



ESCUELA POLITÉCNICA DEL EJÉRCITO  
EXTENSIÓN LATACUNGA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

“DESARROLLO E IMPLEMENTACIÓN DE UN ALGORITMO  
PARA DETECCIÓN DE OBJETOS CON TECNOLOGÍA KINECT”

NOMBRE DEL AUTOR

VELASCO ERAZO NANCY DEL ROCÍO

Tesis presentada como requisito previo a la obtención del grado  
de:

INGENIERO EN ELECTRÓNICA E INSTRUMENTACIÓN

AÑO 2013

ESCUELA POLITÉCNICA DEL EJÉRCITO  
CARRERA DE INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

DECLARACIÓN DE RESPONSABILIDAD

Nancy del Rocío Velasco Erazo

DECLARO QUE:

El proyecto de grado denominado Desarrollo e Implementación de un algoritmo para detección de objetos con tecnología Kinect, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan el pie de las páginas correspondiente, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, Abril del 2013.

---

Nancy Velasco E.

ESCUELA POLITÉCNICA DEL EJÉRCITO  
CARRERA DE INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

**CERTIFICADO**

Ing. Eddie Galarza (Director)

Ing. David Rivas (Codirector)

**CERTIFICAN**

Que el trabajo titulado “Desarrollo e Implementación de un algoritmo para detección de objetos con tecnología Kinect” realizado por Nancy del Rocío Velasco Erazo, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Escuela Politécnica del Ejército.

Debido a que constituye un trabajo de excelente contenido científico que coadyuvará a la aplicación de conocimientos y al desarrollo profesional, SI recomiendan su publicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a Nancy del Rocío Velasco Erazo que lo entregue a la Ing. Nancy Guerrón en su calidad de Directora de la Carrera.

Latacunga, Abril del 2013.

---

Ing. Eddie Galarza  
DIRECTOR

---

Ing. David Rivas  
CODIRECTOR

ESCUELA POLITÉCNICA DEL EJÉRCITO  
CARRERA DE INGENIERÍA ELECTRÓNICA E INSTRUMENTACIÓN

**AUTORIZACIÓN**

Yo, Nancy del Rocío Velasco Erazo

Autorizo a la Escuela Politécnica del Ejército la publicación, en la biblioteca virtual de la Institución del trabajo “Desarrollo e Implementación de un algoritmo para detección de objetos con tecnología Kinect”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Latacunga, Abril del 2013.

---

Nancy Velasco E.

## DEDICATORIA

A mi familia, quienes supieron darme el aliento y la fuerza necesaria para continuar y alcanzar la meta anhelada, depositando su entera confianza en cada reto que se me presentaba, a mis Padre Ramiro, a mi Hermano José Ramiro, y de manera especial a mi madre Carmen, pilar de la familia y mi hermana Ruth, quien siempre estuvo ahí para mí apoyándome en todo, cada uno de ellos me han enseñado a no doblar la espalda cuando hay que erigir el pecho y a no contentarme con sólo desear en lugar de realizar, sin ellos, jamás hubiese podido conseguir lo que hasta ahora, su tenacidad y lucha insaciable han hecho de ellos el gran ejemplo a seguir y destacar porque me han dado todo lo que soy como persona, mis valores, mis principios, mi perseverancia, mi empeño.

A todas las personas por el apoyo que me han proporcionado para la culminación de este propósito ya que con sus frases de aliento y felicitación no me hicieron desmayar en este objetivo.

***Nancy***

## **AGRADECIMIENTO**

*“Serás del tamaño de tus pensamientos, no te permitas fracasar.”*

*David Bustamante*

A Dios, por haberme permitido cumplir con mi meta profesional.

A mi familia por su apoyo incondicional en cada paso de mi vida.

A todos los señores profesores de la especialidad, que con respeto y vocación han hecho todo lo posible por transmitir todos sus conocimientos en las aulas de clases y sobre todo por resaltar los valores que hacen de nosotros verdaderos humanistas, sin dejar de lado a mis asesores de tesis Ing. Eddie Galarza e Ing. David Rivas los cuales respeto y estimo por su ayuda, paciencia, optimismo y colaboración que me han brindado.

A mis compañeros y amigos con quienes compartí varias de las experiencias más enriquecedoras de mi existencia.

A todos aquellos que durante los años que duró este sueño lograron convertirlo en una realidad.

***Nancy***

## ÍNDICE DE CONTENIDOS

DECLARACIÓN DE RESPONSABILIDAD .....	i
CERTIFICADO DE TUTORÍA.....	ii
AUTORIZACIÓN DE PUBLICACIÓN.....	iii
DEDICATORIA .....	iv
AGRADECIMIENTO .....	v
ÍNDICE DE CONTENIDOS.....	vi
LISTADO DE FIGURAS.....	xiii
LISTADO DE TABLAS.....	xviii
LISTADO DE ANEXOS.....	xix
RESUMEN.....	1
ABSTRACT .....	2
1. FUNDAMENTOS TEÓRICOS.....	3
1.1 INTRODUCCIÓN .....	3
1.2 IMPORTANCIA DEL PROYECTO.....	3
1.3 SISTEMAS DE DETECCIÓN DE OBJETOS .....	4
1.3.1 Dispositivos de navegación para no videntes.....	4
<i>a. El bastón blanco.....</i>	<i>4</i>
<i>b. Bastón blanco electrónico.....</i>	<i>5</i>
<i>c. Bastón vibratorio para invidentes .....</i>	<i>5</i>
<i>d. Gafas con sistema GPS.....</i>	<i>6</i>
<i>e. Mi Ultrasonic .....</i>	<i>6</i>

<i>f. Vara para ciegos</i> .....	7
<i>g. Perro guía</i> .....	7
1.3.2 Robots móviles en interiores.....	8
<i>a. Sensor Ultrasónico</i> .....	8
<i>b. Sensor infrarrojo</i> .....	9
<i>c. Sensor radar</i> .....	10
1.4 MICROSOFT KINECT .....	10
1.4.1 Partes fundamentales del Hardware.....	11
1.4.2 Técnica para capturar la profundidad .....	14
1.4.3 Información de profundidad.....	16
1.4.4 Limitaciones.....	19
1.5 SOFTWARE PARA CONTROLAR EL SENSOR KINECT.....	21
1.5.1 Microsoft Kinect SDK.....	22
1.5.2 OpenNI SDK.....	23
1.5.3 Open Kinect.....	23
1.5.4 Microsoft Kinect SDK vs. Open Source .....	23
1.6 PROCESAMIENTO DIGITAL DE IMÁGENES .....	25
1.6.1 Vision Artificial .....	25
1.6.2 Representación digital de una imagen.....	26
1.6.3 Etapas del Procesamiento Digital de Imágenes .....	26
<i>a. Adquisición de la imagen</i> .....	26
<i>b. Procesamiento previo</i> .....	27
<i>c. Extracción de características relevantes</i> .....	27
<i>d. Interpretación</i> .....	28
1.6.4 Tipos de procesamiento .....	28

a. Segmentación .....	28
a.1. Técnicas de Umbralización .....	29
a.2. Técnicas basadas en regiones .....	29
a.3. Detección de Bordes.....	30
b. Binarización.....	31
c. Segmentación basada en morfología matemática .....	31
d. Relleno de región.....	32
e. Determinación de la posición.....	32
f. Etiquetado de una imagen.....	33
1.7 SOFTWARE LABVIEW .....	33
1.8 MICROSOFT SPEECH SDK .....	34
1.8.1 Narrador de Windows.....	34
2. DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO .....	35
2.1 INSTALACIÓN DEL SOFTWARE.....	35
2.1.1 LabVIEW 2011 o superior con NI Vision Development Module. ....	35
2.1.2 Kinect SDK 1.5 .....	35
2.1.3 Microsoft. NET Framework 4.0 .....	37
2.1.4 Microsoft Speech SDK .....	38
2.1.5 Voz adicional en español para Text to Speech. ....	39
2.2 INSTRUCCIONES DEL MICROSOFT KINECT SDK .....	40
2.2.1 Clase KinectSensor .....	40
a. Método Start() .....	41
b. Método Stop.....	41
c. Propiedades .....	41

2.2.2 KinectSensorCollection .....	42
a. Método <i>Get_Item</i> .....	42
b. Propiedad <i>Count</i> .....	43
2.2.3 Clase <i>ColorImageStream</i> .....	43
a. Método <i>Enable</i> .....	43
c. Método <i>OpenNextFrame</i> .....	43
b. Propiedad <i>Format</i> .....	43
2.2.4 <i>DepthImageStream</i> .....	44
a. Método <i>Enable</i> .....	44
b. Método <i>OpenNextFrame</i> .....	44
c. Propiedad <i>Format</i> .....	45
2.2.5 Clase <i>ImageStream</i> .....	45
a. Propiedades .....	45
b. Método <i>Disable</i> .....	45
2.2.6 <i>DepthImageFrame</i> Class .....	46
a. Método <i>CopyDepthImagePixelDataTo</i> .....	46
b. Método <i>CopyPixelDataTo</i> .....	46
2.3 HERRAMIENTAS MICROSOFT SPEECH SDK .....	46
2.4 HERRAMIENTAS DE LABVIEW .....	48
2.4.1 .NET .....	48
2.4.2 Controles ActiveX .....	48
2.2.3 Herramientas del .NET y Controles ActiveX.....	49
a. <i>Automation Open/Close Reference</i> .....	49
b. <i>Property Node/ Invoke Node</i> .....	50
2.4.4 Creación métodos y propiedades de una clase.....	51

2.4.5 Toolkit IMAQ Vision .....	52
a. <i>Crear imagen</i> .....	53
b. <i>Copiar imagen</i> .....	53
c. <i>Imagen a un arreglo</i> .....	54
d. <i>Arreglo a imagen</i> .....	54
e. <i>Simetría de la imagen</i> .....	55
f. <i>Superponer líneas</i> .....	55
g. <i>Crear casco convexo</i> .....	56
h. <i>Contar objetos</i> .....	56
i. <i>Detectar los bordes</i> .....	61
j. <i>Image Display (panel frontal)</i> .....	63
2.4.6 Herramientas adicionales .....	64
a. <i>Simple Error</i> .....	64
b. <i>Error in / Error out</i> .....	64
2.5 ETAPAS DEL ALGORITMO .....	65
2.5.1 Etapa 1: Interface Kinect.....	67
a. <i>Obtener Información</i> .....	67
b. <i>Inicializar el Kinect</i> .....	69
c. <i>Establecer el ángulo de elevación</i> .....	72
d. <i>Obtener la imagen RGB</i> .....	73
e. <i>Obtener la imagen de profundidad</i> .....	74
f. <i>Detener el sensor Kinect</i> .....	76
2.5.2 Etapa 2: Transformar a una profundidad real.....	78
2.5.3 Etapa 3: Eliminar errores de adquisición .....	78
2.5.4 Etapa 4: Filtrar imagen por distancia y detección de bordes .....	79

2.5.5 Etapa 5: Segmentar los objetos .....	80
2.5.6 Etapa 6: Establecer límites de posiciones .....	81
2.5.7 Etapa 7: Determinar la distancia al centro del objeto .....	82
2.5.8 Etapa 8: Texto a voz.....	83
2.5.9 Etapa 9: Párametros de entrada y salida.....	89
2.6 INSERTAR EL VI EN EL DIAGRAMA DE BLOQUES .....	90
3. PRUEBAS Y ANÁLISIS DE RESULTADOS .....	92
3.1 Pruebas Experimentales .....	92
3.1.1 Prueba de adquisición .....	93
3.1.2 Prueba para eliminar errores de adquisición .....	94
3.1.3 Prueba del filtro en profundidad .....	96
3.1.4 Prueba según la localización.....	97
3.1.5 Prueba de forma de objetos .....	100
3.1.6 Prueba de Tamaño.....	105
3.1.7 Prueba de profundidad.....	108
3.1.8 Prueba de materiales .....	110
3.1.9 Prueba objetos colgantes.....	111
3.1.10 Prueba de objetos transparentes .....	113
3.1.11 Prueba de objetos reflectantes .....	114
3.1.12 Prueba de objetos unidos.....	116
4. CONCLUSIONES RECOMENDACIONES.....	118
4.1 CONCLUSIONES.....	118
4.2 RECOMENDACIONES .....	120

BIBLIOGRAFÍA.....	122
ENLACES ELECTRÓNICOS.....	124
ANEXOS .....	129

## LISTADO DE FIGURAS

Figura 1.1: Bastón Blanco.....	4
Figura 1.2: Bastón blanco electrónico .....	5
Figura 1.3: Gafas con sistema GPS.....	6
Figura 1.4: Mi Ultrasonic .....	7
Figura 1.5: Vara para ciegos.....	7
Figura 1.6: Perro guía.....	8
Figura 1.7: Modo de operación del sensor Ultrasónico .....	9
Figura 1.8: Sensor Infrarrojo .....	9
Figura 1.9: Sensor Infrarrojo .....	10
Figura 1.10: Sensor Kinect para la consola Xbox360.....	11
Figura 1.11: Hardware del sensor Kinect .....	12
Figura 1.12: Proyección del patrón infrarrojo .....	15
Figura 1.13: Imagen generada por la cámara del Kinect.....	15
Figura 1.14: Diagrama de la tecnología Kinect patentada por PrimerSense	16
Figura 1.15: Forma de medir la profundidad del Kinect.....	17
Figura 1.16: Ubicación de los bits de profundidad .....	17
Figura 1.17: Rango de distancias admitidas por el sensor Kinect. ....	18
Figura 1.18: Efecto de distancias grandes en el Kinect.....	19
Figura 1.19: Efecto de las sombras en el Kinect .....	20
Figura 1.20: Efecto de superficie reflectante y paralela a los rayos de luz ...	21
Figura 1.21: Sistema de visión artificial.....	25
Figura 1.22: Vecindad a 4 (izquierda); vecindad a 8 (derecha) .....	29
Figura 1.23: Imagen original (izquierda), imagen con filtro sigma (derecha)	30
Figura 1.24: Representación de imágenes binarias .....	31
Figura 1.25: a) Obtención de los puntos extremos de un objeto; b) Obtención del centroide del objeto.....	32

Figura 1.26: Logotipo de Labview .....	33
Figura 2.1: Ventana para la instalación del Kinect SDK.....	36
Figura 2.2: Controladores del Kinect cargados automáticamente .....	37
Figura 2.3: Sitio de descarga del Microsoft .NET 4 .....	38
Figura 2.4: Sitio de descarga del Microsoft Speech SDK .....	39
Figura 2.5: Voz adicional en español instalada .....	40
Figura 2.6: Menú .NET .....	48
Figura 2.7: Menú ActiveX.....	49
Figura 2.8: Close Reference .....	49
Figura 2.9: Automation Open .....	50
Figura 2.10: Property Node.....	50
Figura 2.11: Invoke Node.....	50
Figura 2.12: Ejemplo de creación de Método.....	51
Figura 2.13: Herramientas de Vision and Motion de Labview .....	52
Figura 2.14: IMAQ Create.....	53
Figura 2.15: IMAQ Copy .....	53
Figura 2.16: IMAQ Image To Array .....	54
Figura 2.17: IMAQ Array to Image .....	54
Figura 2.18: IMAQ Symmetry .....	55
Figura 2.19: IMAQ Overlay Line.....	55
Figura 2.20: IMAQ Convex Hull .....	56
Figura 2.21: Cascos convexos a dos partículas independientes .....	56
Figura 2.22: IMAQ Count Objects 2 .....	57
Figura 2.23: IMAQ Convert Rectangle to ROI .....	57
Figura 2.24: IMAQ Extract .....	58
Figura 2.25: IMAQ Threshold.....	58
Figura 2.26: IMAQ RejectBorder .....	59
Figura 2.27: IMAQ FillHole.....	59
Figura 2.28: IMAQ Particle Filter 2 .....	59

Figura 2.29: IMAQ Particle Analysis .....	60
Figura 2.30: IMAQ Overlay ROI .....	60
Figura 2.31: IMAQ Overlay Rectangle .....	61
Figura 2.32: IMAQ Overlay Rectangle .....	61
Figura 2.33: IMAQ Overlay Text .....	61
Figura 2.34: IMAQ Edge Detection .....	62
Figura 2.35: 1) Líneas de búsqueda, 2) bordes .....	62
Figura 2.36: Panel Frontal de IMAQ Display .....	63
Figura 2.37: Simple Error Handler .....	64
Figura 2.38: Panel Frontal del Error In (izquierda) y Error Out (derecha) .....	64
Figura 2.39: Resumen del algoritmo .....	65
Figura 2.40: Etapas del algoritmo .....	66
Figura 2.41: Elementos del paquete Kinect para Labview .....	67
Figura 2.42: Get Device Info.VI.....	68
Figura 2.43: Diagrama de bloques del VI que obtiene información del Kinect. .....	68
Figura 2.44: Initialize Kinect VI.....	69
Figura 2.45: Diagrama de bloques del iniciador del Kinect .....	71
Figura 2.46: Set Elevation Angle VI .....	72
Figura 2.47: Diagrama de bloques del ajuste del ángulo .....	72
Figura 2.48: Get Color Frame .....	73
Figura 2.49: Diagrama de bloques la adquisición de la imagen de color.....	74
Figura 2.50: Get Depth Frame VI.....	74
Figura 2.51: Diagrama de bloques la adquisición de la imagen de profundidad. .....	75
Figura 2.52: Dipose Kinect VI .....	76
Figura 2.53: Diagrama de bloques del VI que detiene el Kinect.....	76
Figura 2.54: Adquisición de datos del Kinect .....	77
Figura 2.55: Transformación a una profundidad real.....	78

Figura 2.56: Eliminar errores de adquisición .....	79
Figura 2.57: Filtrar imagen por distancia y detección de bordes .....	80
Figura 2.58: Detección de objetos presentes .....	80
Figura 2.59: Establecer límites de posiciones .....	81
Figura 2.60: Delimitación de la posición de los objetos .....	82
Figura 2.61: Profundidad al centro de cada objeto.....	83
Figura 2.62: Diagrama de bloques para convertir texto a voz .....	83
Figura 2.63: Texto que va a ser reproducido .....	84
Figura 2.64: Selección de la clase SpeechLib.ISpeechVoice.....	85
Figura 2.65: Revisar las voces disponibles en la pc.....	86
Figura 2.66: Selección de una de las voces disponibles .....	86
Figura 2.67: Reproducción del texto .....	87
Figura 2.68: Programa completo para la Interface texto a voz .....	88
Figura 2.69: Parámetros de entrada y salida .....	90
Figura 2.70: Localización del VI en la paleta de funciones.....	91
Figura 2.71: Ícono en Labview listo para ser utilizado .....	91
Figura 3.1: Prueba de adquisición.....	94
Figura 3.2: Simetría de las imágenes.....	95
Figura 3.3: Eliminación de error lateral en la imagen de profundidad.....	96
Figura 3.4: Prueba del filtro en profundidad .....	96
Figura 3.5: Prueba de localización centro e izquierda .....	97
Figura 3.6: Prueba de localización obstáculo total e izquierda .....	98
Figura 3.7: Prueba de localización centro derecha e izquierda .....	99
Figura 3.8: Prueba de objetos rectangulares .....	101
Figura 3.9: Prueba de objetos circulares.....	102
Figura 3.10: Prueba de objetos cilíndricos .....	103
Figura 3.11: Prueba de otras formas de objetos .....	104
Figura 3.12: Prueba de objetos pequeños .....	106
Figura 3.13: Prueba de objetos grandes .....	107

Figura 3.14: Prueba de profundidad .....	109
Figura 3.15: Prueba de materiales .....	110
Figura 3.16: Prueba de objeto colgante .....	112
Figura 3.17: Prueba de objetos transparentes .....	113
Figura 3.18: Prueba de objetos reflectantes .....	115
Figura 3.19: Prueba de objetos unidos físicamente .....	116
Figura 3.20: Prueba de objetos unidos respecto al ángulo de visión del Kinect .....	117

## LISTADO DE TABLAS

Tabla 1.1: Especificaciones del Kinect.....	13
Tabla 1.2: Ventajas del SDK de Microsoft y del OpenNI. ....	24
Tabla 1.3: Desventajas del SDK de Microsoft y del OpenNI. ....	24
Tabla 2.1: Propiedades de la clase KinectSensor.....	41
Tabla 2.2: Propiedades de la clase ImageStream.....	45
Tabla 2.3: Herramientas del Microsoft Speech SDK.....	47
Tabla 3.1: Resultado de la prueba de localización.....	100
Tabla 3.2: Resultado de la prueba de forma de objetos.....	105
Tabla 3.3: Resumen de la prueba de tamaño.....	108
Tabla 3.4: Resultado de la prueba de materiales.....	111

## LISTADO DE ANEXOS

Anexo A: Especificaciones técnicas del kinect.....	130
Anexo B: Diagrama de bloques del algoritmo de detección de objetos y los SubVIs que lo conforman.....	133
Anexo C:Tamaño de imagen de acuerdo a la altura seleccionada.....	134
Anexo D: Valor mínimo del filtro tras la detección del borde .....	135
Anexo E: Texto a ser hablado en inglés y español .....	137
Anexo F: Respuesta a diferente tipo de objetos.....	138
Anexo G: Especificaciones del algoritmo .....	139
Anexo H: Guía de instalación .....	140
Anexo I: Manual de operación del vi .....	145

## **RESUMEN**

Las tecnologías de visión robótica artificial son una realidad hoy en día y la detección de objetos llega a ser crucial para distinguir entre los obstáculos y el ser humano cuando comparten el mismo espacio de trabajo.

En este proyecto se ha desarrollado un algoritmo para la detección de obstáculos con el sensor Kinect de la consola XBox 360 y tiene adicionalmente una salida de audio sobre la información procesada mediante la ayuda de software Labview y su Toolkit de Visión Artificial.

En el primer capítulo del proyecto se desarrolla los fundamentos teóricos de la detección de obstáculos, el sensor Microsoft Kinect, el procesamiento de imágenes y el conversor de texto a voz de Windows.

En el segundo capítulo, se detalla cada una de las partes del algoritmo en lenguaje G, así como también los requisitos, herramientas y funciones básicas que lo integran para a través de la experimentación obtener los resultados deseados.

En el tercer capítulo se indican las pruebas realizadas y los resultados obtenidos.

El cuarto capítulo muestra las conclusiones y recomendaciones durante toda la ejecución del proyecto.

Finalmente, cabe subrayar que el proyecto contiene figuras, tablas y anexos que ayudan al entendimiento y reproducción total del proyecto, desde el análisis hasta la implementación y funcionamiento, dejando abierta la posibilidad de extender el alcance del algoritmo.

## **ABSTRACT**

The artificial robotic vision technologies are a reality today and object detection becomes crucial to distinguish between obstacles and humans when they share the same workspace.

This project has developed an algorithm for detecting obstacles with Kinect sensor of Xbox 360 and in addition has an audio output on the information processed by the Labview software and Artificial Vision Toolkit.

The first chapter of this project analyzes the theoretical basis of the detection of obstacles, the Microsoft Kinect sensor, image processing and text to speech converter for Windows.

In the second chapter is detailed each of the parts of the algorithm in G language, as well as the requirements, tools and basic functions, for through the experimentation obtain the desired results.

In the third chapter are listed the performed tests and the obtained results.

Chapter fourth shows the conclusions and recommendations obtained during the project implementation.

Finally, it should be emphasized that the project contains figures, tables and appendices that help to understand and reproduce the project, from analysis to implementation and operation, leaving open the possibility of extending the scope of the algorithm.

# **CAPÍTULO 1**

## **FUNDAMENTOS TEÓRICOS**

### **1.1 INTRODUCCIÓN**

La reciente aparición de la herramienta Microsoft Kinect ha sido un rotundo éxito en el mundo de la informática, numerosos equipos de investigación de todo el mundo están desarrollando diferentes aplicaciones para campos cada vez más variados, separándose del campo de los videojuegos que era su objetivo.

Gracias a este sensor tan poderoso y al mismo tiempo económico muchas personas han conseguido hacer desarrollos muy productivos como por ejemplo siguiendo la rama de detección de movimiento, hacer traducción de lenguajes corporales o usarlo para el área de la educación y salud.

En la actualidad con el avance de la electrónica y de la informática se puede diseñar algoritmos realmente potentes, por lo tanto las posibilidades de desarrollo han crecido exponencialmente.

### **1.2 IMPORTANCIA DEL PROYECTO**

En el mundo se estima hay entre 40 y 45 millones de personas con discapacidades visuales, además, gran parte habita en comunidades de bajo nivel socioeconómico. Los invidentes necesitan de herramientas que ayuden en su desplazamiento para que puedan caminar con mayor seguridad y puedan prescindir de ayudas tales como un bastón, un perro o un ayudante personal.

Por otra parte, los robots móviles para interiores utilizados para traslado de material en almacenes o para procesos de producción, tienen precios

elevados, por lo que la necesidad de detectar obstáculos en tiempo real a precios más accesibles se vuelve una tarea primordial.

A través de este proyecto se pretende fomentar en los estudiantes e ingenieros electrónicos en instrumentación la realización de investigaciones de nuevos campos en donde se pueda aplicar los conocimientos adquiridos, mediante el uso de los más recientes avances tecnológicos con las cuales se puede ayudar al bienestar y comodidad de las personas.

### **1.3 SISTEMAS DE DETECCIÓN DE OBJETOS**

#### **1.3.1 Dispositivos de navegación para no videntes<sup>1</sup>**

Los detectores de obstáculos intentan mejorar la orientación y el movimiento del usuario, dando información espacial de elementos relevantes, algunos se detallan a continuación.

##### **a. El bastón blanco**

La información de objetos distantes que podría ser brindada por la vista, es recolectada por receptores táctiles en la mano al tocar el objeto con el bastón. Tiene tres funciones básicas: distintivo, protección e información. Su apariencia se puede observar en la Figura 1.1.



**Figura 1.1: Bastón Blanco<sup>2</sup>**

---

<sup>1</sup> MARTINEZ, Alain. "Bastón blanco para prevenir obstáculos". Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco. México D.F, Agosto 2012.

<sup>2</sup> "Bastón blanco". [En línea]. Recuperado el 08 de marzo del 2013, Disponible en Web: <[http://es.wikipedia.org/wiki/Bast%C3%B3n\\_blanco](http://es.wikipedia.org/wiki/Bast%C3%B3n_blanco)>

Existen varios tipos de bastones, así como una gama de modelos. Los más comunes utilizados en movilidad son el bastón símbolo, el bastón guía y el bastón largo. Los dos primeros se utilizan por deficientes visuales y el último por ciegos.

### **b. Bastón blanco electrónico**

Es un bastón electrónico para ciegos, que mide la distancia de los objetos mediante rayos láser y transmite la información correspondiente al usuario a través de sonidos o vibraciones emitidos por un pequeño dispositivo del tamaño de un mando de televisión que se lleva adherido a la mano. En la Figura 1.2 se puede observar que el bastón electrónico tiene externamente la misma apariencia que el bastón blanco.



**Figura 1.2: Bastón blanco electrónico<sup>3</sup>**

### **c. Bastón vibratorio para invidentes**

Permite avisar al usuario la proximidad de obstáculos en su camino. Esto es posible gracias a un sensor supersónico, que puede detectar obstáculos hasta la altura de la cintura del usuario a una distancia desde los 3

---

<sup>3</sup> “Bastón blanco electrónico teletacto”. [En línea]. Recuperado el 08 de marzo del 2013, Disponible en Web: < <http://familia.bertho.free.fr/ace31/index.php?page=boitier>>

centímetros hasta los 2 metros, vibrando con más intensidad dependiendo de la cercanía del obstáculo.

#### **d. Gafas con sistema GPS**

Prototipos que emplean la visión artificial, el sonido y tecnología GPS para que las personas ciegas reciban información sobre su entorno y puedan ubicar los obstáculos con los que pueden chocar. Consiste en unas gafas que emiten un rayo láser con el que se pueden localizar objetos mediante sonidos a unos 5 metros de distancia. En la Figura 1.3 se ilustra la apariencia de las gafas.



**Figura 1.3: Gafas con sistema GPS<sup>4</sup>**

#### **e. Mi Ultrasonic**

El MINI es solo como un accesorio a la más tradicional de las ayudas como el bastón blanco y el perro guía. Usa el eco mediante ultrasonidos para detectar objetos. Vibra para indicar la distancia a los objetos. Es más rápido de la vibración cuando está cerca del objeto. También tiene una toma de auriculares que puede utilizarse para proporcionar información. En la Figura 1.4 se observa el accesorio.

---

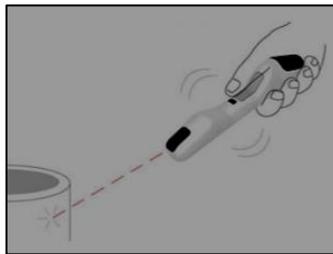
<sup>4</sup> “Lentes para ciegos”. [En línea]. Recuperado el 09 de marzo del 2013, Disponible en Web: <<http://bloguned1.blogspot.com/2009/07/caslip-lentes-para-ciegos.html>>



**Figura 1.4: Mi Ultrasonic<sup>5</sup>**

### **f. Vara para ciegos**

Hace cierta referencia al clásico bastón blanco, sólo que un poco de tecnología que permite detectar los obstáculos sin necesidad de golpearlos véase la Figura 1.5. El tubo tiene sensores incorporados que miden la distancia de los objetos y avisan por medio de vibraciones al portador de cercanía, para que los evite.



**Figura 1.5: Vara para ciegos<sup>6</sup>**

### **g. Perro guía<sup>7</sup>**

El adiestramiento de un perro guía es un proceso bastante complejo y útil para guiar a aquellas personas ciegas o con deficiencia visual grave (véase

---

<sup>5</sup> “Mi ultrasonic”. [En línea]. Recuperado el 26 de febrero del 2013, Disponible en Web: <<http://itzamna.bnct.ipn.mx/dspace/bitstream/123456789/11470/1/21.pdf>>

<sup>6</sup> “Una vara para ciegos”. [En línea]. Recuperado el 09 de marzo del 2013, Disponible en Web: <<http://www.impresionante.net/10-03-2008/general/una-vara-para-ciegos>>

<sup>7</sup> “Perro guía”. [En línea]. Recuperado el 08 de marzo del 2013, Disponible en Web: <<http://www.escuelaperrosguia.com/etapas/etapas.htm>>

Figura 1.6). El perro debe tener capacidad para percatarse de peligros eventuales para el dueño debido por ejemplo a barreras arquitectónicas.



**Figura 1.6: Perro guía<sup>8</sup>**

### **1.3.2 Robots móviles en interiores**

Los robots móviles, dependen de las tecnologías de sensado dado que la manera de actuar de un robot móvil puede ser simplificada si el sistema de detección es suficientemente sofisticado. Emplean detectores como las cámaras de video o sensores ultrasónicos, infrarrojos, radares, entre otros.

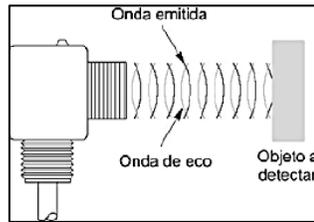
#### **a. Sensor Ultrasónico<sup>9</sup>**

Basan su funcionamiento en la emisión y recepción de una onda de acústica en el rango de 65 a 400 KHz, un valor demasiado alto para ser detectado por el oído humano. Tanto el emisor como el receptor son construidos con materiales piezoeléctricos. La piezoelectricidad es una propiedad que tienen algunos cristales de generar una diferencia de potencial entre sus extremos cuando son sometidos a una deformación mecánica o se deforman cuando se les aplica una diferencia de potencial entre sus extremos.

---

<sup>8</sup> "Perro guía". [En línea]. Recuperado el 09 de marzo del 2013, Disponible en Web: <[http://www.proyectosalohogar.com/Zoologia/Terapias\\_con\\_animales.htm](http://www.proyectosalohogar.com/Zoologia/Terapias_con_animales.htm) >

<sup>9</sup> ANDRIAGO, Miguel. "Sensores de posición, descripción, selección y uso texto". Universidad Nacional Experimental Politécnica Antonio José de Sucre. Barquisimeto, Noviembre 2003.

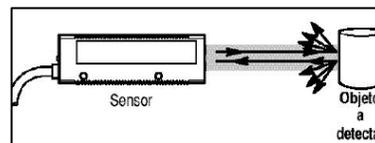


**Figura 1.7: Modo de operación del sensor Ultrasónico<sup>10</sup>**

En la Figura 1.7 se muestra un modo de operación del sensor ultrasónico, el emisor de este sensor genera un pulso ultrasónico, si hay un objeto presente, la onda choca y se regresa hacia el receptor y es examinada por una unidad evaluativa donde se filtran ruidos para asegurarse que el eco corresponde a la de la onda emitida. En los detectores ultrasónicos con salida analógica se cuantifica el tiempo que tarda en llegar el eco para determinar la distancia a la cual se encuentra el objeto del detector.

## **b. Sensor infrarrojo**

Los detectores infrarrojos basan su funcionamiento en la emisión de un haz de luz infrarroja que es interrumpido o reflejado por el objeto a detectar. Tiene muchas aplicaciones en el ámbito de la robótica. En la Figura 1.8 se muestra un sensor infrarrojo que refleja el haz por el objeto a detectar.



**Figura 1.8: Sensor Infrarrojo<sup>11</sup>**

<sup>10</sup> "Principio de operación ultrasónico". [En línea]. Recuperado el 09 de marzo del 2013, Disponible en Web: <<http://www.monografias.com/trabajos67/tecnologia-control/tecnologia-control3.shtml>>

<sup>11</sup> "Detección difusa". [En línea]. Recuperado el 09 de marzo del 2013, Disponible en Web: <<http://www.monografias.com/trabajos67/tecnologia-control/tecnologia-control3.shtml>>

### c. Sensor radar

Radar es un acrónimo de Radio Detection And Ranging (detección y medida de distancias por ondas de radio). Emite la señal hacia una escena, recibe la fracción de energía reflejada por la escena en su misma dirección y observa la intensidad (detección) y el retardo de tiempo (distancia) de las señales o eco de retorno.

En la Figura 1.9 se muestra el principio de funcionamiento del sensor radar. Las ondas se propagarán hasta su colisión con un objeto, propagándose en múltiples direcciones. Parte de estas ondas, retornarán al equipo radar, el cual las analizará y procesará.

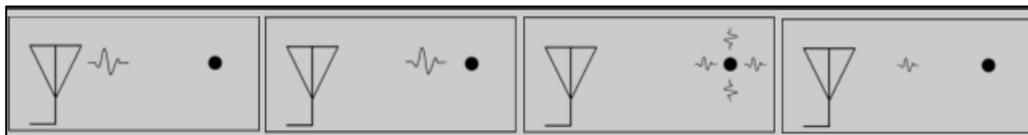


Figura 1.9: Sensor Infrarrojo<sup>12</sup>

## 1.4 MICROSOFT KINECT

Kinect es un dispositivo desarrollado por PrimeSense y distribuido por Microsoft para la videoconsola Xbox 360. Su apariencia se muestra en la Figura 1.10. Kinect es un dispositivo de control por movimiento creado originalmente para jugar sin necesidad de ningún mando o controlador para interactuar con el cuerpo en el videojuego mediante una interfaz natural de usuario que reconoce gestos, comandos de voz e imágenes.

<sup>12</sup> "Sensor con tecnología radar". [En línea]. Recuperado el 09 de marzo del 2013, Disponible en Web: <<http://www.sensorstecnics.net/en/productos/category/244/sensores-y-transmisores/sensores-ultrasonicos/sensor-de-distancia-con-tecnologia-radar>>



**Figura 1.10: Sensor Kinect para la consola Xbox360<sup>13</sup>**

El lanzamiento del sensor Kinect en Noviembre del 2010 supondría una revolución en el uso de este tipo de dispositivos capaces de capturar la profundidad. Su bajo coste y su aceptable precisión en los datos entregados hicieron que pronto éste fuese utilizado en multitud de aplicaciones.

Todos los píxeles que Kinect recibe como IR son convertidos en una escala de colores, haciendo que los cuerpos, dependiendo de la distancia, se capturen como rojos, verdes, azules hasta llegar a tonos grises, que representan a objetos muy lejanos<sup>14</sup>.

### **1.4.1 Partes fundamentales del Hardware**

Kinect es una barra de plástico negro de 30 cm de ancho conectada a un cable que se bifurca en dos, un cable USB y otro un cable eléctrico. Se pueden distinguir cuatro partes fundamentales dentro del hardware:

- Cámara RGB, cámara de video con una resolución de 640x480 y 1280x1024 píxeles a 30 fps.

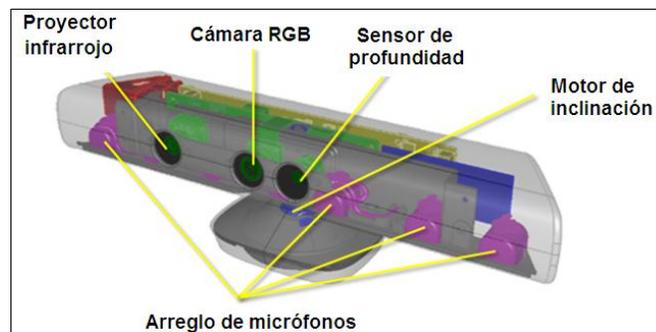
---

<sup>13</sup> "Kinect". [En línea]. Recuperado el 09 de marzo del 2013, Disponible en Web: <<http://www.respuestario.com/como/como-funciona-kinect-guia-de-uso-practica-paso-a-pasor>>

<sup>14</sup> "Conoce con detalles técnicos cómo funciona Kinect". [En línea]. Recuperado el 22 de septiembre del 2012, Disponible en Web: <<http://www.tierragamer.com/index.php/conoce-como-funciona-kinect/>>

- Sensores 3D de profundidad, combinación de un proyector de profundidad con un sensor de profundidad.
- Inclinación monitorizada, permite ajustar la cámara hacia arriba o abajo hasta 27°.
- Micrófono Multi-array, conjunto de cuatro micrófonos que se monta como un solo micrófono.

En la Figura 1.11 se puede distinguir las partes del Kinect.



**Figura 1.11: Hardware del sensor Kinect<sup>15</sup>**

Y aunque no visibles a simple vista, Kinect también posee:

- Memoria RAM de 512 Mb
- Acelerómetro, para estabilizar la imagen cuando se mueve.
- Ventilador, no está encendido continuamente para no interferir con los micrófonos.

Las especificaciones del sensor Kinect se resumen en la Tabla 1.1.

<sup>15</sup> "Sensor Kinect". [En línea]. Recuperado el 10 de marzo del 2013, Disponible en Web: <<http://msdn.microsoft.com/en-us/library/jj131033.aspx>>

**Tabla 1.1: Especificaciones del Kinect**

<b>Kinect</b>	<b>Especificación</b>
Ángulo de visión	43° vertical por 57° horizontal
Rango de inclinación vertical	±27°
Resolución Cámara RGB	640x480(VGA <sup>16</sup> ) y 1280X1024 Píxeles
Resolución Sensor de Profundidad	640x480(VGA), 320x240 (QVGA) y 80x60 píxeles
Velocidad de generación	30 imágenes por segundo (30fps)

Se debe inclinar el Kinect pocas veces como sea posible, para minimizar el desgaste en el sensor y para minimizar el tiempo de inclinación. El motor de inclinación no está diseñado para el movimiento constante o repetitivo, y los intentos de utilizarlo de esa manera pueden causar la degradación de la función motora.

Para reducir el desgaste, la aplicación debe cambiar el ángulo de elevación no más de una vez por segundo. Además, debe permitir por lo menos 20 segundos de descanso después de 15 cambios consecutivos. Si se exceden estos límites, el motor de inclinación puede experimentar un período de bloqueo y se traducirá en un código de error.

---

<sup>16</sup> El formato VGA (Colección de Gráficos de Video) fue diseñado por IBM en 1987 con un tamaño de imagen de 640x480. Debido a estándares comunes para PCs y monitores industriales, VGA ha sido ampliamente usado en los dispositivos con imágenes digitales.

En la imagen de la cámara RGB el valor por defecto es InfraredResolution640x480Fps30, que son datos de 16 bits, cuya resolución es de 640 x 480 y la velocidad de fotogramas es de 30 fotogramas por segundo.

### **1.4.2 Técnica para capturar la profundidad**

La cámara permite generar una imagen tridimensional de lo que tiene delante y además reconocer partes del cuerpo humano. Para ello utiliza un sónar de luz infrarroja.

El proyector láser infrarrojo proyecta sobre la escena un patrón de 50000 puntos invisibles al ojo humano, como se muestra en la Figura 1.12 (la fotografía fue obtenida con una cámara de visión nocturna), luego de rebotar en los objetos de la escena el patrón de puntos es captado por la cámara infrarroja que se encuentra a 7,5cm de separación del proyector.

Entonces el circuito integrado, analiza la disparidad provocada por los objetos de la escena entre el patrón de puntos proyectados y el patrón de puntos captados<sup>17</sup>.

---

<sup>17</sup> MESÍAS, Alejandro; LÓPEZ, Christian. "Diseño e implementación de un prototipo a escala de un robot móvil acompañante". Director: Nelson Sotomayor MSc. Escuela Politécnica Nacional, Facultad de Eléctrica y Electrónica. Quito, Junio 2012.



**Figura 1.12: Proyección del patrón infrarrojo<sup>18</sup>**

Cuando la cámara recibe la luz infrarroja generada por el cañón de infrarrojos se combina con el sensor monocromático CMOS se genera una malla de puntos mediante los cuales se genera una imagen como la Figura 1.13:



**Figura 1.13: Imagen generada por la cámara del Kinect<sup>19</sup>**

El entorno que la cámara enfoca así como la profundidad de los objetos es de la que se parte para un procesado más completo. Acto seguido el chip de procesado de imagen (PrimeSense PS1080) descompone la imagen en los parámetros necesarios para su tratamiento. El proceso se resume en la Figura 1.14.

---

<sup>18</sup> “Proyector láser infrarrojo”. [En línea]. Recuperado el 1 de octubre del 2012, Disponible en Web: <[http://biqfr.blogspot.com/2010\\_11\\_02\\_archive.html](http://biqfr.blogspot.com/2010_11_02_archive.html) >

<sup>19</sup> “Programar sensor Kinect”. [En línea]. Recuperado el 1 de octubre del 2012, Disponible en Web: <[http://biqfr.blogspot.com/2010\\_11\\_02\\_archive.html](http://biqfr.blogspot.com/2010_11_02_archive.html)>



**Figura 1.14: Diagrama de la tecnología Kinect patentada por PrimeSense<sup>20</sup>**

### 1.4.3 Información de profundidad

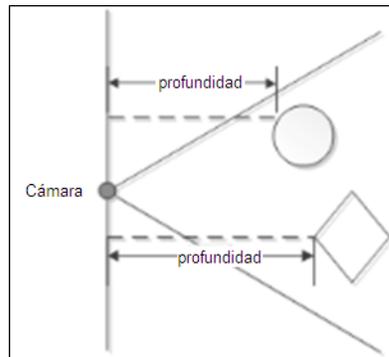
La información de profundidad se devuelve en un mapa de píxeles con una frecuencia máxima de 30 imágenes por segundo. Cada píxel está representado por dos bytes (16 bits), cuyo valor representa teóricamente la distancia del objeto al sensor.

Si el valor de un píxel del mapa es cero, significa que el sensor no pudo estimar la profundidad para esa región debido a las limitaciones del sensor.

Cada píxel contiene la distancia cartesiana, en milímetros, desde el plano de la cámara al objeto más cercano en esa coordenada particular (x,y) como se

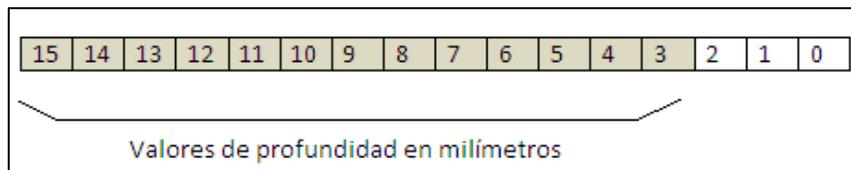
<sup>20</sup> "PrimeSense", [En línea]. Recuperado el 1 de octubre del 2012. Disponible en Web: <<http://www.primesense.com/en/technology/115-the-primesense-3d-sensing-solution>>

muestra en la Figura 1.15. Las coordenadas (x,y) de la imagen de profundidad no representan unidades físicas en la habitación, sino que representan la ubicación de un pixel en la imagen de profundidad.



**Figura 1.15: Forma de medir la profundidad del Kinect<sup>21</sup>**

Los datos de profundidad real son de un tamaño de 12 bits. Los 2 bytes de datos de profundidad incluyen un índice de jugadores en los tres bits menos significativos. Los datos de profundidad por lo tanto, están desplazados a la izquierda por 3 bits. Se debe tener esto en cuenta esto a la hora de leer los bits de profundidad<sup>22</sup>. En la Figura 1.16 se muestra la localización de los datos de profundidad.



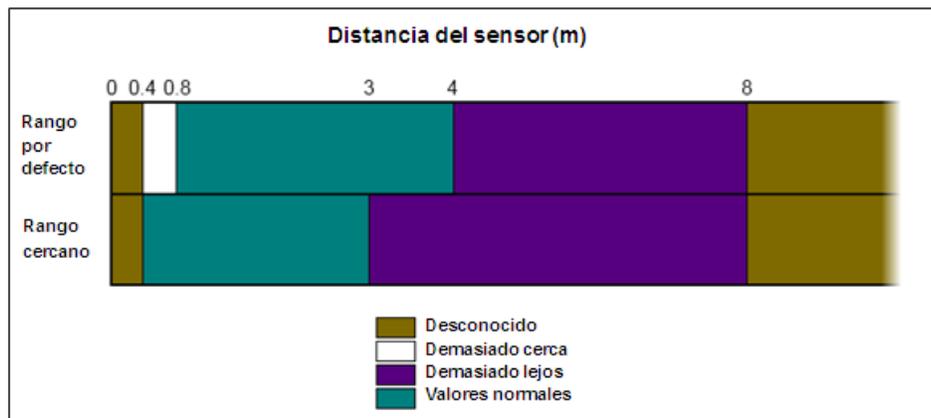
**Figura 1.16: Ubicación de los bits de profundidad**

<sup>21</sup> "Depth Camera". [En línea]. Recuperado el 1 de marzo del 2013, Disponible en Web: <<http://msdn.microsoft.com/en-us/library/hh438997.aspx>>

<sup>22</sup> "Microsoft Kinect SDK". [En línea]. Recuperado el 08 de marzo del 2013. Disponible en Web: <[http://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth\\_Ranges](http://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth_Ranges)>

Hay tres valores que indican que la profundidad no podría determinarse de forma fiable en una ubicación. El valor "demasiado cerca" significa que un objeto fue detectado, pero está muy cerca del sensor para proporcionar una medición de distancia fiable. El valor "demasiado lejos" significa que un objeto fue detectado, pero está muy lejos poder medirse de forma fiable. El valor "desconocido" significa que ningún objeto fue detectado.

El sensor de profundidad tiene dos rangos de profundidad: el rango por defecto y el rango cercano. La Figura 1.17 ilustra los rangos de profundidad del sensor en metros. El rango predeterminado está disponible tanto en el sensor Kinect para Windows y para Xbox 360, el rango cercano está disponible sólo en el sensor Kinect para Windows.



**Figura 1.17: Rango de distancias admitidas por el sensor Kinect.<sup>23</sup>**

<sup>23</sup> "Depth Space Range". [En línea]. Recuperado el 1 de marzo del 2013, Disponible en Web: <[http://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth\\_Ranges](http://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth_Ranges)>

#### 1.4.4 Limitaciones<sup>24</sup>

El sensor Kinect tiene varias limitaciones que hacen que la profundidad de ciertas regiones de la escena no se pueda estimar o si se estima, la fiabilidad de los datos no es aceptable. Estas limitaciones vienen condicionadas tanto por factores internos, debidos a la arquitectura del dispositivo; como externos, debidos a la naturaleza de la escena.

Los puntos de luz no cubren de forma continua la superficie de los objetos (Figura 1.12), lo que conlleva a que algunos píxeles de la imagen de profundidad tienen que ser interpolados. Esto implica que el valor de profundidad de un píxel determinado tiene asociado un margen de error. Este margen es mayor cuanto más alejado está el objeto, puesto que, para una misma superficie, los puntos de luz están más separados.

A mayores distancias, los valores de profundidad devueltos para objetos cercanos entre sí tienden a ser muy similares. Sin embargo, si el objeto está a demasiada distancia del sensor, no se calcula ninguna distancia para ese punto (Figura 1.18).



**Figura 1.18: Efecto de distancias grandes en el Kinect<sup>25</sup>**

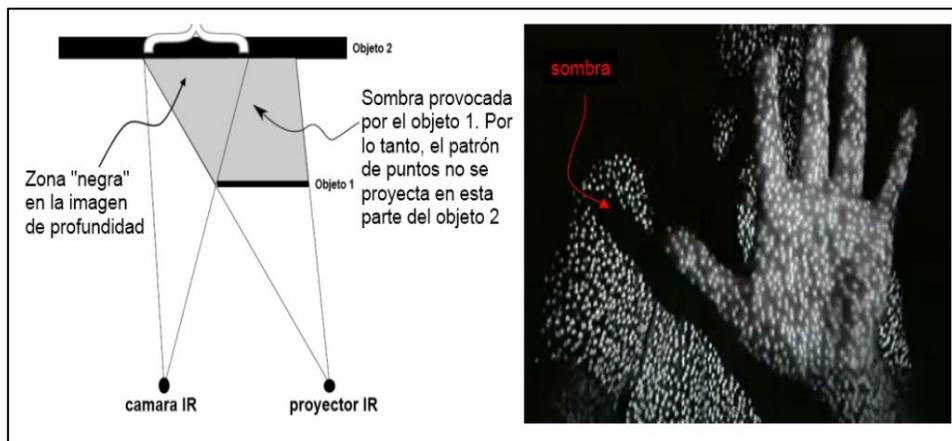
---

<sup>24</sup> CÓRDOVA, Fabricio. "Detección de robo/abandono de objetos en interiores utilizando cámaras de profundidad". Universidad autónoma de Madrid, Diciembre 2012.

<sup>25</sup> "Limitaciones Kinect". [En línea]. Recuperado el 10 de marzo del 2013, Disponible en Web: <<http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20121212FabricioACordovaLucero.pdf>>

Esto ocurre así, porque la potencia de luz del haz de infrarrojos se atenúa en el trayecto recorrido, haciendo que sea imperceptible para el sensor de infrarrojos.

La luz emitida por el proyector de infrarrojos, al impactar sobre un objeto, genera una sombra de éste a mayor distancia, como se puede apreciar en la Figura 1.19. El resultado es que no se puede determinar la profundidad en las zonas afectadas por dichas sombras. Esto se manifiesta como píxeles de valor cero (“zonas negras”) en la imagen de profundidad.



**Figura 1.19: Efecto de las sombras en el Kinect<sup>26</sup>**

Como ocurre con todo sistema óptico, la otra gran limitación viene determinada por las características de los objetos (factores externos). En general, atendiendo a la forma y propiedades de las superficies, se puede hacer la siguiente clasificación de objetos:

- Objetos translúcidos: los puntos de luz que impactan sobre éstos sufren una dispersión, haciendo indistinguible la deformación del patrón para el sensor.

<sup>26</sup> “Detección de objetos con Kinect”. [En línea]. Recuperado el 10 de marzo del 2013, Disponible en Web: <<http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20121212FabricioACordovaLucero.pdf>>

- **Objetos reflectantes:** los puntos de luz tienden a impactar sobre el objeto reflejado.

Otra limitación radica cuando hay un cambio muy fuerte de luminosidad (luz solar) que impide que la luz infrarroja pueda ser detectada por el Sensor IR.

Asimismo, la inclinación de la superficie de los objetos respecto al proyector del haz de luz limita la detección de la profundidad. Si el rayo de luz es casi paralelo a la superficie no podrá incidir sobre la misma, haciendo imposible la estimación de la profundidad. En la Figura 1.20 se puede ver este efecto en la mesa de la esquina inferior derecha y en la superficie del suelo.



**Figura 1.20: Efecto de superficie reflectante y paralela a los rayos de luz<sup>27</sup>**

## 1.5 SOFTWARE PARA CONTROLAR EL SENSOR KINECT

Existen una gran variedad de librerías disponibles, cada uno con sus drivers propios para controlar el sensor Kinect. Esto significa que no es posible combinar dos librerías diferentes (de propietarios diferentes) para crear una aplicación concreta.

<sup>27</sup> "Kinect". [En línea]. Recuperado el 10 de marzo del 2013, Disponible en Web: <<http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20121212FabricioACordovaLucero.pdf>>

## 1.5.1 Microsoft Kinect SDK

Este es el kit de desarrollo oficial lanzado por la corporación de Microsoft. La documentación de Microsoft afirma que el rango de distancias válido para el sensor de profundidad es de 4 a 11.5 pies (1.2 a 3.5 metros). Los valores de profundidad para cada pixel son devueltos en milímetros en dos bytes (16 bits), pero sólo los 12 bits más significativos contienen la información “real” de profundidad. Microsoft no dice nada acerca de la precisión de estos valores (mayores de 3.5 metros); pero, como es lógico, las degradaciones en la información de profundidad serán mayores.

Algunas de las características son<sup>28</sup>:

- Soporte para audio.
- Soporta manejo del motor de inclinación.
- Tracking de todo el cuerpo.
- No necesita calibración previa.
- Funciona mejor con los Joints ocultos.
- Puede manejar varios sensores.
- Instalación sencilla.
- Incluye eventos que saltan cuanto un nuevo frame o trama está disponible.
- Licencia no comercial.
- No da soporte a otros sensores similares a Kinect.
- Sólo soporta Windows 7.

---

<sup>28</sup> IRALDE L. Iñaki; PINA C. Alfredo. “Desarrollo de aplicaciones con Microsoft Kinect”. Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación. Pamplona, Abril 2012.

## 1.5.2 OpenNI SDK

Este SDK pertenece a PrimeSense, compañía líder en “interfaces naturales” y encargada de la fabricación del sensor de profundidad para el dispositivo Kinect. OpenNI es la organización que se encarga del desarrollo del software para todos los dispositivos fabricados por PrimeSense. Al igual que MSDK, el valor de profundidad se devuelve en dos Bytes y en milímetros<sup>29</sup>.

## 1.5.3 Open Kinect

Antes de la aparición de los entornos de desarrollo oficiales (MSDK y OpenNI), desarrolladores de todo el mundo hicieron grandes esfuerzos de ingeniería inversa sobre la Kinect para poder interpretar y manipular los datos entregados por ésta. Una de las comunidades de desarrolladores que se ha consolidado fuertemente es precisamente OpenKinect, cuya labor está centrada en el desarrollo de libfreenect, el paquete software para controlar el sensor Kinect.

## 1.5.4 Microsoft Kinect SDK vs. Open Source

Entre las opciones de software de desarrollo de Kinect existen dos posibles librerías: librerías libres o librerías oficiales. Dentro de las primeras, la más conocida y la más utilizada es la llamada OpenNI que guarda compatibilidad con las librerías de OpenKinect<sup>30</sup>. En la Tabla 1.2 se presentan las ventajas del SDK de Microsoft y OpenNI y en la Tabla 1.3 las desventajas.

---

<sup>29</sup> “Kinect primeros pasos”. [En línea]. Recuperado el 11 de marzo del 2013, Disponible en Web: <[http://alumnos.elo.utfsm.cl/~flopezp/Kinect\\_v1.pdf](http://alumnos.elo.utfsm.cl/~flopezp/Kinect_v1.pdf)>

<sup>30</sup> “Microsoft SDK vs OpenNI”, [En línea]. Recuperado el 09 de marzo del 2013. Disponible en Web: <<http://www.argencon.org.ar/sites/default/files/123.pdf>>

**Tabla 1.2: Ventajas del SDK de Microsoft y del OpenNI.**

SDK de Microsoft	SDK de OpenNI
Soporte para audio	Licencia incluye uso comercial.
Soporte para el motor de inclinación	Reconocimiento de gestos con las manos.
Seguimiento completo	Trackeo de cuerpo entero.
No necesita paso de calibración	Soporta Windows, Linux y MacOSX.
Soporta múltiples sensores	Calcula la rotación de las articulaciones
Instalación simple	Tiene eventos cuando un usuario ingresa o abandona el cuadro.

**Tabla 1.3: Desventajas del SDK de Microsoft y del OpenNI.**

SDK de Microsoft	SDK de OpenNI
Licencia para uso no comercial	Sin soporte para audio
Trackeo cuerpo entero (no exclusivo de manos)	Sin soporte para el motor de inclinación
Calcula posición de las articulaciones pero no rotación	Necesita de una fase de calibración
Solo para Windows 7	Intrincada instalación

## 1.6 PROCESAMIENTO DIGITAL DE IMÁGENES

### 1.6.1 Vision Artificial

La visión artificial conocida también como Visión por Computador consiste en simular el sentido de la visión de un ser humano por medio de programar un computador para que obtenga toda la información de una imagen digital para realizar una tarea determinada. La imagen de entrada es procesada para extraer los atributos, obteniendo como salida una descripción de la imagen analizada (Figura 1.21). Esto se consigue por técnicas como la de reconocimiento de patrones, procesamiento de imágenes, geometría de proyección y otras acciones<sup>31</sup>. La vista es uno de los sentidos más complejos y especializados del cuerpo humano, por medio de la visión, el ser humano es capaz de captar la forma, el volumen, los colores, la luminosidad, el tamaño y demás cualidades de los objetos que lo rodean, lo cual corresponde a tres cuartas partes del total de las percepciones de la persona.

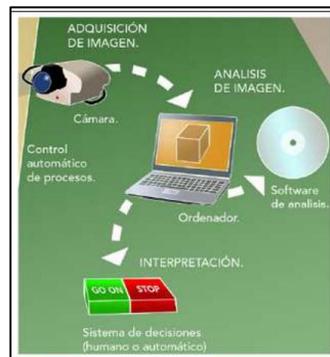


Figura 1.21: Sistema de visión artificial<sup>32</sup>

<sup>31</sup> "Visión artificial", [En línea]. Recuperado el 25 de febrero 2013. Disponible en Web: <<http://bibdigital.epn.edu.ec/bitstream/15000/4119/1/CD-3466.pdf> >

<sup>32</sup> "Sistema de Visión artificial". [En línea]. Recuperado el 25 de febrero del 2013, Disponible en Web: <<http://bibdigital.epn.edu.ec/bitstream/15000/4119/1/CD-3466.pdf>>

El observador humano usa un amplio rango de señales, obtenidas a partir del color, la perspectiva, el sombreado, y una amplia librería de experiencias individuales. La percepción visual depende únicamente de la capacidad humana de realizar juicios. Sin embargo, un sistema de visión artificial no tiene una experiencia de base de la cual tomar decisiones mediante una comparación; todo debe estar específicamente definido.

### **1.6.2 Representación digital de una imagen**

Para visualizar una señal continua o imagen analógica en un computador es necesario transformarla a un formato digital, de esta manera la imagen adquirida por una cámara es representada por una matriz cuadrada de pixeles. Una imagen puede ser definida matemáticamente como una función bidimensional,  $f(x,y)$  donde  $x$  y  $y$  son coordenadas espaciales, y la amplitud de  $f$  en cualquier par de coordenadas  $(x,y)$  es la intensidad o nivel de gris de la imagen en esa coordenada.

Cuando los valores de la amplitud de  $f$  son cantidades finitas, se dice que la imagen es Digital. Una imagen digital está compuesta de un número finito de elementos, cada uno con un lugar y valor específicos. Estos elementos se denominan como picture elements, pels o pixels. El pixel es el elemento más ampliamente utilizado.<sup>33</sup>

### **1.6.3 Etapas del Procesamiento Digital de Imágenes**

#### **a. Adquisición de la imagen**

Cuando se observa una imagen, lo que se está observando es un conjunto de puntos, coloreados de acuerdo a una regla de representación. Para

---

<sup>33</sup> HIDALGO, Víctor. "Implementación de un sistema de autenticación biomédica basado en huellas digitales". Director: Hugo Moreno. Escuela Superior Politécnica de Chimborazo. Riobamba 2010.

almacenar una imagen se debe definir una matriz cuyas entradas sean todos los posibles valores observables. Existen varios tipos de formato de imagen digital, los cuales difieren por la forma en que almacenan la información, la tasa de compresión, etc.

### **b. Procesamiento previo**

Cuando se adquiere una imagen mediante cualquier sistema de captura, por lo general ésta no es directamente utilizable por el sistema de visión. La aparición de variaciones en intensidad debidas al ruido, por deficiencias en la iluminación, o la obtención de imágenes de bajo contraste, hace necesario un pre-procesamiento de la imagen para corregir estos problemas, además de aplicar aquellas transformaciones a la imagen que acentúen las características que se deseen extraer de las mismas.

Para realizar el mejoramiento de una imagen se puede utilizar el dominio espacial en el cual se trabaja en base al plano de la imagen formado por píxeles, de tal manera que se puedan realizar operaciones propias de visión artificial como la etapa de segmentación, extracción de características e interpretación automática de las imágenes.

### **c. Extracción de características relevantes**

La mayoría de las veces, los datos originales en una imagen contienen información irrelevante para una aplicación dada. Los datos se deben simplificar controlando la cantidad y naturaleza de la información que es preservada.

## **d. Interpretación**

Este es el proceso que se realiza una vez que se obtiene la imagen segmentada en regiones de interés. A cada elemento de la región se le asocia un conjunto de valores numéricos o atributos llamado vector de características. Dichos valores constituyen la entrada al sistema de interpretación automática de imágenes. Aquí también se emplea propiedades relacionadas con el total de píxeles que forman el objeto como: métricas, topológicas, texturales y de fronteras.

### **1.6.4 Tipos de procesamiento**

#### **a. Segmentación**

La segmentación es el proceso que divide una imagen en regiones u objetos cuyos píxeles poseen atributos similares, y constituye uno de los procesos más importantes de un sistema automatizado de visión, ya que permite extraer los objetos de la imagen para su posterior descripción y reconocimiento. El nivel de la segmentación depende del problema a ser resuelto, de la precisión de este proceso depende el éxito o fracaso de los procedimientos de análisis computarizados.

Los algoritmos de segmentación se basan en dos propiedades de los valores de intensidad de los píxeles: la discontinuidad y la similitud. En base a la discontinuidad se pueden detectar los bordes. Con la propiedad de similitud se puede segmentar la imagen en regiones similares en base a un criterio determinado.

Las distintas técnicas de segmentación pueden encuadrarse en tres grupos fundamentales: técnicas de umbralización, técnicas basadas en regiones y técnicas basadas en la detección de bordes.

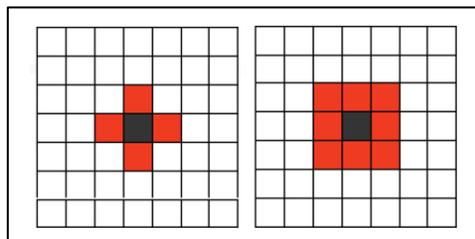
## a.1. Técnicas de Umbralización

El método de segmentación por umbralización tiene en cuenta el valor de intensidad de los píxeles, para decidir si estos forman parte de un objeto de interés.

## a.2. Técnicas basadas en regiones

Las técnicas basadas en regiones tienen en cuenta un conjunto de píxeles de la imagen, en los que se analizan características como la posición en el espacio de intensidades, relaciones topológicas (conectividad) y características de las fronteras entre dos conjuntos. La conectividad de dos píxeles implica que están conectados si existe un camino de uno al otro a través de píxeles vecinos con la misma etiqueta.

La conectividad de un píxel con otro puede ser de 4-conectividad si existe relación entre dos píxeles colindantes horizontalmente o verticalmente, o bien 8-conectividad si los píxeles se tocan en forma diagonal también. Dos píxeles que no son colindantes también pueden tener 4 u 8-conectividad si existe un camino de uno al otro a través de píxeles conectados. En la Figura 1.22 se muestra los tipos de vecindad.



**Figura 1.22: Vecindad a 4 (izquierda); vecindad a 8 (derecha)**<sup>34</sup>

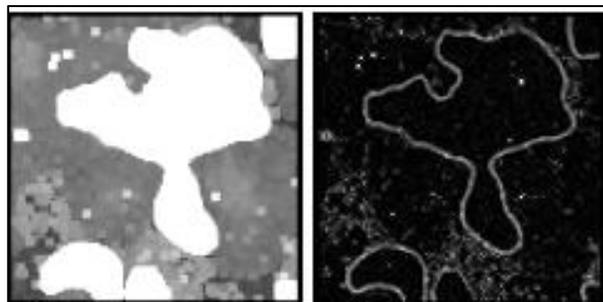
<sup>34</sup> “Técnicas de umbralización”. [En línea]. Recuperado el 25 de febrero 2013. Disponible en Web: <<http://dSPACE.epn.edu.ec/bitstream/15000/9218/1/T11089.pdf>>

### a.3. Detección de Bordes<sup>35</sup>

Es en una técnica que consiste en localizar los lugares donde se produce un cambio significativo de los niveles de intensidad de los píxeles adyacentes.

La detección de bordes se basa en la aplicación de la derivada en un entorno de vecindad. Al aplicar la derivada a la imagen original se puede tener bordes falsos debido al ruido existente en la etapa de captura de la imagen, por lo que es adecuado aplicar una técnica de suavizado de ruido previamente.

Entre los filtros que se utilizan para mejorar la imagen se encuentra el filtro Sigma que pueden preservar satisfactoriamente los detalles finos, el filtro Sigma es un filtro de paso alto, en él se esbozan los contornos y detalles mediante el establecimiento de píxeles con el valor medio encontrado en su vecindario, si su desviación de este valor no es significativo. En la Figura 1.23 se muestra el efecto de aplicar el filtro a una imagen.



**Figura 1.23: Imagen original (izquierda), imagen con filtro sigma (derecha)<sup>36</sup>**

---

<sup>35</sup> ACURIO, Eliana; ENCARNACIÓN Diana. "Diseño e construcción de un módulo didáctico de visión artificial orientada al Control de Calidad de llenado de botellas de vidrio con diferente tipo de líquido". Director Ing. Ana Rodas. Escuela politécnica Nacional. Quito Febrero 2011.

<sup>36</sup> NiVisionConcepts.chm

## b. Binarización

Con éste método es posible convertir una imagen de varios niveles de gris a una nueva imagen con solo dos (blanco y negro), de tal forma que los objetos queden separados del fondo.

Si se considera que una imagen consiste en un conjunto de coordenadas discretas, el conjunto corresponde todos aquellos puntos o pixeles que pertenecen a la imagen (Figura 1.24).

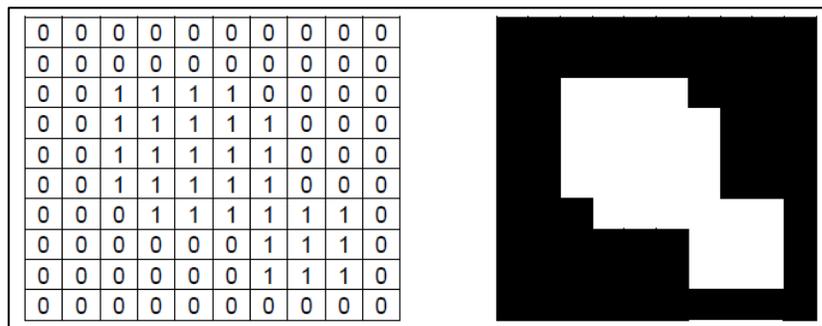


Figura 1.24: Representación de imágenes binarias<sup>37</sup>

## c. Segmentación basada en morfología matemática

La morfología matemática es una herramienta para extraer componentes de una imagen que sean útiles en la representación y descripción de la forma de una región, tales como contornos, esqueletos y cerco convexo. Los operadores morfológicos cambian la forma de las partículas que procesan en base al número de sus vecinos y sus valores.

---

<sup>37</sup> "Representación de una imagen". [En línea]. Recuperado el 26 de febrero del 2013, Disponible en Web: <<http://dSPACE.epn.edu.ec/bitstream/15000/9218/1/T11089.pdf>>

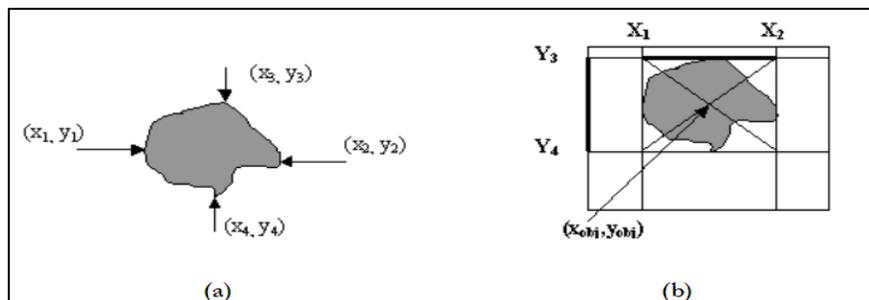
#### d. Relleno de región

En ciertas aplicaciones, es conveniente poder llenar un objeto que está definido por una frontera. Para ello, se puede emplear una operación llamada relleno de región cuyos algoritmos están basados en dilataciones, complementaciones e intersecciones.

#### e. Determinación de la posición<sup>38</sup>.

Para seguir un objeto es necesaria una medida de observación que corresponderá con la posición del centro de masas del objeto. La determinación de la posición basada en los extremos de un objeto se fundamenta en la obtención de los vértices, izquierdo  $(x_1, y_1)$ , derecho  $(x_2, y_2)$ , superior  $(x_3, y_3)$  e inferior  $(x_4, y_4)$  del objeto de la imagen (Figura 1.25 a).

La localización del objeto vendrá determinada por un punto único característico del mismo  $(x_{obj}, y_{obj})$ , que obtendremos como resultado de la intersección de las diagonales de caja contenedora (Figura 1.25 b).



**Figura 1.25: a) Obtención de los puntos extremos de un objeto; b) Obtención del centroide del objeto<sup>39</sup>**

<sup>38</sup> RODRIGUEZ, Patricio. "Aplicación del filtro de Kalman al seguimiento de objetos en secuencias de imágenes". Tutor, Antonio Sanz, Juan José Pantrigo. Universidad Rey Juan Carlos. 2003.

<sup>39</sup> "Localización de objeto". [En línea]. Recuperado el 26 de febrero del 2013, Disponible en Web: <<http://www.escet.urjc.es/~jjpantrigo/PFCs/MemoriaKalmanJun03.pdf>>

## f. Etiquetado de una imagen

Es la discriminación de regiones dentro de la imagen, cuya región corresponde a una agrupación de píxeles con características similares. Se consideran ciertas características para el etiquetado de las regiones como el tipo de vecindad que se encuentra en la región.

## 1.7 SOFTWARE LABVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) creado por National Instruments en 1976, es una plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual gráfico. El logotipo del programa se muestra en la Figura 1.26. Debido a su potente lenguaje un lema tradicional de Labview es "La potencia está en el Software".



**Figura 1.26: Logotipo de Labview<sup>40</sup>**

Permite el desarrollo de programas informáticos complejos facilitando al mismo tiempo la programación y en consecuencia disminuir los plazos de desarrollo. Gracias a sus librerías de funciones dedicadas a la adquisición de datos, la instrumentación, al análisis matemático de las medidas y la visualización, LabVIEW se dedica especialmente a los sistemas de prueba y medida. Desde realizar medidas simples de temperatura hasta controlar el

---

<sup>40</sup> "Logo Labview". [En línea]. Recuperado el 26 de febrero del 2013, Disponible en Web: <<http://www.cetoni.de/development/software.html>>

acelerador de partículas más grande del mundo, los ingenieros y científicos utilizan la plataforma de diseño gráfico de sistemas NI LabVIEW para resolver una amplia variedad de retos en aplicaciones.

LabVIEW consigue combinarse con todo tipo de software y hardware, tanto del propio fabricante (tarjetas de adquisición de datos, Visión, instrumentos) y otro Hardware como de diferentes fabricantes.

## **1.8 MICROSOFT SPEECH SDK**

Microsoft Speech SDK es un kit de desarrollo de software para la construcción de programas que tratan datos de voz y aplicaciones para Microsoft Windows. Incluye una colección de componentes de voz para gestionar directamente el audio, compilador de gramática, recursos, asistente de reconocimiento de voz y asistente texto a voz TTS (text to speech) para control de bajo nivel<sup>41</sup>.

El Microsoft Speech SDK 5.1 añade soporte para la automatización de las funciones de la versión anterior del Speech SDK. Con esto se lo puede utilizar para desarrollar aplicaciones de voz con diferentes lenguajes de automatización.

### **1.8.1 Narrador de Windows**

Desde la versión de Windows XP, el sistema operativo incluye una función conocida como Microsoft Text-to-Speech (texto a voz) que permite que la computadora lea el texto en voz alta con una voz hablada sintetizada. En Microsoft Windows 7, Microsoft Anna es el nombre de la voz por defecto, la cual está en inglés.

---

<sup>41</sup> "Microsoft Speech", [En línea]. Recuperado el 01 de marzo del 2013. Disponible en Web: <<http://msdn.microsoft.com/en-us/library/ms862156.aspx>>

## CAPÍTULO 2

### DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO

#### 2.1 INSTALACIÓN DEL SOFTWARE

Se requiere instalar:

- LabVIEW 2011 o superior con NI Vision Development Module.
- Kinect SDK 1.5
- Microsoft. NET Framework 4.0
- Microsoft Speech SDK.
- Voz adicional en español para Text to Speech.

##### 2.1.1 LabVIEW 2011 o superior con NI Vision Development Module.

LabView es una herramienta de programación gráfica. Los programas creados mediante LabView son los “instrumentos virtuales” o VI (Virtual Instrument) que tienen un panel frontal y un diagrama de bloques. Labview presenta herramientas de procesamiento de imágenes que facilitan la creación de aplicaciones inteligentes de visión y reconocimiento. El Toolkit de visión artificial de Labview es el IMAQ Vision, contiene herramientas para seleccionar la región de interés y extraer la información de cada píxel.

##### 2.1.2 Kinect SDK 1.5<sup>42</sup>

Este Kit simplifica la conexión entre el sensor Kinect y una máquina con Windows 7. El Kit de desarrollo de software (SDK de sus siglas en inglés

---

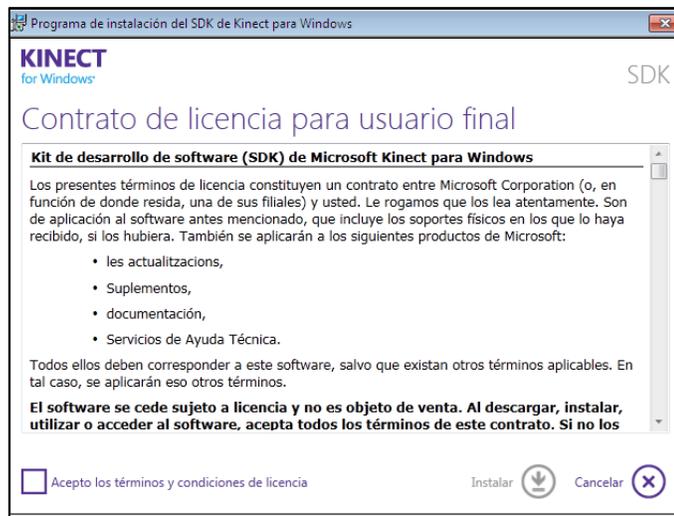
<sup>42</sup> “Microsoft Kinect SDK”. [En línea]. Recuperado el 20 de enero del 2013, Disponible en Web: <<http://msdn.microsoft.com/en-us/library/hh855354.aspx>>

Software Development Kit) provee de herramientas para desarrollar aplicaciones sobre Microsoft Windows. Este Kit requiere:

- Ordenador con un dual-core, procesador de 2,66 GHz o más rápido.
- Tarjeta gráfica compatible con Windows 7.
- 2-GB de RAM (4 GB RAM recomendado)

Para instalar el SDK:

- El sensor de Kinect no debe estar conectado al puerto USB del ordenador.
- El SDK se localiza en: [go.microsoft.com/fwlink/?LinkID=247938](http://go.microsoft.com/fwlink/?LinkID=247938). Se debe ejecutar el instalador y seguir los pasos del Asistente (Figura 2.1).

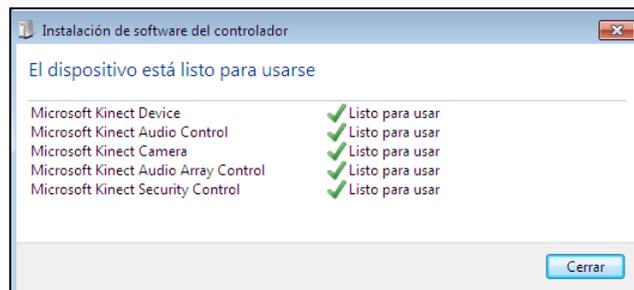


**Figura 2.1: Ventana para la instalación del Kinect SDK**

Una vez que se ha completado la instalación correctamente, se debe asegurar de que el sensor Kinect está conectado a una fuente de alimentación externa y que el sensor esté conectado al puerto USB del PC.

Los controladores se cargarán automáticamente (Figura 2.2). El sensor Kinect ya debería funcionar correctamente. Si el controlador se ha instalado correctamente:

- El LED verde en el sensor Kinect debe estar encendida.
- El sensor Kinect debe aparecer en el Administrador de dispositivos.



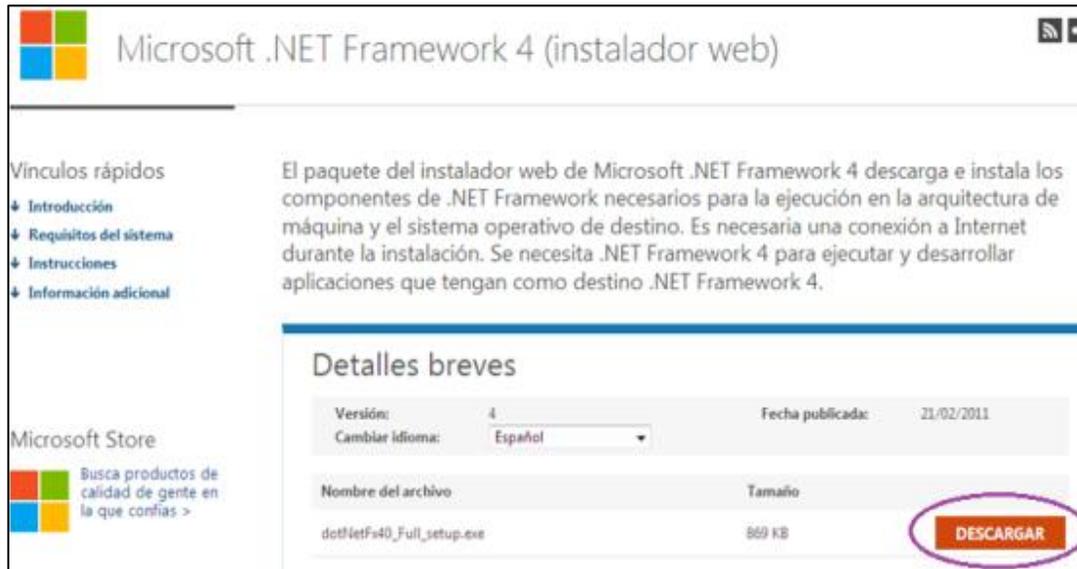
**Figura 2.2: Controladores del Kinect cargados automáticamente**

### 2.1.3 Microsoft. NET Framework 4.0

El Microsoft .NET Framework, es un componente de software que puede ser o es incluido en los sistemas operativos Microsoft Windows. El Sitio para realizar la descarga (Figura 2.3) desde la página de Microsoft es: <http://www.microsoft.com/es-es/download/details.aspx?id=17718>.

Provee soluciones pre-codificadas para requerimientos comunes de los programas y gestiona la ejecución de programas escritos específicamente para este Framework.

.NET Framework es el modelo de programación completo y coherente de Microsoft para compilar aplicaciones que ofrezcan una experiencia visual del usuario, comunicación perfecta y segura, y la capacidad de modelar una amplia gama de procesos empresariales.

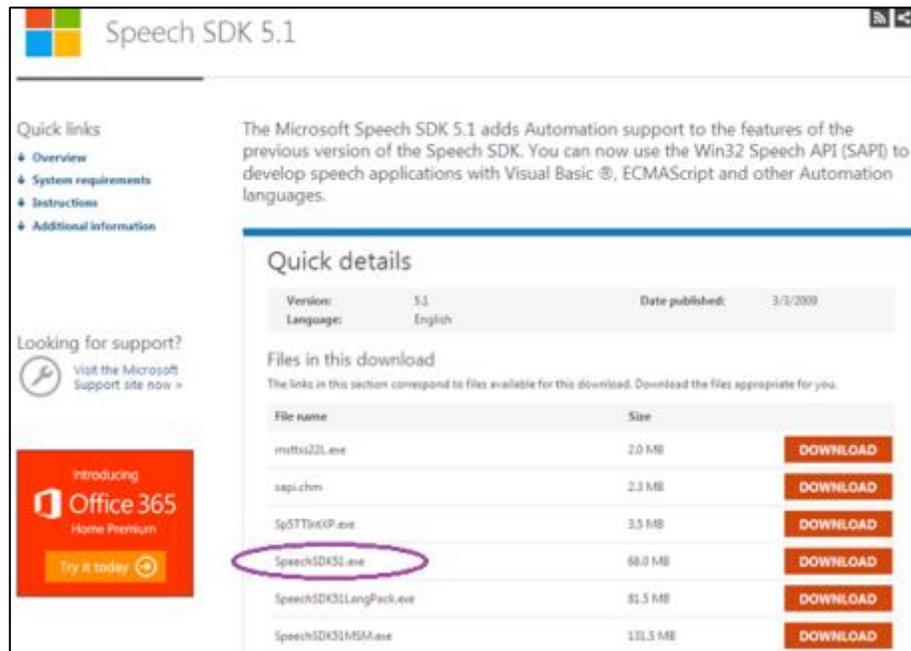


**Figura 2.3: Sitio de descarga del Microsoft .NET 4<sup>43</sup>**

## 2.1.4 Microsoft Speech SDK

El kit permite desarrollar aplicaciones de habla con diferentes lenguajes de automatización, se encuentra disponible en: <http://www.microsoft.com/en-us/download/details.aspx?id=10121>. Se debe descargar el "SpeechSDK51.exe", luego se debe ejecutar el archivo setup.exe (Figura 2.4). Se debe seguir adelante en los pasos del asistente de instalación.

<sup>43</sup> "Microsoft .NET Framework 4.0". [En línea]. Recuperado el 20 de febrero del 2013, Disponible en Web: < <http://www.microsoft.com/es-es/download/details.aspx?id=17718>>



**Figura 2.4: Sitio de descarga del Microsoft Speech SDK<sup>44</sup>**

### 2.1.5 Voz adicional en español para Text to Speech.

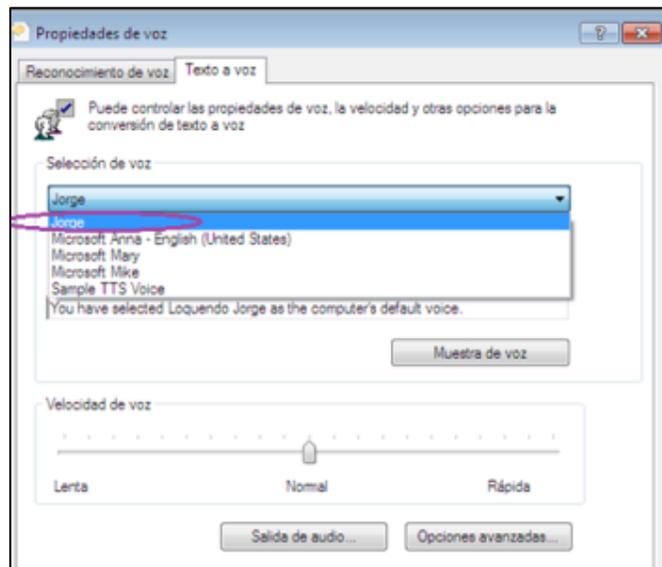
Se instalará una voz adicional, la voz de Jorge, que se encuentra en español.

Para la instalación se debe:

- Descargar e instalar la voz en español del sitio: <http://www.mediafire.com/?u6sehj62xo2wl8x>.
- Ejecutar el archivo Jorge.exe y colocar el archivo del crack LoqTTS6.dll en la ubicación de instalación.
- Hacer clic en Inicio, Panel de Control, Accesibilidad, Reconocimiento de voz, Texto a voz.

<sup>44</sup> "Microsoft Speech SDK". [En línea]. Recuperado el 20 de febrero del 2013, Disponible en Web: <<http://www.microsoft.com/en-us/download/details.aspx?id=10121>>

- Debajo de "Selección de voz" se debe elegir una nueva voz del menú desplegable (Jorge). Se puede cambiar el texto en el campo para hacer que la voz lea una oración diferente de prueba (Figura 2.5).



**Figura 2.5: Voz adicional en español instalada**

## 2.2 INSTRUCCIONES DEL MICROSOFT KINECT SDK<sup>45</sup>

El Microsoft Kinect SDK cuenta con algunas estructuras de programación, específicamente contiene clases, métodos y propiedades, algunas de estas se detallan a continuación.

### 2.2.1 Clase KinectSensor

Representa un sensor Kinect

---

<sup>45</sup> "Microsoft Kinect SDK". [En línea]. Recuperado el 03 de marzo del 2013, Disponible en Web: <<http://msdn.microsoft.com/en-us/library/hh855419.aspx>>

### a. Método Start()

Comienza la transmisión de datos del sensor.

### b. Método Stop

Detiene la transmisión de datos del sensor.

### c. Propiedades

En la Tabla 2.1 se detallan las propiedades de esta clase.

**Tabla 2.1: Propiedades de la clase KinectSensor**

Nombre	Descripción
[S]KinectSensors	Obtiene una colección (arreglo) de los sensores Kinects.
DeviceConnectionId	Obtiene el identificador de conector USB. Cada conector USB devolverá un ID único.
Status	Obtiene el estado del sensor. Puede ser: <ul style="list-style-type: none"><li>• El estado no está definido.</li><li>• Desconectado</li><li>• Conectado</li><li>• El Kinect se está inicializando.</li><li>• Error</li><li>• El Kinect no está encendido.</li><li>• Alguna parte del Kinect no está listo aún.</li><li>• Dispositivo no genuino</li><li>• Dispositivo no soportado</li></ul>

- El conector USB no tiene suficiente ancho de banda.

IsRunning	Obtiene un valor que indica si el sensor está transmitiendo datos. Verdadero si el Kinect transmite datos y falso de otra manera.
ElevationAngle	Obtiene o ajuste el ángulo del Kinect, este valor está limitado entre el mínimo y máximo ángulo de las especificaciones
MinElevationAngle	Obtiene el máximo ángulo de elevación en grados de acuerdo a las especificaciones del Kinect.
MaxElevationAngle	Obtiene el mínimo ángulo de elevación en grados de acuerdo a las especificaciones.
ColorStream	Obtiene una secuencia de datos de color.
DepthStream	Obtiene una secuencia de datos de color.

## 2.2.2 KinectSensorCollection

Representa una colección (arreglo) de sensores Kinect, cada uno de los cuales es representado por un objeto KinectSensor.

### a. Método Get\_Item

Devuelve el objeto con propiedades en el índice conocido en una colección.

## **b. Propiedad Count**

Devuelve el número de sensores Kinect que están conectados al pc.

### **2.2.3 Clase ColorImageStream**

ColorImageFrame es un espacio de memoria para almacenamiento de los datos de color transmitidos de un sensor. También proporciona acceso a la dimensión y formato de los datos de color. La salida de datos se realiza en varios flujos de datos. Cada flujo de datos está representado por un objeto de secuencia de imagen, ya que cada trama de datos se organiza en un bloque rectangular o trama de datos que se pueden abordar de igual forma que acceder a los datos de píxel en una imagen.

#### **a. Método Enable**

Enable es el método para habilitar al sensor a transmitir los datos de color. Tiene dos formas:

- `ColorImageStream.Enable ()` activa el flujo de datos de color con el formato por defecto.
- `ColorImageStream.Enable (ColorImageFormat)` permite el flujo de datos de color con un formato especificado.

#### **c. Método OpenNextFrame**

Abre un nuevo cuadro para la imagen, la cantidad de tiempo de espera para realizarlo se coloca en milisegundos.

## **b. Propiedad Format**

Devuelve el formato (tipo de datos, resolución y velocidad del cuadro).

## 2.2.4 DepthImageStream

Representa una secuencia de datos de profundidad DepthImageFrame, es un espacio de memoria para almacenamiento de los datos de color transmitido de un sensor. También proporciona acceso a la dimensión y formato de los datos de color.

La salida de datos al igual que en el caso del ColorImageStream se realiza en varios flujos de datos. Cada flujo de datos está representado por un objeto de secuencia de imagen, ya que cada trama de datos se organiza en un bloque rectangular o trama de datos, que se pueden abordar de igual forma que acceder a los datos de píxel en una imagen.

### a. Método Enable

Enable es el método para habilitar al sensor para transmitir los datos de color. Tiene dos formas:

- DepthImageStream.Enable (): Habilita el flujo profundidad utilizando el formato por defecto, que es DepthImageFormat.Resolution640x480Fps30.
- DepthImageStream.Enable (DepthImageFormat): Permite el flujo de profundidad utilizando un formato especificado.

### b. Método OpenNextFrame

Abre un nuevo cuadro para la imagen, la cantidad de tiempo de espera para realizarlo se coloca en milisegundos.

### c. Propiedad Format

Obtiene el formato de los datos de profundidad que incluye el tipo de datos, la resolución y la velocidad de fotogramas.

## 2.2.5 Clase ImageStream

La clase base que el flujo de color y de profundidad utilizan para transferir los datos de un sensor.

### a. Propiedades

En la Tabla 2.2 se detallan las propiedades de esta clase.

**Tabla 2.2: Propiedades de la clase ImageStream**

Nombre	Descripción
FramePixelDataLength	Obtiene la longitud de los datos de píxel de cada fotograma.
FrameWidth	Obtiene el ancho de datos (en píxeles) de cada fotograma.
FrameHeight	Altura de datos (en píxeles) del fotograma.

### b. Método Disable

Desactiva los datos enviados desde el Kinect, flujo de datos de color o datos de profundidad, dependiendo de la clase.

## 2.2.6 DepthImageFrame Class

Contiene un espacio de memoria por fotograma para los datos de profundidad que fluyen del sensor. También proporciona acceso a las dimensiones y el formato de los datos.

### a. Método CopyDepthImagePixelDataTo

Método para copiar datos de profundidad por píxel a un arreglo pre-asignado.

### b. Método CopyPixelDataTo

Método para copiar datos de profundidad por píxel a un arreglo de bytes pre-asignado.

## 2.3 HERRAMIENTAS MICROSOFT SPEECH SDK

El centro de la API para text-to-speech (TTS) es ISpVoice. Gracias a esta interfaz, se puede realizar aplicaciones hablar un texto, modificar las características del habla, cambiar las voces, así como responder a eventos en tiempo real mientras habla. De hecho, la mayoría de las aplicaciones necesitan solamente esta única interfaz para lograr el soporte al TTS.

Cuando una aplicación crea primero un objeto ISpVoice, el objeto se inicializa la voz por defecto (definido en las propiedades de voz del panel de control). Un objeto ISpVoice es simplemente una sola instancia de una voz TTS específico. Cada objeto ISpVoice es una voz individual.

En la Tabla 2.3 se detallan las herramientas principales de este SDK.

**Tabla 2.3: Herramientas del Microsoft Speech SDK**

<b>Clase / Interfaz Relacionada</b>	<b>Método y Descripción</b>
	<p><b>GetVoices</b></p> <p>El método GetVoices devuelve una selección de voces disponibles para la voz. El valor devuelto es una colección ISpeechObjectTokens. Los atributos de la colección incluyen "género", "edad", "nombre", "idioma" y "distribuidor".</p>
<p><b>SpVoice / ISpVoice</b></p>	<p><b>Voice</b></p> <p>Obtiene y establece el miembro activo de la colección Voces. Puede ser considerado como la persona del objeto Voice, por ejemplo "Microsoft María" y "Mike Microsoft".</p>
	<p><b>Speak</b></p> <p>Habla el contenido de un texto o un archivo.</p>
<p><b>SpObjectToken /SpObjectTokens</b></p>	<p><b>Item</b></p> <p>Devuelve un miembro de la colección por su índice.</p>
	<p><b>GetDescription</b></p> <p>Devuelve el nombre del recurso representado por el objeto.</p>

## 2.4 HERRAMIENTAS DE LABVIEW

### 2.4.1 .NET

.NET es un conjunto de nuevas tecnologías de Microsoft que permite distribuir software de forma potente y sencilla. El código se distribuye en forma de servicios a los que puede accederse de forma remota; es independiente del lenguaje de programación y de la plataforma<sup>46</sup>. En la Figura 2.6 se muestra el menú de controles en Labview para el .NET.

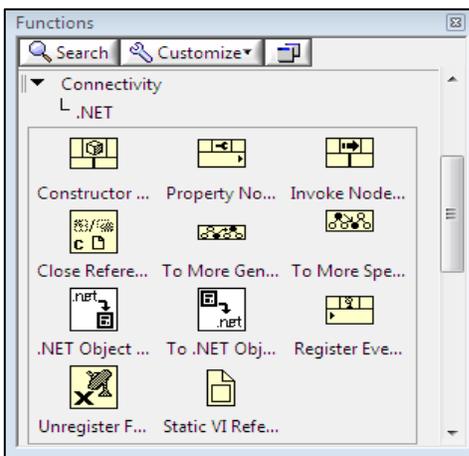


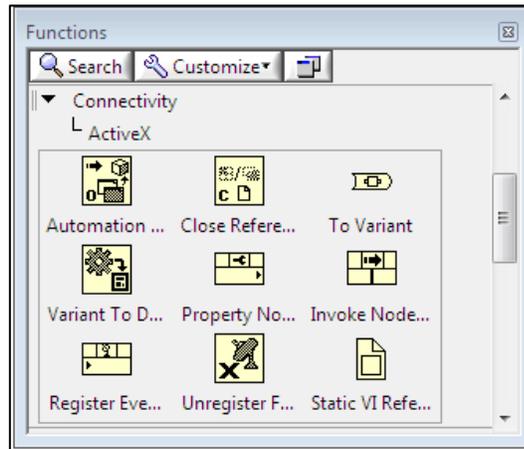
Figura 2.6: Menú .NET<sup>47</sup>

### 2.4.2 Controles ActiveX

Un control ActiveX puede tener propiedades, métodos y eventos asociados. Es un entorno en el que, mediante unos servicios basados en objetos, permite a diversos componentes comunicarse entre sí para reutilizar el código de los demás, de forma que se pueden enlazar unos programas con otros. En la Figura 2.7 se muestra el menú de controles ActiveX en Labview.

<sup>46</sup> Viscaíno, José. Sebastián José. "Labview: Entorno gráfico de programación". Segunda edición, España. 2011.

<sup>47</sup> Labview National Instruments



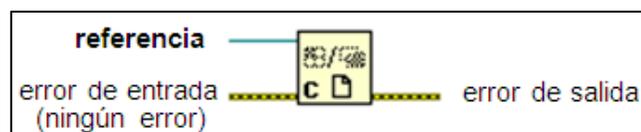
**Figura 2.7: Menú ActiveX<sup>48</sup>**

### 2.2.3 Herramientas del .NET y Controles ActiveX

Ambas herramientas entre algunos de sus controles en sus respectivas paletas tienen:

#### a. Automation Open/Close Reference

Sirve para abrir y cerrar referencias a controles ActiveX o .NET. (Figura 2.8).



**Figura 2.8: Close Reference<sup>49</sup>**

En el Automation Open, el terminal Automation Refnum puede venir de un control (Paleta de controles >Modem > Refnum > Automation Refnum), una constante o un Container.

<sup>48</sup> Labview National Instruments

<sup>49</sup> Labview National Instruments

Se utilizará un container cuando el control tenga una interfaz gráfica que se desee mostrar (Figura 2.9).

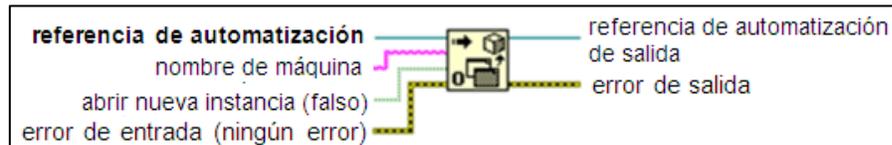


Figura 2.9: Automation Open<sup>50</sup>

## b. Property Node/ Invoke Node

Al cablear una referencia sobre estos nodos aparecerán las propiedades y métodos de los controles ActiveX o .NET. En la Figura 2.10 se muestra los terminales del Property Node y en la Figura 2.11 los del Invoke Node.

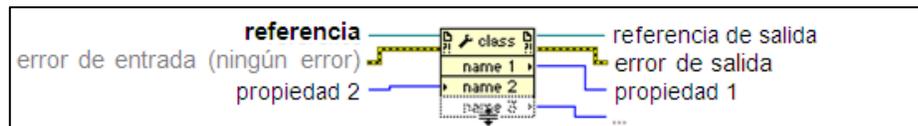


Figura 2.10: Property Node<sup>51</sup>

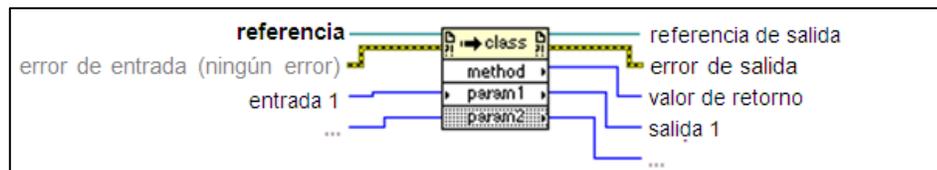


Figura 2.11: Invoke Node<sup>52</sup>

<sup>50</sup> Labview National Instruments

<sup>51</sup> Labview National Instruments

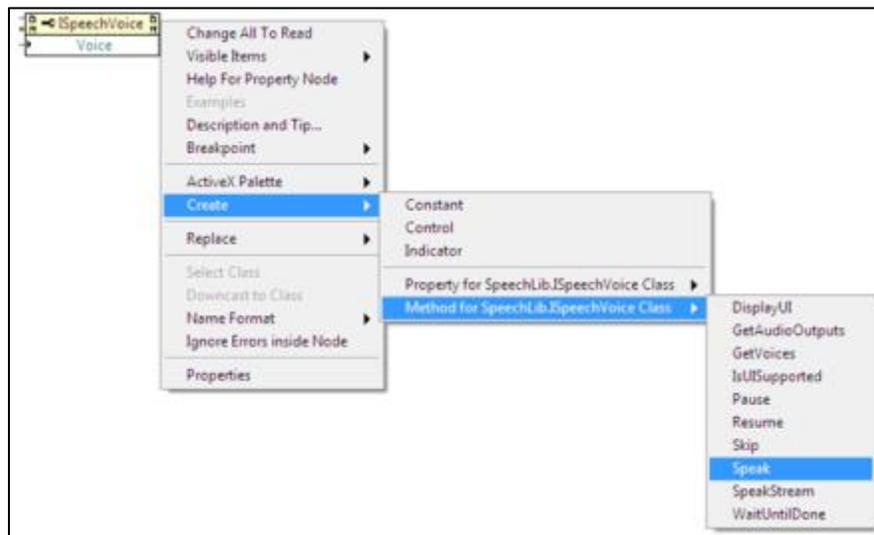
<sup>52</sup> Labview National Instruments

## 2.4.4 Creación métodos y propiedades de una clase

Una clase es una construcción que se utiliza como un modelo para crear objetos de ese tipo. Desde el punto de vista de la programación estructurada, una clase se asemejaría a un módulo, los atributos (propiedades) a las variables globales de dicho módulo, y los métodos a las funciones del módulo<sup>53</sup>. Un método representa acciones o procedimientos declarados dentro de una clase. Una propiedad representa información sobre un objeto.

Para crear un método o una propiedad:

- Se obtiene un Property Node de la paleta de controles.
- Se une el terminal de referencia. La clase se actualizará.
- En el menú desplegable del Property Node se debe navegar en: Create > Method for ... Class o Property for ... Class > y se selecciona el método o propiedad requerida respectivamente. (Figura 2.12).



**Figura 2.12: Ejemplo de creación de Método**

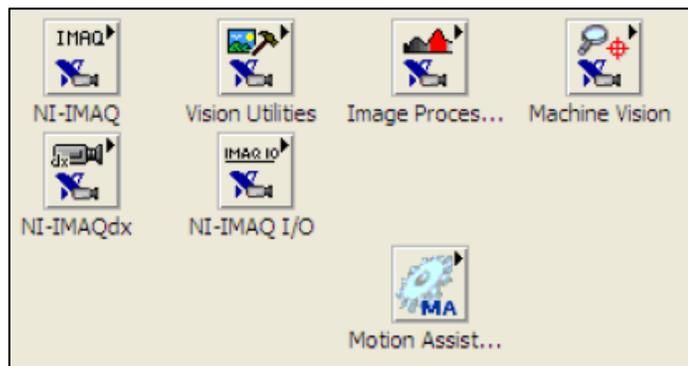
<sup>53</sup> "Clase programación". [En línea]. Recuperado el 25 de febrero del 2013, Disponible en Web: [http://zarza.usal.es/~fgarcia/doc/tuto2/I\\_1.htm](http://zarza.usal.es/~fgarcia/doc/tuto2/I_1.htm)

Un método de la misma clase se puede crear en base a un Property Node ya creado para esa clase.

Para modificar la propiedad seleccionada, se debe dar seleccionar el segmento Property del Property Node y el menú despegable permitirá seleccionar otra propiedad.

## 2.4.5 Toolkit IMAQ Vision

IMAQ Vision es una biblioteca (Figura 2.13) que permite implementar aplicaciones inteligentes de imagen y visión. Se pueden desarrollar instrumentos virtuales para procesamiento digital de imágenes como los detectores de borde y reconocimiento de patrones complejos entre otros. Empleado para realizar rutinas de visión de maquinarias, ajuste de patrones, análisis de partículas, reconocimiento óptico de caracteres y verificación de presencia, por mencionar algunas aplicaciones<sup>54</sup>.



**Figura 2.13: Herramientas de Vision and Motion de Labview<sup>55</sup>**

<sup>54</sup> "IMAQ Toolkit". [En línea]. Recuperado el 20 de enero del 2013, Disponible en Web: <<http://dspace.ups.edu.ec/bitstream/123456789/161/6/Capitulo%205.pdf>>

<sup>55</sup> Labview National Instruments

### a. Crear imagen

Crea una localidad de memoria temporal para una imagen, es decir, permite localizar los recursos necesarios para la imagen y las propiedades que ésta debe tener. Se encuentra en: Functions > Vision and Motion > Vision Utilities > Image Management > IMAQ Create. Los terminales de esta herramienta se ilustran en la Figura 2.14.

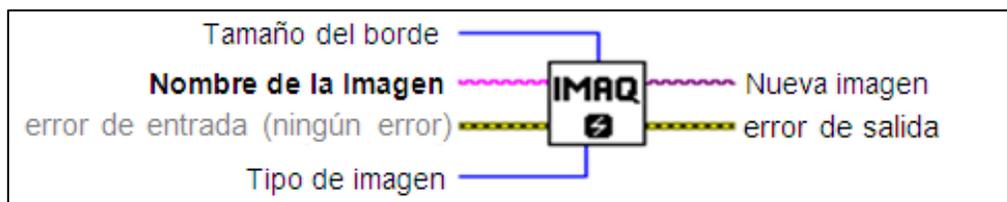


Figura 2.14: IMAQ Create<sup>56</sup>

### b. Copiar imagen

Copia las especificaciones y píxeles de una imagen en otra imagen del mismo tipo. Se puede usar esta función para mantener una copia original de una imagen (por ejemplo antes de procesar la imagen y así poderla comparar al final del proceso). Se encuentra en: Functions > Vision and Motion > Vision Utilities > Image Management > IMAQ Copy (Figura 2.15).

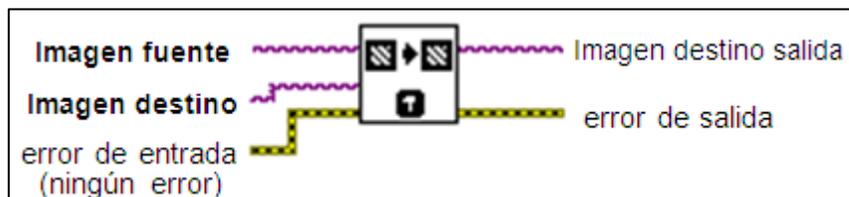


Figura 2.15: IMAQ Copy<sup>57</sup>

<sup>56</sup> Labview National Instruments

<sup>57</sup> Labview National Instruments

### c. Imagen a un arreglo

Extrae los píxeles de una imagen o parte de una imagen en un arreglo de dos dimensiones. Este array es codificado en 8 bits, 16 bits o punto flotante que es determinado por el tipo de la imagen de entrada. Los terminales de esta herramienta se ilustran en la Figura 2.16. Se encuentra en: Functions > Vision and Motion > Vision Utilities > Pixel Manipulation > IMAQ Image to Array.

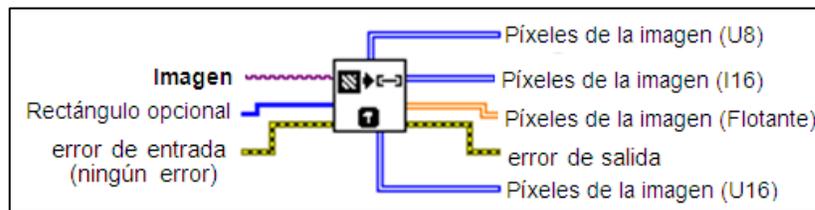


Figura 2.16: IMAQ Image To Array<sup>58</sup>

### d. Arreglo a imagen

Crea una imagen de un arreglo de dos dimensiones. Se encuentra en: Functions > Vision and Motion > Vision Utilities > Pixel Manipulation > IMAQ Array to Image. Los terminales de esta herramienta se ilustran en la Figura 2.17.

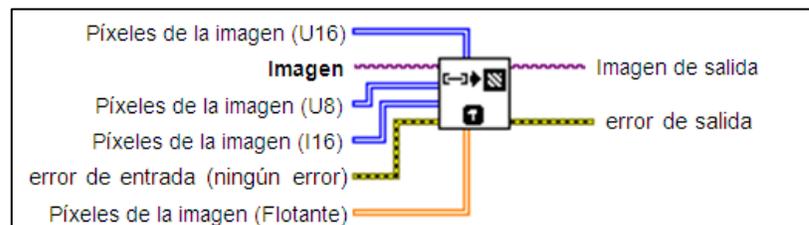
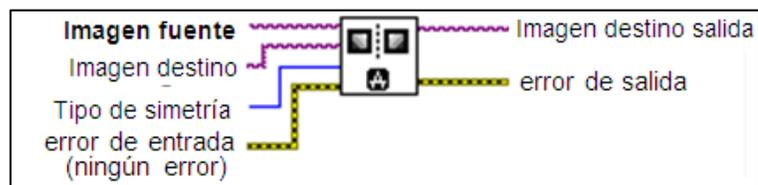


Figura 2.17: IMAQ Array to Image<sup>59</sup>

<sup>58</sup> Labview National Instruments

### e. Simetría de la imagen

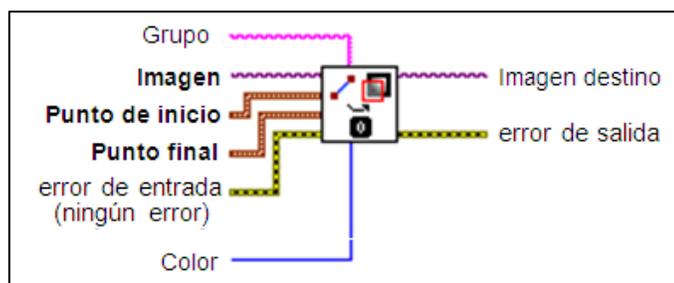
Transforma la imagen hasta su simetría, es decir crea un espejo a través de la reflexión (vuelta) de la imagen. La apariencia de la herramienta se muestra en la Figura 2.18. Se encuentra en **Functions > Vision and Motion > Vision Utilities > Image Manipulation > IMAQ Symmetry**. El tipo de simetría utilizada es vertical, aunque también tiene opción de realizarla horizontal, central o a través de sus diagonales.



**Figura 2.18: IMAQ Symmetry<sup>60</sup>**

### f. Superponer líneas

Superpone una línea en una imagen. Se encuentra en: **Functions > Vision and Motion > Vision Utilities > Overlay > IMAQ Overlay Line**. (Figura 2.19).



**Figura 2.19: IMAQ Overlay Line<sup>61</sup>**

<sup>59</sup> Labview National Instruments

<sup>60</sup> Labview National Instruments

<sup>61</sup> Labview National Instruments

### g. Crear casco convexo

Señala el polígono convexo de cada partícula de la imagen (rellena). Se encuentra en: Functions > Vision and Motion > Image Processing > Morphology > IMAQ Convex Hull (Figura 2.20).

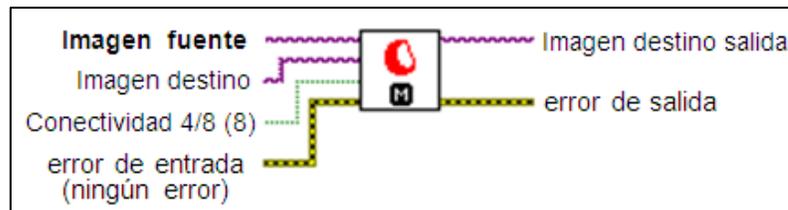


Figura 2.20: IMAQ Convex Hull<sup>62</sup>

Esta herramienta crea un polígono convexo que contiene todos los puntos en la partícula. La Figura 2.21 muestra dos partículas, que se muestran en color gris, y sus cascos convexos respectivos, las zonas delimitadas por las líneas negras.

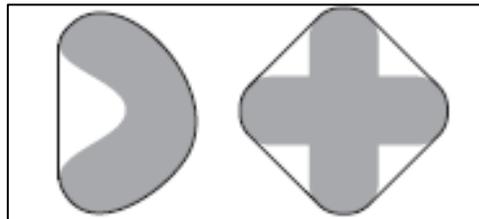


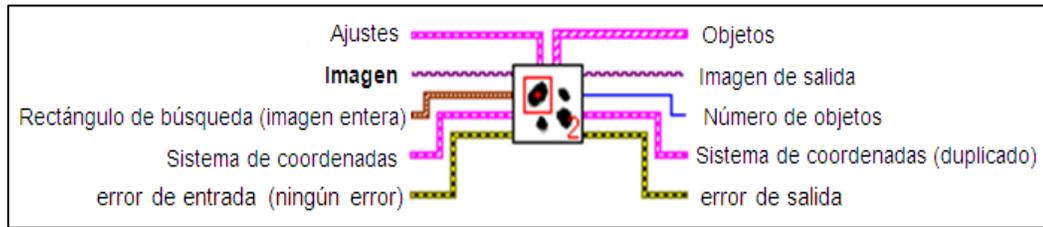
Figura 2.21: Cascos convexos a dos partículas independientes<sup>63</sup>

### h. Contar objetos

Localiza, cuenta y dimensiona objetos en un área de búsqueda rectangular. Se encuentra en: Functions > Vision and Motion > Machine Vision > Count and Measure Objects > IMAQ Count Objects 2 (Figura 2.22).

<sup>62</sup> Labview National Instruments

<sup>63</sup> Labview National Instruments

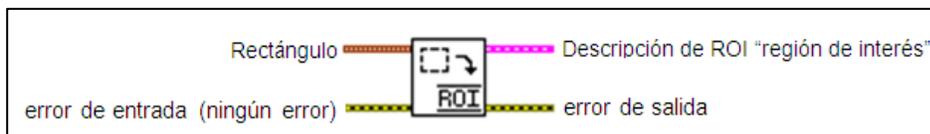


**Figura 2.22: IMAQ Count Objects 2<sup>64</sup>**

Además tiene opción de hacer caso omiso de los objetos más pequeños o más grandes que los tamaños indicados. Otras opciones permiten rechazar los objetos que toquen los bordes de la zona de búsqueda y hacer caso omiso de los agujeros que el proceso de segmentación pudo crear en los objetos. Los objetos segmentados son localizados y medidos. IMAQ Objetos Count 2 puede superponer en la imagen devuelta la posición de la zona buscada, los centros y los rectángulos delimitadores de los objetos detectados.

Esta herramienta realiza los siguientes pasos:

- Transforma la zona de búsqueda en la región de interés. La opción por defecto es la imagen entera. Lo realiza por medio de la herramienta IMAQ Convert Rectangle to ROI (Figura 2.23).

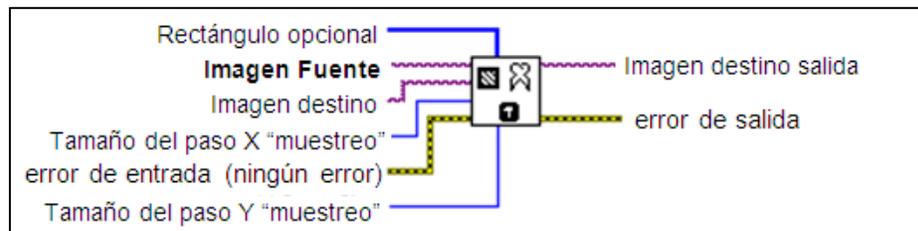


**Figura 2.23: IMAQ Convert Rectangle to ROI<sup>65</sup>**

<sup>64</sup> Labview National Instruments

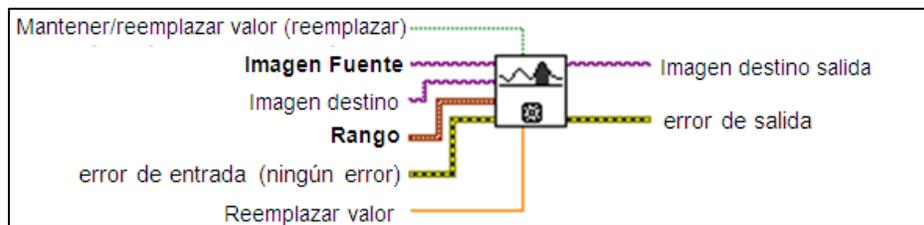
<sup>65</sup> Labview National Instruments

- Extrae la porción de la imagen delimitada en el terminal Search Rect del VI Count Objects 2, lo realiza con la herramienta IMAQ Extract (Figura 2.24).



**Figura 2.24: IMAQ Extract<sup>66</sup>**

- Determina el rango de umbral y lo aplica a la imagen por medio de la herramienta IMAQ Threshold (Figura 2.25). Las partículas se caracterizan por un rango de intensidad. La imagen resultante es una imagen binaria. Píxeles dentro del intervalo de umbral se consideran parte de una partícula. Los píxeles fuera del intervalo de umbral se consideran parte del fondo.

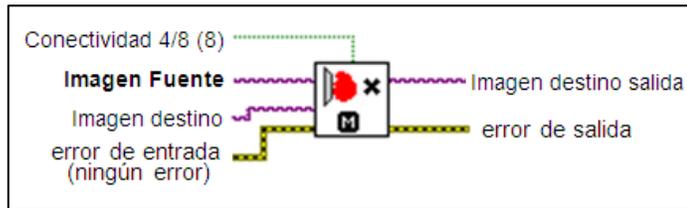


**Figura 2.25: IMAQ Threshold<sup>67</sup>**

- Rechaza los objetos que tocan los bordes si es necesario, de acuerdo a las opciones especificadas. Por medio de la herramienta IMAQRejectBorder dla Figura 2.26.

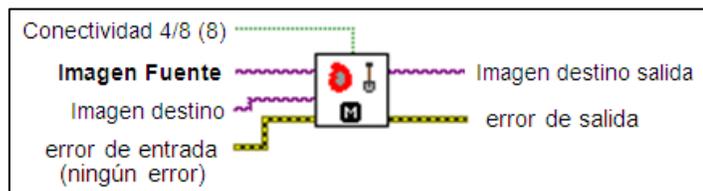
<sup>66</sup> Labview National Instruments

<sup>67</sup> Labview National Instruments



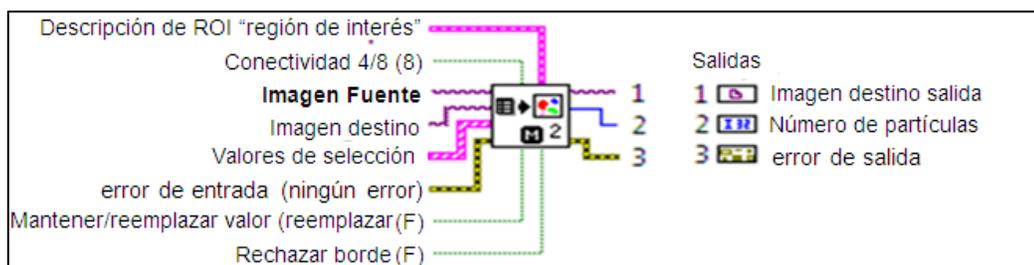
**Figura 2.26: IMAQ RejectBorder<sup>68</sup>**

- Llena los agujeros si es necesario, de acuerdo a las opciones especificadas mediante la herramienta IMAQ FillHole (Figura 2.27).



**Figura 2.27: IMAQ FillHole<sup>69</sup>**

- Filtra las partículas en la imagen de acuerdo a los valores mínimo y máximo de las opciones, con la herramienta IMAQ Particle Filter 2 (Figura 2.28) que mantiene o remueve cada partícula de la imagen.



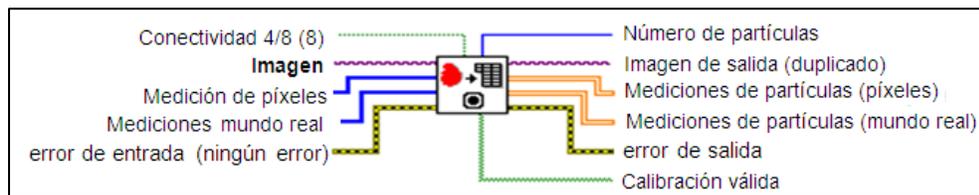
**Figura 2.28: IMAQ Particle Filter 2<sup>70</sup>**

<sup>68</sup> Labview National Instruments

<sup>69</sup> Labview National Instruments

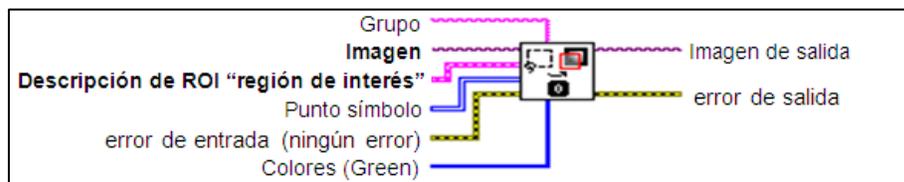
<sup>70</sup> Labview National Instruments

- Obtiene los parámetros de forma de todos los objetos. Mediante la herramienta IMAQ Particle Analysis (Figura 2.29) que retorna el número de partículas detectadas en una imagen binaria y un array de dos dimensiones con las medidas requeridas de cada partícula como centro, vértices delimitadores de la partícula, área.



**Figura 2.29: IMAQ Particle Analysis<sup>71</sup>**

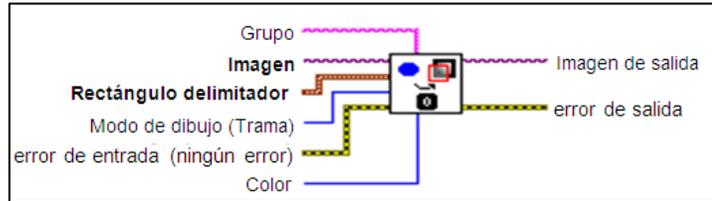
- Sobrepone los resultados a la imagen original. Este lo realiza mediante cuatro herramientas: IMAQ Overlay ROI (Figura 2.30) que superpone la región de interés a la imagen, IMAQ Overlay Oval (Figura 2.31) que sobrepone un óvalo en la imagen para indicar el centro, IMAQ Overlay Rectangle (Figura 2.32), que sobrepone un rectángulo el cual delimita a cada objeto y la herramienta IMAQ Overlay Text (Figura 2.33), que sobrepone un texto a la imagen que indica el índice de objeto detectado.



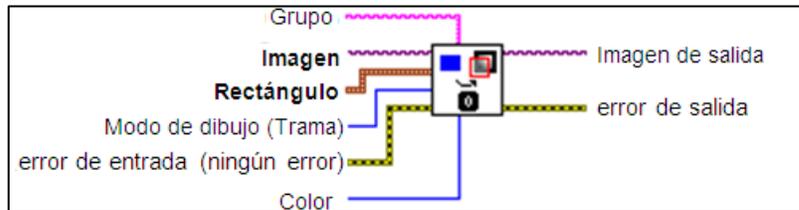
**Figura 2.30: IMAQ Overlay ROI<sup>72</sup>**

<sup>71</sup> Labview National Instruments

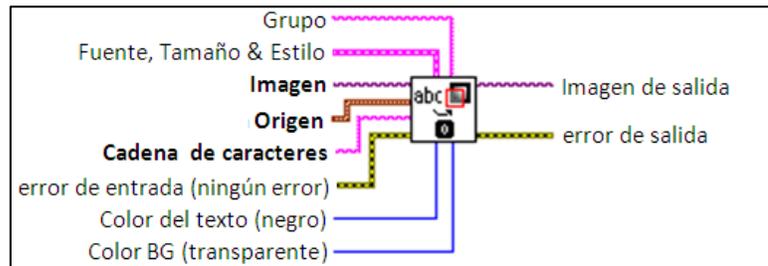
<sup>72</sup> Labview National Instruments



**Figura 2.31: IMAQ Overlay Rectangle<sup>73</sup>**



**Figura 2.32: IMAQ Overlay Rectangle**

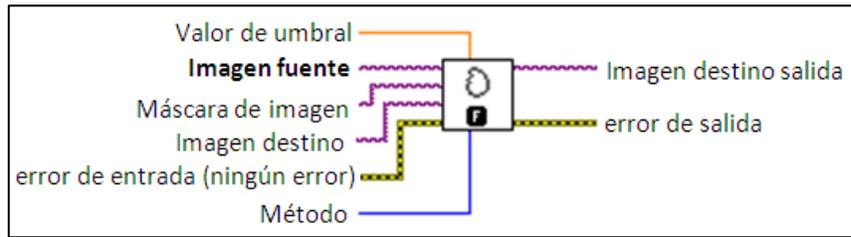


**Figura 2.33: IMAQ Overlay Text**

### **i. Detectar los bordes**

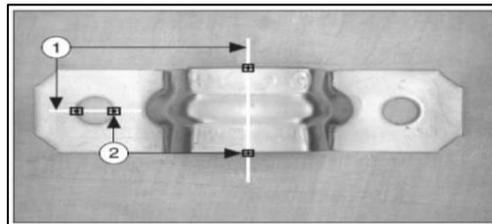
Extrae el contorno en niveles de grises. Se encuentra en: Functions > Vision and Motion > Image Processing > Filters > IMAQ Edge Detection. La herramienta en Labview se muestra en la Figura 2.34.

<sup>73</sup> Labview National Instruments



**Figura 2.34: IMAQ Edge Detection**

El detector de bordes de IMAQ Vision trabaja usando un arreglo de una sola dimensión como se muestra en la Figura 2.35.



**Figura 2.35: 1) Líneas de búsqueda, 2) bordes<sup>74</sup>**

NI Vision proporciona dos métodos de detección de bordes, ambos métodos calculan la intensidad de los pixeles a lo largo de una línea, para determinar si la línea atraviesa o no un borde. Los métodos proporcionados por NI Vision son:

- Detección simple del borde.
- Detección avanzada del borde.

La detección simple consiste en tomar los valores de todos los pixeles que se encuentran en la línea, establecer un valor de umbral con una histéresis que nos permita identificar el valor de inicio y fin del borde.

<sup>74</sup> Labview National Instruments

La detección avanzada del borde se basa en el uso de un kernel (matriz de convolución, sirven para aplicar filtros a una imagen) para calcular la intensidad del borde.

## j. Image Display (panel frontal)

Image Display contiene un área de visualización de la imagen, en la parte lateral izquierda presenta un conjunto de herramientas que permiten hacer ampliaciones o reducciones de la imagen, mientras que en la parte lateral derecha e inferior presenta barras de desplazamiento. Se encuentra en: Controls > Vision > Image Display. Su apariencia en el panel frontal se indica en la Figura 2.36.

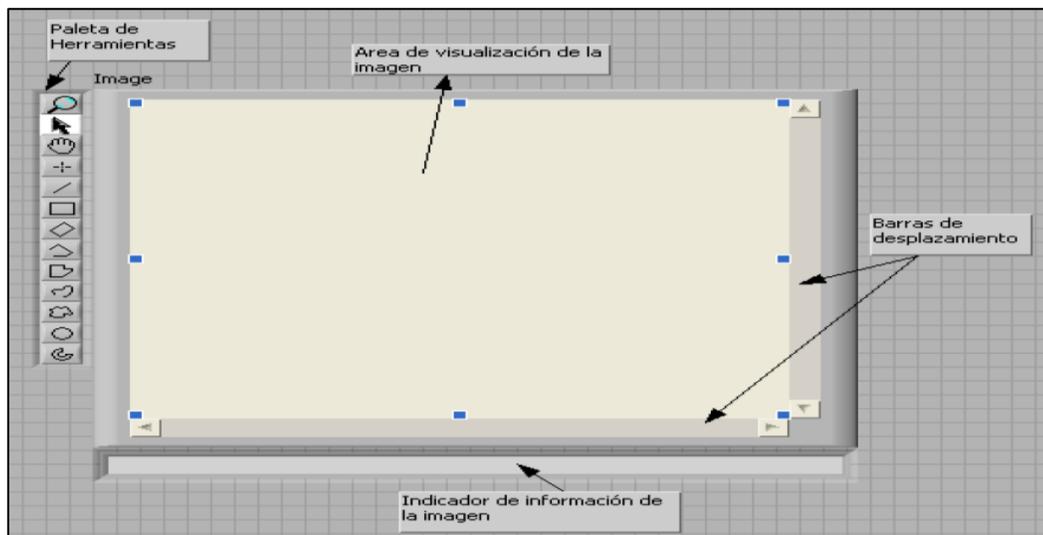


Figura 2.36: Panel Frontal de IMAQ Display<sup>75</sup>

<sup>75</sup> Labview National Instruments

## 2.4.6 Herramientas adicionales

### a. Simple Error

Indica si un error ocurrió. Si un error ocurrió, este VI retorna una descripción del error y opcionalmente despliega un cuadro de diálogo. Se encuentra en Functions > Programming > Dialog & User Interface > Simple Error Handler. Véase Figura 2.37.

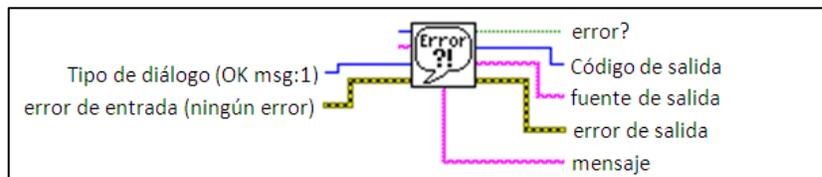


Figura 2.37: Simple Error Handler<sup>76</sup>

### b. Error in / Error out<sup>77</sup>

Los clusters agrupan elementos de datos de diferentes tipos. Por ejemplo el cluster de error de LabVIEW (Figura 2.38) combina un valor Booleano, un valor numérico y uno de cadena de caracteres. Se localiza en: Controls > Modern > Array, Matrix & Cluster.

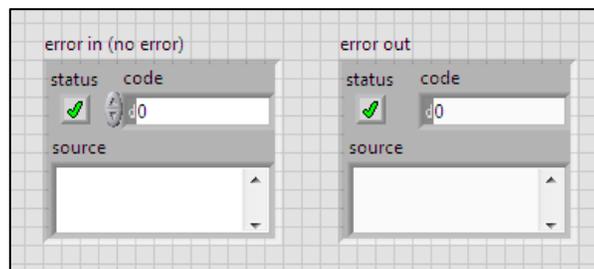


Figura 2.38: Panel Frontal del Error In (izquierda) y Error Out (derecha)<sup>78</sup>

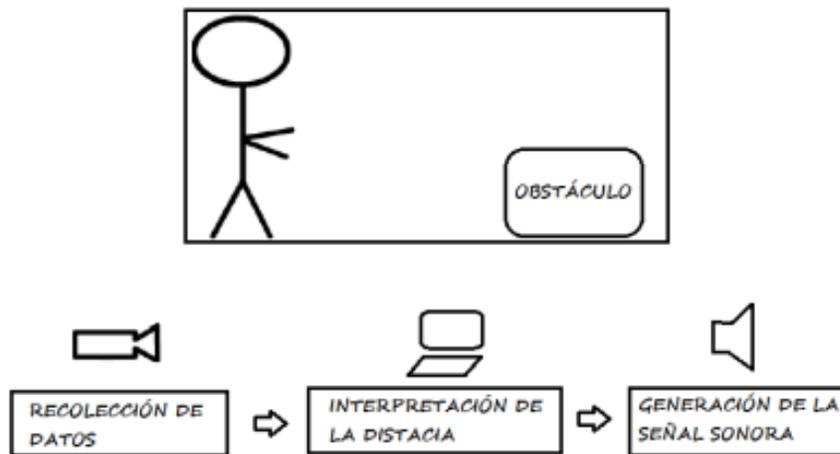
<sup>76</sup> Labview National Instruments

<sup>77</sup> "cluster", [En línea]. Recuperado el 26 de febrero del 2013, Disponible en Web: <<http://www.etitudela.com/entrenadorcomunicaciones/downloads/labviewusodearraystring.pdf>>

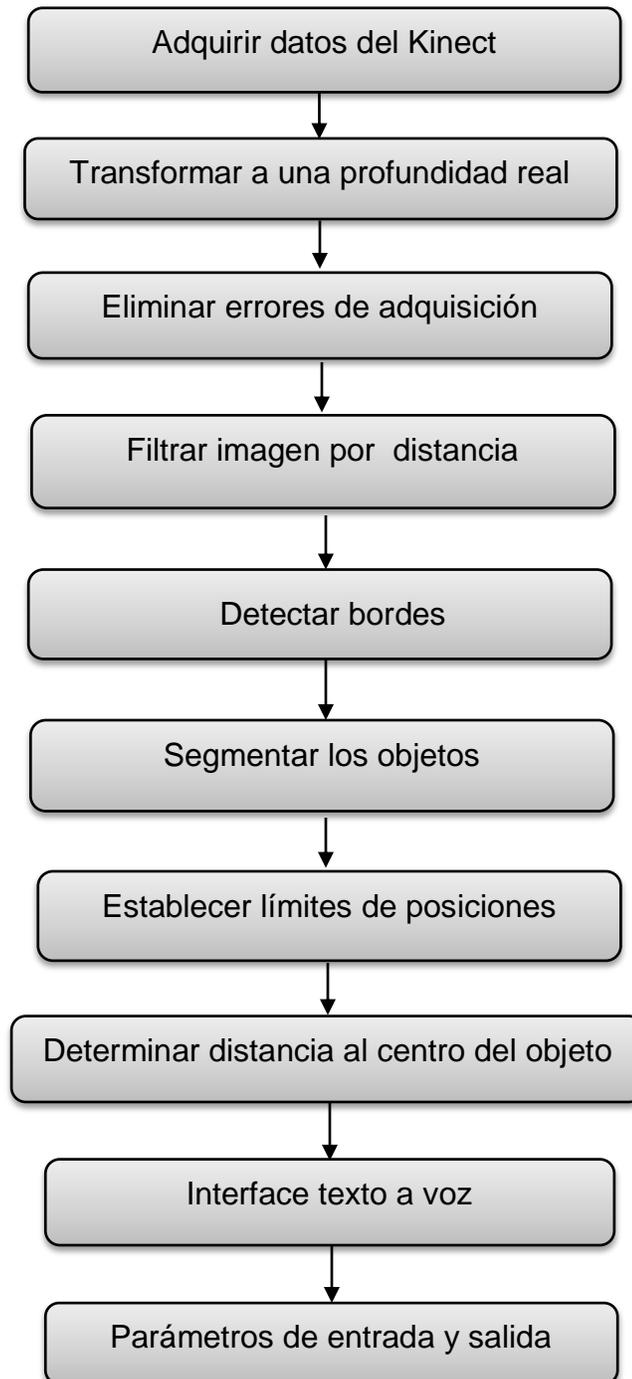
El Estado es VERDADERO (X) si se produce un error o FALSO (✓) para indicar una advertencia o que no hay errores. El Código es el error o código de advertencia. La Fuente describe el origen del error o advertencia.

## 2.5 ETAPAS DEL ALGORITMO

El algoritmo requerido debe ser capaz de identificar los obstáculos presentes y emitir una señal auditiva de esta información como se indica en la Figura 2.39) para ello consta de las etapas descritas en la Figura 2.40.



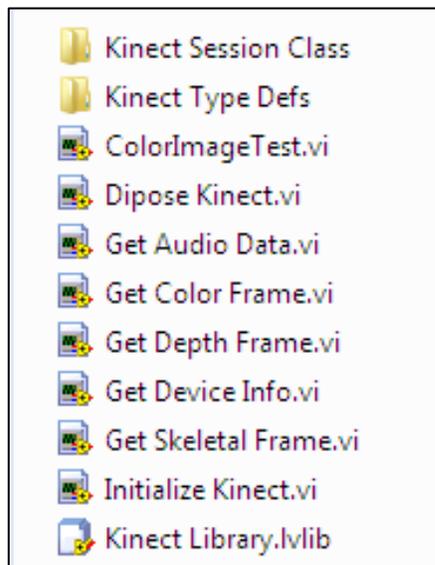
**Figura 2.39: Resumen del algoritmo**



**Figura 2.40: Etapas del algoritmo**

## 2.5.1 Etapa 1: Interface Kinect

La interfaz de Kinect con Labview está basada en el paquete del sensor Kinect para Labview que es el Kinect SDK v1.0.zip, a este se le han realizado algunos cambios, para utilizar solo las herramientas que se necesitan para el proyecto, pero el paquete original (Figura 2.41) se encuentra disponible en: <https://decibel.ni.com/content/servlet/JiveServlet/download/22530-3-47391/Kinect%20SDK%20v1.0.zip>



**Figura 2.41: Elementos del paquete Kinect para Labview**

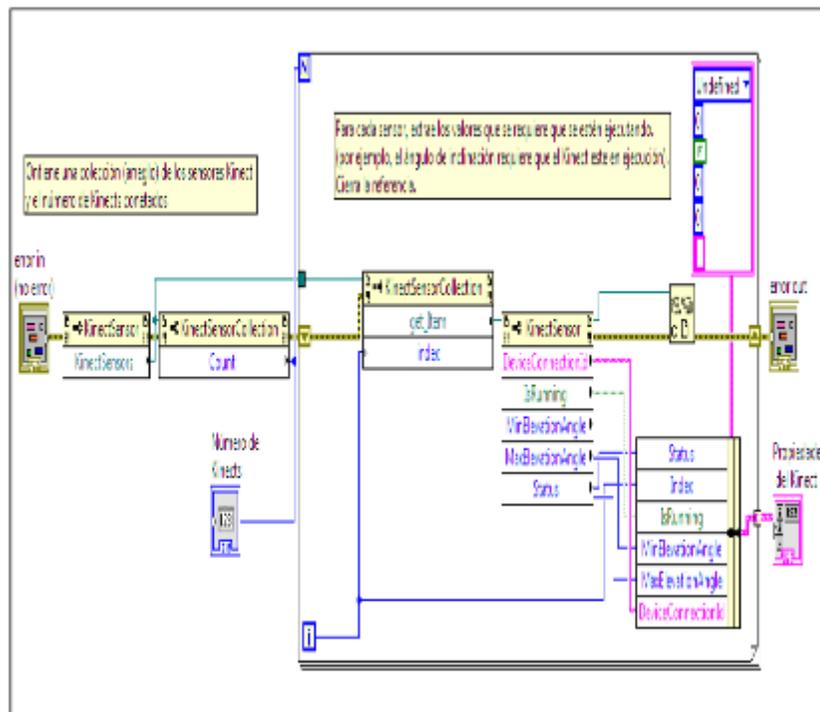
### a. Obtener Información

Retorna un arreglo de la información sobre todos los Kinects conectados al computador, la información puede ser empleada para inicializar el Kinect. El ícono se muestra en la Figura 2.42.



**Figura 2.42: Get Device Info.VI<sup>79</sup>**

En la Figura 2.43 se muestra el diagrama de bloques.



**Figura 2.43: Diagrama de bloques del VI que obtiene información del Kinect.**

Para realizar esta acción el VI:

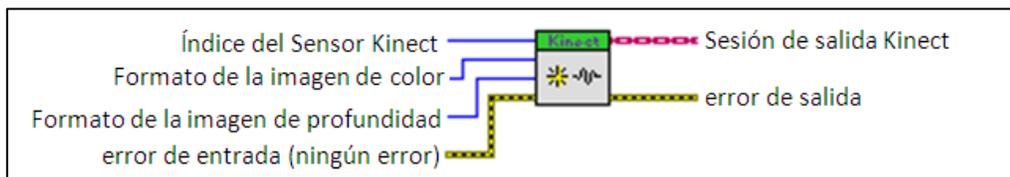
- Representa a cada sensor Kinect con la Clase KinectSensor.

<sup>79</sup> "Kinect SDK v1.0". [En línea]. Recuperado el 20 de octubre del 2012, Disponible en Web: <<https://decibel.ni.com/content/servlet/JiveServlet/download/22530-3-47391/Kinect%20SDK%20v1.0.zip>>

- Obtiene una colección (arreglo) de los sensores Kinects mediante la Clase KinectSensorCollection.
- Devuelve el número de Kinect Conectados con la propiedad Count.
- Para cada sensor, extrae los valores que se requiere que se estén ejecutando. (por ejemplo, el ángulo de inclinación requiere que el Kinect esté en ejecución), lo realiza con el método get\_Item y las propiedades DeviceConeectionId, IsRunning, MinElevationAngle, MaxElevationAngle, Status. Estos datos son agrupados en un clúster y colocado como un terminal de salida del VI.

### b. Inicializar el Kinect

Este es el VI básico de inicialización para la Kinect API. Es utilizado para conectarse a cualquier sensor Kinect conectado al ordenador. Sus terminales de indican en la Figura 2.44.



**Figura 2.44: Initialize Kinect VI<sup>80</sup>**

El diagrama de bloques de indica en la Figura 2.45. En este VI se realizan las siguientes acciones:

- Asegurar que hay un Kinect conectado al computador, por medio de la información del Get Device Info.VI.

<sup>80</sup> "Kinect SDK v1.0". [En línea]. Recuperado el 20 de octubre del 2012, Disponible en Web: <<https://decibel.ni.com/content/servlet/JiveServlet/download/22530-3-47391/Kinect%20SDK%20v1.0.zip>>

- Extraer a un Kinect de la colección con el método `get_Item`.
- Iniciar el Kinect con el método `Start`.
- Recoger toda la información acerca del Kinect con las propiedades de la Clase `KinectSensor`.
- Rellenar la información sobre la imagen de color y activar el sensor de color, con las clases `ColorImageStream` y `DepthImageStream` y las propiedades `Format` y `Enable`.
- Disponer de los datos para el ángulo de inclinación, como máximo y mínimo valor.
- Reservar memoria para las imágenes IMAQ, con la herramienta IMAQ Create del Toolkit Vision and Motion.

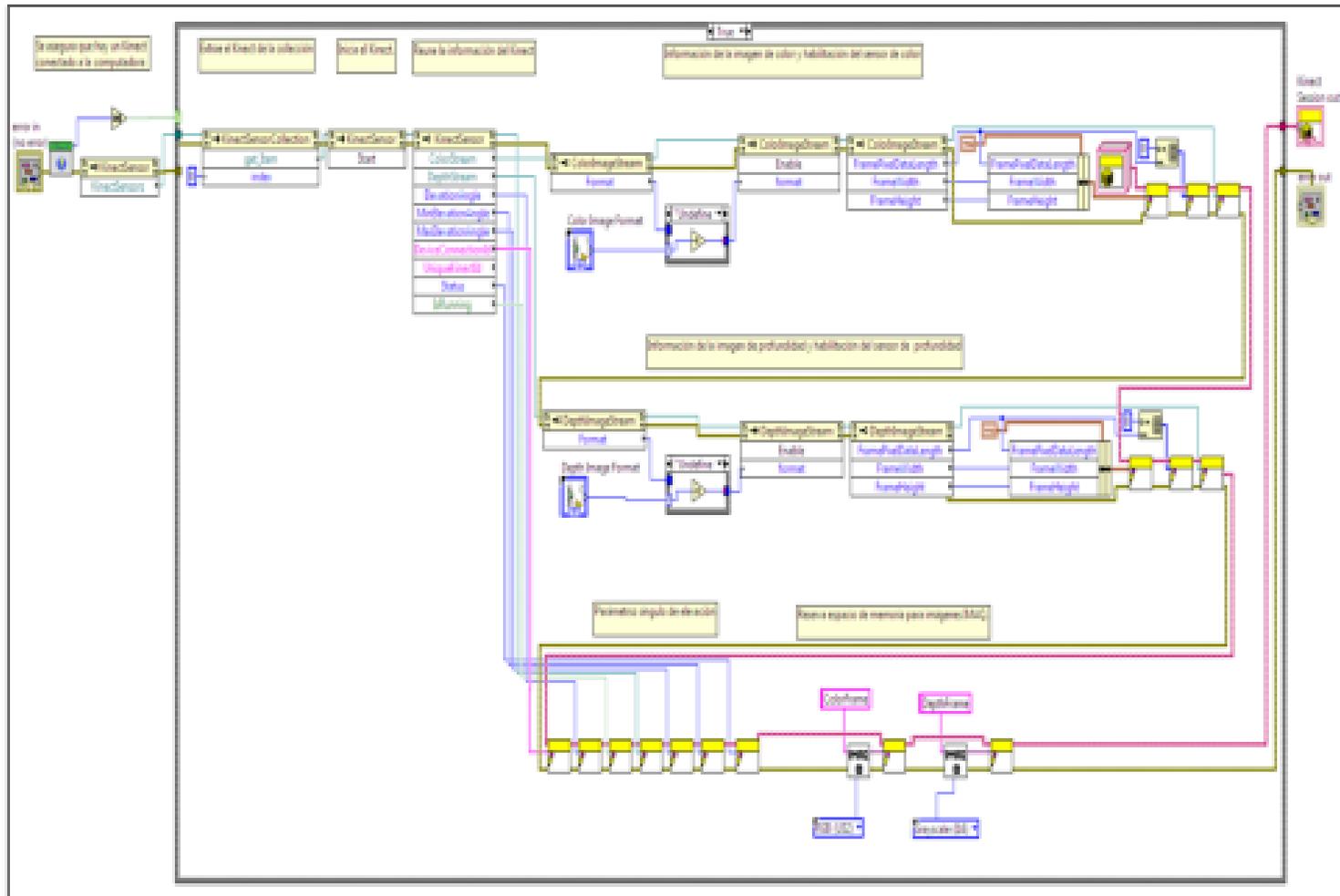


Figura 2.45: Diagrama de bloques del iniciador del Kinect

### c. Establecer el ángulo de elevación

Este VI ajusta el ángulo del motor del Kinect en el eje vertical, si el valor está fuera del rango, este será forzado a valor máximo o mínimo (Figura 2.46).

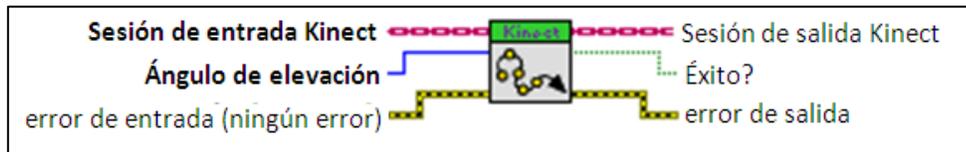


Figura 2.46: Set Elevation Angle VI<sup>81</sup>

A continuación se muestra el diagrama de bloques (Figura 2.47) en el cual se acciona el ángulo de entrada dentro del rango apropiado. Mediante la Propiedad ElevationAngle.

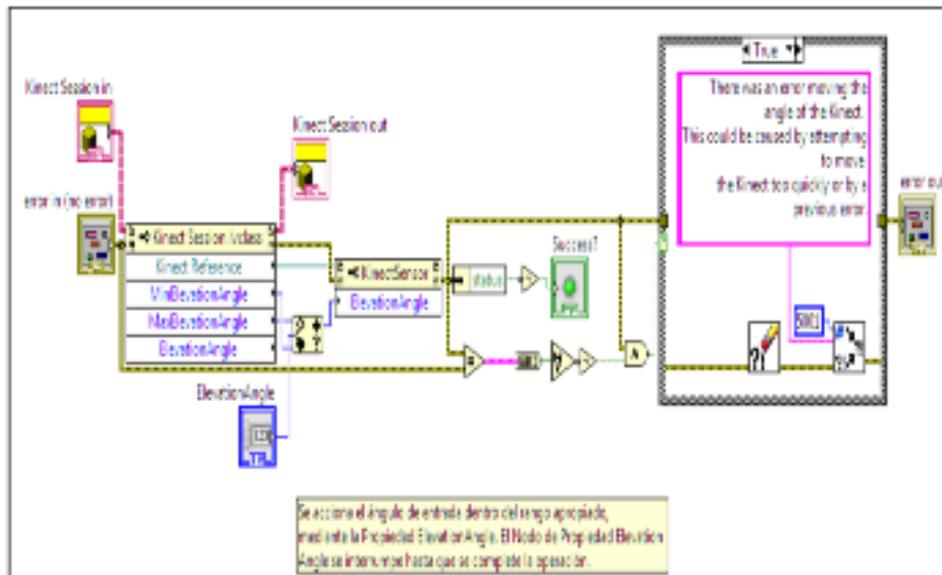


Figura 2.47: Diagrama de bloques del ajuste del ángulo

<sup>81</sup> "Kinect SDK v1.0". [En línea]. Recuperado el 20 de octubre del 2012, Disponible en Web: <<https://decibel.ni.com/content/servlet/JiveServlet/download/22530-3-47391/Kinect%20SDK%20v1.0.zip>>

El Nodo de Propiedad ElevationAngle se interrumpe hasta que se complete la operación. No deben ser pasados errores, la bandera de éxito indica si la propiedad se establece correctamente. Un error se produce por intentar mover el Kinect demasiado rápido o por un error anterior.

#### d. Obtener la imagen RGB

El VI “Get Color Frame” recupera la última imagen de color (Figura 2.48).



**Figura 2.48: Get Color Frame<sup>82</sup>**

Este VI copia los datos de píxel mediante el método CopyPixelDataTo y los convierte en un apropiado arreglo de color de 32 bits mediante herramientas del Toolkit Vision and Motion y arreglos. Este arreglo se convierte en una imagen IMAQ. Puede existir un error al obtener una imagen de Color. Esto puede ser causado por llamar a este VI con demasiada frecuencia o por algún problema con la conexión con el Kinect. En la Figura 2.49 se muestra el diagrama de bloques del VI.

<sup>82</sup> “Kinect SDK v1.0”. [En línea]. Recuperado el 20 de octubre del 2012, Disponible en Web: <<https://decibel.ni.com/content/servlet/JiveServlet/download/22530-3-47391/Kinect%20SDK%20v1.0.zip>>

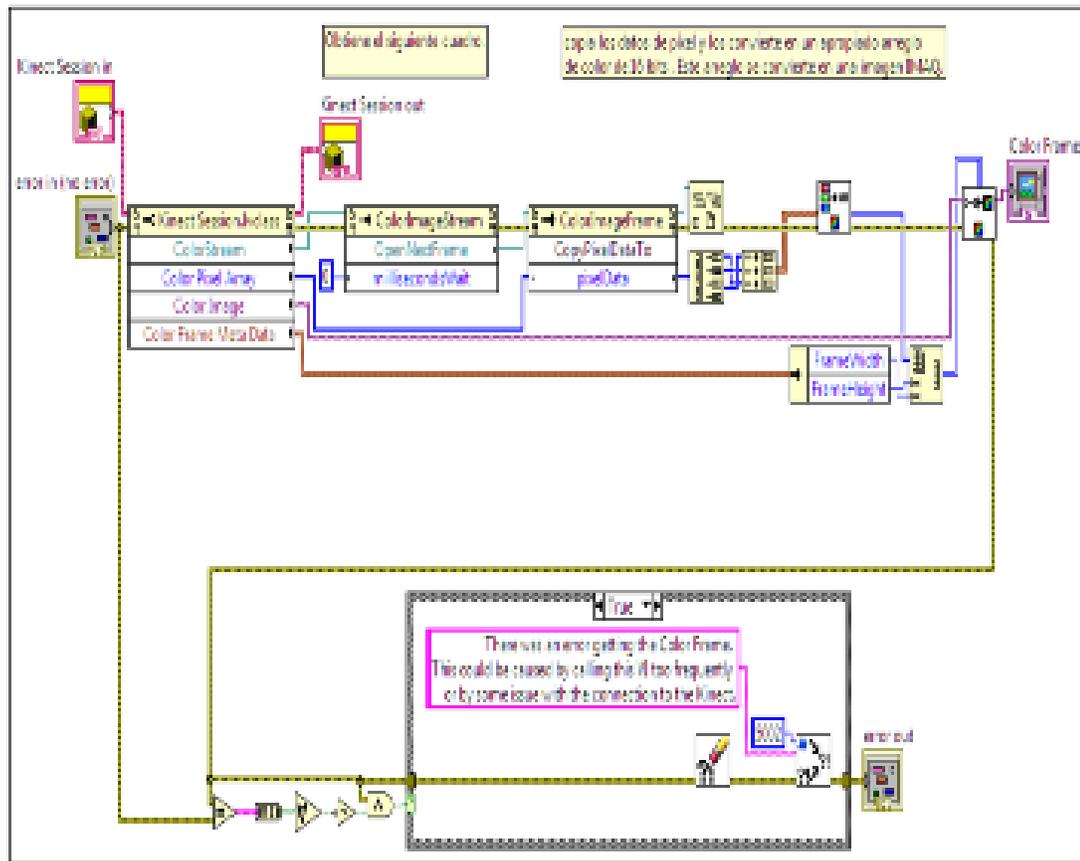


Figura 2.49: Diagrama de bloques la adquisición de la imagen de color

### e. Obtener la imagen de profundidad

El VI “Get Depth Frame” recupera la última imagen (Figura 2.50).

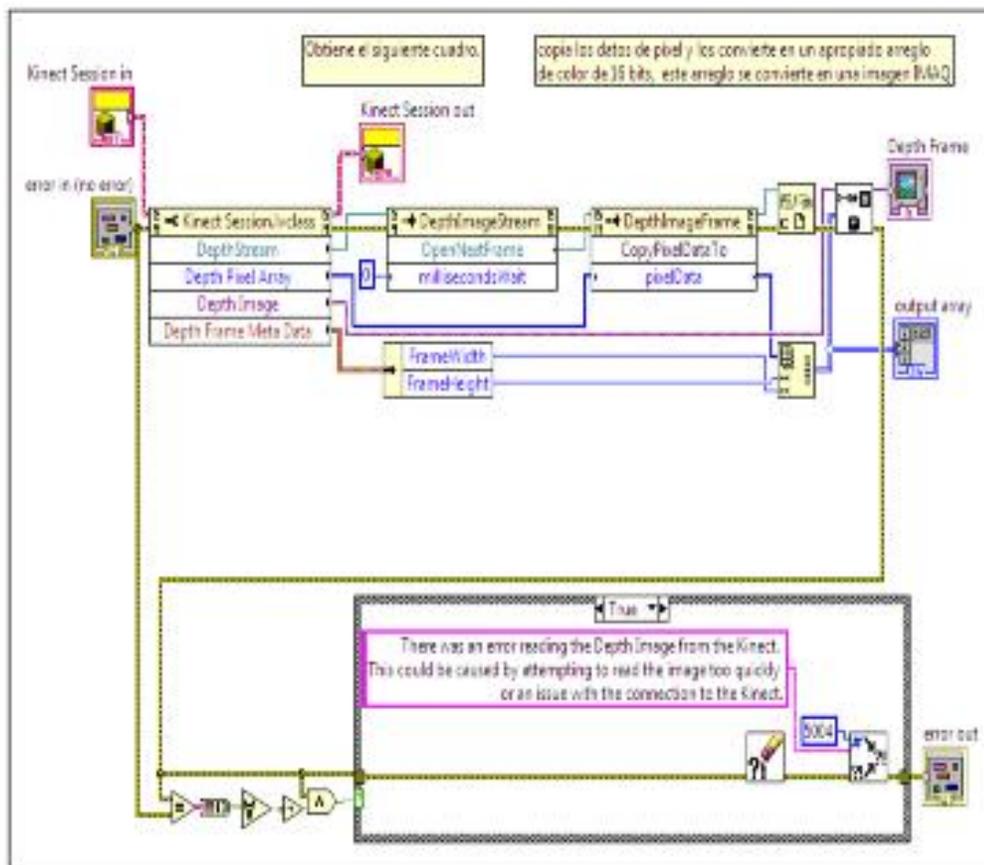


Figura 2.50: Get Depth Frame VI<sup>83</sup>

<sup>83</sup> “Kinect SDK v1.0”. [En línea]. Recuperado el 20 de octubre del 2012, Disponible en Web: <<https://decibel.ni.com/content/servlet/JiveServlet/download/22530-3-47391/Kinect%20SDK%20v1.0.zip>>

Este VI copia los datos de píxel y los convierte en un apropiado arreglo de color de 16 bits con el método CopyPixelFormat. Este arreglo se convierte en una imagen IMAQ, como se muestra en la Figura 2.51.

Al igual que en el VI para obtener la imagen de color, un error se puede producir por llamar a este VI con demasiada frecuencia o por algún problema con la conexión con el Kinect.



**Figura 2.51: Diagrama de bloques la adquisición de la imagen de profundidad.**

## f. Detener el sensor Kinect

Detiene cada sensor y detiene en general el Kinect. Además desecha todos los objetos presentes. El VI se muestra en la Figura 2.52.

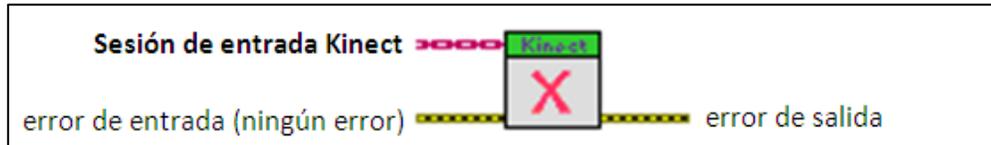


Figura 2.52: Dipose Kinect VI<sup>84</sup>

El diagrama de bloques de indica en la Figura 2.53.

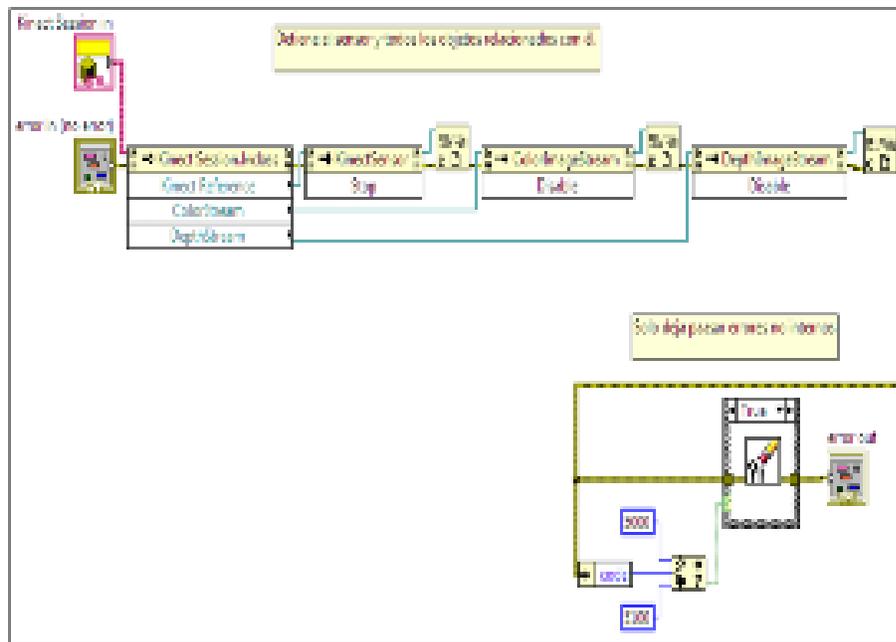


Figura 2.53: Diagrama de bloques del VI que detiene el Kinect

<sup>84</sup> "Kinect SDK v1.0". [En línea]. Recuperado el 20 de octubre del 2012, Disponible en Web: <<https://decibel.ni.com/content/servlet/JiveServlet/download/22530-3-47391/Kinect%20SDK%20v1.0.zip>>

Esta función limpia todas las referencias y detiene el sensor Kinect con el método Stop de la clase KinectSensor y Disable de las clases de color y profundidad. Aunque no es estrictamente requerido llamar a esta función antes de terminar la ejecución del VI, este puede prevenir fugas de memoria ante un requerimiento eventual de Labview a ser reiniciado.

Con los VI antes descritos, en esta fase se inicia el Kinect, se ajusta el ángulo del motor del Kinect en el eje vertical, se obtiene la imagen de color y la de profundidad. Se toman dos imágenes, una de la parte superior y otra de la parte inferior. De cada una se obtendrá la simetría para eliminar el error de espejo a través de la reflexión (vuelta) de la imagen (Figura 2.54).

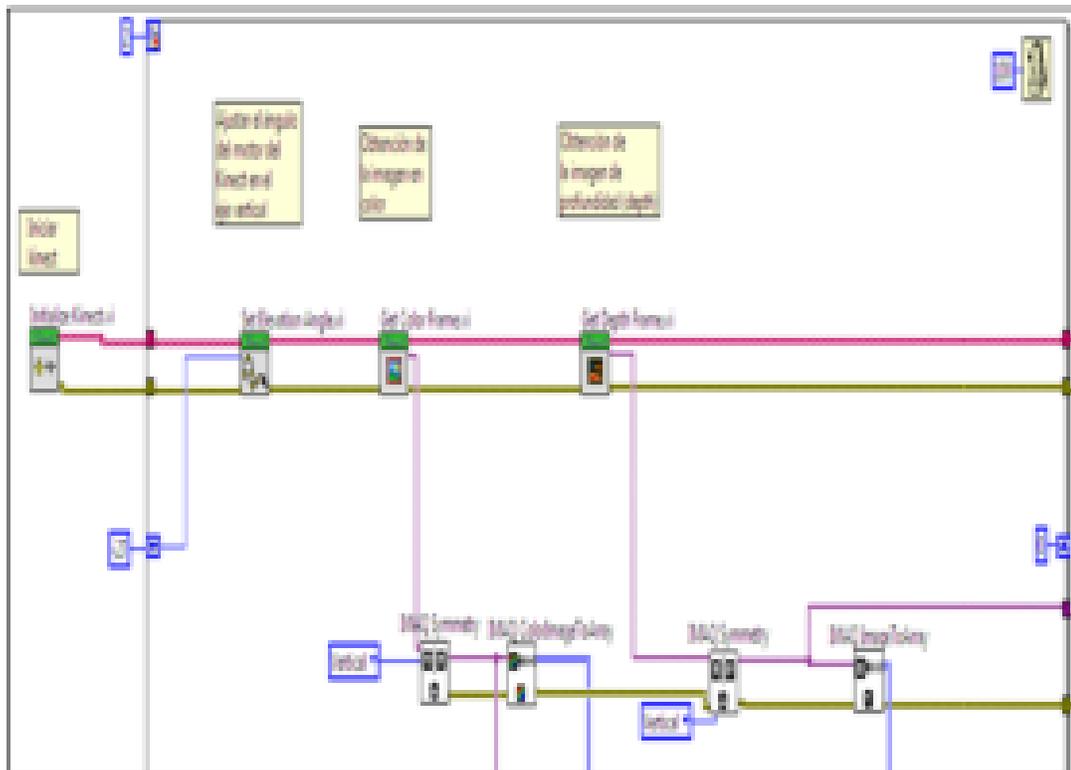


Figura 2.54: Adquisición de datos del Kinect

## 2.5.2 Etapa 2: Transformar a una profundidad real

Se transforma a una profundidad real los datos entregados por el Kinect, obteniendo solo los bits que representan esta magnitud (Figura 2.55).

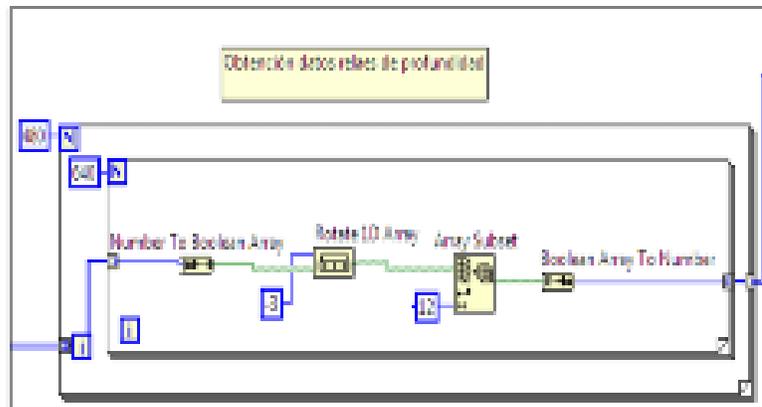
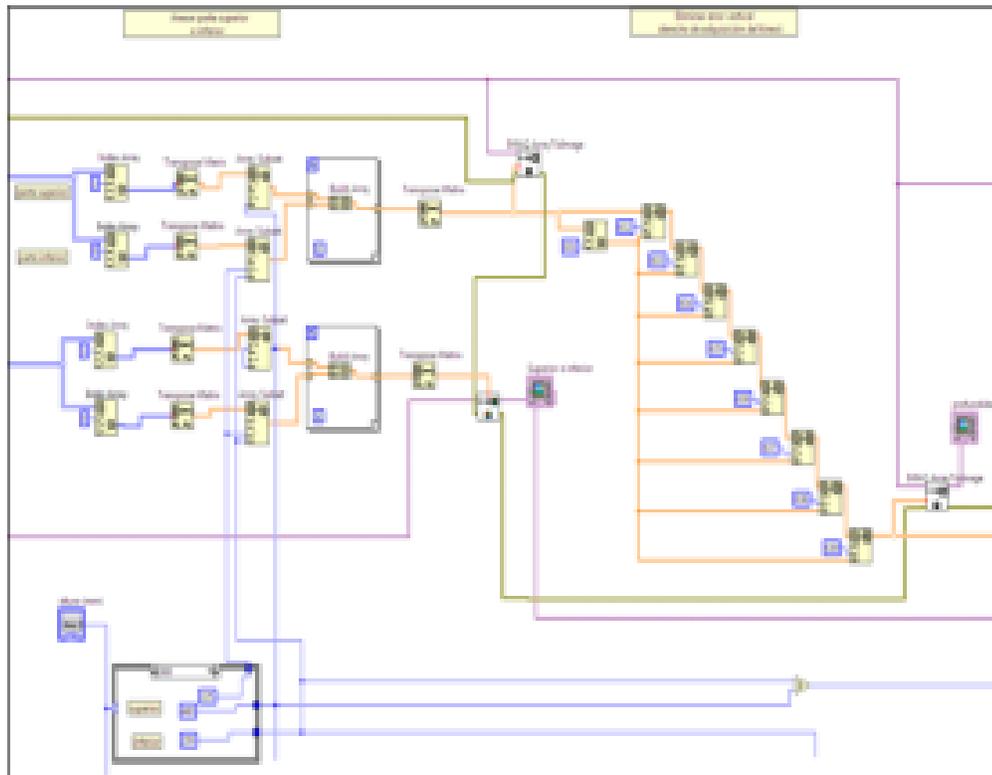


Figura 2.55: Transformación a una profundidad real

## 2.5.3 Etapa 3: Eliminar errores de adquisición

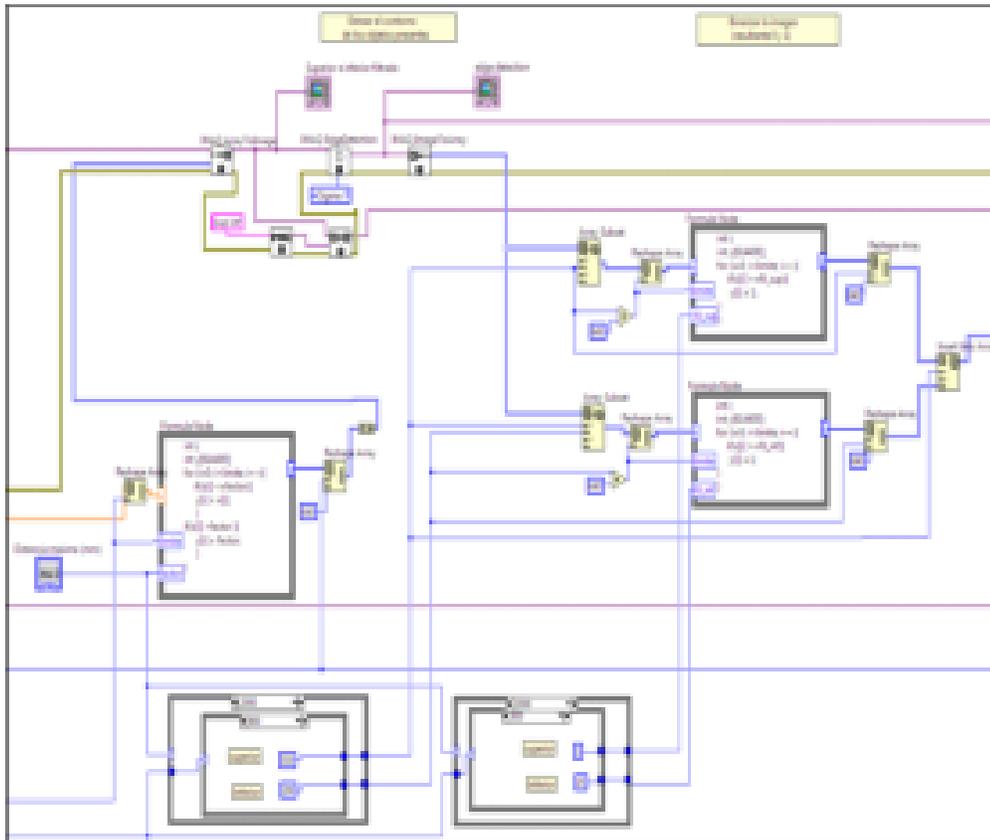
La imagen superior e inferior se visualizan en una sola imagen, tanto para la de color como para la de profundidad, cada imagen tiene una dimensión de 640 x 480. Además se elimina el error de adquisición del Kinect, ya que ocho columnas no son sensadas y serán reemplazadas por la última columna adquirida (Figura 2.56).



**Figura 2.56: Eliminar errores de adquisición**

#### **2.5.4 Etapa 4: Filtrar imagen por distancia y detección de bordes**

Se filtra la imagen según la profundidad, se aplica un filtro sigma para la detección de bordes y se binariza la imagen. En la Figura 2.57 se muestra el desarrollo de esta etapa del algoritmo.



**Figura 2.57: Filtrar imagen por distancia y detección de bordes**

### 2.5.5 Etapa 5: Segmentar los objetos

Se segmentan y contabilizan los objetos, en la Figura 2.58 se muestra el diagrama de bloques de esta etapa.

**Figura 2.58: Detección de objetos presentes**

## 2.5.6 Etapa 6: Establecer límites de posiciones

Se separa las coordenadas del centro y y de las esquinas para un posterior tratamiento, además en base al control del margen de división vertical, sobreponen a la imagen las líneas de división que delimitan los márgenes para establecer si un objeto está al centro, a la derecha o izquierda. En la Figura 2.59 se muestra el diagrama de bloques de esta etapa.

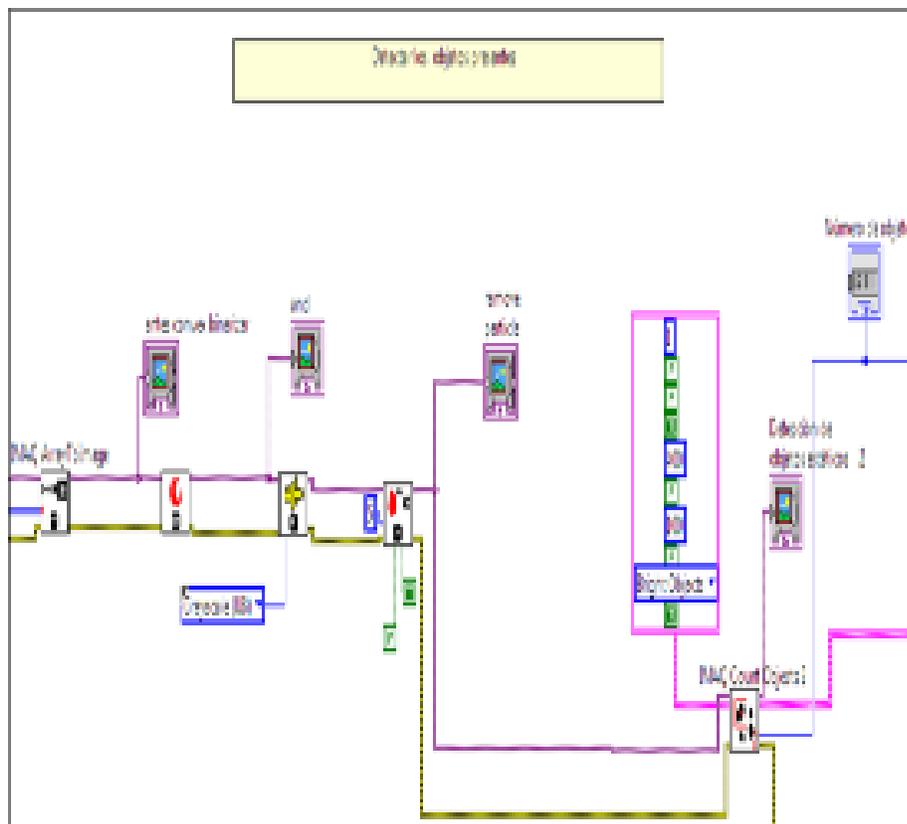
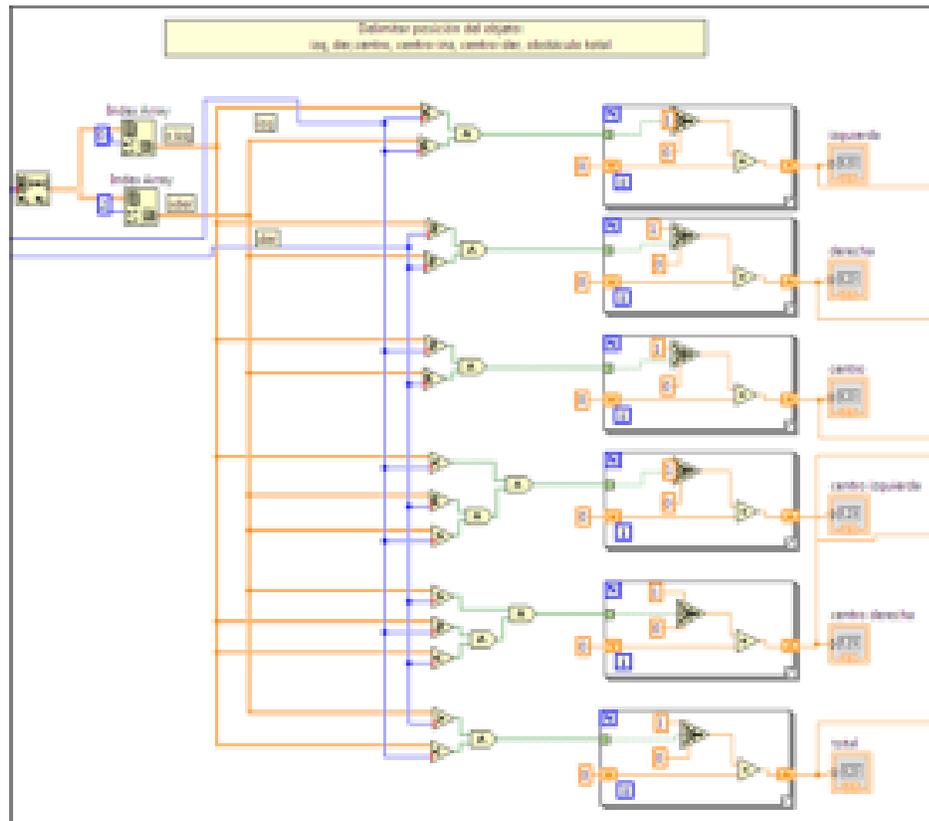


Figura 2.59: Establecer límites de posiciones

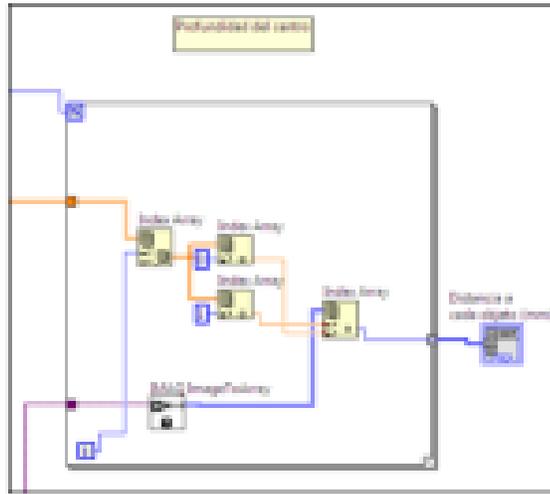
En base a las coordenadas de las esquinas se delimita la ubicación de los objetos como se indica en la Figura 2.60.



**Figura 2.60: Delimitación de la posición de los objetos**

### **2.5.7 Etapa 7: Determinar la distancia al centro del objeto**

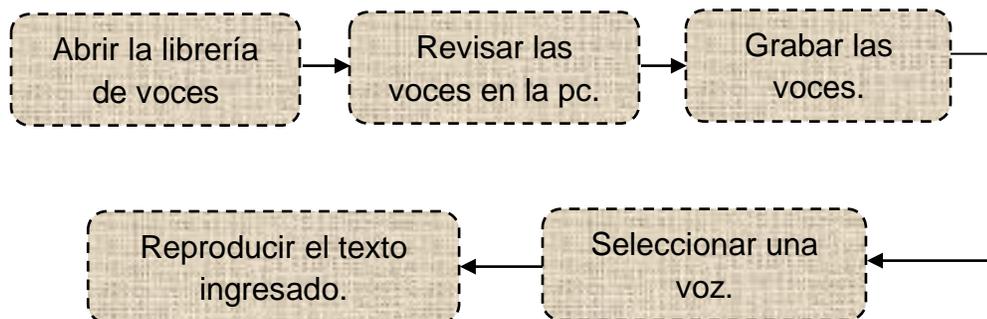
De acuerdo a la coordenada del centro obtenida en la etapa 6 se compara esta posición (x,y) con el arreglo que contiene los datos de profundidad, se extrae el valor y se visualiza con un indicador. (Figura 2.61).



**Figura 2.61: Profundidad al centro de cada objeto**

### 2.5.8 Etapa 8: Texto a voz

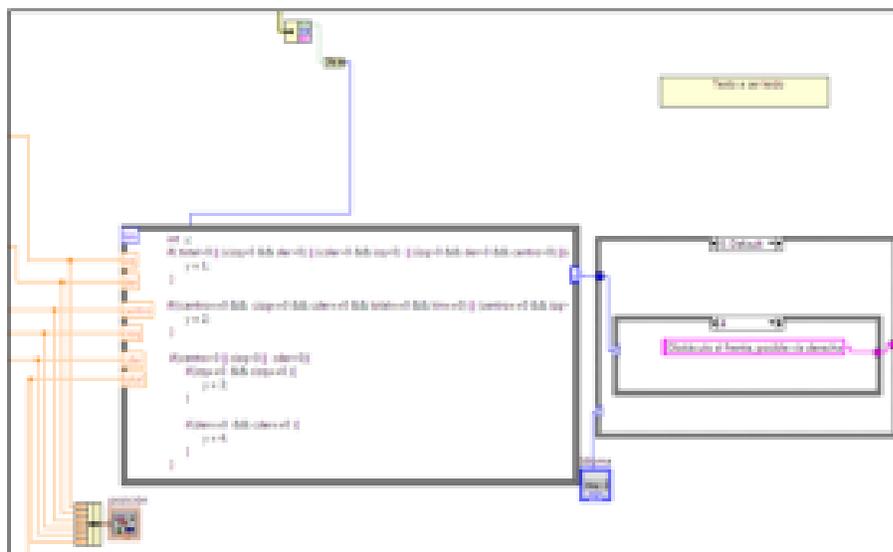
El programa a describirse a continuación está basado en uno encontrado en el foro de National Instrument de Labview<sup>85</sup>. El procedimiento para convertir un texto escrito en LabVIEW en voz audible se sintetiza en el diagrama de bloques de la Figura 2.62 que se ilustra a continuación:



**Figura 2.62: Diagrama de bloques para convertir texto a voz**

<sup>85</sup> "Text to speech". [En línea]. Recuperado el 08 de febrero del 2013, Disponible en Web: <<https://decibel.ni.com/content/docs/DOC-2263>>

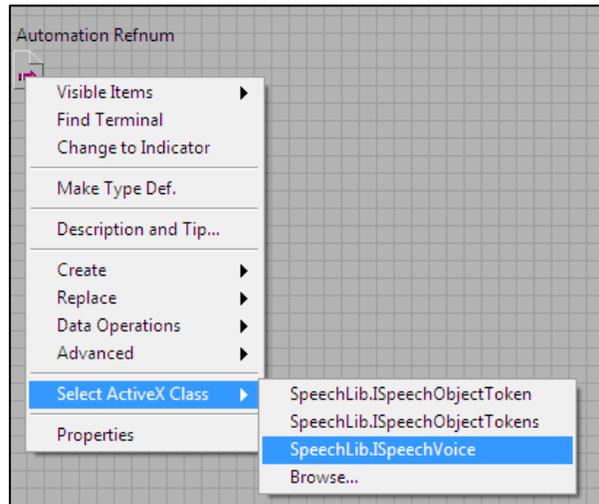
Se utiliza controles ActiveX y nodos de propiedad para convertir un texto escrito en LabVIEW en voz audible. Previamente se ha creado el texto a ser hablado que estará disponible en español e inglés, el texto que se reproducirá será: obstáculo total, vía libre, obstáculo al frente posible vía izquierda, obstáculo al frente posible vía derecha o existe un error (Figura 2.63).



**Figura 2. 63: Texto que va a ser reproducido**

Los pasos a seguir son:

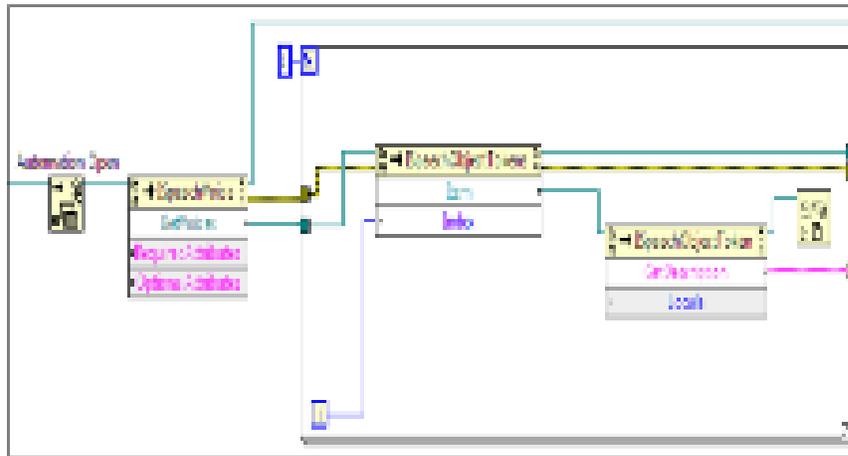
1. Se abre la librería de voces con un Automation Refnum desde el panel frontal y se selecciona la clase ActiveX SpeechLib.ISpeechVoice. (Figura 2.64).



**Figura 2.64: Selección de la clase SpeechLib.ISpeechVoice**

2. Se emplea un Automation Open y un Invoke Node para revisar cuantas voces hay disponibles en la librería de la computadora (0 a 2 por defecto). (Figura 2.65)

Para esto se emplea un Automation Open y Close Reference de los controles ActiveX, se une el Automation Refnum creado en el paso 1 al terminal del mismo nombre del Automation Open. Se crean el método Get Voices de la clase SpeechLib.ISpeechVoice y los métodos Item y GetDescription de la clase SpeechLib.ISpeechObjectTokens.



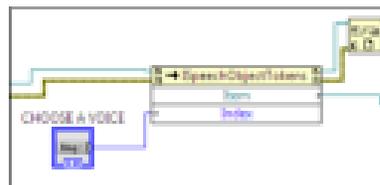
**Figura 2.65: Revisar las voces disponibles en la pc**

3. Las voces disponibles se graban en el botón Choose a voice.

Para esto se emplea un Menu Ring (Controls > Modern > Ring & Enum > Menu Ring) denominarlo Choose a voice, a este se le da la propiedad de Strings[].

4. Se selecciona una voz usando el mismo botón (Figura 2.66).

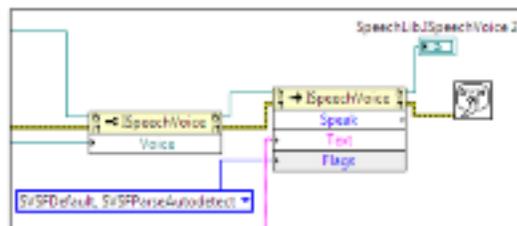
Con este propósito se selecciona el método Item de la Clase SpeechLib.ISpeechObjectTokens se une el terminal Index al menú creado en el paso 3 y se une el terminal de Reference Out a un Close Reference.



**Figura 2.66: Selección de una de las voces disponibles**

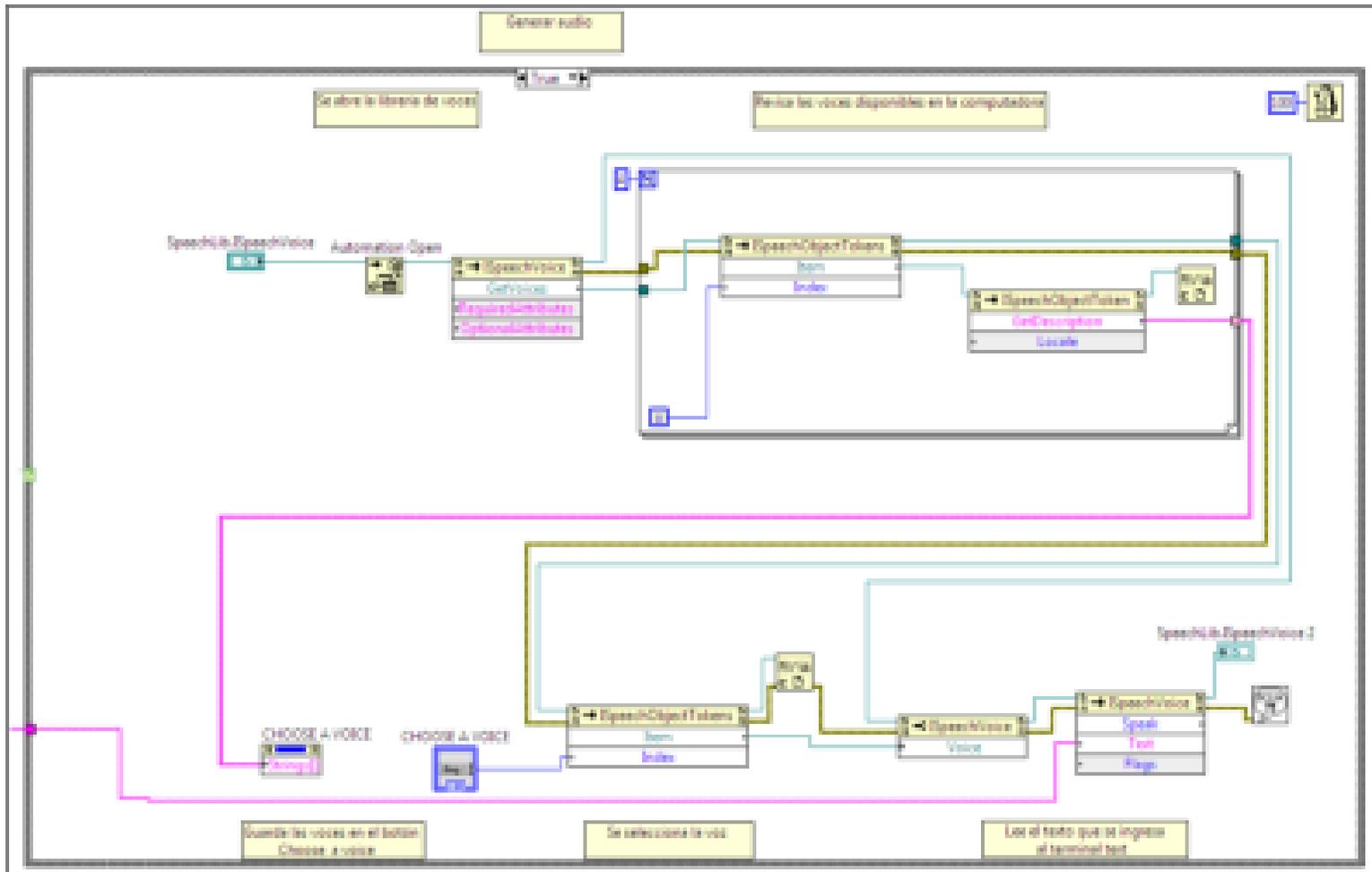
5. Finalmente se lee el texto que esté en el terminal Text del Nodo ISpeechVoice (Figura 2.67).

Para este fin se crea el método Voice de la clase SpeechLib.ISpeechVoice Class en base al Automation Refnum creado en el paso 1. Además se crea el método Speak de la misma clase. Se crea otro Automation Refnum para la referencia de salida del método Speak y se lo cambia a escritura (Change to Write). Se anexa un elemento de error simple.



**Figura 2.67: Reproducción del texto**

En la Figura 2.68 se muestra el programa completo de esta etapa. Tras esta etapa, se finaliza el Kinect.



**Figura 2.68: Programa completo para la Interface texto a voz**

## 2.5.9 Etapa 9: Párametros de entrada y salida

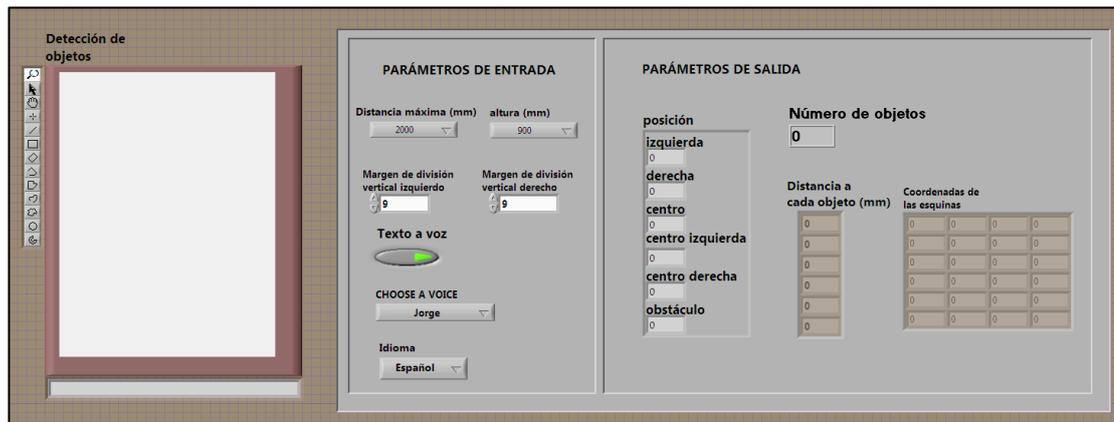
El algoritmo tiene las siguientes entradas:

- Distancia máxima a ser detectada en milímetros por medio de un control, con pasos de 10 centímetros.
- Margen de división de la imagen en grados para indicar desde el centro cuantos grados de tolerancia hay hacia la izquierda y derecha para considerar ese espacio como el centro de la trayectoria.
- Un control booleano que indicará si se desea emplear la etapa de convertir texto a voz.
- Un control para seleccionar de las voces disponibles en la pc, cuál se empleará.
- Un control para seleccionar el idioma inglés o español.

El algoritmo tiene las siguientes salidas:

- Un indicador numérico que muestra el número de obstáculos detectados.
- Un arreglo con la distancia de cada objeto a su centro.
- Un arreglo con las coordenadas de las esquinas de cada objeto en la imagen.
- Una imagen con el resultado de ejecutar el programa.
- Un grupo de indicadores con la información que indica si los objetos están a la izquierda, derecha, centro, centro izquierda, centro derecha y obstáculo total.

Las entradas y salidas se muestran en la Figura 2.69.



**Figura 2.69: Parámetros de entrada y salida**

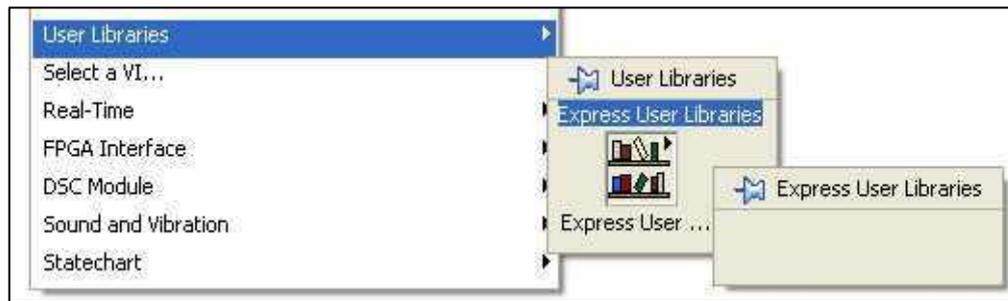
## 2.6 INSERTAR EL VI EN EL DIAGRAMA DE BLOQUES

En Labview, la paleta de Express User Libraries<sup>86</sup> aparece en la paleta de funciones solo si el VI ha sido guardado en el directorio labview\user.lib\\_express.

Para visualizar el vi en la paleta de funciones (Figura 2.70) se sigue los siguientes pasos:

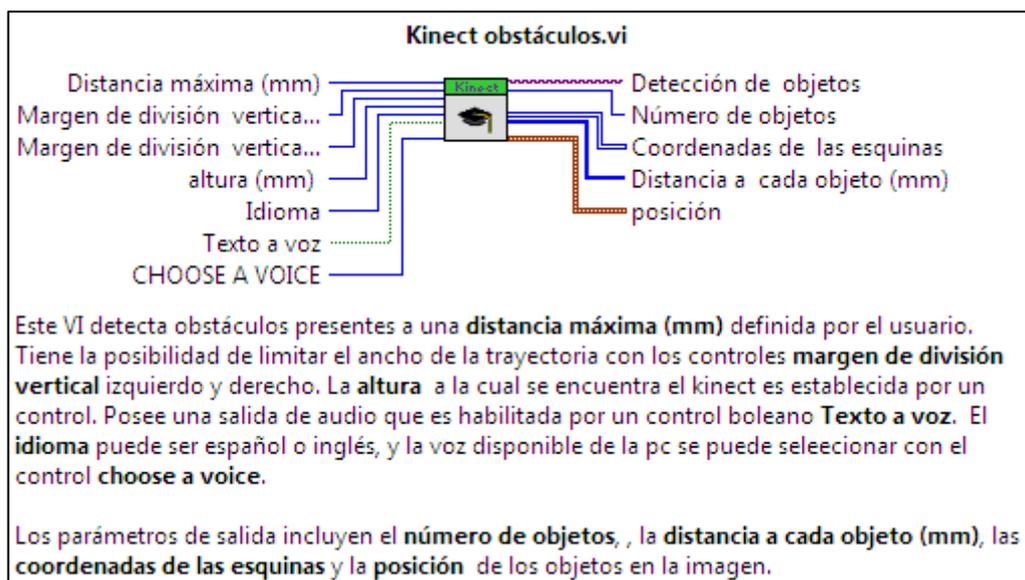
1. Se crea un nuevo VI.
2. Se guarda el VI en el directorio labview\user.lib\\_express.
3. Se reinicia LabVIEW.
4. En el diagrama de bloques, se navega en: Functions > User > Libraries > Express User Libraries. Se puede notar que el VI que se guardó ahora aparece en esta paleta.

<sup>86</sup> "Express user libraries". [En línea]. Recuperado el 26 de febrero del 2013, Disponible en Web: <http://digital.ni.com/public.nsf/allkb/7F0027314F936E82862570F10083C0F3>



**Figura 2.70: Localización del VI en la paleta de funciones**

En la Figura 2.71 se indica el resultado final del VI.



**Figura 2.71: Ícono en Labview listo para ser utilizado**

## **CAPÍTULO 3**

### **PRUEBAS Y ANÁLISIS DE RESULTADOS**

El objetivo principal del proyecto es el diseño e implementación de un algoritmo de detección de obstáculos. Con la finalidad que este objetivo se cumpla se procede a realizar los siguientes escenarios de pruebas para determinar el correcto funcionamiento del algoritmo.

#### **3.1 Pruebas Experimentales**

La prueba de Adquisición de la imagen RGB y de profundidad consiste en obtener las imágenes por separado tanto de la parte superior e inferior con el objetivo de comprobar el correcto funcionamiento de esta etapa del algoritmo.

En la prueba de eliminación de errores de adquisición se prueba que el efecto de espejo de la imagen proporcionada sea corregido y además que el error de adquisición en las últimas columnas a la derecha sea suprimido.

En la prueba del filtro en profundidad se comprueba que el filtro funcione correctamente eliminando todo pixel que no esté dentro del rango de distancia especificado.

Después de estas pruebas previas se toman varias muestras de diferente tipo de objetos para su análisis

En la prueba de la localización de los objetos se comprueba que el algoritmo pueda distinguir que un objeto esté al centro, izquierda, derecha, centro izquierda, centro derecha o sea un obstáculo total.

En la prueba de objetos, según la forma se muestrean objetos circulares, rectangulares, cilíndricos y otras formas y se analiza si el algoritmo es capaz de segmentarlos.

En la prueba del tamaño de los objetos, se colocan objetos pequeños y que progresivamente aumentan su tamaño, así como objetos grandes para verificar que todos sean detectados.

En la prueba de distancia se colocan a diferente profundidad los objetos para que todos sean reconocidos.

En la prueba del material del que está hecho el objeto, se colocan varios objetos de cuero, cerámica, plástico, cartón y tela, para analizar la respuesta del algoritmo ante esta variante.

Los objetos además de encontrarse en el piso o a los laterales, también puede encontrarse en la parte superior suspendido o Colgante, ante esto se prueba que el algoritmo pueda segmentar este tipo de objetos.

Debido a las limitaciones del Kinect especificadas en el capítulo 1, se prueban objetos transparentes, reflectantes y negros para notar la respuesta del algoritmo.

Se colocan objetos unidos físicamente y visualmente respecto al ángulo de visión del Kinect para analizar el resultado de la detección de obstáculos.

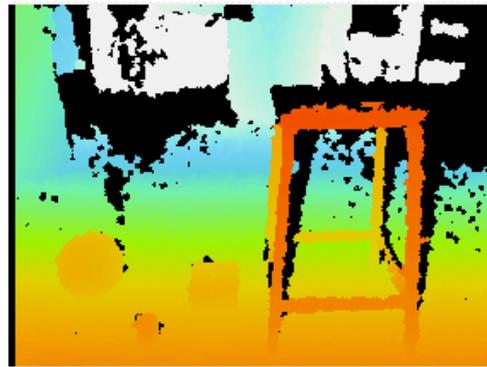
Todo el proceso de detección se realiza sobre la imagen de profundidad del Kinect, y solo para visualizar de mejor manera la detección de los objetos, se coloca este resultado sobre la imagen de color, advirtiéndose que existe un pequeño desfase entre las dos imágenes.

### **3.1.1 Prueba de adquisición**

Las imágenes adquiridas de color y profundidad tanto de la parte superior como inferior se indican en la Figura 3.1.



a) Imagen de color inferior



b) Imagen de profundidad inferior



c) imagen de color superior



d) imagen de profundidad superior

### Figura 3.1: Prueba de adquisición

Se distingue que las cuatro imágenes son adquiridas correctamente, notando las zonas que el Kinect no puede distinguir (zonas negras) por las diferentes limitaciones del sensor y de la escena.

#### 3.1.2 Prueba para eliminar errores de adquisición

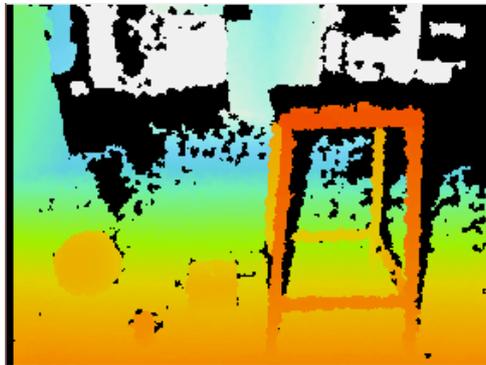
Debido al efecto de espejo se obtiene la simetría de las imágenes (Figura 3.2), se unen las partes superior e inferior y se elimina el error de adquisición lateral en la imagen de profundidad (Figura 3.3).



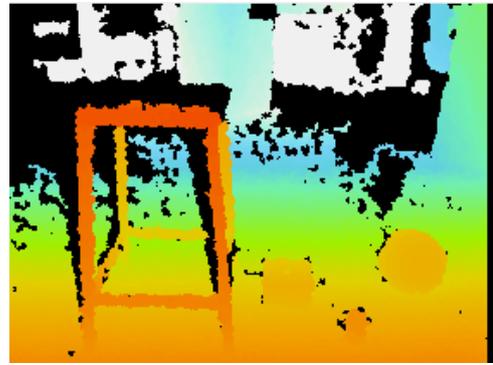
a) Imagen de color sin simetría



b) Imagen de color con simetría



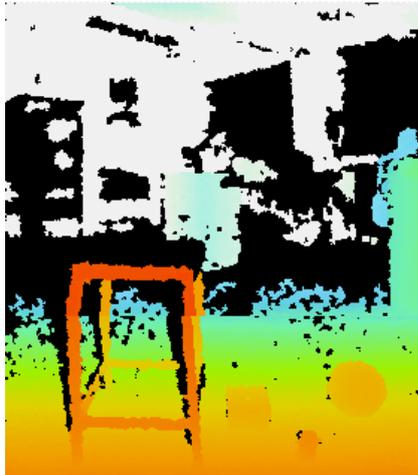
c) imagen de profundidad sin simetría



d) imagen de profundidad con simetría

**Figura 3.2: Simetría de las imágenes**

Se nota que el efecto de espejo ha sido corregido satisfactoriamente.

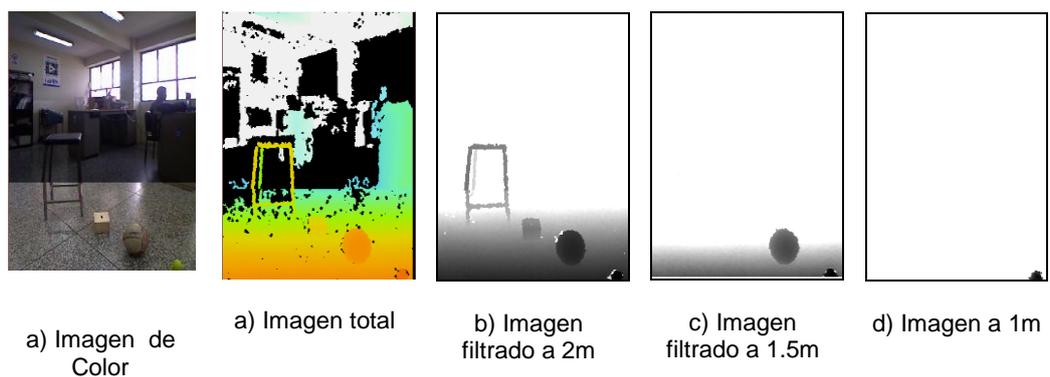


**Figura 3.3: Eliminación de error lateral en la imagen de profundidad**

La franja lateral con valores de la derecha se puede apreciar que ha sido corregida.

### 3.1.3 Prueba del filtro en profundidad

Para que el proceso de segmentación se realice de forma eficiente, se toman solo parte de los datos que puede adquirir el sensor (aproximadamente 4 metros) a máximo 2 metros. Los efectos se indican en la Figura 3.4.



**Figura 3.4: Prueba del filtro en profundidad**

### 3.1.4 Prueba según la localización

La localización de un objeto puede ser: trayectoria (centro), izquierda, derecha, centro-izquierda, centro-derecha, obstáculo total, algunas de estas opciones son distinguidos en la Figura 3.5, Figura 3.6 y Figura 3.7

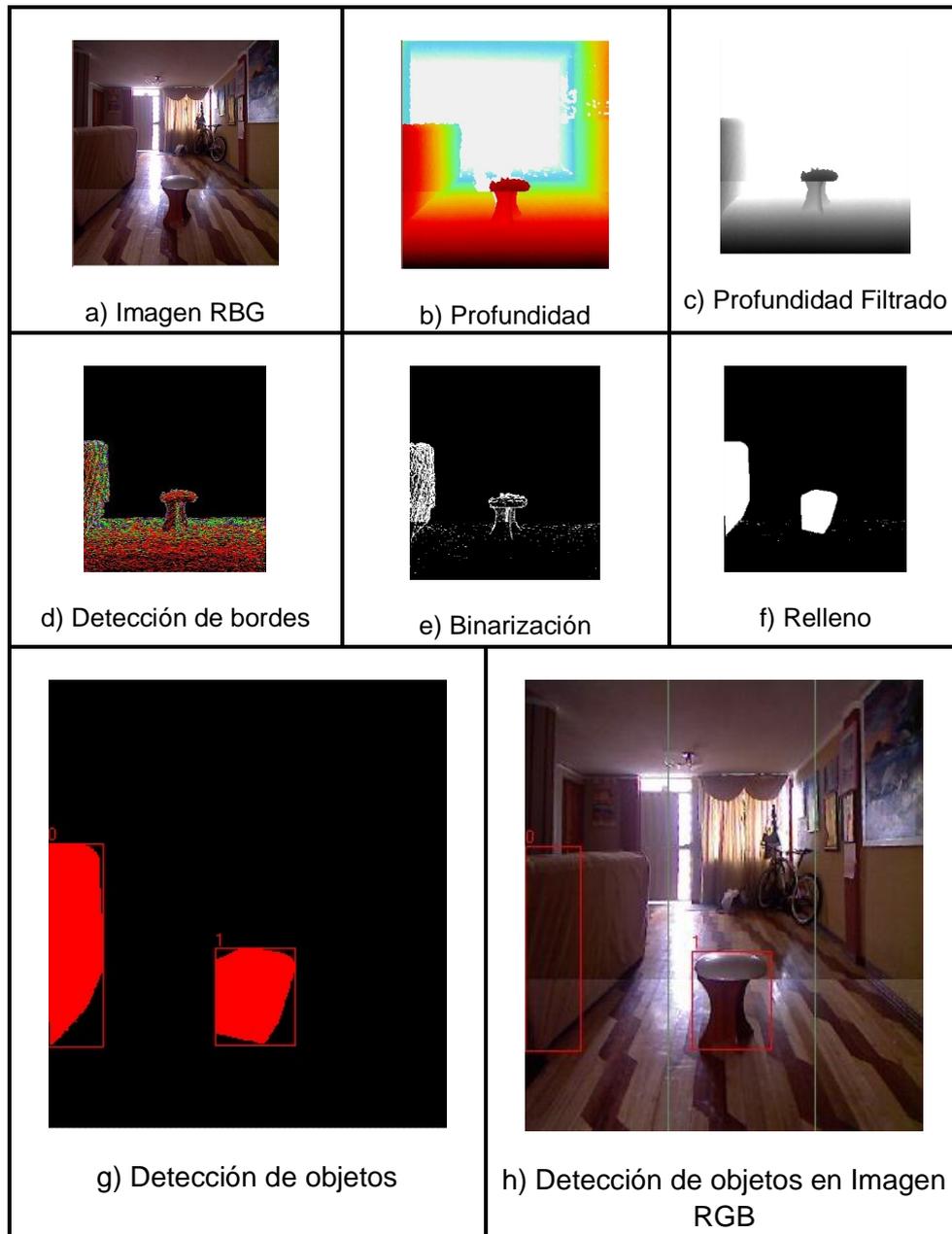
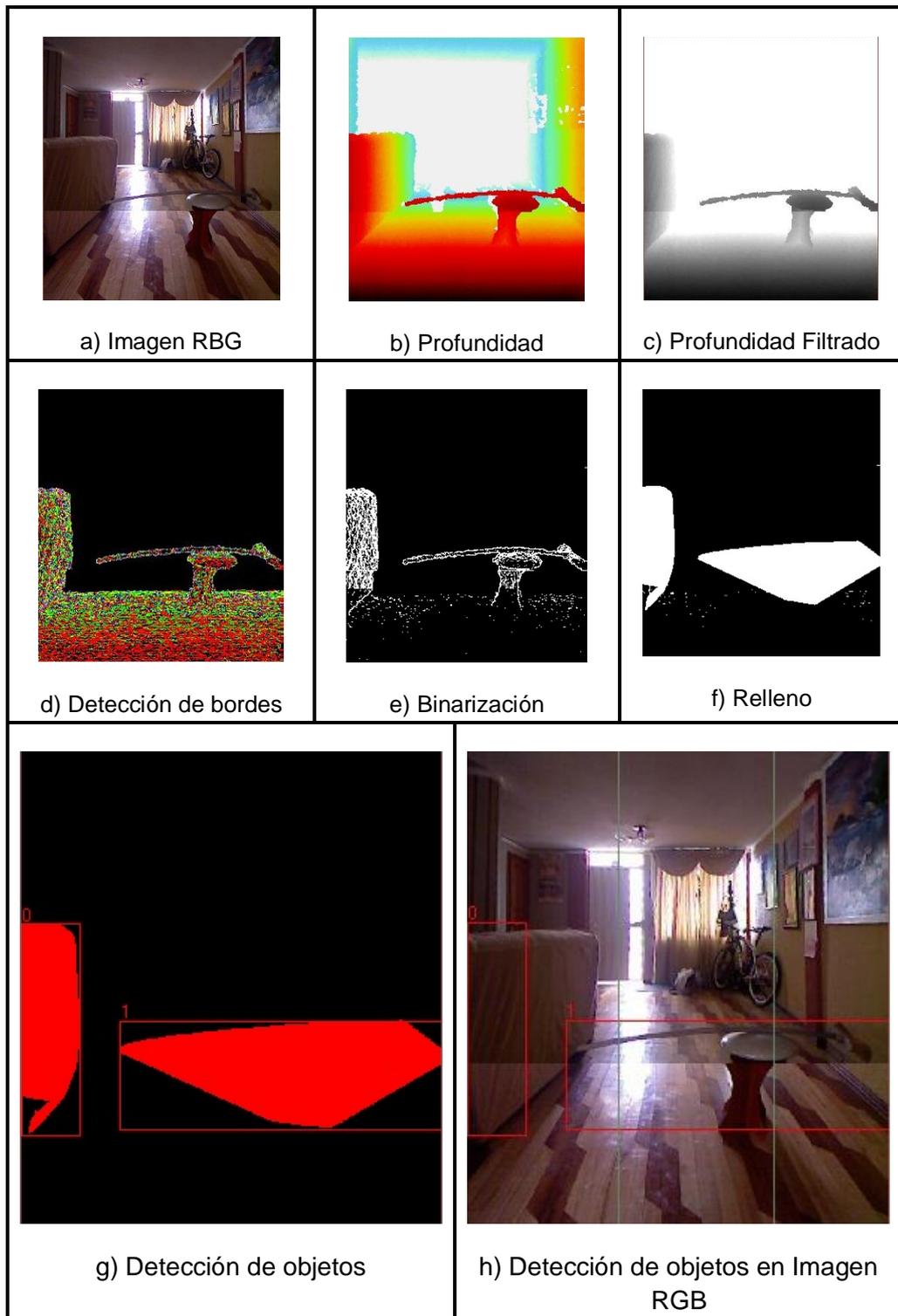
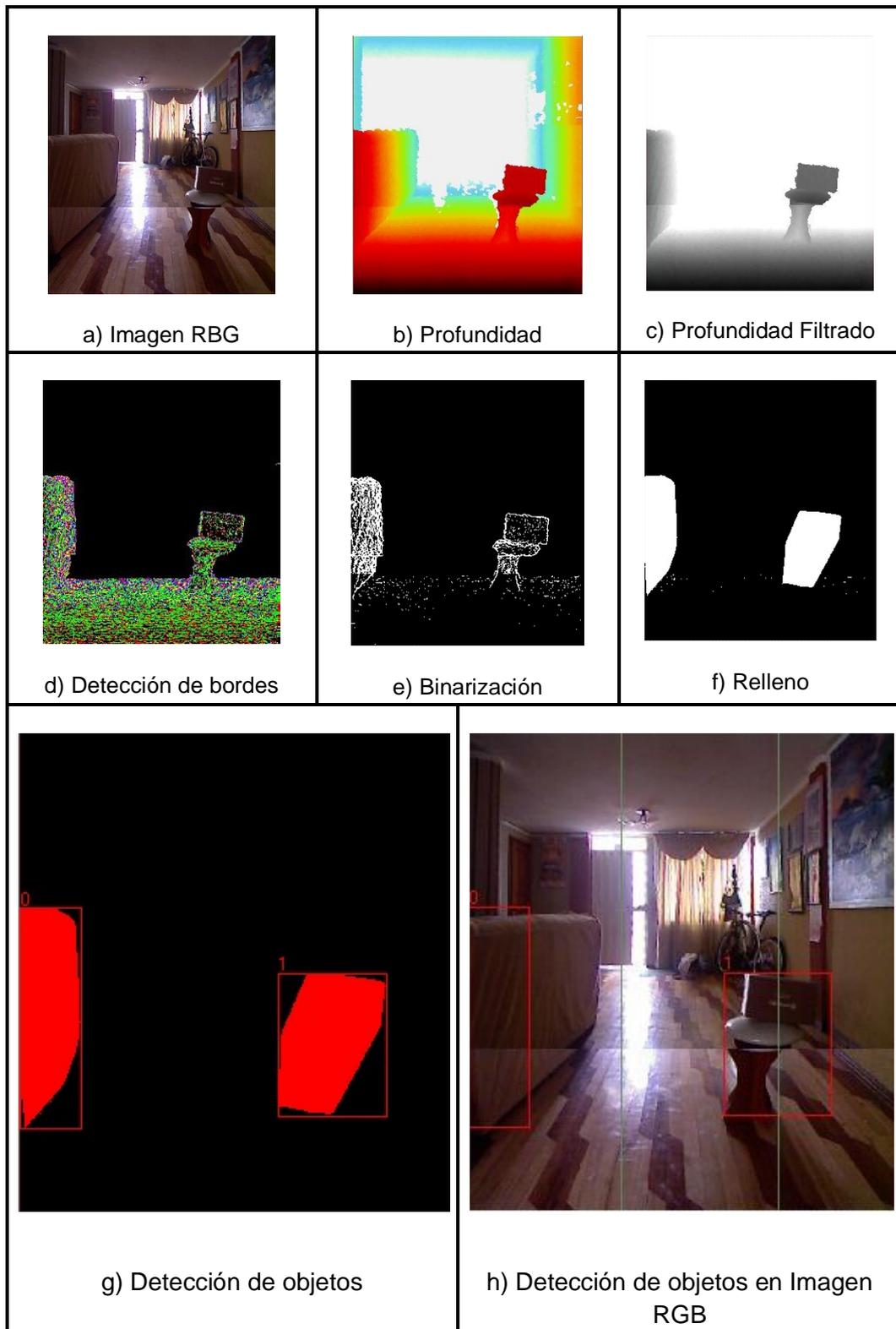


Figura 3.5: Prueba de localización centro e izquierda



**Figura 3.6: Prueba de localización obstáculo total e izquierda**



**Figura 3.7: Prueba de localización centro derecha e izquierda**

El resultado de esta prueba se resume en la Tabla 3.1.

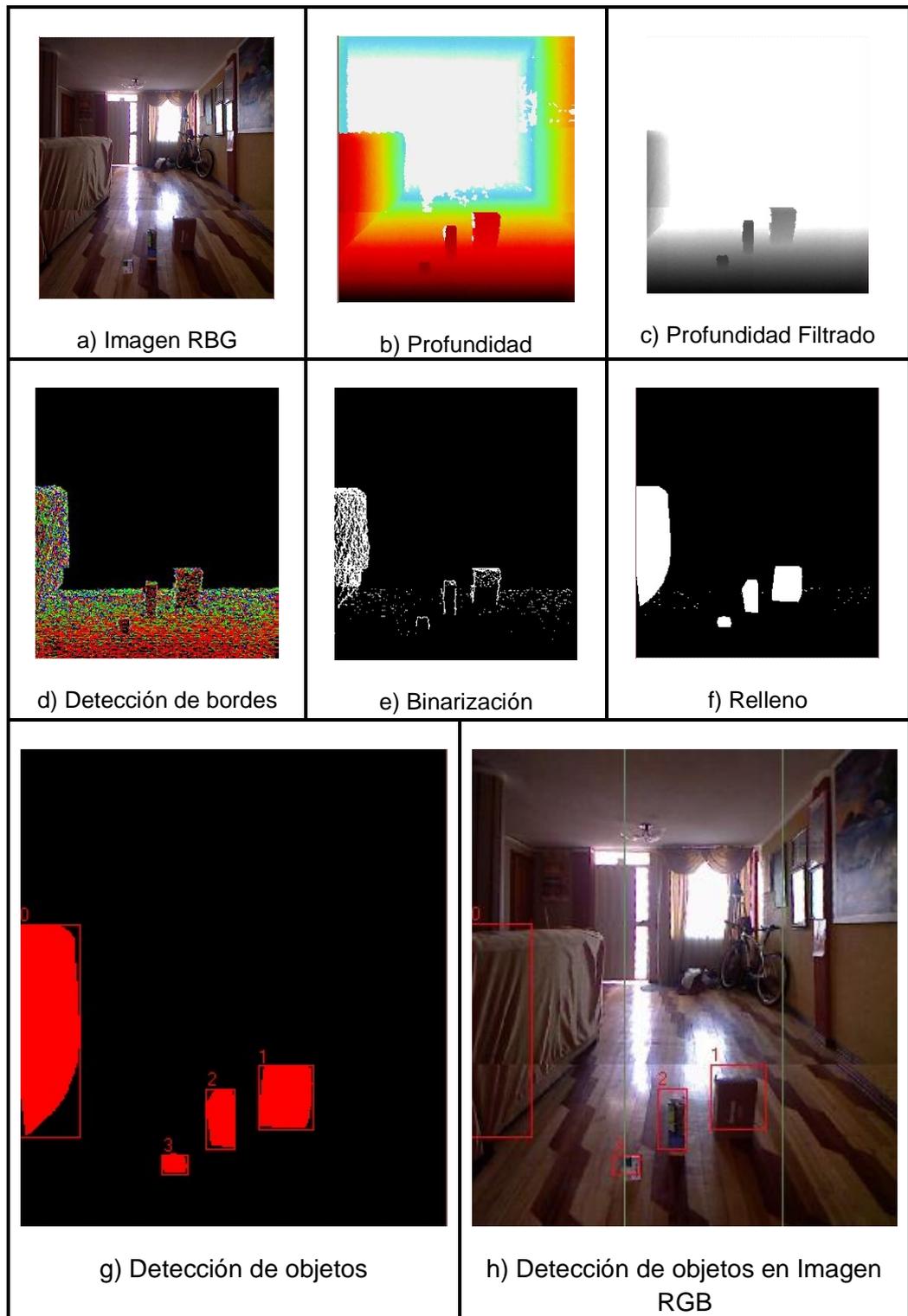
**Tabla 3.1: Resultado de la prueba de localización**

	<b>Detectado</b>	<b>No detectado</b>
<b>Trayectoria</b>	✓	
<b>Izquierda</b>	✓	
<b>Derecha</b>	✓	
<b>Centro-Izquierda</b>	✓	
<b>Centro-Derecha</b>	✓	
<b>Obstáculo Total</b>	✓	

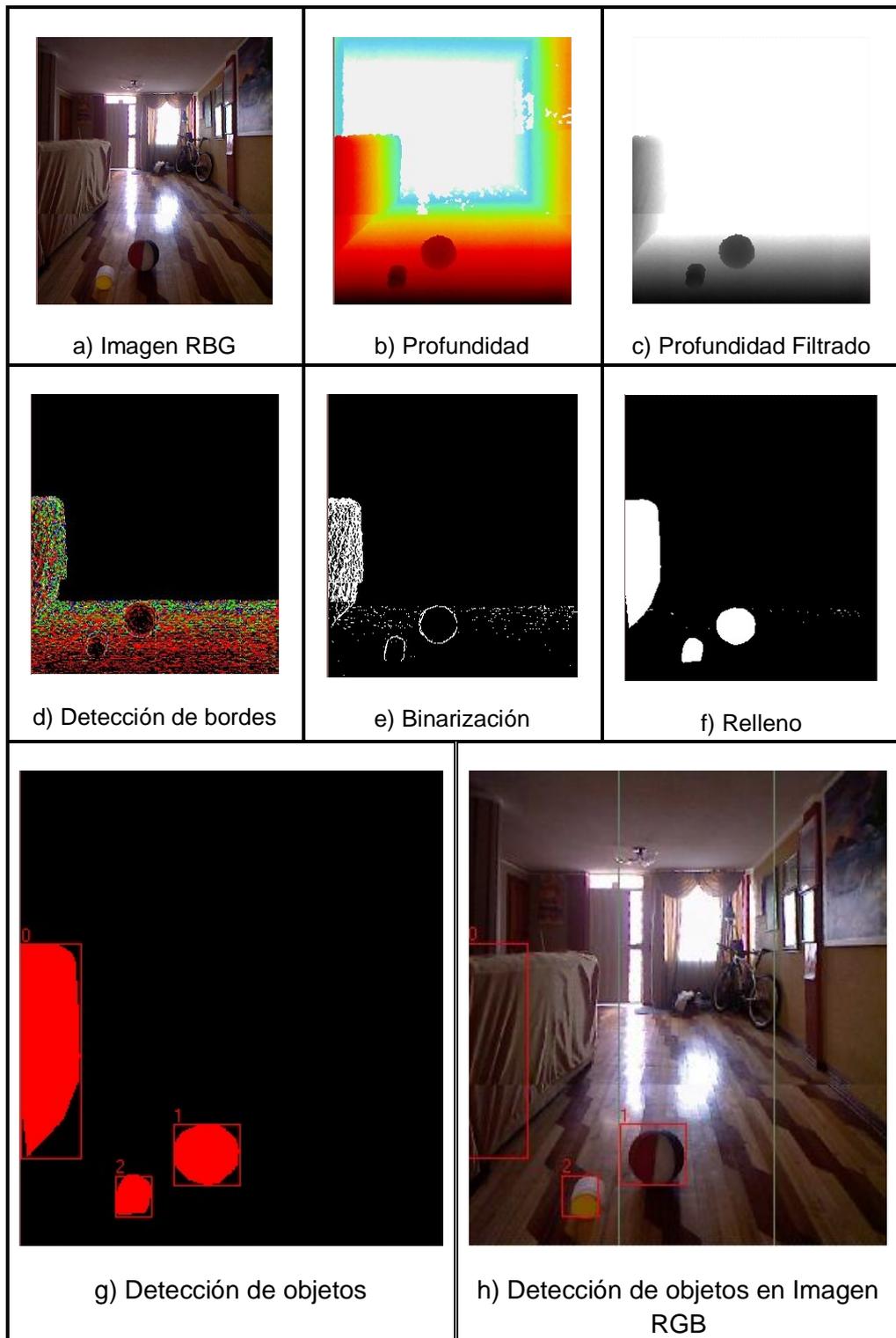
En las opciones de localización presentes, el algoritmo ha sido capaz de detectar a los objetos y distinguir su localización con un 100% de efectividad.

### **3.1.5 Prueba de forma de objetos**

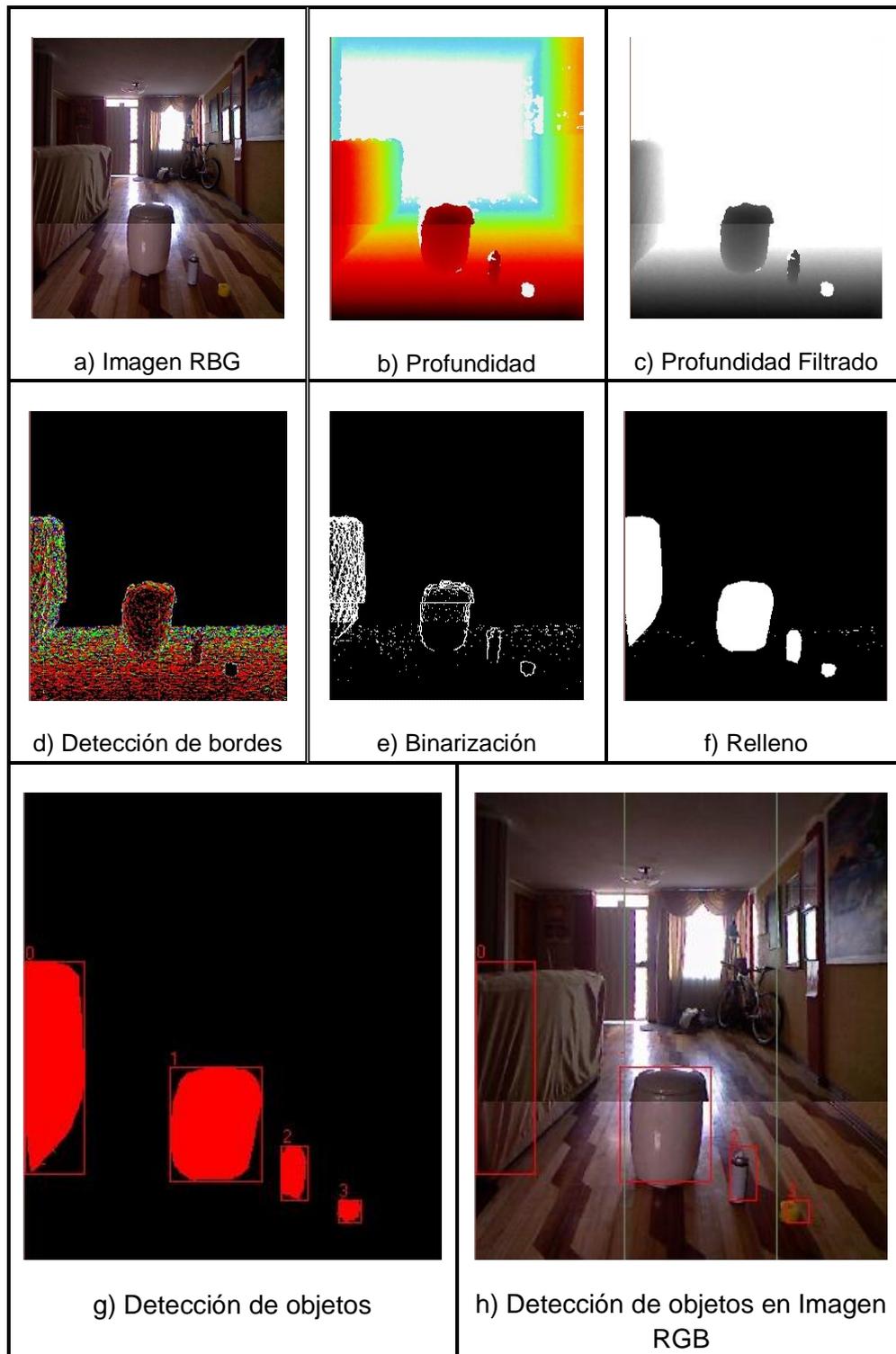
La respuesta del algoritmo es probada ante el parámetro de la forma de los objetos, para este fin se ponen a prueba objetos de tipo rectangular (Figura 3.8), circular (Figura 3.9), cilíndrico (Figura 3.10) y otros como triangular y elíptico (Figura 3.11).



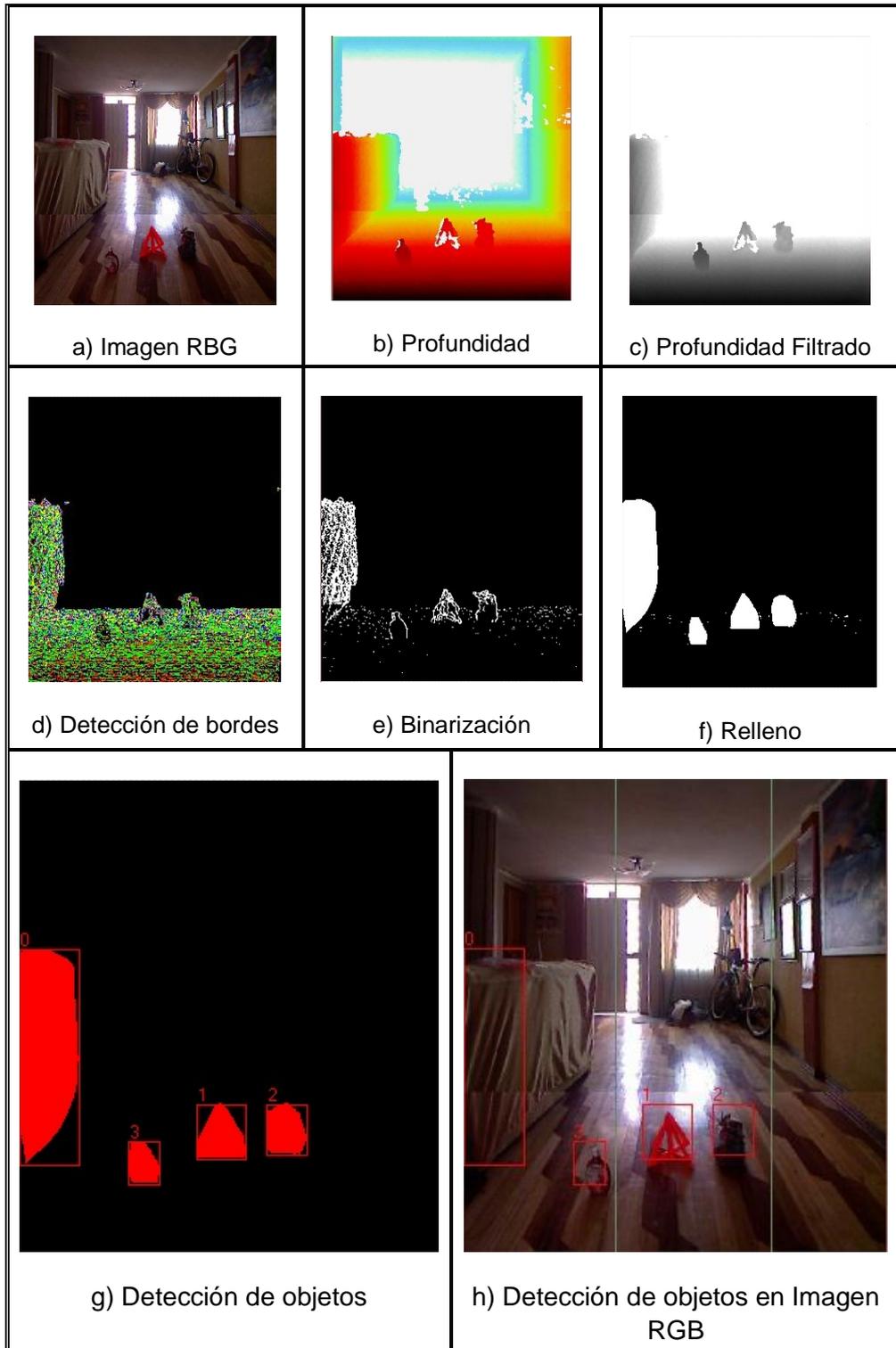
**Figura 3.8: Prueba de objetos rectangulares**



**Figura 3.9: Prueba de objetos circulares**



**Figura 3.10: Prueba de objetos cilíndricos**



**Figura 3.11: Prueba de otras formas de objetos**

El resultado de esta prueba se resume en la Tabla 3.2.

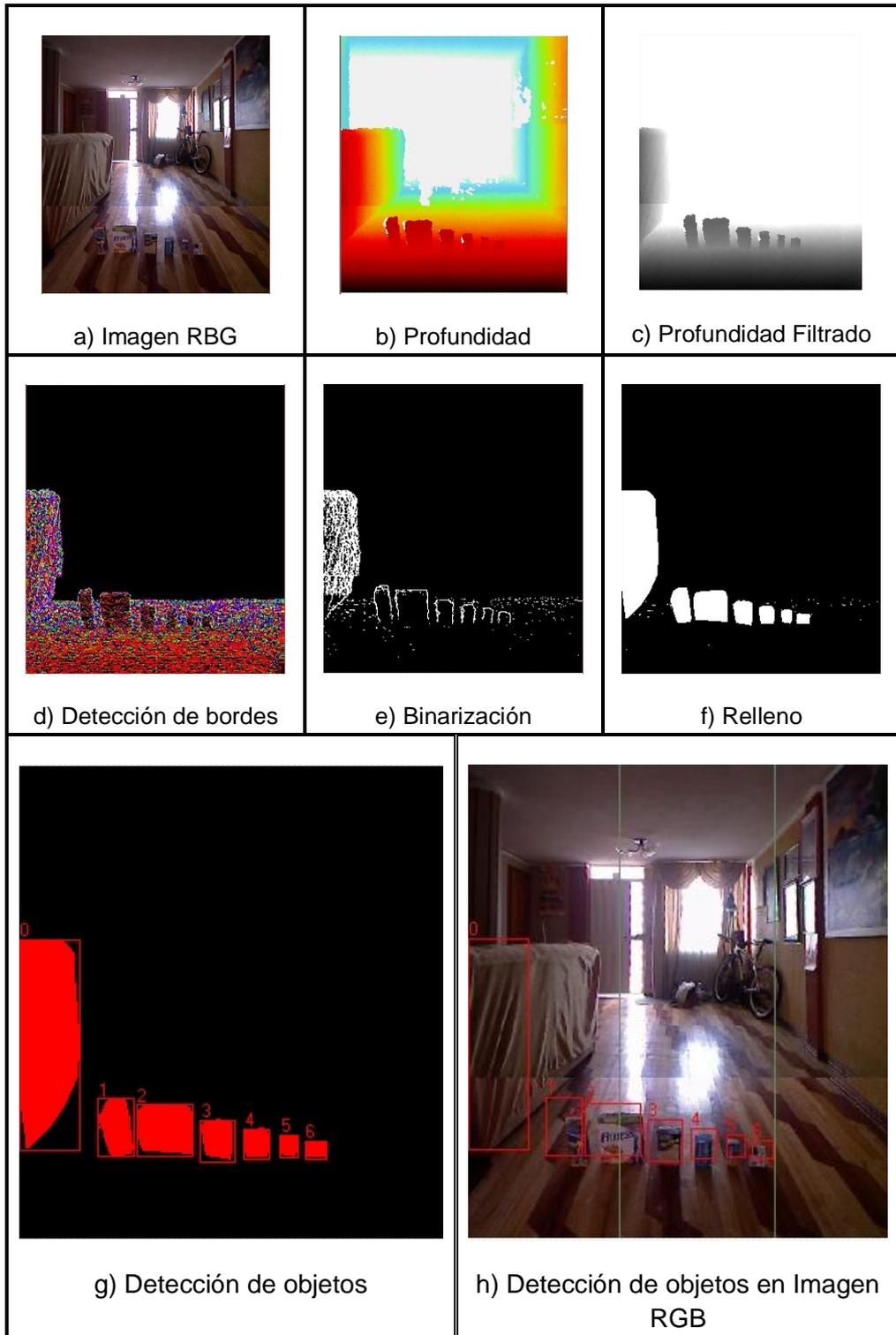
**Tabla 3.2: Resultado de la prueba de forma de objetos**

	<b>Objetos a ser Detectados</b>	<b>Objetos Detectados</b>
<b>Rectangular</b>	3	3
<b>Circular</b>	2	2
<b>Cilíndrico</b>	3	3
<b>Otras formas</b>	3	3

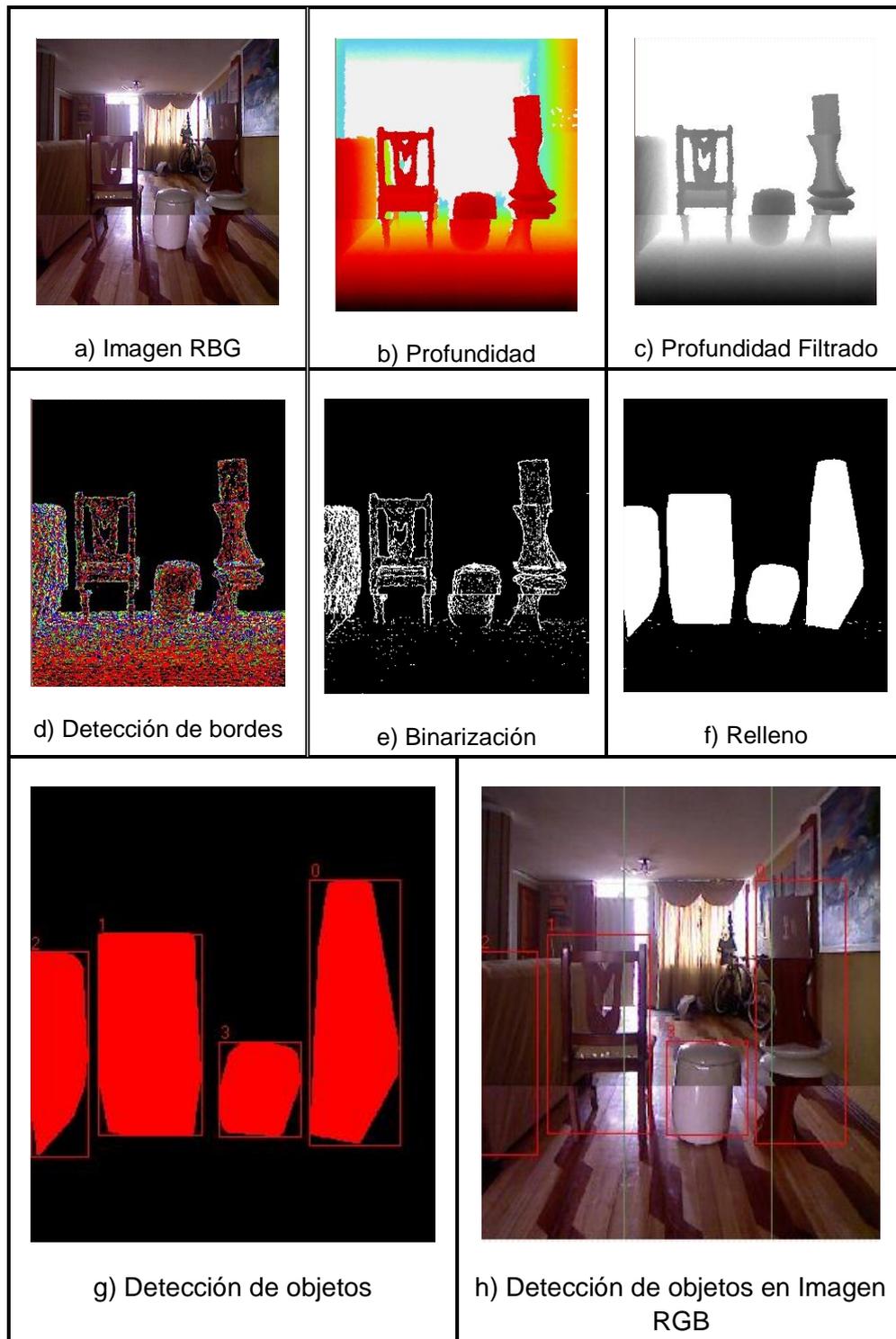
Se distingue que ante las formas presentadas en la prueba, la totalidad de objetos probados son detectados, con lo cual se verifica que la forma de los objetos no es una limitante del algoritmo y que éste el 100% efectivo en este aspecto.

### **3.1.6 Prueba de Tamaño**

Se realizan dos tipos de pruebas con varios objetos a la vez, para objetos pequeños y grandes como se indica en las Figura 3.12 y 3.13 respectivamente.



**Figura 3.12: Prueba de objetos pequeños**



**Figura 3.13: Prueba de objetos grandes**

El resultado de esta prueba se resume en la Tabla 3.3.

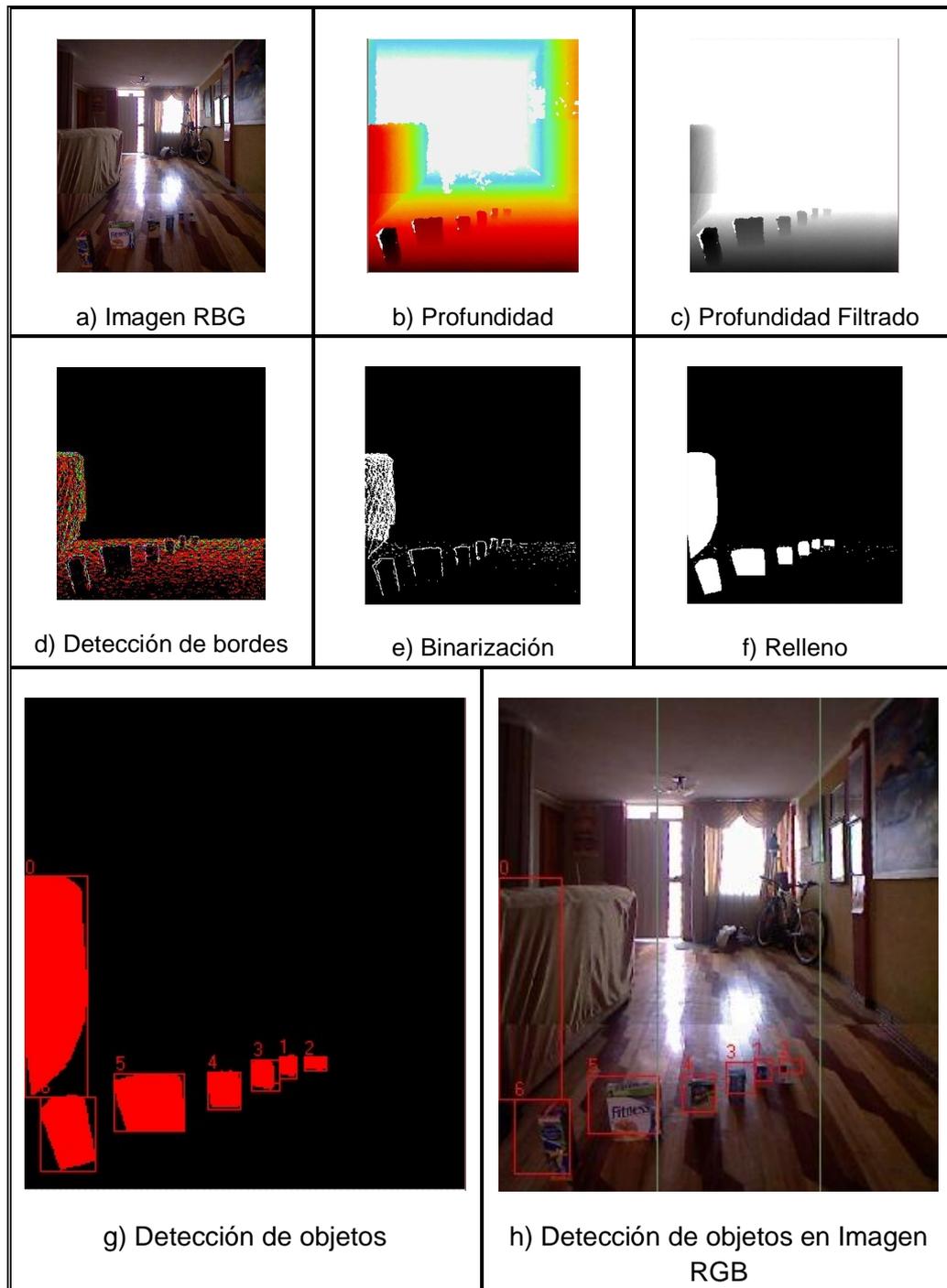
**Tabla 3.3: Resumen de la prueba de tamaño**

	<b>Objetos a ser Detectados</b>	<b>Objetos Detectados</b>
<b>Pequeños</b>	6	6
<b>Grandes</b>	3	3

La totalidad de los objetos presentes son detectados, tanto pequeños como grandes, además, en la prueba de objetos grandes, uno de ellos tiene orificios, lo cual no representa un problema ya que el objeto es segmentado completamente. La altura mínima de los objetos de prueba es de 10cm, ya que en el algoritmo se descartan objetos muy pequeños para que no se confundan con partículas o falsos objetos producto de la adquisición del Kinect.

### **3.1.7 Prueba de profundidad**

Se colocan varios objetos a diferente distancia del sensor Kinect, para probar la respuesta del algoritmo ante esta situación, el resultado es visualizado en la Figura 3.14.

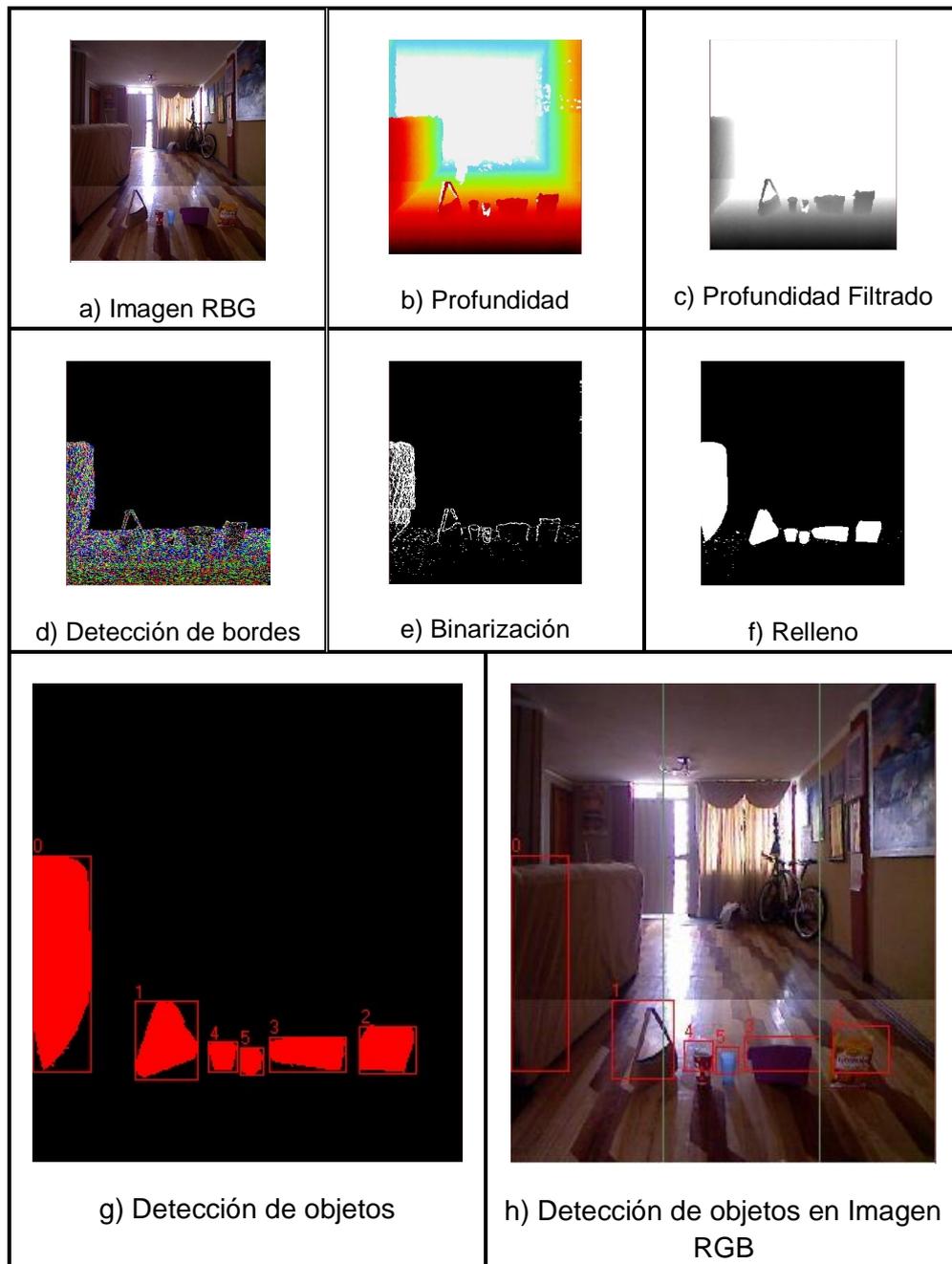


**Figura 3.14: Prueba de profundidad**

Se aprecia que son 7 objetos a ser detectados y todos son segmentados satisfactoriamente.

### 3.1.8 Prueba de materiales

La Figura 3.15 indica la prueba realizada empleando materiales como cuero, cerámica, plástico, tela y cartón.



**Figura 3.15: Prueba de materiales**

El resultado de esta prueba se resume en la Tabla 3.4.

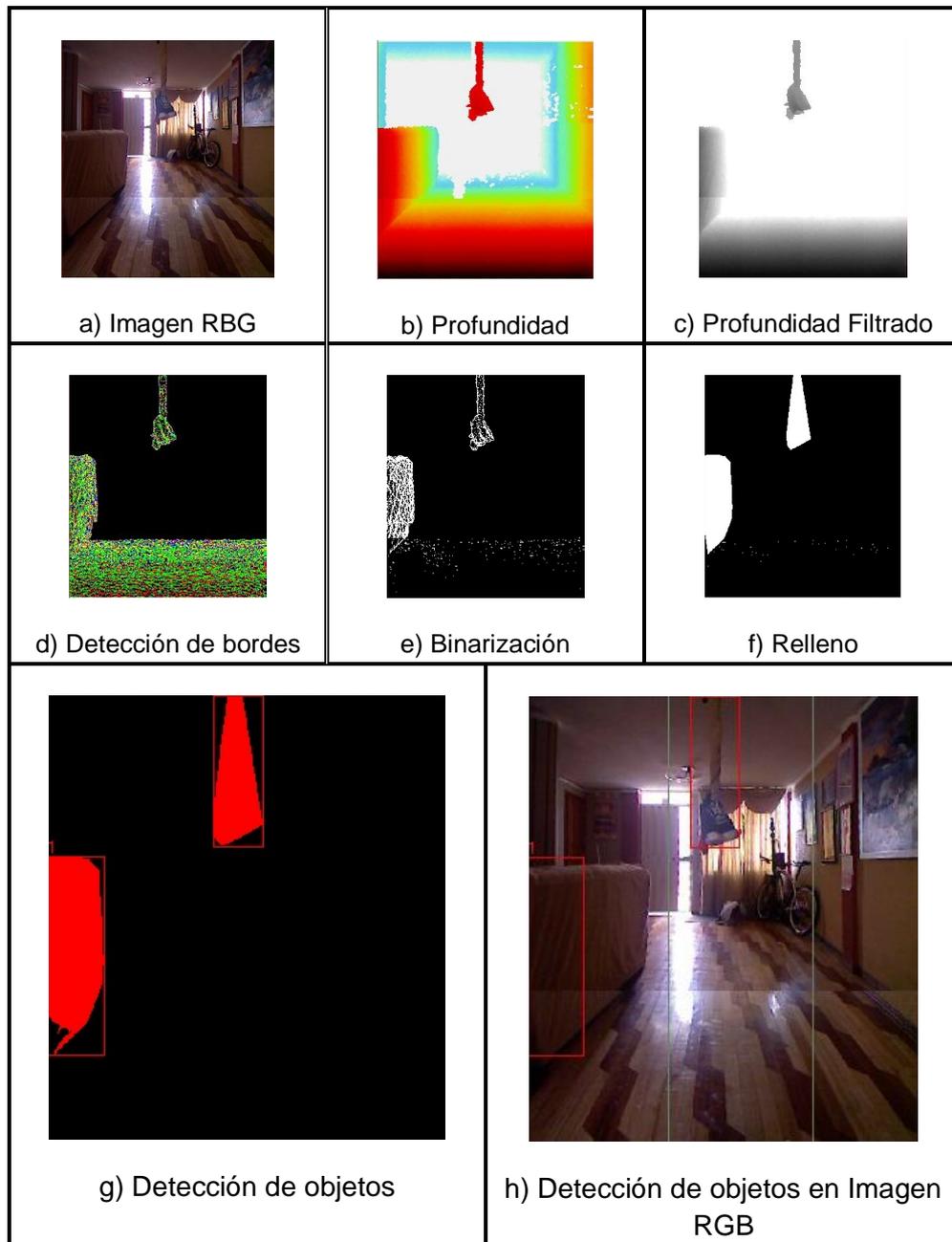
**Tabla 3.4: Resultado de la prueba de materiales**

	<b>Detectado</b>	<b>No detectado</b>
<b>Cuero</b>	✓	
<b>Cerámica</b>	✓	
<b>Plástico</b>	✓	
<b>Tela</b>	✓	
<b>Cartón</b>	✓	

Ante la tabla resumen se concluye que los materiales probados no representan una limitación para detectar al objeto, ya que tanto el de cuero, cerámica, plástico, tela y cartón fueron segmentados satisfactoriamente.

### **3.1.9 Prueba objetos colgantes**

Un objeto además de estar en el piso, también puede encontrarse en la parte superior, por ejemplo suspendida como una lámpara, para verificar el correcto funcionamiento del algoritmo, en la Figura 3.16 se muestra que el objeto colocado es detectado.



**Figura 3.16: Prueba de objeto colgante**

El objeto colocado en la parte superior de la escena es detectado de forma eficiente.

### 3.1.10 Prueba de objetos transparentes

Debido a la limitación del sensor, los objetos transparentes no se aprecian con efectividad (Figura 3.17). El algoritmo le asigna el valor máximo al pixel en esta situación por lo que la distancia a cada objeto no será la real.

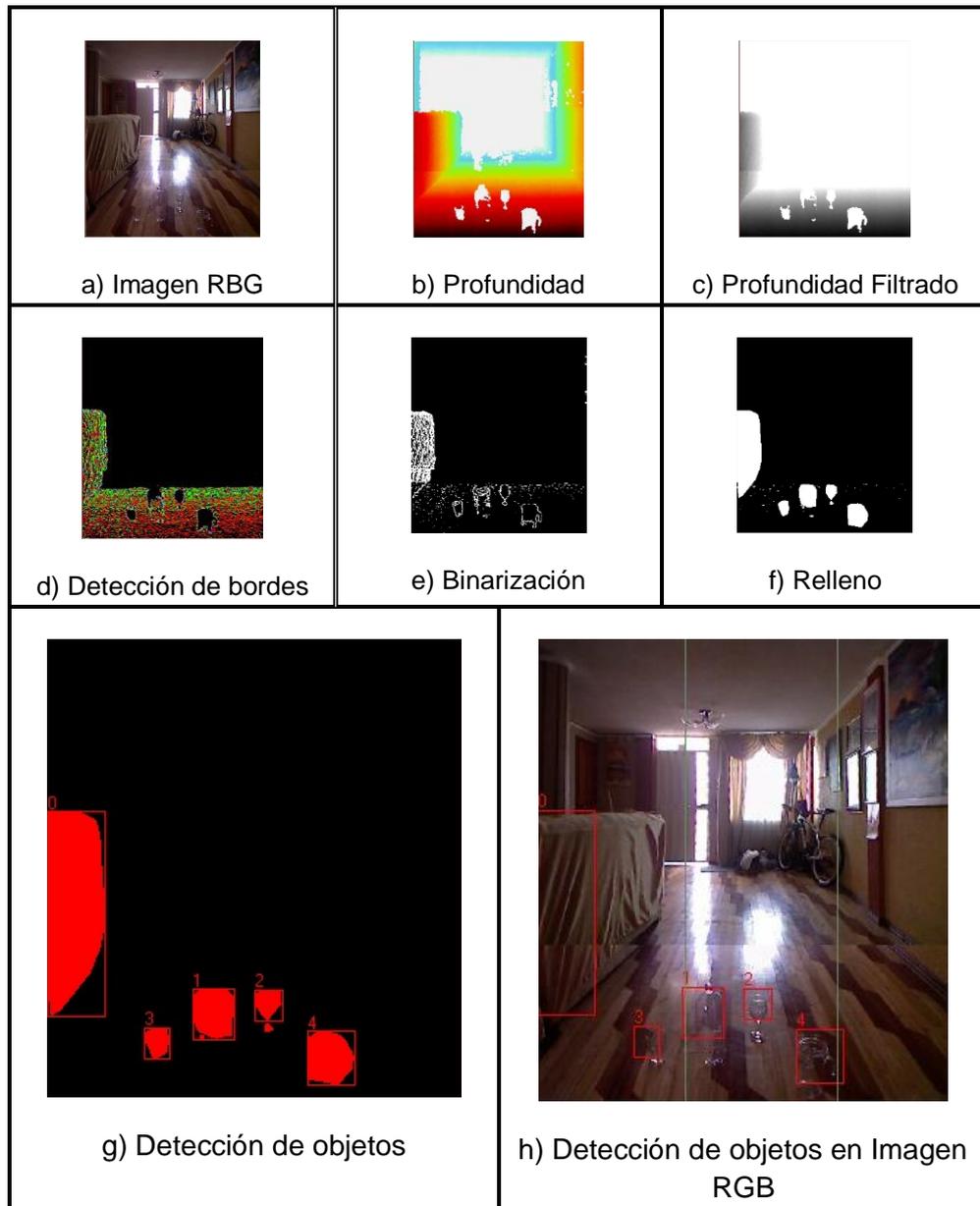


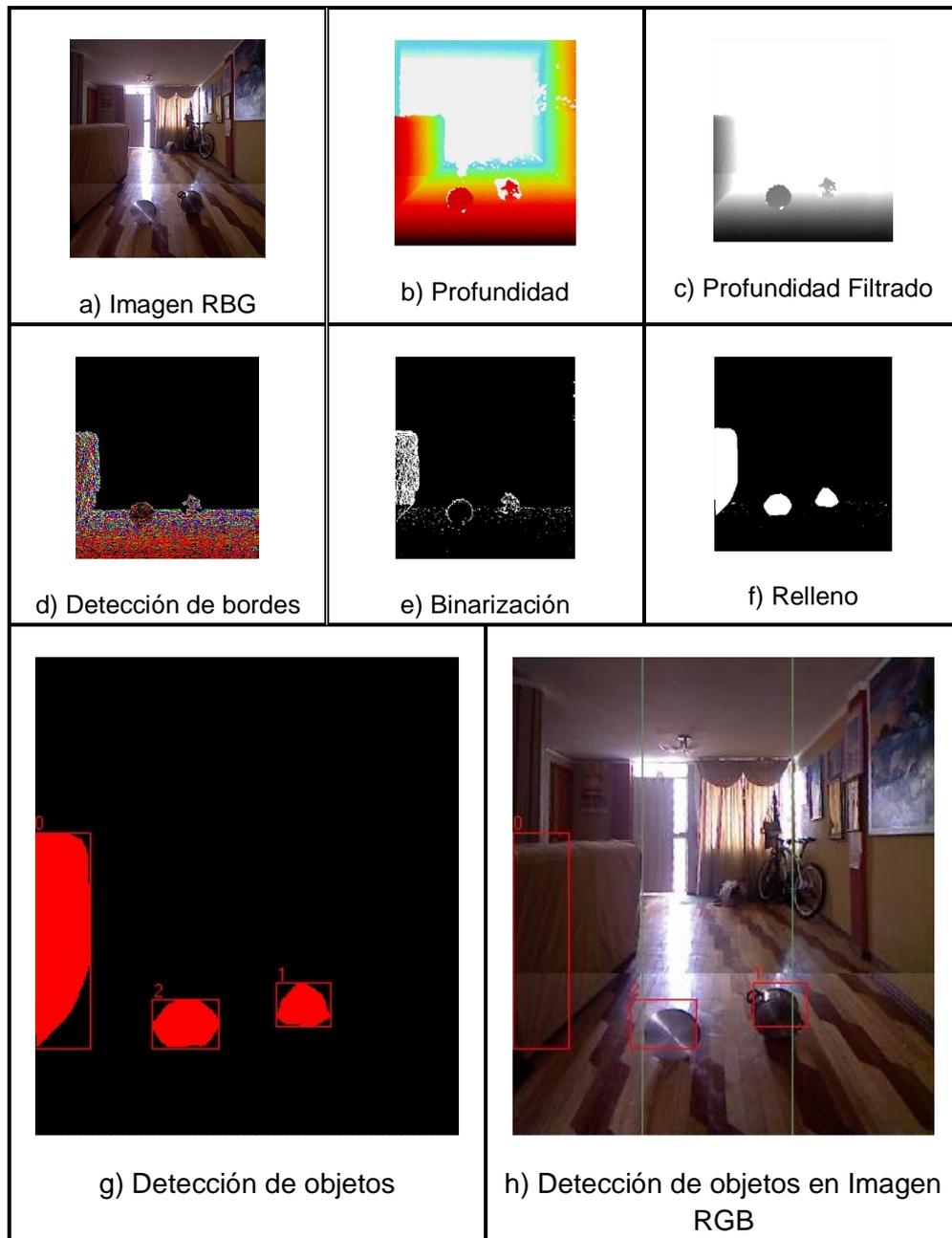
Figura 3.17: Prueba de objetos transparentes

El resultado de la prueba indica que los objetos transparentes no pueden ser detectados totalmente, segmentándose sólo parte de éstos.

### **3.1.11 Prueba de objetos reflectantes**

Ante objetos reflectantes el Kinect no puede estimar la profundidad, el algoritmo le asigna el valor máximo (4095mm) en esta situación con lo que los objetos se aprecian casi completamente (Figura 3.18).

Otra limitación radica cuando hay un cambio muy fuerte de luz solar que impide que la luz infrarroja pueda ser detectada por el Sensor IR (objeto de la derecha en la imagen).



**Figura 3.18: Prueba de objetos reflectantes**

Las pruebas indican que los objetos reflectantes se observan en parte y en mayor proporción siempre y cuando no exista un cambio brusco de luminosidad.

### 3.1.12 Prueba de objetos unidos

Se prueba la respuesta del algoritmo ante objetos unidos físicamente (Figura 3.19) o respecto al ángulo de visión del Kinect (Figura 3.20).

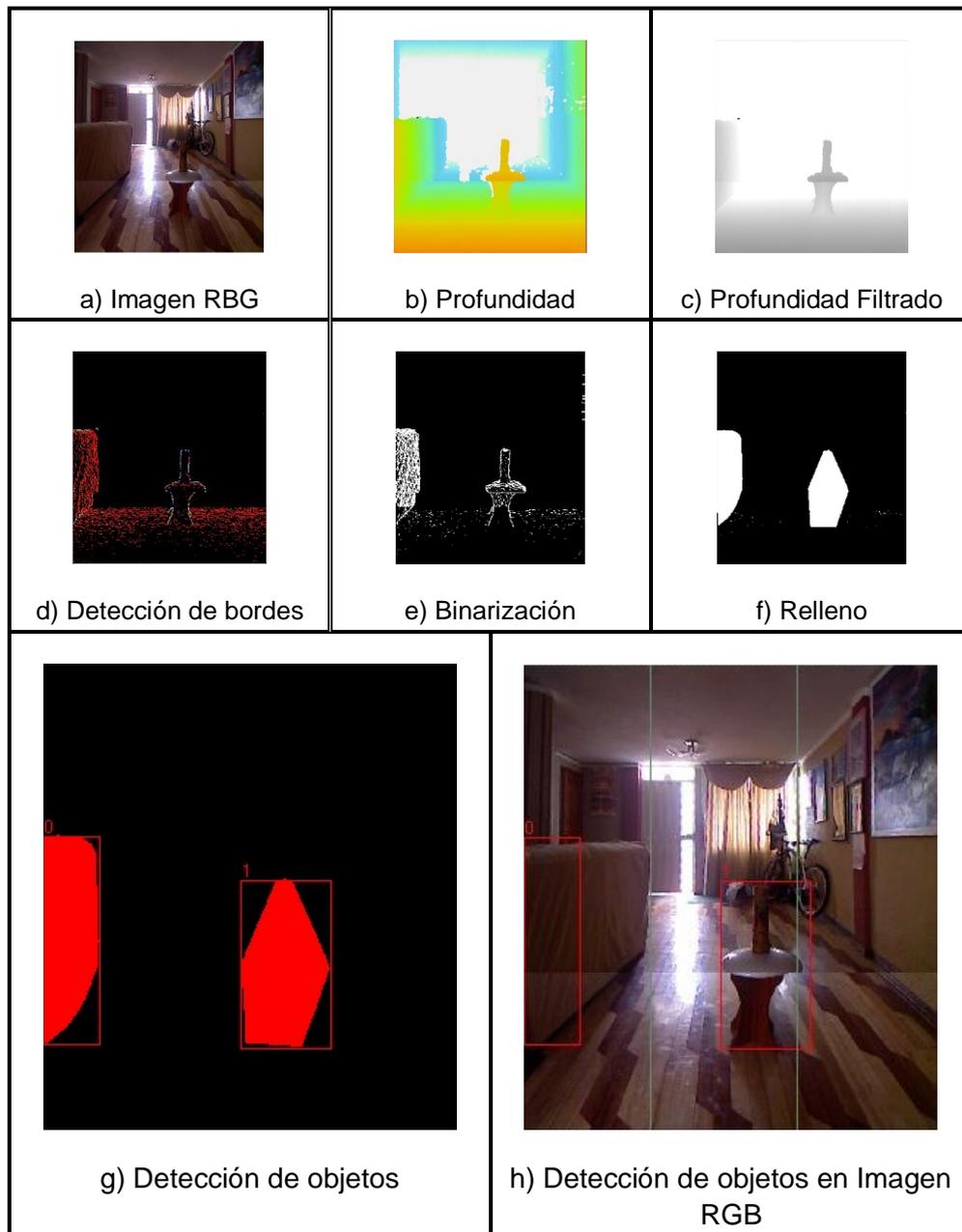
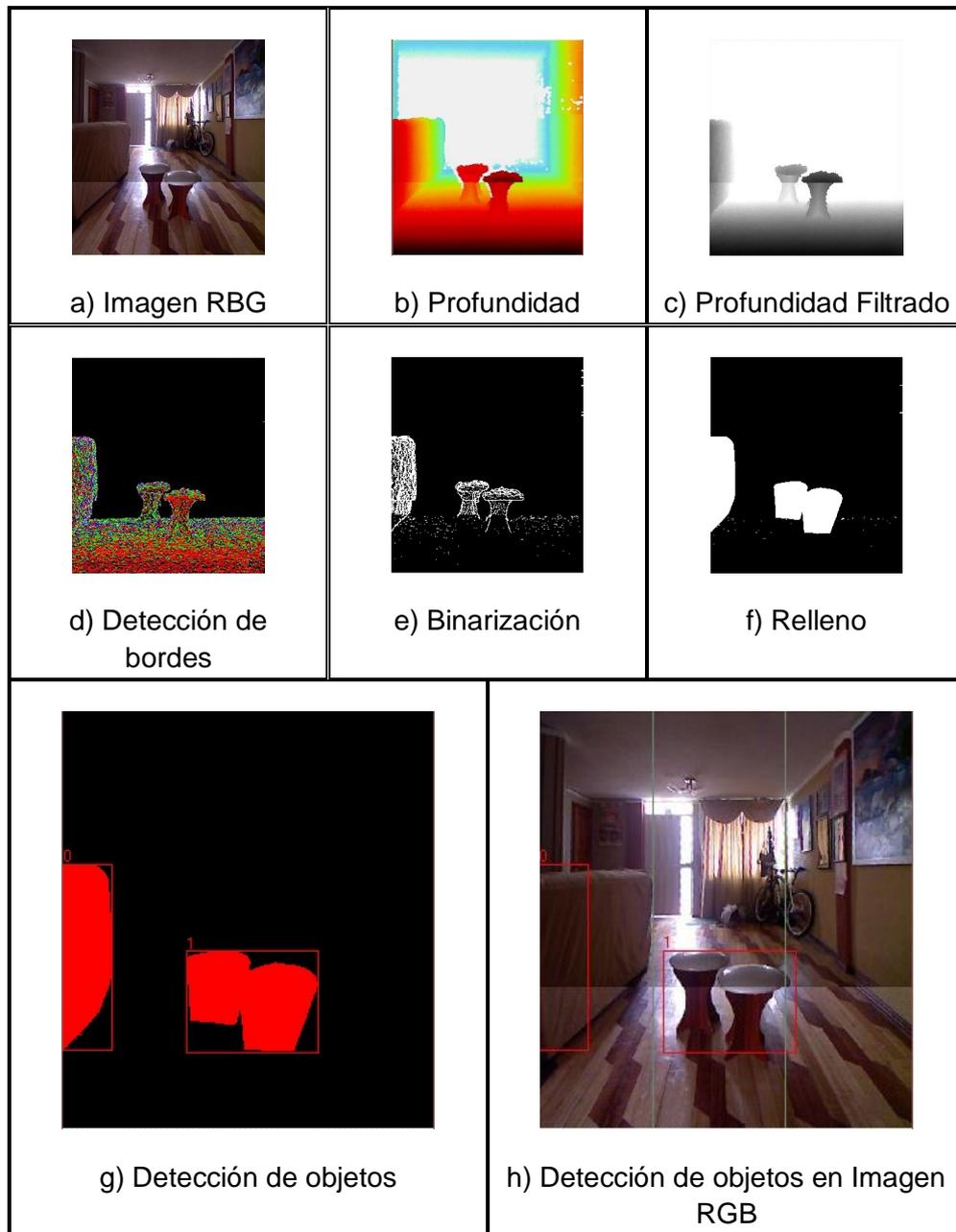


Figura 3.19: Prueba de objetos unidos físicamente



**Figura 3.20: Prueba de objetos unidos respecto al ángulo de visión del Kinect**

Para el algoritmo, los objetos unidos físicamente o visualmente son tomados como uno solo, por lo que se debe tener precaución al tomar un rango de profundidad adecuado para que los objetos no se mezclen con el fondo.

## CAPÍTULO 4

### CONCLUSIONES RECOMENDACIONES

#### 4.1 CONCLUSIONES

- El sensor Kinect proporciona información muy fiable bajo ciertas condiciones del entorno y basta la información de profundidad para clasificar un determinado evento con rápido análisis computacional.
- Otros métodos para detectar objetos a través de la profundidad resultan costosos, o no pueden detectar más de un objeto a la vez, por lo que el sensor Kinect se convierte en una excelente opción.
- El algoritmo puede detectar objetos de materiales como plástico, cuero, tela, cartón, porcelana que son los más comunes.
- Los objetos que presenten forma circular, rectangular, cilíndrica y formas irregulares pueden ser detectadas por el algoritmo.
- El algoritmo puede discernir tanto objetos pequeños, de una altura aproximada de 10cm, así como objetos pequeños y grandes.
- Independientemente de la posición en la que se encuentre el objeto, sea en el piso, al frente, o en la parte superior (colgante como una lámpara), el algoritmo es capaz de detectarlo.
- El filtraje por profundidad entrega una imagen con mejores condiciones para ser segmentada eliminando el fondo de la escena.

- Para la detección de bordes de la imagen se probaron los filtros de Diferenciación, Gradiente, Prewitt, Roberts, Sigma y Sobel, de los cuales el filtro Sigma fue el más adecuado.
- Para optimizar la etapa de filtraje se dividió la imagen en dos partes, de tal manera que los objetos cercanos al piso tienen un mejor filtro al eliminar partículas con valores por debajo del valor de umbral establecido.
- Para evitar que ante objetos que presenten orificios el algoritmo segmente a cada orificio como un objeto individual, se rellena la forma exterior del objeto, logrando así detectar solo uno.
- La mayor efectividad del algoritmo se da cuando los objetos no presentan características de transparencia o reflectancia, ya que no pueden ser detectados y se detectan parcialmente.
- Cuando en la escena existe un cambio brusco de luminosidad, pueden presentarse objetos falsos.
- Los objetos físicamente o visualmente unidos respecto al punto de observación del Kinect, se detectan como uno solo.
- El indicativo de que el sensor no pudo establecer un valor de profundidad para un pixel es que el resultado de profundidad del algoritmo indique 4095mm.
- El algoritmo puede servir como base para ser colocado en robots móviles en interiores, evitando así chocar contra obstáculos presentes en la ruta, o para que una persona no vidente con la presencia de una pc y unos audífonos pueda guiarse en una habitación.

## 4.2 RECOMENDACIONES

- El Kinect debe estar a una distancia considerable de los objetos de acuerdo a las especificaciones propias del hardware, para que el proceso de detección se realice eficientemente.
- El uso de Labview para el desarrollo de aplicaciones de visión es una opción excelente, ya que permite usar y modificar de manera sencilla los diferentes algoritmos, filtros y análisis binarios que forman parte del procesamiento digital de imágenes.
- Seleccionar el valor de altura que corresponda a la altura a la que el Kinect se encontrará físicamente, de tal modo que la segmentación sea exitosa.
- Seleccionar el idioma del texto a voz, de acuerdo a la voz a utilizarse y previamente instalada en la pc, para que la entonación sea la correcto.
- Seleccionar un valor de profundidad de tal forma que los objetos no se mezclen con el fondo.
- Revisar el Anexo I de esta tesis como ayuda del manejo del VI.
- El motor del Kinect no debe ser movido manualmente, para evitar que deje de reaccionar ante una señal enviada por software.
- Para dar mayor eficiencia al programa de LabView es viable que el computador utilizado sea exclusivamente para uso de este proyecto.

- Se recomienda que si el Sensor se va a encontrar a una altura baja y se necesite mejor tiempo de reacción, se elimine la parte del giro del motor del algoritmo.
- Para aplicaciones en que un robot móvil evite los objetos, en base a las salidas de posición de los objetos y posibles vías se pueden enviar instrucciones de activación a motores que controlen el robot.
- Para que una persona no vidente emplee este algoritmo, requiere una computadora, audífonos y un inversor con baterías para el Kinect.

## BIBLIOGRAFÍA

ACURIO, Eliana; ENCARNACIÓN Diana: Diseño e construcción de un módulo didáctico de visión artificial orientada al Control de Calidad de llenado de botellas de vidrio con diferente tipo de líquido. Director Ing. Ana Rodas. Escuela politécnica Nacional. Quito Febrero 2011.

ANDRIAGO, Miguel: Sensores de posición, descripción, selección y uso texto. Universidad Nacional Experimental Politécnica Antonio José de Sucre. Barquisimeto, Noviembre 2003.

CÓRDOVA, Fabricio: Detección de robo/abandono de objetos en interiores utilizando cámaras de profundidad. Universidad autónoma de Madrid, Diciembre 2012.

HIDALGO, Víctor: Implementación de un sistema de autenticación biomédica basado en huellas digitales. Director: Hugo Moreno. Escuela Superior Politécnica de Chimborazo. Riobamba 2010.

IRALDE L. Iñaki; PINA C. Alfredo: Desarrollo de aplicaciones con Microsoft Kinect. Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación. Pamplona, Abril 2012.

Labview National Instruments: IMAQVision.chm y NiVisionConcepts.chm.

MARTINEZ, Alain: Bastón blanco para prevenir obstáculos. Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco. México D.F, Agosto 2012.

MESÍAS, Alejandro; LÓPEZ, Christian: Diseño e implementación de un prototipo a escala de un robot móvil acompañante. Director: Nelson

Sotomayor MSc. Escuela Politécnica Nacional, Facultad de Eléctrica y Electrónica. Quito, Junio 2012.

RODRIGUEZ, Patricio: Aplicación del filtro de Kalman al seguimiento de objetos en secuencias de imágenes. Tutor, Antonio Sanz, Juan José Pantrigo. Universidad Rey Juan Carlos. 2003.

VISCAÍNO, José. SEBASTIÁ José: Labview: Entorno gráfico de programación. Segunda edición. España. 2011.

## ENLACES ELECTRÓNICOS

*Bastón blanco*. [En línea]. Recuperado el 08 de marzo del 2013, Disponible en Web: <[http://es.wikipedia.org/wiki/Bast%C3%B3n\\_blanco](http://es.wikipedia.org/wiki/Bast%C3%B3n_blanco)>

*Bastón blanco electrónico teletacto*. [En línea]. Recuperado el 08 de marzo del 2013, Disponible en Web: <<http://familia.bertho.free.fr/ace31/index.php?page=boitier>>

*Clase programación*. [En línea]. Recuperado el 25 de febrero del 2013, Disponible en Web: [http://zarza.usal.es/~fgarcia/doc/tuto2/I\\_1.htm](http://zarza.usal.es/~fgarcia/doc/tuto2/I_1.htm)

*Clúster Labview National Instruments*, [En línea]. Recuperado el 26 de febrero del 2013, Disponible en Web: <[http://www.etitudela.com/entrenadorcomunicaciones/downloads/labviewu\\_sodearraystring.pdf](http://www.etitudela.com/entrenadorcomunicaciones/downloads/labviewu_sodearraystring.pdf)>

*Conoce con detalles técnicos cómo funciona Kinect*. [En línea]. Recuperado el 22 de septiembre del 2012, Disponible en Web: <<http://www.tierragamer.com/index.php/conoce-como-funciona-kinect/>>

*Depth Camera*. [En línea]. Recuperado el 1 de marzo del 2013, Disponible en Web: <<http://msdn.microsoft.com/en-us/library/hh438997.aspx>>

*Depth Space Range*. [En línea]. Recuperado el 1 de marzo del 2013, Disponible en Web: <[http://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth\\_Ranges](http://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth_Ranges)>

*Detección difusa*. [En línea]. Recuperado el 09 de marzo del 2013, Disponible en Web: <<http://www.monografias.com/trabajos67/tecnologia-control/tecnologia-control3.shtml>>

*Express user Libraries Labview National Instruments.* [En línea]. Recuperado el 26 de febrero del 2013, Disponible en Web: <http://digital.ni.com/public.nsf/allkb/7F0027314F936E82862570F10083C0F3>

*Filtro de Localización de objetos.* [En línea]. Recuperado el 26 de febrero del 2013, Disponible en Web: <http://www.escet.urjc.es/~jjpantrigo/PFCs/MemoriaKalmanJun03.pdf>

*IMAQ Toolkit Labview National Instruments.* [En línea]. Recuperado el 20 de enero del 2013, Disponible en Web: <http://dspace.ups.edu.ec/bitstream/123456789/161/6/Capitulo%205.pdf>

*Kinect.* [En línea]. Recuperado el 09 de marzo del 2013, Disponible en Web: <http://www.respuestario.com/como/como-funciona-kinect-guia-de-uso-practica-paso-a-pasor>

*Kinect SDK v1.0 para Labview Hacker.* [En línea]. Recuperado el 20 de octubre del 2012, Disponible en Web: <https://decibel.ni.com/content/servlet/JiveServlet/download/22530-3-47391/Kinect%20SDK%20v1.0.zip>

*Kinect primeros pasos.* [En línea]. Recuperado el 11 de marzo del 2013, Disponible en Web: [http://alumnos.elo.utfsm.cl/~flopezp/Kinect\\_v1.pdf](http://alumnos.elo.utfsm.cl/~flopezp/Kinect_v1.pdf)

*Lentes para ciegos.* [En línea]. Recuperado el 09 de marzo del 2013, Disponible en Web: <http://bloguned1.blogspot.com/2009/07/caslip-lentes-para-ciegos.html>

*Logo Labview.* [En línea]. Recuperado el 26 de febrero del 2013, Disponible en Web: <http://www.cetoni.de/development/software.html>

*Microsoft Kinect SDK*. [En línea]. Recuperado el 20 de enero del 2013, Disponible en Web: <<http://msdn.microsoft.com/en-us/library/hh855354.aspx>>

*Microsoft Kinect SDK* descarga. [En línea]. Recuperado el 20 de octubre del 2012, Disponible en Web: <[go.microsoft.com/fwlink/?LinkID=247938](http://go.microsoft.com/fwlink/?LinkID=247938)>

*Microsoft Kinect SDK Depth ranges*. [En línea]. Recuperado el 08 de marzo del 2013. Disponible en Web: <[http://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth\\_Ranges](http://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth_Ranges)>

*Microsoft Kinect SDK instrucciones*. [En línea]. Recuperado el 03 de marzo del 2013, Disponible en Web: <<http://msdn.microsoft.com/en-us/library/hh855419.aspx>>

*Microsoft .NET Framework 4.0*. [En línea]. Recuperado el 20 de febrero del 2013, Disponible en Web: <<http://www.microsoft.com/es-es/download/details.aspx?id=17718>>

*Microsoft .NET Framework 4.0* descarga. [En línea]. Recuperado el 20 de octubre del 2012, Disponible en Web: <<http://www.microsoft.com/es-es/download/details.aspx?id=17718>>

*Microsoft SDK vs OpenNI*, [En línea]. Recuperado el 09 de marzo del 2013. Disponible en Web: <<http://www.argencon.org.ar/sites/default/files/123.pdf>>

*Microsoft Speech Library*, [En línea]. Recuperado el 01 de marzo del 2013. Disponible en Web: <<http://msdn.microsoft.com/en-us/library/ms862156.aspx>>

*Microsoft Speech SDK*. [En línea]. Recuperado el 20 de febrero del 2013, Disponible en Web: <<http://www.microsoft.com/en-us/download/details.aspx?id=10121>>

*Microsoft Speech SDK descarga*. [En línea]. Recuperado el 20 de octubre del 2012, Disponible en Web: <<http://www.microsoft.com/en-us/download/details.aspx?id=10121>>

*Mi ultrasonic*. [En línea]. Recuperado el 26 de febrero del 2013, Disponible en: <<http://itzamna.bnct.ipn.mx/dspace/bitstream/123456789/11470/1/21.pdf>>

*Perro guía*. [En línea]. Recuperado el 08 de marzo del 2013, Disponible en Web: <<http://www.escuelaperrosguia.com/etapas/etapas.htm>>

*PrimerSense*, [En línea]. Recuperado el 1 de octubre del 2012. Disponible en Web: <<http://www.primesense.com/en/technology/115-the-primesense-3d-sensing-solution>>

*Principio de operación ultrasónico*. [En línea]. Recuperado el 09 de marzo del 2013, Disponible en Web: <<http://www.monografias.com/trabajos67/tecnologia-control/tecnologia-control3.shtml>>

*Programar el sensor Kinect de la consola Xbox 360*. [En línea]. Recuperado el 1 de octubre del 2012, Disponible en Web: <[http://biqfr.blogspot.com/2010\\_11\\_02\\_archive.html](http://biqfr.blogspot.com/2010_11_02_archive.html)>

*Proyector láser infrarrojo*. [En línea]. Recuperado el 1 de octubre del 2012, Disponible en Web: <[http://biqfr.blogspot.com/2010\\_11\\_02\\_archive.html](http://biqfr.blogspot.com/2010_11_02_archive.html)>

*Representación de una imagen mediante coordenadas.* [En línea]. Recuperado el 26 de febrero del 2013, Disponible en Web: <<http://dspace.epn.edu.ec/bitstream/15000/9218/1/T11089.pdf>>

*Sensor que posee tecnología radar.* [En línea]. Recuperado el 09 de marzo del 2013, Disponible en Web: <<http://www.sensorstecnicos.net/en/productos/category/244/sensores-y-transmisores/sensores-ultrasonicos/sensor-de-distancia-con-tecnologia-radar>>

*Sensor Kinect.* [En línea]. Recuperado el 10 de marzo del 2013, Disponible en Web: <<http://msdn.microsoft.com/en-us/library/jj131033.aspx>>

*Sistema de Visión artificial.* [En línea]. Recuperado el 25 de febrero del 2013, Disponible en Web: <<http://bibdigital.epn.edu.ec/bitstream/15000/4119/1/CD-3466.pdf>>

*Técnicas de umbralización.* [En línea]. Recuperado el 25 de febrero 2013. Disponible en: <<http://dspace.epn.edu.ec/bitstream/15000/9218/1/T11089.pdf>>

*Text to speech.* [En línea]. Recuperado el 08 de febrero del 2013, Disponible en Web: <<https://decibel.ni.com/content/docs/DOC-2263>>

*Una vara para ciegos.* [En línea]. Recuperado el 09 de marzo del 2013, Disponible en Web: <<http://www.impresionante.net/10-03-2008/general/una-vara-para-ciegos>>

*Visión artificial,* [En línea]. Recuperado el 25 de febrero 2013. Disponible en Web: <<http://bibdigital.epn.edu.ec/bitstream/15000/4119/1/CD-3466.pdf>>

*Voz en español,* [En línea]. Recuperado el 25 de febrero 2013. Disponible en Web: <<http://www.mediafire.com/?u6sehj62xo2wl8x>>

<b>ANEXO A</b>	<b>ESPECIFICACIONES TÉCNICAS DEL KINECT</b>	<b>A – 1</b>
----------------	---	--------------



**Figura A.1: Kinect por dentro**

### **Sensores**

Lentes de color y sensación de profundidad

Micrófono multi-arreglo

Ajuste de sensor con su motor de inclinación

Totalmente compatible con las consolas existentes de Xbox 360

### **Campo de visión**

Campo de visión horizontal: 57 grados

Campo de visión vertical: 43 grados

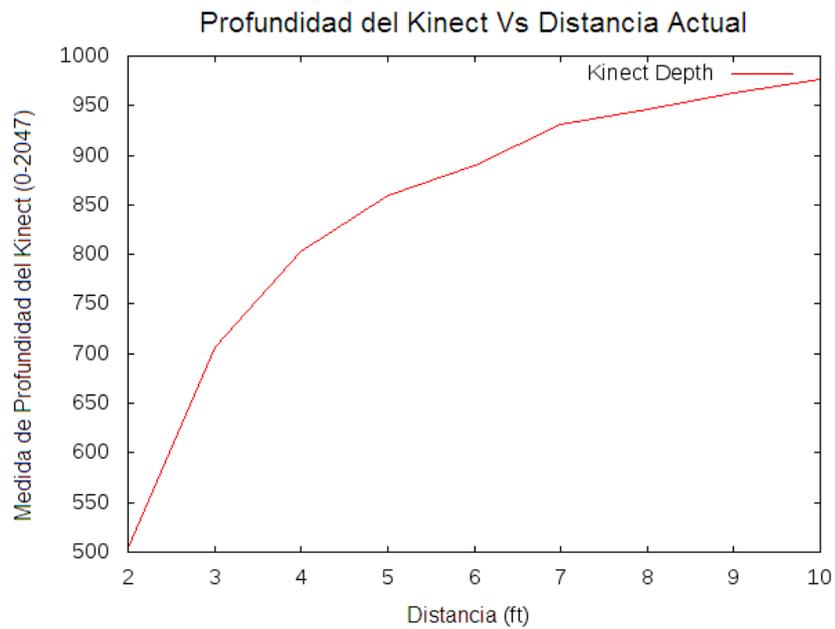
Rango de inclinación física:  $\pm 27$  grados

<b>ANEXO A</b>	<b>ESPECIFICACIONES TÉCNICAS DEL KINECT</b>	<b>A – 2</b>
<p>Rango de profundidad del sensor: 1,2 – 3,5 metros</p> <p><b>Data Streams (Flujo de datos)</b></p> <p>320 x 240 a 16 bits de profundidad @ 30fps</p> <p>640 x 480 32-bit de color @30fps</p> <p>Audio de 16-bit @ 16 kHz</p> <p><b>Sistema de Seguimiento</b></p> <p>Rastrea hasta 6 personas, incluyendo 2 jugadores activos</p> <p>Rastrea 20 articulaciones por jugador activo</p> <p>Capacidad para mapear jugadores activos en Live Avatars</p> <p><b>Sistema de audio</b></p> <p>Chat en vivo y voz dentro del juego (requiere Xbox Live Gold)</p> <p>Sistema de cancelación de eco que aumenta la entrada de voz</p> <p>Reconocimiento de voz múltiple.</p> <p><b>Dimensiones</b></p> <p>Ancho: 14cm</p> <p>Alto: 3.5cm</p> <p>Profundidad: 5cm</p>		

**Interface de datos**

USB 2.0

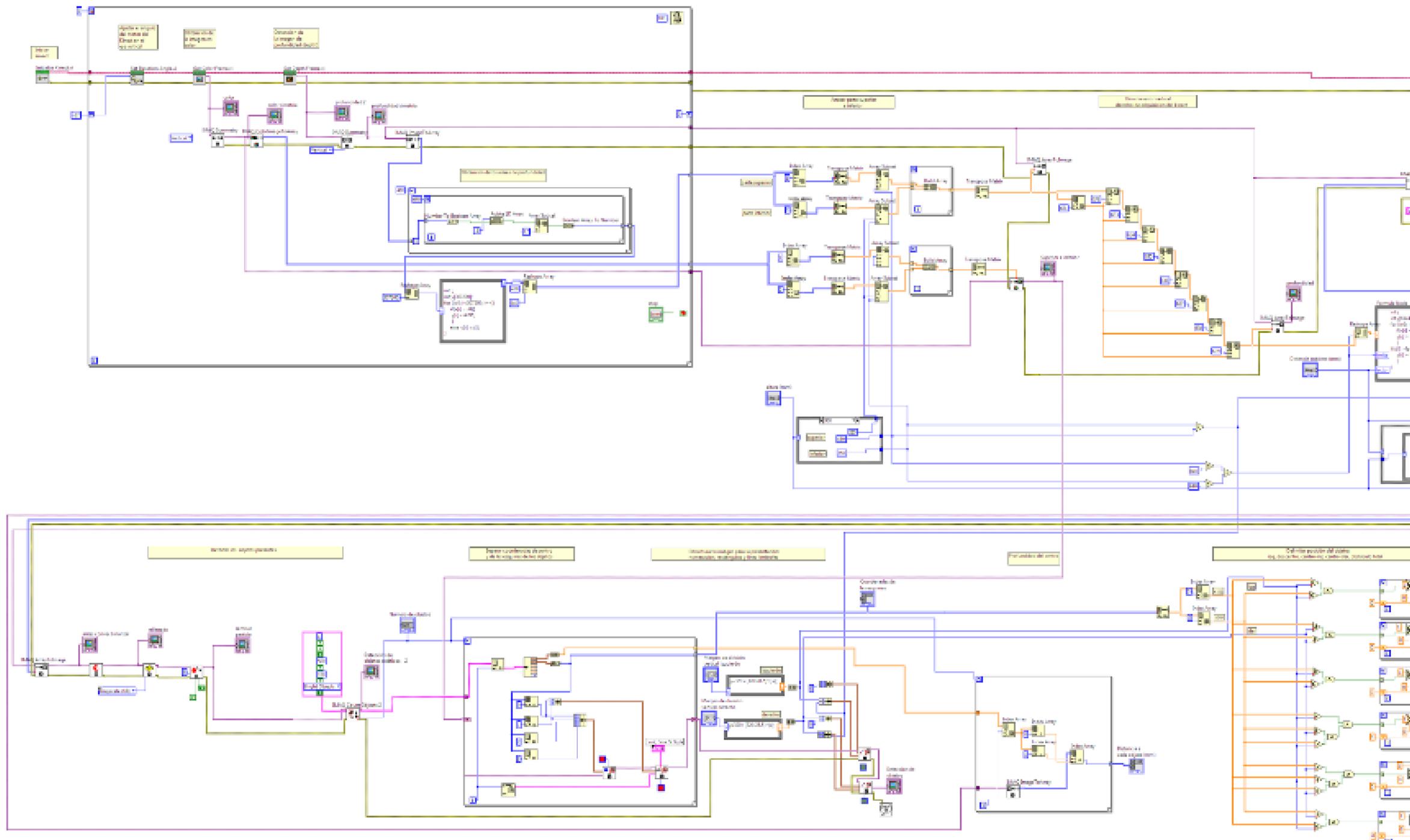
La medición de la profundidad del Kinect en función de la distancia no es lineal, mientras más se aleja del sensor, menor información detallada se obtendrá de los objetos.



**Figura A.2 Profundidad del Kinect Vs Distancia real**

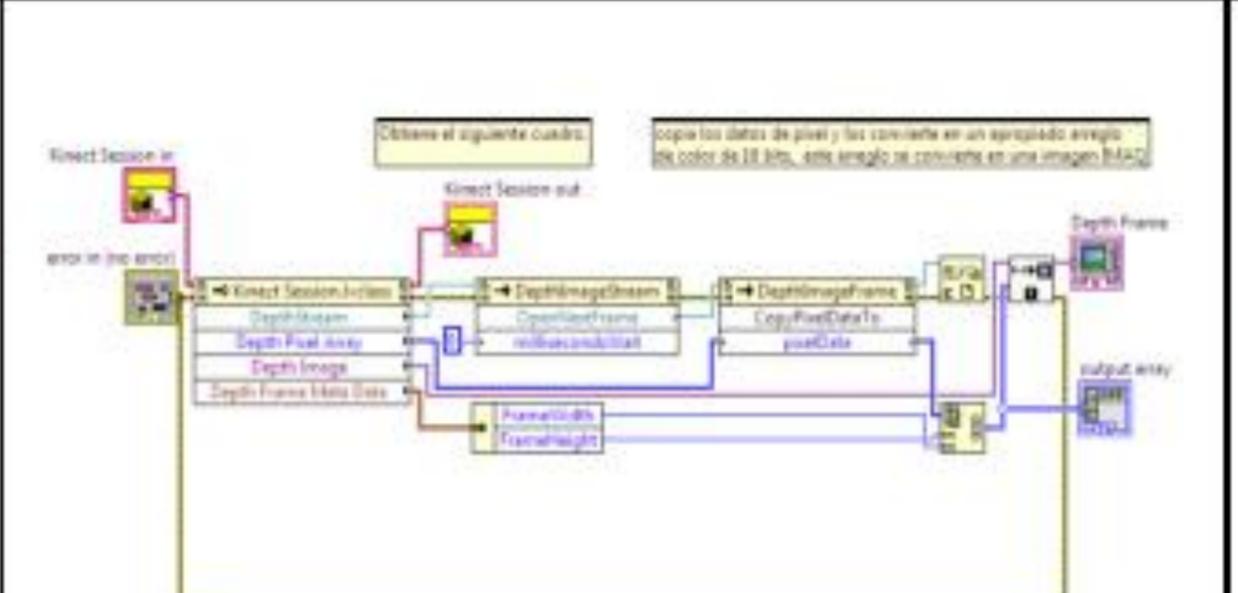
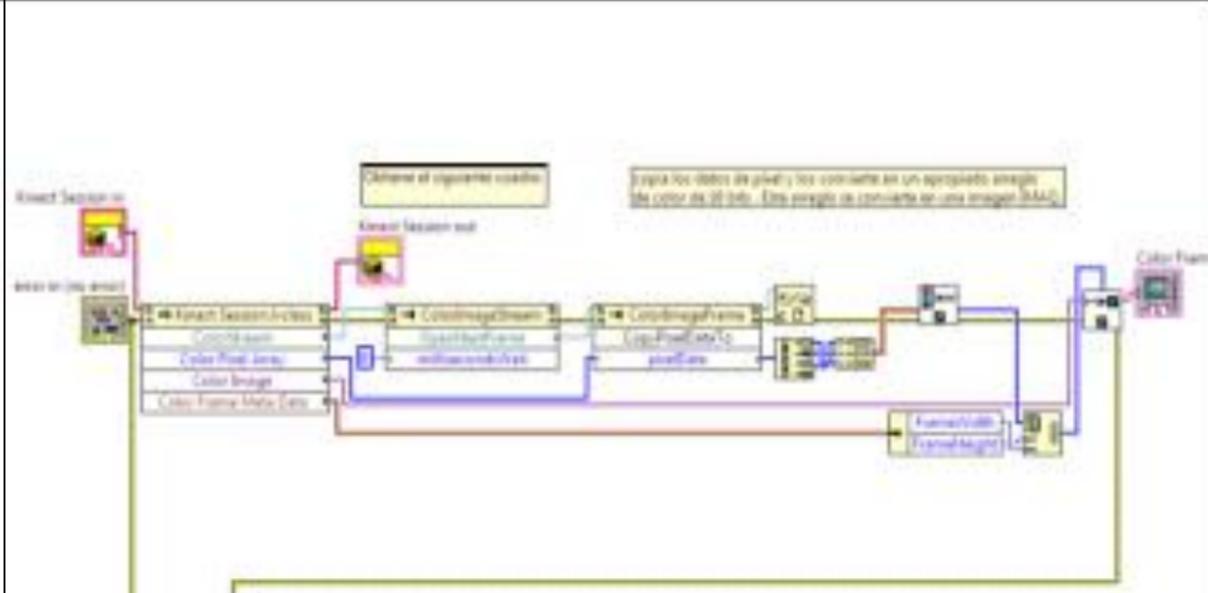
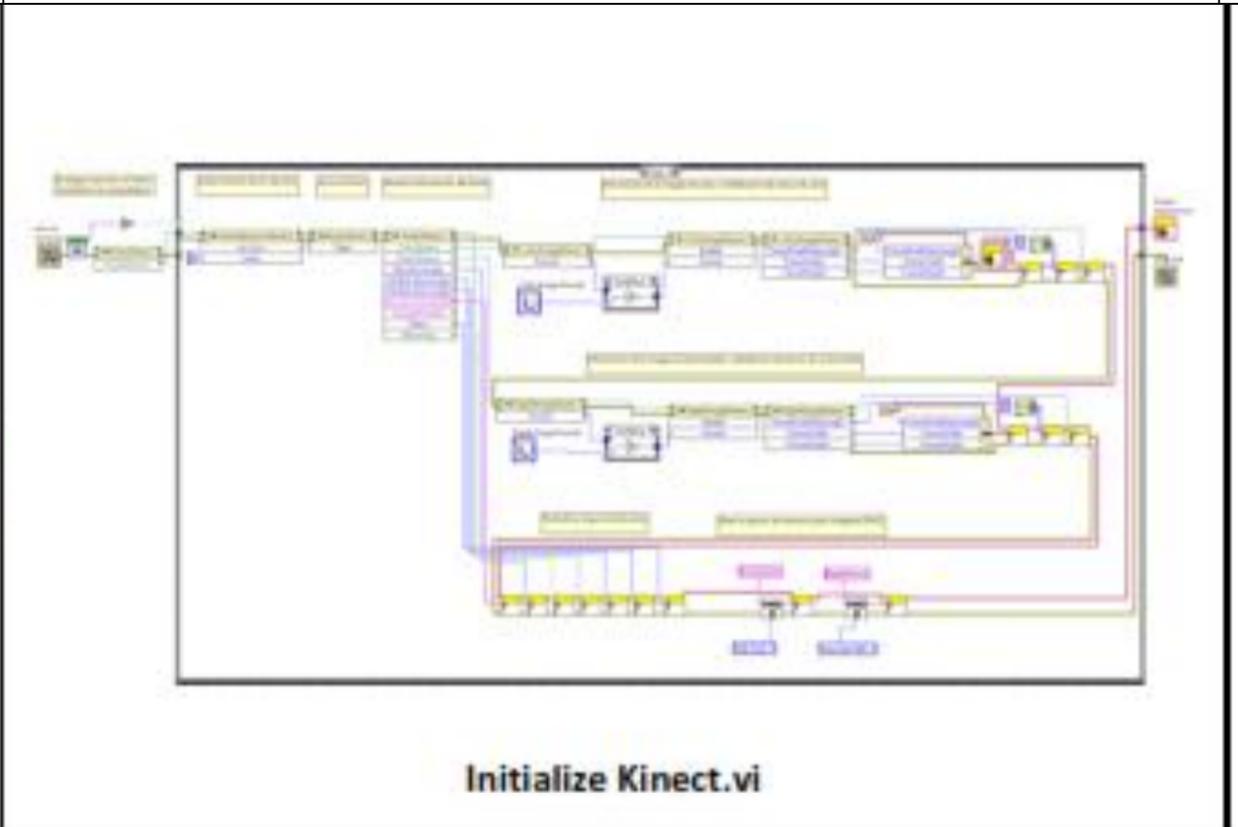
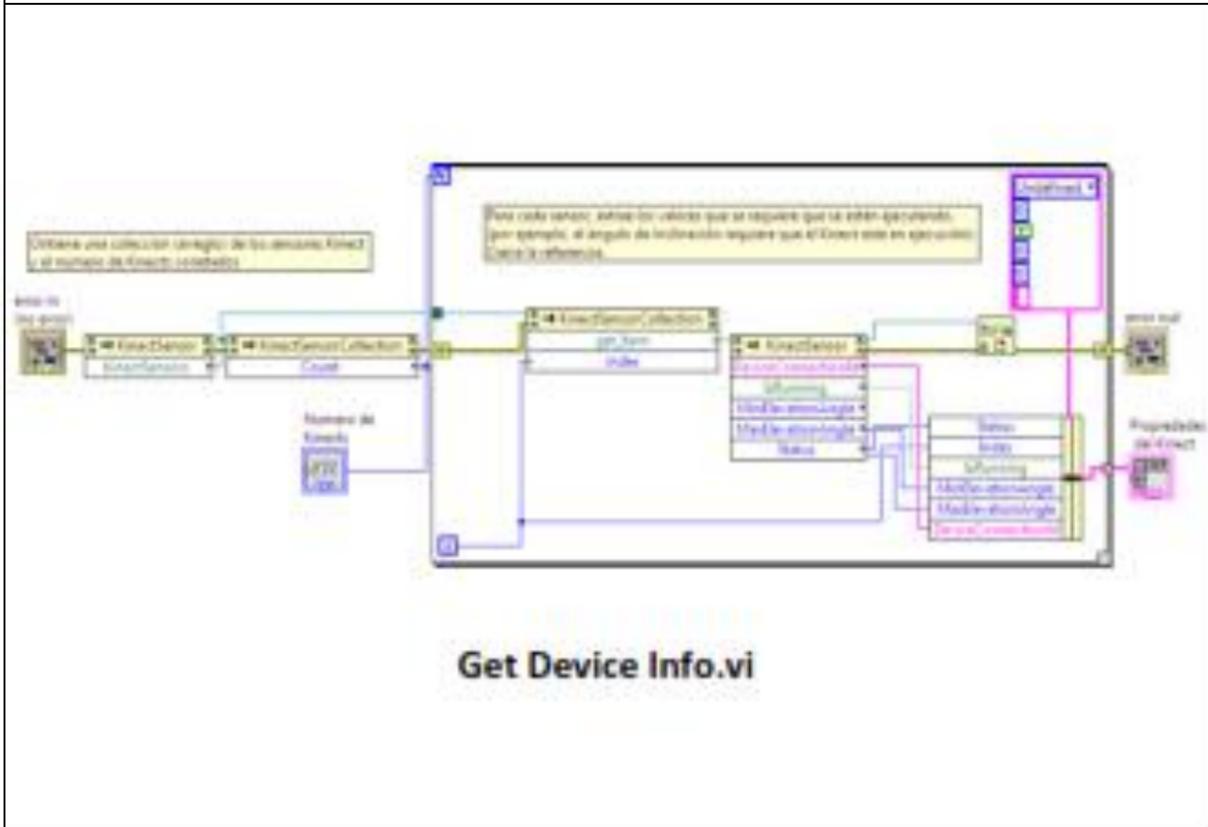
# ANEXO B

## Diagrama de bloques del algoritmo de detección de objetos y los subVI que lo conforman.



# ANEXO C

Diagrama de bloques del algoritmo de detección de objetos y los subVI que lo conforman.



<b>ANEXO D</b>	<b>TAMAÑO DE IMAGEN DE ACUERDO A LA ALTURA SELECCIONADA</b>	<b>C – 1</b>
----------------	---	--------------

**TABLA C.1: Tamaño de la imagen por la altura**

Tamaño Altura (mm)	Filas imagen superior	Filas Imagen inferior	Inicio de filas imagen inferior	Total
100	430	0	0	430
300	340	160	90	500
500	400	250	160	650
700	390	290	160	680
900	480	250	225	730
1000	480	245	235	725

<b>ANEXO E</b>	<b>VALOR MÍNIMO DEL FILTRO TRAS LA DETECCIÓN DE BORDES</b>	<b>D – 1</b>
----------------	--	--------------

Para la imagen superior el valor de umbral es 1.

**TABLA D.1: VALOR DE UMBRAL IMAGEN INFERIOR**

Altura (mm) Distancia	100	300	500	700	900	1000
1000	0	10	4	1	8	1
1200	0	12	10	5	8	1
1400	0	14	9	12	8	10
1600	0	18	11	12	8	10
1800	0	13	14	12	8	12
2000	1	14	15	12	12	13

**TABLA D.2: TAMAÑO DE LA IMAGEN SUPERIOR PARA EL FILTRO**

Altura (mm) Distancia	100	300	500	700	900	1000
1000	430	460	555	590	730	725
1200	430	420	500	600	730	715
1400	430	380	460	520	620	690
1600	430	370	440	490	590	630
1800	430	360	420	450	570	580
2000	430	340	400	440	530	540

<b>ANEXO D</b>	<b>VALOR MÍNIMO DEL FILTRO TRAS LA DETECCIÓN DE BORDES</b>	<b>D – 2</b>
----------------	--	--------------

**TABLA D.3: TAMAÑO DE LA IMAGEN SUPERIOR PARA EL FILTRO**

Altura (mm) Distancia	100	300	500	700	900	1000
1000	0	40	95	90	0	0
1200	0	80	150	80	0	10
1400	0	120	190	160	110	35
1600	0	130	210	190	140	95
1800	0	140	160	200	160	145
2000	0	160	250	240	200	185

**ANEXO F**

**TEXTO A SER HABLADO EN  
INGLÉS Y ESPAÑOL**

**E – 1**

**Tabla E.1: Texto a ser hablado**

<b>Español</b>	<b>Inglés</b>
Vía libre	Go
Obstáculo total	Total Obstacle
Obstáculo al frente, posible vía izquierda	Obstacle in front, possible via: left
Obstáculo al frente, posible vía izquierda	Obstacle in front, possible via: right
Error de Proceso	Process error

**Tabla F.1 Porcentaje de respuesta del algoritmo ante diferentes objetos**

Cantidad Tipo de objeto	Objetos a ser detectados	Objetos detectados	Porcentaje de Efectividad
Localización	10	10	100%
Forma	10	10	100%
Tamaño	10	10	100%
Distancia	10	10	100%
Profundidad	10	10	100%
Materiales	10	10	100%
Transparente	10	5	50% (5 no localizados)
Reflectante	10	7	70% (3 no localizados)
Negros	10	8	80% (2 no localizados)
Unidos	10	10	100%
Con cambio brusco de luminosidad	10	12	2 objetos falsos

ANEXO H	ESPECIFICACIONES DEL ALGORITMO	G – 1
<p><b>1. Distancia máxima en milímetros</b></p> <p>Rango entre 1000 y 2000mm en línea recta. Distancia desde el piso aproximadamente 1000m. Debido al rango de inclinación el valor de distancia del Kinect tiene una variación del - 20%.</p> <p><b>2. Velocidad de ejecución</b></p> <p>8 segundos / muestra.</p> <p><b>3. Altura del Kinect</b></p> <p>Valores de 100, 300, 500, 700, 900 y 1000 mm.</p> <p><b>4. Idioma</b></p> <p>Inglés y español</p> <p><b>5. Limitaciones</b></p> <p>Propias del sensor, objetos no detectados o parcialmente detectados por ser transparentes o reflectantes, negros, generar sobra y falsos objetos detectados por cambios bruscos de luminosidad. Propias del algoritmo, objetos parcialmente detectados ante un límite de distancia no adecuado. Un valor de 4095mm en profundidad significa que el Kinect no pudo estimar la distancia a ese punto.</p> <p><b>6. Tamaño mínimo del objeto a ser detectado</b></p> <p>Altura de 10cm.</p>		

Para utilizar este VI se requiere instalar:

- LabVIEW 2011 o superior con NI Vision Development Module.
- Kinect SDK 1.5
- Microsoft .NET Framework 4.0
- Microsoft Speech SDK.
- Voz adicional en español para Text to Speech.

### LabVIEW 2011

Para la instalación se debe seguir los pasos de las siguientes figuras:



Figura H.1 Instalación de Labview paso 1



Figura H.2 Instalación de Labview paso 2

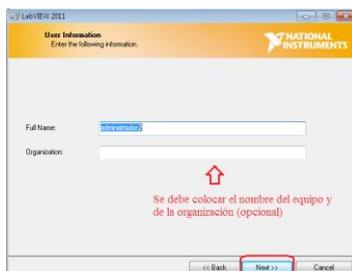


Figura H.3 Instalación de Labview paso 3

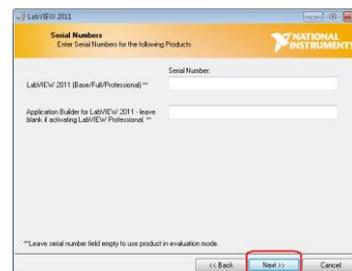


Figura H.4 Instalación de Labview paso 4

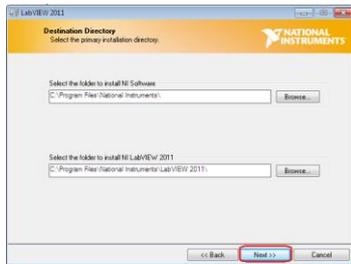


Figura H.5 Instalación de Labview paso 5

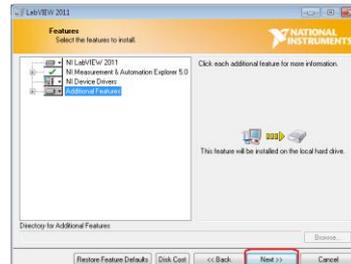


Figura H.6 Instalación de Labview paso 6



Figura H.7 Instalación de Labview paso 7



Figura H.8 Instalación de Labview paso 8

## NI Vision Development Module

Se debe seguir los pasos descritos en las figuras siguientes:



Figura H.9 Instalación NI Vision Development Module paso1

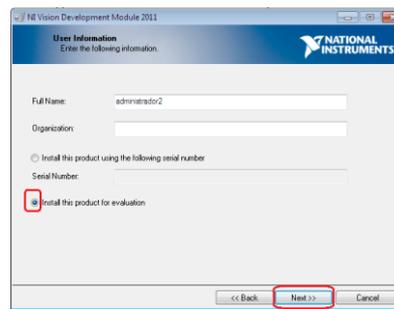


Figura H.10 Instalación NI Vision Development Module paso 2

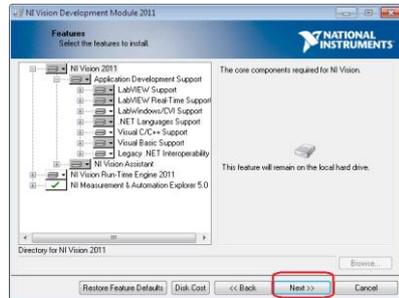


Figura H.11 Instalación NI Vision Development Module paso 3

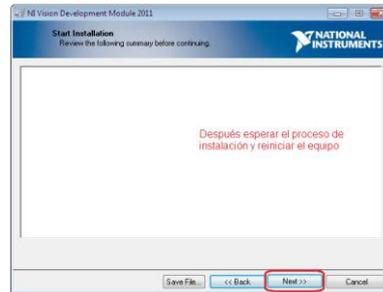


Figura H.12 Instalación NI Vision Development Module paso 4

Kinect SDK 1.5

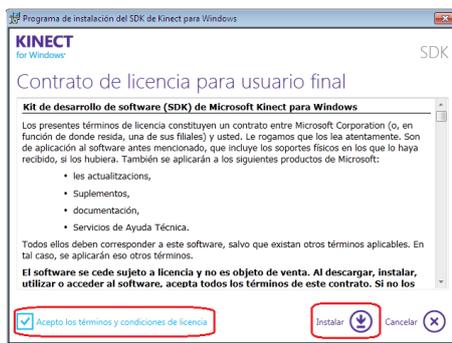


Figura H.13 Instalación Kinect SDK 1.5 paso 1

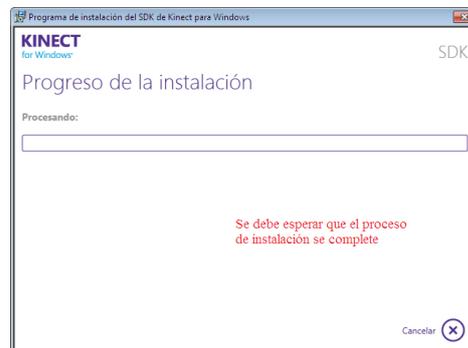


Figura H.14 Instalación Kinect SDK 1.5 paso 2

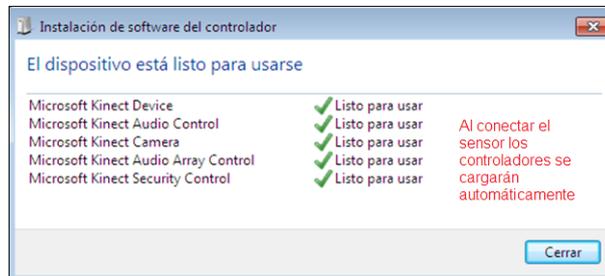
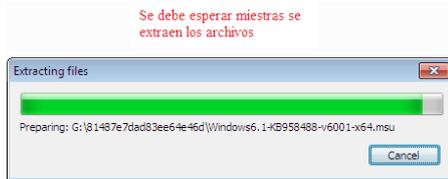
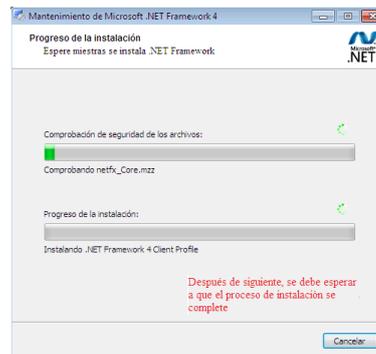


Figura H.15 Instalación Kinect SDK 1.5 paso 3

**Microsoft. NET Framework 4.0**

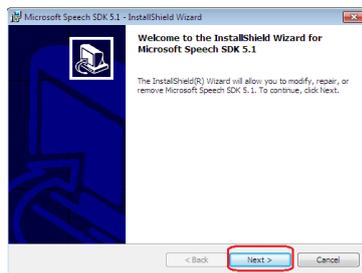


**Figura H.16 Instalación Microsoft. NET Framework 4.0 paso 1**

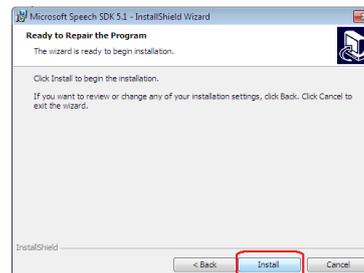


**Figura H.17 Instalación Microsoft. NET Framework 4.0 paso 2**

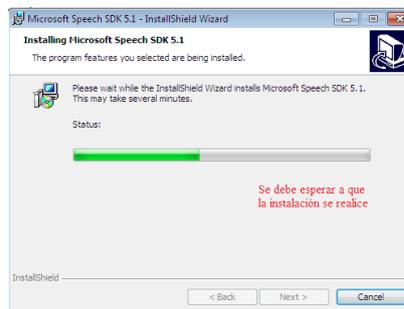
**Microsoft Speech SDK.**



**Figura H.18 Instalación Microsoft Speech SDK paso 1**



**Figura H.19 Instalación Microsoft Speech SDK paso 2**

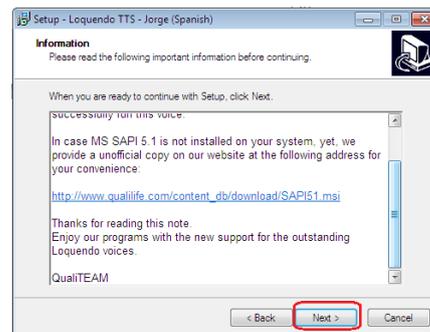


**Figura H.20 Instalación Microsoft Speech SDK paso 3**

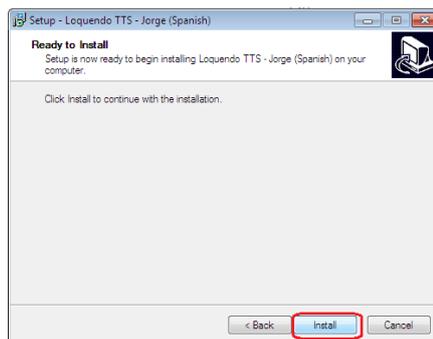
## Voz adicional en español para Text to Speech



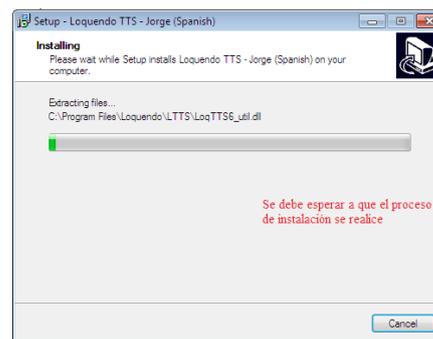
**Figura H.21 Instalación Voz es español  
paso 1**



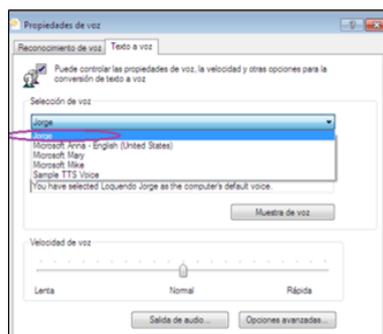
**Figura H.22 Instalación Voz es español  
paso 2**



**Figura H.23 Instalación Voz es español  
paso 3**

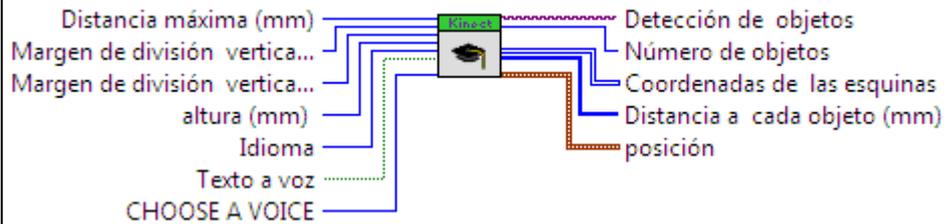


**Figura H.24 Instalación Voz es español  
paso 5**



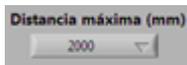
**Figura H.25 Instalación Voz es español paso 5, localización en la pc**

## Kinect obstáculos.vi



Este VI detecta obstáculos presentes a una **distancia máxima (mm)** definida por el usuario. Tiene la posibilidad de limitar el ancho de la trayectoria con los controles **margen de división vertical izquierdo** y **derecho**. La **altura** a la cual se encuentra el kinect es establecida por un control. Posee una salida de audio que es habilitada por un control boleano **Texto a voz**. El **idioma** puede ser español o inglés, y la voz disponible de la pc se puede seleccionar con el control **choose a voice**.

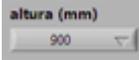
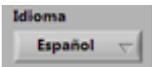
Los parámetros de salida incluyen el **número de objetos**, la **distancia a cada objeto (mm)**, las **coordenadas de las esquinas** y la **posición** de los objetos en la imagen.

1. **Distancia máxima en milímetros**

El sensor Kinect proporciona una medida de profundidad en milímetros, calculada hasta un objeto presente en la escena. En este control se puede seleccionar las distancias de 1000, 1200, 1400, 1600, 1800 y 2000 mm, para delimitar el rango en el cual se va a segmentar los objetos. Por las especificaciones físicas del Kinect, el valor mínimo de detección es 800mm en línea recta. El valor por defecto es 1800mm.

2. **Margen de división vertical izquierdo y derecho**

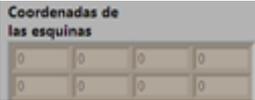
El ángulo de visión del Kinect horizontalmente es 57°, estos controles permiten limitar el espacio para la trayectoria, cada uno está en el rango de 0 a 26° en saltos de 1°. El valor por defecto es 9° para el izquierdo y 11° para el derecho.

ANEXO I	MANUAL DE OPERACIÓN DEL VI	I-2
<p data-bbox="316 479 683 546">  <b>3. Altura (mm)</b> </p> <p data-bbox="316 600 1361 689">Este control representa la altura a la cual se encuentra el sensor. Puede ser: 100, 300, 500, 700, 900 y 1000mm. El valor por defecto es 1000mm.</p> <p data-bbox="316 734 616 808">  <b>4. Idioma</b> </p> <p data-bbox="316 864 1361 1010">Las opciones son inglés y español. De no existir la voz en español, el texto será leído por la voz en inglés presente en la pc, pero la claridad no será eficiente. La opción por defecto es español.</p> <p data-bbox="316 1055 667 1111">  <b>5. Texto a voz</b> </p> <p data-bbox="316 1167 1361 1256">El control permite habilitar la fase de texto a voz o deshabilitarla, la opción por defecto es habilitada.</p> <p data-bbox="316 1301 791 1375">  <b>6. Choose a voice</b> </p> <p data-bbox="316 1431 1361 1632">La pc puede contener más de una voz disponible, por lo que de acuerdo a las voces presente se puede seleccionar la que más se adecue al usuario. La voz por defecto es Jorge, también existe Microsoft Anna y Sam.</p> <p data-bbox="316 1677 770 1778">  <b>7. Detección de objetos</b> </p> <p data-bbox="316 1834 1329 1868">La imagen muestra la visualización de los objetos contabilizados sobre</p>		

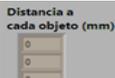
la imagen RGB, con un ligero desfase las imágenes de profundidad y color.

**8.  Número de objetos**

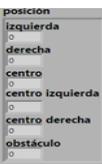
Indica el total de objetos encontrados en la escena.

**9.  Coordenadas de las esquinas**

Muestra las coordenadas de las esquinas de los objetos en la imagen. Las coordenadas (x,y) no representan unidades físicas en la escena, sino que representan la ubicación de un pixel en la imagen de profundidad. El orden es: xizquierda, yizquierda, xiderecha, yderecha.

**10.  Distancia a cada objeto (mm)**

Este valor indica la distancia dada por el Kinect al centro de cada objeto.

**11.  Posición**

La posición de un objeto en la imagen puede estar dada por: izquierda, derecha, centro, centro-izquierda, centro. derecha, u obstáculo total. La delimitación se hace a partir de controles de los márgenes de división vertical izquierdo y derecho.

## **AUTORÍA**

Latacunga, Abril del 2013

### **ELABORADO POR:**

---

Nancy Velasco E.

### **APROBADO POR:**

---

Ing. Nancy Guerrón  
DIRECTORA DE LA CARRERA DE INGENIERÍA EN ELECTRÓNICA E  
INSTRUMENTACIÓN

### **CERTIFICADO POR:**

---

Dr. Rodrigo Vaca  
SECRETARIO ACADÉMICO  
UNIDAD ADMISIÓN Y REGISTRO