

# Sistema de visión para conteo de paquetes deformables en una pila

Guillermo Steiner<sup>1</sup>, Guillermo Forte<sup>2</sup>, Gastón Araguás<sup>1</sup>

<sup>1</sup> Centro de Investigación en Informática para la Ingeniería  
Universidad Tecnológica Nacional, Regional Córdoba  
Maestro Lopez esq. Cruz Roja Argentina, Córdoba, Argentina

<sup>2</sup> Incubadora de empresas  
Universidad Nacional de Córdoba  
Av. Haya de la Torre s/n

**Resumen** En este trabajo se presenta un método novedoso de detección y conteo de paquetes de cabos de escoba mediante visión por computadora, en el que se aplica una estrategia de segmentado robusta ante la gran variabilidad de color, forma y posición de los paquetes. La detección y conteo de partes mediante cámaras es un tópico de mucho interés en la industria, entre otras razones por ser un sistema de medición libre de contacto. El principal inconveniente que presenta es la segmentación de la imagen para identificar las partes a contar, especialmente cuando poseen poca regularidad o el entorno es poco uniforme. El método propuesto aplica primero múltiples estrategias para la estimación inicial de las zonas de interés (paquetes), que luego se segmentan con una técnica de cortes de grafos. En este caso el segmentado se realiza utilizando el método llamado *GrabCut*, el cual requiere una estimación inicial de la zona a segmentar como frente o *foreground* y de la zona de fondo o *background*. Se describe también el sistema de visión propuesto y se muestran resultados reales del conteo. Finalmente, se analizan algunos casos problemáticos y se discuten posibles soluciones.

## 1. Introducción

Detectar y contar objetos utilizando visión por computadora es muchas veces la mejor solución o la única posible en determinado contexto. Ejemplo de ello se tiene cuando el conteo debe hacerse sin contacto físico, cuando se deben contar objetos distantes, o simplemente cuando no se cuenta con otra información que la visual sobre los objetos a contar. El conteo, definido como la estimación del número de objetos en una imagen estática o en un video, puede ser en sí mismo un problema [6], sin embargo la tarea de detectar y localizar los objetos a contar es casi siempre la parte más compleja. Cuando se trata de contar objetos en movimiento es posible utilizar la auto-correlación entre imágenes para extracción del fondo [5, 9], o su velocidad y dirección como información extra, lo que hace al proceso más robusto ante oclusiones, falsas detecciones y no detecciones [10]. En el caso de objetos estáticos la detección puede ser abordada como un problema de segmentación de imagen.

La segmentación de imágenes es un problema de la visión por computadora que aún carece de una solución general. Existen una gran cantidad de técnicas y algoritmos para segmentación, y cada uno presenta ventajas y desventajas según el contexto de aplicación [3, 2, 7]. Recientemente han tenido un fuerte desarrollo los métodos basados en grafos [11, 4, 12, 1], cuya estrategia de segmentación combina las características del contorno con propiedades de la región.

El problema a resolver consiste en el conteo de paquetes de cabos de escoba provenientes de una línea de producción, que son apilados manualmente sobre una plataforma rotante donde se realiza el paletizado final. Debido a la intervención de un operario en el apilado el conteo no puede hacerse a la salida de la línea sino que debe hacerse sobre la pila antes de ser paletizado. El apilado de los paquetes sobre la plataforma rotante dificulta la instalación de celdas de carga para realizar un conteo por peso. Además, debido a que la producción de estos cabos se realiza con bajas exigencias sobre el espesor de la chapa, el conteo indirecto por peso se hace muy impreciso. En la Fig. 1 se muestran ejemplos de pilas de paquetes sobre la plataforma de paletizado.

En el presente trabajo se propone un método de conteo basado en visión, que determina antes del paletizado si la cantidad de paquetes por fila es la correcta. El método aplica primero múltiples estrategias para la detección de las filas y sobre cada una de ellas realiza una estimación inicial de las zonas de interés (paquete), que luego se segmenta con una técnica de cortes de grafos. La técnica de segmentado se llama *GrabCut* [8], la cual requiere una estimación inicial de la zona a segmentar como frente o *foreground* y de la zona de fondo o *background*.

El trabajo se organiza de la siguiente manera: en la sección 2 se describe la línea de producción de los cabos y el empaquetado sobre la cual actúa el sistema de conteo, en la sección 3 se detalla el algoritmo de conteo, en la sección 4 se muestran algunos resultados sobre imágenes reales y finalmente en la sección 5 se comentan algunas conclusiones, dificultades y el trabajo futuro.

## 2. Descripción de la línea de producción

La línea de producción de cabos de escobas realiza en forma automática la fabricación de los cabos y el fraccionamiento en paquetes de 24 unidades. Los paquetes se apilan manualmente sobre una plataforma rotante para su posterior paletizado. El sistema propuesto realiza el conteo de paquetes de esta pila (Fig. 1), asumiendo conocidas la cantidad de filas y columnas. Cuando el conteo es diferente al esperado, el sistema indica por pantalla las filas con exceso o diferencia de paquetes e inhabilita el paletizado.

### 2.1. Variabilidad del empaquetado

El empaquetado de los cabos de escoba se realiza utilizando un polímero transparente. El cierre de los extremos se logra aplicando calor sobre el material sobrante, haciendo que la vista frontal de los paquetes presente una textura de gran irregularidad. Sumado a esto, la deformación de cada paquete debido al



**Fig.1:** Paquetes de cabos de escoba apilados para su paletizado.

apilado hace que se pierda también la regularidad en su forma. Debido a esta deformación la frontera entre los paquetes se diluye, dificultando la detección visual de los bordes.

Por otro lado, los cabos de escoba fabricados por esta máquina llevan inserto en sus extremos un capuchón y una rosca de plástico de colores varios. Estos plásticos de colores son los que luego quedan visibles en el apilado, que se realiza alternando la orientación del paquete para lograr mantener la pila derecha. Como consecuencia, los paquetes fabricados pueden ser de una gran variedad de colores, incluyendo la posibilidad de alternar de color entre filas de una misma pila en los casos en que el capuchón y la rosca son diferentes.

Por estas razones los algoritmos clásicos de segmentación no pueden ser aplicados en forma directa. En la Fig. 2 pueden verse ejemplos de casos de paquetes desalineados, dañados y de diferentes colores.

### 3. Algoritmo de conteo

Dada la variabilidad del empaquetado el algoritmo propuesto realiza el conteo de paquetes segmentando primero las filas que conforman la pila y luego los paquetes por fila. La segmentación por filas se realiza comenzando desde abajo ya que la deformación inicial es menor debido a que los primeros paquetes se apoyan sobre una superficie plana. Esta segmentación no implica independencia entre filas, puesto que los paquetes detectados en una fila son utilizados como información para determinar la posición de los paquetes de la fila superior.

#### 3.1. Detección de intersticios

La primera etapa en la segmentación de filas se realiza en base a los intersticios, es decir los espacios vacíos que quedan entre los paquetes durante el apilado, que por las características de la iluminación aparecen en la imagen como zonas oscuras. Los intersticios detectados son utilizados luego como anclaje



(a) Paquetes desalineados



(b) Paquetes rotos

**Fig.2:** Paquetes de cabos de alta variabilidad en tamaño, textura y alineación.

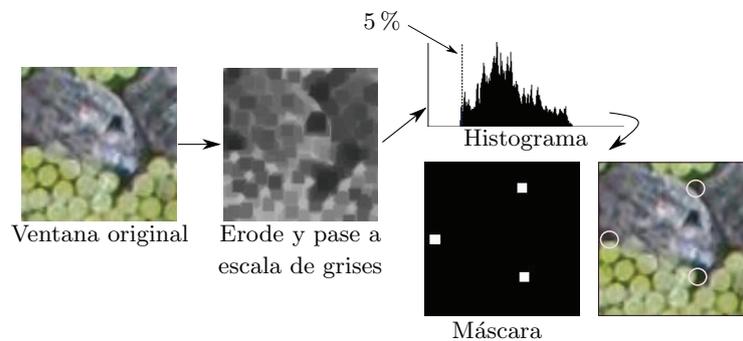
para la determinación de fronteras entre filas. Esta etapa puede considerarse un preproceso debido a que no utiliza información previa de filas ni paquetes, sino que realiza una estimación de intersticios a nivel global en toda la imagen. El proceso consiste en determinar la posición y superficie de las zonas oscuras presentes en la imagen binarizada. El binarizado se lleva a cabo mediante una serie de pasos según se muestran en la Fig. 3. Inicialmente se erosiona la imagen para magnificar los posibles intersticios. Luego se realiza una transformación de la imagen de 3 canales (RGB) a otra de 1 solo canal, utilizando la transformación clásica a escala de grises dada por la ecuación 1.

$$g = 0,299R + 0,587G + 0,114B \quad (1)$$

Sobre esta imagen en escala de grises se determina el umbral de binarización en forma adaptiva mediante una ventana deslizante. La elección del umbral se realiza en base al histograma de la imagen en la ventana asumiendo que:

- en cada ventana hay algún intersticio,
- en cada ventana no todo es intersticio.

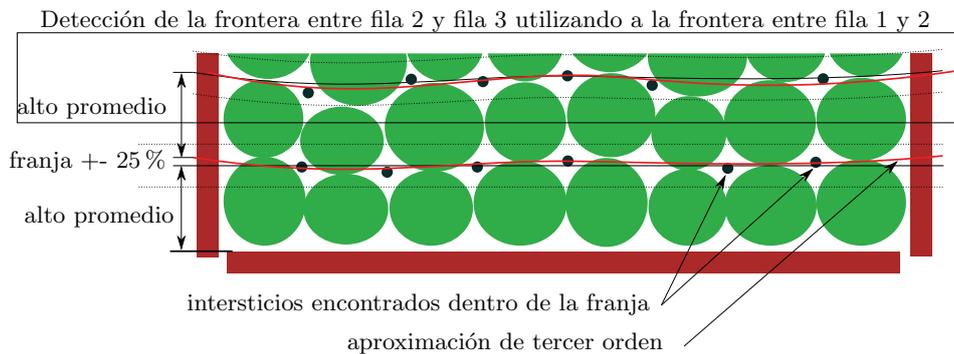
Luego se determina en el histograma el nivel de gris más cercano al negro que involucre menos del 5% de todos los píxeles de la imagen. Este valor de gris es el umbral que determina en cada posición de la ventana deslizante qué zona es intersticio y cuál no. El resultado final es una binarización adaptiva de toda la imagen que permite segmentar los intersticios de forma robusta ante las variaciones de iluminación.



**Fig.3:** Proceso de detección de intersticios mediante binarizado adaptivo.

### 3.2. Segmentado de filas

El siguiente paso es el segmentado de las filas, para el cual se considera que la primera de ellas se encuentra en la base de la imagen. Esta consideración implica una calibración inicial del sistema de visión que consiste en alinear la base de la imagen con el límite inferior de la pila de paquetes. Luego, comenzando desde la parte inferior de la imagen se buscan todos los intersticios que se encuentran a una altura de paquete (altura promedio debido a la deformación de los paquetes por apilado) y dentro una franja de  $\pm 25\%$  de este alto. Con el conjunto de puntos formados por los centros de los intersticios seleccionados se calcula el polinomio que mejor los aproxime. Este polinomio es la frontera que divide la primera fila de la segunda. En base a esta frontera y aplicando el mismo procedimiento a la próxima fila se determina la siguiente frontera. De esta manera se absorben las deformaciones progresivas que se producen en las filas por el apilado, lográndose una segmentación más robusta. El proceso se repite hasta completar la pila. En la Fig. 4 se muestra esquemáticamente el primer y segundo paso de esta segmentación.



**Fig.4:** Detalle del proceso de división entre filas.

### 3.3. Segmentado de paquetes

La etapa siguiente consiste en segmentar los paquetes dentro de cada fila. La segmentación de los paquetes se realiza utilizando una ventana deslizante del doble de ancho y de alto que un paquete medio, la cual se centra entre las dos fronteras de la fila comenzando por uno de los bordes de la imagen. Dentro de la ventana deslizante se busca el paquete y se estiman sus dimensiones, que en conjunto con su ancho medio se utiliza para calcular la nueva posición de la ventana. La operación se repite hasta que el centro del paquete a detectar se encuentra fuera de la pila de paquetes.

La detección del paquete dentro de la ventana se realiza utilizando un método de segmentado basado en grafos que se denomina *GrabCut* [8]. Este método permite determinar la frontera entre dos zonas de la imagen, una considerada *primer plano* o *foreground* y la otra *fondo* o *background*. Para esto se deben conocer previamente algunos píxeles o zonas de la imagen que pertenezcan al frente y otros que pertenezcan al fondo, y en que zona o conjunto de píxeles se encuentra la frontera. Esta información se pasa como entrada al método mediante una matriz del tamaño de la imagen, donde cada elemento de esa matriz representa el conjunto al cual pertenece el píxel correspondiente en la imagen. Considerando como primer plano o foreground al paquete y fondo o background al resto, los cuatro conjuntos necesarios son:

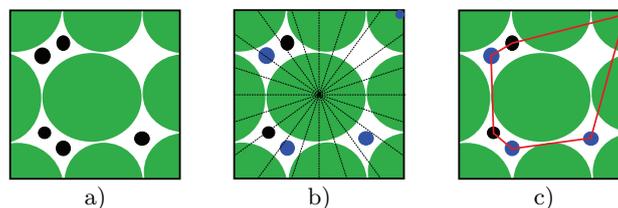
- Conjunto background: píxeles que con certeza no corresponde a un paquete.
- Conjunto foreground: píxeles que con certeza sí corresponde a un paquete.
- Conjunto probable background: píxeles que con cierta probabilidad no corresponde a un paquete.
- Conjunto probable foreground: píxeles que pueden corresponder a un paquete.

La salida del proceso es la misma matriz pero conteniendo sólo los dos primeros conjuntos. El éxito del método es altamente dependiente de la conformación de estos conjuntos de entrada, por lo que deben ser determinados con la mayor precisión posible. Suponiendo que la ventana de búsqueda del paquete se encuentra bien posicionada, existe una alta probabilidad de que su centro pertenezca a un paquete, por lo que la zona central se puede asumir como parte del conjunto foreground. Sin embargo, esto no siempre es correcto debido a la gran variabilidad presente en el apilado y en los paquetes. De igual forma se puede suponer que los píxeles del borde de la ventana pertenecen al background, aunque debido a que el tamaño de búsqueda de la ventana es mayor al de un paquete medio, en sus bordes se encuentran generalmente porciones de paquetes vecinos. La determinación de los conjuntos foreground y background es la parte más crítica del proceso y su descripción se realiza en detalle en la próxima subsección. Los conjuntos restantes se forman con los píxeles que no pertenezcan a ninguno de los dos conjuntos principales, y según su distancia al centro y el tamaño medio de paquete se los califica como posible foreground o posible background. Estos dos conjuntos son los que se reasignan al conjunto foreground o background durante el proceso, y su elección no introduce diferencias notables en el resultado final.

**Determinación de los conjuntos background y foreground.** La información más segura que se tiene sobre qué parte de la imagen pertenece al fondo o background está en los intersticios. Para utilizar esta información se debe asegurar antes que los intersticios involucrados no sean detecciones fallidas. Una característica que deben cumplir los intersticios válidos es que se encuentren en la zona de frontera entre paquetes, es decir fuera del paquete que se está segmentando. Para determinar si un intersticio se encuentra en la zona de frontera se realiza una secuencia de pasos representados en la Fig. 5. En la representación se muestra una situación inicial donde se consideran dos posibles fallas comunes, la múltiple detección de un mismo intersticio (representado por los dos puntos negros en la Fig. 5a) y la no detección de un intersticio (el correspondiente al cuadrante superior derecho de la misma Fig. 5a). Las reglas a aplicar son:

1. se divide radialmente la imagen en 20 sectores iguales, y en cada uno de ellos se detecta como máximo un intersticio,
2. se agrupan los sectores de cada cuadrante y se define un intersticio representante del cuadrante,
3. si un cuadrante no tiene ningún intersticio detectado se asigna como punto representante el punto esquina del cuadrante (puntos en azul de la Fig. 5b).

Luego, la unión de los intersticios y los puntos agregados definen la frontera del paquete, como se muestra en la Fig. 5c. El conjunto de píxeles que conforman esta frontera constituyen el conjunto background inicial.



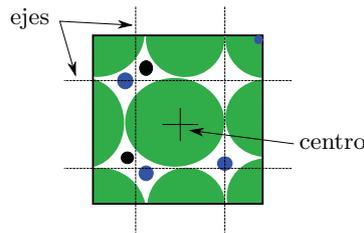
**Fig.5:** Secuencia para la definición de la frontera a través de los intersticios.

Con los intersticios detectados se puede determinar la posición del centro del paquete, esto permite obtener una superficie con alta probabilidad de pertenecer al paquete y determinar el conjunto de píxeles foreground. Para encontrar el centro se consideran los intersticios representativos de cada cuadrante. En base a ellos se calculan los ejes verticales y horizontales que unen a los cuadrantes contiguos. En este proceso se pueden presentar tres situaciones diferentes:

- que ambos cuadrantes contiguos posean intersticios representativo. Esta es la mejor situación donde simplemente el eje se calcula con el promedio de las dos coordenadas correspondientes ( $x$  para el eje vertical e  $y$  para el eje horizontal) de ambos intersticios,
- que un cuadrante posea intersticio y el otro no. En este caso se toma la coordenada del intersticio encontrado como coordenada del eje,

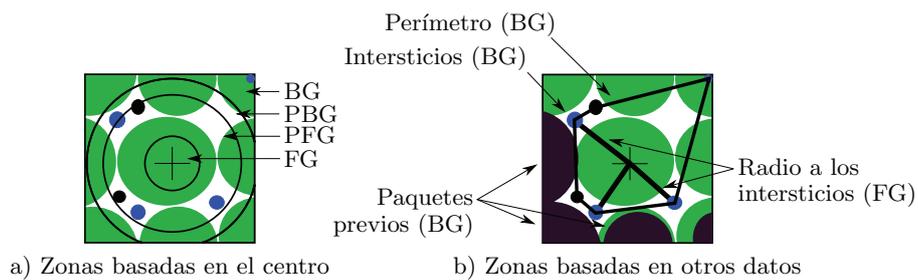
- que falten ambos intersticios. Se utiliza un eje ubicado en una posición relativa a los intersticios encontrados y fijada por el tamaño medio de paquete. En el caso extremo de no existir ningún intersticio en la ventana los ejes son ubicados en posiciones fijas establecidas a priori.

En la Fig. 6 se muestra esquemáticamente la determinación de los ejes, con los cuales se ajustan la posición y tamaño del paquete actual. A partir de esta infor-



**Fig.6:** Cálculo de los ejes y del centro del paquete.

mación actualizada se define una zona circular en el centro del paquete y de la mitad de su superficie cuyos píxeles constituyen el conjunto foreground inicial, como se muestra en la Fig. 7a (indicado como FG). Luego se traza un nuevo círculo concéntrico y de radio tal que incluya a todos los intersticios válidos. Los píxeles dentro del disco formado constituyen el conjunto de posibles foreground (PFG). Un tercer círculo concéntrico y del mayor radio posible tal que no sobrepase los límites de la ventana define el disco de píxeles que conforman el conjunto de posibles background (PBG). Los píxeles restantes (correspondientes a la zona entre el círculo de radio máximo y los límites de la ventana) se agregan al conjunto background inicial (BG).

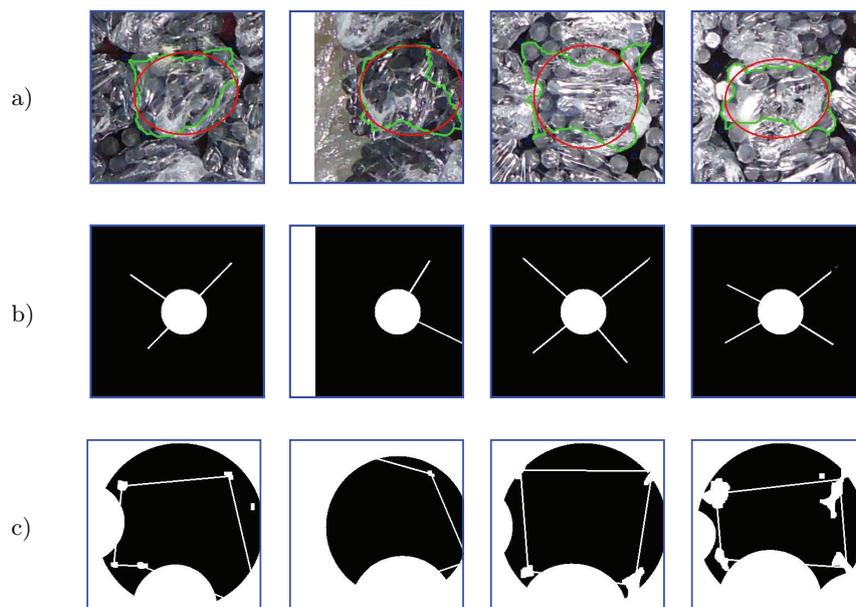


**Fig.7:** Conjuntos de píxeles que conforman la matriz de entrada del segmentador *GrabCut*.

Finalmente, en base a las siguientes consideraciones, se incorpora información extra a los conjuntos ya existentes:

- los píxeles que pertenecen a un intersticio válido son incorporados al conjunto background,
- el conjunto de píxeles pertenecientes a la línea que conecta el intersticios representativo de cada cuadrante y el centro actual del paquete forma parte del conjunto foreground (Fig. 7b),
- los paquetes vecinos ya segmentados que aparecen parcialmente en la ventana son incorporados al conjunto background.

Una vez conformada la matriz de entrada con toda la información disponible se aplica el método *GrabCut* para determinar la frontera entre los conjuntos foreground y background. El contorno del conjunto foreground es el contorno del paquete detectado, cuyo tamaño y posición se vuelven a actualizar en base a esta nueva información. En la Fig. 8 se muestran varios ejemplos donde se puede observar el resultado del segmentado (8a), el conjunto foreground utilizado como entrada al segmentador (8b) y el conjunto background (8c).



**Fig.8:** Ejemplos de detecciones de paquetes mediante *GrabCut*, en a) se muestran en verde el contorno detectado por el segmentador y la elipse que aproxima el tamaño del paquete, en b) se muestra en blanco el conjunto de píxeles de entrada al segmentador definidos como foreground, y en c) se muestra el conjunto background.

## 4. Resultados

La Fig. 9 muestra dos ejemplos de resultados satisfactorios obtenidos en dos pilas de diferente altura y conformada por paquetes de diversos colores, unos con extremo rojo y negro y el otro de extremos verde y negros. En el proceso exitoso se logra un 100% de correctas detecciones y 0% de falsas detecciones. En los ejemplos se puede observar la robustez del método frente a las deformaciones en las filas, a la variabilidad de tamaño y forma de los paquetes, y a algunas fallas del empaquetado.

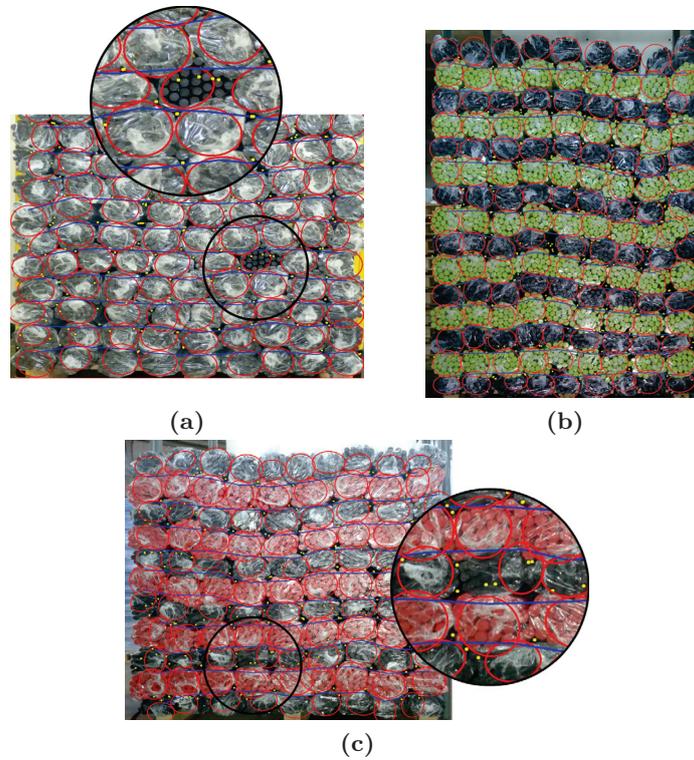
En la zona ampliada de la Fig. 9a se muestra un paquete dañado que logra ser bien segmentado a pesar de los múltiples intersticios que se detectan en su interior, los cuales son descartados por su cercanía con el centro estimado del paquete. También se observa (en la zona ampliada de la Fig. 9c) que la interdependencia de las diferentes etapas del segmentado, es decir la estimación de una fila considerando la anterior y luego la segmentación de cada paquete considerando los paquetes vecinos ya detectados, logra corregir pequeños errores en posición o tamaño de los paquetes. El error en el segmentado de este paquete en particular se debe a los intersticios detectados incorrectamente en su interior. Sin embargo aparecen algunos fallos cuando el proceso de empaquetado deja huecos en el centro, de forma que la situación mostrada en la Fig. 9a se repite a lo largo de toda la fila. Este defecto del empaquetado deja expuestas las puntas de los cabos, lo que provoca dos situaciones: 1) una detección múltiple de intersticios, 2) un segmentado parcial del paquete, como se ve en la Fig. 10a. En la Fig. 10b se muestra un ejemplo de esta problemática en múltiples paquetes, y el error de conteo en la fila defectuosa.

## 5. Conclusiones y trabajos futuros

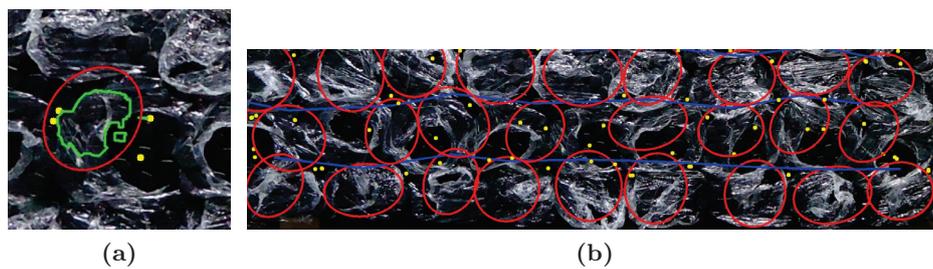
Se presentó un método novedoso de detección y conteo de paquetes de cabos de escobas en una pila. Debido a las características del apilado y otros factores como deformaciones y variabilidad en el tamaño de los paquetes, la aplicación cualquier método convencional de segmentado o conteo resultó inviable.

El método propuesto está basado en una serie de etapas que permiten segmentar los paquetes de cada fila utilizando el método de segmentado *GrabCut*. La segmentación inicial de filas basada en los intersticios mostró una gran robustez frente a las deformaciones de la pila. El segmentado de cada paquete considerando también estos intersticios resultó igualmente robusto frente a las variaciones de tamaño y forma.

Los resultados presentados sobre imágenes reales mostraron un buen funcionamiento del método bajo las condiciones descritas, sin embargo aparecen algunos fallos cuando la deformación de los paquetes aumenta, o cuando el proceso de empaquetado deja huecos en el centro (Fig. 10). Como trabajo futuro se plantea mejorar la robustez del segmentado para este caso particular, a partir de considerar las formas circulares presentes en el defecto mismo del empaquetado.



**Fig.9:** Detección y cuenta de paquetes de cabos de escoba apilados para su posterior paletizado. En la zona ampliada de (a) se muestra un empaquetado roto que logra ser bien segmentado a pesar de los múltiples intersticios que se detectan en su interior. En (c) puede verse un par de intersticios validados incorrectamente al centro de un paquete que provocan un error en el segmentado. Estos errores aislados son absorbidos por el método debido a la interdependencia entre filas y paquetes en el proceso de segmentado.



**Fig.10:** Paquetes con empaquetado defectuoso. En (a) se muestra en detalle cómo el defecto del empaquetado provoca una detección múltiple de intersticios y un segmentado parcial del paquete, con la consecuente falla en el conteo que aparece en la fila del centro en (b).

## Referencias

1. Suman Deb and Subarna Sinha. Comparative improvement of image segmentation performance with graph based method over watershed transform image segmentation. In Raja Natarajan, editor, *Distributed Computing and Internet Technology*, number 8337 in Lecture Notes in Computer Science, pages 322–332. Springer International Publishing, 2014.
2. M. Feixas, /A Bardera, J. Rigau, and Q. Xu. *Information Theory Tools for Image Processing*. Morgan & Claypool, 2014.
3. M. Haidekker. *Survey of Fundamental Image Processing Operators*. Wiley-IEEE Press, 2011.
4. Shoudong Han, Wenbing Tao, Desheng Wang, Xue-Cheng Tai, and Xianglin Wu. Image segmentation based on GrabCut framework integrating multiscale nonlinear structure tensor. *Image Processing, IEEE Transactions on*, 18(10):2289–2302, 2009.
5. Takumi Kobayashi, Tadaaki Hosaka, Shu Mimura, Takashi Hayashi, and Nobuyuki Otsu. Hlac approach to automatic object counting. In *Bio-inspired Learning and Intelligent Systems for Security, 2008. BLISS'08. ECSIS Symposium on*, pages 40–45. IEEE, 2008.
6. Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in Neural Information Processing Systems*, pages 1324–1332, 2010.
7. Yuyu Liang, Mengjie Zhang, and Will N. Browne. Image segmentation: A survey of methods based on evolutionary computation. In *Simulated Evolution and Learning*, number 8886 in Lecture Notes in Computer Science, pages 847–859. Springer International Publishing, December 2014.
8. Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004.
9. Jen-Chao Tai and Kai-Tai Song. Background segmentation and its application to traffic monitoring using modified histogram. In *Networking, Sensing and Control, 2004 IEEE International Conference on*, volume 1, pages 13–18, 2004.
10. Junliang Xing, Haizhou Ai, Liwei Liu, and Shihong Lao. Robust crowd counting using detection flow. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2061–2064. IEEE, 2011.
11. Ning Xu, Narendra Ahuja, and Ravi Bansal. Object segmentation using graph cuts based active contours. *Computer Vision and Image Understanding*, 107(3):210–224, September 2007.
12. Faliu Yi and Inkyu Moon. Image segmentation: A survey of graph-cut methods. In *Systems and Informatics (ICSAI), 2012 International Conference on*, pages 1936–1941. IEEE, 2012.