [9] [10] [11]. A la hora de detectar los objetos a sobre los que se realiza el tracking se encuentran trabajos con diferentes tipos de segmentación de la imagen de video, destacando aquellos realizados mediante color [12] [13] [14], donde algunos además incorporan texturas [15][16] o formas [13] [8], o mediante histogramas [17], o trabajan directamente sin segmentación [18]. Sumado a ello en cuanto a filtros a la hora de realizar el tracking hemos de destacar la utilización del filtro de Kalman [10] [19], filtro de partículas [20] y algoritmo de Kuhn-Munkres [21].

En base al análisis del estado del arte mencionado y a las características particulares del problema a resolver, fundamentalmente las características de la escena, y de su entorno, se define un algoritmo de tracking invertido en el que cada paso y cada método utilizado ha sido ajustado de modo de conseguir eficiencia con el mínimo costo computacional asociado. En este sentido, ha sido necesario definir y caracterizar claramente la escena, los objetos a detectar, los sensores y la estructura de hardware que los soportan. El desarrollo del trabajo expone el resultado de estas decisiones y el sustento de las mismas.

II. MATERIALES

A. Sobre la escena

El estudio se realizó sobre diferentes ensayos de maíz en diferentes estadios, es decir aplicados a diferentes etapas del crecimiento del cultivo, principalmente en estadios tempranos (V2/V3), donde la numeración acompañada indica la cantidad de hojas que ha logrado desarrollar la planta hasta el momento.

Los ensayos se plantean de forma regular en un área determinada del campo y se realizan en base a las necesidades que se tenga respecto de los datos fenológicos a relevar. Un ensayo es en líneas generales un conjunto de canteros. Los canteros tienen un largo variable (entre 100 y 400 metros por ejemplo), y una separación entre ellos llamada caminos de 1 m. de ancho por el largo del cantero. Están compuestos por un número preestablecido de parcelas donde una parcela es un conjunto de dos surcos con 70 cm. de distancia entre ellos; cada parcela tiene 5 m. de largo x aproximadamente 1,40 m. La distancia entre una parcela y otra es de 1 m. Un surco es una línea de siembra que tendrá un largo de 5 m. Dado que la decisión de la separación entre plantas está estipulado de antemano se conoce a priori el número de las mismas que debería contener cada surco.

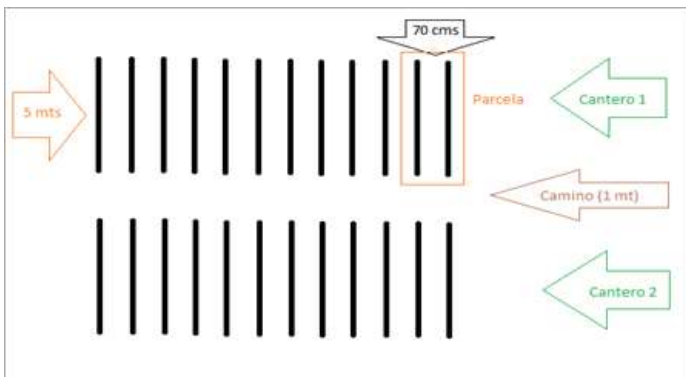


Fig. 1. Diagrama que ilustra la estructura de los ensayos

Para esta estructura de sembrado, se define un recorrido lineal, utilizando los caminos entre canteros, de modo que los sensores abarquen en cada paso la totalidad de cada parcela tanto a lo largo como a lo ancho de las mismas.

B. Sobre el dispositivo y los sensores

La plataforma móvil cuenta con tres cámaras, donde dos de ellas se utilizan para capturar imágenes que permiten la obtención de datos, el monitoreo y evaluación, y la restante que permite capturar el video que se procesa en línea (mediante la utilización de un algoritmo

de tracking invertido) para la detección de los surcos y la generación automática del disparo de las otras dos cámaras según corresponda. Para la obtención de las imágenes que luego serán procesadas para el monitoreo y análisis de parámetros fenológicos se utiliza una cámara de luz visible WebCam Genius WideCam 1050 con gran angular (que llamaremos en adelante S1_WCWC), con una resolución desde 640x480 hasta 1280x1204 píxeles [28]. Ésta, mediante una óptica gran angular de 120° permite capturar una parcela completa (en largo y ancho) incluyendo todos los plantines sembrados en la misma. También se utiliza como segundo sensor una cámara multispectral Tetracam ADC Lite (que llamaremos en adelante S2_MC) la cual permite obtener una imagen con 3 bandas de frecuencia diferentes (Red, Green e IRC) con una resolución que va desde 1024 x 768 hasta 2048 x 1536 píxeles [27]. Esta cámara permite obtener información espectral de la misma escena capturada por la cámara de luz visible, la que a su vez permitirá resolver índices de vigorocidad de cultivo mediante el cálculo de Índices Verdes (NDVI) [29]. Estas dos cámaras se disponen en la plataforma en forma volada sobre las parcelas, una al lado de la otra, con soportes individuales que permiten que ambas puedan regular su altura.

Para la obtención de la señal de video, que permite determinar bajo procesamiento digital los shot-points para los sensores S1_WCWC y S2_MC, se utiliza una cámara de luz visible Webcam Genius Facecam 321 (que llamaremos en adelante S3_WCV), con una resolución de video de 640 x 480 píxeles y de 30fps [30].

Para la administración de las tres cámaras se utiliza un dispositivo RaspBerry Pi 3 Modelo B [22], que posee un procesador quad-core ARMv8 de 1.2GHz y 64-bit y una memoria RAM de 1GB. El mismo cuenta con Sistema Operativo RaspBian [23] basado en Debian.

Para el procesamiento de video en tiempo real se utiliza el lenguaje de programación Python 3 [24], junto con librerías OpenCV 3.1.0 [25].

C. Sobre los objetos a detectar

Estos objetos son las plantas de maíz por surco y por parcela dentro de un ensayo, y están sujetos a las condiciones climáticas (viento, iluminación diferente, humedad, etc.) reinantes en el momento de la captura. Existen varios aspectos a tener en cuenta por el algoritmo en relación a ellos.

Una primera cuestión es la evolución fenológica del cultivo, dado que no será lo mismo evaluar los surcos en estados iniciales (v2/v3) que en estados más avanzados donde la cobertura del suelo será mayor y existirá una diferenciación casi nula entre los cultivos de un surco y de otro.



Fig. 2. (a),(b),(c),(d). Escenas que el algoritmo de tracking invertido tiene que evaluar. a) surcos con cultivos en estadio v7/v8, b) surcos con cultivos en estadio v3/v4, c) surcos con cultivos en estadio v1/v2, indicados con línea amarilla d) Surcos con cultivos en estadio v5/v6.

Dada la variabilidad de la escena sobre la que se debe trabajar, la captura de video para el algoritmo de tracking invertido siempre contemplará la inclusión del inicio de los surcos donde está presente para todos los estadios del cultivo el factor suelo y el detalle de los tallos para los estadios más avanzados (Fig.3a,3b) que son aquellos en los que ya no será posible discriminar a partir de las hojas de la planta.

Otro aspecto que se tiene en cuenta para la definición del algoritmo es la determinación de cuál o cuáles serán las características del objeto que se debe detectar en el video. En nuestro análisis, el atributo que muestra mayor diferencia en todas las escenas, aún en toda la curva evolutiva del cultivo, es el color. Esta característica no sólo mantiene un rango distinguible en la escena sino que también se mantiene diferente respecto del resto de los objetos que forman la misma, como es la cobertura del suelo (por ejemplo, suelo desnudo o rastrojo). Sin embargo, es importante comentar que la escena no siempre contiene solamente cultivo y suelo desnudo o rastrojo (Fig.3a,3b). Es común encontrar entre ambos, otros objetos que llamaremos malezas o yuyos, que comparten el atributo color mencionado anteriormente. En consecuencia se incorporan como nuevos atributos tanto la forma como el área ocupada de los objetos detectables, que tienen como objetivo colaborar en el ajuste de la detección de las plantas por surco y que también son adoptados por el algoritmo.



Fig. 3. (a), (b) Escenas donde puede observarse que el suelo está cubierto por malezas que comparten la característica de rango de color con los cultivos, lo que dificulta la detección basada sólo en este atributo.

III. DEFINICIONES Y PARÁMETROS UTILIZADOS

Dado el problema a resolver, el algoritmo debe determinar automáticamente en qué momento del recorrido de la plataforma se deben sensar las parcelas. Dicho de otra forma, debe ser capaz de detectar surcos diferentes correspondientes a parcelas diferentes para generar la captura de imágenes de las mismas.

A. Shot-Points

Los shot-points correspondientes a las zonas correctas de disparo se muestran en la figura 4. El sensado de toda la parcela (tanto en ancho como en largo) se registra en una única pasada de la plataforma por cada camino.

B. Cantidad de Frames

El algoritmo no procesa la totalidad de frames que captura el sensor S3_WCV, sino una parte de ellos teniendo siempre en cuenta el último que fue almacenado en el buffer del mismo. Esta decisión se basa en dos aspectos. Por una lado, la poca variabilidad de la escena lo que hace que la diferencia de datos entre frames contiguos no sea sustancial para lo que se busca y por otro el limitante impuesto por el hardware utilizado, lo que llevaría a un procesamiento mucho más lento que el requerido para la solución del problema.

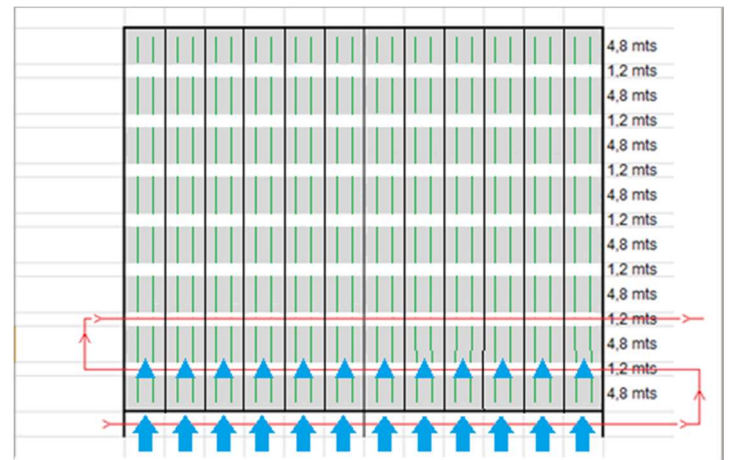


Fig. 4. Shot-points (flechas en color celeste) que el algoritmo detecta en forma automática mediante procesamiento de imágenes aplicado a frames determinados de la señal de video.

El mecanismo de trabajo se basa en tomar el primer frame del buffer y procesarlo. A la par, mientras se realiza este procesamiento un thread continua leyendo los frames que va capturando la cámara y cuando termina de procesar vuelve a tomar un nuevo frame del buffer que será siempre el último frame leído por el thread. Este proceso se repite hasta que el sistema genera la señal de stop.

C. Procesamiento de cada frame

El algoritmo no trabaja sobre el frame completo, o sea que no procesa los 640 x 480 pixeles del mismo. Esta decisión se basa en el hecho de que para la detección del surco no es necesario tomar toda la línea de cultivos que capta la resolución de la cámara, sino basta con el inicio de ella. El algoritmo sólo procesa un porcentaje de la altura del frame que se ajusta en función de un parámetro del

sistema que indica el estadio del cultivo antes de iniciar el proceso de relevamiento de datos. Esta variación se basa en una relación inversamente proporcional; a mayores estadios alcanzará con un menor porcentaje de filas de la imagen.

Dado $f(x,y) = i$, donde $1 \leq x \leq n$, $1 \leq y \leq m$, con n filas y m columnas, un frame cualesquiera de la señal de video. Dado $pe = ve$, valor entero $ve = 4,7,10$, número de estadio de la planta de cultivo representativo (en relación al tamaño) del resto de los estadios (2 al 13). Se obtienen las ecuaciones para el valor porcentual que se aplica al frame (1), y para el recorte del frame que se procesa en realidad en cada paso (2).

$$p = (100/pe) \%, \quad pe = 4,7,10 \quad 0 \leq p \leq 100 \quad (1)$$

$$Rf = f(u, v) = i, \quad \text{donde } (n \cdot p) \leq u \leq n, 1 \leq v \leq m \quad (2)$$

Rf es entonces el recorte del frame en curso de procesamiento tomando siempre desde la fila indicada por el % aplicado hasta la última y para todas las columnas.

IV. TÉCNICAS DE PROCESAMIENTO DIGITAL DE SEÑALES APLICADAS

El algoritmo utiliza procesamiento digital de señales. En particular en nuestro caso el mismo aplica diferentes técnicas de procesamiento digital de imágenes y procesamiento digital de video. Los datos sobre los que se aplican éstas técnicas son los obtenidos por el sensor S3_WCV. Se trata de un stream de video con una resolución de 30 fps, del que se procesan sólo recortes de acuerdo con (2).

Para el procesamiento digital de la señal de video se utilizan librerías del paquete OpenCV que permiten tanto visualizar el stream, ajustarlo a los parámetros que el sistema administra, y recolectar los frames que lo conforman a través de un buffer con el que cuenta el sensor. En este caso las operaciones utilizadas sobre la señal son las básicas del paquete y no han sido transformadas.

Para el procesamiento digital de imágenes, se parte de la señal de video, y se trabaja sobre los frames que van quedando en el buffer a medida que corre el video. En particular la etapa de procesamiento implica técnicas basadas en conversión a esquemas de color HSI, umbralización basada en color, morfología matemática binaria, segmentación basada en contornos, y la etapa de análisis se basa en técnicas de identificación de características y reconocimiento de patrones. Los pasos del procesamiento implican 1) convertir recorte de RGB a HSI, 2) aplicar umbralización basada en color, 3) aplicar erosión y dilatación binaria para descartar elementos no deseados, 4) aplicar técnicas de detección de objetos basada en contornos, 5) aplicar técnicas de análisis estructural y descriptores de formas.

Para el paso 1), cada recorte que debe procesarse, utiliza el esquema de color RGB que es el que por defecto aplica el sensor. Dado que la escena que debe procesarse contienen poca variabilidad de colores (básicamente verdes del cultivo y marrones del suelo) se decide enfatizar estas características mediante la conversión a HSI (Hue-Saturation-Intensity) [26], dado que este modelo separa el componente de intensidad de la información de color de la imagen, que queda especificado en la capa de tono (H). En particular el tono está asociado con la longitud de onda dominante y representa el color dominante tal y como lo percibimos.

Para el paso 2) se utiliza un rango definido de tonos representativos de los cultivos, los que se encuentran parametrizados en el sistema. Esto se debe a la posibilidad de cubrir diferentes variedades de cultivos en los que, si bien se mantiene la característica de

color, ya no serán los mismos.

Con los tonos establecidos se aplica una umbralización por rango, cuyo resultado es la imagen binaria, tal como muestra la figura 6.

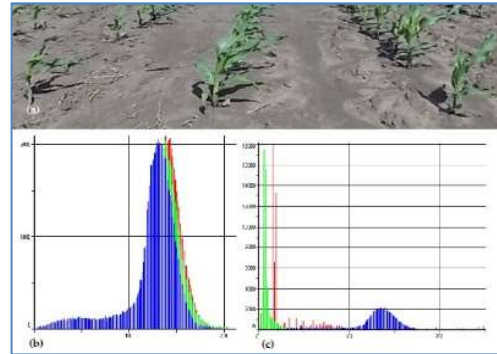


Fig. 5. a) Recorte de frame en estadios V4, b) histograma RGB, c) histograma HIS.

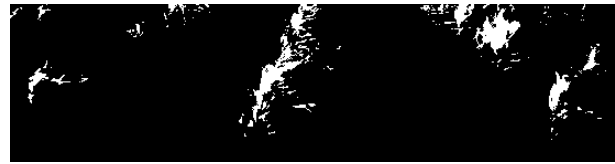


Fig. 6. Recorte binarizado mediante umbralización por color

Una vez que el recorte quedó binarizado, se inicia el paso 3) en el que se aplican operaciones simples (dilatación y erosión) de morfología matemática. Se utiliza un elemento estructurante (EE) tipo rombo de igual alto que ancho para que nos permita eliminar aquellos objetos que no cumplen con ser parte del cultivo. En éste punto es importante resaltar que nuevamente se utiliza el parámetro asociado al estadio actual que se releva. De acuerdo con éste parámetro se define el tamaño del elemento estructurante:

$$T(EE) = \begin{cases} pe, & \text{si } pe = 2 \\ pe - 1, & \text{si } pe = 3,4 \\ pe - 2, & \text{si } pe = 5 \\ pe - 3, & \text{si } pe = 6 \\ (pe/2)+1, & \text{si } pe \geq 7, pe \leq 13 \end{cases} \quad (3)$$

Donde T es el tamaño del EE, y pe es el parámetro relacionado a los estadios del cultivo. De acuerdo con la ecuación (3), los pares (estadio - tamaño EE) quedan (2-2), (3-2), (4-3), (5-3), (6-3), (7-4), (8-5), (9-5), (10-6), (11-6), (12-7), (13-7).



Fig. 7. Recorte binarizado y aplicadas operaciones de dilatación y erosión para quitar los elementos indeseados

En el siguiente paso (paso 4), se aplica un método de búsqueda de contornos sobre la imagen binaria resultante, el que retorna los objetos detectados en forma individual (figura 8). Cada uno de estos objetos serán analizados en el siguiente paso para determinar si corresponden a las características de los objetos que se pretenden

detectar, o sea cultivos.



Fig. 8. Detección de contornos sobre la imagen color.

En el siguiente y ultimo punto (paso 5), es necesario volver a tener en cuenta dos nuevos parámetros, uno para el área y otro para la forma. Éstos son parte de los descriptores de los objetos buscados, y nuevamente son dependientes del estadio actual del cultivo en el momento del sensado a campo.

En el caso de el área, la misma se cuenta en píxeles y la relación establece que a menor estadio menor área. Sean $Ob=(ob1,ob2,...,obn)$ los n objetos detectados en el paso 4, sea $A(ob_i)$ el área del i -ésimo objeto detectado, y sea pe el estadio actual del cultivo a sensar. La relación establece que un objeto será aceptado como tal, si cumple con:

$$A(ob_i) > AP(Ob) - [pe * 10], 1 \leq i \leq n \quad (4)$$

donde $AP(Ob)$ es el área promedio de todos los objetos detectados.

En el caso de la forma, hay que hacer una diferenciación entre los estadios menores y los mayores, donde como ya hemos explicado, tendremos visibilidad de los tallos y no de las hojas. En los estadios menores donde se observan las hojas, se tiene en cuenta para esta característica, el área del mínimo polígono que contiene a la forma. Esta referencia se utiliza para distinguirlo de los llamados yuyos o rastros. Y en aquellos estadios donde se observa el tallo, se tiene en cuenta la relación alto/ancho. Por lo tanto se define que un objeto será aceptado como tal, si cumple con:

$$Si 2 \leq pe \leq 8$$

$$\text{Min}(AMPC(ob_i)) > n-1 \sum_{i=1}^n Ampc(ob_i), 1 \leq i \leq n \quad (5)$$

donde $AMPC$ es el área del mínimo polígono que contiene al objeto. En éste caso el objeto deberá cumplir con que ésta área sea mayor que el promedio de las áreas de los mínimos polígonos que contienen a cada objeto. En cambio, Si $9 \leq pe \leq 13$:

$$Al(ob_i) > An(ob_i), 1 \leq i \leq n \quad (6)$$

donde Al se refiere al alto del objeto en pixels y An al ancho.

V. ALGORITMO DE TRACKING INVERTIDO

El objetivo del algoritmo de tracking invertido es determinar con la mayor exactitud posible la zona de disparo de las otras cámaras ($S1_WCWC$, $S2_MC$) y que el sistema se encargue de generarlos. Para esto se dispone el sensor $S3_WCV$ sobre la plataforma de navegación de forma de poder captar siempre suelo y cultivo, se toma esta señal de video, se toman un número mínimo de frames para su procesamiento digital, y se aplican técnicas de procesamiento de imagen para determinar los inicios de surcos y las parcelas dónde deben dispararse las otras cámaras.

El algoritmo captura la señal de video de manera continua durante el recorrido de la plataforma. Capturará, mientras no encuentre la señal de stop, todos los frames de video, los que de manera simultánea va procesando.

El algoritmo trabaja sobre dos buffers de datos. Uno es el buffer propio del sensor, en el que se van dejando los frames que captura durante la toma del video, que llamaremos buffer de frames. El otro buffer, que indicaremos como buffer de tracks, se utiliza para almacenar aquellos tracks que se obtienen del procesamiento de cada frame.

A la hora del procesamiento siempre tomará el último frame leído del buffer del sensor. A partir del frame obtenido, se selecciona el área a procesar y luego se binariza la imagen a partir de una umbralización por color en formato HSI. Los objetos encontrados en esta etapa se llaman *contornos*.

Una vez obtenida la umbralización se aplica morfología matemática, búsqueda de contornos y filtro de formas para obtener los objetos a realizar el tracking, en este caso plantines de maíz, discriminados de malezas o yuyos. Estos objetos resultantes del paso de los filtros indicados se consideran detecciones válidas y se llaman *detecciones*.

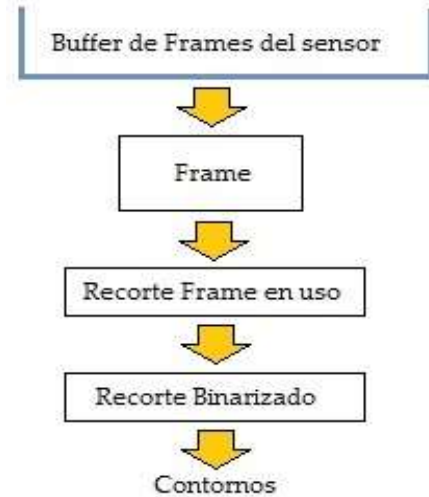


Fig. 9. Esquema del algoritmo de tracking invertido hasta la detección de objetos en el recorte binarizado mediante técnicas de detección de contornos

Una vez obtenidos los objetos candidatos a ser plantines (detecciones), se procede a la asignación de las detecciones a los tracks existentes en el buffer de tracks. Esta parte del proceso se lleva a cabo en dos pasos. Primero se genera una matriz de costos que contempla las distancias de las detecciones en relación con los tracks del buffer. Luego con esta matriz de distancias se aplica un algoritmo de optimización combinatoria de asignaciones conocido como Algoritmo de Kuhn-Munkres [21]. Si la detección que se analiza corresponde a un track existente (DAT), o sea que fue asociada por el algoritmo de Kuhn-Munkres, se actualiza el track correspondiente con su posición. Si la detección no fue asignada entonces se la considera como un nuevo track (TN) y se incorpora el mismo al buffer de tracks para que quede disponible para la evaluación del siguiente frame. Puede darse el caso de que queden tracks del buffer de tracks que no tengan nuevas asignaciones durante la evaluación de algún frame. Estos tracks que llamaremos tracks perdidos (TP), se mantendrán por un ciclo predeterminado en el buffer y luego serán desestimados (Fig.10).

Una vez finalizado el proceso de obtención y asignación de detecciones a tracks (figuras 9 y 10), el algoritmo analiza si hay al

menos dos tracks que se encuentren en el área de detección de parcela, parámetro establecido al comienzo del recorrido en base a la distancia entre surcos. Si se cumple esta condición se realizarán los disparos de las cámaras correspondientes y se marcarán los tracks como evaluados para que en el procesamiento del próximo frame no sean analizados.

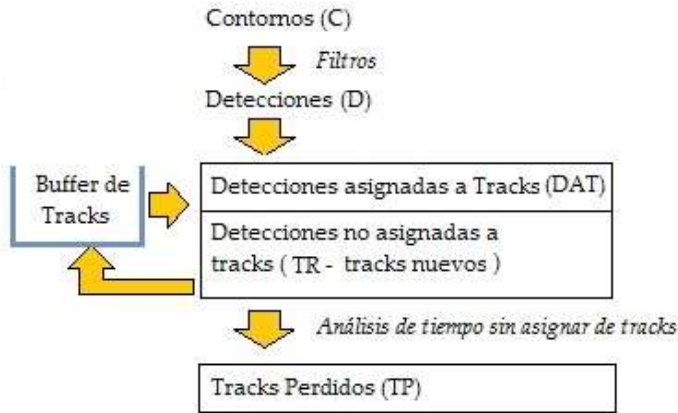


Fig. 10. Esquema del algoritmo de tracking invertido desde la obtención de detecciones válidas a partir de los contornos de objetos, hasta la asignación o no a tracks existentes en el buffer de tracks

Las figuras 11, 12 y 13 describen un pseudocódigo del algoritmo.

Algoritmo 1: Algoritmo principal

```

algoritmo_de_tracking():
1: Lectura de Señal de Inicio
2: área_de_disparo = Establece un área de disparo, establecido por dos líneas, parámetros establecidos al inicio (ancho de surco)
3: listado_de_tracks = []
4: Iniciar en paralelo la captura de video
5: Mientras no encuentre la señal de Stop hacer
6:   frame = leer el último frame
7:   objetos = procesar_frame(frame)
8:   listado_de_tracks_asignados,
   listado_de_tracks_sin_asignar = detección_de_tracks(listado_de_objetos)
8:   Si hay tracks en el listado_de_tracks_asignados entonces
9:     Actualización de Tracks de listado_de_tracks
10:  fin si
11:  Si hay tracks en el listado_de_tracks_sin_asignar entonces
12:    Creación de nuevos Tracks de listado_de_tracks
13:  fin si
14:  Si hay dos tracks del listado_de_tracks dentro de la zona_de_disparo entonces
15:    Realizar disparos de las cámaras
16:  fin si
17: fin mientras
18: fin

```

Fig. 11. Estructura principal del algoritmo

Algoritmo 2: Algoritmo de detección de tracks

```

detección_de_tracks(listado_de_objetos):
Entrada: Listado de objetos
Salida: Listado de tracks asignados y listado de tracks sin asignar
1: matriz_de_costos = Creación de la matriz de costos a partir de los tracks anteriormente detectados (el primer frame no va a contar con tracks) y el listado_de_objetos
2: tracks_detectados = Aplicación de filtro de Kuhn-Munkers a partir de la matriz_de_costos con las últimas detecciones y los tracks anteriores
3: listado_de_tracks_asignados = []
4: listado_de_tracks_sin_asignar = []
5: Para cada tracks_detectados:
6:   Si el track está en el listado_de_tracks entonces
7:     listado_de_tracks_asignados.append(track)
8:   sino
9:     listado_de_tracks_sin_asignar.append(track)
10: fin si
10: devolver listado_de_tracks_asignados,
   listado_de_tracks_sin_asignar

```

Fig. 12. Función de detección de Tracks

Algoritmo 3: Algoritmo que procesa un frame

```

procesar_frame(frame):
Entrada: Frame de video
Salida: Listado de objetos detectados
1: frame_recortado = Recortar frame en base a parámetros
2: frame_binarizado = Binariza el frame_recortado mediante un umbral por color con HSV
3: frame_con_morfología = Aplicación de Morfología Matemática mediante Erosión/Dilatación a frame_binarizado para eliminar malezas
4: frame_con_contornos = Búsqueda de Contornos en el frame_binarizado
5: lista_de_objetos = Filtro de formas por área de objetos (usando promedio de todas las áreas de los objetos detectados) en el frame_con_contornos
6: devolver lista_de_objetos

```

Fig. 13. Función que procesa el frame leído en el algoritmo principal

VI. RESULTADOS OBTENIDOS

El algoritmo que se presenta, se encuentra en el marco de un trabajo mucho más amplio, que implica la coordinación de varios sensores para el relevamiento a campo de datos fenológicos de cultivos, en particular en nuestro caso de maíz. Tal como se mencionó, las pruebas se realizaron fundamentalmente en la etapa temprana de desarrollo del cultivo, estadios V2/V3 y V4/V5.

Se trabajó sobre ensayos posibles, dado que no siempre coincidieron las épocas de sembrados y desarrollo de los mismos con el desarrollo de la plataforma. En este sentido también se realizaron pruebas de video en laboratorio, con videos armados utilizando imágenes de ensayos.

Se implementó una interfaz sobre la unidad de control de los sensores, que permite al operador administrar los parámetros de configuración, calibrar las cámaras en caso de ser necesario, e iniciar y detener el proceso de relevamiento automático durante el recorrido de la plataforma.

En el momento en que la plataforma se encuentre ubicada al inicio del ensayo y el operador indique el inicio del relevamiento

de datos, el sensor S3_WCV comenzará a capturar video y el algoritmo comienza a procesarlo. Sobre la pantalla de la unidad de control puede visualizarse en línea esta situación para que el operador verifique que el relevamiento es correcto o en caso contrario detenga la plataforma y reconfigure lo que sea necesario. Por ejemplo, puede configurarse el ancho de la ventana de disparo, dado que la misma está sujeta no sólo a la distancia aproximada a la que se siembran los surcos sino también a la distancia con la que la cámara tomará el video.



Fig. 14. Interfaz implementada sobre la unidad raspberry

Como puede observarse en la figura 15, la ventana de disparo (área encerrada entre líneas celestes) no enmarca dos surcos por lo que el disparo no será efectuado. En la siguiente imagen (Fig. 16), se observa la situación contraria en la que coinciden dos surcos dentro de la ventana y entonces el algoritmo dará la orden a los otros sensores habilitados sobre la plataforma (S1_WCWC y S2_MC), para que capturen la escena.

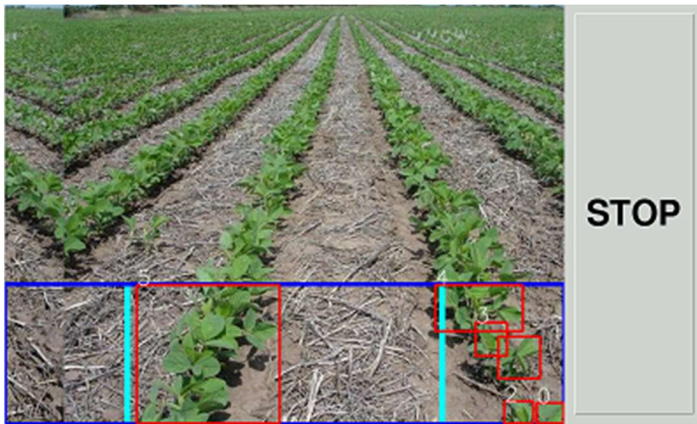


Fig. 15. Algoritmo de tracking invertido corriendo sobre un video de prueba armado en laboratorio. Puede observarse los tracks detectados etiquetados con números, y la ventana de disparo enmarcada en líneas celestes.



Fig. 16. Algoritmo de tracking invertido corriendo sobre un video de prueba armado en laboratorio. Los tracks detectados coinciden sobre las líneas de los surcos dentro de la ventana de disparo.

Las pruebas realizadas a campo también fueron satisfactorias respecto de la detección de las zonas disparo. En este caso se simuló el movimiento de la plataforma con una estructura auxiliar. El sistema en general y el algoritmo en particular resultaron efectivos a la hora de resolver el procesamiento de imagen y video en condiciones a campo. Las pruebas se realizaron con todos los sensores incluyendo el S1_WCWC (cámara de luz visible) y el S2_MC (cámara multiespectral) que capturaron correctamente la escena.



Fig. 17. Algoritmo de tracking invertido corriendo en un ensayo a campo, sobre una plataforma auxiliar. En el recuadro rojo puedo observarse la cámara que capta el video que el algoritmo procesa

Para evaluar la eficiencia del algoritmo, se realizó un análisis por frame procesado registrando los contornos encontrados (C), las detecciones (D), las detecciones asignadas a tracks existentes (DAT), detecciones no asignadas a tracks existentes o los que llamamos tracks nuevos (TN) y los tracks perdidos (TP) para cada recorte. Los datos fueron relevados de la corrida de 4 videos.

Como puede observarse en la tabla 1, en el caso del V1, sobre un promedio de 17 contornos por frame, y 6 detecciones válidas (resultantes del proceso de filtrado sobre los 17 contornos), el porcentaje de acierto para la vinculación de estas detección a tracks existentes es de un 100%. En promedio para los 4 videos, el porcentaje de detecciones asignadas a tracks (DAT) es del 97%.

TABLA 1
PORCENTAJE PROMEDIO DE DETECCIONES POR FRAME QUE SE ASIGNAN A TRACKS

videos	Cant. Frames analizados	Prom. Tracks en el buffer por frame	Prom. Detecciones por frame (D)	Prom. Detecciones asignadas a tracks por frame (DAT)	% DAT por frame
V1	533	8	6	6	100%
V2	327	10	8	7	88%
V3	865	5	4	4	100%
V4	807	7	6	6	100%
					97%

Datos relevados por frame procesado de la corrida del algoritmo de tracking invertido para los videos V1, V2, V3 y V4.

En la tabla 2 se analiza el porcentaje promedio de tracks existentes en el buffer de tracks que no reciben nuevas asignaciones, convirtiéndose en tracks nuevos o en tracks en espera de ser descartados una vez completado el ciclo de permanencia definido de antemano.

TABLA 2
% PROMEDIO DE TRACKS DEL BUFFER SIN ASIGNACIONES POR FRAME

videos	Cant. Frames analizados	Prom. Tracks en el buffer por frame	Prom. Detecciones por frame (D)	Prom. Detecciones no asignadas a tracks por frame (DAT)	% tracks del buffer sin asignaciones por frame
V1	533	8	6	2	25%
V2	327	10	8	2	20%
V3	865	5	4	1	20%
V4	807	7	6	1	14%
					20%

Esta tabla muestra que de todos los tracks del buffer en el momento del procesamiento de un frame, en promedio el 20% de los mismos no reciben nuevas asignaciones. Esto puede deberse a tracks que se pierden por ciclo y/o a la generación de nuevos tracks.

Respecto de la cantidad de disparos que el algoritmo genera, del análisis de los 4 videos se concluye una fuerte dependencia del ajuste de los parámetros indicados durante el desarrollo de este trabajo. Éstos son el estadio del cultivo, la distancia entre surcos dentro de una parcela, y la distancia desde el sensor S3_WCV al inicio de la parcela. Se detectó que el algoritmo responde con un porcentaje de éxito superior al 95% en relación a la cantidad de disparos requeridos si se configuran estos parámetros correctamente. Se probó el algoritmo sin configurar alguno de estos parámetros y se detectó que el algoritmo falla en más de un 30%. El origen de estas fallas se basa en obtener más detecciones que las reales (leer más surcos que los que existen). Y esto es producto de la configuración incorrecta de alguno de los parámetros mencionados. La tabla 3 muestra estos resultados para los videos estudiados. Es importante destacar que no se generan errores de disparo por menos de los necesarios. Esto significa que el algoritmo no pierde surcos en el recorrido.

TABLA 3
% DE DISPAROS GENERADOS POR EL ALGORITMO DE TRACKING INVERTIDO

videos	Cant. Disparos que deberían ejecutarse	Cant. Disparos generados	Cant. Disparos generados incorrectamente	% Disparos correctos	% Disparos incorrectos
V1	4	5	1	75%	25%
V2	3	4	1	67%	33%
V3	6	6	0	100%	0%
V4	6	7	1	83%	17%
				81%	19%

Esta tabla analiza los disparos generados por el algoritmo en relación a los disparos que debería realizar. En promedio el 81% de disparos se generan en forma correcta y cerca de un 20% se generan de más.

VII. CONCLUSIONES Y TRABAJOS FUTUROS

En el marco de este trabajo se desarrolló un algoritmo de tracking encargado de detectar objetos fijos mediante un sensor móvil que capta señal de video en escenas outdoor. Si bien el algoritmo se propone como solución a una problemática particular como es el sensado de datos de ensayos a campo, el mismo puede aplicarse a otras problemáticas que compartan sus características.

El diseño del algoritmo de tracking invertido se basa en el uso de un sensor móvil, sujeto a una plataforma robótica, y en el uso de una estructura de conectividad con otros sensores que dependen de su ejecución.

La mayor problemática a resolver es el procesamiento digital y análisis e interpretación de resultados en línea de los frames capturados por la cámara, debido a que la plataforma debe mantenerse en *movimiento continuo* durante el relevamiento de datos por parte de los otros sensores de la plataforma (cámara de luz visible y cámara multispectral). Esta problemática, sumada los recursos de hardware posibles para gestionar las 3 cámaras, limitan el tipo de procesamiento digital que puede aplicarse a la solución de la detección de objetos y lo dejan sujeto principalmente a los tiempos de ejecución. Desde este punto de vista también cuenta la resolución espacial y radiométrica de la imagen que debe procesarse.

Las decisiones sobre las que se sustenta el diseño del algoritmo apuntan básicamente a obtener una solución global a la definición eficiente de los puntos de disparo del resto de los sensores que debe controlar atendiendo estas problemáticas. Desde este punto de vista, los resultados obtenidos en las pruebas realizadas indican el camino correcto.

Por otro lado, se plantean como trabajos futuros, continuar realizando pruebas para los estadios más avanzados del cultivo (V6 a V13) y probar el uso del algoritmo sobre la plataforma concluida, lo que va a permitir reducir la inestabilidad de los videos desde el punto de vista del movimiento.

También se postula como trabajo futuro, resolver mediante procesamiento digital de señales la detección del final del cantero y el inicio del próximo para que el algoritmo detenga y reinicie los procesos de resolución de los disparos en forma automática, sin tener que detener la plataforma ni la ejecución de la aplicación.

REFERENCIAS

- [1] J. L. Araus and J. E. Cairns, "Field high-throughput phenotyping: The new crop breeding frontier," *Trends Plant Sci.*, vol. 19, no. 1, pp. 52–61, 2014.
- [2] G. Breccia and G. Nestares, "Next-generation phenotyping in plants: old problems, new promises," *BAG. J. basic Appl. Genet.*, vol. 25, no. 1, pp. 5–8, 2014.
- [3] J. Fuentes, J. Ruiz, and J. M. Rendon, "Salient Point Tracking for Key Frames Selection in Outdoor Image Sequences," *IEEE Lat. Am. Trans.*, vol. 14, no. 5, pp. 2461–2469, 2016.
- [4] H. Q. Pardo, J. L. Calvo Rolle, and O. Fontenla Romero, *Aplicación de un robot comercial de bajo coste en tareas de seguimiento de objetos.*, vol. 79, no. 175. 2006, Revista DYNA, 2012.
- [5] H. Ghazouani, "Shape and Color Object Tracking for Real-Time Robotic Navigation," Oct. 2014.
- [6] S. Kamate and N. Yilmazer, "Application of Object Detection and Tracking Techniques for Unmanned Aerial Vehicles," *Procedia Comput. Sci.*, vol. 61, pp. 436–441, 2015.
- [7] M.-X. Jiang, Z.-J. Shao, and H.-Y. Wang, "Real-time object tracking algorithm with cameras mounted on moving platforms," *Int. J. Image Graph.*, vol. 12, no. 3, p. 1250020, Jul. 2012.
- [8] J. Arróspeide and L. Salgado, "Video based vehicle detection and tracking for driver assistance systems," *Secur. Vialis*, vol. 7, no. 1–3, pp. 41–49, Dec. 2015.
- [9] F. Javier, G. Funes, D. Ivonne, and B. Trujillo, "Detección de movimiento de objetos mediante secuencias de video," INSTITUTO POLITÉCNICO NACIONAL, MEXICO, 2012.
- [10] W.-C. Hu, C.-H. Chen, T.-Y. Chen, D.-Y. Huang, and Z.-C. Wu, "Moving object detection and tracking from video captured by moving camera," *J. Vis. Commun. Image Represent.*, vol. 30, pp. 164–180, 2015.
- [11] Y. Wan, X. Wang, and H. Hu, "Automatic Moving Object Segmentation for Freely Moving Cameras," *Math. Probl. Eng.*, vol. 2014, pp. 1–11, 2014.
- [12] M. Bianchi, "Seguimiento de Objetos en Secuencias de Imágenes RGB-D," Universidad de Buenos Aires Facultad, 2015.
- [13] J. A. Etxeberria, "Algoritmo de reconocimiento de forma y color para una plataforma robótica," Universidad del País Vasco, 2010.
- [14] H. C. Barbosa, "Modelado y Seguimiento de Objetos por Medio de Distribución de Color," 2001.
- [15] E. Ozyildiz, N. Krahnstöver, and R. Sharma, "Adaptive texture and color segmentation for tracking moving objects," *Pattern Recognit.*, vol. 35, no. 10, pp. 2013–2029, 2002.
- [16] M. S. Allili and D. Ziou, "Active contours for video object tracking using region, boundary and shape information," *Signal, Image Video Process.*, vol. 1, no. 2, pp. 101–117, Jun. 2007.
- [17] M. Prieto *et al.*, "Algoritmo de seguimiento de objetos en imágenes mediante reconstrucción iterativa de histograma en tiempo real," *Jornadas Argentinas de Robótica*, no. November, p. 155, Nov. 2014.
- [18] H. T. Sogaard and H. J. Olsen, "Determination of crop rows by image analysis without segmentation," *Comput. Electron. Agric.*, vol. 38, no. 2, pp. 141–158, 2003.
- [19] E. C. J, D. Z. N, and R. Rojas, "Filtering for vision tracking," *Rev. Electrónica y Tecnológica e-Gnosis*, no. 4, pp. 0–18, 2006.
- [20] A. Rodríguez Moya, "Estudio del filtro de partículas aplicado al seguimiento de objetos en secuencias de imágenes," Universidad Carlos III de Madrid, 2009.
- [21] C. Hong, J. Zhang, C. Chungfeng, and C. Qinyu, "Solving large-scale assignment problems by Kuhn-Munkres algorithm," *2nd Int. Conf. Adv. Mech. Eng. Ind. Informatics (AMEII 2016)*, no. Ameii, pp. 822–827, 2016.
- [22] "Raspberry," 2017. [Online]. Available: <https://www.raspberrypi.org/>. [Accessed: 20-Apr-2017].
- [23] "Raspbian," 2017. [Online]. Available: <https://www.raspbian.org/>. [Accessed: 20-Apr-2017].
- [24] "Python," 2017. [Online]. Available: <https://www.python.org/>. [Accessed: 20-Apr-2017].
- [25] "OpenCV," 2017. [Online]. Available: <http://opencv.org/>. [Accessed: 20-Apr-2017].
- [26] R. C. Gonzalez and R. E. (Richard E. Woods, *Digital image processing*. Addison-Wesley, 1992.
- [27] TETRACAM ADC Lite. [Online]. Available: http://www.tetracam.com/Products-ADC_Lite.htm
- [28] GENIUS Widecam 1050. Manual Técnico Funcional. Genius.
- [29] J. Barreiros, N. Magne, D. Chavez, E. Alvarez. "Multispectral NDVI aerial image system for vegetation analysis by using a consumer camera". Power, Electronics and Computing (ROPEC), IEEE International Autumn Meeting on. 2014.
- [30] GENIUS Webcam Facecam 321. Manual Técnico Funcional. Genius.