

UNIVERSIDAD DE CONCEPCIÓN

FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA



Profesor Patrocinante:

Dr. Daniel Sbarbaro H.

Informe de Memoria de Título
para optar al título de:

Ingeniero Civil Electrónico

Algoritmo de Identificación de las Posiciones Para la Poda de las Parras Mediante Técnicas de Procesamiento de Imágenes Usando Sensor Kinect.

UNIVERSIDAD DE CONCEPCIÓN
Facultad de Ingeniería
Departamento de Ingeniería Eléctrica

Profesor Patrocinante:
Dr. Daniel Sbárbaro H.

Algoritmo de Identificación de las Posiciones Para la Poda de las Parras Mediante Técnicas de Procesamiento de Imágenes Usando Sensor Kinect.

análisis de imagen



Iván Andrés Faúndez Báez

Informe de Memoria de Título
para optar al Título de

Ingeniero Civil Electrónico

Diciembre 2014

Resumen

En la presente Memoria de Título se desarrolla un nuevo sistema de visión artificial capaz de identificar los puntos óptimos de poda para las parras de uva. Para llevar a cabo este concepto se trabajó con el dispositivo Microsoft Kinect, aprovechando así sus potenciales características que nos entregan valiosa información tridimensional de la escena enfocada.

Para la calibración del dispositivo se utilizó el toolbox para Matlab desarrollado por Daniel Herrera[16] junto con la interfaz gráfica desarrollada en la Memoria de Título de C. Hernandez [7] con la cual se obtiene la data en unidades de ingeniería.

Los datos son procesados en un computador portátil con el sistema operativo Linux Debían Squeeze 6.0.5. Además se instaló el software Matlab R2012a para la programación del algoritmo y creación de la interfaz de usuario. El set up experimental se constituyó con el sensor montado sobre un trípode frente a una parra real fijada en un pedestal y un telón blanco fue fijado en una de las paredes del Laboratorio de Control del DIE-UDEC.

El algoritmo se encuentra basado primeramente en la binarización de la imagen, logrando así mayor eficiencia de cálculo. Luego en el análisis de imágenes se crearon diversos métodos para localizar las principales secciones de la planta como lo son el cordón, las ramas y los nodos. Además se llevó a cabo la reconstrucción en tres dimensiones de la parra, aprovechando de esta forma las imágenes de disparidad entregadas por el sensor Kinect. Finalmente se presenta una interfaz de usuario, donde se asignaron las diversas funciones del algoritmo a cada uno de los botones en este HMI.

Los resultados obtenidos permiten concluir que a través de la visión por computador y eligiendo buenas técnicas de análisis de imagen se logra la creación de un algoritmo capaz de tomar buenas decisiones en la poda de la vid, esto siempre y cuando se sigan de buena manera las consideraciones del algoritmo, como por ejemplo elegir un umbral correcto para la binarización de la imagen, disponer de un telón para la segmentación de esta y tomar los resguardos indicados para localizar correctamente cada sección de la parra. Además, la reconstrucción 3D se llevó a cabo con éxito mediante las imágenes de disparidad proporcionadas por el dispositivo Microsoft Kinect, usando así uno de los avances tecnológicos mas actuales que ha salido al mercado de las videoconsolas, pero empleado en esta ocasión para el desarrollo del algoritmo que se describe en este informe.



Dedico este trabajo a Dios, mis padres y a Viviana.

Agradecimientos

Primeramente a Dios por guiar mi vida, ayudarme en cada paso que doy y sostenerme en mis debilidades. A mis padres Rafael y Rosa por apoyarme en todo momento, en donde a través de esfuerzos y sacrificios, han hecho de todo para que yo pueda ser feliz y sobre todo gracias por entregarme la herramienta mas valiosa que puedo tener en esta vida, los valores cristianos que me han inculcado desde niño los llevaré por siempre en mi corazón, sin duda quiero ser como ustedes. A la persona más importante que Dios entrego a mi vida, Viviana gracias por tu amor incondicional, tu apoyo siempre estuvo presente y siempre que te necesité estuviste a mi lado para entregarme una palabra de ánimo, llenas mis días de alegría.

A mis hermanos, primos y amigos, les agradezco porque en cada uno de ustedes he encontrado la amistad verdadera, han creído y confiado en mí, ustedes me han regalado vivencias en este camino de la vida que jamás olvidare.

A mis compañeros y amigos de universidad Andrés, Agustín, Jorgito y Renzo, los "Pesadillas", ustedes fueron los que me acompañaron en esta etapa, les quiero agradecer por todos los momentos que pasamos juntos, sin duda vivimos muchas gratas experiencias que hoy son grandiosos recuerdos que atesoraré en mi corazón por siempre, gracias por ayudarme cuando los necesite y por su amistad en las buenas y en las malas.

A mi profesor guía, el Sr. Daniel Sbárbaro, por sus sugerencias y disposición a ayudarme durante todo este proyecto.

Tabla de Contenidos

LISTA DE TABLAS	VIII
LISTA DE FIGURAS	IX
NOMENCLATURA.....	XI
ABREVIACIONES.....	XII
CAPÍTULO 1. INTRODUCCIÓN	1
1.1. INTRODUCCIÓN GENERAL.....	1
1.2. TÉCNICAS DE RECONSTRUCCIÓN 3D.....	2
1.2.1 Cámaras Estereoscópicas.....	2
1.2.2 Cámaras Time of Flight(ToF).....	3
1.2.3 Cámaras basadas en luz estructurada.....	3
1.3. LA PODA DE PARRAS.....	4
1.3.1 La Parra.....	4
1.3.2 Poda Segundo brote.....	5
1.4. TRABAJOS PREVIOS	6
1.4.1 Revisión Bibliográfica.....	7
1.4.2 Procesamiento y análisis de imágenes.....	7
1.4.3 Reconstrucción 3D y sensor Kinect.....	9
1.4.4 Calibración Kinect.....	11
1.4.5 La poda.....	13
1.4.6 Discusión	14
1.5. HIPÓTESIS DE TRABAJO	15
1.6. OBJETIVOS	15
1.6.1 Objetivo General	15
1.6.2 Objetivos Específicos.....	15
1.7. ALCANCES Y LIMITACIONES	15
1.8. TEMARIO Y METODOLOGÍA	15
CAPÍTULO 2. ADQUISICIÓN DE IMÁGENES.....	17
2.1. INTRODUCCIÓN	17
2.2. SENSOR KINECT.....	17
2.2.1 Principio de Funcionamiento	18
2.3. CALIBRACIÓN CÁMARA MICROSOFT KINECT.....	21
2.3.1 Parámetros Intrínsecos cámara RGB.....	21
2.3.2 Parámetros Intrínsecos Cámara de Profundidad.....	23
2.3.3 Parámetros Extrínsecos Kinect	24
2.4. ADQUISICIÓN DE IMÁGENES.....	25
2.4.1 Configuración del sensor con el PC.....	25
CAPÍTULO 3. ALGORITMO DE IDENTIFICACIÓN.....	31
3.1. INTRODUCCIÓN	31
3.2. PROCESAMIENTO DE IMÁGENES.....	32
3.2.1 Imagen RGB en escala de grises.....	32
3.2.2 Imagen en Escala de Grises a Imagen Binaria.....	37
3.3. ANÁLISIS DE IMÁGENES.....	40
3.3.1 Localización del Cordón.....	40
3.3.2 Procedimiento.....	41
3.3.3 Resultado.....	43
3.3.4 Discusión.....	44
3.3.5 Localización de las Ramas.....	45
3.3.6 Resultados y Discusión.....	50
3.3.7 Localización de los nodos.....	51

3.3.8	<i>Discusión y Resultados</i>	53
3.4.	RECONSTRUCCIÓN 3D.....	55
3.4.1	<i>Introducción</i>	55
3.4.2	<i>Imagen Rango 3D de la Parra</i>	55
CAPÍTULO 4. PRUEBAS Y RESULTADOS		58
4.1.	INTRODUCCIÓN	58
4.1.1	<i>Funcionamiento del HMI</i>	58
CAPÍTULO 5. CONCLUSIONES		71
5.1.	INTRODUCCIÓN GENERAL.....	71
5.2.	CONCLUSIONES.....	71
5.3.	TRABAJO FUTURO.....	72
BIBLIOGRAFÍA		74



Lista de Tablas

TABLA3.1 Características de Imagen RGB y Escala de Grises34



Lista de Figuras

Fig. 1.1 Cámara Estereoscópica[18].....	2
Fig. 1.1 Cámara Estereoscópica[18]	3
Fig. 1.1 Cámara Estereoscópica[18].....	3
Fig. 1.1 Cámara Estereoscópica[18].....	4
Fig. 1.1 Cámara Estereoscópica[18].....	6
Fig. 1.1 Cámara Estereoscópica[18].....	16
Fig. 1.1 Cámara Estereoscópica[18].....	18
Fig. 1.1 Cámara Estereoscópica[18].....	19
Fig. 1.1 Cámara Estereoscópica[18]Ejemplo de patrón de puntos proyectado [28].....	20
Fig. 1.1 Cámara Estereoscópica[18]Imágenes Kinect [28]	21
Fig. 1.1 Cámara Estereoscópica[18].....	22
Fig. 1.1 Cámara Estereoscópica[18].....	24
Fig. 1.1 Cámara Estereoscópica[18].....	26
Fig. 1.1 Cámara Estereoscópica[18].....	28
Fig. 1.1 Cámara Estereoscópica[18].....	29
Fig. 1.1 Cámara Estereoscópica[18].....	29
Fig. 1.1 Cámara Estereoscópica[18].....	31
Fig. 1.1 Cámara Estereoscópica[18].....	32
Fig. 1.1 Cámara Estereoscópica[18].....	34
Fig. 1.1 Cámara Estereoscópica[18].....	35
Fig. 1.1 Cámara Estereoscópica[18].....	36
Fig. 1.1 Cámara Estereoscópica[18].....	38
Fig. 1.1 Cámara Estereoscópica[18].....	39
Fig. 1.1 Cámara Estereoscópica[18].....	39
Fig. 1.1 Cámara Estereoscópica[18].....	40
Fig. 1.1 Cámara Estereoscópica[18].....	42
Fig. 1.1 Cámara Estereoscópica[18].....	43
Fig. 1.1 Cámara Estereoscópica[18].....	43
Fig. 1.1 Cámara Estereoscópica[18].....	45
Fig. 1.1 Cámara Estereoscópica[18].....	46
Fig. 1.1 Cámara Estereoscópica[18].....	46
Fig. 1.1 Cámara Estereoscópica[18].....	46
Fig. 1.1 Cámara Estereoscópica[18].....	47
Fig. 1.1 Cámara Estereoscópica[18].....	48
Fig. 1.1 Cámara Estereoscópica[18].....	48
Fig. 1.1 Cámara Estereoscópica[18].....	58
Fig. 1.1 Cámara Estereoscópica[18].....	50
Fig. 1.1 Cámara Estereoscópica[18].....	52
Fig. 1.1 Cámara Estereoscópica[18].....	53
Fig. 1.1 Cámara Estereoscópica[18].....	54
Fig. 1.1 Cámara Estereoscópica[18].....	55
Fig. 1.1 Cámara Estereoscópica[18].....	56
Fig. 1.1 Cámara Estereoscópica[18].....	57
Fig. 1.1 Cámara Estereoscópica[18].....	59
Fig. 1.1 Cámara Estereoscópica[18].....	60



Fig. 1.1 Cámara Estereoscópica[18]..... 60
Fig. 1.1 Cámara Estereoscópica[18]..... 61
Fig. 1.1 Cámara Estereoscópica[18]..... 62
Fig. 1.1 Cámara Estereoscópica[18]..... 63
Fig. 1.1 Cámara Estereoscópica[18]..... 64
Fig. 1.1 Cámara Estereoscópica[18]..... 64
Fig. 1.1 Cámara Estereoscópica[18] 65
Fig. 1.1 Cámara Estereoscópica[18] 65
Fig. 1.1 Cámara Estereoscópica[18]..... 66
Fig. 1.1 Cámara Estereoscópica[18]..... 67
Fig. 1.1 Cámara Estereoscópica[18]..... 67
Fig. 1.1 Cámara Estereoscópica[18] 68
Fig. 1.1 Cámara Estereoscópica[18]..... 69
Fig. 1.1 Cámara Estereoscópica[18]..... 70



Nomenclatura

Escalares

d	: disparidad sin corrección de distorsión[kdu]
d_k	: disparidad con corrección de distorsión [kdu].
c_0, c_1	: parámetros intrínsecos de la cámara de profundidad kinect.
Z_d	: profundidad con corrección de distorsión[mm]
Lim Min	: distancia kinect-parra
Lim Max	: distancia kinect-fondo
Lim Medio	: distancia kinect-espacio entre parra y fondo
umbral	: parámetro entre 0 y 1 que define la binarización de la imagen.
inicio	: parámetro que define posición de inicio en escaneo de la imagen.

Vectores

P_{rgb}	: posición proyectada de la esquina de la imagen RGB Kinect.
Pi_{rgb}	: proyección $\tilde{P}i_{rgb}$ considerando distorsión RGB Kinect $[px]Pi_{rgb} = [u_{i_{rgb}}, v_{i_{rgb}}]^T$
P_C	: punto en el sistema de coordenadas mundo, $P_C = [X_C, Y_C, Z_C]^T$
$\tilde{P}i_{rgb}$: proyección sobre el plano imagen cámara color, $\tilde{P}i_{rgb} = \begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \end{bmatrix} = \frac{f}{Z_C} \begin{bmatrix} X_C \\ Y_C \end{bmatrix}$
$\delta\tilde{P}^r_{i_{rgb}}$: distorsión radial cámara color Kinect $\delta\tilde{P}^r_{i_{rgb}} = \begin{bmatrix} \delta\tilde{u}^r_{i_{rgb}} \\ \delta\tilde{v}^r_{i_{rgb}} \end{bmatrix}$
$\delta\tilde{P}^t_{i_{rgb}}$: distorsión tangencial cámara color Kinect $\delta\tilde{P}^t_{i_{rgb}} = \begin{bmatrix} \delta\tilde{u}^t_{i_{rgb}} \\ \delta\tilde{v}^t_{i_{rgb}} \end{bmatrix}$
$P_{i_{rgb}}$: proyección considerando distorsión cámara color, $P_{i_{rgb}} = \begin{bmatrix} u_{i_{rgb}} \\ v_{i_{rgb}} \end{bmatrix}$
f_{rgb}	: distancias focales en píxeles, $f_{rgb} = [f_{u_{rgb}}, f_{v_{rgb}}]^T$
cc_{rgb}	: punto principal de la cámara RGB del Kinect en píxeles. $cc_{rgb} = [u_{0_{rgb}}, v_{0_{rgb}}]^T$
t	: traslación desde coordenadas mundo a coordenadas cámara de color.

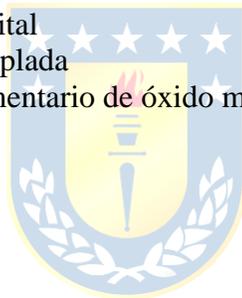
Matrices

R	: matriz de rotación de dimensión 3:3.
D_δ	: patrón de distorsión espacial de disparidad de dimensión 640:480.

Abreviaciones

Mayúsculas

3D	: tres dimensiones.
2D	: dos dimensiones.
ToF	: time of flight.
RGB-D	: rojo-verde-azul-profundidad(red-green-blue-depth).
RGB	: rojo-verde-azul(red-green-blue).
IR	: infrarrojo.
HMI	: Human Machine Interface.
RAW	: formato digital de imágenes.
HRI	: Human robot interaction.
IEEE	: Instituto de Ingenieros Eléctricos y Electrónicos.
DIE	: Departamento de Ingeniería Eléctrica.
UDEC	: Universidad de Concepción.
R	: rotación.
VGA	: adaptador grafico de video
Hz	: Hertz
ADC	: convertidor analogo digital
CCD	: dispositivo de carga acoplada
CMOS	: semiconductor complementario de óxido metálico



Minúsculas

f	: distancia focal.
cc	: eje óptico.
alpha	: coeficiente de torsión.
k	: coeficiente de distorsión.
du,dv	: factores de conversión pixel-milímetros.
su	: factor de escala.
t	: traslación.
kdu	: unidades de disparidad kinect.
fps	: frames por segundo
cm	: centimetro
mm	: milimetro

Capítulo 1. Introducción

1.1. Introducción General

Dentro de la Ingeniería Electrónica existe una diversidad de técnicas que son ocupadas para lograr grandes aportes en relación a avances tecnológicos, con respecto a esto, podemos encontrar fácilmente muchas investigaciones o proyectos relacionados al desarrollo de la visión artificial o visión por computador, ya que su aplicación ha sido muy utilizada en los últimos años, encontrando lugar en distintas áreas de ciencias e ingeniería, tales como la robótica, la representación de estructura 3D, controlen distintas áreas de la industria , por nombrar algunas.

La visión por computador tiene como objetivo programar un computador para que este sea capaz de interpretar y entender una imagen, las características de esta y así analizar en detalle toda la información que puede ser obtenida a partir de un conjunto de vistas de una escena. Pero esta técnica se compone de importantes procesos como adquisición de imágenes, procesamiento de imágenes, análisis de imágenes y representación de estructura 3D, las cuales en este proyecto se irán desarrollando para lograr los objetivos.

Ahora la aplicación de nuestro proyecto va dirigido hacia la agricultura, específicamente a la poda de las parras de uvas. Al igual que varios tipos de árboles frutales, la vid debe ser podada en cierta época del año, de lo contrario, si no se han podado o se ha podado incorrectamente en un lugar inadecuado de la rama, traerá consecuencias negativas, como la mala calidad del fruto, dificultades para madurar y reducción en la producción de la uva. Hoy en día esta operación la llevan a cabo personas que no son formalmente capacitadas.

Con respecto a esto último, es que en este proyecto se utilizará la técnica de visión por computador anteriormente descrito con el objetivo de encontrar el punto de corte óptimo y así ayudar a que el proceso de la poda de las parras se lleve a cabo de una manera correcta. Con el desarrollo y eventuales resultados de este proyecto tendremos a corto plazo una aplicación inmediata en lo que se refiere la poda de las parras, como también una aplicación a largo plazo, ya que este proyecto servirá como base para lograr la automatización total del proceso de la poda.

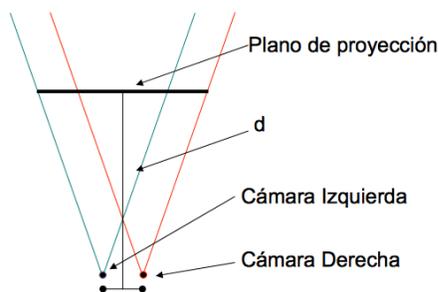
1.2. Técnicas de Reconstrucción 3D.

Uno de los objetivos de la visión artificial es conseguir que un ordenador llegue a analizar una escena real como lo haría una persona. Para conseguir este propósito, es necesario crear un modelo 3D de dicha escena. La reconstrucción tridimensional tiene varias aplicaciones y en esta oportunidad se aprovechará de usar esta herramienta en el área de la agricultura.

Lo que se pretende con este proyecto es conseguir un algoritmo que, a partir de imágenes, llegue a crear en un modelo tridimensional de un objeto, en nuestro caso la parra, esto es uno de los procesos desarrollados en este estudio, además de la creación del algoritmo que identifique el punto óptimo de poda en las imágenes 2D. Para conseguir el objetivo se han estudiado las diferentes técnicas desarrolladas para la reconstrucción 3D que son capaces de reproducir modelos muy exactos y precisos, para conocer las posibilidades existentes y a modo de introducción se verán las principales técnicas.

1.2.1 Cámaras Estereoscópicas.

La visión binocular humana, produce dos imágenes, una para cada ojo que aun siendo parecidas, tienen distintos ángulos, que luego se mezclan en el cerebro creando la imagen 3D. Estas cámaras, intentan imitar este comportamiento, utilizando dos objetivos o dos cámaras separadas estratégicamente, captando así la fotografía en el mismo instante. Con esto haciendo relaciones entre fotografías y determinando la profundidad de cada pixel es posible llegar a hacer una reconstrucción 3D de la escena.



(a)



(b)

Fig. 1.1 Cámara Estereoscópica[18].

(a) Dos objetivos; (b) Cámara estéreo.

1.2.2 Cámaras Time of Flight (ToF).

Este tipo de cámaras además de capturar la imagen RGB, calculan la distancia a la que se encuentra el objetivo en cada pixel, esto mediante la medición del tiempo de retardo o el tiempo que se demora un haz de luz enviado desde la cámara hacia el objetivo.

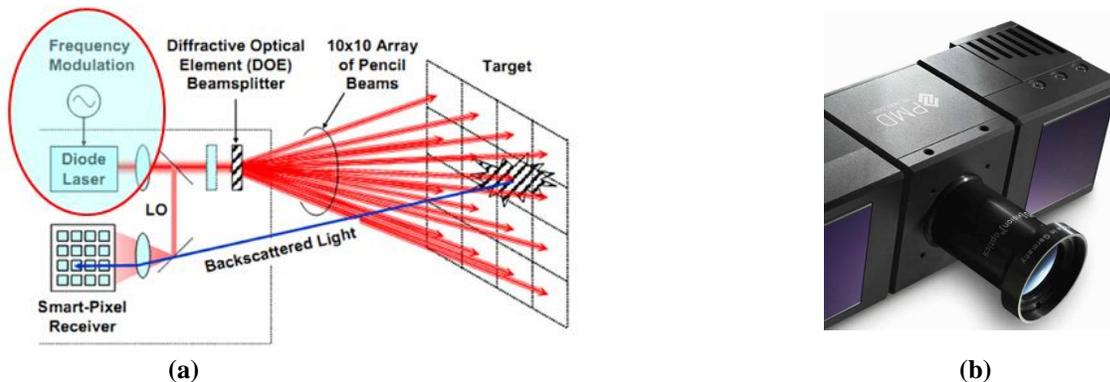


Fig. 1.2 Cámaras Time of Flight (ToF)[18].

(a) Principio de funcionamiento (b) Cámara Time of Flight.

1.2.3 Cámaras basadas en luz estructurada.

Para conocer la distancia a la que se encuentra cada pixel de la imagen de profundidad en este tipo de cámaras se emite una constelación de puntos con el emisor infrarrojo. Así la cámara infrarroja detecta esta constelación y la cámara calcula la disparidad para cada pixel o la diferencia entre donde estaba el punto al proyectarlo a donde está en la proyección. A esto se le llama cámara de luz estructurada.



Fig. 1.3 Cámaras basadas en luz estructurada[18].

(a) Principio de funcionamiento. (b) Kinect.

1.3. La Poda de Parras.

La poda, en sentido amplio, consiste en la supresión total o parcial de sarmientos, hojas, racimos u otras partes vegetativas de la planta. Se puede dividir en dos grandes grupos: poda en seco y poda en verde. La primera se realiza durante la fase de reposo vegetativo y la segunda durante el periodo de actividad de la planta. La poda en seco o de invierno consiste básicamente en la supresión de sarmientos en reposo. Las operaciones en verde conciernen un mayor número o variedad de actuaciones sobre la planta. En este informe nos concentraremos en las técnicas de poda en seco o en periodo de invierno.

1.3.1 La Parra.

La Fig 1.4 ilustra la estructura de una parra. La cual está formada por el tronco principal, brazos (también llamado cordones o bastones) y los brotes que crecen de los brazos.

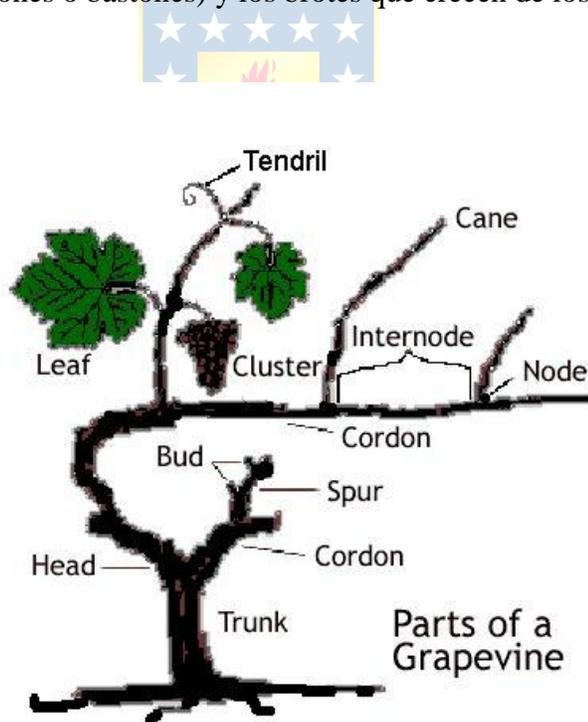


Fig. 1.4 Estructura de la Parra [8]

1.3.2 Poda Segundo brote.

La poda consiste en los cortes de los brotes y bastones. Se refiere principalmente a la poda de los bastones inactivos y los brotes de verano, normalmente llevadas a cabo de forma manual utilizando tijeras de podar y por lo tanto tiene una alta precisión. En este estudio, el método de poda segundo brote se seguirá para localizar la posición de la poda. La poda se puede aplicar en cualquier variedad, pero esta técnica es ideal para las variedades que sean fructíferas en el área basal del brote. Normalmente un espolón de dos yemas producirá dos brotes en la primavera. En la mayoría de los casos, los brotes inferiores se conservan como un estímulo y los mejores canes se retiran, quedando 2 a 3 yemas de los bastones de la temporada pasada, que se puede ver en la Fig. 1.5.

La Fig 1.5 representa la secuencia de corte. Las cuatro imágenes muestran la poda segundo brote en donde funcionó desde el primer año hasta el cuarto año. En el primer año, sólo hay una rama principal por encima del cordón y dos ramas secundarias, que están creciendo en diferentes direcciones. Esta poda se realiza primero en la posición media de la rama principal. A continuación, se opera entre el segundo y tercer espolón en las ramas secundarias. Habrá dos espolones en la rama secundaria después de un año. Durante el segundo año, (Fig 1.5, arriba a la derecha) la rama secundaria del año pasado se ha convertido en una rama principal. En primer lugar, podar en la posición media de la rama de segundo año. Entonces, la operación de poda se llevará a cabo entre el segundo y el tercero espolón de la rama secundaria. Habrá otros dos espolones en la rama secundaria después del tercer año. La operación de poda también se lleva a cabo como se describió anteriormente, como se muestra en la Fig 1.5 (abajo a la derecha y abajo a la izquierda).

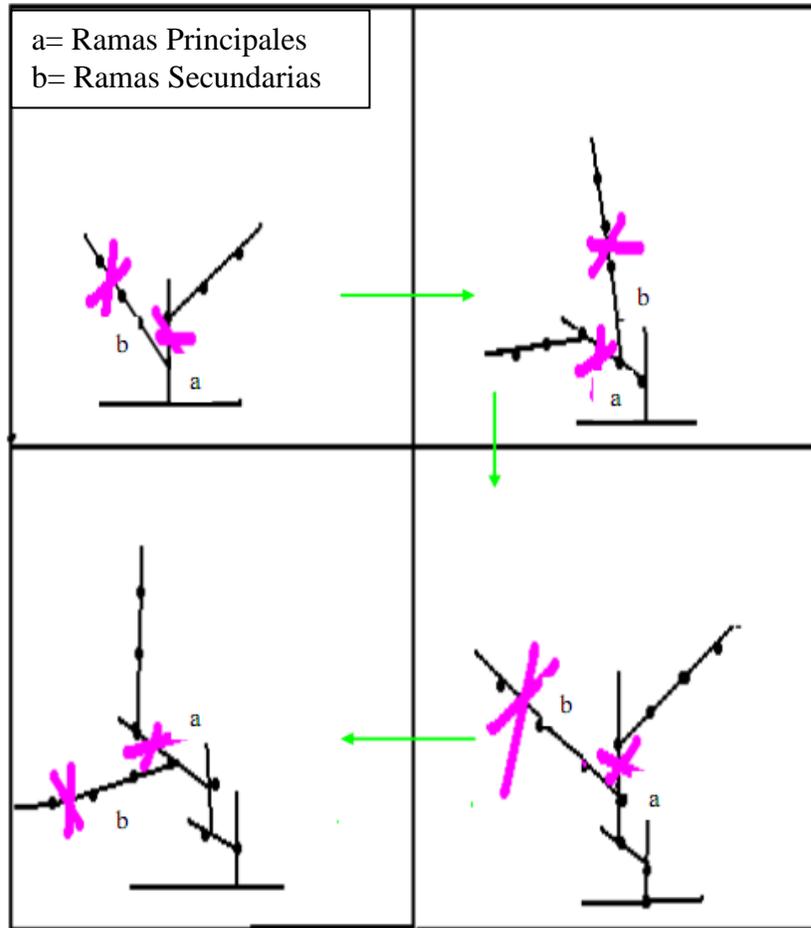


Fig. 1.5 Poda Segundo Brote [8]

1.4. Trabajos Previos

El enfoque de nuestra revisión bibliográfica principalmente está dirigida al campo de la visión por computador y a los respectivos procesos que componen esta técnica. A modo de introducción y para comprender lo que se estudiará en este proyecto, haremos una breve descripción de ciertos procesos y definiremos ciertos términos que se abordarán en el transcurso de esta investigación.

El primer proceso viene dado por la adquisición de imágenes, donde se obtiene la mejor imagen del objeto de estudio, dependiendo de la aplicación, esta puede ser una fotografía, radiografía, etc. El procesamiento de imágenes es otro de estos procesos, y tiene como fin tomar una

imagen y producir una versión modificada de esta, ya sea segmentando la imagen donde generalmente se pretende separar o contrastar el objeto de estudio con el fondo de la imagen. También podemos encontrar el proceso de análisis de imágenes, mediante el cual a partir de una imagen se obtiene una medición y posterior interpretación de ciertos patrones de interés del objeto de estudio. La fase final viene dada por la reconstrucción 3D, esto se logra mediante técnicas como triangulación a partir de las vistas obtenidas, para luego obtener el modelo 3D del objeto de estudio.

1.4.1 Revisión Bibliográfica.

Ya abordando a lo que se refiere revisión bibliográfica, se debe detallar con detención los principales aspectos a analizar dentro de los documentos revisados, para esto, clasificaremos en distintos ítems las referencias en los ítems de visión por computador, Kinect y la poda de parras.

1.4.2 Procesamiento y análisis de imágenes.

- ♣ D. Bulanon, T. Kataoka, Y. Ota, and T. Hirao "A segmentation algorithm for automatic recognition of Fujiapples at harvest," *The United Graduate School of Agricultural Sciences, Biosystems Engineering* (2002), vol., no., pp. 405-412, 14 Aug. 2002, [1].

En este paper se desarrolla un algoritmo para el reconocimiento de las manzanas Fuji para un sistema de recolección robótica. El sistema de visión por computador se compone de un dispositivo de carga y recolección, acoplado con una cámara a color para capturar imágenes de las manzanas y un computador personal para procesar imágenes para el reconocimiento y la localización de la fruta. La imagen de la fruta fue realizada usando la diferencia de color rojo entre los objetos de la imagen, así, la varianza máxima de nivel de gris de la diferencia de color rojo entre la fruta y el fondo determina el umbral óptimo. Los resultados de la segmentación, utilizando este umbral, mostraron resultados de una tasa de éxito de más del 88%, sin embargo la tasa de error fue de más del 18% para las imágenes donde se reflejaba mucha luz en el fondo de la imagen.

- ♣ “Machine visión: theory, algorithms, practicalities” E. Davies.,*2da edicion*, Elsevier Inc. 2005,[2].

Este libro se introduce ampliamente en el campo de la visión artificial, ahondando en lo que se refiere a redes neuronales, análisis y detección de objetos. El autor se encarga de examinar aspectos teóricos y prácticos en lo referente a visión por computador, donde le da un énfasis considerable al análisis de imágenes para superar problemas de oclusión, ruido, distorsión, entre otras. Define el papel que toman las redes neuronales dentro de la visión artificial y proporciona estrategias básicas para trabajar imágenes 2D y 3D.

- ♣ D. Mery. “Visión por Computador,” Departamento de Ciencia de la Computación, Universidad Católica de Chile, Santiago de Chile, Agosto. 2004,[3].

En esta investigación se detalla y se explica todo lo referente a Visión por Computador, donde se describe la teoría que mejor modela la formación de las imágenes. Se habla de geometría proyectiva, donde se enseñan los distintos modelos existentes para establecer la función de transferencia 3D a 2D en un sistema de Visión por Computador y se analizan los algoritmos mas conocidos para llevar a cabo la calibración del sistema. Se muestran los principales algoritmos para hacer una reconstrucción 3D de una escena a partir de sus vistas. Se habla de la técnica Matching y Tracking, en el cual se estudia cómo encontrar la correspondencia entre las distintas imágenes de un objeto. Finalmente se muestran algunas aplicaciones de la Visión por Computador en distintas áreas de la Ingeniería.

- ♣ W. Burger, M. Burge. “Digital ImageProcessing,” ISBN: 978-1-84628-379-6, 2008, pp. 13,[4].

En este trabajo se tratan las variadas técnicas de procesamiento digital de imágenes, como también se describen los procesos de obtención de imágenes, procesamiento y análisis, se introduce en el termino de histograma, definiéndolo primeramente para luego señalar la guía para interpretarlos, además de señalar varios ítems referente a este tema. Con respecto a las imágenes, categoriza cada uno de los formatos, así como también el filtrado, segmentación y detección de los objetivos en las escenas captadas.

- ♣ G. Ming. "ImageProcessing and AnalysisforAutonomousGrapevinePruning",2006 *IEEE, International ConferenceonMechatronics and Automation*), vol., no., pp. 922-927, 25-26 June. 2006,[5].

Esta tesis trata sobre la automatización en la operación de la poda de las parras mediante una maquina robótica que sea capaz de responder ante los cambios en el entorno físico del medio ambiente para realizar una correcta poda. El objetivo de esta investigación es desarrollar un algoritmo utilizando procesamiento y análisis de imágenes para determinar las posiciones de poda y hacer posible la poda automática de la vid. Con el fin de obtener las posiciones a podar con precisión, las imágenes tomadas desde dos cámaras se procesan y se analizan, usando técnicas de visión artificial. En primer lugar, se realiza la fase de procesamiento de imágenes, con el fin de obtener la imagen binaria a partir de la escena original. En segundo lugar, el análisis de imágenes se emplea para identificar las diferentes partes de la vid y obtener las posiciones 2D de los puntos de corte. Se desarrollan nuevos algoritmos para identificar el cordón, la rama y nodos, lo que deriva al último paso donde se determina el punto de poda 3D que es informado a la maquina robótica que realiza la poda.

- ♣ Vincent, L.; Soille, P.; , "Watersheds in digital spaces: anefficientalgorithmbasedonimmersionsimulations," *PatternAnalysis and Machine Intelligence, IEEE Transactionson* , vol.13, no.6, pp.583-598, Jun 1991, [11].

Con respecto a la transformada deWatershed en este paper se propone un nuevo método de cálculo de dicha transformada para imágenes digitales en escala de grises y así disminuir la sobresegmentación. Se prueba la efectividad de este método en base a la comparación de algoritmos existentes con diferentes ejemplos de un algoritmo programado en un lenguaje similar a C.

1.4.3 Reconstrucción 3D y sensor Kinect.

- ♣ J. Smisek, M. Jancosek, T.Pajdla. "3D with Kinect". *2011 IEEE International ConferenceonComputerVision (ICCV Workshops)*, pp. 1154-1160, 6-13 Nov. 2011, [6].

En este paper se analiza el sensor Kinect como un dispositivo de medición 3D. Se propone un modelo geométrico y su procedimiento de calibración. El modelo geométrico de la cámara RGB y la cámara IR son análogos y corresponden al que utilizan los procedimientos de calibración para cámaras digitales, por ello su descripción no se detalla en este trabajo. El sensor de profundidad del dispositivo Kinect es asociado a la geometría de la cámara IR, para lo cual se presenta un modelo inverso de profundidad a lo largo del eje z por cada pixel de la cámara IR. Además se hace una comparación cuantitativa de precisión con un sistema de estero visión y una cámara 3D, siendo el sensor Kinect el de mayor precisión de acuerdo al análisis experimental realizado.

- ♣ P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. “RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments”. *The International Journal of Robotics Research*, 2012, [9].

En esta investigación se habla específicamente de los sensores que captan imágenes RGB, además de la información de profundidad por pixel, dichos dispositivos conocidos también como cámaras RGB-D, como lo es el sensor Microsoft Kinect. Se investiga como estas cámaras se pueden utilizar para construcción de mapas 3D en ambientes interiores. Señala la aplicación que tienen estos sistemas tanto como en la navegación de robots, manipulación y tele presencia. Se presenta una reconstrucción 3D que utiliza un nuevo algoritmo de optimización conjunta, que combina características visuales y de alineación basado principalmente en la forma de los objetos de la escena. Se evalúa y se compara el mapa RGB-D en dos ambientes interiores, donde se combina eficazmente la información de la visual y la forma desde dichas cámaras.

- ♣ K. Khoshelham, “Accuracy analysis of Kinect depth data”, *Remote Sensing and Spatial Information Sciences, ISPRS Calgary 2011 Workshop*, vol., no., pp.133-138, Aug.2011, [10].

Este artículo presenta una investigación de la calidad geométrica de los datos de profundidad obtenidos por el sensor Kinect. Basado en el modelo matemático de la medición de la profundidad por el sensor se presenta un análisis de error teórico, que proporciona una visión de los factores que influyen en la exactitud de los datos. Los resultados experimentales muestran que el error aleatorio de profundidad medida aumenta con el incremento de la distancia del sensor, además se presenta un modelo matemático para la obtención de las coordenadas de objetos 3D, a partir de las mediciones

de imagen RAW, y se discuten los parámetros de calibración que intervienen en el modelo.

- ♣ U. Dugan, R. Dongseok, K. Myungjoon, "Multiple Intensity Differentiation for 3-D Surface Reconstruction With Mono-Vision Infrared Proximity Array Sensor," *Sensors Journal, IEEE*, vol.11, no.12, pp.3352-3358, Dec. 2011, [12].

Este paper trata de la tecnología de los sensores de profundidad, recalcando que tienen un gran potencial para el avance tecnológico en diversas áreas de aplicación como el sector industrial, médico y sistema de visión exterior. Da a conocer las ventajas del dispositivo Microsoft Kinect, en donde censando la profundidad de los movimientos del cuerpo humano se pueden implementar entradas de gestos intuitivos para interfaz HMI (Human Machine Interface), así como también HRI (Human robot interaction) en un nivel más avanzado. Además se compara el desarrollo propuesto con un sistema Mono-Vision basado en un sensor de proximidad infrarrojo.

- ♣ Abramov, A.; Pauwels, K.; Papon, J.; Worgotter, F.; Dellen, B.; , "Depth-supported real-time video segmentation with the Kinect," *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, vol., no., pp.457-464, 9-11 Jan. 2012. [13]

En este trabajo se presenta un método para la segmentación utilizando un algoritmo en paralelo implementado en una GPU. Se aprovecha las características de la cámara RGB y de profundidad que proporciona Microsoft Kinect para realizar las medidas correspondientes y así mejorar en gran medida el proceso de segmentación, obteniendo resultados satisfactorios al probar el método en una escena de humanos manipulando diversos objetos. Este método puede ser de gran ayuda a la hora de implementar un sistema de visión para robot.

1.4.4 Calibración Kinect.

- ♣ C. Hernández, "Desarrollo de Procedimientos Calibración de Escáner 3D de Luz Infrarroja Estructurada para Caracterización de Distribución de Tamaño Granulométrico en Correas Transportadoras", Memoria de Título, Ingeniero Civil Electrónico, Marzo 2013, Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Concepción, [7].

En esta tesis principalmente se hace énfasis al sensor Kinect, siendo este trabajo muy útil para la calibración de este, referente al dispositivo explica además su principio de funcionamiento y la arquitectura que lo compone. También hace reseña a la calibración de las cámaras RGB, la obtención de los parámetros que definen las condiciones de la formación de la imagen, conocidos como parámetros intrínsecos y extrínsecos, los cuales detalla los pasos para la identificación de estos parámetros. Todo esto, que son pasos previos, lo realiza a modo de introducción para entender posteriormente el mecanismo de calibración de la Kinect, con esto el objetivo general de esta tesis es determinar un procedimiento y algoritmo de calibración que permita asegurar la calidad de las medidas de una cámara Microsoft Kinect para una posterior caracterización de tamaño granulométrico.

- ♣ D. Herrera, J. Kannala, and J. Heikkilä. “Accurate and Practical Calibration of a Depth and Color Camera Pair,” Proceedings of the 14th International Conference, CAIP 2011, Part II, vol. 6855, pp. 437-445, 29-31 Aug. 2011, [14].

En este trabajo se presenta un algoritmo que calibra simultáneamente una cámara de color, una cámara de profundidad y la relativa posición entre ellas. Se realiza la calibración en conjunto, debido a que llevar a cabo la calibración de manera independiente, esto puede provocar que los parámetros intrínsecos y extrínsecos no sean los óptimos. El método requiere solamente una superficie plana para la obtención de imágenes desde diferentes orientaciones. Consiste en el modelo de pin-hole con corrección de distorsión radial y tangencial. Para el sensor de profundidad se presenta una ecuación que relaciona el valor de disparidad con la profundidad. Se llevaron a cabo experimentos con el sensor de profundidad y se logró la misma precisión que el procedimiento de calibración del fabricante.

- ♣ D. Herrera, J. Kannala, and J. Heikkilä. “Joint Depth and Color Camera Calibration with Distortion Correction,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.34, no.10, pp. 2058-2064, Oct. 2012, [15].

Este paper es de los mismos autores que el paper [14], en donde al igual que en el paper anterior se presenta un algoritmo que calibra simultáneamente las cámaras y la posición relativa entre ellas, pero la diferencia es que en esta oportunidad agrega una cámara de color externa de alta

resolución, además de la cámara de profundidad que viene incluida con el sensor. El motivo para agregar una segunda cámara RGB es para obtener una calibración aun más exacta y precisa. El método de calibración hace una mejora en la distorsión de la información de profundidad entregada por el sensor. En esta oportunidad los resultados obtenidos de los experimentos mostraron indiscutibles mejoras y se logró una mayor precisión con respecto a la calibración del fabricante.

- ♣ Daniel Herrera. Center for Machine Vision Research, University of Oulu. [Online] <http://www.ee.oulu.fi/~dherrera/>, [16].

En esta pagina web se describe la herramienta de matlab “Kinect Calibration Toolbox”, para la calibración de la kinect ya vista en los paper [14] y [15] respectivamente. Desde esta página web se puede descargar el complemento o la herramienta para matlab, así como también una descripción e instrucción detallada de la calibración.

- ♣ M. Saiz, “Reconstrucción Tridimensional Mediante Visión Estéreo y Técnicas de Optimización”, Proyecto Fin de Carrera, Ingeniero Industrial, Junio 2010, Escuela Técnica Superior de Ingeniería (ICAI), Universidad Pontificia Comillas, [17].

En esta tesis se hace un estudio de la reconstrucción tridimensional de objetos y escenas, esto a través de un mecanismo análogo al que se produce en el ojo humano. A partir de dos imágenes en estéreo, se determinan los dos puntos en cada una de esas imágenes que se corresponden con el mismo punto en la realidad, estos dos puntos se llaman puntos correspondientes. Este proyecto es de mucha utilidad en uno de nuestros procesos, ya que hace énfasis especial en el proceso de calibración de la cámara, define el proceso mediante el cual se calculan los parámetros intrínsecos y extrínsecos de la cámara, a partir de un conjunto de puntos de control, conocidas las coordenadas tridimensionales de esos puntos y midiendo las correspondientes coordenadas de imagen, en la imagen obtenida con dicha cámara.

1.4.5 La poda.

- ♣ G. Perez. “Operaciones Manuales en Viñedo,” *Centros de Formacion Agraria Castilla y*

Leon, Dec. 2005, [8]

El presente manual esta enfocado para ser una herramienta o una guía para que las personas conozcan conceptos y características básicas de la vid, como también proporciona conocimientos técnicos para realizar procesos avanzados como la poda en los viñedos. Comienza con una introducción en la anatomía practica básica de la vid, en el cual se define y explica cada parte de la estructura de la vid, se describe el proceso de formación de esta y por último con respecto a la poda, deja claro las diferencias de podar en seco y en verde. También da a conocer los diferentes métodos y criterios que se usan para realizar la poda. Este material es de gran ayuda para este proyecto, ya que proporciona los conocimientos suficientes para conocer los procesos de la vid.

1.4.6 Discusión

Luego de haber estudiado y revisado investigaciones, tesis, papers, etc. Podemos ver la posibilidad cierta de concretar la idea y el objetivo de nuestro proyecto, identificar los puntos de corte óptimo para la poda de las parras usando técnicas de procesamiento de imágenes. Se logra identificar el problema de que cada vez en la agricultura existe una escasez de personas que realicen de manera correcta y adecuada este trabajo, en este sentido los viñedos ven una problemática con respecto a este proceso, ya que cada vez hay menos expertos. Por medio de este problema es donde nace la oportunidad de poder cubrir ese vacío en el cual no hay personas que lo puedan cubrir, también bajo la necesidad de crear un sistema de ayuda para que las personas que lo realizan lo hagan de forma adecuada. Aprovechando las características positivas del sensor Kinect, con una buena calibración es posible hacer mediciones precisas y mejor que otros dispositivos visuales, esto unido al procesamiento y posterior análisis de las imágenes se logrará hacer la reconstrucción 3D.

En este proyecto se analizarán las técnicas de visión por computador y algoritmos de procesamiento de imagen para la búsqueda del punto óptimo mediante un set-up experimental que se implementará en el laboratorio, esto constará de la conexión, calibración e instalación de los respectivos dispositivos a utilizar, como también la sujeción de parras reales a pedestales que harán que esta tenga su forma natural y asemeje la situación real de una planta plantada, con esto se logrará llevar a cabo la tarea de mejor forma.

1.5. Hipótesis de Trabajo

Mediante un algoritmo basado en técnicas de visión por computador es posible realizar una correcta identificación de los puntos de poda en las vides utilizando los avances tecnológicos actuales, como lo es el sensor Kinect.

1.6. Objetivos

1.6.1 Objetivo General

Desarrollar una metodología para procesamiento de imágenes, que permita identificar los puntos de poda óptimos en las parras de uva.

1.6.2 Objetivos Específicos

- Desarrollar un algoritmo capaz de identificar los puntos de poda en las parras de uva para aportar con los avances tecnológicos en la agricultura.
- Diseñar set-up experimental.
- Creación de interfaz gráfica de usuario.
- Evaluar la efectividad del algoritmo desarrollado



1.7. Alcances y Limitaciones

- La adquisición de las imágenes se hará utilizando Kinect y el procesamiento de imágenes se realizará a través del software Matlab.

- Se utilizarán plantas de parra reales para las pruebas.

- La exactitud de los datos estará sujeta a las limitaciones técnicas del dispositivo Microsoft Kinect.

1.8. Temario y Metodología

Se implementará una estructura que se compone de parras reales acomodadas en pedestales, frente a estas se posicionará la Kinect a modo de asegurar la obtención de las mejores imágenes, además para poder segmentar la imagen en el procesamiento se instalará un telón de manera que se pueda contrastar la parra con el fondo de la imagen, tal como se muestra en la Fig. 1.6.



Fig. 1.6Set-up experimental



Capítulo 2. Adquisición de Imágenes

2.1. Introducción

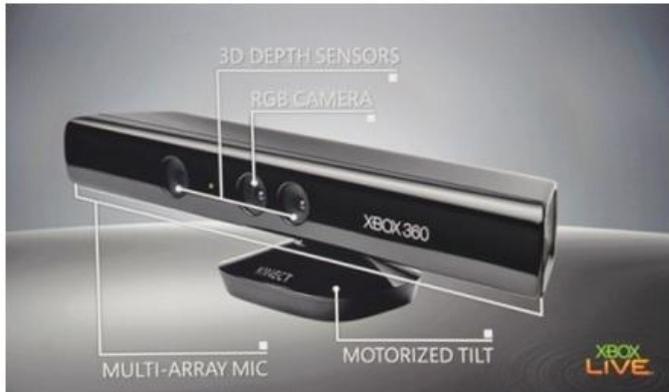
Hasta aquí se ha llevado a cabo la introducción al tema principal del proyecto y la revisión bibliográfica, abarcando todos los temas relacionados con este proyecto. Ahora, en este capítulo se hará una revisión al principio de funcionamiento del sensor que utilizaremos y a grandes rasgos se procederá a explicar el método de calibración, que para efectos de procesamiento y análisis de imagen es un proceso primordial a la hora de obtener una data lo más representativa posible del entorno. Esta data, una vez adquirida a través de las imágenes capturadas por la cámara, se procesan en Matlab.

2.2. Sensor Kinect.

El sensor de Kinect es una barra horizontal de aproximadamente 23 cm conectada a una pequeña base circular con un eje de articulación de rótula, y está diseñado para ser colocado longitudinalmente por encima o por debajo de alguna plataforma. Su creación estaba destinada únicamente a su utilización junto a la videoconsola "Xbox 360" y para extraer la información tan valiosa que obtiene, hubo que recurrir a la ingeniería inversa. Con este proceso se pudo empezar a utilizar el dispositivo con ordenadores para diferentes aplicaciones como en el reconocimiento de objetos.

Las partes básicas de Kinect como podemos ver en la Fig. 2.1, son:

- Una cámara tradicional (Resolución 640x480 RGB 30fps VGA).
- Un emisor de infrarrojos.
- Una cámara de infrarrojos.
- 4 Micrófonos (16bit sampling rate: 16Hz).
- Un motor.



(a)



(b)

Fig. 2.1 Microsoft Kinect [27]

(a) Partes que componen la Kinect; (b) Kinect desmontada

2.2.1 Principio de Funcionamiento

El sensor Kinect es capaz de capturar el mundo que lo rodea en 3D mediante combinación de la información obtenida de la cámara RGB y del sensor de profundidad. El resultado de esta combinación es una imagen RGB-D (color + profundidad) con una resolución de 640x480, donde a cada uno de los píxeles se le asigna una información de color y de profundidad.

Cámara RGB

El funcionamiento de la cámara RGB que monta la Kinect es como el de una cámara digital estándar. La luz atraviesa un lente que la dirige a un filtro encargado de separarla en los colores primarios, los cuales son proyectados sobre un sensor fotosensible. Este sensor genera una señal eléctrica en función de la intensidad de la señal que incide sobre él. Posteriormente, esta señal es convertida a digital mediante un ADC (Analog Digital Convert), que más tarde es analizada y reconstruida para su almacenamiento. Esto se consigue gracias a la interpolación, que permite rellenar aquellos espacios en los que falta información.

El problema del sensor es que no distingue los colores, sino variaciones de intensidad, por tanto para obtener una imagen en color es necesario descomponer la imagen en los colores primarios (rojo, verde y azul). Estos son proyectados sobre distintas zonas del sensor, el cual

reconoce la cantidad de intensidad de cada uno de ellos por separado, como se puede observar en la Fig. 2.2.

Estos sensores según que tecnología empleen se clasifican en CCD (Charge Couple Device) y CMOS (Complementary Metal Oxide Semiconductor). La diferencia viene dada por la forma en la que es transmitida la información.

En el caso de los CCD, esta se envía a los extremos del sensor digital y de ahí al ADC, mientras que en el CMOS los valores se conducen directamente en formato digital, por lo que no precisa de un ADC.

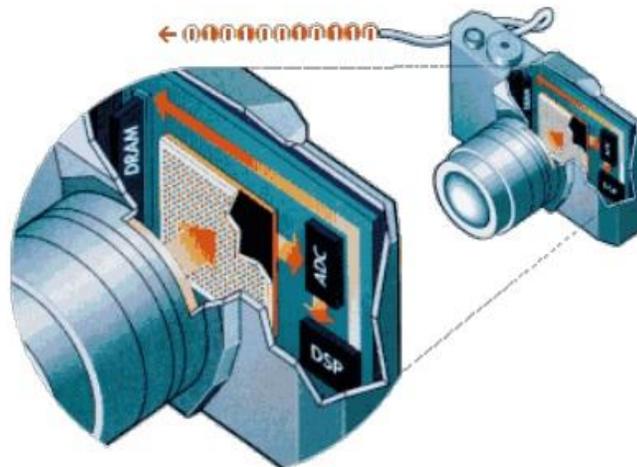


Fig. 2.2 Funcionamiento interno de la cámara digital [27]

Sensor de Profundidad

Por su parte, el sensor de profundidad está formado por dos componentes: un proyector de luz infrarroja (IR) y un sensor CMOS monocromo estándar. Ambos se encuentran alineados a lo largo del eje X del dispositivo, a una distancia (denominada “línea base”) de 75mm, con ejes ópticos paralelos, ver Fig 2.1. Esta disposición dentro de Kinect facilita los cálculos de profundidad, que se basan en un principio similar al de triangulación activa entre emisor y cámara, esto es, entre los rayos visuales de los puntos proyectados y sus correspondientes proyecciones en la imagen.

A diferencia de los métodos tradicionales de triangulación activa, los desarrolladores de la tecnología de la cámara de rango presente en Kinect proponen una técnica ingeniosa para la obtención de información 3D, denominada Codificación de Luz (Light Coding)[27]. La idea principal consiste en un proceso en dos fases, una primera de calibración, y otra de funcionamiento. En la fase de calibración, se emplea el proyector de luz infrarroja para proyectar un patrón de puntos sobre un plano de la escena, variando su distancia entre posiciones conocidas. A su vez, la cámara captura una imagen del patrón proyectado sobre el plano para cada una de estas distancias. Las imágenes obtenidas se denominan imágenes de referencia y se almacenan en el sensor.



Fig. 2.3Ejemplo de patrón de puntos proyectado [28]

En la fase de funcionamiento se emplean las imágenes de referencia para sustituir “virtualmente” al emisor del patrón IR, de tal manera que para cada nueva imagen capturada por el sensor, el cálculo de profundidad se resume a un problema de visión estéreo con configuración ideal: cámaras idénticas, ejes alineados y separados una distancia base de 75 mm En cuanto al error cometido por las mediciones, este es menor a los 10cm a distancias superiores a los 4m, y menor a los 2cm en mediciones inferiores a los 2.5m.

Entonces la cámara infrarroja detecta esta constelación y Kinect calcula la disparidad para cada píxel (la diferencia entre donde estaba el punto al proyectarlo a donde está en la proyección). A esto se le llama cámara de luz estructurada.



Fig. 2.4 Imágenes Kinect [28]
(a) Imagen de disparidad; (b) Imagen RGB

2.3. Calibración Cámara Microsoft Kinect.

2.3.1 Parámetros Intrínsecos cámara RGB

Se utiliza un modelo intrínseco, en donde la proyección de un punto de coordenadas cámara de color $\mathbf{P}_{rgb} = [X_{rgb}, Y_{rgb}, Z_{rgb}]^T$ hacia coordenadas imagen color $\mathbf{P}_{i_{rgb}} = [u_{i_{rgb}}, v_{i_{rgb}}]^T$ es obtenida mediante el siguiente análisis de ecuaciones.

Primeramente definimos que el punto transformado desde el sistema de coordenadas del objeto al sistema de coordenadas de la cámara esta denotado por \mathbf{P}_C . Con lo cual, es necesario transformar ahora este punto $\mathbf{P}_C = [X_C, Y_C, Z_C]^T$, desde el sistema de coordenadas de la cámara al plano imagen, mediante la proyección de este punto sobre dicho plano, Fig. 2.5.

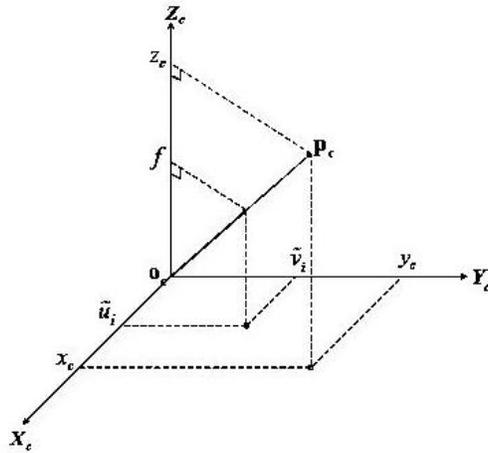


Fig. 2.5 Proyección del punto P_c sobre el plano de la imagen

Matemáticamente la proyección queda definida por la ecuación (2.1).

$$\tilde{\mathbf{P}}_{i_{rgb}} = \begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \end{bmatrix} = \frac{f}{Z_c} \begin{bmatrix} x_c \\ y_c \end{bmatrix} \quad (2.1)$$

Utilizamos esta ecuación de proyección, normalizada por la distancia focal:

$$\tilde{\mathbf{P}}_{i_{rgb}} = \begin{bmatrix} \tilde{u}_{i_{rgb}} \\ \tilde{v}_{i_{rgb}} \end{bmatrix} = \frac{1}{Z_{rgb}} \begin{bmatrix} x_{rgb} \\ y_{rgb} \end{bmatrix} \quad (2.2)$$

Luego, se aplica sobre el punto anterior el modelo de distorsión, la cual como se sabe este modelo de distorsión de una cámara color tiene dos componentes: una radial y otra tangencial. Así, la componente de distorsión radial está dada por:

$$\delta \tilde{\mathbf{P}}_{i_{rgb}}^r = \begin{bmatrix} \delta \tilde{u}_{i_{rgb}}^r \\ \delta \tilde{v}_{i_{rgb}}^r \end{bmatrix} = \left\{ k_1 r_{i_{rgb}}^2 + k_2 r_{i_{rgb}}^4 + k_5 r_{i_{rgb}}^6 \right\} \tilde{\mathbf{P}}_{i_{rgb}} \quad (2.3)$$

La componente de distorsión tangencial está dada por:

$$\delta \tilde{\mathbf{P}}_{i_{rgb}}^t = \begin{bmatrix} \delta \tilde{u}_{i_{rgb}}^t \\ \delta \tilde{v}_{i_{rgb}}^t \end{bmatrix} = \begin{bmatrix} 2k_3 \tilde{u}_{i_{rgb}} \tilde{v}_{i_{rgb}} + k_4 (r_{i_{rgb}}^2 + 2\tilde{u}_{i_{rgb}}^2) \\ k_3 (r_{i_{rgb}}^2 + 2\tilde{v}_{i_{rgb}}^2) + 2k_4 \tilde{u}_{i_{rgb}} \tilde{v}_{i_{rgb}} \end{bmatrix} \quad (2.4)$$

$$r_{i_{rgb}} = \sqrt{\tilde{u}_{i_{rgb}}^2 + \tilde{v}_{i_{rgb}}^2} \quad (2.5)$$

donde,

$\mathbf{k}_{rgb} = [k_1, \dots, k_5]^T$:vector que contiene los coeficientes de distorsión.

Aplicando la corrección de distorsión radial y tangencial, el punto proyectado queda como:

$$\mathbf{P}_k = [u_k, v_k]^T = \tilde{\mathbf{P}}_{i_{rgb}} + \delta\tilde{\mathbf{P}}_{i_{rgb}}^r + \delta\tilde{\mathbf{P}}_{i_{rgb}}^t \quad (2.6)$$

Luego, usando la ecuación (2.19), se tiene que las coordenadas de la imagen están dadas por:

$$\mathbf{P}_{i_{rgb}} = \begin{bmatrix} u_{i_{rgb}} \\ v_{i_{rgb}} \end{bmatrix} = \begin{bmatrix} f_{u_{rgb}} & 0 \\ 0 & f_{v_{rgb}} \end{bmatrix} \begin{bmatrix} u_k \\ v_k \end{bmatrix} + \begin{bmatrix} u_{0_{rgb}} \\ v_{0_{rgb}} \end{bmatrix} \quad (2.7)$$

donde,

$\mathbf{f}_{rgb} = [f_{u_{rgb}}, f_{v_{rgb}}]^T = [f_{rgb} d_u s_u, f_{rgb} d_v]^T$:distancias focales en píxeles.

$\mathbf{cc}_{rgb} = [u_{0_{rgb}}, v_{0_{rgb}}]^T$: punto principal de la cámara RGB del Kinect en píxeles.

2.3.2 Parámetros Intrínsecos Cámara de Profundidad

Se debe transformar las coordenadas del sistema cámara de profundidad a las coordenadas de profundidad de la imagen. Se aplica un modelo de cámara pin-hole con corrección de distorsión radial y tangencial para el mapa de disparidad[15].

Para obtener un patrón de error fijo en las mediciones de profundidad se debe calibrar el valor de la disparidad, entregado por el Kinect. La profundidad en metros está conformada por dos adaptaciones, un escalamiento de inversión y una corrección de disparidad. La corrección de disparidad es innecesaria cuando se trabaja con las librerías proporcionadas en Linux, debido a que el toolbox posee una corrección interna de la data. El escalamiento inverso se representa por

$$z_d = \frac{1}{c_1 d_k + c_0} \quad (2.8)$$

Donde,

c_1 y c_0 : Parámetros intrínsecos que son resultado del proceso de calibración

d_k : Mapa de disparidad entregada por el dispositivo

Para el desarrollo del algoritmo de visión volumétrica será necesario obtener este dato con la mayor precisión posible, debido a su importancia a la hora de estimar el volumen de un cuerpo. Si lo relacionamos con un error en la coordenada z, éste se amplificará por la superficie que recorra ese error, en otras palabras, si vemos un caso límite en el cual el cuerpo al cual se requiera estimar volumen es relativamente plano (coordenada z del orden 1cm), y las coordenadas x e y del espacio sean de un orden mucho mayor (30cms), un pequeño error en la coordenada z provocará una gran diferencia en la estimación volumétrica correspondiente. Por esta razón de sensibilidad en la coordenada z, la calibración del dispositivo Kinect debe ser lo más precisa posible

2.3.3 Parámetros Extrínsecos Kinect

La Fig. 2.6. muestra los diferentes sistemas de referencia presentes en una escena. $\{C\}, \{D\}$ y $\{E\}$ son los sistemas de referencia de la cámara RGB, profundidad y externa respectivamente. $\{V_i\}$ es el marco de referencia anclado al plano de calibración y $\{W_i\}$ es el marco de referencia del mundo anclado al patrón de calibración.

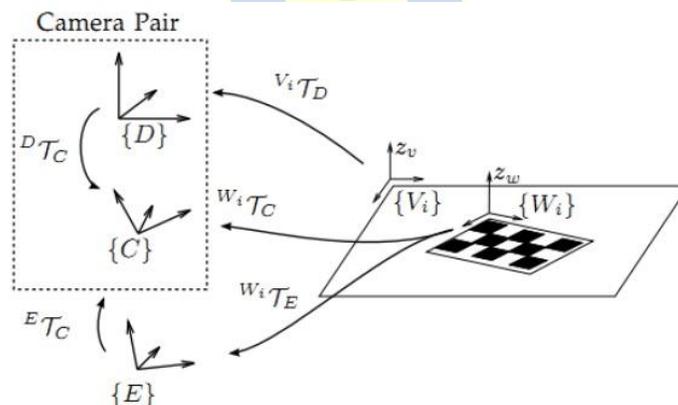


Fig. 2.6 Sistemas de referencia y transformaciones presentes en una escena [13]

Los puntos desde un sistema de referencia pueden ser transformados a otro mediante una transformación rígida, transformación que no cambia el tamaño o forma de los puntos originales, denotada por $\tau = \{\mathbf{R}, \mathbf{t}\}$, donde \mathbf{R} es una rotación y \mathbf{t} una traslación [7]. Por ejemplo, la transformación de un punto \mathbf{P}_w desde el sistema de coordenadas del mundo $\{W\}$ al sistema de coordenadas de la cámara color $\{C\}$ sigue la siguiente estructura:

$$\mathbf{P}_c = \mathbf{R}_c \mathbf{P}_w + \mathbf{t}_c \quad (2.9)$$

La posiciones relativas ${}^D\tau_C$ y ${}^E\tau_C$ son constantes, mientras que cada imagen tiene su propia posición relativa con respecto a la cámara ${}^{W^i}\tau_C$.

Si consideramos los ejes de referencia {E} y {C} constantes podremos obtener los parámetros extrínsecos de la cámara de profundidad y adaptarlos a la calibración mediante una transformación como en la ecuación (2.9).

2.4. Adquisición de Imágenes

Para obtener una eficaz adquisición de imágenes, primero es necesario conectar el sensor Kinect al computador y realizar los pasos de instalación y configuración de este dispositivo, para luego proceder a utilizar el algoritmo de calibración que se encuentra disponible para este proceso y finalmente con esto, estar en óptimas condiciones y de manera confiable adquirir una imagen. Con todo lo precedido hasta aquí en los capítulos anteriores, a continuación se desarrolla lo mencionado anteriormente.

2.4.1 Configuración del sensor con el PC

Como es de conocimiento mundial, existen varios sistemas operativos que permiten al usuario interactuar con el computador, como lo son Windows, Mac Os, Linux y Ubuntu entre otros. En cada uno de estos sistemas es posible instalar drivers del fabricante o librerías de funciones que son necesarias y son el único método para la conexión y configuración del sensor Kinect, con el fin de obtener las imágenes en el computador que las cámaras de este sensor captan del entorno. Sin duda el software más amigable y más utilizado por la mayoría es el sistema operativo Microsoft Windows, que entre sus diversas versiones ofrece una interfaz que permite llevar a cabo todas las tareas y procesos en pasos más sencillos.

El Kinect se puede instalar en Windows mediante drivers OpenNI open source y conectarse a Matlab mediante una librería de funciones en C++, exportadas a través del compilador MEX y con esto se logran visualizar las imágenes entregadas por el sensor, como lo son la imagen RGB y de profundidad. Pero hay un inconveniente mediante este método, ya que la información de profundidad no viene en la data pura del sensor, esto es en unidades de disparidad Kinect (kdu), codificada en 11 bits. Es de suma importancia este tipo de información, sobre todo en el proceso de calibración, ya que usamos la herramienta "Kinect Calibration Toolbox for Matlab" [15], en donde se considera la distorsión de la data de disparidad y por ende la data pura.

Existe otra forma de poder visualizar y obtener las imágenes en su data pura como se desea, mediante el sistema operativo Debian Linux y la librería "libfreenect", además es posible la obtención de estos datos en Matlab mediante un archivo MEX [14]. A continuación, se muestra un esquema con lo dicho hasta relacionado con el método adquisición de imágenes RGB-D desde la Kinect.

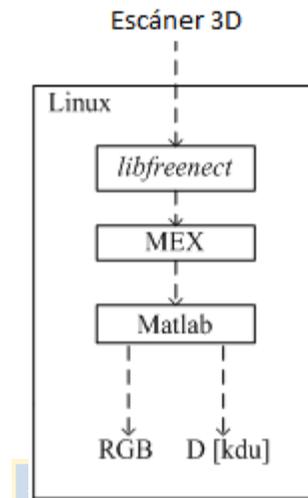


Fig. 2.7 Método de adquisición de imágenes RGB-D

Para detallar todo el proceso de instalación de la librería antes mencionada, primero se detalla la lista de software usados:

- Matlab R2012a para Linux.
- Debian "squeeze" 6.0.5 de 32 bits.

Los pasos para la instalación del sensor kinect en Debian "squeeze", se describen a continuación.

1. Instalar los siguientes paquetes desde una consola como usuario root.

- apt-get install git-core
- apt-get install cmake
- apt-get install libglut3-dev
- apt-get install pkg-config
- apt-get install build-essential
- apt-get install libxmu-dev
- apt-get install libxi-dev
- apt-get install libusb-1.0-0-dev

2. Clonación del repositorio libfreenect.

Nos ubicamos en una carpeta que tenga permisos de escritura y lectura desde consola, luego ubicado en la carpeta destino, realizamos la clonación con el siguiente comando.

- <https://github.com/OpenKinect/libfreenect.git>

3. Ingresar a la carpeta "libfreenect", luego crear una nueva carpeta (build), entrar en la carpeta mediante los siguientes comandos.

- cd libfreenect/

- mkdir build

-cd build/

4. Verificar requerimientos, compilación e instalación de programa, con los siguientes comandos.

- cmake ..

- make

- make install



5. Finalmente para montar las librerías y realizar la conexión con el sensor kinect y así obtener las imágenes, se digitan los últimos comandos.

- ldconfig /usr/local/lib/

- glview

Una vez realizados todos los pasos anteriores se puede notar que la instalación se ha completado con éxito y ya podemos obtener las capturas de imágenes RGB-D que la Kinect obtiene del entorno, en la siguiente imagen notamos lo dicho anteriormente.

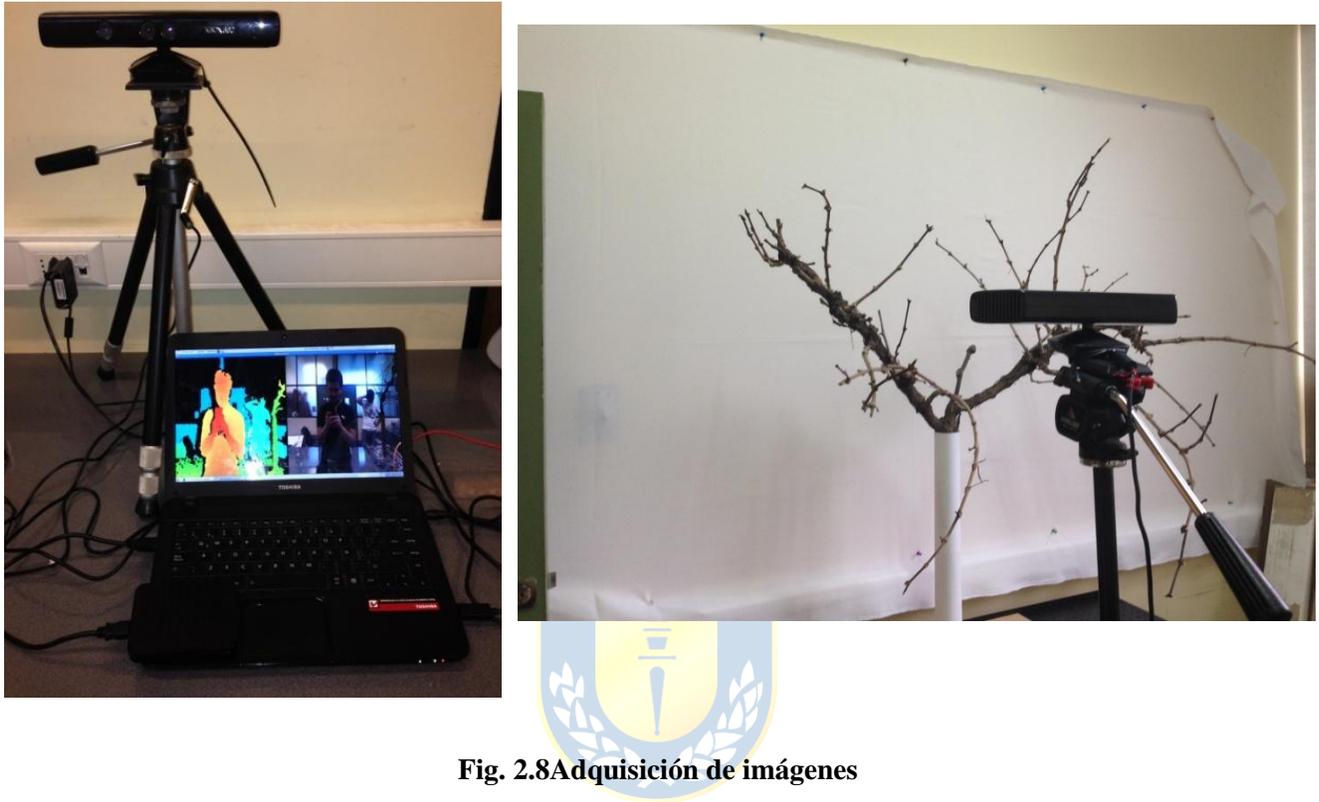


Fig. 2.8 Adquisición de imágenes

Ahora que ya tenemos acceso a la información que entrega el sensor, estamos en condiciones de realizar la calibración del dispositivo, como se ha mencionado anteriormente se utilizó la herramienta "Kinect Calibration Toolbox for Matlab", referente al toolbox desarrollado por Daniel Herrera[15]. Por otro lado, también se implementó un patrón de calibración que es utilizado comúnmente para la calibración de las cámaras RGB. Las medidas de este patrón son de cuadrados de 34mm.

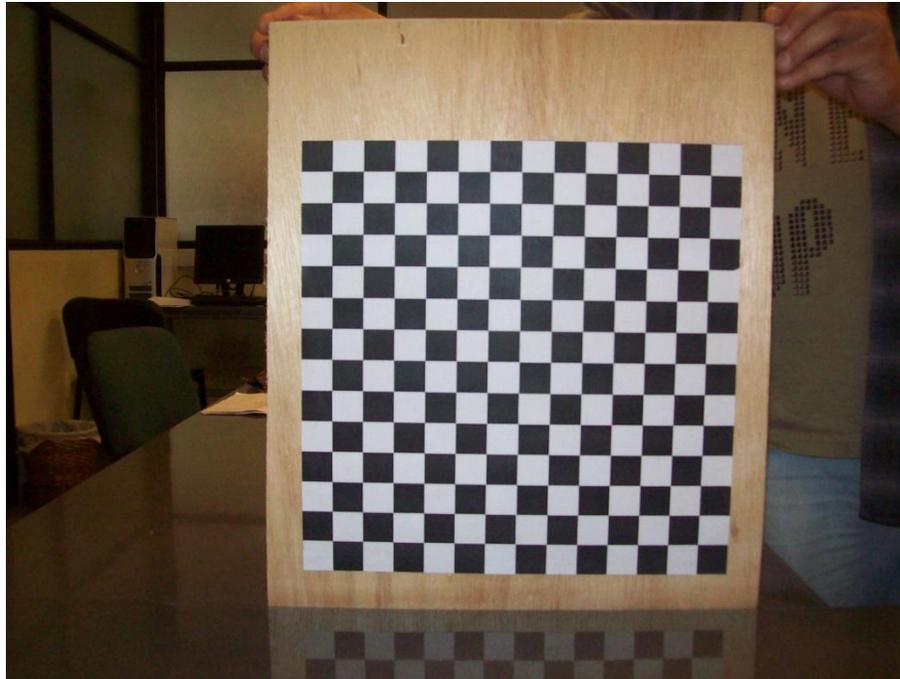


Fig. 2.9 Patrón de calibración para cámaras RGB

Se tomaron varias imágenes a este patrón, aproximadamente alrededor de 35 capturas, esto con el fin de asegurar una buena calibración. Estas imágenes se adquirieron a diferentes distancias, en plano frontal, plano rotado alrededor del eje x e y.

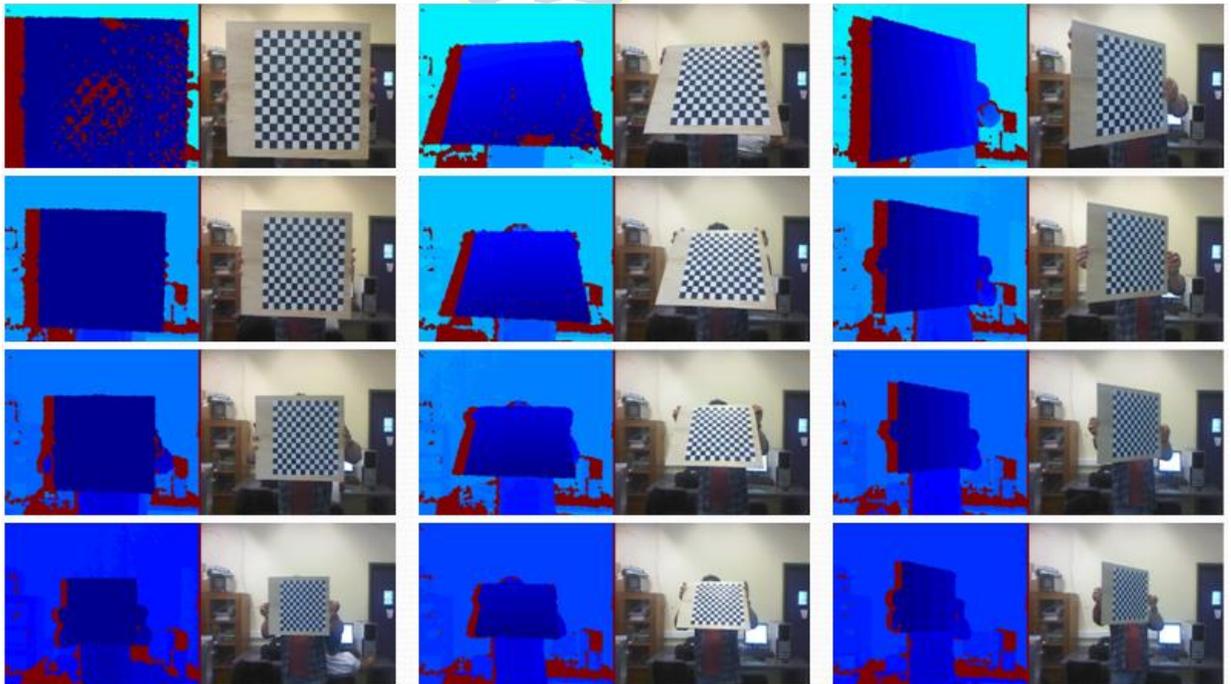


Fig. 2.20 Captura de imágenes RGB-D

La herramienta de calibración para ambas cámaras proporcionada por la interfaz generada en la Memoria de Título de Carlos Hernández[7], además de el toolbox desarrollado por Daniel Herrera[16], nos dan la posibilidad tanto de realizar el proceso de calibración como también de guardar las imágenes que han sido retornadas por la librería "libfreenect". Estas se guardan mediante la función *imwrite* de matlab. Una vez obtenida la calibración en el espacio de interés, se comienza con la etapa de adquisición de imágenes de la parra. Todo el procedimiento de calibración para las cámaras del dispositivo Microsoft Kinect y la utilización de la interfaz de usuario para llevar a cabo esta etapa, son descritas paso a paso en la memoria de título antes mencionada.



Capítulo 3. Algoritmo de Identificación

3.1. Introducción

Para prepararse para la poda de la parra, se deben llevar a cabo los siguientes procedimientos:

- Procesamiento de imágenes: Este paso consiste en preparar la imagen para el posterior análisis.
- Análisis de imágenes: Localizar el cordón, las ramas y los nodos.
- Reconstrucción 3D: Se derivan las coordenadas en tres dimensiones para hacer la reconstrucción de la parra y los datos de interés.

A continuación se describen los métodos utilizados en este estudio para procesar y analizar las imágenes. Después del tratamiento de las técnicas descritas en este capítulo, se obtendrán las posiciones de poda en 2D. La metodología tridimensional y la estructura de la interfaz gráfica se detallan en el capítulo siguiente.

Existen tres pasos principales para localizar las posiciones de poda en dos dimensiones. En primer lugar, las imágenes de la cámara se cargan en el computador. Luego se aplican técnicas de procesamiento de imágenes en donde se puede realizar la transformación de color para procesar la imagen y prepararla. Finalmente, el algoritmo de análisis de imágenes está diseñado para extraer las posiciones de poda. La Fig 3.1 muestra el diagrama de flujo de la fase de análisis 2D antes mencionada.

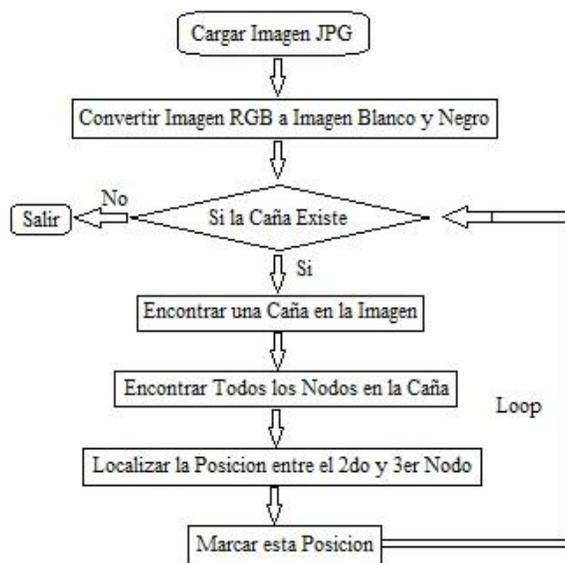


Fig. 3.1 Diagrama de Análisis de Imagen[18].

Para simplificar la discusión, haremos dos supuestos que son razonables en el mundo real y que se han establecido para el estudio metodología presentada en este estudio:

- El fondo se procesa y se simplifica en este estudio para hacer que el algoritmo de análisis de imagen sea más eficiente.
- El nivel de iluminación se supone que es constante.

3.2. Procesamiento de Imágenes.

En la etapa de procesamiento de imágenes, la imagen RGB se convierte a una imagen de escala de grises, y luego la imagen convertida se convierte además en una imagen binaria (blanco y negro) que reduce en gran medida la potencia de cálculo necesaria para su posterior análisis. Específicamente, el algoritmo de análisis de imagen podría distinguir los cordones, las ramas y a partir de entonces los nodos basados en la imagen binaria.

3.2.1 Imagen RGB en escala de grises.

Se utilizó el sensor Kinect, el cual tiene una resolución de 640x480 para capturar imágenes RGB. Las imágenes en color RGB contenían una matriz de $N \times M \times 3$ de píxeles de colores, en donde cada color se ubica en un espacio específico que puede ser representado por un determinado valor de componentes rojo, verde y azul. Esta es una aplicación de un principio óptico en el campo de visión por computador. Este principio establece que cualquier color se puede describir dándole valores a los componentes rojo, verde y azul. Figura 3.2 muestra el modelo de espacio de color RGB.

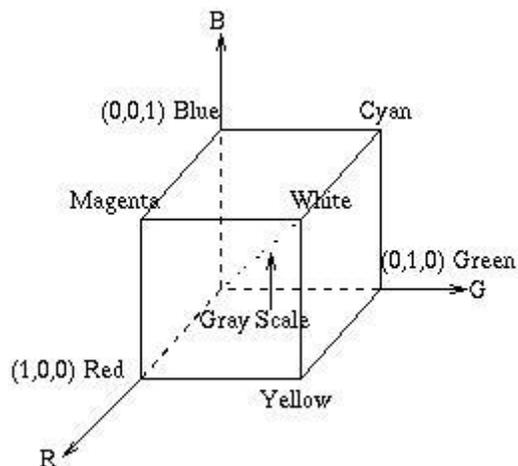


Fig. 3.2 Modelo Espacio Imagen RGB[18].

Sin embargo, una imagen de escala de grises (o-nivel de gris) es una imagen en el que sólo hay un tipo de base color gris y los objetos de este tipo de imagen se pueden distinguir por las diferencias en los valores del nivel de gris. La razón por la diferenciación de este tipo de imágenes a partir de cualquier otro tipo de imágenes en color es que hay menos información que debe facilitarse para cada píxel. De hecho, una de color gris es una en la que los componentes rojo, verde y azul, todos tienen igual intensidad en el espacio RGB (Fig.3.2). Por lo tanto, lo único que necesita es especificar un único valor de intensidad para cada píxel, en lugar de la recogida de tres valores de intensidad para especificar un píxel en una imagen a todo color.

Si una imagen se ha convertido de una imagen RGB a una imagen en escala de grises, es mucho más fácil y más rápido poder realizar cálculos de procesamiento de imágenes. Por la misma razón, en este estudio, se utilizó una cortina en blanco para separar el fondo mientras que la imagen era capturada. La Fig.3.3 ilustra la imagen se cambia de un RGB de una imagen en escala de grises.



(a)



(b)

Fig. 3.3 Imagen de RGB a escala de grises

(a) Imagen RGB. (b) Imagen Escala de Grises.

La Tabla 3.1 compara las características de estas dos imágenes. Siendo la misma imagen de entrada y salida durante el procesamiento, mientras que el tamaño y los bytes de la imagen se reducen tanto a 1/3 de la imagen original.

TABLA3.1 Características de Imagen RGB y Escala de Grises.

Imagen	Size	Bytes	Clase
RGB	480x640x3	8439234	Unit8 array
Escala de Grises	480x640	2830945	Unit8 array

Una vez que la imagen se ha convertido en escala de grises, el rango de valor de color para un píxel es [0, 255]. Al utilizar el comando "imtool" de Matlab en la ventana de visor de imágenes (Fig.3.4, a), se encuentra un píxel más preciso en la imagen y este se puede observar en la ventana de región de píxeles (Fig.3.4, b).

Notemos que en la Fig.3.4 (b), los valores de píxel de nivel de gris son alrededor 229. Esto indica que el valor del nivel de gris de los píxeles del fondo es de alrededor de 229. Además

podemos repetir la operación en un píxel diferente, como por ejemplo en las Fig.3.4 (c) y (d) muestran que el valor de píxel de nivel de gris de la rama es inferior a 90.

En este estudio, la imagen de histograma nos sirve para mostrar la distribución de intensidades de una imagen. Fig.3.5 ilustra la imagen de escala de grises después de usar el imhist función Matlab.

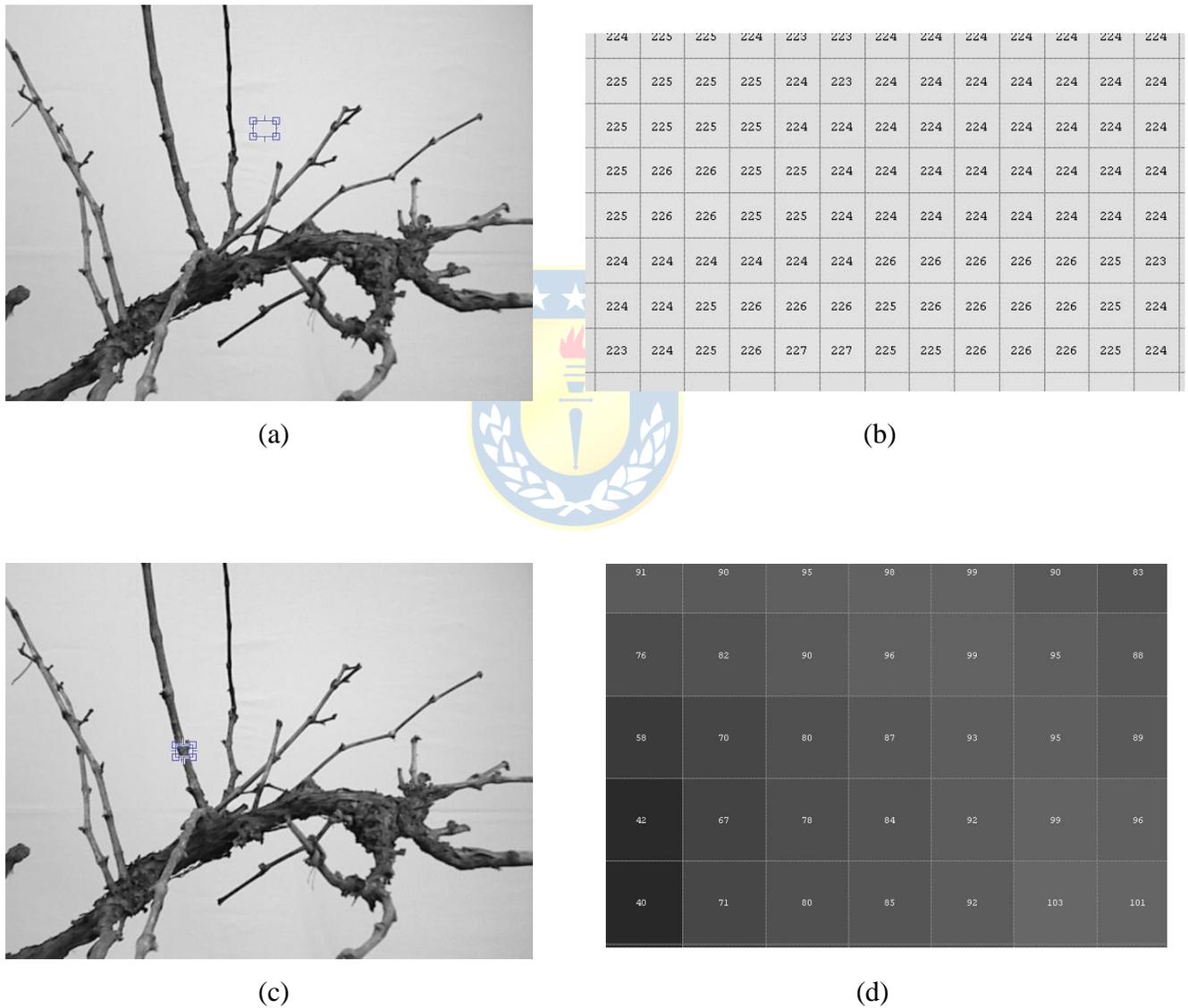


Fig. 3.4 Nivel de grises en la imagen

(a,c) Imagen Escala de Grises. **(b,d)** Valor de Pixel en la Región Marcada.

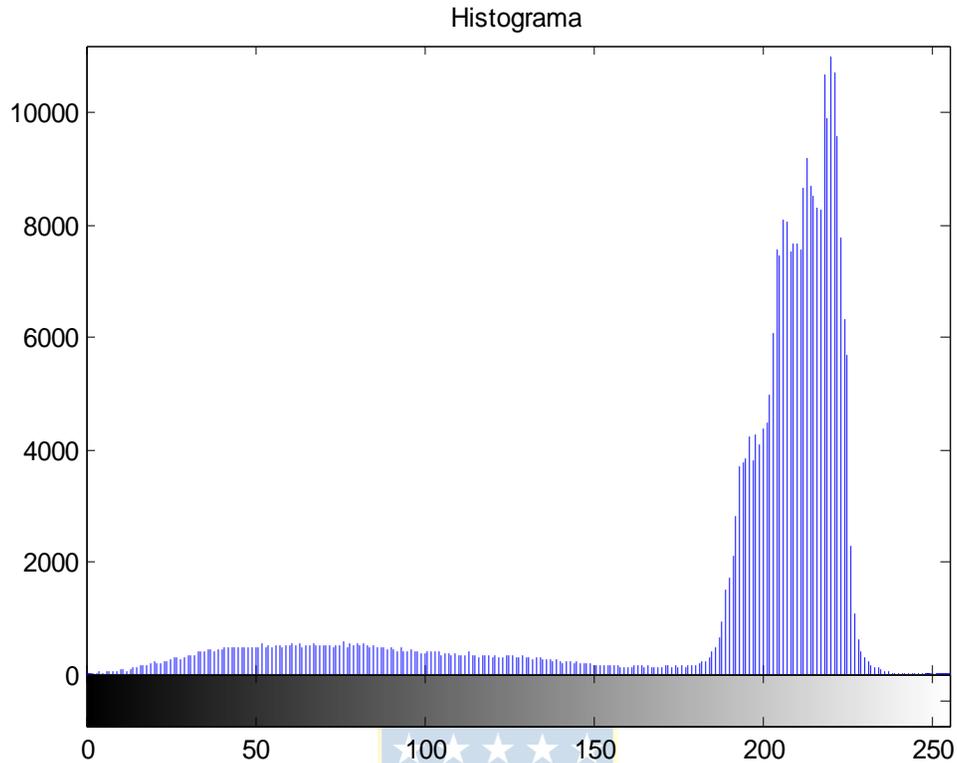


Fig. 3.5 Histograma de la Imagen Escala de Grises.

La Fig.3.5 da a conocer el histograma de la imagen de la figura 3.3 (b). De las Fig.3.3 y 3.4 es fácil notar que los píxeles en los cuales los valores de nivel de gris son menor a 160 ($p < 160$) son los que contienen casi todos los cordones y ramas.

Los píxeles que tienen valores superiores a 90 ($p > 90$ niveles de gris) son el fondo (Fig.3.3 y 3.4). Ha sido probado en todas las imágenes. Con el fin de separar los cordones y las ramas del fondo de la imagen, existe la opción de tener las ventajas que ofrece la imagen binaria que será presentada en la siguiente sección.

3.2.2 Imagen en Escala de Grises a Imagen Binaria.

La imagen binaria (blanco y negro) es un tipo especial de imagen con diferentes valores de intensidad. Los valores de nivel de gris de los píxeles en una imagen binaria sólo pueden ser 0 (negro) o 1 (blanco). Eso significa que hay sólo dos colores en la imagen, el color utilizado para los objetos (color de primer plano) y el color usado para las piezas del resto de la imagen (color de fondo). La imagen binaria, también es conocida como bi-nivel o de dos niveles, se utiliza a menudo en el análisis de imágenes. También se puede considerar como el resultado de algunas operaciones.

Aquí, en este estudio, puede verse como el resultado de la operación de umbralización. Umbralización de la imagen, así como el nombre lo dice, significa un proceso en el que se elige un valor de umbral y se establece como el punto de demarcación. Hay varios pasos en una operación de umbralización comunes que se enumeran a continuación:

- a) Se elige un valor umbral U en función de la situación en donde se captara la imagen.
- b) Se segmenta la imagen en los objetos (en primer plano) y el fondo, de acuerdo con el valor del umbral. Dos conjuntos de píxeles se crean en este proceso.

$$\begin{cases} P_1 = \{f(m, n): f(m, n) > U\} & (\text{Objetivo Principal}) \\ P_2 = \{f(m, n): f(m, n) \leq U\} & (\text{Fondo}) \end{cases}$$

Donde $f(m, n)$ es el valor de píxel del punto situado en la columna m y la fila n .

- c) Se asigna un nuevo valor a todos los píxeles del conjunto P_1 . De manera similar, se asigna un nuevo valor a todos los píxeles del conjunto P_2 .



Fig. 3.6 Telón

Para el algoritmo hay varias funciones tales como funciones iterativas que se pueden utilizar en el proceso de umbral para hacer la operación más precisa. La Fig.3.7 ilustra la operación de umbral a través de la comparación entre la imagen original y la imagen después de la operación de umbralización.



(a)



(b)

Fig. 3.7 Binarización de la imagen

(a) Imagen Original. (b) Imagen después de la operación de umbralización.

En este trabajo el valor de umbral se elige para que sea 90, basándose en el resultado de la Fig.3.5. En las imágenes de este estudio, los píxeles que tienen valores inferiores a 90 se restablecen a 0 (negro) en las imágenes binarias, mientras que los píxeles con valores superiores a los 90 se restablecen a ser 1 (blanco) en las imágenes binarias. Es mucho más fácil distinguir los cordones y las ramas del fondo en las imágenes binarias. Al igual que, es mucho más conveniente para poder encontrar los cordones en el siguiente paso. La figura 3.8 ilustra la imagen en blanco y negro que sólo contiene 0 y 1



Fig. 3.8 Imagen Binaria Parra.

3.3. Análisis de Imágenes.

Se han utilizado técnicas de análisis en muchos campos para analizar las imágenes procesadas y así extraer información desde las imágenes. En este informe, se requiere determinar las posiciones de los cordones, los bastones y los nodos. El ítem de análisis de imagen en este estudio se centra principalmente en estas tres partes. La figura 3.9 ilustra el diagrama de flujo de tres pasos principales en análisis de imagen.

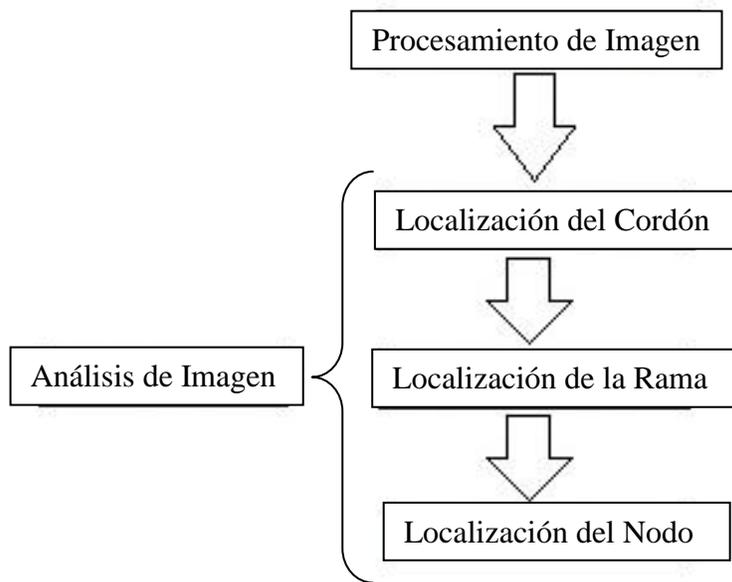


Fig. 3.9 *Diagrama de Flujo de Análisis de Imagen.*

3.3.1 Localización del Cordón.

Un objetivo importante de este trabajo es encontrar las posiciones de los nodos en las ramas. Esto es difícil debido al cambio de pendientes de las ramas, es decir, algunos de los nodos son considerablemente más suaves o difíciles de reconocer que otros. De acuerdo con el método de la poda segundo brote, el punto de poda depende de las posiciones de los nodos. Si el nodo no puede ser identificado por el algoritmo como resultado del cambio de la pendiente de la rama o el sistema encuentra más nodos que en realidad existían debido a los posibles ruidos de la imagen, el punto de corte no se podrá determinar con precisión. Esto conllevaría a la poda incorrecta, lo que afectará el crecimiento de la parra en el año siguiente.

La localización del cordón es el primer paso en la etapa de análisis de imagen. Esto se decide por el ciclo vegetativo de la parra, ya que en el proceso de crecimiento de una parra, las ramas y los nodos sólo aparecen después de la derivación del cordón. Y el cordón es la única parte permanente en una parra entre estas tres partes. Lo que es más, el cordón es la parte más baja de toda la imagen si el fondo no se toma en consideración. Una vez que se encuentra la posición del cordón, a la vez estamos identificando la línea de base en donde nace la rama, la cual es por encima del cordón. El área debajo de esta línea no se consideraría, con lo que se ahorrará tiempo para el cálculo.

3.3.2 Procedimiento.

De acuerdo con la Fig.3.8, ya se ha decidido que las ramas y los cordones serán el objetivo principal de la imagen de color negro (valor de píxel = 0) y todo el fondo será de color blanco (valor de píxel = 1). Dado que las imágenes originales son capturadas por el sensor Kinect con una resolución de 480x640, lo que significa que el $Y_{max}=480$ y el $X_{max}=640$. Esto de igual manera pueden entenderse que, en la dirección paralela al eje de las X, el cordón contiene más píxeles negros en comparación con las otras partes de la imagen de la parra.

Sobre la base de este descubrimiento, en la Fig.3.10, se muestra el esquema del ciclo que se usó para encontrar el cordón. Primero se comprueba el número de píxeles para el cual los valores son 0 en la dirección X para cada línea definida por el alto de la imagen, que se denota por un cierto valor Y, a contar desde la línea de fondo de la imagen. Se graba la posición Y cuando el número contador sea superior a $X_{max}/2=320$, y se restablecen todos los valores de píxeles de las línea a 1 (blanco) cuando el número de recuento sea inferior a $X_{max}/2=320$. El diagrama de flujo de este programa es el siguiente:

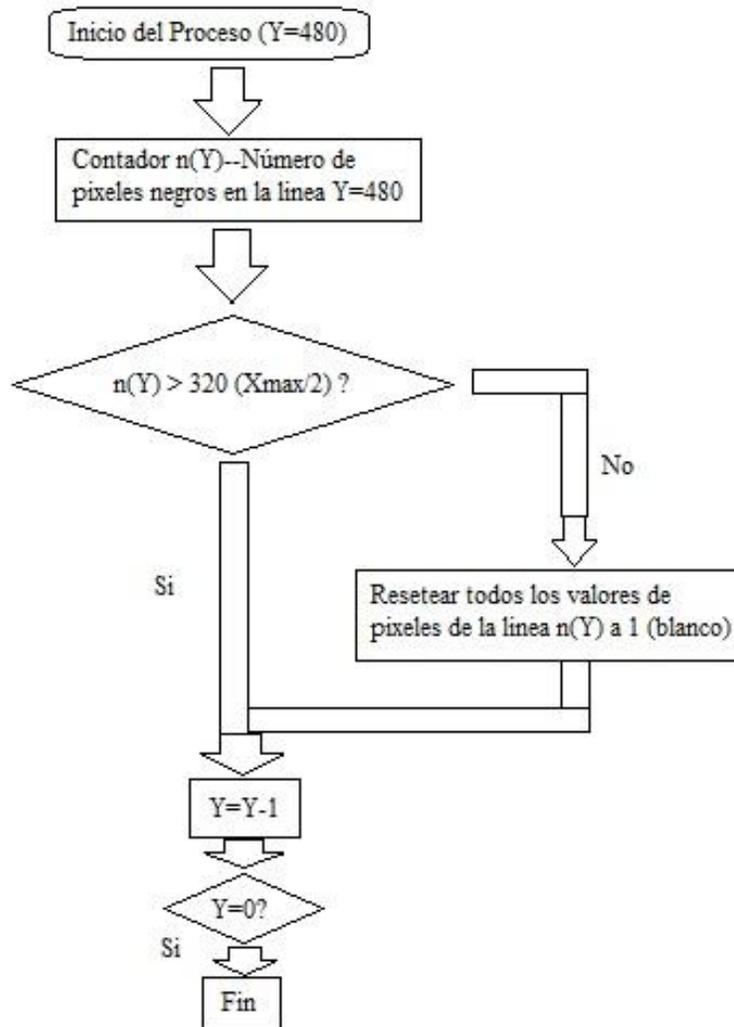


Fig. 3.10Diagrama de Flujo de Localización del Cordón.

La posición donde Y es la parte más baja, donde nace la rama desde el cordón, esto se determino haciendo el procedimiento antes explicado. La coordenada resultante Y es 367. Este resultado se puede ver en la Fig.3.11.

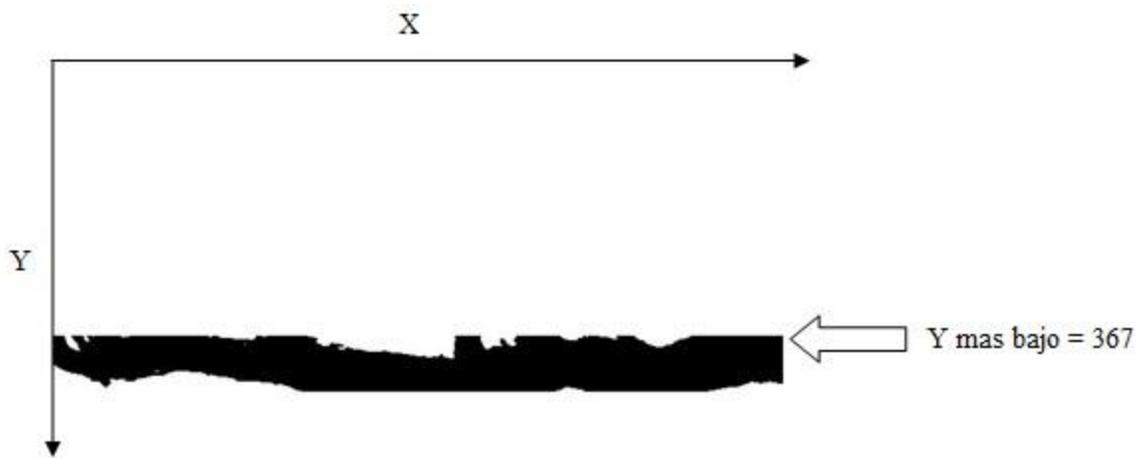


Fig. 3.11 *Cordón.*

3.3.3 Resultado.

Al final de la Sección 3.3.2, en la imagen se muestra que el cordón se puede ver claramente, además la posición más baja de Y también se conoció ($Y = 367$). La Fig.3.12 muestra otro ejemplo utilizando el mismo método pero con otra imagen. En este caso, la posición del cordón se encuentra en el valor 582 (en la dirección Y).



Fig. 3.12 Localización del *Cordón.*

3.3.4 Discusión.

Los cordones se han encontrado con éxito en las dos imágenes que se muestran arriba. Sin embargo, las condiciones de las imágenes varían mucho en diferentes situaciones. Aquí otro problema surge, puede haber dos partes de un cordón en la imagen después de el procesamiento de la imagen. La Fig.3.13 (c) ilustra una imagen que tiene un problema así.

Esto se debe a que hay demasiadas ramas cerca del cordón. En el punto 3.3.1, como se ha explicado, en la dirección X, cuando los valores de píxel negro son más de la mitad de X_{max} se muestran en las imágenes y son considerados como parte del cordón. Sin embargo, si hay demasiadas ramas cerca del cordón, tal como se muestra en la Fig.3.13 (b), es posible que el programa tome las líneas de píxeles que tienen ramas densas en píxeles negros en las direcciones de X como parte del cordón después de la ejecución del algoritmo. Esto se debe a que los valores de nivel de gris de las ramas y el cordón son 0 (negro) y una vez que la cantidad de píxeles negros es más alta que el 50% de X_{max} , será reconocido como parte del cordón por el algoritmo. En la misma línea de la dirección X, si el valor del umbral de los píxeles negros se cambia para que sea más de 70% (las líneas en donde la cantidad de píxeles negros son mayores que 70% de X_{max} serán considerados como parte del cordón), con esta solución, tal problema no volverá a ocurrir (Fig.3.13 (d)). Se ha probado en otras 10 imágenes, y se logró una tasa de éxito del 100%. Después de la localización precisa del cordón, el siguiente paso es localizar las ramas.



(a)



(b)

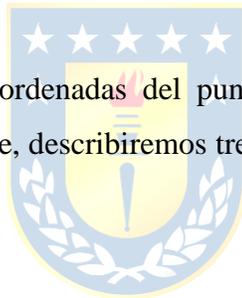


Fig. 3.13 Cordon

(a)Imagen. (b)Escala de Grises. (c)Dos Cordones. (d)Cordón.

3.3.5 Localización de las Ramas.

Con el fin de encontrar las coordenadas del punto de corte en la imagen, es necesario primero localizar las ramas. En esta parte, describiremos tres métodos diferentes y se analizan con el fin de localizar las ramas.



- **Método uno**

La forma y los valores de píxel de nivel de gris se utilizan para localizar las ramas en el Método uno. El procedimiento es el siguiente:

- Localización de una parte de la rama.
- Segmentación de los puntos de la rama.
- Encontrar toda la rama sobre la base de la parte de la rama que se ha localizado.

El cordón se encuentra y se considera como la línea de partida a partir del cual las ramas se encuentran. La Fig.3.14 muestra el resultado del cordón que se encontró en la última sección y la Fig.3.15 muestra el resultado de la localización de las partes de las ramas que crecen desde el cordón.



Fig. 3.14 Cordón.

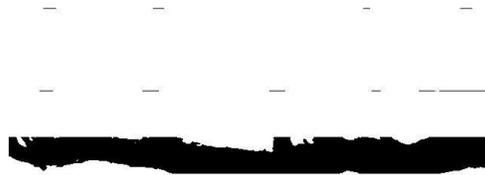
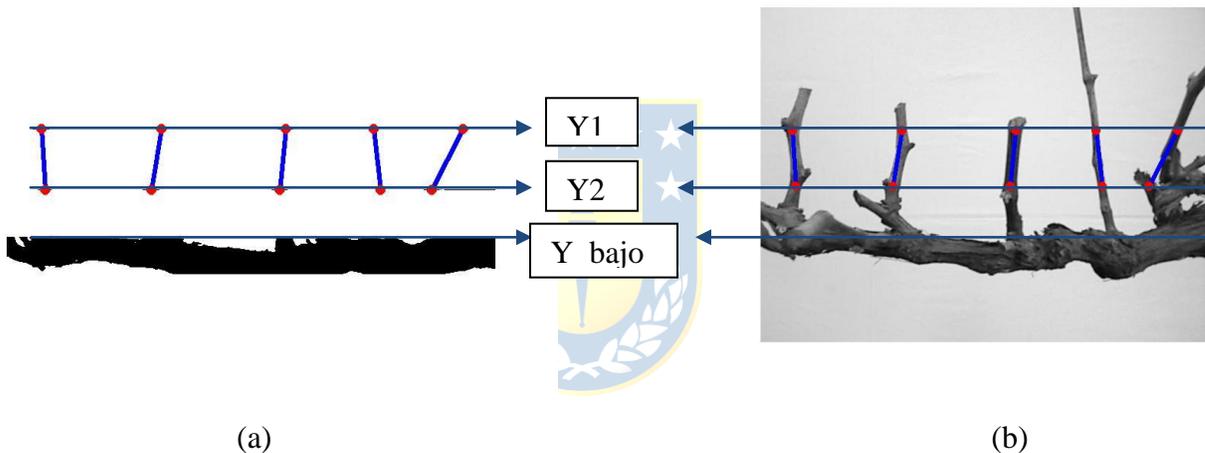


Fig. 3.15 Partes de las ramas.



(a)

(b)

Fig. 3.16 Localización de ramas método 1

(a) Localizar ramas. (b) Comprobación Localización de Ramas.

Después de encontrar el cordón, se conoce la posición más baja del cordón, denotado como Ybajo (Fig.3.16 (a)). Al marcar dos líneas al azar por encima del Y antes mencionado, es decir, trisecar la distancia desde $y = 0$ a Y más bajo. Habrá tres líneas: Y1, Y2 y el Ybajo (Fig.3.16, (a)). Así partes de las ramas se pueden mostrar entre las líneas $Y = Y1$ e $Y = Y2$ (Fig.3.16, (b)). Habrán algunos puntos en las líneas $Y = Y1$ e $Y = Y2$. Se tomo el primer punto en la línea Y1, y se intento encontrar el punto correspondido en la línea Y2.

Como se describe anteriormente, la imagen puede segmentarse y toda la rama puede ser determinada dando los pequeños puntos y sus líneas de conexión. Luego, se obtiene la posición de las ramas en todas las áreas posibles encima de Ybajo.

- **Método Dos.**

En este método, se requiere la intervención de alguna maquina, robot o persona para mover una cierta distancia la rama que se quiere localizar. Por esta razón es necesario tomar dos imágenes desde el sensor Kinect, la primera imagen se toma antes del movimiento de la rama y la otra se toma después de este movimiento. Esto permite aislar una única rama para la poda y así identificar el punto de corte. La Fig.3.17 ilustra el diagrama de flujo seguido para procesar las imágenes.

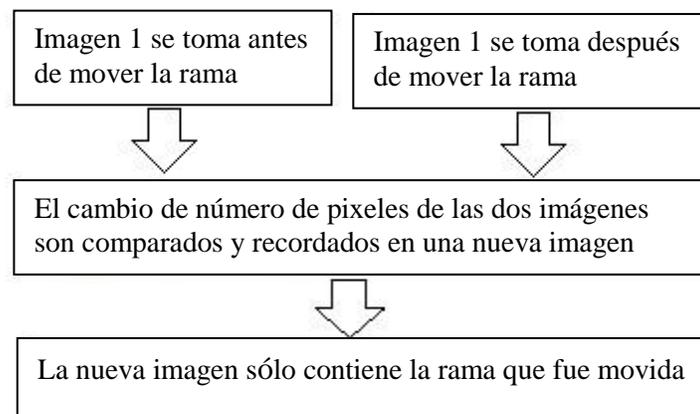


Fig. 3.17 *Diagrama de Flujo de Localización de la Rama.*

Después de esto, se muestra un ejemplo de la ejecución de estos pasos en una imagen de una parra en el laboratorio. En primer lugar, la imagen (1) fue cargada y se ha cambiado de RGB a blanco y negro (Fig.3.18, a). Luego, la imagen (2) se ha cargado y al igual que la imagen anterior, se cambió de formato RGB a blanco y negro (Fig.3.19). Por último, se compararon los valores de los píxeles de las dos imágenes. El algoritmo logra descifrar que algunos de los valores de los píxeles han cambiado debido a que la rama ha sido movida. Sobre la base de este proceso, una nueva imagen que sólo contenía los píxeles que han cambiado puede ser creada (Fig.3.20).

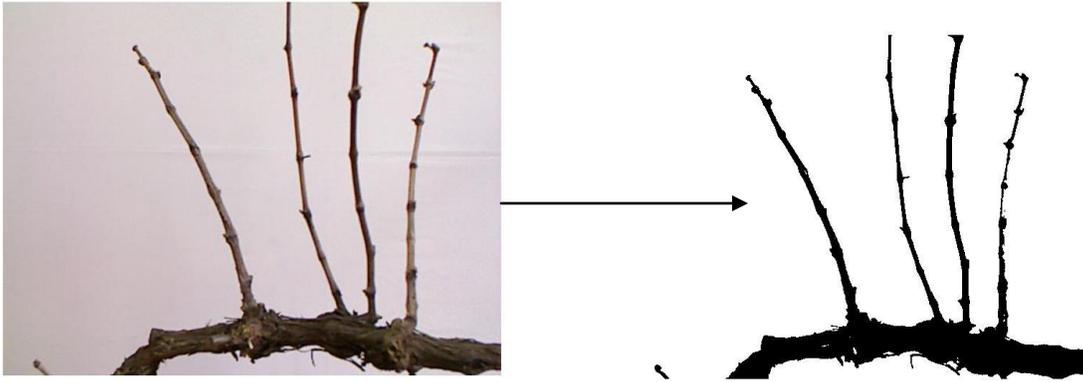


Fig. 3.18 Antes de Mover la Rama, desde RGB a Binaria.

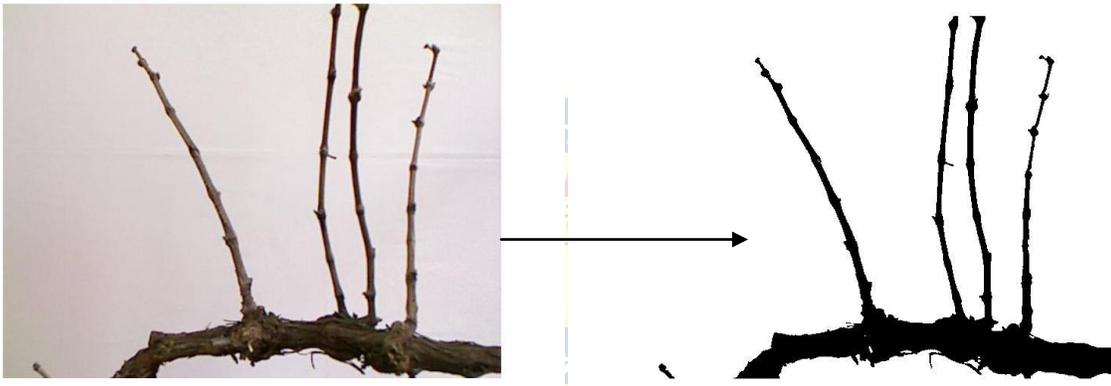


Fig. 3.19 Después de Mover la Rama, desde RGB a Binaria.

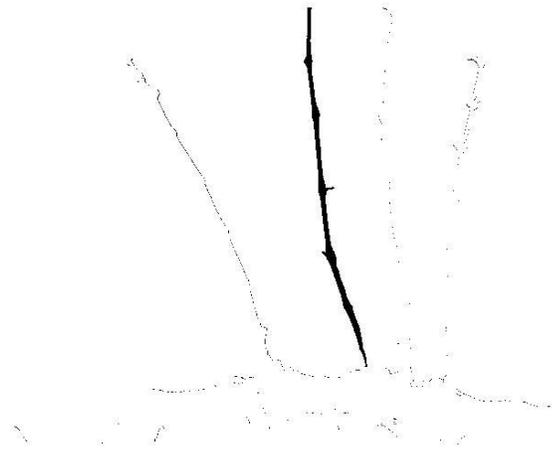
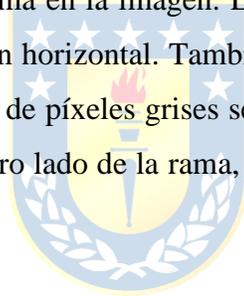


Fig. 3.20 Rama que se movió.

- **Método Tres(Cambio de Pendientes).**

La Figura 3.19 ilustra las ramas que tienen un valor de píxel de 0 (negro) de la imagen. Definimos la parte superior izquierda como el punto de inicio de toda la imagen. Esto con el fin de localizar todos las cañas de la imagen incluyendo las más cortas, el programa puede iniciar la búsqueda a partir de $1/3$ de la altura de la imagen. Un tercio de la altura de la imagen es una constante empírica que se eligió después de ensayo y error. Si el punto de partida es superior a $1/3$, algunas ramas pueden no considerarse, ya que se encontrarían bajo la línea de partida. Con esto, no hay necesidad de escanear desde una posición inferior a $1/3$. Además de tener en consideración que la eficiencia de cálculo del algoritmo es mayor, esto ya que no se escanea toda la imagen, sólo la sección de interés. El programa busca desde el punto de partida hasta que encuentra el primer píxel de que el valor cambia de 1 a 0 en la dirección horizontal. Este píxel se considera que es el punto límite lateral izquierdo de la primera rama en la imagen. La anchura de la rama se consideró como una pequeña línea en la misma dirección horizontal. También es necesario encontrar el otro lado de la rama. Como todas las ramas de nivel de píxeles grises se ha establecido que sean de color negro, es fácil para el programa encontrar el otro lado de la rama, donde los valores de los píxeles cambian de 0 a 1 (0 es negro, 1 es blanco).



Después de que los puntos de los límites de la rama encontrada son localizados, se calcula el punto medio de esta en la misma dirección horizontal, a continuación, se puede identificar la posición (Y_{medio} , X_{medio}). El algoritmo se mueve verticalmente hacia abajo a la posición siguiente ($Y_{medio}+1$, X_{medio}) y busca de nuevo tratando de encontrar los límites de la caña en esta nueva línea. Se guarda el punto medio de la caña y se mueve verticalmente hacia abajo de nuevo para encontrar el siguiente punto. Finalmente, toda la forma de la rama se puede determinar, y todos los puntos medios de esta son grabados.

Como no es probable que la unión de los puntos medios de la rama sea una línea recta, es importante evaluarlos cambios de pendientes de la rama después de haber explorado una corta distancia de esta, la figura 3.21 ilustra el cambio de pendiente de la rama. Después de haber explorado una corta distancia de la línea media de la rama, se evalúan los cambios de dirección de

esta seguidilla de puntos que conforman los puntos medios para así tener toda la información de la rama sobre los quiebres, anchuras cambios de pendiente, entre otros a disposición.

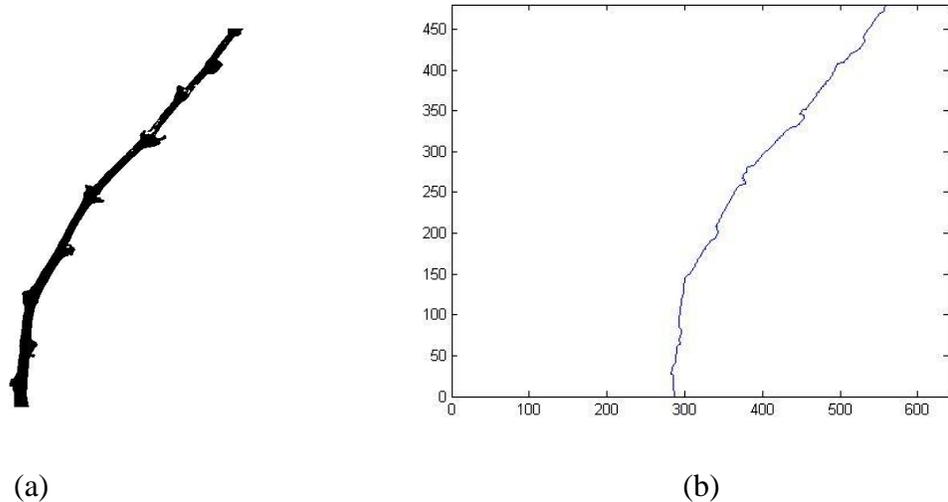


Fig. 3.21 Escaneo de la Rama

(a) Parte de una Rama. **(b)** Puntos Medios de la Rama

3.3.6 Resultados y Discusión.

Tres métodos se han analizado en este capítulo. Si comparamos estos tres métodos, el último es la mejor opción para ubicar la rama. Esto se debe a que para el método de uno, primero se segmenta la imagen y se intenta recolectar los puntos y conectarlos a una línea nueva. Para esto se necesita determinar si la línea es una rama real o no. Sin embargo, este método tiene que depender de la posición del cordón.

Si se encuentran dos o más cordones, este método puede fallar. Además, durante la conexión de las partes de las ramas, puede existir más de un número de opciones en la línea $Y = Y2$ por un punto en $Y = Y1$ a conectar, lo que es difícil de determinar, ya que depende de la situación. Por último, para la determinación de la rama se necesita volver a escanear la imagen binaria original y así averiguar si existen píxeles negros a lo largo de la línea conectada entre los puntos $Y1$ e $Y2$. Sin embargo, los píxeles de otras ramas pueden confundirse como la que se espera obtener. Por lo tanto, el programa hará una determinación incorrecta.

De acuerdo con el método dos, se requiere la intervención de un agente externo para mover la rama. No obstante, en el diseño de este método, se hace muy fácil localizar las ramas, pero muchas fotos tienen que ser cargadas en el computador para procesarlas, lo que provoca una ineficiencia en el programa. Además, si se pasan a llevar otras ramas el objetivo no se cumplirá. En comparación con el método de uno y tres, el segundo método tiene un paso más que tiene que llevar a cabo.

En el método tres, la ubicación de las ramas es más precisa cuando se mira a través de toda la imagen y desde el punto de inicio. Además, todo el programa está funcionando con sólo una imagen, no hay necesidad de comprobar la imagen original o procesar la imagen en varias ocasiones que permitirá ahorrar mucho tiempo para la unidad aritmética del sistema. También, se obtiene mayor información de los cambios de pendiente de las ramas que es muy útil y conveniente para nuestro siguiente objetivo, que es la localización de los nodos y que se analizará en la siguiente sección.

3.3.7 Localización de los nodos.

Una vez que las ramas han sido localizadas, el programa se centrará en la localización de los nodos. Según el método de poda espolón 2do brote, el punto de poda es en la región media entre el segundo y tercer nodo a contar de la parte inferior de la rama. Según el estudio de todas las imágenes capturadas, es obvio que el nodo es mucho más amplio en la misma dirección horizontal que cualquier otra parte de la rama. Esto significa que el método utilizado para encontrar peaks de acuerdo con el cambio de la pendiente podría ser utilizado para localizar los nodos, donde los cambios de dirección en estas partes son más notorios que cualquier otra parte de la rama. Varios pasos se deben hacer en el proceso de localización:

- a) Inicio del programa desde el lado cerca del cordón.
- b) Comprobación de los cambios de dirección en el vector de puntos medios de la rama.
- c) Grabación de los cambios máximos de pendiente y las coordenadas correspondientes.
- d) Coincidir las coordenadas con los nodos correspondientes a fin que la información de la posición de los nodos pueda ser determinada.

Todos los pasos anteriormente definidos se lograron mediante la ejecución del programa, los cuales se utilizaron para la búsqueda de los nodos, se puede ver todo el procedimiento a través de la siguiente imagen:

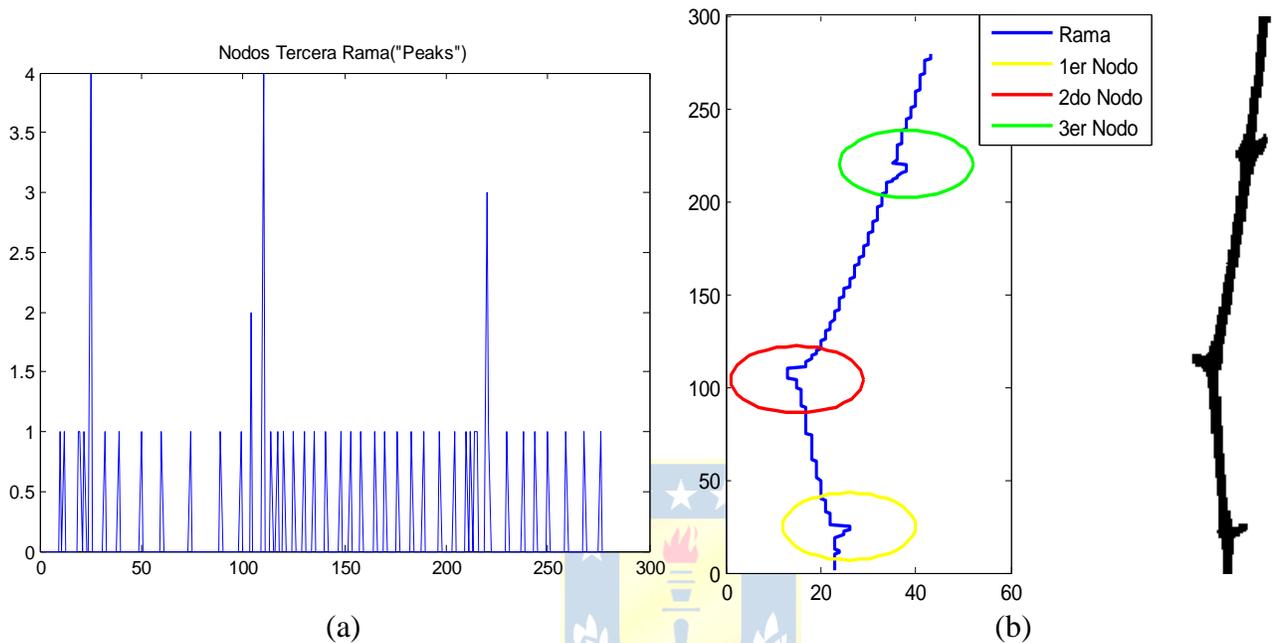


Fig. 3.22 Localización de nodos.
(a) Peaks ; (b) Identificación de nodos.

La localización fue satisfactoria, el método para ubicar los nodos mediante la evaluación de todos los puntos medios que habían sido guardados en un vector, permitió al programa definir claramente la posición de los peaks, esto lo notamos en la Fig.3.22,(a) lo cual además se ve reflejado en la Fig.3.22,(b) donde mediante unas elipses de colores se encierran los nodos localizados.

3.3.8 Discusión y Resultados.

Los nodos se han localizado de manera exitosa en la imagen de la parra analizada. No obstante, existen situaciones donde no es posible identificarlo debido a que el nodo no cumple con las características de los demás nodos encontrados o básicamente no marca una diferencia notoria con respecto al resto de la rama y por esta razón el algoritmo no lo considera. El problema dado a conocer previamente es de suma importancia para la búsqueda del punto óptimo de poda, ya que esta requiere fundamentalmente la localización exacta del primer, segundo y tercer nodo de cada rama. Las figuras siguientes ilustra un caso que posee un problema de esta índole.

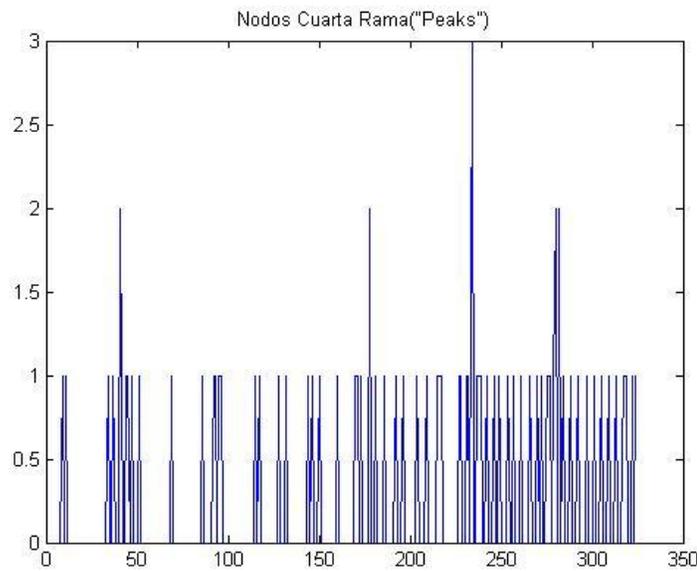


Fig. 3.23peaks de posición de nodos

En la Fig.3.23 vemos que el algoritmo logra identificar 4 nodos a lo largo de la rama, pero fijémonos bien en la Figura 3.24. que hay un nodo que no es considerado (Encerrado en un círculo rojo). La razón principal por la cual el escaneo de la rama mediante el comando "diff" omitió el nodo es porque como se dijo anteriormente, no se logra hacer una diferencia clara entre nodo y rama, debido a que son muy similares en su ancho.

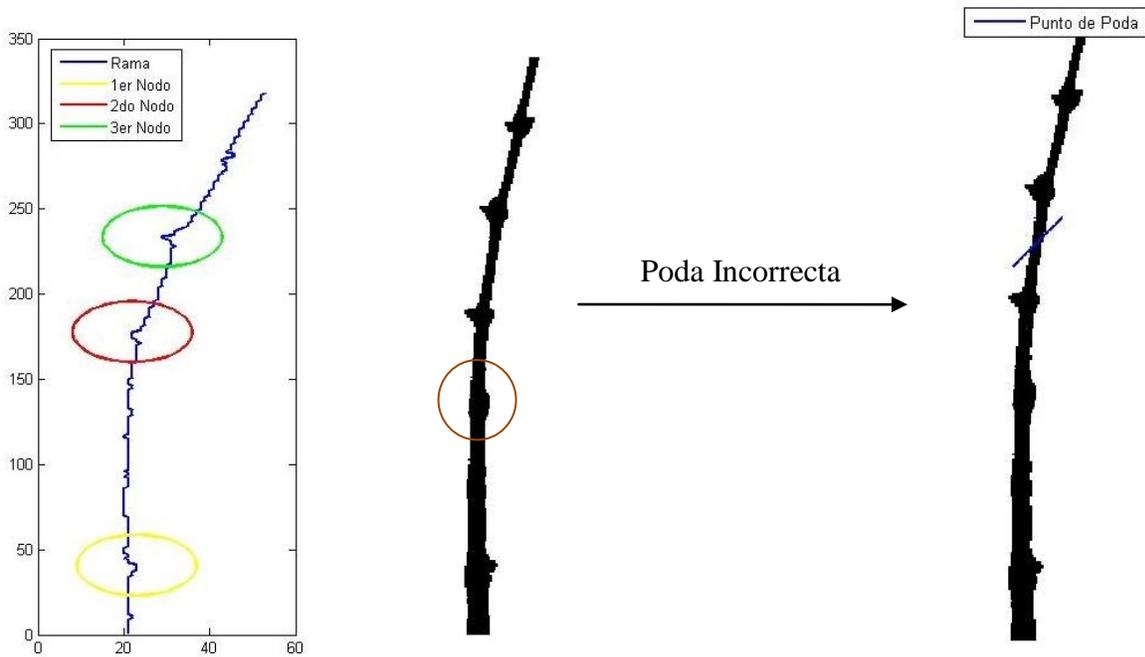


Fig. 3.24Falla al reconocer nodo

Por otro lado el diagrama de flujo del procesamiento de imágenes ha sido descrito a fondo al comienzo de este capítulo. La operación de umbralización debe llevarse a cabo con el fin de convertir una imagen de escala de grises en una imagen binaria. Esto hace que la adquisición de la información de la imagen sea mucho más rápida. El procesamiento de imágenes y análisis también se han introducido en este capítulo. El programa comprueba los píxeles que tienen un valor de 0 (negro) a lo largo de píxel de cada línea horizontal paralelo al eje X en la imagen y restablece el valor de todos los píxeles en esta línea a 0 para localizar el cordón. Tres métodos diferentes sobre cómo localizar las ramas han sido analizados. En la sección 3.3.5, el método tres es finalmente elegido para la localización de la rama.

3.4. Reconstrucción 3D.

3.4.1 Introducción.

En esta sección, el método empleado en la obtención de la posición de la poda para tres dimensiones se experimenta en un conjunto de datos reales mundo. Se utilizó la imagen de profundidad del sensor Kinect para capturar las imágenes de las parras en el laboratorio. Acotaremos que el dispositivo que adquiere las imágenes esta fijo sobre un trípode y mantiene una distancia entre la cámara y la vid de 700 mm horizontalmente, esto se puede apreciar en la Fig.3.25.



Fig. 3.25 Rama de parra

3.4.2 Imagen Rango 3D de la Parra.

Usualmente, la imagen obtenida por una cámara convencional representa la distribución de intensidades de una escena observada. En cambio, una imagen de rango de la misma escena representa las distancias de la cámara a los puntos de la escena, que en nuestro caso corresponde a la parra con el fondo conformado por un telón. Una imagen de rango puede ser considerada como un mapa de profundidades, la cual el algoritmo lo representa mediante la variable $depth(x,y)$, donde "depth" es la medida de profundidad o rango de la escena y (x,y) son las coordenadas en el plano de formación de la imagen, ver esto en la Fig. 3.26.

La principal ventaja de las imágenes de rango es que en ellas se tiene explícitamente información de las características geométricas de las superficies medidas. Estetipo de imagen es especialmente útil para los procesos de inspección de superficies ya que los cambios o defectos

concaracterísticas 3D, como cavidades o salientes, son representados con alto contraste y pueden ser distinguidos notoriamente.

En la Fig.3.25 vemos una parte de la parra que corresponde a una rama. Utilizaremos las imágenes de disparidad captadas por la Kinect a esta rama con el fin de lograr la reconstrucción 3D.

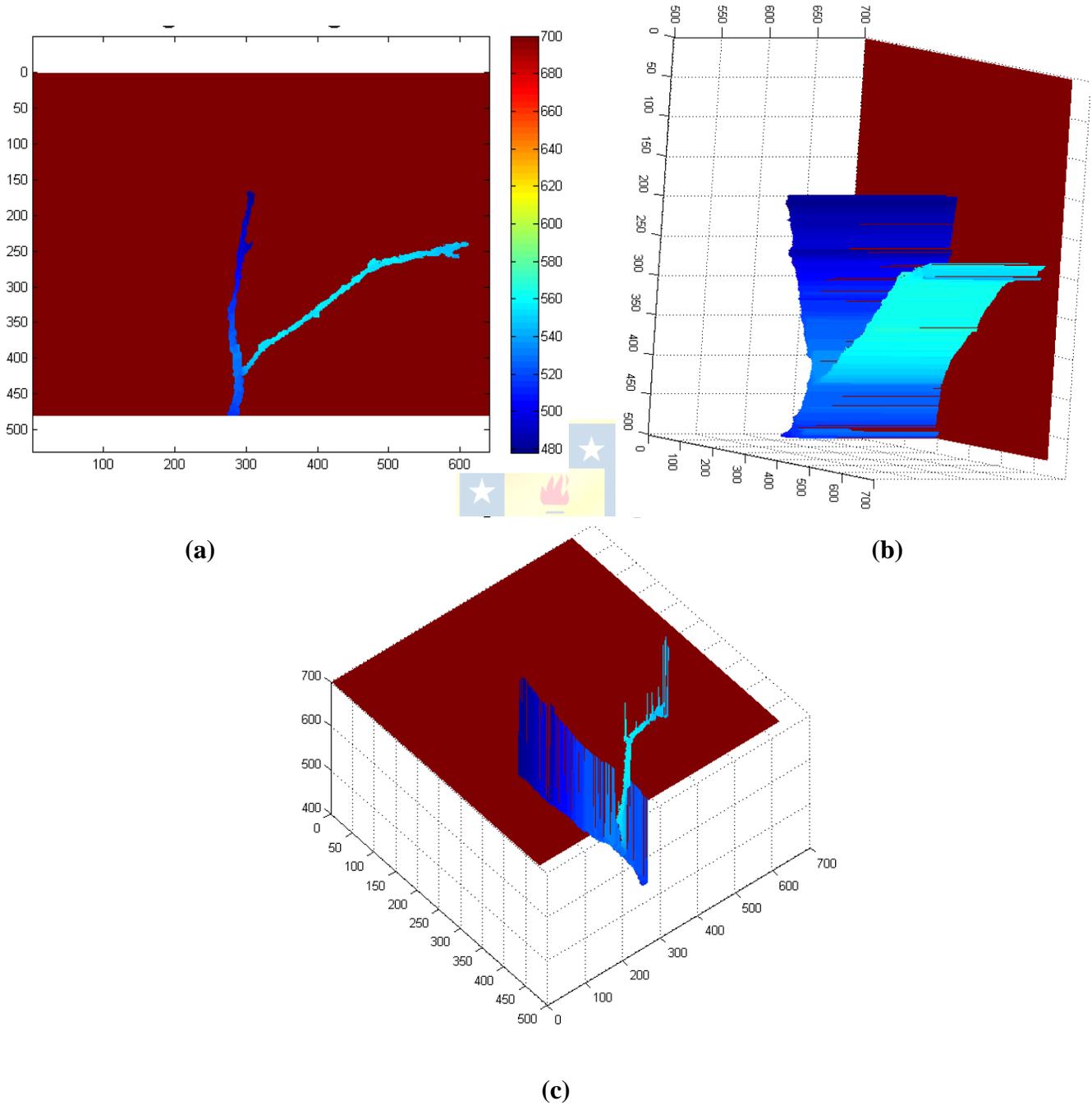


Fig. 3.26 Reconstrucción 3D rama-fondo.

(a) Imagen de rango 3D; (a) Mapa tridimensional; (c) Mapa tridimensional 2da vista

Hemos visto la imagen de rango acompañada de imágenes correspondientes a mapas tridimensionales, todas ellas muestran la rama y el fondo de la escena, pero a nosotros solamente nos interesa la rama, es por esta razón que eliminaremos el fondo guardando en un vector solamente los datos que se relacionan con la distancia existente entre la Kinect y alguna distancia posterior de la rama o una distancia anterior al fondo. Con esto, no consideraremos los datos de profundidad equivalentes al telón y de esta forma graficamos solamente los datos que fueron guardados en el vector, las gráficas que representa esto se muestran en la Fig.3.27.

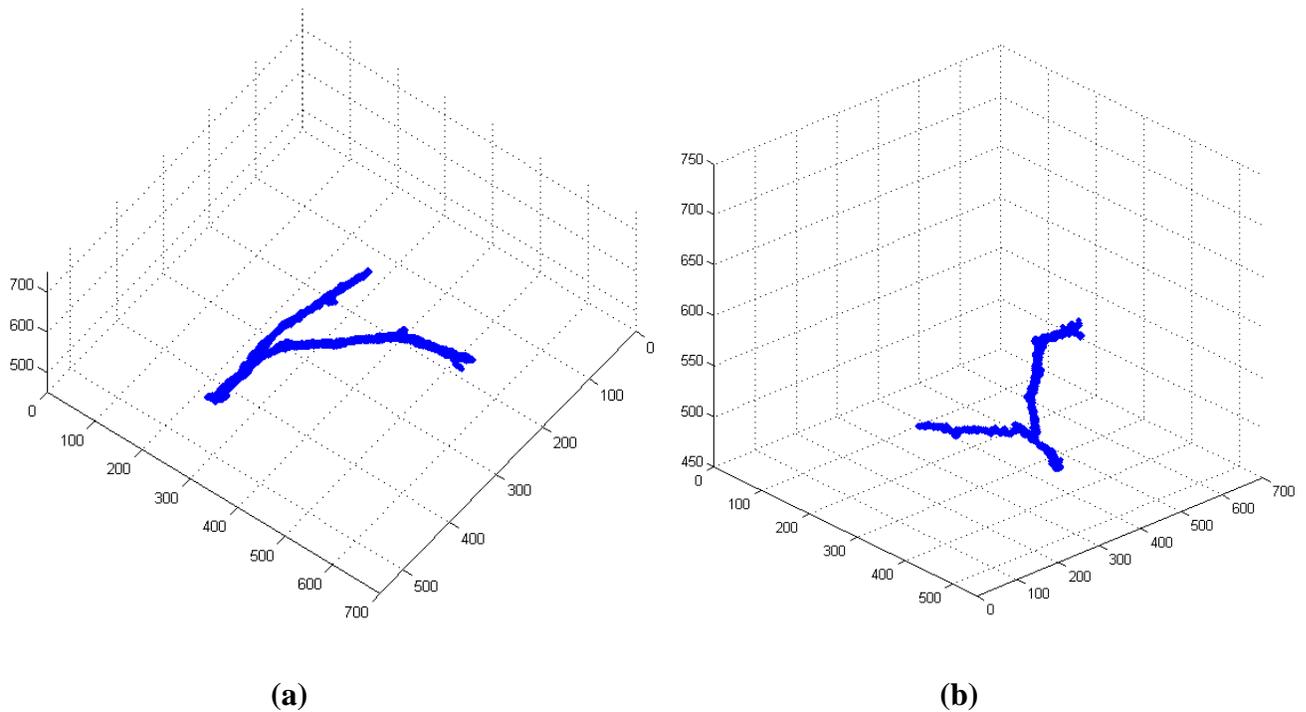


Fig. 3.27 Reconstrucción 3D rama
(a) Mapa tridimensional 1era vista; (b) Mapa tridimensional 2da vista

Capítulo 4. Pruebas y Resultados

4.1. Introducción

En el actual proyecto, hemos estudiado y realizado una serie de procesos que nos permiten llegar a cumplir nuestro objetivo principal. Ahora, para complementar de mejor manera lo precedido, veremos paso a paso los resultados obtenidos, los cuales se presentarán en este capítulo incluyendo además la explicación y el funcionamiento de la interfaz gráfica del usuario que se desarrollo. Esta interfaz tiene por objetivo realizar todas las etapas previamente analizadas y ser el puente de comunicación entre el algoritmo y el usuario, en una plataforma que sea fácil de entender y fácil de accionar.

Hay que mencionar que todo lo referente a la calibración del dispositivo Kinect se trabajo con el toolbox para Matlab desarrollado por Daniel Herrera [16] junto a la interfaz gráfica desarrollada en la Memoria de Título de C. Hernández [7], en donde se obtiene la data en unidades de ingeniería. Esta interfaz mencionada está ligada con la interfaz de este proyecto, como se verá en este capítulo, mediante el botón "*Calibrar*" podremos acceder a las ventanas de calibración antes mencionadas y así complementar nuestra Interfaz de mejor forma .

4.1.1 Funcionamiento del HMI.

Paso 1:

En Matlab R2012a, iniciamos la interfaz gráfica de usuario para comenzar con la etapa de adquisición de imágenes ejecutando la instrucción *HMI_adquisición* en la ventana de comandos de Matlab. Con esto se desplegará la ventana que vemos en la Fig.4.1.

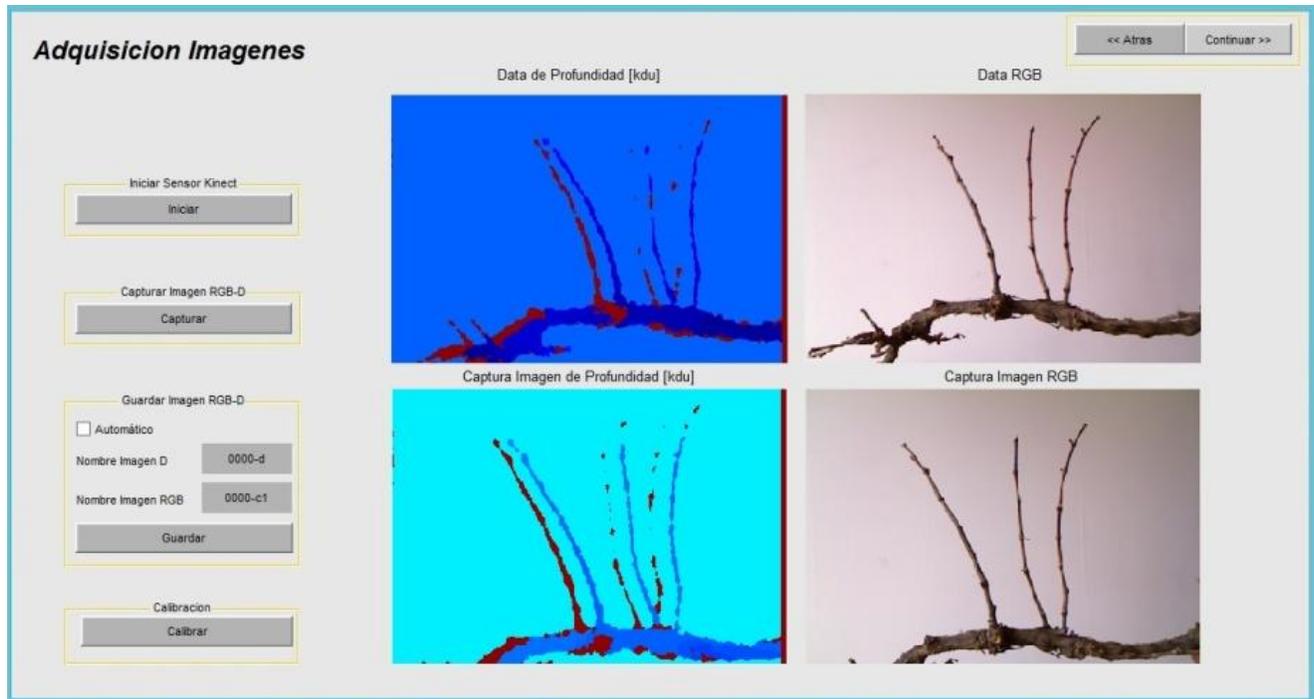


Fig. 4.3 Hmi adquisición imágenes

Paso 2:

Antes de realizar clic en cualquier botón, tenemos que conectar el sensor Kinect al ordenador mediante USB y además energizar este dispositivo mediante la red eléctrica. Luego presionar el botón "Iniciar" para realizar la comunicación con el sensor y comenzar a visualizar las imágenes RGB-D en la interfaz, en el lado izquierdo se muestra la imagen de profundidad y en el lado derecho la imagen de color. Una vez enfocada la parra proceder a hacer clic en el botón capturar, en donde las imágenes quedarán congeladas, para ver si la captura fue la correcta.

Para guardar las imágenes, la interfaz nos brinda dos opciones. Una de ellas es la de guardar cada imagen con el nombre que nosotros queramos, la otra se activa seleccionando el casillero "automático" y se guardarán las imágenes sin tener que ingresar algún nombre para cada una de estas.

Paso 3:

La interfaz nos da la opción de llevar a cabo la calibración del dispositivo haciendo clic en el botón "Calibrar", esto nos llevará a las ventanas de calibración que se muestran en las Fig.4.2.

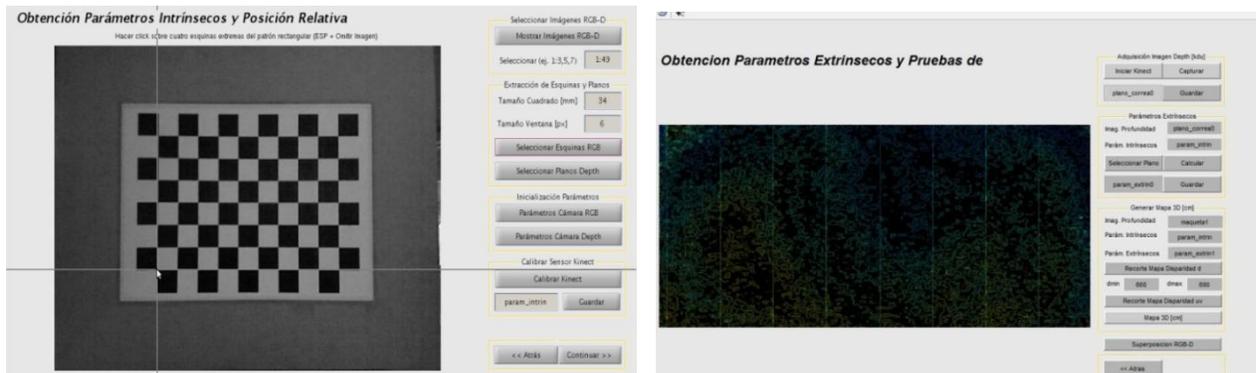


Fig. 4.2 Interfaz Calibración Microsoft Kinect [7]

Como se mencionó en la introducción del presente capítulo, todo lo referente a la calibración del dispositivo Kinect, como esta interfaz relacionada con la obtención de parámetros intrínsecos y extrínsecos que fueron desarrolladas en la Memoria de Título de C. Hernández [7], en donde podremos encontrar paso a paso las instrucciones del procedimiento de calibración y la utilización de la interfaz gráfica de usuario para estas dos ventanas.

Paso 4:

En la ventana de adquisición de imágenes hacer clic en el botón "*Continuar*" para ir a la siguiente ventana llamada identificación puntos de poda que se ilustra en la Fig.4.3. En esta, se podrá acceder a todas las funciones que ofrece el algoritmo con respecto a la búsqueda del punto a podar en las imágenes 2D, éstas se irán detallando a continuación.

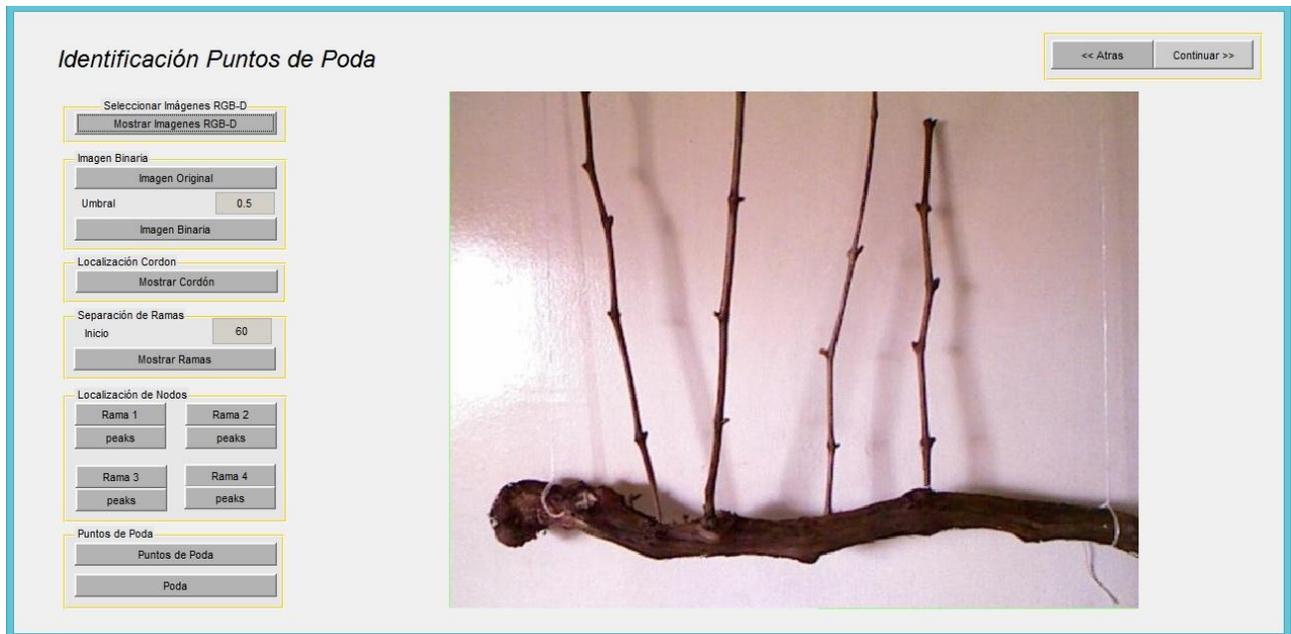


Fig. 4.3HMI identificación puntos de poda

En el paso 3 se guardaron las imágenes capturadas desde la Kinect, ahora con el botón "*Mostrar Imágenes RGB-D*" podemos buscar aquellas en el directorio en el cual fueron archivadas. Las funciones que realizan el resto de los botones en esta ventana se irán detallando uno por uno, los cuales irán respaldados por las imágenes que muestran el resultado de dichas funciones y las cuales también aparecen en la interfaz mostrada en la Fig.4.3.

Paso 5:

Una de las características que tiene el algoritmo, es que se trabaja con la binarización de la imagen, esto para poder realizar los procedimientos de cálculo de manera más sencilla. En este paso hay que ingresar el valor requerido del parámetro "*Umbral*" para tener una imagen binaria correcta, considerando que este valor debe estar en el rango entre 0 y 1. En la Fig.4.4. se muestra el resultado.

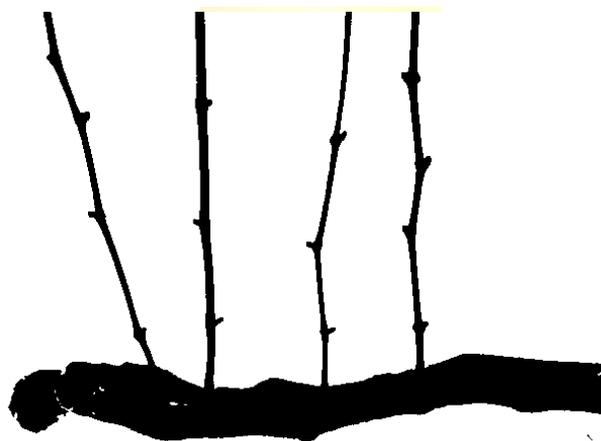


Fig. 4.4Resultado imagen binaria

Paso 6:

El siguiente botón "*Mostrar Cordón*" está relacionado con la ubicación del cordón de la parra. Este paso se realiza con el fin de localizar la línea de la imagen donde empiezan o nacen las ramas desde el cordón.

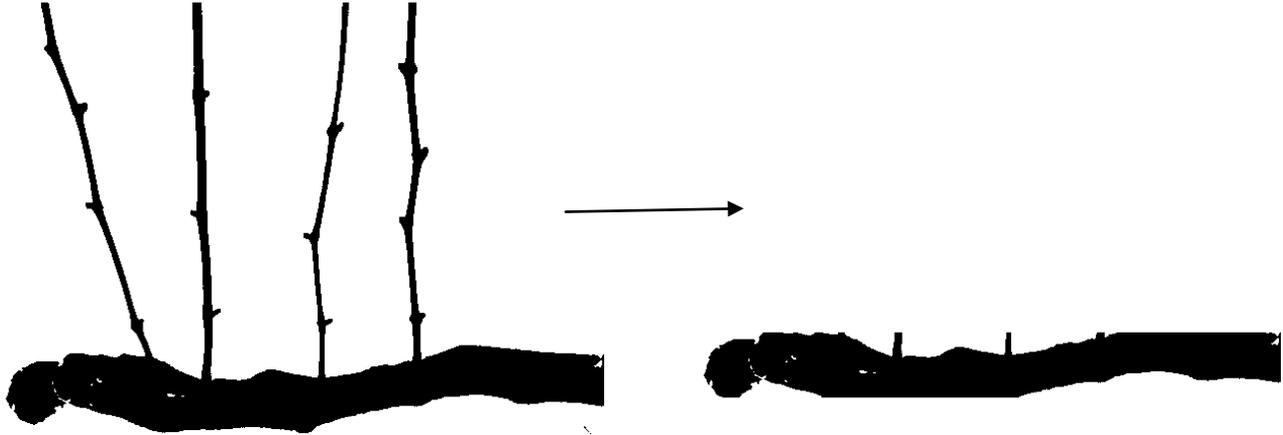


Fig. 4.5 Localización del Cordón.

Paso 7:

Para comprobar que las ramas han sido localizadas con éxito, se pone a disposición el botón "*Mostrar Ramas*". Este requiere ingresar el valor del parámetro "*Inicio*", este parámetro está relacionado con la posición de la coordenada Y de la imagen, el programa inicia un escaneo en esta línea, esto con el objetivo de saber cuántas ramas hay. El motivo por el que se agrega este parámetro es porque en algunas ocasiones existen parras que contienen ramas más cortas que otras y para no pasar por alto alguna rama es la razón por el que se impone este requerimiento.

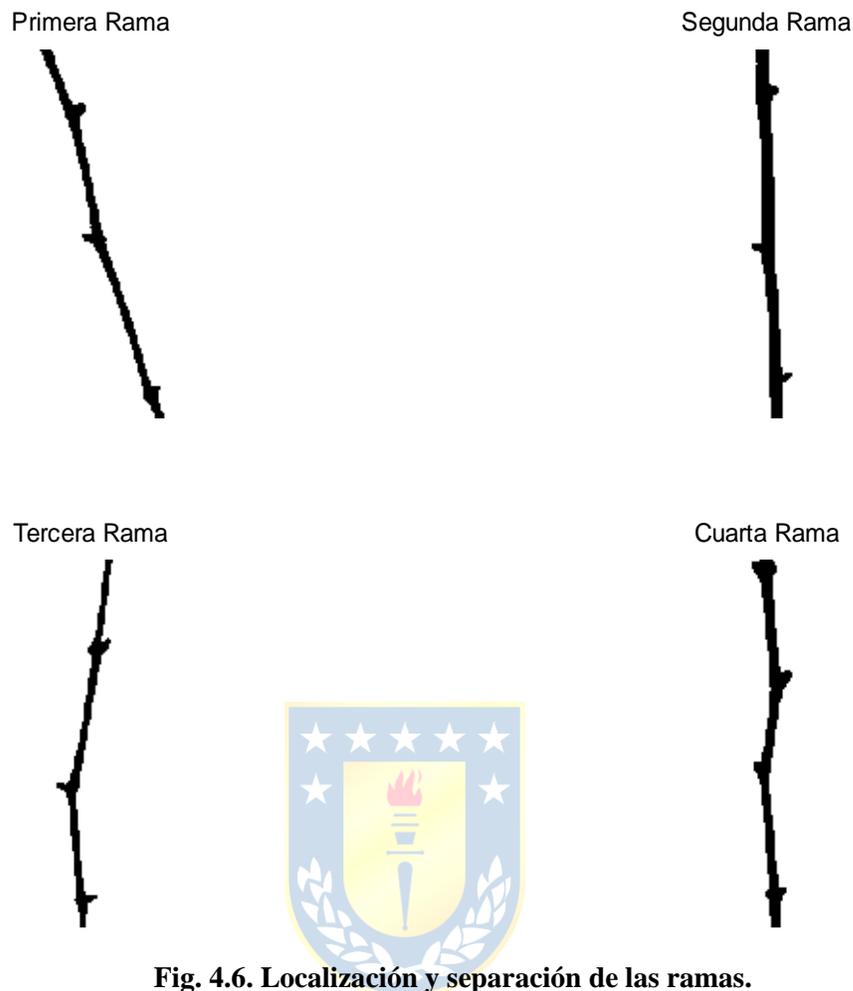


Fig. 4.6. Localización y separación de las ramas.

Paso 8:

La sección denominada "*Localizar Nodos*" nos ofrece la opción analizar cada rama por separado, donde en cada una de ellas se le indicará mediante una elipse la ubicación de sus nodos. El algoritmo identificará los 3 primeros nodos desde el nacimiento de la rama en el cordón, solamente 3 nodos se encontrarán, ya que el método de poda 2do brote considera el punto de corte de la rama entre el segundo y tercer nodo. Además tendremos a disposición a través del botón "*peaks*" las gráficas donde se notan claramente los peaks que corresponden a los cambios de dirección al momento de recorrer con un escaneo los puntos medios de la rama, de aquí se obtienen las coordenadas que se usan para la localización de los nodos.

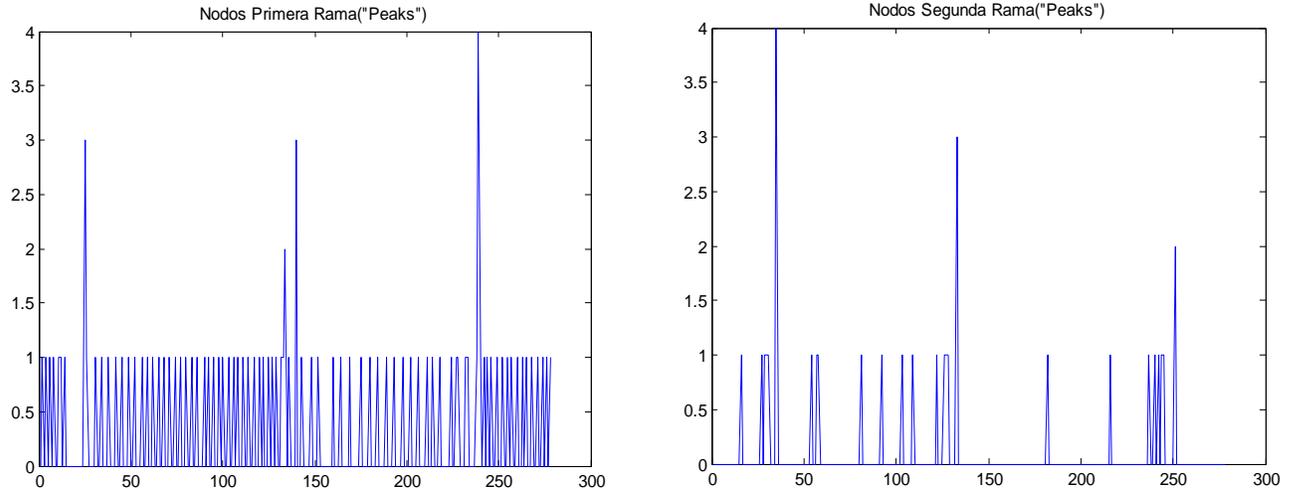


Fig. 4.7. Localización de 1er y 2do nodo mediante Peaks.

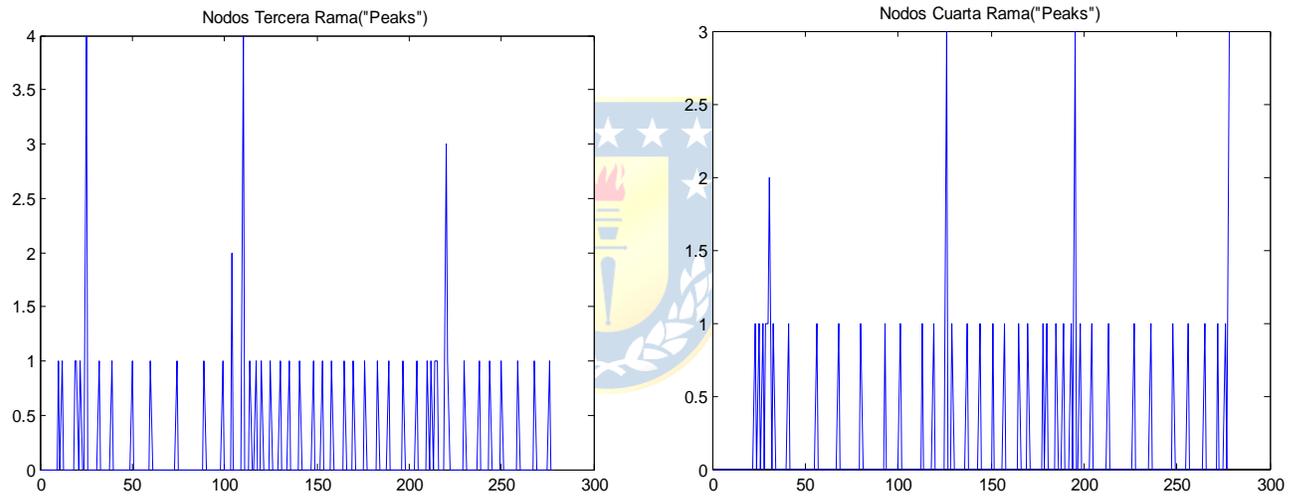


Fig. 4.8. Localización de 3er y 4to nodo mediante Peaks.

Hay que acotar que estas imágenes de las Fig.4.9 y Fig.4.10 que han sido agrupadas por pares el conjunto de cuatro ramas, en la interfaz se muestran cada una por separado según el botón de cada rama que se quiera escoger.

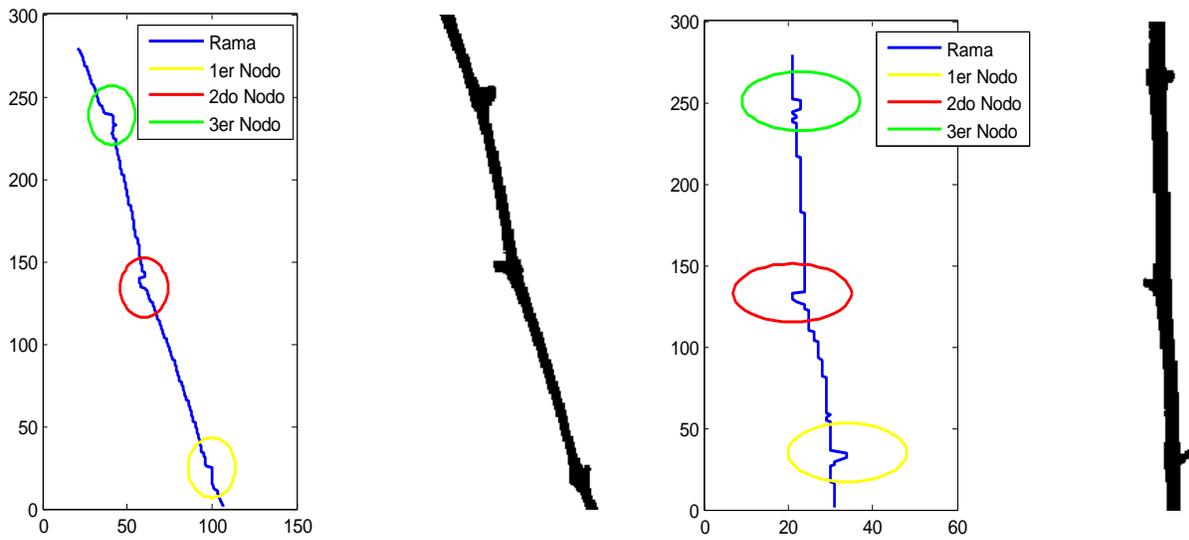


Fig. 4.9. Nodos Identificados 1era y 2da Rama.

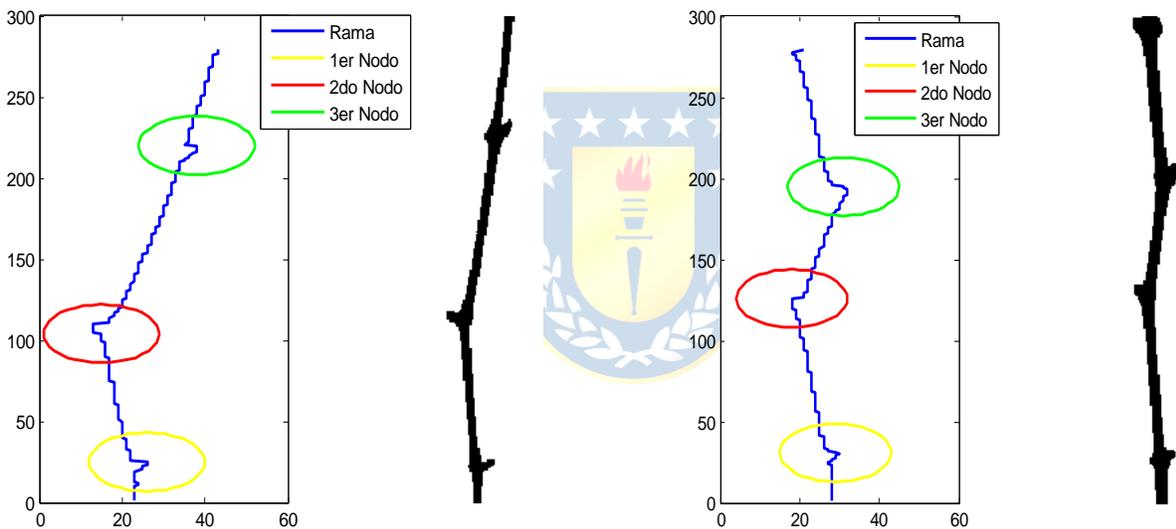


Fig. 4.10. Nodos Identificados 3era y 4ta Rama.

Hasta aquí, estamos en condiciones para definir el principal objetivo de este proyecto, la identificación del punto de poda de cada rama se pudo localizar satisfactoriamente. A continuación en la Fig.4.11 damos a conocer este punto y además el corte de la rama o la poda de esta.

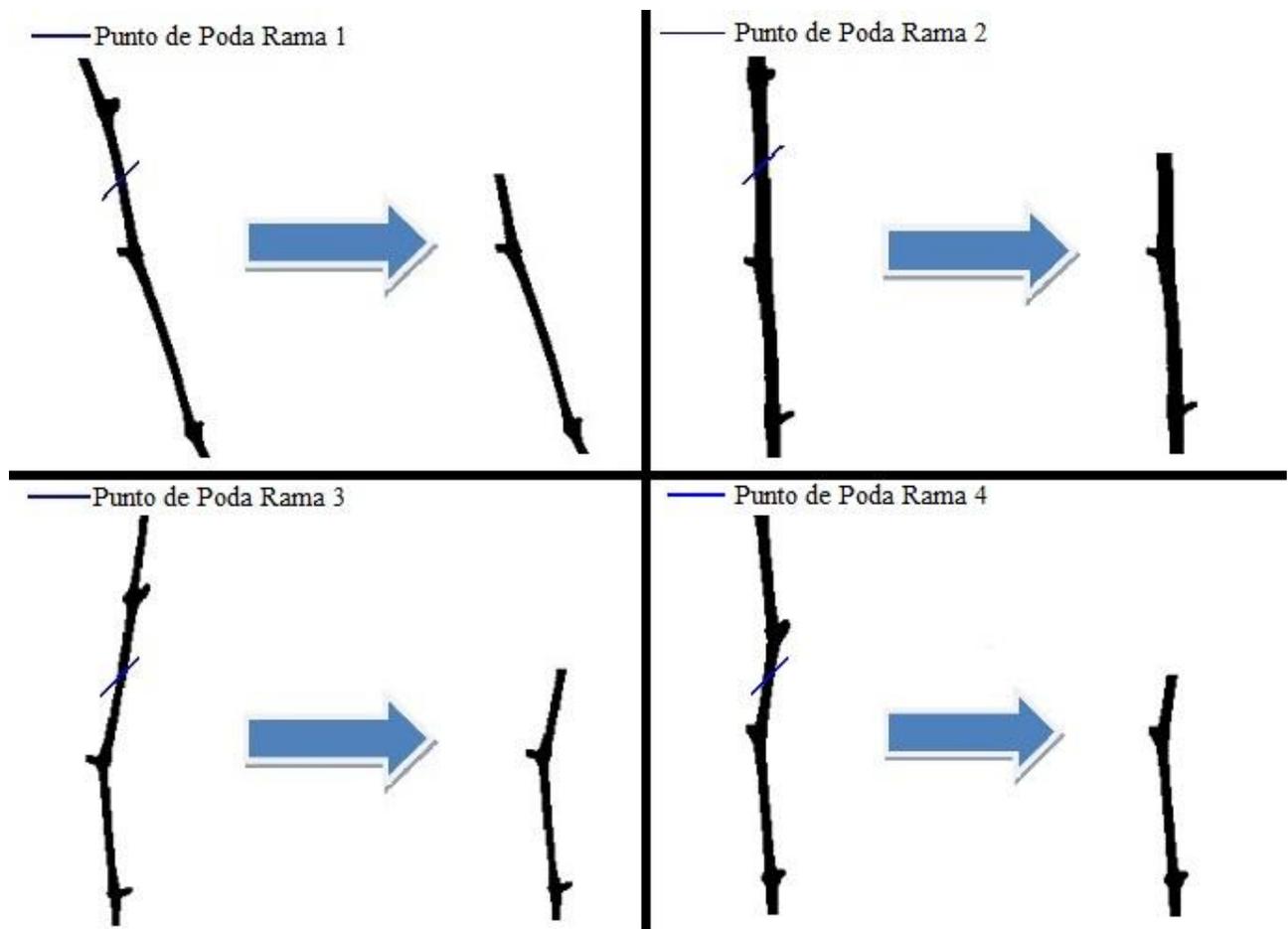


Fig. 4.41 Poda en cada rama

Paso 9:

Para finalizar los procesos que se llevan a cabo en esta ventana de la interfaz, este último paso ofrece al usuario poder ver la parra completa, pero esta vez con la indicación de donde podar las ramas y con la opción de poder ver la parra podada por el algoritmo.

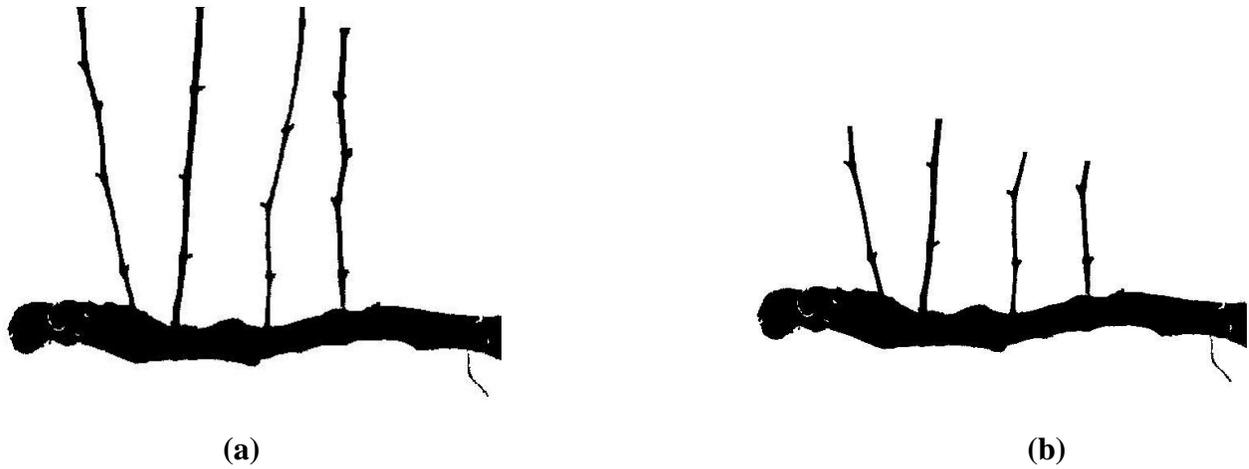


Fig. 4.52 Rectificador Fuente de Voltaje
 (a) Parra; (b) Parra podada

Paso 10:

Una vez finalizada la operación de identificación de los puntos de poda en las imágenes RGB, hacemos clic en el botón "Continuar" para acceder a la última ventana de esta interfaz, con esto se desplegará en la pantalla la ventana de reconstrucción 3D, la cual tiene por finalidad mostrar los resultados hasta aquí logrados pero en una reconstrucción de la escena en tres dimensiones. Esta parte del HMI y sus respectivos botones se muestra en la Fig.4.13.

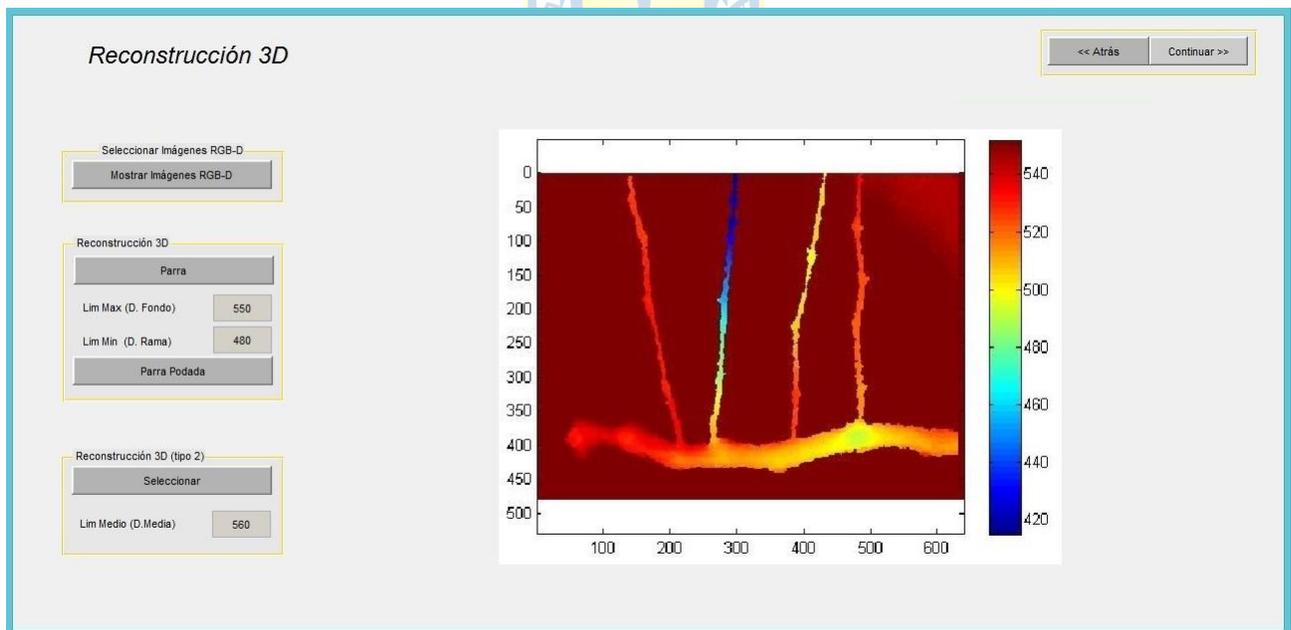


Fig. 4.63 HMI reconstrucción 3D

Al igual que en el paso 4, buscamos las imágenes en el directorio, pero en este caso seleccionamos la imagen de profundidad.

Paso 11:

Debemos ingresar ciertos valores en este paso, estos son "*LimMax*" y "*Lim Min*". El primero está relacionado con la distancia desde donde se encuentra el sensor Kinect hasta donde está el telón blanco y el segundo corresponde a la distancia medida entre el sensor y la parra. Estos valores permitirán al algoritmo desplegar en pantalla la imagen de rango 3D cuando se haga clic en el botón "*Parra*", la cual se muestra en la Fig.4.13 y la figura que se desplegará en el caso de hacer clic en "*Parra Podada*" la vemos en la Fig.4.14. Además, al hacer clic en los botones antes mencionados se desplegará en pantalla nuevas ventanas que contendrán gráficas de mapas tridimensionales, los cuales se podrán rotar para ver diferentes perspectivas de la parra, en la Fig.4.15 notamos esto.

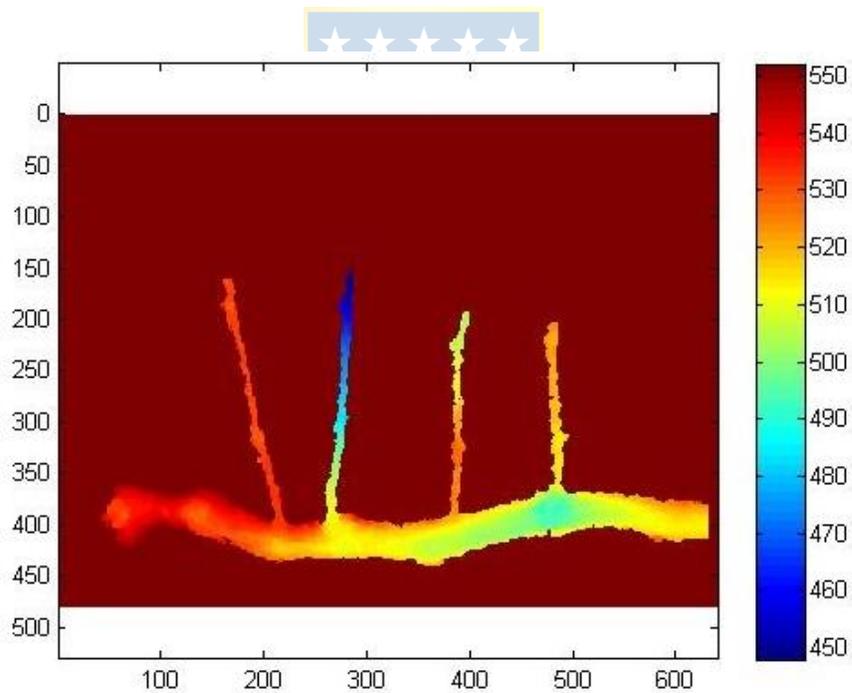
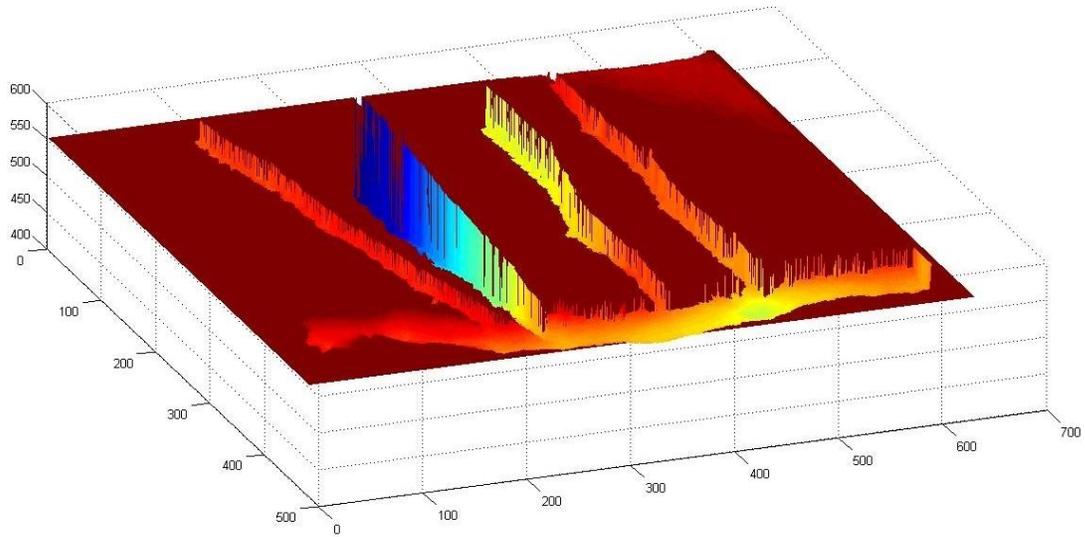
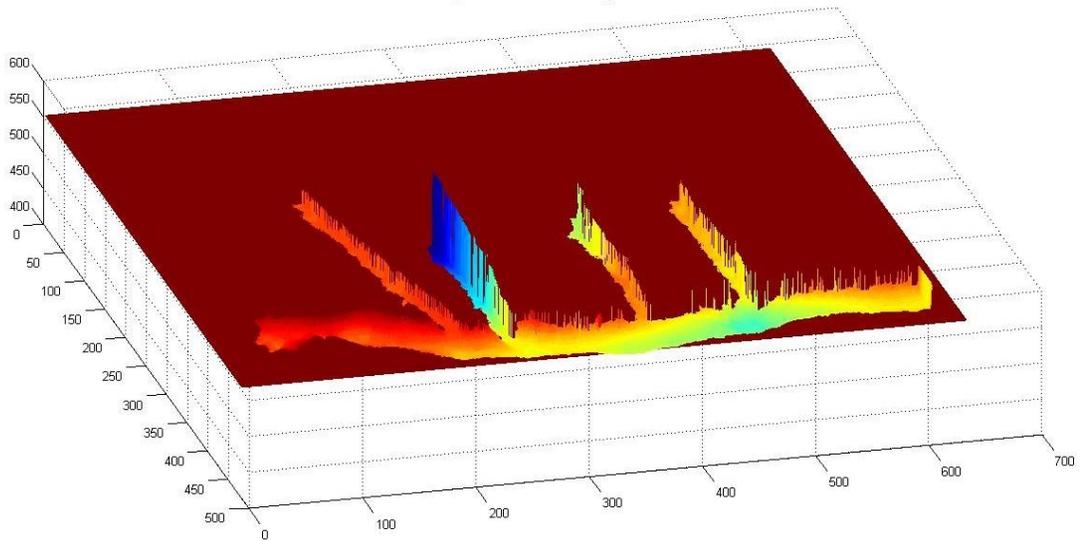


Fig. 4.14. Imagen de rango de profundidad.



(a)



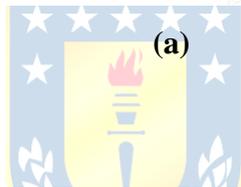
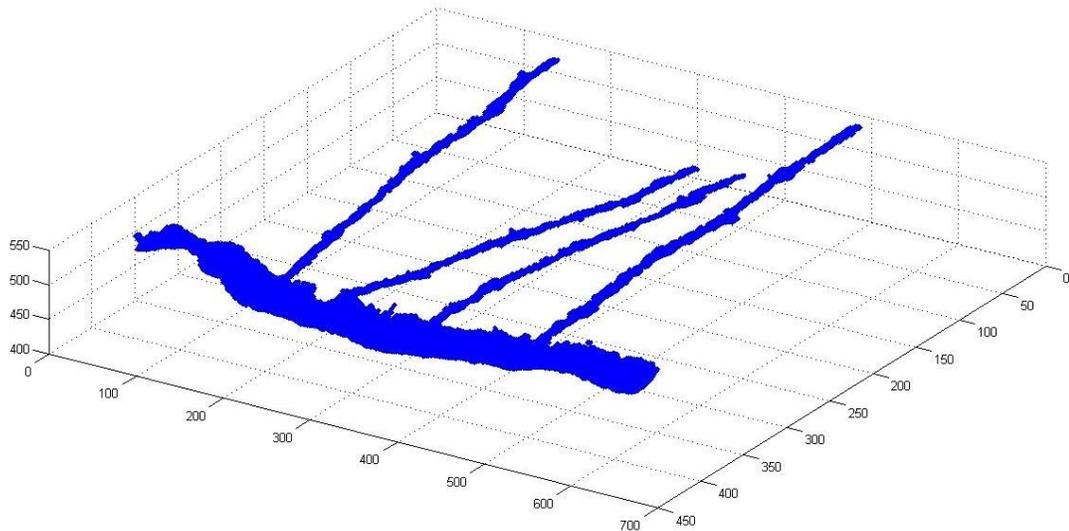
(b)

Fig. 4.75 Mapa tridimensional
(a) Parra; (b) Parra podada

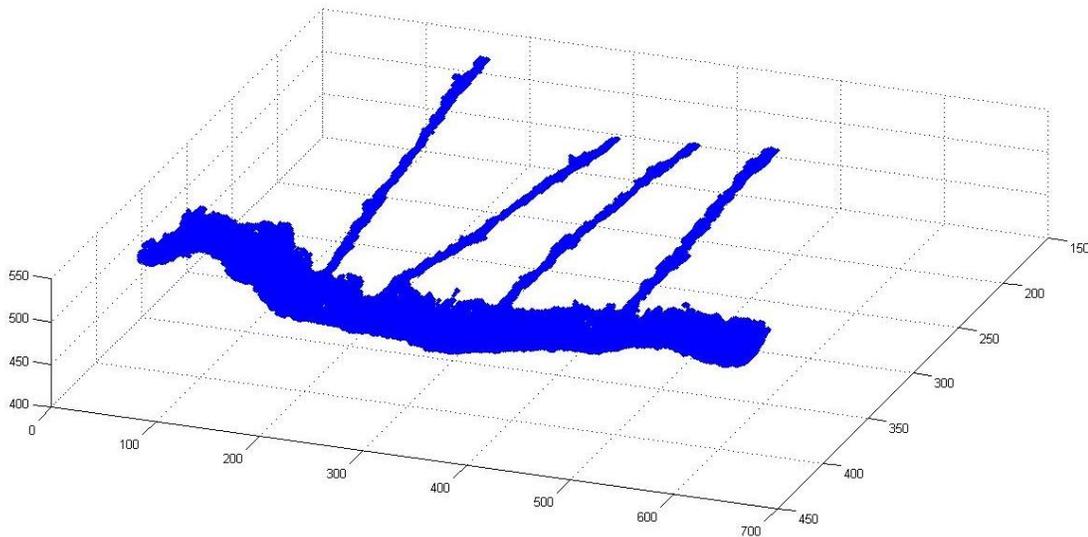
Paso 12:

Este es el último paso de la interfaz desarrollada en este proyecto y al igual que el paso anterior, consiste en desplegar la imagen de la reconstrucción en tres dimensiones de la parra pero eliminando el fondo, y al igual como se vió en la sección de reconstrucción 3D obtener solamente la parra en tres dimensiones. El parámetro impuesto para esta función es "*Lim Medio*", que se vincula netamente con la distancia que hay entre el sensor Kinect y alguna distancia que hay en el espacio

existente entre la parra y el telón, así se puede eliminar la pared y dejar solamente los datos de la parra. En las siguientes figuras veremos la parra podada y sin podar.



(a)



(b)

Fig. 4.8 Reconstrucción 3D
(a) Parra; (b) Parra podada.

Capítulo 5. Conclusiones

5.1. Introducción General

Luego de haber finalizado con todo el estudio y pruebas experimentales, el propósito de este capítulo es concluir el trabajo hasta aquí realizado haciendo una evaluación sobre sus objetivos y los aportes o contribuciones que este nos pudo proporcionar.

5.2. Conclusiones

En el presente estudio, un nuevo método se presenta para la poda de las parras de uva. En concreto, los algoritmos que utilizan procesamiento de imágenes, análisis de imagen y la reconstrucción 3D se desarrollaron para localizar las posiciones de corte en la parra, siendo la técnica de poda 2do brote el método escogido para realizar esta tarea.

Al inicio, se presentó una revisión sobre los trabajos previos relacionados con los temas a tratar, seguido de una descripción sobre la metodología de este trabajo. La metodología incluye la introducción al procesamiento de imágenes, las técnicas usadas para el análisis de estas y la reconstrucción 3D. El propósito de la etapa de procesamiento es preparar las imágenes para el posterior análisis que se lleva a cabo y los resultados obtenidos en estos procesos se muestran en el capítulo 4, y se logra con el objetivo principal en la localización de las posiciones de poda. Hay que acotar que un ejemplar de la parra fue llevada al laboratorio de control, en el Departamento de Ingeniería Eléctrica, donde se montó sobre un pedestal, además de un telón que fue fijado sobre una de las paredes del laboratorio, por otro lado la Kinect estaba fija sobre un pedestal al momento de capturar las imágenes, con esto se diseñó el set-up experimental para realizar el estudio.

El algoritmo sobre el análisis de imagen en la ubicación de la posición de los puntos de corte se ha desarrollado con éxito, este fue realizado sobre las imágenes RGB o 2D. Además, se incluyó el dispositivo Microsoft Kinect en este estudio, que es considerado un avance tecnológico de los últimos años y se aprovecharon sus potenciales características para lograr llevar a cabo una tarea que es muy común y tradicional como lo es la poda. En este ámbito, hay que notar que existen cada vez menos expertos en poda de las vides, lo cual ha causado preocupación en el medio, considerando esto, el actual trabajo puede ser un gran aporte y puede dar estos beneficios a personas que no saben

podar una parra o también el HMI puede servir para capacitar a estas personas.

Luego en la reconstrucción 3D también se logró realizar esta representación mediante las imágenes de profundidad proporcionadas por el dispositivo Microsoft Kinect, aunque hubo algunas dificultades con respecto a este punto. Debido a lo delgadas que son algunas ramas, a la hora de capturar las imágenes, se apreció el problema de que en las imágenes de disparidad salían ramas cortadas o solamente parte de estas. Esto se debe a que cuando el proyector de luz infrarroja proyectaba el patrón de puntos sobre el plano de la escena, como hay ramas muy delgadas, la cámara infrarroja lograba captar solamente algunas partes de estas, por lo que para la finalidad de este proyecto era un gran problema, ya que se necesita la visualización de la rama completa. El año 2013 fue lanzado a la venta el dispositivo Kinect 2.0, que es una versión mejorada de la que se usó en este estudio, la cual presenta varias mejoras donde dispone de una cámara de resolución 2080p con un nivel de precisión y reconocimiento superior, además permite procesar 2GB de datos por segundo y captura 30 imágenes por segundo. Con esto se podría mejorar sustancialmente el funcionamiento del algoritmo de visión desarrollado, sobre todo para la reconstrucción 3D.

Por último, el objetivo del telón blanco es fundamental para el algoritmo, con este se logra distinguir mejor el fondo de la parra. Esto se realiza porque el fondo es generalmente muy complejo y las cámaras pueden capturar otros objetos que se mezclarían con la parra si es que no existiera un telón, lo que conllevaría a errores en el algoritmo.

5.3. Trabajo Futuro

1. Desarrollar de mejor forma el método utilizado en la ubicación del punto óptimo en las imágenes 2D. Esto se refiere, a usar otros criterios para la localización de este punto, ya que como se vio en el capítulo 3, con el algoritmo desarrollado en este estudio existen algunos problemas en algunas ocasiones para localizar el cordón o localizar los nodos.
2. Aprovechar las características de resolución y rapidez en la adquisición de datos de la nueva versión del dispositivo Microsoft Kinect 2.0.
3. Realizar pruebas del algoritmo al aire libre, en el entorno de las parras de uva.
4. Automatizar la poda de las parras, aprovechando la tecnología Kinect y realizar este trabajo en tiempo real.

5. Utilizar varios sensores Kinect desde diferentes ángulos de visión para así obtener diversas datos tridimensionales y con esto de lograr una representación 3D completa de todos los ángulos y no solamente de la parte frontal de la parra.



Bibliografía

- [1] D. Bulanon, T. Kataoka, Y. Ota, and T. Hirao. "A segmentation algorithm for automatic recognition of Fuji apples at harvest," *The United Graduate School of Agricultural Sciences, Biosystems Engineering(2002)*, vol., no., pp. 405-412, 14 Aug. 2002.
- [2] "Machine vision: theory, algorithms, practicalities" E. Davies., *2da edicion*, Elsevier Inc. 2005.
- [3] D. Mery. "Vision por Computador," Departamento de Ciencia de la Computacion, Universidad Catolica de Chile, Santiago de Chile, Agosto. 2004.
- [4] W. Burger, M. Burge. "Digital ImageProcessing," ISBN: 978-1-84628-379-6, 2008, pp. 13.
- [5] G. Ming. "Image Processing and Analysis for Autonomous Grapevine Pruning", *2006 IEEE, International Conference onMechatronics and Automation*), vol., no., pp. 922-927, 25-26 June. 2006.
- [6] J. Smisek, M. Jancosek, T. Pajdla. "3D with Kinect". *2011 IEEE International Conference on Computer Vision (ICCV Workshops)*, pp. 1154-1160, 6-13 Nov. 2011.
- [7] C. Hernández, "Desarrollo de Procedimientos Calibración de Escaner 3D de Luz Infrarroja Estructurada para Caracterización de Distribución de Tamaño Granulométrico en Correas Transportadoras", Memoria de Título, Ingeniero Civil Electrónico, Marzo 2013, Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Concepción.
- [8] G. Perez. "Operaciones Manuales en Viñedo," *Centros de Formacion Agraria Castilla y Leon*, Dec. 2005, [2].
- [9] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor enviroments". *The International Journal of RoboticsResearch*, 2012.
- [10] K Khoshelham, "Accuracy analysis of Kinect depth data", *Remote Sensing and Spatia lInformation Sciences, ISPRS Calgary 2011 Workshop*, vol., no., pp.133-138, Aug.2011.
- [11] Vincent, L.; Soille, P.; , "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *Pattern Analysis and Machine Intelligence, IEEE Transactionson* , vol.13, no.6, pp.583-598, Jun 1991.

- [12] U. Dugan, R. Dongseok, K. Myungjoon, "Multiple Intensity Differentiation for 3-D Surface Reconstruction With Mono-Vision Infrared Proximity Array Sensor," *Sensors Journal, IEEE*, vol.11, no.12, pp.3352-3358, Dec. 2011 [9]
- [13] Abramov, A.; Pauwels, K.; Papon, J.; Worgotter, F.; Dellen, B.; , "Depth-supported real-time video segmentation with the Kinect," *Applications of Computer Vision (WACV), 2012 IEEE Work shop on*, vol., no., pp.457-464, 9-11 Jan. 2012. [10]
- [14] D. Herrera, J. Kannala, and J. Heikkilä. "Accurate and Practical Calibration of a Depth and Color Camera Pair," *Proceedings of the 14th International Conference, CAIP 2011, Part II*, vol. 6855, pp. 437-445, 29-31 Aug. 2011, [10].
- [15] D. Herrera, J. Kannala, and J. Heikkilä. "Joint Depth and Color Camera Calibration with Distortion Correction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.34, no.10, pp. 2058-2064, Oct. 2012, [11].
- [16] Daniel Herrera. Center for Machine Vision Research, University of Oulu. [Online] <http://www.ee.oulu.fi/~dherrera/>.
- [17] M. Saiz, "Reconstrucción Tridimensional Mediante Visión Estéreo y Técnicas de Optimización", Proyecto Fin de Carrera, Ingeniero Industrial, Junio 2010, Escuela Técnica Superior de Ingeniería(ICAI), Universidad Pontificia Comillas.
- [18] Corbett-Davies, S; Botterill, T; Green, R; Saxton, V. "An expert system for automatically pruning", *IVCNZ'12*, November 26-28 2012, Dunedin, New Zealand 2012 ACM 978-1-4503-1473-2/12/11.
- [19] Cristóbal Obregón Pérez, "Diseño de sistema visión para monitoreo", Memoria de Título, Ingeniero Civil Electrónico, Abril 2009, Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Concepción.
- [20] Gokturk, S.B.; Yalcin, H.; Bamji, C.; , "A Time-Of-Flight Depth Sensor - System Description, Issues and Solutions," *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, vol., no., pp. 35, 27-02 June 2004.

- [21] Lopez, M.; Vilan, J.A.; Matias, J.M.; Taboada, J.; , "Quality Control of Wood-Pulp Chips Using A 3D Laser Scanner and Functional Pattern Recognition," *Industrial Electronics*, 2007. ISIE 2007. IEEE International Symposium on , vol., no., pp.1773-1778, 4-7 June 2007.
- [22] Pedro Ignacio Orellana Rueda, "Desarrollo de Sistema para la Medición de Distribución Granulométrica con un Escáner Tridimensional", Memoria de Título, Ingeniero Civil Electrónico, Abril 2013, Departamento de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Concepción.
- [23] W. Qi, F. Li, and L. Zhenzhong. "Review on Camera Calibration," *2010 Chinese Control and Decision Conference (CCDC)*, pp. 3354-3358, 26-28 May. 2010.
- [24] J. Heikkilä and O. Silvén. "Calibration Procedure for Short Focal Length Off-the-Shelf CCD Cameras," *Proceedings of the 13th International Conference on Pattern Recognition, 1996.*, vol. 1, pp. 166-170, 25-29 Aug. 1996.
- [25] J. Heikkilä and O. Silvén. "A Four-step Camera Calibration Procedure with Implicit Image Correction," *1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997. Proceedings.*, pp. 1106-1112, 17-19 Jun. 1997.
- [26] Pagina Kinect y Matlab, <http://www.mathworks.com/matlabcentral/fileexchange/30242-kinect-matlab>.
- [27] Javier Nuño Simon, "Reconocimiento de objetos mediante sensor 3D Kinect", Proyecto Fin de Carrera, Ingeniería Electrónica y Automática, Septiembre 2012, Departamento de Ingeniería de Sistemas y Automática, Universidad de Madrid.
- [28] Zhengyou Zhang. "Microsoft Kinect Sensor and Its Effect," *MultiMedia IEEE*, vol.19, no. 2, pp. 4-10, Feb. 2012.