Universidad Zaragoza
1542

Trabajo Fin de Grado

VISIÓN POR COMPUTADOR CON CÁMARA RGB-D PARA DETECCIÓN DE PERSONAS EN APLICACIONES DE ROBÓTICA

Autor

Juan Felipe Montesinos García

Director/es

José Jesús Guerrero Campos

Escuela de Ingeniería y Arquitectura

2017

RESUMEN

La normativa de robótica contempla varios criterios de seguridad que permiten prevenir accidentes en los que se vea implicado un ser humano. Hasta 2016 su aplicación se traducía en entornos de trabajo automatizados en donde no hubiese presencia humana o donde los robots estuviesen enjaulados para prevenir colisiones o lanzamientos accidentales de objetos.

La colaboración entre humanos y robots en entornos industriales es posible tras la publicación del estándar *ISO/TS 15066:2016, ROBOTS AND ROBOTIC DEVICES - COLLABORATIVE ROBOTS* [2]. En esta nueva normativa se especifican las medidas de seguridad necesarias para una interacción segura con robots industriales. Esta tendencia se ha visto potenciada por el crecimiento de la capacidad de cáculo de los ordenadores, lo que permite ofrecer seguridad mediante el procesamiento de imagen en vivo *(Computer Vision)*, capacidad de auto-aprendizaje *(Machine Learning)*, la instalación de mayor número de sensores y, por supuesto, la mejora de software y hardware de los propios robots.

El trabajo elaborado en este TFG se centra en el uso de la cámara RGB-D desarrollada por Microsoft© conocida como Kinect V2. Este tipo de cámaras permiten conocer la profundidad a la que se encuentran cada uno de los píxeles que conforman la imagen, siendo posible establecer la localización espacial de los puntos proyectados en cada uno de los píxeles desde un sistema de referencia relativo a la cámara. Si además se aplica procesado de imagen para reconocer seres humanos (o cualquier objeto u entorno que sea menester), el resultado es la localización espacial de las diferentes personas que se encuentren en el campo de visión de la cámara y, en consecución, un sistema que permita la interacción segura de un ser humano con un robot industrial.

El trabajo elaborado en este documento aborda el problema de la conexión offline entre la Kinect V2 y el robot industrial ABB. Para ello se segmenta el problema en distintos apartados. Inicialmente se realiza una documentación sobre la cámara y sus posibilidades para posteriormente evaluar de forma teórica la viabilidad de utilizarla para interactuar con un robot industrial. Hecho esto se procede a programar la cámara y todas las herramientas necesarias para detectar personas y realizar diferentes tareas como capturas de imagen, grabación de video o generación de imagen

de profundidad. El siguiente segmento a abordar es la realización de ensayos para evaluar las condiciones óptimas de localización de la cámara respecto del robot. Por último se diseña un método de calibración que permita obtener la transformación homogénea entre la Kinect V2 y el robot. El proceso de diseño consta tanto de una parte teórica, de formulación de una solución y simulaciones en Matlab para evaluar su viabilidad, como de una parte práctica que es su implementación y prueba con un robot industrial real.

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./Dª. JUAN FELIPE MONTESINOS GARCÍA,

con nº de DNI 73017814J en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Grado en ingenería de Tecnologías Industriales, (Título del Trabajo) VISIÓN POR COMPUTADOR CON CÁMARA RGB-D PARA DETECCIÓN DE PERSONAS EN APLICACIONES DE ROBÓTICA,

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 1 de septiembre de 2017

Fdo: JUAN FELIPE MONTESINOS GARCÍA

Índice General

Índice General IX

1. Introducción 1
1.1. Objetivos y Alcance 2

2. Sistemas ópticos de medición de la profundidad 3
2.1. Introducción 3
2.2. Visión estéreo 4
2.3. Luz estructurada 5
2.4. Tiempo de Vuelo, ToF 6
2.4.1. Principio físico de funcionamiento 7
2.4.2. Rango de trabajo sin ambigüedad 7
2.4.3. Características generales de la tecnología ToF 8
2.4.4. Otros sistemas de medición 9

3. KINECT V2: Tecnología y características 11
3.1. Introducción 11
3.2. Características de la Kinect V2 12
3.2.1. El sensor IR 12
3.3. Precisión de medición en profundidad 13
3.3.1. Distribución espacial del error de medición 13
3.3.2. Transitorio pre-temperatura de operación 14
3.3.3. Influencia de materiales y colores: Superficies negras y reflectantes 15
3.3.4. Eficiencia en exteriores 17

4. Análisis de viabilidad de la captura dinámica con Kinect V2 19
4.1. Introducción 19

4.1.1. Escenario dinámico 19
4.1.2. Procesamiento de imagen 19
4.1.3. Contornos de los objetos 20
4.2. Nomenclatura 20
4.3. Modelo de cámara 21
4.3.1. Cálculo de la profundidad 22
4.3.2. Obtención de coordenadas en SR Píxel 23
4.3.3. Parámetros intrínsecos típicos 24
4.4. Análisis de las limitaciones teóricas de Kinect V2 24
4.4.1. Análisis de la velocidad registrable 24
4.4.2. Análisis de tamaño de objeto 26
4.4.3. Heurística entre tamaño de objeto y velocidad 26
4.4.4. Conclusiones 28

5. Software desarrollado para Kinect V2 29
5.1. Introducción 29
5.2. Sistemas de referencia en Kinect V2 y otras consideraciones 30
5.3. Reconocimiento del esqueleto 33
5.4. Clases y funciones implementadas 34

6. Calibración del robot respecto de la cámara 37
6.1. Planteamiento de la resolución 38
6.2. Definición y análisis del algoritmo utilizado 38
6.3. Análisis numérico del algoritmo: Descripción general 39
6.4. Planteamiento, resultados y discusión de las simulaciones 41
6.4.1. Planteamiento 41
6.4.2. Resultados y Discusión 42
6.5. Implementación del método de calibración 44
6.5.1. Introducción y planteamiento 44
6.5.2. Herramienta 1, descripción y resultados 46
6.5.3. Herramienta 2, descripción y resultados 48

7. Efecto de la perspectiva en el tracking 53
7.1. Introducción 53
7.2. Experimento 1 53
7.2.1. Resultados y conclusiones 55
7.3. Experimento 2 61

8. conclusiones 63
8.1. Trabajo Futuro 64

Anexos **67**

A. Puntos utilizados en la calibración **69**

Índice de figuras

2.1. Clasificación de sistemas de medición 3
2.2. Concepto de triangulación para visión en estéreo 5
2.3. Ejemplo de patrón en visión por luz estructurada 6
2.4. Desfase entre onda emitida y recibida en tecnoloía ToF [3] 7

3.1. Kinect V2 desmontada 11
3.2. Distribución espacial del error de medición en Kinect V2: Cono elíptico [4] 14
3.3. Transitorio en la medición de profundidad de Kinect V2 [5] 15
3.4. Error de medición de la profundidad sobre un CD paralelo a la cámara [5]. 15
3.5. Relación entre el brillo de una superficie y la intensidad captada por el sensor IR en escala de grises. Escala de muestra (izquierda), intensidad en escala de grises (derecha) [5]. 16
3.6. (a) Nube de puntos en ambiente exterior. (b) Píxeles flotantes (rojo) y escena captada (blanco) [5]. 17

4.1. Representación de las bases 20
4.2. Proyección de un objeto en imagen planar 21
4.3. Proyección de espacio en el plano YZ 22
4.4. Heurística velocidad-distancia-tamaño de objeto 27

5.1. Sistema de referencia espacial de Kinect V2, situado en el centro del sensor IR 30
5.2. Esquema general de los SR en Kinect V2 31

5.3. Perspectiva del mismo escenario para los dos sensores de Kinect V2. Imagen RGB (izquierda), Imagen IR y Mapa de profundidad (derecha). 32
5.4. Posición de las articulaciones sin corregir las diferencias de resolución y escenario. Recuadrado en naranja el tamaño del mapa de profundidad 32
5.5. Esquema del sistema de seguimiento de Kinect V2 para SDK 1.8 . . . 33
5.6. Diagrama simplificado de clases y funciones utilizadas 34

6.1. Esquema de ejecución del análisis numérico sobre la capacidad de resolución de la función *fminsearch*. 40
6.2. Convergencia del algoritmo de la función *fminsearch* para la ecuación planteada 41
6.3. Error relativo en orientación entre la solución calculada y la solución exacta (izquierda, °) y espacial (derecha, mm) al aumentar el número de puntos para ruido de media 0 y desviación típica 4mm 43
6.4. Error relativo en orientación entre la solución calculada y la solución exacta (izquierda, °) y espacial (derecha, mm) al aumentar el ruido inducido. 44
6.5. Espacio de trabajo del robot visto desde la cámara. 45
6.6. Mapa de profundidad del espacio de trabajo del robot. 45
6.7. Herramienta basada en prisma cuadrangular de 29 mm de lado. Poliestireno expandido. 47
6.8. Error cometido en la calibración y la medición debido al volumen de la herramienta. 48
6.9. Proyección de la corrección sobre el SR de la cámara, aplicada a las coordenadas medidas por esta pre y post calibración. 49
6.10. Cálculo de los ángulos α y β a partir de la posición de la esfera en el espacio bajo hipótesis de medición ideal. 50

7.1. Ilustración de posiciones para la medición de perspectivas. 54
7.2. Ejemplo de correcto seguimiento para perspectivas verticales para alturas comprendidas entre 1,5 y 2,5 metros. 55
7.3. Ejemplo de incorrecto seguimiento para perspectivas verticales. 56
7.4. Tiempo necesario para realizar la primera detección para cada ubicación. En las columnas se refleja la distancia a la que se situa el usuario de la cámara. En las filas la altura a la que situa la cámara respecto del suelo. 4 medidas para cada par distancia - altura. 57

7.5. Composición de cada perspectiva de 0,5 a 1,5 metros. 58
7.6. Composición de cada perspectiva de 2 a 3 metros. 59
7.7. Composición de cada perspectiva de 3,5 a 4,5 metros. 60
7.8. Función de probabilidad acumulada (izquierda) y función de densidad (derecha) 61

Capítulo 1

Introducción

El auge de la robótica colaborativa hace necesario el desarrollo en paralelo de sistemas de protección más avanzados. Existen precedentes de accidentes fatales para el operario; como el caso de Robert Williams, empleado de Ford fallecido en Michigan en 1979 por el impacto de un brazo robótico en la cabeza o el caso de Kenji Urada, empleado de Kawasaki fallecido en 1981 realizando el mantenimiento de un robot. En general, la mayoría de accidentes suelen derivar en lesiones al usuario. A mayor integración humano-maquina las probabilidades de sufrir un accidente mortal se incrementan significativamente lo que exige mejoras acorde que satisfagan la seguridad .

En la actualidad la mejor tecnología y desarrollo de sistemas de detección va de la mano de la industria automovilística y los coches autónomos. Tanto los vehículos de Google con Waymo y los vehículos de Tesla Motors emplean sendos sistemas de cámaras y sensores que permiten, entre otras cosas, detectar personas con robustez.

En este TFG va a ahondar en la programación de la cámara RGB-D Kinect V2 de Microsoft y a proporcionar las herramientas necesarias para detectar y seguir personas incidiendo en una pequeña aplicación en el ámbito de la robótica industrial utilizando el robot IRB 120 de ABB, siendo extrapolable a diversas aplicaciones adicionales.

1.1. Objetivos y Alcance

El objetivo principal de este TFG es dominar las características y algoritmos de la Kinect V2 para lograr la detección de personas y, desarrollando un método de calibración adecuado, lograr una interacción offline con un robot. Para ello se descompone dicho proposito en los siguientes hitos:

- Realizar una revisión bibliográfica exhaustiva para comprender el funcionamiento y las prestaciones de la Kinect V2. De esta forma se pueden conocer los entornos más adecuados para su utilización y los posibles errores que se pueden encontrar.
- Realizar un estudio de viabilidad de la captura de movimiento en función de la velocidad y la distancia a la cámara para poder definir los límites teóricos de la detección.
- Generar una capa de software que permita utilizar las herramientas adecuadas para la detección, a saber: el mapa de profundidad, la imagen infraroja, la imagen RGB y los algoritmos de detección de y localización de articulaciones.
- Generar una capa de software que permita grabar, capturar imagenes y obtener las distintas herramientas para realizar experimentos.
- Desarrollar un método de calibración para obtener la rotacÂ¡ón y traslación entre la referencia de la cámara y la del robot.
- Realizar una simulación del método de calibración para determinar el efecto del error de medición y determinar el número de puntos a utilizar en la calibración.
- Realizar un experimento mediante el cual se observa el efecto de variar la perspectiva y la posición de la cámara respecto del usuario y su efecto en el posterior seguimiento.
- Realizar la calibración entre la cámara y el robot utilizando el robot ABB.

Para realizar todo ello se han utilizado tres librerías principales: NtKinect, Kinect SDK y OpenCV bajo el lenguaje de programación C++. En el apartado de calibración los datos se han exportado y tratado con MatLab.

Capítulo 2

Sistemas ópticos de medición de la profundidad

2.1. Introducción

A día de hoy existen diversas tecnologías que permiten obtener la profundidad a la que se encuentran puntos en el espacio respecto de un dispositivo:

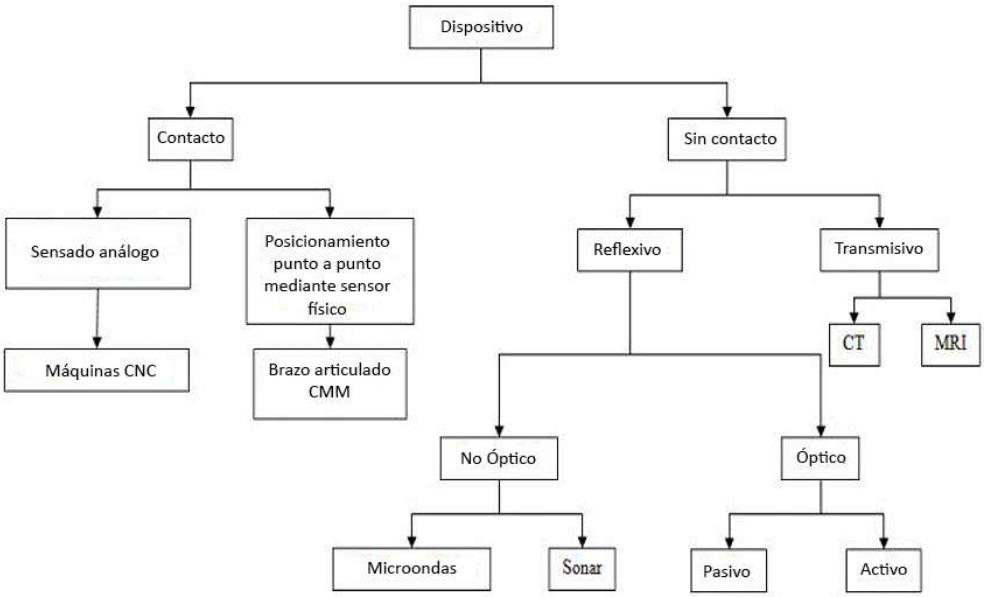

Figura 2.1: Clasificación de sistemas de medición

Dentro de la medición sin contacto (Figura 2.1) existen dos categorías, la reflexiva y la transmisiva. Reflexivo es aquel tipo de medición en el que el objeto a medir refleja ondas electromagnéticas generadas extérnamente, usualmente luz solar, luz de interior o luz infraroja generada por un emisor. Si la radiación es generada por el propio dispositivo (mediante cámaras infrarojas, por ejemplo), se denomina activo, mientras que si aprovecha la luz inherente al escenário se denomina pasivo. Transmisivo es aquel tipo de medición donde se aprovecha la radiación emitida por un cuerpo.
La diferencia entre sistemas ópticos y no ópticos es la longitud de onda empledada. Los médios no ópticos se ven limitados fuertemente por la difracción y son utilizados en sistemas GPS o SAR. Los métodos ópticos permiten alta precisión, del orden de mm. En adelante este TFG se centrará exclusivamente en métodos de medición ópticos. Los principales sistemas de medición englobados en esta categoría son:

- Luz estructurada.
- Visión estéreo.
- Interferometría
- Tiempo de Vuelo

2.2. Visión estéreo

La visión estéro se fundamenta en el análisis de un mismo escenario captado desde dos o más perspectivas diferentes utilizando diversas cámaras, cuya posición relativa ha sido previamente calibrada utilizando el modelo "pinhole"(que se expondrá posteriormente en el capítulo 4.3). Mediante el procesado por software se pueden distinguir puntos comunes del mismo escenario proyectados en diferentes imágenes y, por triangulación, deducir su localización espacial (3D). Este sistema permite utilizar sensores de muy alta resolución, con la ventaja añadida de su bajo coste y alta disponibilidad comercial. Funciona perfectamente en exteriores y evita errores típicos de otros métodos de medición, que poseen inconsistencias de medición en superficies reflectantes, absorbentes o con geometrías complejas.
Sin embargo el sistema plantea problemas ante escenarios en exceso homogéneos, pues no es posible identificar los puntos en cada una de las imágenes mediante procesado. Para ello el sistema se puede complementar con patrones luz aleatorios que permiten texturizar las zonas más homogéneas. Además es necesario que todas

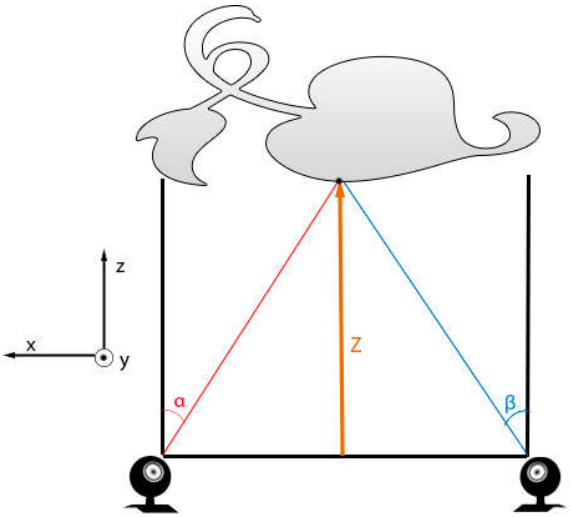

Figura 2.2: Concepto de triangulación para visión en estéreo

las zonas visibles de un escenario para un dispositivo sean visibles para al menos otro de los dispositivos de cara a poder triangular su posición.

Utilizando este método se obtiene una imagen a color y otra de profundidad ambas con la misma resolución. Con un número suficiente de cámaras se podría realizar una reconstrucción 3D.

2.3. Luz estructurada

Es un método de medición activo que utiliza un único dispositivo emisor para proyectar un patrón conocido sobre el escenario. Este patrón ha sido previamente calibrado sobre una superficie plana a una distancia fijada. A partir de las deformaciones que sufre al ser proyectado se puede calcular la distancia a la que estan cada uno de los puntos o zonas que lo conforman comparándo con el patrón calibrado. Para evitar que el usuario observe el patron este se suele emitir en el espectro infrarojo. Este patrón puede estar compuesto por cualquier distribución de elementos, desde puntos hasta lineas, cuadrados o circunferencias. La figura 2.3 ilustra un ejemplo de patron con franjas lineales.

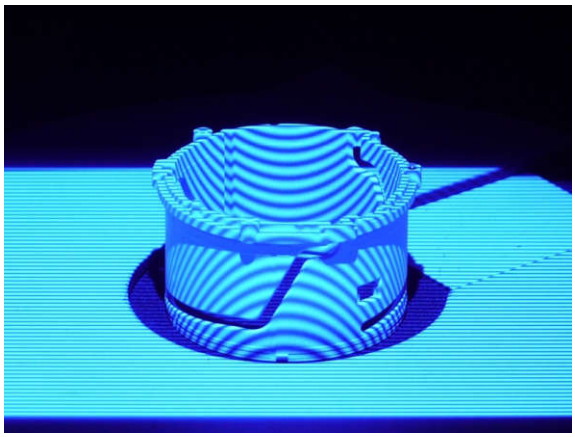

Figura 2.3: Ejemplo de patrón en visión por luz estructurada

La visión estructurada funciona muy bien en interiores y en entornos estáticos, aunque puede presentar dificultades frente a geometrías intrincadas pudiendo no llegar a detectar ciertos detalles. En exteriores, los sensores se ven saturados por la luz solar. En relación con otras soluciones como la visión estéreo o ToF presenta menor resolución, puesto que la mínima unidad de información procesable no viene dado por la resolución del sensor si no por la geometría del patrón y su procesado.

2.4. Tiempo de Vuelo, ToF

La tecnología ToF se basa en la emisión de ondas IR y la medición del tiempo de ida y vuelta de estas entre el sensor y el escenario para calcular la distancia a la que se encuentran los distintos puntos del escenario siendo conocida la velocidad de propagación de dichas ondas, es decir, la velocidad de la luz. Existen dos métodos para medir este intervalo de tiempo: el medición directa y medición indirecta.

- **Directa:** La luz se emite en pulsos cuya intensidad es medida en el sensor. La distancia a la que se encuentra el escenario es directamente proporcional a la intensidad lumínica.

- **Indirecta:** Se emiten ondas IR moduladas en frecuencia y se mide el desfase entre la onda emitida y la onda recibida. La profundidad es inferible a partir de este desfase. Este es el método más utilizado.

Por ser esta la tecnología utilizada en este TFG será explicada en mayor profundidad.

2.4.1. Principio físico de funcionamiento

La medición de la distancia a la que se encuentra un objeto se basa en el desfase entre la onda emitida y la onda recibida, de acuerdo a la Figura 2.4.

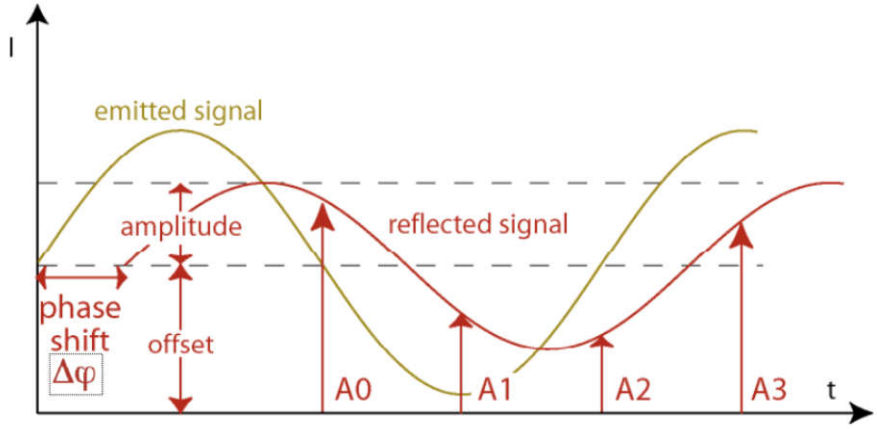

Figura 2.4: Desfase entre onda emitida y recibida en tecnoloía ToF [3]

De esta forma el desfase se calcula como:

$$d = \frac{c \cdot \Delta\varphi}{4\pi f_{mod}} \tag{2.1}$$

Donde f_{mod} es la frecuencia de modulación (frecuencia a la que se emite en IR), c es la velocidad de la luz y $\Delta\varphi$ el desfase. Cabe resaltar que d es la distancia bruta entre un punto y su correspondiente proyección en un píxel físico y no la profundidad, que sería la proyección de dicha distancia sobre el eje Z. Este cálculo se realiza para cada uno de los píxeles del sensor. Es trivial observar que el desfase de una onda es una función periódica, ergo existe un rango de funcionamiento limitado por un desfase máximo 2π. Para ángulos superiores se produce ambigüedad puesto que trigonometricamente los ángulos φ y $(\varphi + 2\pi k)$ son equivalentes.

2.4.2. Rango de trabajo sin ambigüedad

Como se ha explicado en la sección anterior la señal de modulación es una señal periódica, por lo que el desfase estará comprendido entre 0° y 360°. Es por ello que cualquier objeto situado a una distancia superior a la máxima medible para este intervalo dará como resultado un error de medida. Se puede demostrar [6] que esta

distancia es:

$$d_{max} = \frac{c}{2f_{mod}} \tag{2.2}$$

Por lo que es posible tabular la distancia máxima medible en función de la frencuencia de modulación.

f_{mod} (MHz)	40	35	30	25	21	20	19	18	17	16
f_{mod} (MHz)	3,75	4,28	5	6	7,14	7,5	7,89	8,33	8,82	9,37
f_{mod} (MHz)	15	14	13	12	11,5	11	10,1	10	9,5	9
f_{mod} (MHz)	10	10,71	11,54	12,5	13,04	13,63	14,28	15	15,79	16,66
f_{mod} (MHz)	8	7	6	5	4	3	2	1	0,5	0,1
f_{mod} (MHz)	18,75	21,43	25	30	37,5	50	75	150	300	1500

Tabla 2.1: Tabla paramétrica de distancia máxima medible y frecuencia de modulación necesaria [6].

Si bien a priori puede parecer que utilizar una baja frecuencia de modulación mejora las prestaciones del dispositivo se ha de considerar que ha de existir una intensidad mínima de la señal recibida para que esta sea captada por el sensor y que, además, la precisión disminuye al disminuir la frecuencia de modulación.

Es posible trabajar con varias frecuencias de modulación (si así lo permite el equipo) para aumentar la distancia máxima medible sin ambigüedad a la par que se mantiene la precisión. Para ello se filtran aquellas frecuencias que no sean de interés a fin de no interferir unas con otras y evitar ruido proveniente del espectro IR de la luz solar. Utilizando una frecuencia baja que permita medir distancias elevadas y una frecuencia alta que aporte precisión se puede determinar en que ciclo periódico está midiendo cada frecuencia y aprovechar los datos para paliar las desventajas de aumentar o disminuir la frecuencia de modulación. Estas características dependerán del número de frecuencias utilizadas y su longitud de onda.

2.4.3. Características generales de la tecnología ToF

La tecnología ToF genera un mapa denso en todo el escenario y calcula la distancia a la que se encuentra la proyección de cada punto para cada píxel. Por ello ofrece más precisión que la luz estructurada y tanta precisión o más que la visión estéreo, dependiendo de la resolución del sensor. Funciona en entornos exteriores no saturados, aunque de ser posible es preferible eliminar el factor luz solar, pues

induce ruido en el sensor. Esta luz de fondo suele ser filtrada.
Si bien no le afectan los escenarios homogéneos sí lo hacen las superficies reflectantes, provocando altos errores de medición; así como las superficies absorbentes, que impiden que se refleje la radiación con suficiente intensidad como para ser captada por el sensor.
La resolución y la precisión aumentan al aumentar la frecuencia de modulación y al aumentar la potencia de iluminación, por lo que entornos con luz homogénea y bien limitados son adecuados. También es notable el ángulo de incidencia de la luz IR Aunque se ha hablado de las bondades de aumentar la frecuencia de modulación esto dificulta el tomar un buen contraste de la imagen y valores excesivamente altos disminuyen la precisión de la medida. Por último cabe mencionar que se suelen realizar varias medidas para aumentar la precisión, por lo que en entornos dinámicos se encuentra en desventaja frente a otras tecnologías que solo necesitan de una única.

2.4.4. Otros sistemas de medición

Se ha hablado previamente de la interferometría y la triangulación laser. La interferometría consiste en la emisión de ondas mediante dos focos coherentes a una distancia conocida para formar un patrón de interferencia. Utilizando longitudes de onda muy cortas pueden detectarse diferencias en la trayectoria seguida por ambos rayos lo que implica una longitud. Es de muy alta precisión pero tiene tiene utilidad únicamente para la medición de distancias muy pequeñas, del orden de los cm, por lo que se emplea en procesos de calidad. Fue una técnica muy potenciada con el desarrollo del laser.

Capítulo 3

KINECT V2: Tecnología y características

3.1. Introducción

Kinect V2 es un dispositivo desarrollado por Microsoft como complemento para la consola Xbox One. Fue lanzado el 22 de noviembre de 2013 y posteriormente se creó un SDK para desarrolladores conocido como Kinect for windows V2 SDK. El dispositivo cuenta con dos sensores, uno infrarojo y otro RGB (Figura 3.1), además de una serie de microfonos que permiten conocer la dirección de la que proviene el sonido. El sensor infrarojo permite medir la profundidad mediante la tecnología ToF *(Time of Flight)*. La innovación principal de esta cámara radica en el bajo precio y la difusión que ha tenido respecto a otros modelos enfocados a entornos profesionales con precios prohibitivos y prestaciones similares.

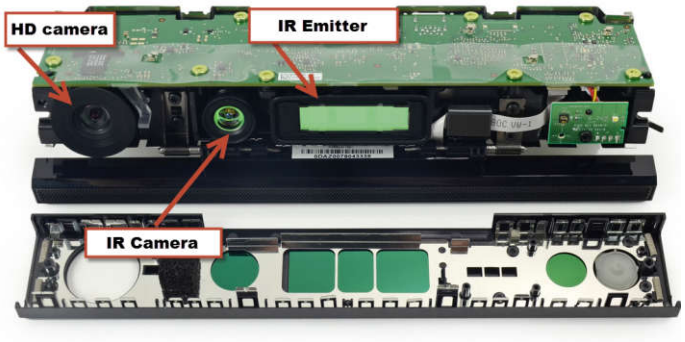

Figura 3.1: Kinect V2 desmontada

3.2. Características de la Kinect V2

Las especificaciones de Kinect V2 son las siguientes: [7]

Resolución RGB [pixeles]	1920 x 1080 30 Hz
Resolución IR [pixeles]	512 x 424 30 Hz
Máx. profundidad medible	4.5 m
Min. profundidad medible	0.5 m
FoV horizontal x vertical	70 x 60 °
USB estándar	3.0
Sistema operativo	Windows 8, 8.1, 10

Tabla 3.1: Especificaciones Kinect for Windows V2

Como se puede observar el sensor RGB cuenta con una resolución full HD perfecta para cualquier aplicación, sin embargo el sensor IR tiene una resolución bastante más pobre, con un ratio $\frac{rRGB}{rIR} \approx 4$ lo que implica una correspondencia de 4 píxeles RGB por cada píxel IR lo que a la postre merma las aplicaciones de la cámara.

3.2.1. El sensor IR

El sensor IR de la Kinect V2 monta una resolución de 512 x 424 píxeles utilizando diodos tipo CMOS (Tabla 3.2). Cada píxel esta compuesto por dos fotodiodos controlados por la misma señal de reloj, de tal forma que si uno está encendido el otro está apagado. Operando ambas señales se puede obtener la profundidad, la imagen a escala de grises y la imagen a escala de grises con independencia de la luz ambiente. [8]

Tecnología	TSMC 0.13 1P5M
Resolución	512 x 424
Tamaño de chip	8,2 x 14,2 mm^2
Rango dinámico	>2500 = 68 dB
Modulación de contraste	68 % @ 860 nm @ 50MHz
Modulación de frecuencia	10 - 130 MHz
Promedio de modulación de frecuencia	80 MHz
FOV	70 (h) x 60(V) °
Error de profundidad	<0.5 % del rango
Rango de medición	0.5 - 4.5 m
Longitud de onda	860 nm
FPS	30 fps
ADC	2 GS/s
Reflectividad	15 % - 95 %
Potencia del chip	2,1 W

Tabla 3.2: Características del sensor IR de Kinect V2 [9].

En la Tabla 3.2 se exponen algunas de las características principales del sensor. Se utiliza un filtro de banda estrecha para filtrar toda la luz cuya longitud de onda sea distinta de los 860 nm. que por defecto usa Kinect V2 y realiza varias mediciones para cada píxel eligiendo el mejor valor.

3.3. Precisión de medición en profundidad

Como en toda medición la Kinect V2 esta sujeta a un rango de trabajo a través del cual varía la precisión y el error cometidos al realizar la medición. En esta sección se plasman resultados de diferentes experimentos realizados con el fin de exponer parámetros de interes al trabajar con Kinect V2. Para una documentación más extensa sobre la realización de estos referirse a la bibliografía citada.

3.3.1. Distribución espacial del error de medición

El error de medición de la Kinect V2 satisface un cono elíptico de 70 x 60 ° de acuerdo a la figura.

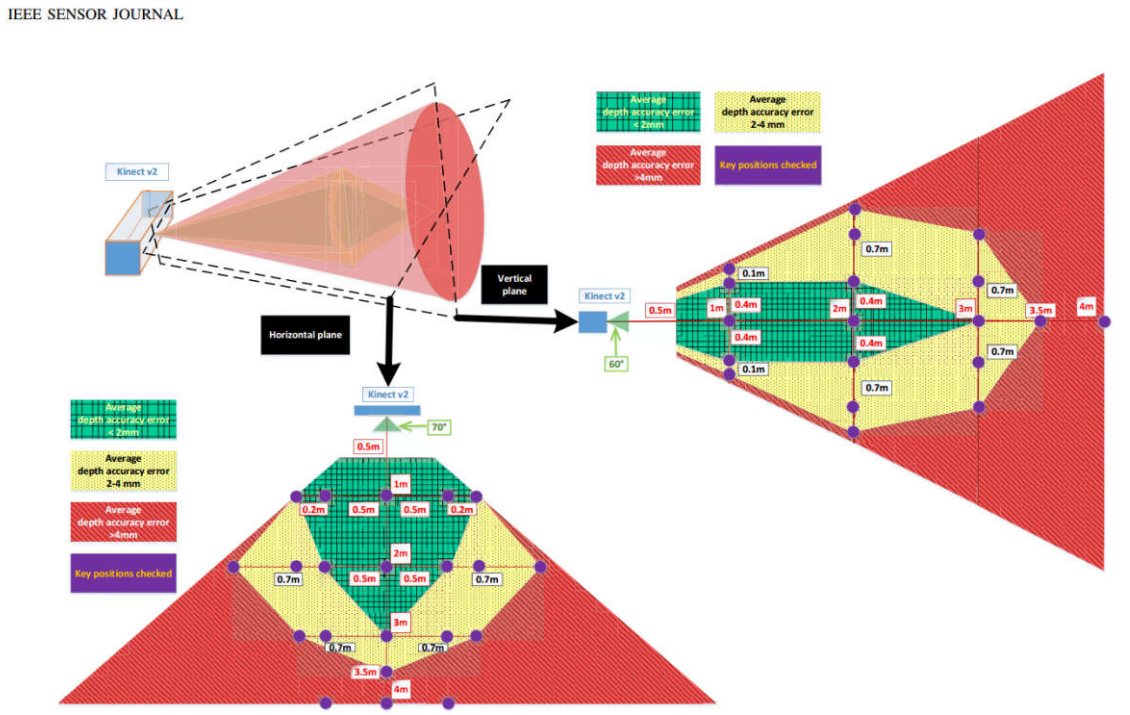

Figura 3.2: Distribución espacial del error de medición en Kinect V2: Cono elíptico [4]

El error cometido en la zona correspondiente al penacho verde es inferior a 2 mm, y cuando más nos alejemos de este mayor es el error de medición, que esencialmente aumenta al aumentar la profuncidad llegando a superar los 4mm a parir de los 3,5 m de distancia. Según especificaciones de Microsoft el error es menor que el 0.5 % del rango, aproximadamente 20 mm.

3.3.2. Transitorio pre-temperatura de operación

El sensor se ve sometido a un periodo transitorio debido al calentamiento de los diferentes componentes hasta alcanzar un régimen estacionario con temperatura estable [5]. Este periodo transitorio asume una distribución logarítmica durante la cual las mediciones de profundidad varían notablemente. Dicho comportamiento se puede observar en la figura 3.3, en la cual la distancia varia desde 5 hasta 30 mm tras 30 minutos de uso hasta volverse aproximadamente constante.

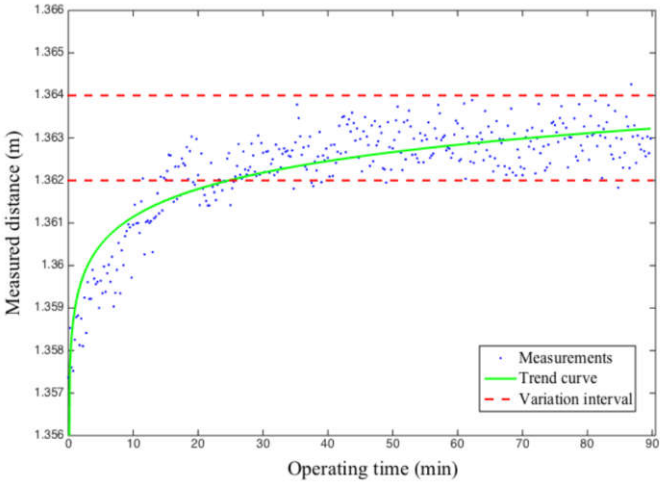

Figura 3.3: Transitorio en la medición de profundidad de Kinect V2 [5]

3.3.3. Influencia de materiales y colores: Superficies negras y reflectantes

Se estudia la influencia de diferentes materiales enfocandose en la rugosidad, el color y la reflectancia. Tanto los materiales negros como aquellos con superficies reflectantes reflejan una onda con intensidad bastante baja, lo que se traduce en elevados errores en la medición de la profundidad, con distancias superiores a lo esperado. Este efecto queda patente en la Figura 3.4, donde el error de medición alcanza los 6 cm mientras que el orden de magnitud típico es de 2 mm.

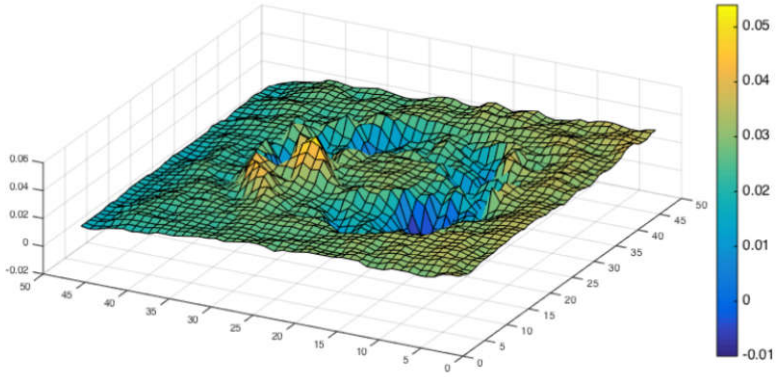

Figura 3.4: Error de medición de la profundidad sobre un CD paralelo a la cámara [5].

Los efectos del brillo del material se comprueban utilizando una escala de muestra 3.5. La intensidad incrementa al incrementar el brillo de la superficie del material (salvo que este sea transparente o reflectante).No hay resultados concluyentes en cuanto a la influencia de la rugosidad del material.

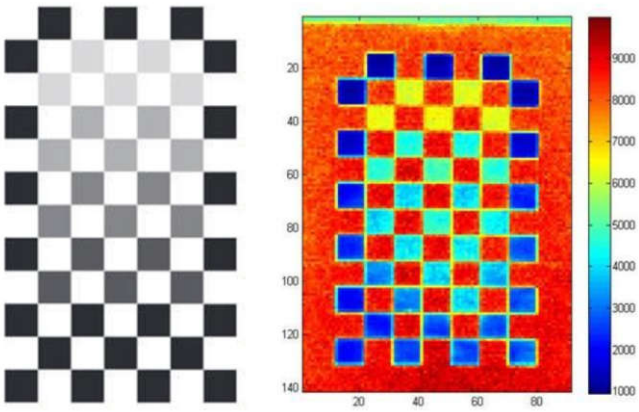

Figura 3.5: Relación entre el brillo de una superficie y la intensidad captada por el sensor IR en escala de grises. Escala de muestra (izquierda), intensidad en escala de grises (derecha) [5].

3.3.4. Eficiencia en exteriores

Como ya se explicó en el apartado 2.4 la luz solar posee radiación en la frecuencia de trabajo de Kinect V2, lo que induce ruido. Sin embargo la cámara puede operar a la luz del Sol siempre y cuando no esté bajo iluminación directa, puesto que esto provoca en muchos casos la desconexión del sensor [5]. En condiciones de penumbra se observa el sensor genera píxeles flotantes (no correspondientes a ningún punto real), particularmente en los límites del campo de visión.

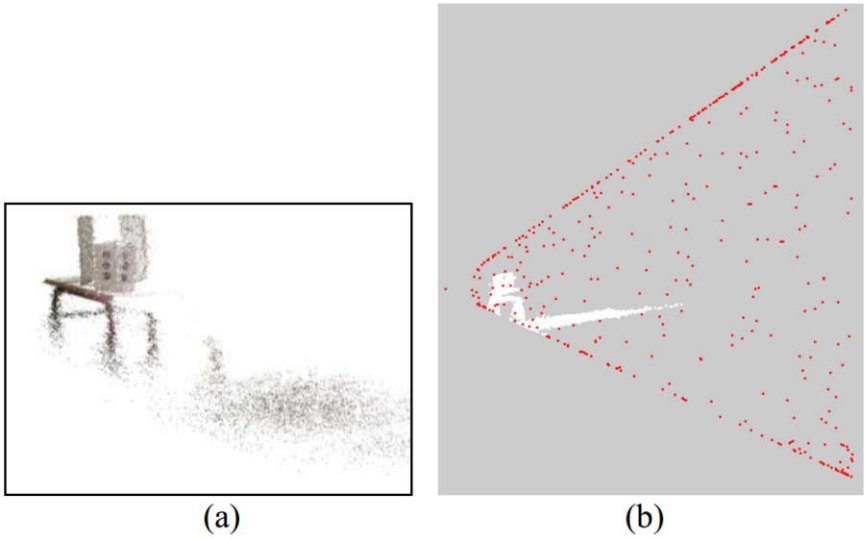

Figura 3.6: (a) Nube de puntos en ambiente exterior. (b) Píxeles flotantes (rojo) y escena captada (blanco) [5].

Capítulo 4

Análisis de viabilidad de la captura dinámica con Kinect V2

4.1. Introducción

4.1.1. Escenario dinámico

La captura realizada con la Kinect V2 asume un entorno estático. El sistema de procesado de la Kinect V2 realiza varias mediciones de la profundidad por cada pixel, para lo cual se supone que durante el proceso de adquisición cada pixel se corresponde con un único punto y este no varía durante el procesado.

El hecho de realizar una captura dinámica rompe dicha premisa lo que da lugar a unas limitaciones de velocidad que guardan relación directa con la distancia a la que se encuentre el objeto a capturar. Es decir, existe una Velocidad máxima Vmax = f(Z).

4.1.2. Procesamiento de imagen

Por otro lado existe una limitación en el mínimo tamaño de objeto que se puede capturar. Esta limitación se genera por el procesamiento de la imagen. Un parámetro adecuado (y arbitrario) a considerar es un tamaño mínimo de objeto, una vez proyectado en el sensor, de 5-10 píxeles.

Además, el tamaño correspondiente en la imagen planar de un objeto en el espacio

es inversamente proporcional a la distancia entre el objeto y la cámara.

4.1.3. Contornos de los objetos

El lugar geométrico de los puntos conocido como contorno tiene la caracteística de estar situado a dos distancias de la cámara simultáneamente, la del punto perteneciente al objeto y la del punto perteneciente al plano por detrás de este. Esta circunstancia hace que en la realidad se obtenga información mezclada de ambas profundidades, dando lugar a incosistencias de medida.

4.2. Nomenclatura

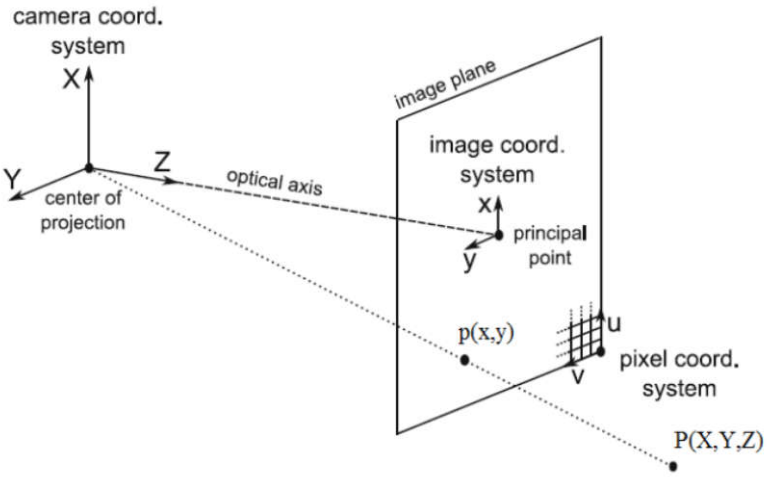

Figura 4.1: Representación de las bases

En adelante nos referiremos a un sistema de referencia con las siglas SR. La figura 4.1 ilustra los distintos sistemas de referencia utilizados en la explicación conceptual del funcionamiento de la Kinect V2. [9]
SR Principal : SR de referencia definido por la Base Principal y el punto conocido como centro de proyección.

Base Principal: Base ortonormal en R^3 definido como $B_p<\hat{i},\hat{j},\hat{k}>$. Denotaremos como P(X,Y,Z) a las coordenadas de un punto en el espacio, en SR principal.

SR Pixel : SR de referencia definido por la Base Píxel y un punto arbitrario de la imagen planar, típicamente situado en alguna de las esquinas de la imagen planar.

Base píxel Sistema ortonormal de referencia en R^2 definido como $B_{px}<\hat{u},\hat{v}>$ en la figura 4.1. Denotaremos como px(u,v,1) a las coordenadas de la proyección de un punto del espacio en la imagen planar, en SR pixel.

Base imagen Sistema ortonormal de referencia en R^2 definido como $B_{im}<\hat{i},\hat{j}>$ en la figura 4.1. Denotaremos como p(x,y,1) a las coordenadas de la proyección de un punto del espacio en la imagen planar, en SR imagen. Se puede observar que el punto principal es la proyección del Centro de Proyección sobre la imagen planar el cual se corresponde a las coordenadas p(0,0,1).

4.3. Modelo de cámara

El modelo *Pinhole camera* es un modelo ideal de fotografía en el cual todos los haces de luz provenientes de un objeto se cortan en un único punto conocido como *Centro de Proyección*. Para capturar posteriormente la imagen basta con colocar una superficie plana de proyección a cualquier distancia del centro de proyección. Dicha superficie se denomina imagen planar.

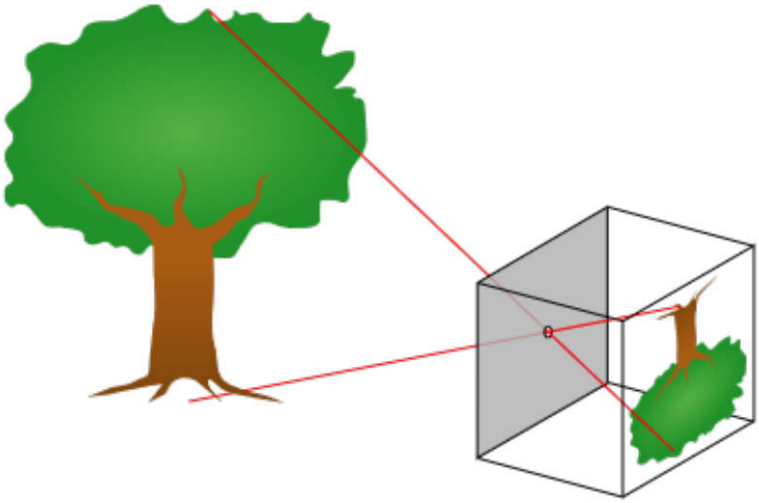

Figura 4.2: Proyección de un objeto en imagen planar

Matemáticamente el modelo pinhole utiliza una transformación sobreyectiva q: $R^3 \longrightarrow R^2$ para definir la proyección de un punto P en el espacio sobre un plano. En términos físicos dicho plano sería la película o chip fotosensible de la cámara.

Se determina el espacio vectorial de R^3 mediante un sistema generador. Por simplicidad, se toma como dicho sistema la base canónica de R^3, a la que se denota como $B_p = \{\hat{i},\hat{j},\hat{k}\}$. La dirección del vector $\hat{k}$ es la perpendicular a la imagen planar, como se observa en la Figura 4.1.

Se define el SR principal a partir de la base B_p y el centro de proyección C. El sentido del vector $\hat{k}$ sera de C a la imagen planar. Los vectores $\hat{j}$ e $\hat{i}$ se seleccionan de tal forma que el SR sea dextrógiro. En el caso de la Kinect V2 el vector $\hat{j}$ es perpendicular al suelo. Los vectores $\hat{u}$ y $\hat{v}$ son pararelos a los vectores $\hat{i}$ y $\hat{j}$ respectivamente.

4.3.1. Cálculo de la profundidad

Si se toma una proyección bidimensional del espacio R^3 y algunos de sus puntos característicos en el plano YZ se obtiene el esquema de la Figura 4.3

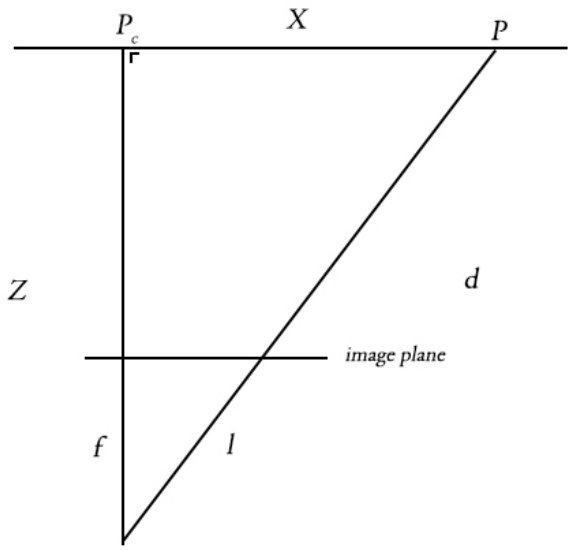

Figura 4.3: Proyección de espacio en el plano YZ

Siendo f la *distancia focal* (distancia de C a su proyección en la imagen planar) y d la distancia desde el *Centro de Proyección* hasta un punto P generico, por semejanza de triangulos se obtiene que:

$$\frac{f}{l} = \frac{Z}{d} \Rightarrow Z = d\frac{f}{l} = d\frac{f}{\sqrt{f^2 + x^2}} \tag{4.1}$$

De esta forma se puede obtener la coordenada Z (profundidad) de un punto mediante parámetros puramente geométricos.

4.3.2. Obtención de coordenadas en SR Píxel

Dado un punto P en el espacio y su proyección p_i en la imagen planar, siguiendo un razonamiento análogo al planteado en la ecuacion 4.1 se puede deducir que:

$$x = f\frac{X}{Z}; y = f\frac{Y}{Z} \tag{4.2}$$

El cambio de base entre el SR pixel y el SR imagen se obtiene de las siguiente expresiones:

$$u = k_u(x + x_o) = k_u \cdot f\frac{X}{Z} + k_u \cdot x_o \tag{4.3}$$

$$v = k_v(y + y_o) = k_v \cdot f\frac{Y}{Z} + k_y \cdot y_o \tag{4.4}$$

Donde k_u y k_v son la densidad lineal de píxeles en las direcciones principales. Es decir, pixeles/espacio.
De esta forma la aplicación q se puede definir a partir de las ecuaciones 4.3 y 4.4 como:

$$q : \vec{p} = \bar{\bar{K}}\vec{P} \tag{4.5}$$

$$q : \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{Z}\begin{pmatrix} k_u & 0 & k_u x_o \\ 0 & k_v f & k_v y_o \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \tag{4.6}$$

Como se puede observar la matriz K es una matriz de constantes que depende de parámetros puramente geométricos, denominados *parámetros intrínsecos de la cámara*, cuyas unidades son los píxeles. Reescribiendo la ecuacion 4.6:

$$q : \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{Z}\begin{pmatrix} \alpha_u & 0 & u_o \\ 0 & \alpha_v & v_o \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \tag{4.7}$$

4.3.3. Parámetros intrínsecos típicos

El estudio de prestaciones de la Kinect V2 se realizará mediante un estudio numérico utilizando los siguientes parámetros [5] como representativos para la Kinect V2.

{píxeles}	α_u	α_v	u_o	v_o
RGB	365	365	255	204
IR	365	365	251	207

Tabla 4.1: Parametros caracteristicos

4.4. Análisis de las limitaciones teóricas de Kinect V2

4.4.1. Análisis de la velocidad registrable

Como en todo sensor del mercado la Kinect V2 esta sujeta a una velocidad máxima de movimiento de los objetos dentro de su campo de visión antes de que se produzca desenfoque. Esta velocidad esta directamente ligada a la *velocidad de obturación* de la cámara, parámetro que indica (para cualquier cámara) la inversa del tiempo durante el cual está abierto el obturador. Cuanto mayor sea esta velocidad de obturación mayor podrá ser la velocidad de la imagen sin producirse desenfoque. A la par esta velocidad es directamente proporcional a la distancia a la que se encuentre el objeto a captar puesto que, cuanto más lejos se encuentre, menor será la velocidad efectiva que tenga que captar el sensor.

La derivada temporal de un vector expresador en una referencia Ref_1, respecto a una referencia Ref_2 que se mueve recpecto de Ref_1 es [10]:

$$\{\frac{d}{dt}\vec{u}|_{Ref_1}\}_{Ref_2} = \frac{d}{dt}\{\vec{u}\}_{Ref_2} + \{\vec{\Omega}_{Ref_1}(Ref_2) \times \vec{u}\} \tag{4.8}$$

Asumiendo una posición fija o estática para la cámara durante la captura de imagen, la derivada temporal del vector posición se puede expresar como la derivada de cada uno de sus términos.

Cabe descatar que el estudio analiza la velocidad en el plano XY, por lo que la coordenada Z se puede considerar constante.
De esta forma se obtiene:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ 0 \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} \alpha_u & 0 & u_o \\ 0 & \alpha_v & v_o \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{X} \\ \dot{Y} \\ 0 \end{pmatrix} \tag{4.9}$$

Aproximando la derivada temporal como incremento por unidad de tiempo:

$$\frac{d}{dt} \approx \frac{\Delta \vec{u}}{T} \tag{4.10}$$

Y sabiendo que la camara muestrea a 30 FPS T = 1 / frec, se puede hallar el desplazamiento (en píxeles) que se produce entre dos fotogramas para un objeto con movimiento el el plano XY a una distancia Z. El tomar como velocidad de obturación la frecuencia de captura de imagenes es la peor de las situaciones posibles, por lo que cualquier resultado práctico posterior sólo puede ser mejor que este.
Combinando las ecuaciones 4.9 y 4.10 se obtiene que:

$$\dot{\vec{p}} \approx frec \cdot \Delta \vec{p} \Rightarrow \Delta \vec{p} = \frac{1}{Z \cdot f_m} \begin{pmatrix} \alpha_u & 0 & u_o \\ 0 & \alpha_v f & v_o \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{X} \\ \dot{Y} \\ 0 \end{pmatrix} \tag{4.11}$$

Como criterio para no generar pérdida de resolución se establece Δp $\leq$ 1 píxel. Se puede calcular la velocidad máxima del punto P en función de la distancia Z. Reescribiendo la ecuacion 4.11

$$V_{max} = f(Z)$$

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ 0 \end{pmatrix} = Z \cdot f_m \begin{pmatrix} 1/\alpha_u & 0 & -u_o/\alpha_u \\ 0 & 1/\alpha_v & -v_o\alpha_v \\ 0 & 0 & 1 \end{pmatrix} \Delta \vec{p}, \Delta \vec{p} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \tag{4.12}$$

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = Z \cdot f_m \begin{pmatrix} 1/\alpha_u \\ 1/\alpha_v \end{pmatrix} \tag{4.13}$$

Asumiendo $\alpha_u \approx \alpha_v$ se puede considerar una velocidad genera V tal que

$$V_{max} = Z \cdot \frac{f_m}{\alpha} = 0{,}0822Z \tag{4.14}$$

4.4.2. Análisis de tamaño de objeto

Es trivial observar que se puede utilizar la transformación no solo para la proyección de puntos en la imagen planar si no para la proyección de dimensiones.

$$q : \begin{pmatrix} s_u \\ s_v \\ 1 \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} \alpha_u & 0 & u_o \\ 0 & \alpha_v f & v_o \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} S_x \\ S_y \\ 1 \end{pmatrix} \tag{4.15}$$

De esta forma se pueden transformar longitudes en el plano XY en longitudes en píxeles en la imagen planar.

Asumiendo que los algoritmos de detección van a restar resolución a la imagen se considera empíricamente que el mínimo tamaño de detección de una longitud en la imagen planar es de 5 píxeles.

4.4.3. Heurística entre tamaño de objeto y velocidad

A la hora de seleccionar una distancia a la que posicionar el objeto a capturar se genera una dicotomía. Por un lado, es conveniente una baja velocidad de movimiento en la imagen planar para evitar desenfoque, lo que conlleva alejar el objeto de la cámara y, por otro, es conveniente que el objeto proyectado en la imagen planar sea del mayor tamaño posible de cara a aumentar la resolución y consecuentemente favorecer el trabajo de los algoritmos de detección, para lo cual es conveniente acercar el objeto a la cámara.

Al ser un problema bidimensional obtendríamos diferentes soluciones dependiendo de si tratamos una longitud en el eje X o Y, tomamos el peor de los casos que es una longitud en el eje Y, puesto que es la dirección en la que la cámara cuenta con menor resolución. De esta forma:

$$s_v = \frac{\alpha_v \cdot S_v + v_o}{Z} \tag{4.16}$$

Del análisis de velocidad se deduce que:

$$V_{max} = f(Z) \Leftrightarrow Z = f(V_{max})$$

De esta forma se puede generar una función:

$$s_v = f(V_{max}, S_y) \tag{4.17}$$

Como se puede observar en la ecuación 4.17 el tamaño máximo que puede ocupar un objeto sin provocar desenfoque limita la velocidad (máxima) a la que se puede desplazar el objeto y acota el tamaño que puede tener en función de la distancia.

Por consiguiente y tras las suposiciones realizadas anteriormente se obtiene:

$$s_v = \frac{\alpha \cdot S_y + v_o}{\alpha \cdot V_{max}} f_{max} \tag{4.18}$$

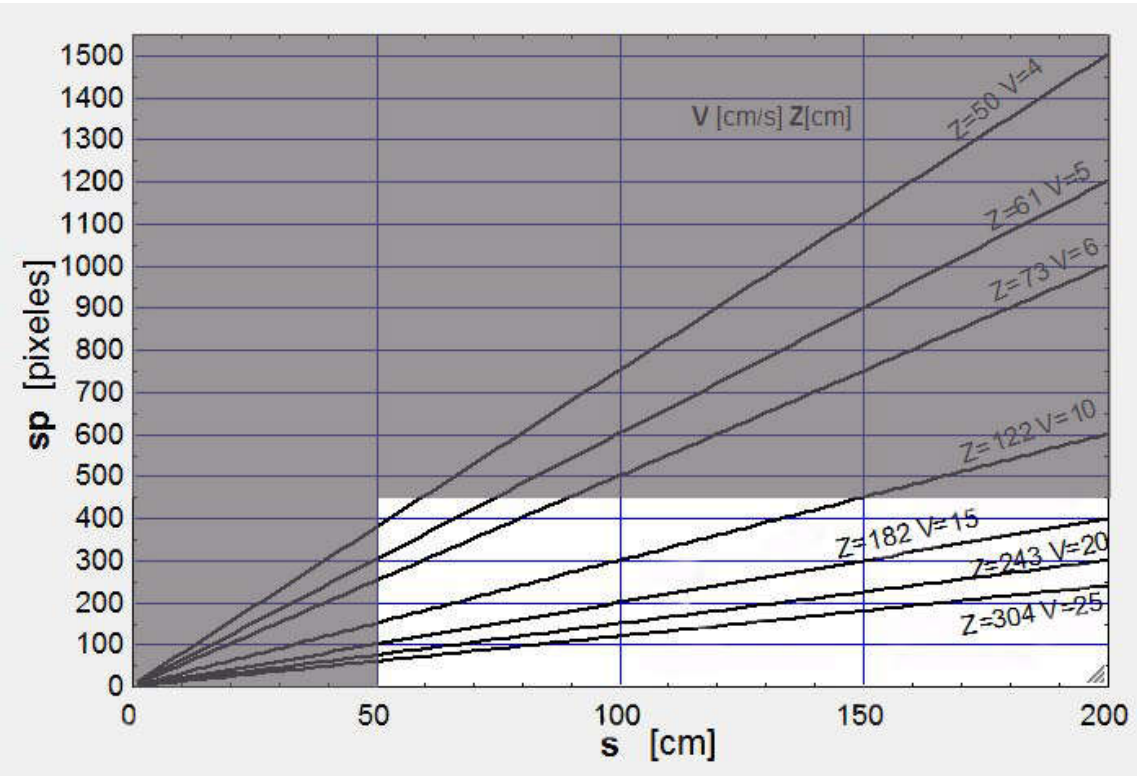

Figura 4.4: Heurística velocidad-distancia-tamaño de objeto

La ecuación 4.18 no es válida para todo el rango por limitaciones de la Kinect V2. Recordamos la mínima distancia a la que puede medir profundidad es 0,5 m. A su vez la resolución vertical del sensor de profundidad es 424 píxeles.

4.4.4. Conclusiones

Existen unas limitaciones físicas causadas por el sensor y otras limitaciones causadas por el procesado de imagen que dependen fuertemente de dos parámetros: la profundidad a la que se encuentre el objeto y la velocidad a la que se mueva. Una menor distancia del objeto a la cámara produce que esta ocupe mayor tamaño en la imagen, lo que facilita su procesado software y disminuye el error cometido por el sensor al calcular la profundidad. Por contra, una mayor cercanía a la cámara reduce la velocidad máxima a la que el objeto se puede desplazar sin provocar desenfoque y limita el tamaño de objeto que cabe en pantalla.
Valorando los resultados se observa que (aproximadamente) una persona de 1.90 m situada a una distancia de 2.5m de la cámara abarcaría la practica totalidad del sensor y podría moverse a una velocidad de 20 cm/s, lo cual limita las potenciales aplicaciones de la Kinect V2. Como dato comparativo la raqueta de un tenista profesional puede alcanzar los 120 km/h $\approx$ 32 m/s, [11] 2 órdendes de magnitud superior a lo que la cámara puede captar. Si bien puede parecer que la Kinect V2, a priori, cuenta con fuertes limitaciones cabe destacar que estas se ven suplidas por el sensor RGB y aunque no sea posible obtener datos durante la ejecución de un movimiento a alta velocidad es posible recuperar el tracking del objeto instantes después y seguir el movimiento en la imagen RGB.
La distancia ideal es por lo tanto la más cercana posible a la cámara permitiendo movimientos sin generar desenfoque y captando el objeto entero.

Capítulo 5

Software desarrollado para Kinect V2

5.1. Introducción

La programación de Kinect V2 realizada en este TFG se fundamenta en tres pilares esenciales: el SDK Kinect v2 creado por Microsoft, la librería NtKinect creada por Yoshihsa Nitta [1] y las librerías OpenCV 3.1. Se ha empleado lenguaje C++ para poder implementar la librería OpenCV y el entorno de desarrollo Visual Studio 2017 desarrollado también por Microsoft. Mientras que el SDK proporciona las herramientas adecuadas para controlar la cámara la librería NtKinect auna todo el codigo implícito en el SDK en funciones simples y concisas en aras de una mayor claridad y menor longitud del código. La librería OpenCV proporciona las herramientas para un adecuado tratamiento de la imagen recibida de la cámara.

El objetivo final es programar las herramientas necesarias para obtener los tres tipos de imágenes existentes para Kinect V2: la imagen a color, la imagen infraroja y el mapa de profundidad y, a partir de ella, mediante algoritmos de visión por computador y programación, reconocer el esqueleto, gestos realizados con las manos y una interfaz operada por comandos que nos permita realizar la calibración y la experimentación previamente explicados. Para alcanzar estos objetivos se ha buscado la descomposición del problema en hitos sencillos:

1. Obtener imagen en tiempo real RGB, IR y de profundidad por separado en una interfaz simple.

2. Capturar imágenes y video.
3. Reconocer el esqueleto y gestos con la mano.
4. Interfaz que permita obtener las coordenadas de la imagen utilizando el cursor y almacenarlas en archivos de texto mediante clicks.
5. Programa para realizar la calibración.
6. Programa para realizar el experimento angular.
7. Integración conjunta y comandos por teclado.

5.2. Sistemas de referencia en Kinect V2 y otras consideraciones

En esta sección se va a utilizar la nomenclatura citada en el apartado 4.2: Nomenclatura. La Kinect V2 utiliza dos sensores con diferentes resoluciones, lo que implica dos SR pixel distintos, uno para cada sensor. Adicionalmente la cámara cuenta con un SR Principal, lo que hacen un total de 3 SR distintos. Las estructuras de datos creadas por Microsoft para cada SR se encuentran en el encabezado "Kinect.h".

El SR principal se corresponde con el SR espacial de la cámara. Su origen está situado sobre el centro del sensor IR con la dirección y sentidos definidos en la Figura 5.1. Este SR se asocia con la estructura *CameraSpacePoint*, que permite almacenar tres coordenadas en variables tipo float. Cualquier coordenada espacial vendrá definida sobre este SR y con esta estructura de datos. Cabe destacar que los datos son adquiridos y representados como si de una imagen reflejada en un espejo se tratase.

Figura 5.1: Sistema de referencia espacial de Kinect V2, situado en el centro del sensor IR

Puesto que hay dos sensores y dos SR tipo píxel, uno para la imagen RGB y otro para la imagen IR/Depth Microsoft define dos estructuras de datos: *ColorSpacePoint* para la imagen RGB y *DepthSpacePoint* para la imagen IR/Depth. Los SR píxel se orientan como muestra la Figura 5.2, con origen en la esquina superior izquierda.

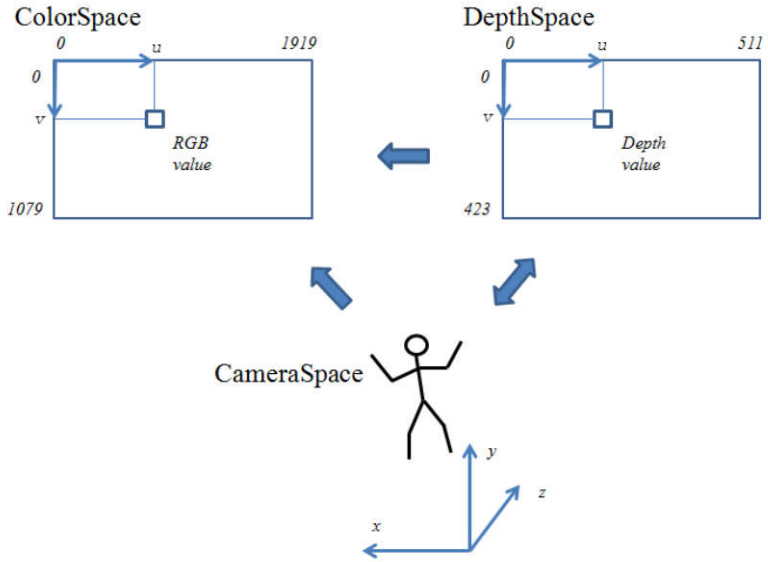

Figura 5.2: Esquema general de los SR en Kinect V2

Kinect V2 requiere de un mecanismo para calcular la equivalencia entre los 3 SR debido, por un lado, a la diferente resolución de los sensores, y por otro, al hecho de que estos enfocan distintos escenarios ya que poseen distinto campo de visión y no se situan espacialmente en el mismo punto. El SR principal (*CameraSpacePoint*) se relaciona con el SR Depth a traves de los parámetros intrínsecos de la cámara y el modelo *PINHOLE* (apartado 4.3) y existen algoritmos para trasladar datos del SR Depth al SR RGB así como del SR principal al SR RGB. En la Figura 5.3 se puede observar el cambio de perspectiva entre ambos sensores fruto de diferentes FOV, y el tamaño a escala de cada una de las imágenes.

Figura 5.3: Perspectiva del mismo escenario para los dos sensores de Kinect V2. Imagen RGB (izquierda), Imagen IR y Mapa de profundidad (derecha).

La detección del esqueleto humano se realiza a partir de la profundidad sin que intervenga la imagen RGB. Por ello, si no se utiliza la necesaria transformación el esqueleto no se corresponde con la posición en la imagen RGB, como se puede observar en la Figura 5.4.

Figura 5.4: Posición de las articulaciones sin corregir las diferencias de resolución y escenario. Recuadrado en naranja el tamaño del mapa de profundidad

5.3. Reconocimiento del esqueleto

El software desarrollado por Kinect V2 permite el reconocimiento de hasta seis personas y el seguimiento de dos [12]. El seguimiento esta optimizado para reconocer personas de pie o sentadas, de frente al sensor. Las posiciones laterales generan ciertos problemas ya que una parte del cuerpo no es visible. En la figura 5.5 se puede observar un diagrama con un ejemplo del seguimiento realizado por Kinect V2 con el SDK 1.8. En la versión del SDK 2.0 añade puntos adicional en los extremos del pulgar y del dedo corazón, lo que define de una forma más completa la mano. En total el cuerpo lo conforman 25 articulaciones o puntos.

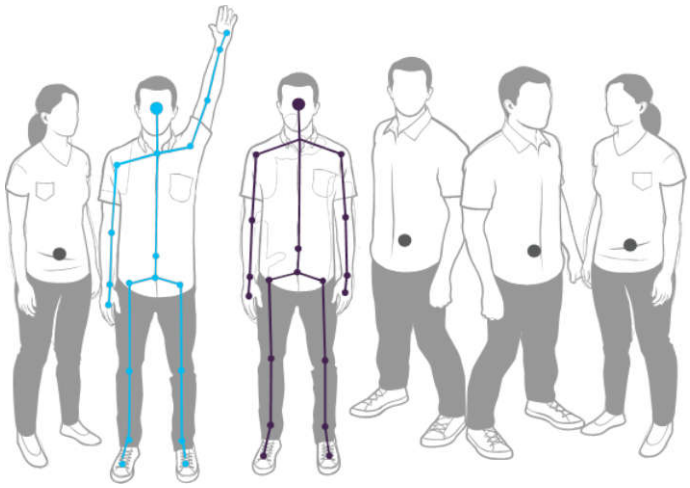

Figura 5.5: Esquema del sistema de seguimiento de Kinect V2 para SDK 1.8

El sistema de seguimiento tiene efecto memoria. El reconocimiento no es instantáneo, pero una vez está hecho el sistema conserva los parámetros para un reconocimiento posterior bastante más rápido que el inicial, lo que se traduce en un seguimiento fiable y un reconocimiento inmediato aunque el usuario detectado salga de la imagen y vuelva a entrar.

La información que proporciona el SDK 2.0 de las articulaciones es: su localización en la referencia *CameraSpacePoint*, la orienciación de las articulaciones en forma de matriz de rotación o quaterniones, el estado del seguimiento de la articulación (tracked/ no tracked). Con esta cantidad de información es posible construir un detector de colisiones que se base en un modelo de movilidad articular realista.

5.4. Clases y funciones implementadas

Se utiliza programación orientada a objetos para poder acceder a cualquier función y utilidad con independencia de la tarea que se esté ejecutando. El entramado y la recursividad de funciones es demasiado complejo para ser representado, mas se ha desglosado una simplificación en la Figura 5.6, que a su vez refleja el camino seguido para alcanzar los hitos y desarrollo de este TFG. En este apartado nos centramos en el código en C++.

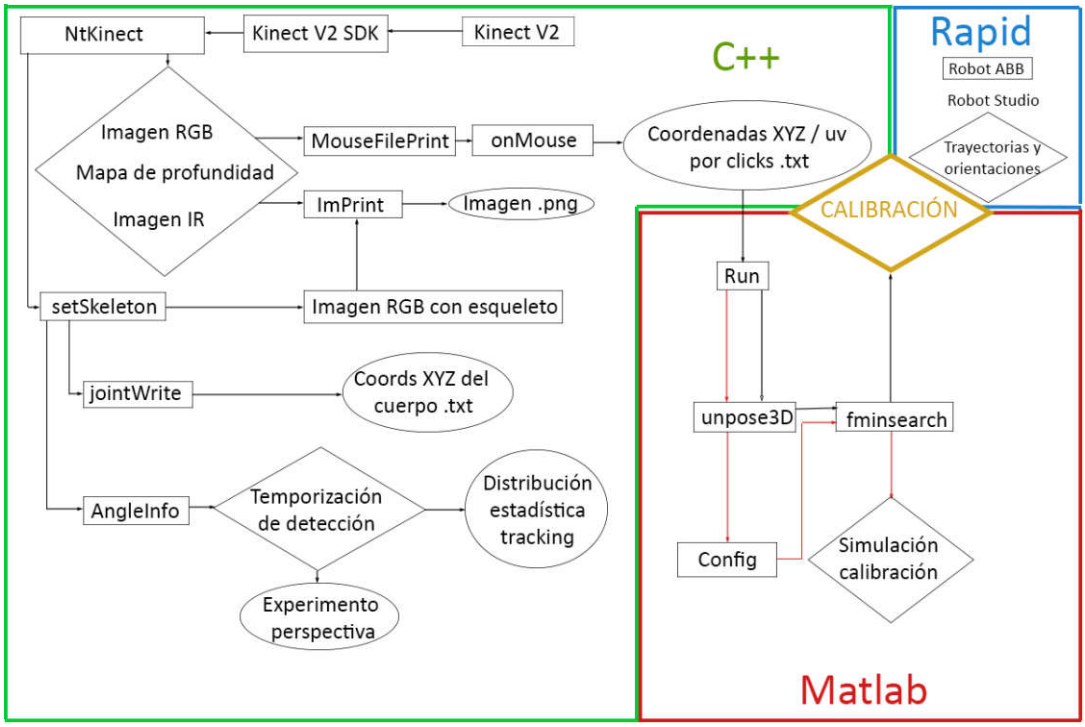

Figura 5.6: Diagrama simplificado de clases y funciones utilizadas

El codigo en C++ se estructura en un programa principal que se ejecuta en bucle. Dentro de este programa se disponen de una serie de comandos que permiten acceder a las diferentes funciones disponibles, a saber:

- q: Exit
- p: Realiza una captura de la imagen, ya sea IR, mapa de profundidad o RGB y la almacena en un archivo png.
- o:Activa la captura y almacenamiento en fichero de texto de las coordenadas XYZ, uv de la imagen en los puntos seleccionados por clicks.

- t: Activa o desactiva el seguimiento.
- a: Accede al recurso de temporización del seguimiento.
- r: Comienza/para un archivo de video (.avi)
- m: Se muestran las coordenadas del píxel y las coordenadas XYZ donde se situa el ratón. Varía entre mostrarse en la esquina superior izquierda, al lado del cursor o no mostrarse.
- j: Almacena las coordenadas XYZ de cada una de las articulaciones del cuerpo en una archivo de texto.
- d: Muestra el mapa de profundidad o la imagen IR si este está ya abierto.

A continuación se van a detallar algunas de las clases y funciones ilustradas en la Figura 5.6.

- La imagen tanto RGB, IR y de profundidad se obtienen de la capa de software NtKinect. Se muestran mediante una interfaz sencilla elaborada a partir de OpenCV.
- *onMouse* es una función que desempeña las acciones relacionadas con el ratón. Si este se mueve actualiza la información por pantalla y si se clica almacena las coordenadas ya explicadas. Para ello se integra con la clase MouseFilePrint, que es la encargada de crear los ficheros de texto, darles un nombre único y, supeditado a onMouse y a los comandos por teclado almacenar la información y cerrar el archivo.
- La clase *ImPrint* proporciona, mediante OpenCV, herramientas para realizar capturas de pantalla para cualquier tipo de imagen.
- La clase *AngleInfo* posee los mecanismos para realizar el experimento de perspectivas. Obtiene información de un timer no asociado al procesador principal. Ejecuta el seguimiento y cronometra el tiempo que se tarda en detectar el cuerpo entero. No es ejecutable en paralelo con ninguna otra función para no aumentar indebidamente el tiempo de detección.
- La clase *jointWrite* permite almacenar las coordenadas espaciales de las articulaciones en un fichero de texto, proporcionando las herramientas para ello.

Capítulo 6

Calibración del robot respecto de la cámara

Para transmitir datos sobre localizaciones espaciales al robot industrial, con el fin de interactuar con él, es necesario hacerlo en su SR. Se pueden aplicar dos métodos de calibración. El primero es el problema de estimación de pose de la cámara, en el cual a partir de proyecciones de puntos en la imagen y los parámetros intrinsecos de la propia cámara y su localización espacial en SR robot es posible obtener el cambio de SR entre ambos. Desafortunadamente no se poseen los parámetros intrínsecos de la cámara. El segundo metodo es utilizar la imagen de profundidad, por lo que se obtendrían coordenadas en el espacio desde la referencia de la cámara y desde la referencia del robot para un mismo punto.

Para resolver estre problema se van a expresar las transformaciones entre SR como matrices homogéneas, que de forma genérica se pueden expresar como:

$$T = \begin{pmatrix} n_x & o_x & a_x & t_x \\ n_y & o_y & a_y & t_y \\ n_z & o_z & a_z & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.1}$$

Reformulando la ecuacion 6.1

$$T = \begin{pmatrix} \mathbf{R} & \mathrm{t} \\ 0 & 1 \end{pmatrix} \tag{6.2}$$

La submatriz R hace referencia a la componente rotacional de la transformación.

Se han utilizado ángulos de Euler. De esta forma la matriz de rotación se expresa como:

$$\mathrm{Euler}(\phi,\theta,\psi) = \mathrm{Rot}(\mathbf{z},\phi) * \mathrm{Rot}(\mathbf{y},\theta) * \mathrm{Rot}(\mathbf{z},\psi) =$$
$$= \begin{pmatrix} C\phi\, C\theta\, C\psi - S\phi\, S\psi & -C\phi\, C\theta\, S\psi - S\phi\, C\psi & C\phi S\theta & 0 \\ S\phi\, C\theta\, C\psi + C\phi\, S\psi & -S\phi\, C\theta\, S\psi + C\phi\, C\psi & S\phi S\theta & 0 \\ -S\theta\, C\psi & S\theta\, S\psi & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

El vector t representa la traslación entre ambos SR.
Esta matriz de transformación permite obtener las coordendas de un punto en SR robot tal y como se expresa en la ec 6.5

$$\vec{P}_{robot} = \bar{\bar{T}}\vec{P}_{camara} \tag{6.3}$$

Reformulando la ec 6.5

$$\begin{pmatrix} P_x^r \\ P_y^r \\ P_z^r \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P_x^c \\ P_y^c \\ P_z^c \\ 1 \end{pmatrix} \tag{6.4}$$

6.1. Planteamiento de la resolución

El problema a resolver se plantea como un sistema de ecuaciones no lineales con 6 incógnitas, los tres ángulos de Euler y las translación en cada uno de los ejes. Se disponen de tantas parejas de puntos cámara-robot como sean necesarios. La resolución se basa en un problema minimización de error, para lo cual se plantearà el sistema para cada pareja de puntos y mediante métodos numéricos se halla la solución que mejor conjugue todas las parejas de puntos.

$$(\vec{P}_{robot} - \bar{\bar{T}}\vec{P}_{camara})^t(\vec{P}_{robot} - \bar{\bar{T}}\vec{P}_{camara}) = (Error_x)^2 + (Error_y)^2 + (Error_z)^2 \tag{6.5}$$

6.2. Definición y análisis del algoritmo utilizado

La función *fminsearch* de Matlab es una función iteratíva para la resolución de problemas de mínimos cuadrados. Se puede encontrar más información en

[https://es.mathworks.com/help/matlab/ref/fminsearch.html]. En los siguientes apartados se va a experimentar con dicha función para observar la adecuación y calidad para la resolución del problema de calibración planteado en el apartado 6. Las incógnitas a abordar son:

- Comportamiento del algoritmo al incrementar el número de puntos.
- Comportamiento del algoritmo al incrementar el ruido de medición de los puntos.
- Comportamiento del algoritmo al variar la estimación inicial.

Los dos primeros apartados se resolverán de forma cuantitativa mediante ensayos numéricos mientras que el segundo se realizará de forma cualitativa observando la convergencia de la función.

6.3. Análisis numérico del algoritmo: Descripción general

Para evaluar el comportamiento de la función *fminsearch* se van a realizar distintas simulaciones en las cuales se proporcionan puntos al algoritmo y se le solicita la resolución de la matriz de transformación que los relaciona. El proceso es el siguiente: Se definen arbitrariamente unos ángulos de Euler y un vector traslación para componer una matriz homogénea de transformación. A continuación se generan unos puntos aleatorios utilizando la función *randi* de Matlab que emulan puntos captados por la cámara. Esta función genera números enteros aleatorios siguiendo una distribución uniforme. Se recuerda que cualquier punto captado por la cámara se situará entre 0.5 y 4.5 m de distancia de esta y, por situación de su SR, tomarán siempre valores positivos en el eje Z. Los puntos se almacenan como indica la ecuación 6.6. Una última matriz de puntos, p, resultado de la transformación de los puntos almacenados en P_{or} representa las coordenadas de los puntos P en el SR robot. Se utilizan los puntos P_{or} asumiendo que el robot no comete errores significativos en la medición de dichos puntos.

$$P_{or} = \begin{pmatrix} P_x^1 & P_y^1 & P_z^1 & 1 \\ P_x^2 & P_y^2 & P_z^2 & 1 \\ P_x^3 & P_y^3 1 & P_z^3 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ P_x^n & P_y^n & P_z^n & 1 \end{pmatrix} \tag{6.6}$$

Acto seguido se genera una matriz aleatoria de coeficientes que responden a una distribución normal con media cero y desviación típica a convenir, con las mismas dimensiones que P_{or} . Sumando esta matriz a la matriz P_{or} se emula ruido derivado del proceso de la detección de puntos en la cámara. Esta matriz se crea utilizando la función *normrnd* de Matlab. El resultado de dicho adicción se almacena en una variable con el mismo formato que la mostrada en 6.6 denominada P.

El objetivo de este análisis numérico es evaluar la capacidad que el algoritmo posee para calcular la matriz de transformación que hemos definido arbitrariamente y estudiar las parámetros relevantes. Groso modo el esquema planteado para esta evaluación responde al representado en la Figura 6.1.

Para las distintas simulaciones se ha tomado como ángulos de Euler, expresados en radianes: $\varphi = 0, \theta = 0{,}7, \psi = 4{,}8$; y como vector traslación $t_x = 250, t_y = 90. t_z = 101$ Como solución inicial se ha utilizado $(\phi, \theta, \psi, t_x, t_y, t_z) = (0{,}3, 0{,}3, 0{,}3, 50, 60, 60)$. Se han evaluado cualitativamente más soluciones iniciales y el algoritmo es convergente con el suficiente número de iteraciones.

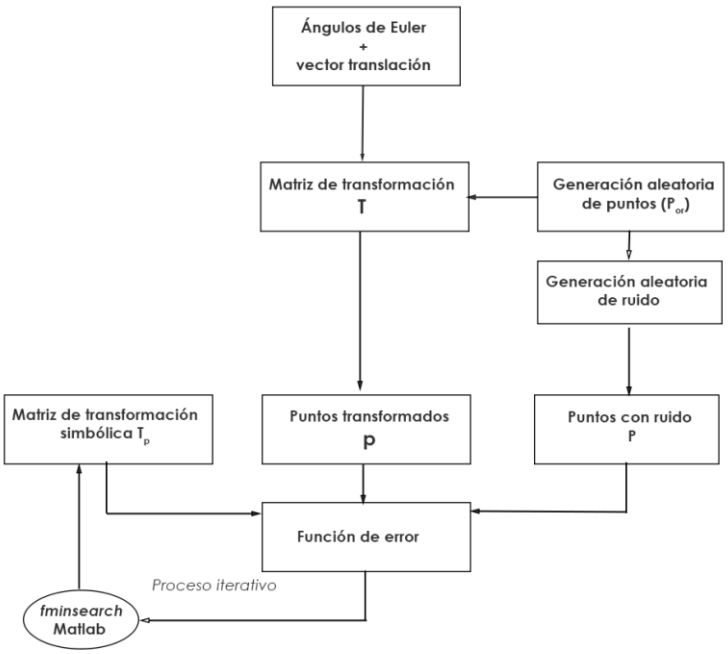

Figura 6.1: Esquema de ejecución del análisis numérico sobre la capacidad de resolución de la función *fminsearch*.

6.4. Planteamiento, resultados y discusión de las simulaciones

6.4.1. Planteamiento

Se han realizado dos simulaciones principalmente. En la primera se resuelve el problema n veces incrementando el número de puntos que utiliza el algorimo en cada resolución, desde 1 hasta n, induciendo un ruido aleatorio con distribución normal de media 0 y desviación típica 4mm. Evaluando el error absoluto se pretende discernir relación alguna entre el número de puntos utilizado y la precisión alcanzable. En la segunda se varía la desviación típica desde 1mm hasta 10mm en incrementos de 1 mm y se resuelve el problema para 3, 5, 10 y 20 puntos. En la Figura 6.2 se refleja la convergencia del algoritmo para n=500. Para el vector solución inicial existe gran error y se produce una rápida convergencia, momento tras el cual se afina la solucición lentamente. Un incremento del número de puntos implica un mayor número de iteraciones hasta alcanzar la solución, sin embargo, un escaso número de puntos puede superar el número de iteraciones máximo de igual forma.

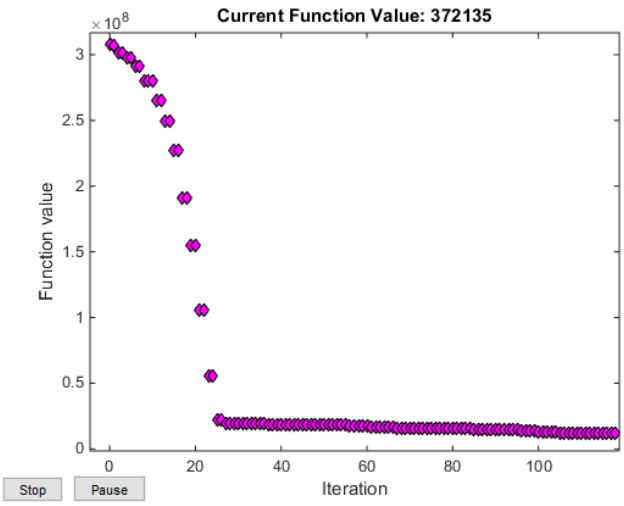

Figura 6.2: Convergencia del algoritmo de la función *fminsearch* para la ecuación planteada

La resolución de la matriz de transformación es un problema definido por 6 incógnitas, 3 parámetros relativos a la orientación y 3 parámetros relativos a la traslación. Sin embargo, los resultados de este estudio se van a parametrizar en 2 variables

principales. Para el error de traslación se utiliza la distancia entre el SR solución y el SR arbitrario, puesto que experimentalmente se observa un comportamiento homogéneo del error en cualquier dirección. Para el error de rotación se expresa el giro relativo entre el SR solución y el SR arbitrario y se calcula el eje de giro entre ambos.

6.4.2. Resultados y Discusión

Se comprueba que si no se induce ruido en la medición el algoritmo únicamente requiere 3 puntos para la perfecta resolución del sistema, lo que equivaldría a un sistema compatible determinado. Utilizando 1 o 2 puntos el algoritmo converge pero con errores del orden de las centenas de grados para la orientación y metros para la traslación.

Contrastada la correcta resolución del sistema bajo criterio de medición sin ruido, se realiza el primer experimento. De las 80 veces que se ha ejecutado la simulación 16 no se ha obtenido convergencia en la solución. Dado que la calibración no es un proceso crítico no se evalua la no convergencia del algoritmo.

En el primer experimento(Figura 6.3) se puede observar tendencia a la disminución del error, tanto angular como espacial, al aumentarse el número de puntos. Cabe destacar que el orden del error angular es de centésimas de grado, mientras que el error de traslación es del orden de mm. Este error, en adición al propio error cometido por la cámara, permite utilizar este sistema para la aplicación de control de un entorno robotizado, mas no es apto para aplicaciones en las que se requiera alta precisión.

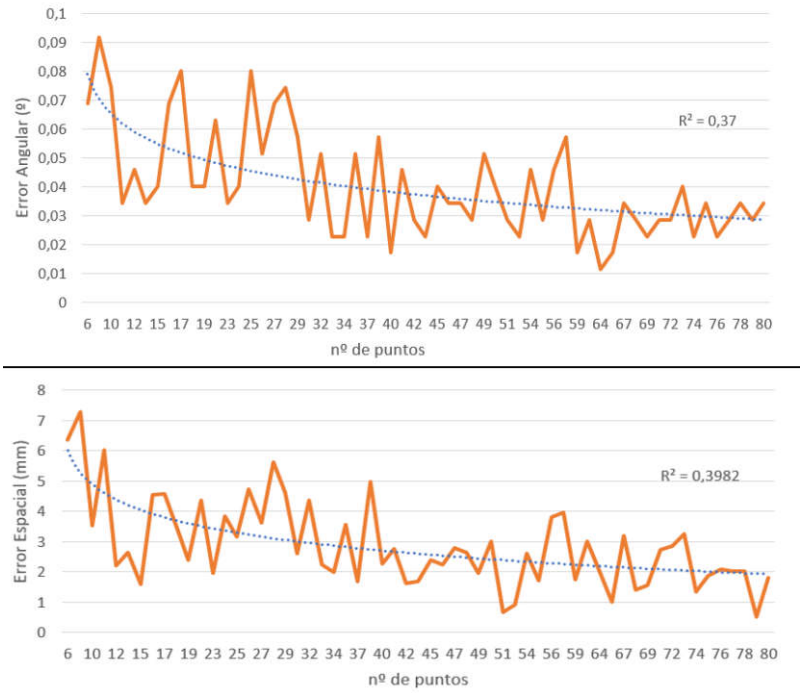

Figura 6.3: Error relativo en orientación entre la solución calculada y la solución exacta (izquierda, °) y espacial (derecha, mm) al aumentar el número de puntos para ruido de media 0 y desviación típica 4mm

En el segundo experimento (Figura 6.4) se observa que a mayor ruido inducido mayor error de medición se experimenta, como cabría esperar. A la par se observa una relación directa entre el número de puntos utilizado y el error producido. Un número elevado de puntos mitiga sustancialmente el efecto de la dispersión generado por el ruido. Como se expone en capítulos anteriores la precisión de la cámara decrece al aumentar la distancia. Este efecto nos permite aumentar la distancia cámara - robot manteniendo el error en el cambio de base acotado a pesar de dispararse el error de medición generado por la cámara. También permite paliar el error de calibración fruto tanto del método de calibración como de los errores en la toma de puntos.

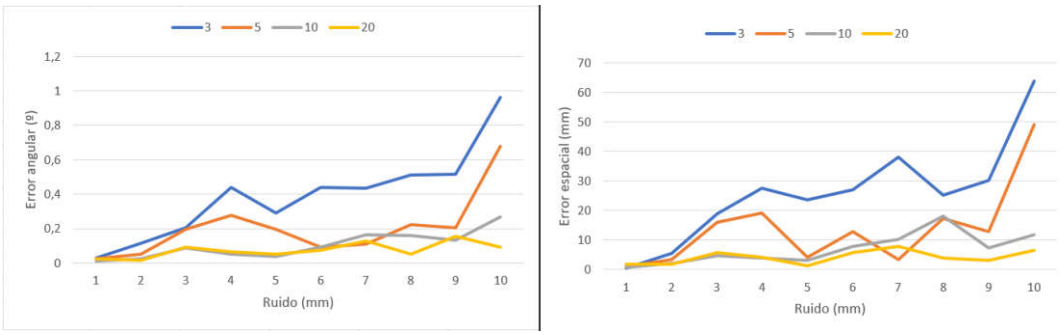

Figura 6.4: Error relativo en orientación entre la solución calculada y la solución exacta (izquierda, °) y espacial (derecha, mm) al aumentar el ruido inducido.

El sistema propuesto para calcular la transformación cámara - robot ofrece buenos resultados considerando se trata de una calibración óptica. Como bondad del método es subrayable la solidez de la solución frente al incremento del ruido con un número suficiente de puntos. Como contrapartida, el error de traslación asociado al cálculo de la transformación es del orden del error de medición de la cámara, por lo que en total disminuimos sustancialmente la precisión del sistema en su conjunto.

6.5. Implementación del método de calibración

6.5.1. Introducción y planteamiento

Tras el ensayo virtual del método de calibración se procede a su implantación experimental. De acuerdo a las conclusiones teóricas y a las características de la perspectiva estudiadas en el capítulo 7. La cámara se ha situado a una altura de 1.90 metros sobre el suelo y a una distancia de 2.30 metros del robot aproximadamente. La zona de trabajo se puede observar en la Figura 6.5 .

Figura 6.5: Espacio de trabajo del robot visto desde la cámara.

La resolución del sensor es limitada, lo que hace necesario utilizar una herramienta voluminosa que permita su clara visibilidad en pantalla. Se ha de considerar que el tamaño de la herramienta visto en la imagen es mínimo y, además, en los contornos se produce segmentación que resta aún más definición. El mapa de profundidad del espacio de trabajo se corresponde con la Figura 6.6. La superficie proyectada de la herramienta es de apenas 5 píxels y una incorrecta selección supone un alto error en la captura del punto y, por ende, en la solución final.

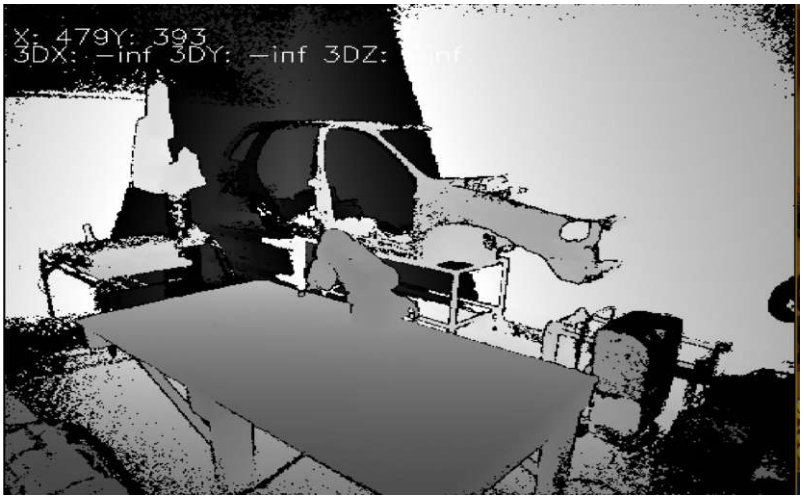

Figura 6.6: Mapa de profundidad del espacio de trabajo del robot.

Puesto que la precisión de la cámara es del orden de mm es importante planificar un proceso y unas medidas correctoras que permitan minimizar el error experimental.

Para ello el desarrollo de la calibración se ha realizado en varias fases:

- Ensayo de dos geometrías para la herramienta: esfera y poliedro.
- Corrección del error inducido por factores geométricos.
- Proceso iterativo

Para realizar esta calibración se dispone de 4 conjuntos de puntos seleccionados en ubicación y orientación a partir de RobotStudio. Estos conjuntos constan de: un conjunto de puntos de control (utilizados para contrastar los resultados de la calibración), dos conjuntos de puntos utilizados en la calibración de la herramienta dos; uno de ellos distribuido por todo el espacio de trabajo y el otro limitado a la zona de alta precisión de la cámara. Por último se dispone de un conjunto de puntos utilizados para calibrar el robot con la herramienta 1. Las orientaciones seleccionadas para cada punto son aquellas que permiten una clara visibilidad de la herramienta en el mapa de profundidad. Se pueden consultar las coordenadas de cada pareja de puntos cámara - robot para cada conjunto en el Anexo A.

6.5.2. Herramienta 1, descripción y resultados

La primera herramienta utilizada consiste en una punta con un prisma cuadrangular de lado 29mm (Figura 6.7). Fabricada en poliestireno expandido, de color blanco para asegurar la máxima reflexión de los rayos IR.

La transformación solución a esta primera calibración es, Rot(Euler, rad) = (-0.8568, 4.3873, -1.6364); Trans(mm)=(1625.22, -1628.654, 1138.409).

Figura 6.7: Herramienta basada en prisma cuadrangular de 29 mm de lado. Poliestireno expandido.

Una vez resulta la matriz de transformación se comprueba el resultado comparando las coordenadas ofrecidas para dos puntos aleatorios por el robot y por la cámara. Tabla 6.1

[mm]	X	Y	Z
camara	614.27	44.69	311.07
robot	618.83	58.39	297.9
error	4.56	13.7	-13.7
camara	66.69	-528.48	778.53
robot	55.6	-555.97	767.33
error	-11.09	-27.49	-11.2

Tabla 6.1: Comprobación de calibración comparando coordenadas obtenidas desde la cámara y el robot para la herramienta 1

El error varía desde 0.5 cm hasta 3 cm en cada eje. Apoyandose en los estudios previos este error se descompone de la siguiente forma: Incertidumbre de la cámara 4mm, error del proceso de cálculo de la transformación (3-8mm). Error superior al descrito es atribuible a la voluminosidad de la herramienta, ya que con el método elegido no es posible obtener las coordendas exactas del punto calibrado de la herramienta, ni conocer con precisión de que parte de la herramienta se han obtenido las

coordenadas. Ello puede suponer una desviación de hasta 1.5 cm en la captación de cada punto por parte de la cámara, que genera una solución peor que la supuesta en las simulaciones (en las que no se consideraba la voluminosidad de la herramienta) y posteriormente vuelve a inducir error en la comprobación de la solución por este mismo hecho. Esta idea se ilustra con mayor claridad en la figura 6.8

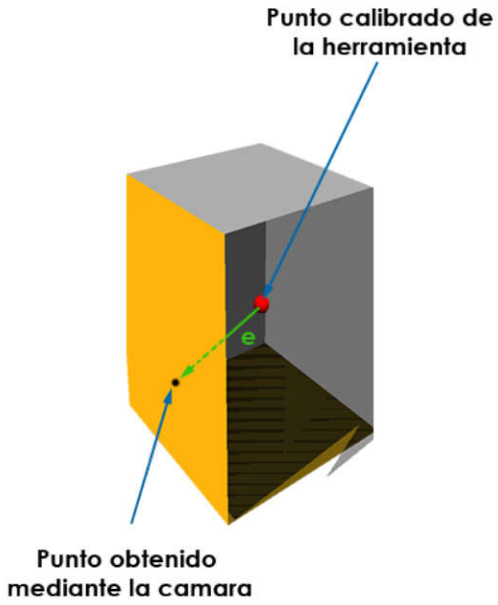

Figura 6.8: Error cometido en la calibración y la medición debido al volumen de la herramienta.

6.5.3. Herramienta 2, descripción y resultados

Se ensaya la calibración con una segunda herramienta para la cual se desarrolla un sistema de corrección del error derivado del volumen de la herramienta. En vez de un prisma se opta por una esfera de plastilina blanca de 17 mm de diametro. Para corregir el error derivado del volumen de esta se parten de dos hipótesis:

- El error de medición cometido es igual al radio de la esfera. Esta hipótesis asume que, en la medida de lo posible, el dato de profundidad obtenido por la cámara se recibe de un punto alineado con el origen del SR de la cámara y el centro de la esfera.

- Las coordendas obtenidas por la cámara son ideales, sin ruido.

Bajo estas hipótesis y con un ejercicio de geometría que se describirá a continuación es posible obtener las componentes cartesianas del error generado por el volumen del objeto y corregirlo. A su vez, esto nos permite compensar el error una vez la cámara se halla calibrada. Esto se ilustra en la Figura 6.9. Los ángulos α y β descritos se pueden calcular a partir de la posición de la herramienta en el espacio, Figura 6.10.

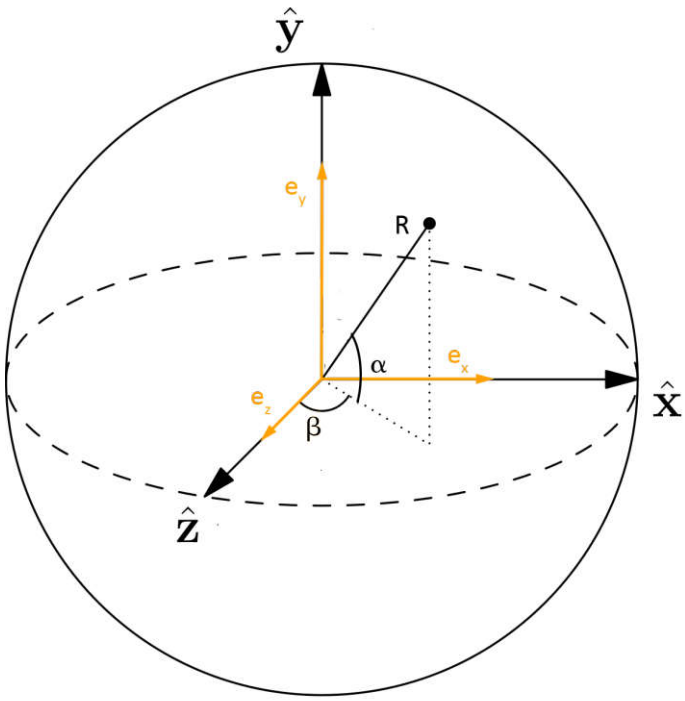

Figura 6.9: Proyección de la corrección sobre el SR de la cámara, aplicada a las coordenadas medidas por esta pre y post calibración.

Las ecuaciones necesarias para obtener la corrección para la herramienta son las siguientes

$$\beta = atan2\frac{Z}{Y} \tag{6.7}$$

$$d_{yz} = \frac{Y}{cos\beta} \tag{6.8}$$

$$\alpha = atan2\frac{X}{d_{yz}} \tag{6.9}$$

$$(e_x, e_y, e_z) = R(sin\alpha, cos\alpha \cdot cos\beta, cos\alpha \cdot sin\beta) \tag{6.10}$$

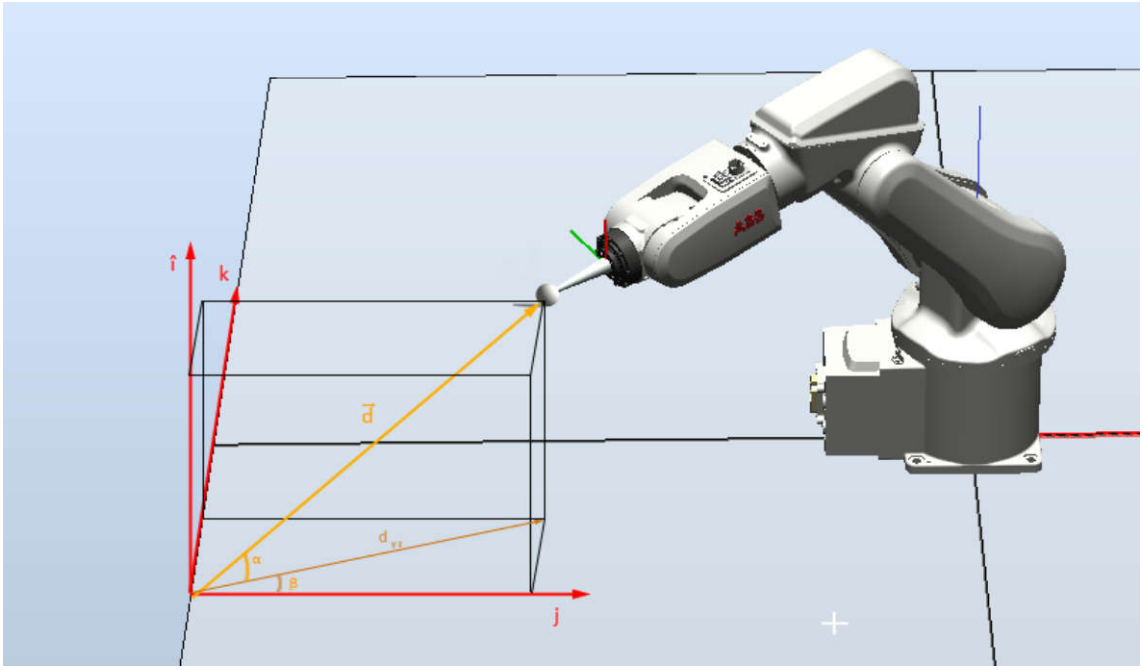

Figura 6.10: Cálculo de los ángulos α y β a partir de la posición de la esfera en el espacio bajo hipótesis de medición ideal.

Se comparan los resultados de la calibración con y sin corrección para una herramienta de 17mm de diámetro. Se calcula la media de error en terminos absolutos para cada uno de los ejes y en global.

$$\overline{e_x} = \frac{1}{n}\Sigma_i^n x_i; \overline{e_y} = \frac{1}{n}\Sigma_i^n y_i; \overline{e_z} = \frac{1}{n}\Sigma_i^n z_i \tag{6.11}$$

$$e_P = \frac{1}{n}\Sigma^n \sqrt{e_x i^2 + e_y^2 + e_z^2} \tag{6.12}$$

Los resultados se exponen en la tabla

mm	$\overline{e_x}$	$\overline{e_y}$	$\overline{e_z}$	$\overline{e_p}$
inalterado	5.88	7.01	7.57	12.63
compensado en calibración	8.32	9.21	8.50	15.88
compensado en calibración y medición	5.82	6.76	7.54	12.46

Tabla 6.2: Resultados de calibración con herramienta esférica de 17mm de diametro.

El error de calibración promedio es de aproximadamente 13 mm. Este se puede desglosar en varias componentes:el error de medición de la cámara (1-4 mm), aunque este se ve paliado por el efecto nube de puntos; el error de calibración inherente al algoritmo, que si bien tiende a disminuir al aumentar el número de puntos es susceptible de ser muy variable aunque acotado a 10 mm; y el error experimental y de violación de las hipótesis de corrección.

Capítulo 7

Efecto de la perspectiva en el tracking

7.1. Introducción

Los algoritmos de Kinect V2 se basan en la imagen de profundidad para realizar la detección de personas. Dado que en última instancia se va a utilizar la cámara para controlar un entorno robótico es conveniente saber cómo se desenvuelve bajo diferentes perspectivas con objeto de colocarla a una altura y profundidad óptimas para obtener la menor tasa de fallo en la detección. Además se quiere observar cual es el tiempo promedio que tarda en detectar e identificar a una persona así como inferir si hay alguna distribución estadística que nos permita cuantificar su comportamiento. Para ello se van a realizar sendos experimentos.

7.2. Experimento 1

Como se explica anteriormente la Kinect V2 sólo permite medir profundidad hasta un máximo de 4,5 metros. En este primer experimento se busca comprobar cual es la posición óptima para colocar la Kinect respecto del objetivo a detectar. Para ello se dispone la cámara a diferentes alturas, que varían de 0,5 en 0,5 metros desde los 0,5 hasta los 4,5 metros de altura. Para cada altura de la cámara se ejecuta el algoritmo de detección con el usuario colocado a diferentes distancias desde los 0,5 hasta los 4,5 metros como se puede observar en la figura 7.1.

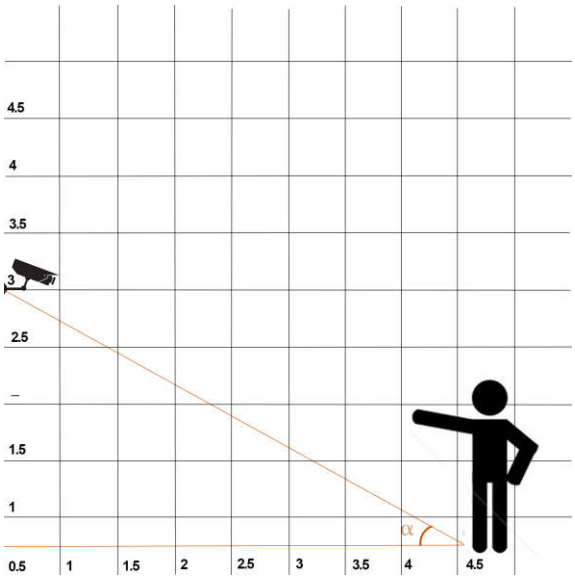

Figura 7.1: Ilustración de posiciones para la medición de perspectivas.

Para cada ubicación se ejecuta el algoritmo de detección 4 veces y se mide el tiempo que tarda en detectar al usuario. Esta medición ha sido programada con un timer independiente de la ejecución del algoritmo principal para no incrementar el tiempo de detección. En algunos casos por cuestienes de cercanía o perspectiva se ha cronometrado manualmente. La detección se considera fallida si no se realiza antes de 30s.

A la par se ha tomado una fotografía en cada ubicación para observar cualitativamente la evolución de la perspectiva y si los algoritmos situan bien las diferentes articulaciones. Esta evolución de las perspectivas se refleja en las Figura 7.5, 7.6 y 7.7.

Al realizar las mediciones se han seguido los siguientes criterios de colocación de usuario y cámara que guardan prioridad según el orden descrito.

- El usuario se coloca de frente a la cámara y facilita tanto como sea posible la detección extendiendo las palmas o colocandose de forma favorable.
- La cámara abarca la mayor cantidad del cuerpo de usuario que sea posible dando prioridad al torso-cabeza.
- La cámara se coloca en la posición más horizontal posible.

7.2.1. Resultados y conclusiones

Como ya se ha explicado con anterioridad se mide el tiempo que le cuesta realizar la primera detección 4 veces para cada ubicación. Los resultados se pueden observar en la figura 7.4. Existe una progresión de color desde verde hasta rojo en función del tiempo cronometrado, siendo el verde asignado a los menores valores.

El punto más adecuado para la cámara de acuerdo con el tiempo que se tarda en realizar la primera detección sería colocarla entre 1 y 3 metros de altura con los usuarios a detectar situados a entre 1 y 3 metros de distancia (Figura 7.2). Cabe recordar que el tiempo que se tarda en realizar la primera detección no afecta al posterior seguimiento del usuario, aun así se interpreta que una primera detección temprana es consecuencia de un mejor funcionamiento de los algoritmos.

Figura 7.2: Ejemplo de correcto seguimiento para perspectivas verticales para alturas comprendidas entre 1,5 y 2,5 metros.

En cuanto a la perspectiva se puede observar que entre 1 y 2.5 metros de altura los algoritmos situan correctamente las articulaciones en la mayoria de situaciones. Para alturas superiores es capaz de realizar el seguimiento pero detecta el cuerpo en posiciones equivocas generando una alta tasa de fallos. Existe a la par una eficacia mayor en la detección cuando el usuario se situa perpendicular al eje Z de la cámara, observado cualitativamente durante la realización de estos. En la Figura 7.3 se

observa como el seguimiento empeora al variar a una perspectiva más vertical.

Figura 7.3: Ejemplo de incorrecto seguimiento para perspectivas verticales.

Combinando ambas interpretaciones se concluye que la cámara trabaja en su punto de operación óptimo situando la cámara a la altura del torso y perpendicular al usuario. Es admisible con buen resultado elevar la posición para la aplicación de un entorno robótizado o bajarla para otras aplicaciones. El tiempo minimo de detección es de 0.05 segundos alcanzando en ocasiones cifras de 15 a 20 segundos. El tiempo promedio es de 2 a 3 segundos.

	0,5	1	1,5	2	2,5	3	3,5	4	4,5	
0,5										
	30	30	0,05	6,24	3,817	16,528	19,24	0,051	30	
	30	30	5,42	0,054	3,542	0,051	19,035	3,368	30	
	30	30	2,27	2,791	2,769	0,052	0,056	0,067	30	
	30	30	6,513	3,629	2,96	0,049	0,054	4,776	30	
1										
	30	30	1,248	7,655	7,369	0,053	7,413	2,271	1,734	
	30	3	4,752	4,757	2,357	4,05	0,069	0,054	3,392	
	30	0,05	0,064	1,731	0,053	2,154	0,076	1,732	2,038	
	30	30	0,078	2,502	0,114	1,888	1,328	0,233	4,845	
1,5										
	30	30	0,047	2,537	5,178	2,885	2,943	1,775	2,993	
	30	30	5,858	0,078	0,051	0,049	6,282	3,64	0,053	
	30	30	0,054	16,195	1,632	0,048	0,586	0,055	6,689	
	30	30	0,047	0,061	1,562	0,051	5,858	2,758	1,882	
2										
	30	8,657	0,07	1,485	0,057	0,046	0,05	30	30	
	30	4,183	0,059	3,765	15,479	17,786	3,9	30	30	
	30	7,728	2,668	0,236	0,049	7,949	27,769	30	30	
	30	0,321	1,937	0,053	12,302	0,228	1,816	30	30	
2,35										
	30	30	0,053	0,045	0,05	5,005	0,078	0,052		
	30	6,805	2,477	2,191	6,017	0,92	5,126	0,072		
	30	11,873	0,05	6,674	6,26	9,669	4,122	3,12		
	30	1,786	2,516	4,162	2,016	2,551	4,752	6,63		
3,1										
	5,974	3,887	0,051	7,016	0,052	1,585	1,885		30	
	30	1,397	4,679	1,735	3,904	27,255	1,889		30	
	2,054	0,971	0,924	1,562	2,379	15,538	4,125		30	
	3,693	0,283	4,683	0,86	3,253	9,998	2,689		30	
3,57										
	0,079	0,057	13,62	0,054	5,58	0,05	0,05			
	5,54	0,051	0,947	20,16	0,05	30	0,05			
	1	2,93	5,78	13,57	14,42	0,05	1,23			
	0,102	18,92	12	2,13	0,05	30	0,165			
4										
	30	0,079	30	22,47	0,052	0,049	4,42	30	30	
	30	5,54	7,86	11,6	1,195	1,43	0,052	30	30	
	30	1	7,04	3,17	2,12	2,3	7,25	30	30	
	30	0,102	30	6,13	9,209	0,052	0,35	30	30	
			2,99							

Figura 7.4: Tiempo necesario para realizar la primera detección para cada ubicación. En las columnas se refleja la distancia a la que se situa el usuario de la cámara. En las filas la altura a la que situa la cámara respecto del suelo. 4 medidas para cada par distancia - altura.

Figura 7.5: Composición de cada perspectiva de 0,5 a 1,5 metros.

Figura 7.6: Composición de cada perspectiva de 2 a 3 metros.

Figura 7.7: Composición de cada perspectiva de 3,5 a 4,5 metros.

7.3. Experimento 2

Se va calcular la distribución estadística que rige el tiempo que se tarda en realizar la primera detección. Para ello se va a tomar una muestra de n = 50 mediciones temporizadas equivalentes a las realizadas en el experimento 1. La cámara se ha situado a 1,35 m. del suelo y a 1,5 metros de distancia del usuario, que se situa de pie, frente a la cámara en posición neutra.

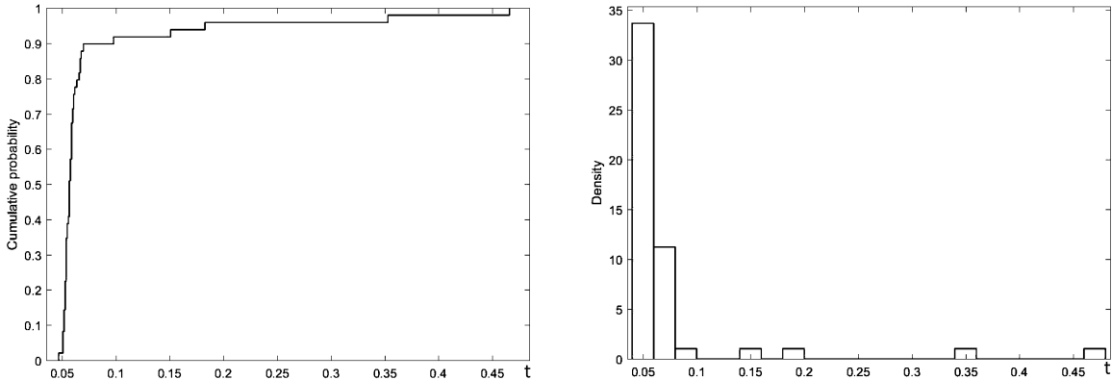

Figura 7.8: Función de probabilidad acumulada (izquierda) y función de densidad (derecha)

Como se puede observar en la Figura 7.8 la probabilidad de haber realizado la primera detección antes de 0,1 segundos es aproximadamente del 95 %. La mayoría de las ocasiones la detección tarda en mínimo registrado, 0,05 segundos. Todo ello refleja el correcto funcionamiento de los algoritmos en la zona designada como más adecuada a partir del experimento 1. Si consideramos todas las mediciones realizadas en el experimento 1 es factible suponer que esta distribución no se mantendrá para cualquier posción relativa cámara - usuario.

Capítulo 8

CONCLUSIONES

En este Trabajo Fin de Grado se ha realizado el estudio para la utilización de una cámara RGB-D como la Kinect V2 para detectar personas que puedan interaccionar con un robot manipulador como el IRB 120 de ABB.

En primer lugar se parte de una revisión bibliográfica en lo referente a tecnologías ópticas de medición de profundidad, en el que se destaca la visión estéreo, la luz estructurada y el tiempo de vuelo, siendo este último en el que se basa la tecnología de Kinect V2. A continuación se describen las prestaciones técnicas de los sensores tanto IR como RGB de la cámara y se profundiza en la adecuación de los entornos a la hora de utilizar la Kinect V2 siendo clave su uso en entornos interiores, sin superficies reflectantes y en el centro de la imagen para obtener los mejores resultados. Cubiertos los aspectos técnicos y tecnológicos se realiza un análisis de viabilidad cuyo objetivo es conocer cuales son las velocidades máximas que la cámara permite capturar en función del tamaño del objeto a capturar y la distancia a la que se situa de la cámara, bajo un enfoque teórico basado en el modelo *pinhole*. Como resultado de dicho análisis se generó una tabla heurística.

A partir de este punto se pasa del plano teórico al plano práctico, comenzando por configurar y programar la cámara en lenguaje C++. Se utilizan algoritmos de detección del esqueleto basados en mapas de profundidad, generando multitud de clases y funciones propias que permiten acceder a todas las características hardware y software de la cámara basadas en capas de software como el SDK de Kinect V2 u OpenCV. Una vez dominadas todas las características intrínsecas a la cámara se plantea un método de calibración entre la Kinect V2 y el robot industrial IRB

120 de ABB, que permita obtener la rotación y traslación entre los sistemas de referencia propios de cada elemento. Para ello se opta por resolver un problema de resolución mínimos cuadrados en el que se minimiza el error cartesiano entre las coordenadas de puntos del robot industrial y de los mismos puntos (en la base del robot) obtenidos a traves de la cámara. Se selecciona la función *fminsearch* de Matlab para resolver el problema y se simula para observar el comportamiento de la función ante el incremento del número de puntos y del ruido de medición de la Kinect V2. Se concluye que el incremento de puntos es positivo puesto que atenúa el error generado por el ruido de medición y da lugar a una tendencia a la baja del error producido durante la resolución. Tras consensuar un método de calibración se persigue definir cual es la posición cámara - usuario que mejor funcionamiento de los algoritmos de detección del esqueleto permita. Para ello se realiza un experimento de perspectivas en el que se varía la posición relativa cámara - usuario. Se concluye que el lugar la posición relativa más adecuada es aquella en la que la cámara se situa de frente al usuario, perpendicular a este, aproximadamente a 1,5 m de distancia y a 1,5 m de altura respecto del suelo. Sin embargo, es posible localizarla a 2 - 2,5 metros del suelo en aras de una mayor utilizadad y manteniendo su correcto funcionamiento. Por último, conociendo la posición relativa más adecuada se procede a implantar a nivel práctico la calibración. Se realiza con puntos reales obtenidos del robot industrial ABB y de la cámara y se obtiene la transformación cámara robot. Los resultados experimentales ofrecen un error medio de 13 mm, un resultado satisfactorio considerando que es un método óptico de calibración.

8.1. Trabajo Futuro

Este TFG se ha versado en el dominio de la cámara y los conceptos de visión por computador. Una línea futura sería realizar la conexión cámara-robot vía ethernet para poder trabajar con una interacción en tiempo real. También se podría controlar un entorno robótico bajo los criterios de la *ISO/TS 15066:201 ROBOTS AND ROBOTIC DEVICES - COLLABORATIVE ROBOTS*, que son esencialmente restrictivos con el robot y el entorno, así como un detector de colisiones.

Bibliografía

[1] Yoshihisa Nitta. Ntkinect : C++ class library for kinect v2. In *172-th conference, SIG Human Computer Interaction, IPSJ*, 2017.

[2] Maria Lazaro. *Robots and humans can work together with new ISO guidance.* ISO, 2016.

[3] Infaimon. Sensores cmos para obtención de imágenes 3d en tiempo real. In *XXXI Jornadas de Automática industrial, Universidad de Jaen*, 2010.

[4] Abdulmotaleb El Saddik Lin Yang, Longyu Zhang. Evaluating and improving the depth accuracy of kinect for windows v2. *IEEE Sensors Journal*, 2015.

[5] Photogrammetry and Geomatics Group of ICube Laboratory. First experiences with kinect v2 sensor for close range 3d modelling. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-5/W4:93–100, 2015.

[6] Sílvia Rodríguez Jiménezl. *Desarrollo y Evaluación de estrategias para aplicaciones de percepción 3D usando cámaras de rango.* PhD thesis, Universidad Carlos III de Madrid, 2015.

[7] Sandra Demitseval. *GESTURE BASED COMPUTER CONTROLLING USING KINECT CAMERA.* PhD thesis, University of Tartu, 2015.

[8] The Xbox One system on a chip and Kinect sensor. First experiences with kinect v2 sensor for close range 3d modelling. *IEEE Micro*, 2:40–55, 2014.

[9] Lembit Valgmal. *3D reconstruction using Kinect V2 camera.* PhD thesis, University of Tartu, 2016.

[10] Beatriz Sánchez Tabuenca Juan Llado París. *MECÁNICA. Grado en ingeniería de tecnologías industriales.* Copy Center Zaragoza, 2014.

[11] Rendey Horacio. *Tenis: potencia, velocidad y movilidad.* INDE, 2004.

[12] Skeletal Tracking Microsoft. https://msdn.microsoft.com/en-us/library/hh973074.aspx.

Anexos

Apéndice A

Puntos utilizados en la calibración

```
Puntos estandar CAMARA (metros)
0.0321804 -0.541183 2.036 1
-0.255122 -0.505756 1.985 2
-0.464033 -0.466137 2.1 3
-0.693294 -0.388352 2.568 4
-0.514939 -0.341587 2.69 5
-0.732374 -0.145929 2.738 6
-0.812656 -0.144607 2.465 7
-0.703102 -0.302395 2.002 8
-0.368118 -0.374816 1.798 9
0.0435882 -0.290039 1.813 10
0.309815 -0.351435 1.966 11
0.321888 0.0187954 2.004 12
-0.225295 -0.0173927 1.784 13
-0.53629 0.00737798 1.89 14
-0.736212 0.0491991 2.145 15
-0.570555 0.246548 2.134 16
-0.472323 0.219014 1.988 17
-0.173927 0.258982 1.918 18
0.0145948 0.200217 1.905 19
0.204525 0.224709 2.034 20

Puntos estandar ROBOT (centimetros)
288.15,-516.55,126.46
```

```
500.16,-319.51,162.66
543.23,-84.06,166.77
341.54,391.45,117.82
117.37,347.76,131.37
188.02,579.49,297.99
433.62,486.27,364.26
723.42,90.63,331.38
681.41,-307.79,322.02
380.91,-600.62,415.06
118.98,-729.49,334.15
4.06,-633.22,673.28
524.52,-344.53,674.66
648.92,-44.74,665.21
589.02,268.74,635.79
452.44,169.24,835.98
495.23,1.90,848.78
340.97,-247.20,910.91
246.11,-424.24,863.84
27.44,-489.69,863.83

Puntos alta precision CAMARA (metros)
-0.16361 -0.0717874 1.802 1
-0.198202 -0.248081 1.762 2
0.0135981 -0.245434 1.776 3
-0.369772 -0.211417 1.78 4
-0.43745 -0.346075 1.802 5
-0.426122 -0.467975 1.839 6
-0.162902 -0.488567 1.802 7
-0.150399 -0.123249 1.763 8
0.0778116 -0.116048 1.8 9
0.082423 -0.233345 1.795 10
0.0929486 -0.329082 1.811 11
0.093323 -0.399897 1.82 12
-0.129839 -0.254822 2.14 13
-0.248983 0.171095 1.931 14
-0.316501 0.18726 1.992 15
-0.372205 0.196263 2.03 16
```

```
-0.493561 0.144643 2.008 17
-0.127351 -0.168028 1.778 18
0.023712 -0.126256 1.805 19
0.0692294 -0.0580181 1.833 20
-0.317799 -0.233557 2.109 21
-0.424229 -0.303795 2.043 22
-0.429754 -0.236963 2.042 23
-0.163782 -0.0918222 2.474 24
-0.284426 -0.0522376 2.525 25

Puntos alta precision ROBOT (centimetros)
495.17,-409.90,647.92
565.34,-431.22,479.09
409.64,-581.17,479.07
654.32,-276.44,499.21
700.39,-230.75,357.32
696.23,-254.53,234.61
560.34,-485.31,234.59
510.91,-442.90,598.86
330.63,-589.81,598.85
350.45,-624.75,488.63
359.98,-641.58,396.67
363.39,-647.59,324.13
286.38,-290.77,402.52
400.32,-206.43,842.27
400.30,-125.27,842.27
415.17,-59.76,842.26
505.83,18.40,795.43
500.68,-463.91,557.67
366.81,-544.31,591.81
299.45,-551.59,647.76
386.39,-131.38,402.50
513.96,-96.61,346.39
515.43,-88.43,413.37
30.34,-25.85,471.06
20.09,147.09,474.02
```

```
Puntos de control CAMARA (metros)
0.418116 -0.485684 2.11 1
0.183439 -0.535358 1.936 2
-0.0460037 -0.53797 1.837 3
-0.249666 -0.534991 1.828 4
-0.454406 -0.506639 1.853 5
-0.66048 -0.444503 2.005 6
-0.812146 -0.39097 2.194 7
-0.807382 -0.252695 2.134 8
-0.667145 -0.274782 1.96 9
-0.475944 -0.393658 1.819 10
-0.665812 -0.356475 1.941 11
-0.84571 -0.298639 2.173 12
-0.806303 -0.19139 2.162 13
-0.66868 -0.223793 1.98 14
-0.531846 -0.260702 1.858 15
-0.361434 -0.309644 1.788 16
-0.17924 -0.320749 1.766 17
0.0735566 -0.315894 1.82 18
0.272305 -0.260555 1.926 19
0.200209 -0.249224 2.047 20
0.0682241 -0.258827 1.95 21
-0.121369 -0.269784 1.915 22
-0.43217 -0.242523 1.953 23
-0.610373 -0.196817 2.092 24
-0.163065 -0.0715484 1.796 25

Puntos de control ROBOT (centimetros)
-51.33,-728.37,182.61
258.35,-682.94,182.60
468.74,-559.85,182.58
605.74,-407.71,182.56
709.78,-229.71,196.90
727.72,28.59,218.52
674.43,274.83,218.51
675.08,275.19,353.58
723.33,51.88,385.21
```

```
726.60,-202.58,306.45
754.22,20.80,307.83
685.89,314.38,307.81
650.22,300.49,405.76
707.23,70.02,436.02
706.19,-110.99,425.27
654.12,-293.22,396.41
556.36,-452.01,396.39
360.47,-619.60,396.38
147.69,-697.49,441.88
113.69,-575.53,425.49
263.12,-524.34,425.48
416.51,-413.14,425.48
571.09,-134.23,425.46
577.24,104.66,425.46
495.17,-409.90,647.92

Puntos de calibracion herramienta 1, CAMARA (metros)
-0.171086 0.0552748 2.142
0.165718 -0.06761 1.967
-0.0920528 -0.190753 2.078
-0.304215 -0.00365134 2.358
-0.29543 -0.0233117 2.391
0.449686 -0.124054 2.016
0.0159871 -0.334205 2.089
-0.469219 -0.206698 2.258
-0.186848 0.395752 2.003
0.423414 0.285135 1.855
0.54783 -0.264327 1.921
-0.0823044 -0.328064 1.861
-0.469181 -0.276618 2.383
-0.0837697 0.487423 2.022
0.301027 0.579228 2.642
0.541752 0.388586 2.057
0.160116 -0.159003 1.733
0.11278 0.626437 2.203
-0.201513 -0.0516838 1.789
```

```
-0.564591 -0.0373261 2.459

Puntos de calibracion herramienta 1, CAMARA (centimetros)
41.705 2.552 50.345
30.279 -34.779 42.876
45.594 -11.970 28.442
39.407 25.865 38.406
37.287 29.972 36.957
8.727 -53.084 35.189
40.160 -22.567 14.646
64.285 25.789 24.204
42.730 1.266 90.816
7.266 -53.613 79.258
9.227 -68.275 22.590
61.580 -30.894 22.589
57.371 35.480 13.594
30.412 -3.059 98.153
-43.115 19.865 84.224
-17.625 -45.511 83.048
48.269 -52.907 42.194
0.798 -0.705 104.490
68.070 -21.457 53.617
53.175 53.249 34.638
```