Escuela Politécnica Superior

Análisis comparativo de métodos de calibrado para sensores RGB-D y su influencia en el registro de múltiples vistas

Grado en Ingeniería Informática

Trabajo de Fin de Grado

Autor:
Víctor Villena Martínez

Tutores:
Jorge Azorín López
Marcelo Saval Calvo
Andrés Fuster Guilló

Universitat d'Alacant
Universidad de Alicante

Junio 2015

U N I V E R S I D A D D E A L I C A N T E

Departamento de Tecnología Informática y Computación

Trabajo de Fin de Grado

Análisis comparativo de métodos de calibrado para sensores RGB-D y su influencia en el registro de múltiples vistas

Víctor Villena Martínez
vvillena@dtic.ua.es

Tutores

Dr. Jorge Azorín López

Dr. Marcelo Saval Calvo

Dr. Andrés Fuster Guilló

Memoria presentada para aspirar al grado de:
GRADUADO EN INGENIERÍA INFORMÁTICA
Alicante, 25 de Junio de 2015

A mis abuelos

Hay ausencias visibles toda la vida

Siempre hay que aprender.

Juan Antonio Villena S.

AGRADECIMIENTOS

En estas líneas me gustaría agradecer a todas aquellas personas que me han apoyado y enseñado a lo largo de este camino.

A mis padres y a mi hermana, por todo el apoyo, esfuerzo y enseñanzas recibidas, lo que me ha permitido llegar hasta aquí, y a mi familia, por todos esos conocimientos y buenos momentos que han compartido conmigo, sólo puedo disculparme cuando no he podido estar ahí. Tampoco puedo olvidarme de mis abuelos, gratitud que no puedo expresar en unas pocas líneas.

Profundamente agradecido a mis tutores, Jorge, Andrés y Marcelo, por darme la oportunidad, los conocimientos y los consejos que me han permitido realizar este trabajo.

A mis amigos, porque han seguido ahí, incluso después de pasar meses sin hablar con ellos, y a mis compañeros, que han hecho que aquellas mañanas en "la D27" fuesen más llevaderas, sin olvidarme de todos a los que he conocido en el laboratorio del DTIC.

Finalmente, quiero pedir disculpas a todas aquellas personas que se merecen mi agradecimiento pero no se sienten identificadas en estas líneas. Resulta imposible nombraros a todos.

Te agradezco a ti que tengas este trabajo en tus manos.

RESUMEN

Los sensores de propósito general RGB-D son dispositivos capaces de proporcionar información de color y de profundidad de la escena. Debido al amplio rango de aplicación que tienen estos sensores, despiertan gran interés en múltiples áreas, provocando que en algunos casos funcionen al límite de sensibilidad. Los métodos de calibración resultan más importantes, si cabe, para este tipo de sensores para mejorar la precisión de los datos adquiridos.

Por esta razón, resulta de enorme transcendencia analizar y estudiar el calibrado de estos sensores RGB-D de propósito general. En este trabajo se ha realizado un estudio de las diferentes tecnologías empleadas para determinar la profundidad, siendo la luz estructurada y el tiempo de vuelo las más comunes. Además, se ha analizado y estudiado aquellos parámetros del sensor que influyen en la obtención de los datos con precisión adecuada dependiendo del problema a tratar. El calibrado determina, como primer elemento del proceso de visión, los parámetros característicos que definen un sistema de visión artificial, en este caso, aquellos que permiten mejorar la exactitud y precisión de los datos aportados.

En este trabajo se han analizado tres algoritmos de calibración, tanto de propósito general como de propósito específico, para llevar a cabo el proceso de calibrado de tres sensores ampliamente utilizados: Microsoft Kinect, PrimeSense Carmine 1.09 y Microsoft Kinect v2. Los dos primeros utilizan la tecnología de luz estructurada para determinar la profundidad, mientras que el tercero utiliza tiempo de vuelo.

La experimentación realizada permite determinar de manera cuantitativa la exactitud y la precisión de los sensores y su mejora durante el proceso de calibrado, aportando los

mejores resultados para cada caso.

Finalmente, y con el objetivo, de mostrar el proceso de calibrado en un sistema de registro global, diferentes pruebas han sido realizadas con el método de registro μ-MAR. Se ha utilizado inspección visual para determinar el comportamiento de los datos de captura corregidos según los resultados de los diferentes algoritmos de calibrado. Este hecho permite observar la importancia de disponer de datos exactos para ciertas aplicaciones como el registro 3D de una escena.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1

INTRODUCCIÓN

1.1 MOTIVACIÓN Y CONTEXTO

Desde pequeño siempre he tenido curiosidad por el funcionamiento de las cámaras, que por aquel entonces, "las de carrete", eran las mas extendidas en los hogares. ¿Cómo un dispositivo de proporciones, en su momento, reducidas era capaz de obtener una representación tan fiel de la realidad? Esta incertidumbre se acentuó con la llegada de las cámaras digitales, cada vez de tamaños más reducidos y con mejores prestaciones.

Años más tarde, en las lecciones de óptica de la asignatura de física pude aprender los principios ópticos básicos de estos aparatos. Sin embargo, no fue hasta la universidad cuando empecé a conocer la visión por computador, un campo inmenso tanto en conocimientos como aplicaciones, y es que fue allí donde pude observar los primeros robots haciendo uso de cámaras para determinar sus acciones, además de conocer los fundamentos básicos de la adquisición y el posterior tratamiento de imágenes.

Cuando hace un año tuve la oportunidad de colaborar junto a Andrés Fuster, Jorge Azorín y Marcelo Saval en su grupo de investigación de visión artificial, no lo dudé. Entonces descubrí la trayectoria del grupo, iniciada con la tesis de Andrés "Modelado de sistemas para visión realista en condiciones adversas y escenas sin estructura" [Fuster Guilló, 2003] en la que se proponen soluciones para aplicar a diferentes situaciones donde no es posible obtener una imagen en la que se distingan los elementos que aparecen en la misma. Por ejemplo, objetos que aparezcan en penumbra o deslumbrados.

Continuando el estudio de los problemas de visión en situaciones adversas, Jorge Azorín propone en su tesis soluciones que permiten mejorar la inspección visual automática en situaciones de especularidad, titulada "Modelado de sistemas para visión de objetos especulares. Inspección visual automática en producción industrial" [Azorín López, 2007], analizando los requisitos que debe cumplir un sistema que se enfrente a estas situaciones y proponiendo técnicas basadas en proyecciones de luz para percibir defectos.

Dada la evolución de los sistemas de visión artificial en los últimos años hacia una percepción multidimensional del entorno, se está mostrando un mayor interés por la visión tridimensional. Por ello, se están desarrollando nuevas tecnologías y sistemas de percepción que permiten obtener una vista 3D del entorno, es decir, añaden la dimensión de profundidad a las imágenes 2D tradicionales. Esto ha dado lugar a que aparezcan cámaras con tecnologías de propósito general, conocidas también como "de bajo coste" o categorizadas como cámaras o sensores RGB-D (*Red Green Blue and Depth*), siendo Microsoft Kinect o PrimeSense Carmine los ejemplos más cercanos.

Siguiendo la línea de investigación del grupo, Marcelo Saval propone en su tesis "Methodology based on registration techniques for representing subjects and their deformations acquired from general purpose 3D sensors" [Saval Calvo, 2015] un conjunto de métodos para mejorar la adquisición y el registro de datos 3D obtenidos a través de cámaras RGB-D en situaciones adversas, donde el proceso se desarrolla en el límite de la sensibilidad del sensor, ya que se estudian sujetos con características difícilmente perceptibles.

En los sistemas de visión 3D por computador dedicados al análisis de la escena, a rasgos generales, se diferencian tres fases principales incluyendo la adquisición, el registro de las diferentes vistas y el análisis final, representadas en el diagrama de la Figura 1.1, donde la adquisición es la fase inicial donde se toman los datos, y por lo tanto donde intervienen los sensores, y donde se centra este trabajo. Cada una de estas fases está relacionada con la aplicación final, siendo la adquisición la que menos influenciada está y el análisis la que

tiene una vinculación más directa. Es por esto que los sensores suelen calibrarse con parámetros que permiten un rango de percepción amplio, para permitir su uso en un conjunto mayor de situaciones. El primer paso consiste en la obtención de los datos por parte del sensor. A continuación, en la parte de registro se realizan las modificaciones necesarias sobre los datos obtenidos en la parte de adquisición para obtener un modelo completo de la escena. Este paso puede dividirse en dos partes, la primera referente al registro rígido que transforma los datos del mismo modo para llevarlos a un mismo sistema de coordenadas. La segunda, el registro no rígido permite alinear los datos de forma independiente. Esto puede ser útil para analizar cambios en la forma, por ejemplo, la deformación producida en la forma de una planta por el paso del tiempo, o si se alinea el color, reducir efectos de iluminación como brillos o sombras. Finalmente, en la parte de análisis se extrae la información necesaria de los datos anteriores para el propósito específico del problema, siendo éste el que define los requerimientos de cada una de las tres fases anteriores. Un ejemplo de todo el proceso podría ser el análisis de la forma de un sujeto, el primer paso es obtener los datos desde distintos puntos de vista del mismo. A continuación, se registran todas las vistas para que formen el modelo completo, y en caso de ser necesario, aplicar el registro no rígido para refinar los datos (reducir los efectos de movimiento del sujeto durante la adquisición, como mover la cabeza si el sujeto es una persona). Finalmente el análisis permite obtener datos útiles de la reconstrucción anterior, como altura, volumen, colores, geometría, etc.

La adquisición de los datos resulta una parte crítica del sistema, dado que el resultado proporcionado por las fases posteriores depende en gran medida de los datos adquiridos. Esta fase depende de tres factores principales [Fuster Guilló, 2003, Azorín López, 2007], incluyendo los parámetros del entorno (iluminación, medio de transmisión...), los parámetros del motivo (tamaño, color, forma...) y los parámetros de la cámara (distancia focal, tamaño del sensor...). La cámara, o el calibrado de la misma, condicionan la adquisición debido a la sensibilidad que ofrece y la tecnología que utiliza para estimar las distancias.

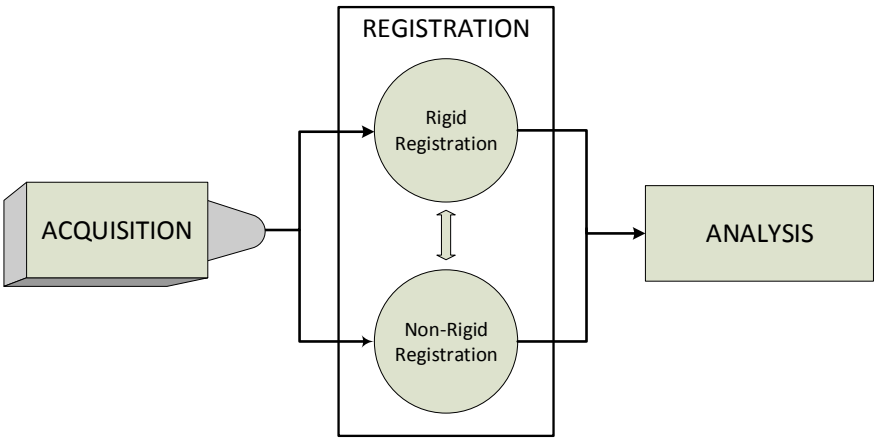


Figura 1.1: Esquema general de un sistema 3D

La fase de adquisición también se ve afectada por el entorno. Factores como la luz, las sombras y el punto de vista, entre otros, pueden afectar de manera significativa a los datos obtenidos. Por ejemplo, en el caso de los sensores que utilizan una proyección para calcular la profundidad su funcionamiento está restringido a lugares cerrados, ya que en abiertos, la luz solar es muy intensa y puede dificultar la correcta identificación del patrón proyectado proporcionando datos erróneos.

Por último, el motivo (sujeto u objeto de interés) también tienen un impacto importante en la fase de adquisición. Por ejemplo, en el caso de los sensores que proyectan colores sobre la escena para determinar la profundidad, pueden verse afectados por superficies especulares [Azorín López, 2007] o del mismo color que el patrón, dificultando la obtención de los datos. Por otra parte, si el motivo es intrincado, pueden producirse oclusiones que también dificulten su completa adquisición.

1.2 ESTADO DEL ARTE

El primer paso en los sistemas de visión 3D es la adquisición de datos. Actualmente existen varios tipos de sensores, los cuales se pueden dividir en dos grupos:

- De contacto. Obtiene la información 3D de un objeto a través del contacto directo con éste.
- Sin contacto. Son capaces de obtener la información 3D desde la distancia.

La mayoría de los sensores dentro del segundo grupo no están restringidos a situarse en una posición concreta, además suelen ser más rápidos y pueden proporcionar otros datos como el color o sonido (por ejemplo, el sensor Microsoft Kinect contiene cuatro micrófonos que permiten determinar la orientación del origen de un sonido, y junto con la detección de personas, saber que sujeto está hablando). Dentro del grupo de sensores sin contacto se pueden diferenciar cinco categorías diferentes:

- LIDAR (*Laser Illuminated Detection And Ranging*). Es una tecnología que se basa en proyectar un laser en la escena y analizar la reflexión para determinar la distancia y por tanto obtener un mapa de profundidad [Schwarz, 2010].
- Tiempo de vuelo (*Time-of-Flight, ToF*). Proyecta un haz de luz para determinar la distancia midiendo el tiempo transcurrido entre la emisión y el retorno [Foix et al., 2011, Cui et al., 2010].
- Cámaras estereoscópicas. Se trata de una tecnología pasiva que utiliza dos o más cámaras calibradas para estimar la profundidad calculando la disparidad entre las imágenes obtenidas por cada cámara y, conociendo los parámetros de calibrado, estimar la distancia [Lazaros et al., 2008].
- Luz estructurada (*structured light*). Al contrario que las cámaras esteroscópicas, se trata de una tecnología activa que proyecta un patrón conocido en la escena y calcula la disparidad entre el patrón observado y el conocido. Para ello es necesario conocer la distancia que existe entre el emisor y el receptor [Salvi et al., 2004, 2010, Herakleous and Poullis, 2014].
- Sensores RGB-D de propósito general. Son sensores que combinan diferentes técnicas de las nombradas (como luz estructurada o ToF) con cámaras de color para proporcionar de manera casi simultánea el color y mapa de profundidad de la escena. Se

caracterizan por ser sensores económicamente asequibles [Lai et al., 2013, Khoshelham and Elberink, 2012, Henry et al., 2012].

(a) Lidar

(b) ToF

(c) Luz estructurada

(d) Estereoscópica

(e) ToF

Figura 1.2: Diferentes sensores 3D

Algunos sistemas láser no proporcionan información del color de la escena, por lo que no se pueden obtener características basadas en el color. Otros pueden proporcionar color pero tienen un coste muy elevado. Sin embargo, existen otras alternativas para obtener el color utilizando estos sistemas, como por ejemplo añadir una cámara RGB y calibrarla para proporcionar la información de intensidad y profundidad alineadas. Además, el proceso de adquisición en algunos de estos sistemas es muy lento, añadiendo la restricción de que los objetos en la escena deben permanecer estáticos, por lo que limita su campo de aplicación. Las cámaras de tiempo de vuelo (en ocasiones clasificadas como una subcategoría de LIDAR) también suelen utilizar láser y tampoco suelen proporcionar información del color.

Las cámaras estereoscópicas se han usado para reconstrucción 3D, como en [Kasper et al., 2012], donde se emplea una Konica Minolta Vi-900 3D para ello. Esta tecnología es capaz de proporcionar información de color además de la de profundidad. Sin embargo, es necesario calibrar ambos sensores cada vez que se varía su emplazamiento, lo que dificulta su portabilidad. La distancia entre las cámaras se puede variar para ajustar el rango de trabajo del sistema. Por ejemplo, aumentado esa distancia se puede ampliar el alcance del sistema, pero también se incrementa la distancia mínima. Otro problema que presenta esta tecnología es la necesidad de textura para poder obtener la información del 3D de la escena, por lo que los resultados pueden no ser correctos si se aplica a una superficie sin texturas.

Los sensores basados en tecnología de luz estructurada proyectan un patrón para obtener el mapa de profundidad de la escena. Existen diferentes tipos de patrones de distintas formas, por ejemplo, basados en líneas. Uno de los inconvenientes de esta tecnología es su limitación a trabajar en situaciones donde la iluminación (luz solar o colores similares entre el patrón y el motivo) puede dificultar la identificación del patrón; otro es la proyección del patrón donde no se pueda distinguir las distintas partes del motivo (por ejemplo, si se utilizan patrones basados en puntos, las partes de los sujetos que haya entre los puntos no pueden ser detectados).

La categoría de sensores RGB-D se refiere a cámaras de propósito general de bajo coste que pueden proporcionar la información de color y profundidad de la escena de manera casi simultanea. A menudo se nombran como sensores 3D comerciales o sensores de bajo coste, pero generalmente se conocen como sensores RGB-D. Este tipo de sensores implementa alguna de las tecnologías nombradas anteriormente combinada con una cámara de color, pudiendo ser las tecnologías, por ejemplo, luz estructurada (Microsoft Kinect o PrimeSense) o tiempo de vuelo (Microsoft Kinect v2). Se caracterizan por ser de bajos coste y fácilmente manipulables, capaces de proporcionan información adecuada para múltiples aplicaciones, aportando una gran flexibilidad. La información de color y profundidad se proporciona de manera casi simultánea, con una diferencia de tiempo tan reducida que se puede asumir como simultánea para muchas aplicaciones.

Khoshelham and Elberink [Khoshelham and Elberink, 2012] han realizado un estudio de la precisión del sensor Microsoft Kinect. Además, existen varios artículos que proponen algoritmos y aplicaciones utilizando los sensores RGB-D [Han et al., 2013, Morell-Gimenez et al., 2014, Shao et al., 2014]. Por ejemplo, en [Weiss et al., 2011] utilizan una cámara Microsoft Kinect para obtener un modelo 3D de una persona a través de múltiples vistas rotando alrededor cuerpo, obteniendo buenos resultados y evitando usar dispositivos más costosos. En [Lovato et al., 2014] utilizan el sensor Carmine 1.09 de PrimeSense para realizar reconstrucciones 3D del pie, tomando varias capturas alrededor del objetivo y utilizando realidad aumentada. En [Jedvert, 2013] y [Paier, 2011] también utilizan el sensor Microsoft Kinect para obtener un modelo 3D de la cabeza, en el caso del primero con el objetivo de obtener un modelo con una alta calidad de texturas utilizando GPUs y una variante de Kinect Fusion [Izadi et al., 2011] con altas restricciones temporales. En el caso del segundo, se realiza el modelo para obtener una identificación del sujeto en un sistema de seguridad. Además, en [Smisek et al., 2011] demuestran que los parámetros por defecto que utiliza el sensor Microsoft Kinect no son lo suficientemente precisos para muchas aplicaciones.

Por lo tanto, para obtener unos resultados precisos con estos sensores, la mayoría de estos trabajos realizan un proceso de calibrado. [Herrera C. et al., 2011] propone un algoritmos para calibrar los parámetros intrínsecos de ambas cámaras y además los necesarios para convertir la disparidad a metros. [Zhang and Zhang, 2011] extiende este trabajo añadiendo la necesidad de buscar las correspondencias entre el color y la profundidad de las esquinas del patrón de calibrado.. Al mismo tiempo [Burrus, 2012] utiliza OpenCV para realizar el calibrado intrínseco y extrínseco del sensor Microsoft Kinect. [Smisek et al., 2011] observó errores residuales en la profundidad, estimando un error fijo (en la profundidad) para cada píxel, lo que permite determinar un patrón de corrección para la información obtenida. Más tarde, [Daniel Herrera C et al., 2012] propone una corrección de la distorsión en la profundidad. [Raposo et al., 2013] mejora esta propuesta obteniendo resultados similares con un número menor de imágenes. Recientemente [Staranowicz et al., 2014] propone un método para estimar los parámetros a partir de un objeto esférico, como un balón, utilizando la transformada de Hough y realizando una minimización no lineal para obtener los resultados.

Tras el estudio del estado del arte referente a sensores 3D, concretamente en los RGB-D, y los calibrados propuestos, se pueden concluir distintos aspectos.

Existen cinco tipos generales de sensores 3D con distintas características. Particularmente, los sensores de propósito general RGB-D han demostrado ser capaces de adaptarse a un amplio espectro de situaciones, sin embargo, su sensibilidad no es apta para ciertos problemas donde se requiere de mayor exactitud de los datos. Por ello se han propuesto distintos métodos de calibrado de estas cámaras.

De ahí que sea necesario realizar un estudio del calibrado para sensores RGB-D, analizando la tecnología que hace posible su funcionamiento para detectar donde se pueden encontrar posibles mejoras. Además, se observa interés de disponer de comparativas entre algunos de los algoritmos más conocidos, incluyendo desde calibraciones para cámaras a

nivel general [Bouguet, 2004], como otros más específicos para estos sensores [Daniel Herrera C et al., 2012].

1.3 OBJETIVOS

El objetivo de este trabajo es estudiar y analizar cuantitativamente el calibrado de sensores RGB-D, para ello se plantean tres subobjetivos:

- Dado el alto interés de este tipo de dispositivos, el primer subobjetivo es conocer las tecnologías que utilizan y los parámetros del calibrado de los sensores RGB-D más utilizados, además, el funcionamiento para obtener información tridimensional de la escena.
- Analizar y comparar los métodos de calibrado más comunes para este tipo de sensores.
- Estudiar su influencia en el registro de secuencias 3D como caso práctico de un sistema de visión por computador.

1.4 PROPUESTA FORMAL

El Modelo de Visión Activa propuesto por Andrés Fuster [Fuster Guilló, 2003] presenta una formulación para la adquisición bajo condiciones adversas. Más tarde fue ampliada por Jorge Azorín [Azorín López, 2007] para la adquisición de superficies especulares. Recientemente, Marcelo Saval [Saval Calvo, 2015] ha actualizado la formulación para representación 3D usando sensores RGB-D.

El objetivo principal de este modelo es describir el proceso de visión artificial para definir formalmente los elementos que intervienen en el proceso de adquisición y además

proporcionar soluciones a los problemas en entornos reales y/o en condiciones adversas. Se entiende por situación o condición adversa aquellas adquisiciones en las que los parámetros a adquirir son de difícil percepción para el sistema de adquisición dado.

Una imagen I se define como una representación bidimensional proporcionada por F, siendo F la función que modela el Sistema de Adquisición Visual (SAV). Esta función incluye el equipo y la configuración de la escena necesarios para captura una imagen: iluminación, posiciones, puntos de vista, cámaras, etc. El espacio P representa las magnitudes físicas que intervienen en el proceso de percepción visual y está compuesto por vectores ρ que representan las magnitudes de la escena que contribuyen a la formación de I (Ecuación 1.1).

$$I(x, y) = F(\rho), \quad \rho = (\phi_1, \phi_2, \phi_3, ..., \phi_n) \in P \tag{1.1}$$

Los componentes ϕ_i de los vectores de las magnitudes de la escena pueden ser: escala, punto de vista, intensidad de la luz, frecuencia, saturación, etc. Cada componente puede ser modelado como una función dependiente de tres entradas (Ecuación 1.2): el motivo de interés, o, en la escena (objeto que será adquirido), el entorno, e, en el que se sitúa el motivo y la cámara, y finalmente la cámara c que obtiene imágenes I de la escena. Cada ϕ_i está compuesto por estas tres entradas, por lo tanto, ρ compuesto por varios ϕ_i, también se verá afectado por estas entradas (Ecuación 1.2).

$$\phi_i = \phi_i(o, e, c), \quad \rho = \rho(o, e, c) \tag{1.2}$$

La composición de cada elemento (o,e,c) se puede expresar como un vector con varias magnitudes: ϵ_i para el entorno (Ecuación 1.3), γ_i para la cámara (Ecuación 1.4) y μ_i relacionado con el motivo (Ecuación 1.5). La intensidad y la longitud de onda de la fuente de luz, el medio de transmisión, la posición relativa entre la escena y la cámara son ejemplos de elementos del entorno ϵ_i. Respecto a las magnitudes de la cámara γ_i, intervienen las características del sensor, la óptica y los elementos electrónicos: zoom, enfoque, diafragma,

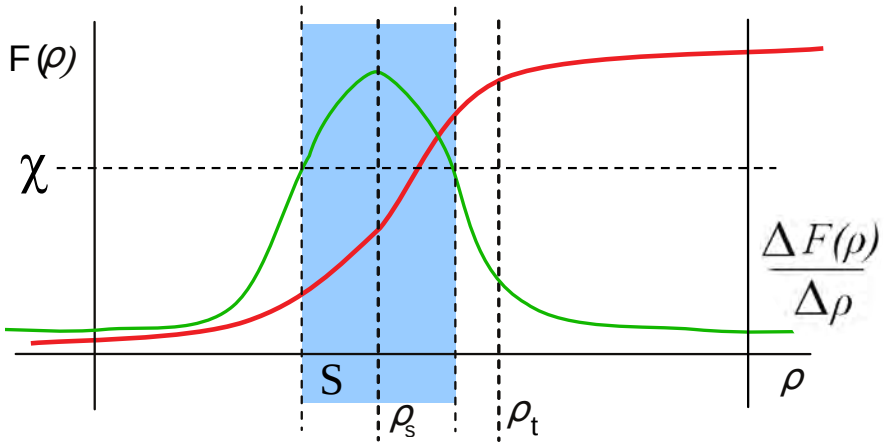


Figura 1.3: Curva de calibración (rojo) y sensibilidad (verde) para un Sistema de Adquisición Visual incluyendo el punto de sintonización ρ_s, el punto de trabajo ρ_t y el área S.

tamaño del sensor, etc. Finalmente, el color, la forma o la topología son ejemplos de elementos μ_i pertenecientes al motivo. Los valores que componen cada vector son elementos del conjunto O para el motivo, E para el entorno y Γ para la cámara.

$$E = \{e_0, e_1, e_2, ...\}, e_i = (\epsilon_1, \epsilon_2, ..., \epsilon_n) \tag{1.3}$$

$$\Gamma = \{c_0, c_1, c_2, ...\}, c_i = (\gamma_1, \gamma_2, ..., \gamma_l) \tag{1.4}$$

$$O = \{o_0, o_1, o_2, ...\}, o_i = (\mu_1, \mu_2, ..., \mu_m) \tag{1.5}$$

Para definir el modelo es importante el concepto de sensibilidad como una característica estática del sensor. La sensibilidad es la pendiente de la curva de calibración (Figura 1.3), esta última refiriéndose al cambio detectable en la salida para el mínimo cambio en las magnitudes de entrada. En este caso, la salida detectable en F para el mínimo cambio de las magnitudes de la escena se expresa en la ecuación Ecuación 1.6.

$$sensitivity = \frac{\Delta F(\rho)}{\Delta \rho} \tag{1.6}$$

Los parámetros por defecto que emplea una cámara están optimizados para un subconjunto de ρ pertenecientes a P. El punto de sintonización ρ_s, es el punto en el que las

magnitudes de la escena en el espacio P para cada cámara del conjunto Γ en el que la sensibilidad del Sistema de Adquisición Visual es óptima.

El punto de trabajo ρ_t es el punto en el que las magnitudes de la escena definen un punto en P en el que trabaja el sistema.

Por lo tanto, en un Sistema de Adquisición Visual en condiciones idóneas, el punto de trabajo ρ_t debe ser cercano al punto de sintonización ρs. Cuando se encuentran muy alejados, entonces el sistema trabaja bajo condiciones adversas.

Para un punto ρ_i específico, $\Omega^{\Phi}_{\rho_i}$ se define como el conjunto de magnitudes distinguibles en F, es decir, $F(\rho_j)$ debe ser diferenciable de $F(\rho_i)$ para ser incluido en $\Omega^{\Phi}_{\rho_i}$.

S se define como el conjunto de magnitudes de la escena que pueden ser distinguibles en F dado un *threshold* χ dependiente del problema a resolver. Esto es, para que una magnitud ρ_i se incluya en S debe ser detectado de forma individual por el Sistema de Adquisición Visual, o lo que es lo mismo, ser mayor que χ (estar por encima de la sensibilidad requerida). El conjunto de magnitudes ρ_j que no se incluyen en S se encuentran bajo condiciones adversas S^c y no son adquiribles por el sistema, siendo S y S^c complementarios.

Por lo tanto $F(\rho_i) = F(\rho_j)$ si $\rho_i, \rho j \in S^c$. Para conseguir que el sistema diferencie ρ_i y ρ_j el Modelo de Visión Activa propone modificar las entradas que intervienen en las magnitudes del sistema: motivo, entorno y cámara. El motivo (objeto o sujeto) se considera constante por ser el objetivo de adquisición del sistema. Sin embargo, las condiciones del entorno y las características de la cámara pueden ser modificadas para incluir en S vectores de magnitudes $\rho \in S^c$, es decir, mejorar la adquisición bajo condiciones adversas. Esto se puede llevar a cabo de dos formas distintas:

- Minimizando la distancia entre el punto de trabajo ρ_t y el punto de sintonización ρ_s.
- Condicionar la adquisición del Sistema de Adquisición Visual.

Por un lado, minimizar la distancia entre el punto de trabajo ρ_t y el punto de sintonización ρ_s consiste en desplazar cualquiera de los dos puntos de forma que el punto de trabajo sea un elemento del conjunto S ($\rho_t \in S$). Por otro lado, el objetivo de condicionar

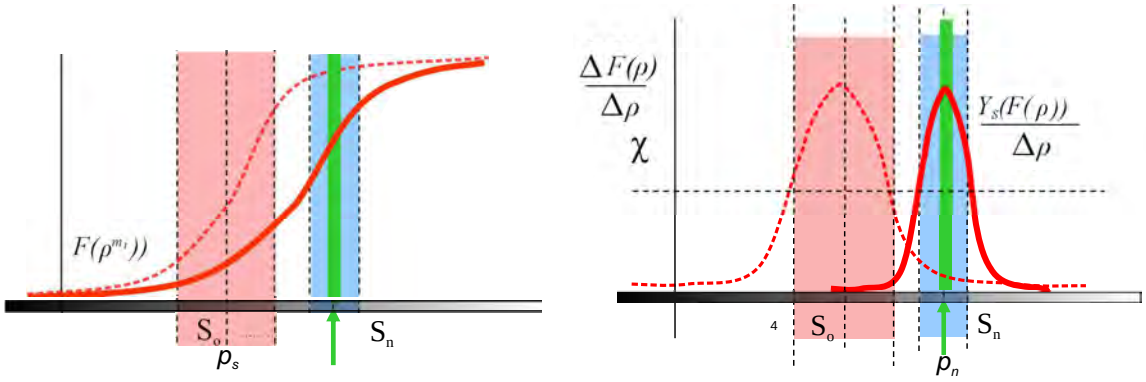


Figura 1.4: Curvas de calibración y sensibilidad para una transformación γ_S desplazando el punto de sintonización ρ_s al punto de trabajo ρ_t. Las lineas discontinuas y zonas sombreadas en rojo representan la calibración antigua y S_o, respectivamente. Las lineas rojas continuas y la zonas sombreadas en azul representan la nueva curva de calibración y S_n. El punto de sintonización antiguo esta representado por ρ_s y ρ_n el nuevo correspondiente después de aplicar la transformación.

la adquisición del SAV ampliar la distancia entre las magnitudes de escena a percibir. Esto puede llevarse a cabo modificando los parámetros ϵ_i, como la propuesta de [Azorín López, 2007] que proyecta patrones de luz; o modificando los parámetros γi, como la propuesta de [Saval Calvo, 2015], que introduce nuevos espacios en la adquisición (color, forma, orientación...).

Este trabajo se centra en la parte del sensor, por lo tanto la propuesta se centra en el caso de reducir la distancia entre ρ_s y ρ_t, concretamente llevando ρ_s hacia ρ_t. Esto, representado en la Figura 1.4, puede entenderse como encontrar los valores γi (parámetros de la cámara) que se adaptan al problema o escena dados. Para ello hay distintas técnicas que se explican en el Capítulo 2.

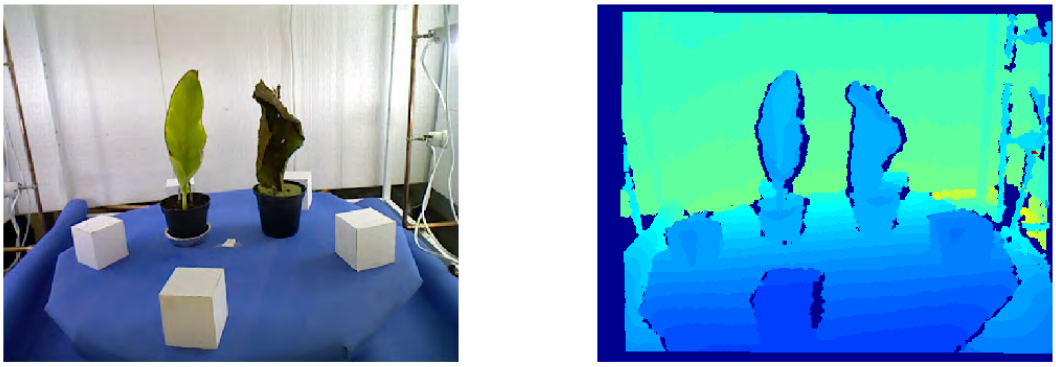
2

CALIBRADO DE SENSORES RGB-D

Calibrar es el proceso para determinar los parámetros que definen al sistema. En el caso de un sistema de adquisición, una cámara, sería encontrar los parámetros como distorsión, tamaño del sensor, etc.

Habitualmente, los sistemas llevan una calibración por defecto, por lo tanto, el calibrado suele ser el proceso de ajustar un determinado instrumento o sistema para trabajar con la mayor exactitud posible. Sin embargo, esta definición de calibrar es muy amplia, por ejemplo, se puede hablar de calibrar un monitor de ordenador o de calibrar los frenos de una bicicleta y se trataría de dos procesos completamente distintos pero con un mismo objetivo, ajustar un dispositivo o sistema para que su funcionamiento sea óptimo.

En ese sentido, con el calibrado de sensores RGB-D se pretende encontrar aquellos parámetros característicos de un sensor determinado permitiendo mejorar su exactitud. Estos sensores son capaces de obtener de la escena información de color (Figura 2.1a), y de profundidad (Figura 2.1b).



(a) Imagen de color (b) Imagen de profundidad

Figura 2.1: Representación del punto principal

2.1 ESTUDIO DE PARÁMETROS DE CALIBRADO DE SENSORES RGB-D

En términos generales, los sistemas de adquisición tienen varios parámetros que son en muchos casos independientes de la tecnología utilizada para estimar la profundidad.

En este apartado se explican los principales parámetros a tener en cuenta, tanto de la cámara de color (RGB) como la de profundidad (*Depth*).

2.1.1 *Distancia focal*

La distancia focal es una propiedad física de la lente que determina la distancia a la que se encuentra el punto a través del cual pasan todos los rayos paralelos al eje óptico que incidan sobre la lente, Figura 2.2.

Sin embargo, en fotografía, la distancia focal es la distancia que existe entre el centro óptico y el sensor fotosensible, situado en el plano focal, en el cual se formará la imagen,

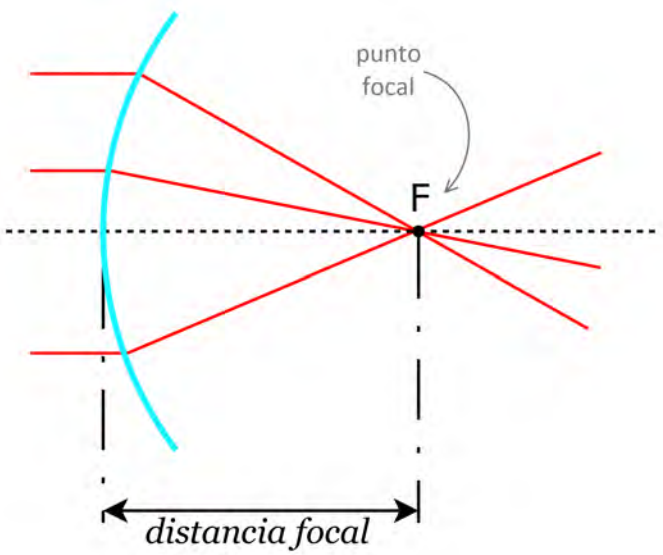


Figura 2.2: Distancia focal de una lente

Figura 2.3.

Los sensores de las cámaras se basan en el efecto fotoeléctrico, que es la capacidad de un material a emitir electricidad cuando incide sobre él una radiación electromagnética, y están compuestos por varias células denominadas fotositos, concretamente uno para cada píxel. Esta célula produce una corriente eléctrica que varía en función de la intensidad de luz recibida.

Las células en los sensores de bajos coste no son perfectamente cuadradas, son rectangulares. Por esta razón se manejan dos magnitudes para la distancia focal, una horizontal y otra vertical, f_x y f_y. Siendo F la distancia focal y s_x, s_y la dimensión de la célula (tamaño del píxel), tenemos que las distancias focales f_x y f_y se definen con las Ecuaciones 2.1 y 2.2.

$$f_x = F s_x \tag{2.1}$$

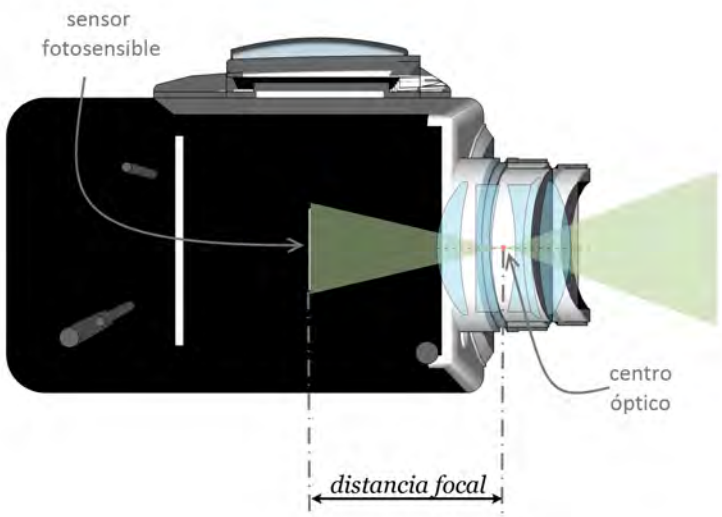


Figura 2.3: Distancia focal

$$f_y = Fs_y \tag{2.2}$$

Es importante destacar que s_x y s_y no pueden ser obtenidos a través de ningún proceso calibración, como tampoco se puede obtener la distancia focal F. Actualmente, sólo es posible estimar f_x y f_y a través de métodos de calibrado. Para conocer en detalle los valores de F, s_x y s_y sería necesario desmontar la cámara y medir los componentes directamente [Zelinsky, 2009].

La distancia focal permite estimar el campo de visión (*field of view, FoV*) de la cámara conociendo el formato del sensor. El campo de visión es la distancia cubierta por la cámara a una cierta profundidad. Por otro lado, se define el ángulo de visión como la porción de la escena incluida en la imagen, es decir, el número de grados incluidos en la imagen. Por ejemplo, una cámara con una ángulo de visión de 90° a un metro de distancia tendrá un campo de visión de 2 metros (Figura 2.4). Estos valores se asocian con la Ecuación 2.3.

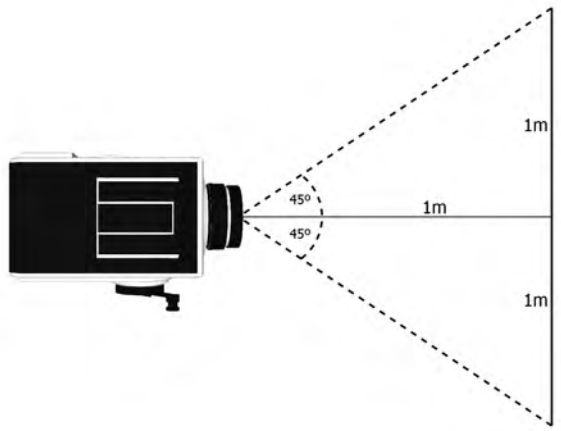


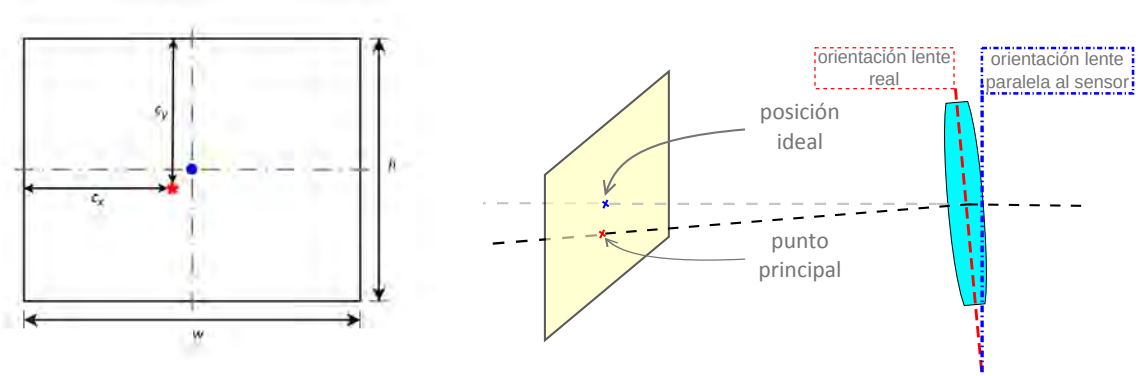
Figura 2.4: Campo de visión de una cámara

$$FoV = 2 * arctan(\frac{size}{2 * f}) \tag{2.3}$$

2.1.2 *Punto principal*

El punto principal representa el desplazamiento del eje óptico respecto al sensor, lo que provoca un desplazamiento del centro de proyección en la imagen. Como se muestra en la Figura 2.5a, en una imagen de dimensiones $w \times h$ el centro de proyección debería estar situado en el punto $(\frac{w}{2}, \frac{h}{2})$ (circulo azul), sin embargo, debido al desplazamiento del eje óptico, se encuentra en el (c_x, c_y) (asterisco rojo).

La Figura 2.5b muestra un ejemplo en el que se aprecia, en azul, un alineamiento perfecto del eje óptico, y por lo tanto del punto principal con el centroide del sensor, y en rojo un caso en el que eje no está perfectamente alineado debido a la orientación de la lente, provocando un desplazamiento del punto principal respecto al centroide del sensor.



(a) (b)

Figura 2.5: Representación del punto principal

A menudo, el punto principal se expresa junto a la distancia focal en una matriz de 3×3 denominada matriz de intrínsecos, como se muestra en la Ecuación 2.4, donde f_x, f_y son las distancias focales y c_x, c_y el punto principal.

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

2.1.3 *Distorsiones ópticas*

Otro parámetro γ_i del calibrado de la cámara son las distorsiones ópticas, que se producen como resultado de defectos o fallos en el diseño de las lentes. Los dos tipos de distorsión más comunes son la distorsión radial y la tangencial. La distorsión radial se produce debido a que las lentes no son perfectamente parabólicas, especialmente en cámaras de bajo coste, donde suelen ser esféricas, ya que son menos costosas de fabricar. Como se muestra en la Figura 2.6, los haces de luz que inciden en una posición alejada del centro en la lente esférica (izquierda) no convergen en el mismo punto que los haces más cercanos al centro,

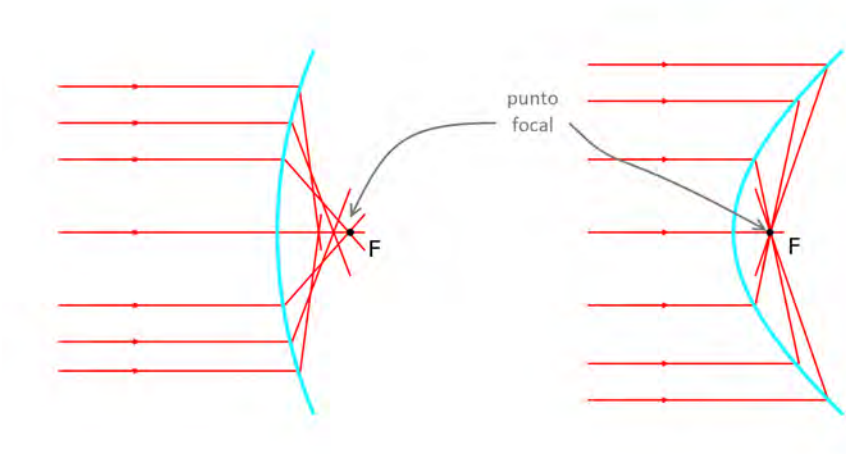


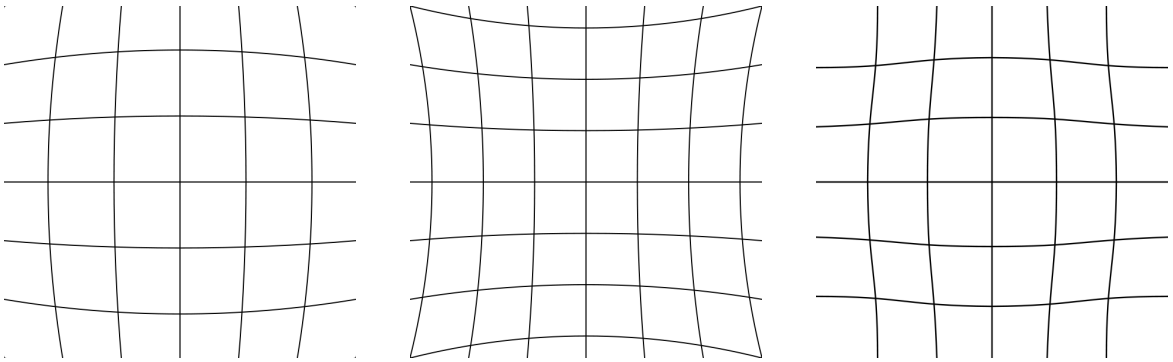
Figura 2.6: Diferencia entre lente esférica (izquierda), y parabólica (derecha)

provocando que parte de la imagen no se forme en el plano correspondiente. No ocurre lo mismo cuando la forma de la lente es parabólica (derecha), en la que los haces de luz alejados del centro convergen en el mismo punto que el resto, el punto focal.

Entre los efectos que tiene la distorsión radial sobre la geometría de la imagen, destacan: de barril, cojín y bigote. La distorsión de barril (*barrel or fish eye distortion*), Figura 2.7a, provoca que los píxeles se alejen del centro de la imagen a medida que se incrementa la distancia con el punto principal de la imagen, es el efecto más común en los sensores RGB-D. La distorsión de cojín (*pincushion distortion*), Figura 2.7b, produce el efecto contrario que la distorsión de barril, desplazando los píxeles hacia el centro de la imagen a medida que se incrementa la distancia con el punto principal. Por último, la distorsión de bigote (*moustache distortion*), Figura 2.7c, produce un efecto que es la combinación de los dos anteriores, provocando una distorsión de barril que se convierte en distorsión de cojín a medida que se incrementa el radio con el punto principal.

Por otro lado, la distorsión tangencial se produce debido al descentrado de la lente, es decir, la lente no se encuentra totalmente paralela al sensor. Como consecuencia, estas dis-

torsiones tiene unos efectos sobre la imagen final, entre ellos, el desplazamiento del punto principal [Ghosh, 2005] (Figura 2.5). La Figura 2.8 muestra un ejemplo de los efectos de las distorsiones radial y tangencial. La primera produce un desplazamiento en el radio dr, tomando como centro el punto principal. El desplazamiento de la segunda es angular dt, teniendo como eje de giro, también, el punto principal.



(a) Distorsión de barril (b) Distorsión de cojín (c) Distorsión de bigote

Figura 2.7: Distorsiones ópticas

Las distorsiones de una cámara se suelen expresar en un vector de distorsiones, como se muestra en la Ecuación 2.5, siendo k_1, k_2, k_3 los parámetros de distorsión radial y p_1, p_2 los de distorsión tangencial.

$$\begin{bmatrix} k_1 & k_2 & p_1 & p_2 & k_3 \end{bmatrix} \tag{2.5}$$

La distorsión radial es cero en el punto principal, y se incrementa a medida que aumenta la distancia entre el píxel y el punto principal. Se caracteriza por los primeros términos de la serie de Taylor. Sin embargo, puede simplificarse como en las Ecuaciones 2.6 y 2.7, para la distorsión radial, y en las Ecuaciones 2.8 y 2.9 para la tangencial [Zelinsky, 2009]. Siendo x e y la posición original de un píxel, $x_{corrected}$ e $y_{corrected}$ la nueva posición para ese píxel y k_1, k_2, p_1, p_2 y k_3 el vector con los coeficientes de distorsión.

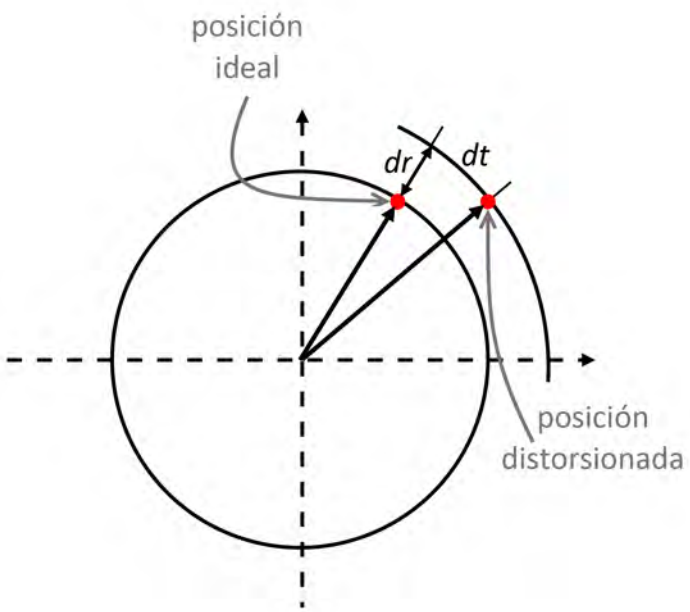


Figura 2.8: Ejemplo del desplazamiento de un píxel producido por las distorsiones [Weng et al., 1992]

$$x_{corrected} = x * (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2.6}$$

$$y_{corrected} = y * (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2.7}$$

$$x_{corrected} = x + [2p_1 y + p_2(r^2 + 2x^2)] \tag{2.8}$$

$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 x] \tag{2.9}$$

2.1.4 *Nube de puntos*

Conociendo los parámetros intrínsecos de la cámara infrarroja, se puede obtener una nube de puntos a partir de los píxeles de la imagen de profundidad. La nube de puntos es la representación 3D de la escena, en la que cada punto se caracteriza por una posición concreta dada por unas coordenadas (x, y, z). Por ejemplo, un cubo situado a un metro de la cámara tiene un tamaño 100 píxeles en la imagen y mide 10 centímetros en la nube de puntos, sin embargo, el mismo cubo situado a una distancia de 3 metros ocupa píxeles en la imagen pero sigue midiendo 10 centímetros en la nube de puntos. Estás coordenadas se pueden obtener a partir de un píxel de la imagen en la posición (u, v) utilizando las Ecuaciones 2.10, 2.11 y 2.12.

$$P3D.x = \frac{(u - c_{xd}) * Z}{f_{xd}} \tag{2.10}$$

$$P3D.y = \frac{(v - c_{yd}) * Z}{f_{yd}} \tag{2.11}$$

$$P3D.z = Z \tag{2.12}$$

Siendo f_{xd}, f_{yd} y c_{xd}, c_{yd} la distancia focal y el punto principal de la cámara infrarroja; P3D un punto de la nube de puntos con coordenadas (x, y, z); y Z_d la profundidad en el píxel (u, v) de la imagen de profundidad.

2.1.5 *Transformaciones*

Los sistemas formados por dos o más cámaras provocan que se obtengan varias imágenes con sistemas de referencia distintos. Por ello es necesario aplicar transformaciones sobre las imágenes obtenidas de manera que estén situadas en un eje de coordenadas común. Estas transformaciones que se aplican dependen de la distancia que separa ambas cámaras, conocida como *baseline*, y de la orientación de cada una ellas, ya que no suelen estar situadas completamente paralelas (Figura 2.9).

El proceso de situar las imágenes obtenidas por ambas cámaras en un mismo sistema de referencia se conoce como alineamiento. Para ello es necesario conocer las matrices de rotación y traslación que permiten realizar este proceso. Por ejemplo, en la Figura 2.10 se muestran dos conjuntos de puntos, cada uno con un sistema de referencia distinto. Para alinear el conjunto de la izquierda al de la derecha, es necesario aplicar una rotación que permita que los puntos de ambos conjuntos tengan la misma orientación, y una traslación que los desplace y sitúe en la misma posición.

Estas transformaciones se pueden aplicar a la nube de puntos para alinearlos con la imagen de color. La Ecuación 2.13 muestra como se aplica una rotación y traslación a un punto 3D P3D, siendo R una matriz de 3×3 para la rotación y T un vector de tamaño 3×1 indicando la rotación.

$$P3D' = R * P3D + T \tag{2.13}$$

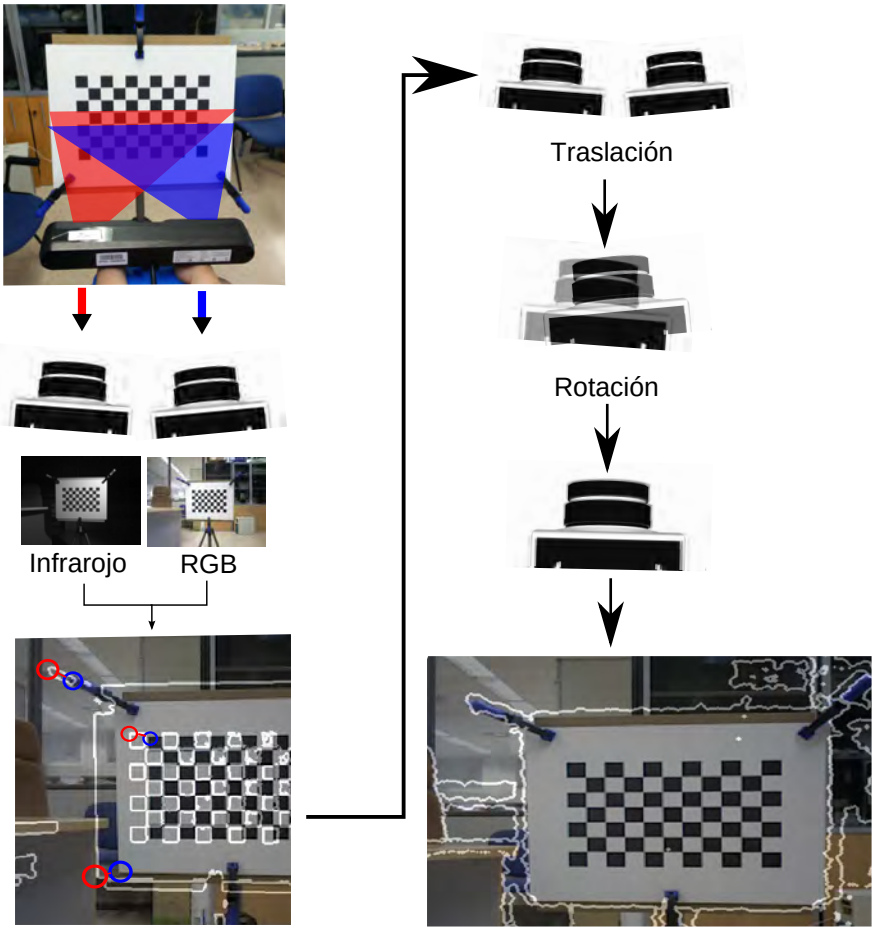


Figura 2.9: Proceso de alineamiento extrínseco entre imágenes de profundidad y RGB

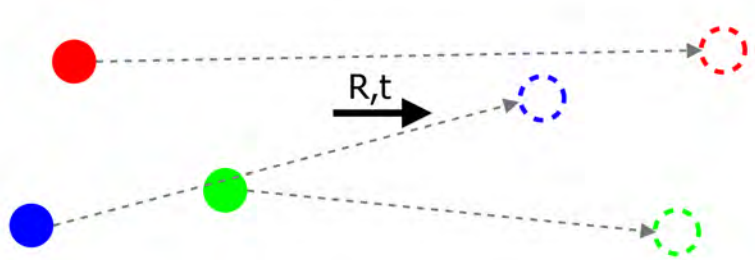


Figura 2.10: Transformaciones para alinear dos conjuntos de puntos

Obteniendo un punto $P3D'$ que ha sido desplazado y rotado respecto a un sistema de coordenadas. A continuación, es posible proyectar la nube de puntos sobre la imagen para conocer las correspondencias entre la nube de puntos y el color, utilizando las Ecuaciones 2.14 y 2.15.

$$P2D.i = (\frac{P3D'.x * f_{xrgb}}{P3D'.z}) + c_{xrgb} \tag{2.14}$$

$$P2D.j = (\frac{P3D'.y * f_{yrgb}}{P3D'.z}) + c_{yrgb} \tag{2.15}$$

Siendo f_{xrgb}, f_{yrgb} y c_{xrgb}, c_{yrgb} la distancia focal y el punto principal de la cámara RGB. Como resultado se obtiene un punto P2D de coordenadas (i, j) indicando el píxel de color correspondiente a cada $P3D'$.

2.2 TECNOLOGÍAS DE SENSORES RGB-D

Dependiendo de la tecnología utilizada para la obtención del 3D, los sensores tienen unas características determinadas. En esta sección se pretende repasar las tecnologías moyormente utilizadas en sensores RGB-D.

2.2.1 *Luz estructurada*

Los sensores RGB-D basados en luz estructurada están compuestos por una cámara RGB, una infrarroja y un emisor infrarrojo como se muestra en la Figura 2.11[1]. La técnica de luz estructurada se basa en el reconocimiento de un patrón conocido, previamente proyectado en la escena, para determinar la distancia entre el objeto y la cámara analizando las defor-

1 Imagen obtenida de iFixit *https://www.ifixit.com/Teardown/Microsoft+Kinect+Teardown/4066* última visita: 1 de Abril, 2015

Figura 2.11: Componentes del sensor Microsoft Kinect

maciones que sufre el patrón sobre la superficie del objeto (Figura 2.12).

Estos sensores proyectan sobre la escena un patrón de motas conocido (*speckle pattern* en inglés), como el de la Figura 2.13, utilizando el emisor infrarrojo. Posteriormente se obtiene a través de la cámara infrarroja y se analizan las variaciones de intensidad que se producen en cada punto para determinar la distancia. Este análisis se conoce con el nombre de interferometría de moteado. La Figura 2.14, fila inferior, muestra el proceso de estimación de la profundidad en las imágenes. Primero se obtiene el infrarrojo, a continuación se estima la disparidad y finalmente se obtiene la profundidad.

La relación entre esa disparidad y la profundidad se muestra en la Figura 2.15, donde para determinar la profundidad Z_k de un punto k del plano del objeto, el sensor compara el patrón adquirido con uno de referencia previamente conocido a una distancia Z_o, siendo o un punto del patrón de referencia que se corresponde con k en el plano del objeto a una distancia Z_k, provocando un desplazamiento horizontal D en el plano del objeto que es percibido por el sensor como una disparidad d. Mediante tiangulación es posible determinar la distancia Z_k a la que se encuentra el objeto a través de las igualdades expresadas en las Ecuaciones . 2.16 y 2.17.

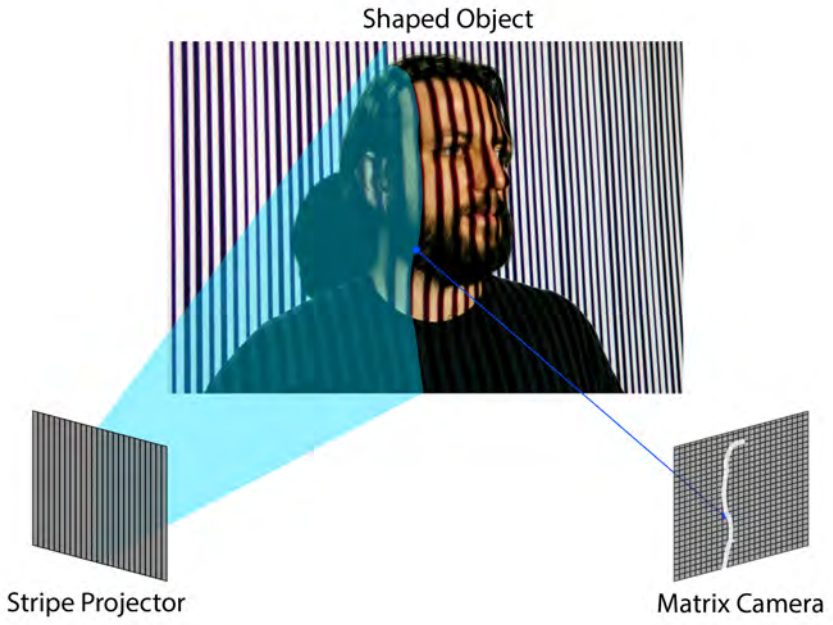


Figura 2.12: Técnica de luz estructurada

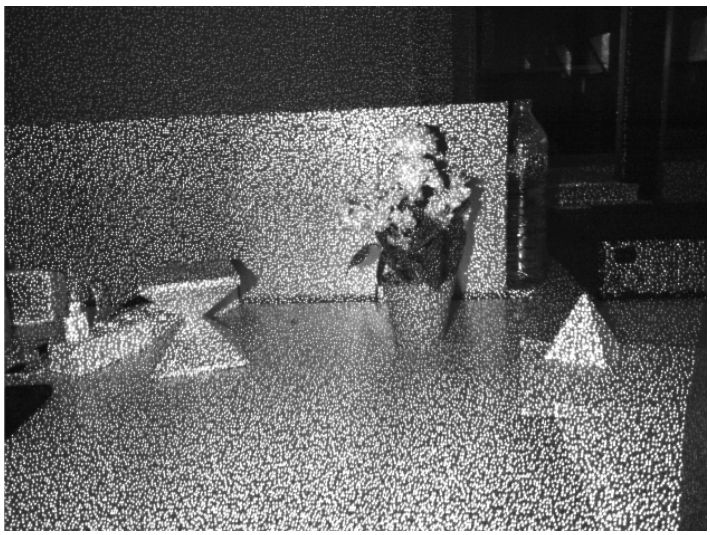


Figura 2.13: Patrón proyectado

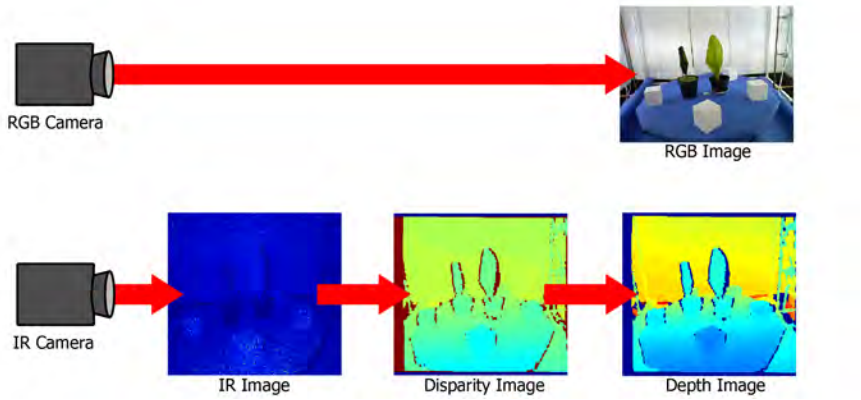


Figura 2.14: Proceso de adquisición

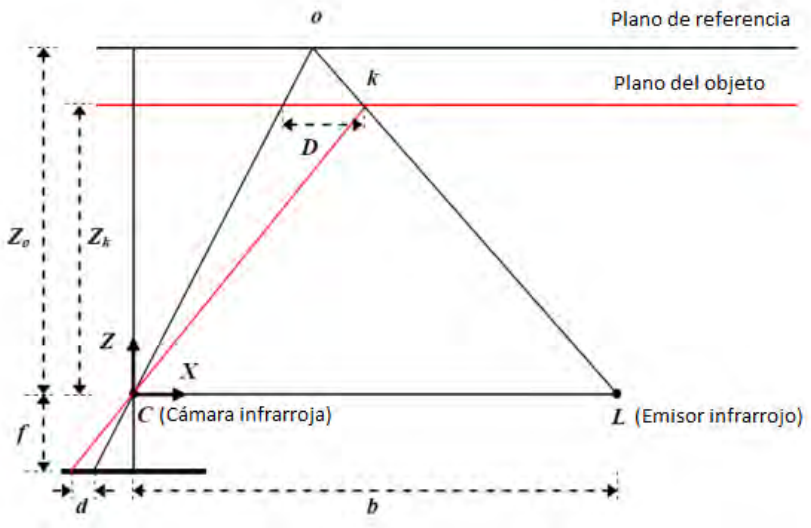


Figura 2.15: Relación entre la profundidad y la disparidad obtenida

$$\frac{D}{b} = \frac{Z_o - Z_k}{Z_o} \tag{2.16}$$

$$\frac{d}{f} = \frac{D}{Z_k} \tag{2.17}$$

Donde Z_k es la distancia al punto k en el plano del objeto, b es *baseline* o distancia entre la cámara y el emisor infrarrojos, y f es la distancia focal de la cámara infrarroja. Sustituyendo D de la ecuación 2 en la ecuación 1, se obtiene el valor de Z_k expresado en la Ecuación 2.18:

$$Z_k = \frac{Z_o}{1 + \frac{Z_o}{f} d} \tag{2.18}$$

Sin embargo, la disparidad obtenida por el sensor se normaliza en el rango de 0 a 2047 para posteriormente transmitirla utilizando 11 bits. Esta normalización es lineal, siendo del tipo $d = md' + n$ con d la disparidad sin normalizar y d' la disparidad normalizada. Por lo tanto, reemplazando la normalización anterior en la Ecuación 2.18, se obtiene una nueva Z_k^{-1} de la forma expresada en la Ecuación 2.19.

$$Z_k^{-1} = (\frac{m}{fb})d' + (Z_o^{-1} + \frac{n}{fb}) \tag{2.19}$$

La Ecuación 2.19 expresa la relación linear entre la inversa de la profundidad de un punto y su disparidad normalizada correspondiente. Aunque los coeficientes de esta relación lineal pueden ser estimados observando la disparidad correspondiente a varios puntos situados a distancias conocidas, al incluir los parámetros de normalización no se puede determinar b y Z_o por separado.

Dentro de los sensores RGB-D que utilizan esta tecnología destacan: Kinect, Figura 2.16a, el sensor desarrollado por Microsoft; Carmine 1.09 de PrimeSense, Figura 2.16b y Xtion de Asus, Figura 2.16c. Todos ellos se basan en la patente de PrimeSense [Freedman et al., 2012], por lo que utilizan el procesador PS1080, desarrollado por esta misma compañía, para la adquisición de los datos.

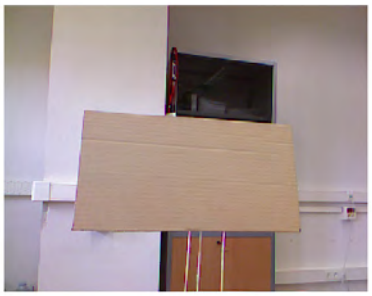
(a) Microsoft Kinect

(b) PrimeSense Carmine 1.09

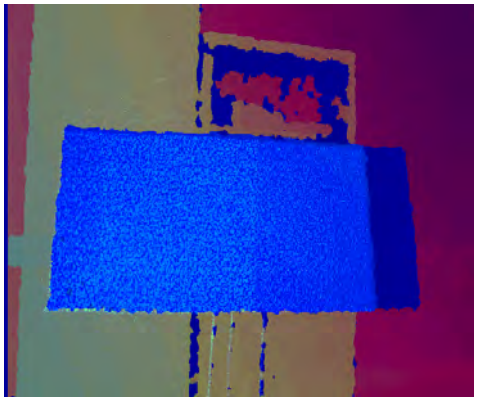
(c) Asus Xtion Pro Live

Figura 2.16: Sensores RGB-D que utilizan luz estructurada para determinar la profundidad

Los sensores que utilizan esta tecnología presentan un desplazamiento entre la imagen infrarroja y de profundidad del orden de 2 a 4 píxeles. Puede ser corregida obteniendo ambas imágenes de una escena en la que se puedan detectar fácilmente los bordes, tanto en la imagen de profundidad como la infrarroja, y calculando la media del desplazamiento entre píxeles correspondientes de ambas imágenes. En la Figura 2.17b se puede observar este desplazamiento comparando el lado derecho y superior, en la que la profundidad sobresale de la forma descrita por la imagen infrarroja, con el inferior y el izquierdo.



(a) Imagen de color



(b) Superposición de la imagen de profundidad a la infrarroja

Figura 2.17: Desplazamiento entre las imágenes infrarrojas y de profundidad

2.2.2 *Tiempo de vuelo (ToF)*

Las cámaras de tiempo de vuelo se basan en el principio de interferometría de modulación, buscando la correlación entre la señal óptica recibida

Las cámaras de tiempo de vuelo se basan en el principio de interferometría de modulación. Un emisor acoplado a la cámara emite luz incoherente cercana al infrarrojo (*near-infrared, NIR*), Ecuación 2.20, que es sinusoidal modulada con una frecuencia ω.

$$g(t) = \cos(\omega t) \tag{2.20}$$

Esta luz ilumina la escena 3D provocando que se refleje y remita una señal difuminada $s(t)$, Ecuación 2.21, con desfase ϕ respecto a la señal $g(t)$ emitida.

$$s(t) = k + a\cos(\omega t + \phi) \tag{2.21}$$

Además, la señal remitida tiene una amplitud a y una constante k procedente de la iluminación del entorno que modifica la modulación. El desfase ϕ entre las señales $g(t)$ y $s(t)$ puede ser estimado a partir de la correlación $c(\tau)$ entre la señal emitida y remitida, Ecuación 2.22.

$$c(\tau) = (s * g)(\tau) = h + \frac{a}{2}\cos(\omega\tau + \phi) \tag{2.22}$$

Esta correlación se calcula para cuatro diferencias de fase $\tau_p = p\frac{\pi}{2}, p = 0, 1, 2, 3$. Dada la velocidad de la luz c y la frecuencia de modulación ω, se generan cuatro correlaciones $R_p = c(\tau_p)$ para calcular la diferencia de fase ϕ Ecuación 2.23, y la amplitud a, Ecuación 2.24.

$$\phi = \arctan\left(\frac{R_1 - R_3}{R_0 - R_2}\right) \tag{2.23}$$

$$a = \frac{\sqrt{(R_1 - R_3)^2 + (R_0 - R_2)^2}}{2} \tag{2.24}$$

Figura 2.18: Microsoft Kinect v2

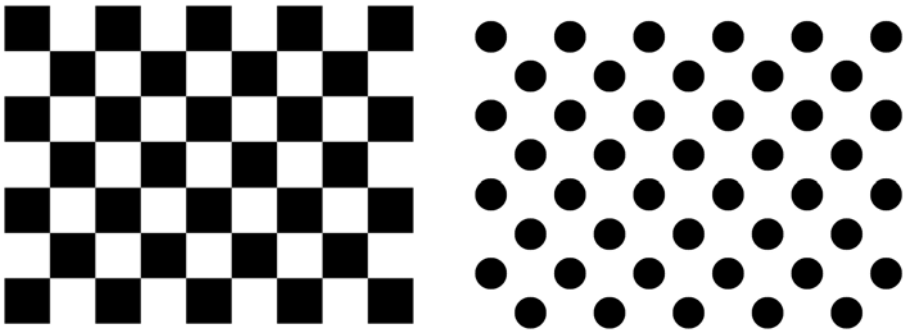
Finalmente, se obtiene la distancia de un determinado píxel, Ecuación 2.25, y formar un mapa de profundidad.

$$d = \frac{c\phi}{4\pi\omega} \tag{2.25}$$

Dentro de este tipo de sensores, el más conocido es Kinect v2 de Microsoft, Figura 2.18. Compuesto por una cámara RGB, otra infrarroja y un emisor infrarrojo, proporciona información de color y de profundidad de la escena. La diferencia principal con la versión anterior de Kinect, es el uso de la tecnología de tiempo de vuelo para determinar las distancias.

2.3 MÉTODOS DE CALIBRADO DE SENSORES RGB-D

Actualmente se pueden encontrar varios algoritmos de calibrado para los sensores RGB-D en la comunidad científica, algunos de ellos específicos para la tecnología que utiliza el sensor, y otros menos específicos. También se pueden encontrar algoritmos generales para el calibrado de cámaras, que no se centran en ninguna tecnología ni característica γ_i de la



(a) Patrón de tablero de ajedrez (b) Patrón de circulos

Figura 2.19: Patrones de calibrado

cámara.

La mayoría de estos algoritmos utilizan un patrón en forma de tablero de ajedrez de tamaño conocido, Figura 2.19a denominado *chessboard* o *checkerboard* en inglés. Aunque también se pueden encontrar otros tipos de patrones basados en círculos, Figura 2.19b, o marcadores. Los patrones como forma de tablero de ajedrez tienen la ventaja de permitir estimar las esquinas de los cuadrados utilizando técnicas de estimación de bordes. Por otro lado, los basados en círculos permiten obtener el centro de cada círculo de forma robusta a cambios en la orientación, mediante técnicas de centro de masas. El objetivo es tomar varias capturas de ese patrón en diferentes orientaciones y posiciones de la imagen para determinar los parámetros de la cámara conociendo las características del patrón y su tamaño (Figura 2.20).

Según los parámetros, se pueden diferenciar dos tipos de calibrado: intrínseco y extrínseco. El primero es aquel calibrado que obtiene los parámetros internos del sensor, como distancia focal, punto principal, coeficientes de distorsión, etc. Mientras que el segundo obtiene características externas, como por ejemplo las transformaciones necesarias para alinear dos imágenes.

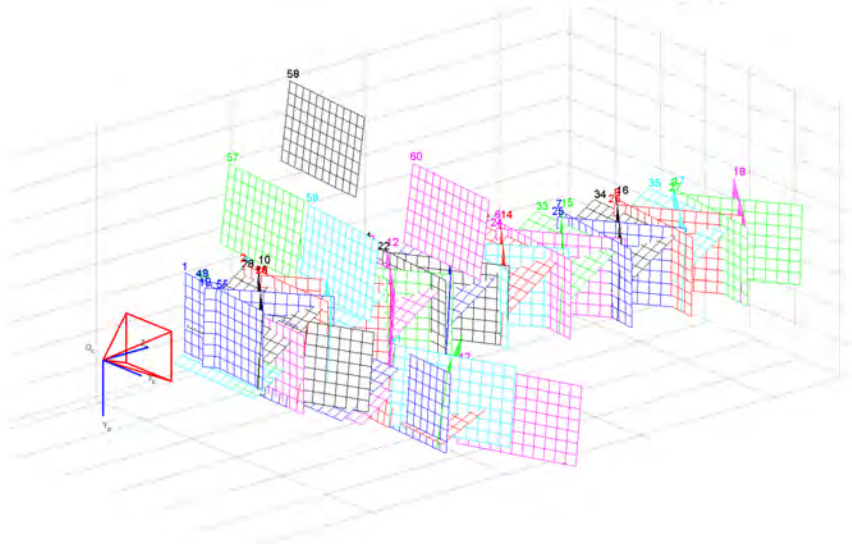
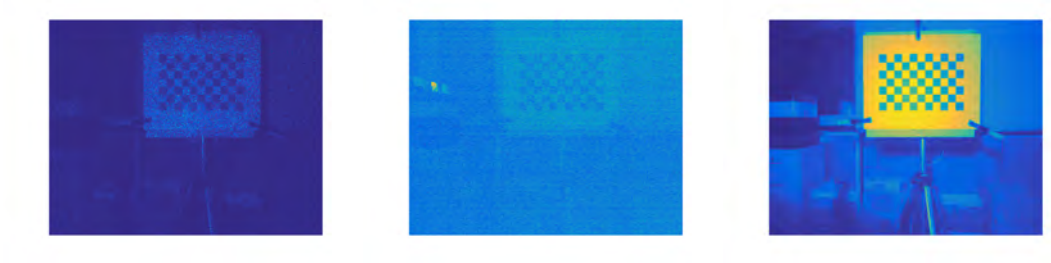


Figura 2.20: Diferentes orientaciones del patrón respecto a la cámara

2.3.1 *Método de Bouguet*

Camera calibration toolbox for Matlab [Bouguet, 2004] es una herramienta genérica para el calibrado de cámaras que incluye un apartado para el calibrado de un sistema estereoscópico. Esto permite calibrar las cámaras RGB y de profundidad de un sensor RGB-D como si se tratase de un sistema estéreo, pero en lugar utilizar las transformaciones entre ambas cámaras de color para alinear las imágenes 2D, se utilizan para obtener las correspondencias entre el mapa de profundidad y la imagen de color. En [Smisek et al., 2011, Van Den Bergh and Van Gool, 2011] se utiliza esta herramienta para calibrar el sensor Microsoft Kinect, y una cámara de tiempo de vuelo acoplada a otra de color, respectivamente. Por lo tanto se han utilizado las imágenes de color e infrarrojas para realizar el proceso de calibrado.

Para detectar las esquinas del patrón en la imagen de profundidad, se utiliza la imagen infrarroja, ya que este es el sensor donde se orienta la imagen de profundidad. Como la cámara emite el patrón de motas, la imagen de infrarrojos es muy ruidosa (Figura 2.21a). Con el objetivo de obtener una imagen donde se pudiese detectar el patrón más fácilmente,



(a) Imagen infrarroja del patrón (b) Imagen infrarroja del patrón sin emisor infrarrojo (c) Imagen infrarroja del patrón sin emisor infrarrojo añadiendo un foco

Figura 2.21: Imágenes infrarrojas

se ha bloqueado el emisor infrarrojo del dispositivo, sin embargo, no hay suficiente iluminación para obtener una imagen en la que se pueda observar el patrón con claridad, Figura 2.21b. Por ello se ha utilizado un foco para mejorar su visibilidad, Figura 2.21c, y facilitar el proceso de calibrado.

Es importante destacar que el conjunto de imágenes para ambas cámaras debe ser obtenido simultáneamente, de lo contrario las correspondencias entre ambas cámaras obtenidas por el algoritmo serían incorrectas.

El primer paso del proceso consiste en obtener el calibrado de cada cámara de forma individual, seleccionando las esquinas del patrón en cada una de las imágenes. El número de esquinas y el orden en el que se seleccionan debe ser el mismo para ambas, de lo contrario no se podrán encontrar correspondencias entre ambas cámaras.

Posteriormente se utilizan los calibrados individuales de ambas cámaras para realizar el calibrado estereoscópico. Este proceso realiza un ajuste para refinar los parámetros intrínsecos de ambas cámaras y obtiene las transformaciones entre ellas a partir de las correspondencias de las esquinas del patrón entre ambas cámaras.

2.3.2 *Métdodo de Burrus*

RGBDemo [Burrus, 2012] proporciona un conjunto de herramientas y librerías para trabajar con los datos de Microsoft Kinect (aunque se puede usar con sensores que soporten el mismo *driver*) y desarrollar aplicaciones de visión por computador. Entre las herramientas que contiene, incluye una para el calibrado de este tipo de sensores. Ha sido desarrollado utilizando OpenCV y es *opensource*, lo que facilita el acceso a su código.

El proceso de calibración se realiza como si se tratase de un sistema estéreo, como en [Bouguet, 2004]. Los intrínsecos de la cámara RGB se obtienen a partir de la localización de las esquinas del patrón de calibrado en la imagen de color, y los de la cámara infrarroja se detectan automáticamente a partir de las imagen infrarroja y de disparidad. A partir de la información del patrón en ambas cámaras, se realiza una calibración estéreo común para obtener los parámetros extrínsecos que permiten alinear las imágenes de ambas cámaras.

2.3.3 *Método de Herrera*

Es un método propuesto por [Daniel Herrera C et al., 2012] para calibrar, simultáneamente, dos cámaras de color, una cámara de profundidad y la posición relativa entre ellas. El algoritmo ha sido diseñado con el objetivo de ser preciso, práctico, y aplicable a varios sensores. El algoritmo está preparado para trabajar con Microsoft Kinect, ya que el modelo intrínseco que contiene para la cámara de profundidad es para este sensor. Sin embargo se puede reemplazar este modelo y utilizar el algoritmo para otra cámara con modelo distinto (como la Kinect V2, que utiliza tiempo de vuelo).

El proceso de calibrado espera como entradas alrededor de 30 imágenes de color y de disparidad con la información de profundidad, de un *chessboard* de dimensiones conocidas sobre una superficie plana de mayor tamaño que el patrón. A continuación es necesario

extraer la localización de las esquinas en las imágenes de color y el plano que contiene el patrón. Para inicializar los parámetros intrínsecos de la cámara de profundidad es necesario seleccionar las esquinas en las imágenes de profundidad. Este último paso se debe realizar de forma arbitraria, puesto que el patrón no es distinguible en las imágenes de profundidad. Después de estos pasos, se realiza un calibrado inicial individual para cada imagen, basado en la inicialización previa. Finalmente se realiza un ajusto iterativo no lineal para obtener el resultado final del calibrado.

Este algoritmo incluye un nuevo modelo para corregir la distorsión de la cámara de profundidad en la disparidad, basado en un error constante que muestran en las medidas este tipo de sensores RGB-D. Además, este error decrece a medida que se incrementa la distancia con el sensor. Para aplicar esta corrección, entre los resultados que aporta el algoritmo se encuentra la matriz con el patrón de distorsión espacial mostrado en la Figura 2.22, que es del mismo tamaño que la imágenes de profundidad, y dos valores α_0, α_1 que representan la decadencia del efecto de distorsión con la distancia. Esta corrección se puede aplicar utilizando la Ecuación 2.26, donde d es la disparidad devuelta por el sensor en el píxel (u, v), D_σ contiene el valor del patrón de distorsión espacial para el pixel (u, v), d_k es la disparidad corregida y α_o, α_1 son los valores que modelan la decadencia de la distorsión.

$$d_k = d + D_\sigma(u, v) * exp(\alpha_0 - \alpha_1 * d) \tag{2.26}$$

Una vez corregida la disparidad se puede obtener la distancia a un punto con la Ecuación 2.27, donde Z_d la distancia entre el punto y la cámara, d_k la disparidad corregida y c_0, c_1 los parámetros que permiten convertir la disparidad a metros. Estos últimos parámetros están relacionados con la Ecuación 2.19, siendo c_0, c_1 las igualdades representadas en las Ecuaciones 2.28 y 2.29, respectivamente.

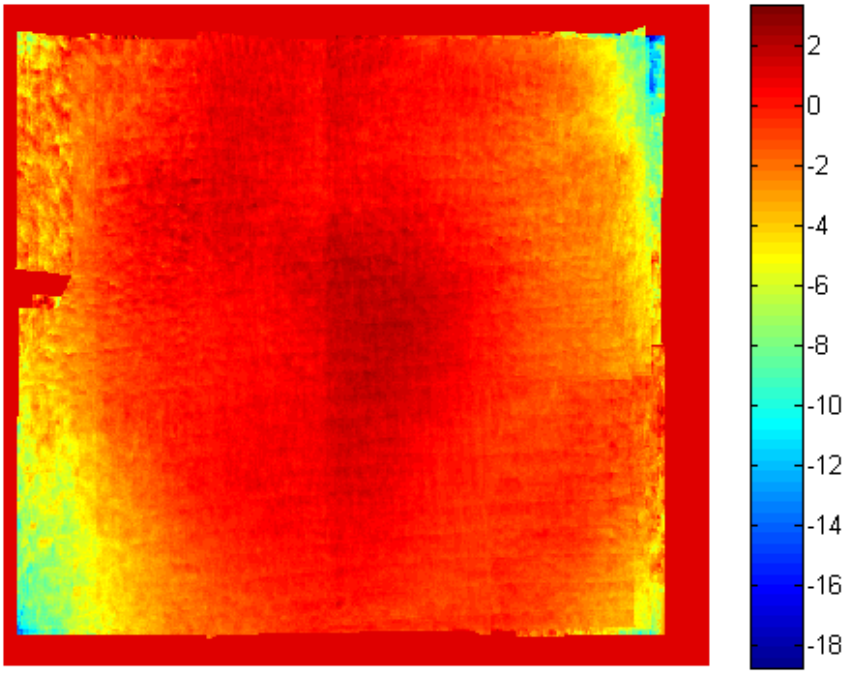


Figura 2.22: Patrón de distorsión espacial

$$Z_d = \frac{1}{c_1 * d_k + c_0} \tag{2.27}$$

$$c_0 = Z_o^-1 + \frac{n}{fb} \tag{2.28}$$

$$c_1 = \frac{m}{fb} \tag{2.29}$$

3

EXPERIMENTACIÓN

En este capitulo se van a explicar los resultados de la experimentación para el análisis de los distintos algoritmos de calibrado y cámaras. En primer lugar se muestran los resultados obtenidos para cada cámara y a continuación, se muestran los resultados de varias pruebas para ver la corrección de los datos debido al calibrado de los distintos algoritmos.

Los algoritmos de Bouguet [Bouguet, 2004], Herrera [Daniel Herrera C et al., 2012] y Burrus [Burrus, 2012], mencionados anteriormente, se han aplicado a tres tipos de sensores: Microsoft Kinect (refiriéndose a la versión 1 del dispositivo), PrimeSense Carmine 1.09 y Microsoft Kinect v2. A excepción del algoritmo de Herrera que solo se ha aplicado a los dos primeros, debido a que el modelo de error que implementa es para la tecnología de luz estructurada y no tiempo de vuelo.

Para obtener los diferentes calibrados, se han realizado varias capturas de un patrón de calibrado del tipo de tablero de ajedrez en diferentes posiciones y orientaciones de la cámara. El patrón de calibrado tiene 7×11 cuadrados de $0{,}034\text{m}$ de lado. A continuación se ha seleccionado un subconjunto de 60 imágenes para cada cámara, el cual se ha utilizado en los distintos algoritmos.

Con el objetivo de facilitar este proceso, se han desarrollado unas herramientas que facilitan la captura de imágenes, la conversión a distintos formatos y la aplicación de los resultados del calibrado a las imágenes, que se mostrarán en la Sección 3.1.

3.1 HERRAMIENTAS DESARROLLADAS

Las herramientas que se muestran a continuación se han desarrollado con el objetivo de facilitar la adquisición de imágenes y el proceso de calibrado.

Para la adquisición han sido necesarios dos programas, uno para obtener las imágenes utilizando el *driver* OpenNI para los sensores Microsoft Kinect y PrimeSense Carmine 1.09, y otro utilizando el SDK de *Kinect for Windows* [1] para la cámara Microsoft Kinect v2. OpenNi es una herramienta desarrollada por PrimeSense que permite a sus dispositivos (para Microsoft Kinect se requiere software de terceros adicional) ser utilizados en computadores comunes.

El primero (Figura 3.1) se ha desarrollado en Matlab utilizando *Kinect for Matlab* [2], que proporciona un conjunto de funciones MEX (Matlab Executable) para utilizar las funciones de la API de OpenNI. La interfaz permite adquirir, previsualizar y guardar las imágenes de color, profundidad, disparidad e infrarroja, además de la nube de puntos.

La segunda herramienta de adquisición para Kinect v2 (Figura 3.2) utiliza el SDK publicado por microsoft, por lo que se ha desarrollado utilizando C# y también permite almacenar las imágenes de color, profundidad, infrarroja y la nube de puntos.

Aunque hay algoritmos de calibrado que aceptan los datos en varios formatos, hay otros que los necesitan con un formato concreto. Por ello, otra herramienta (Figura 3.3) facilita la lectura de las imágenes adquiridas por los programas anteriores y la exportación a diferentes formatos de forma masiva.

1 *https://www.microsoft.com/en-us/kinectforwindows/develop/default.aspx* última visita: 24 de Junio, 2015
2 *http://sourceforge.net/projects/kinect-mex/* última visita: 24 de Junio, 2015

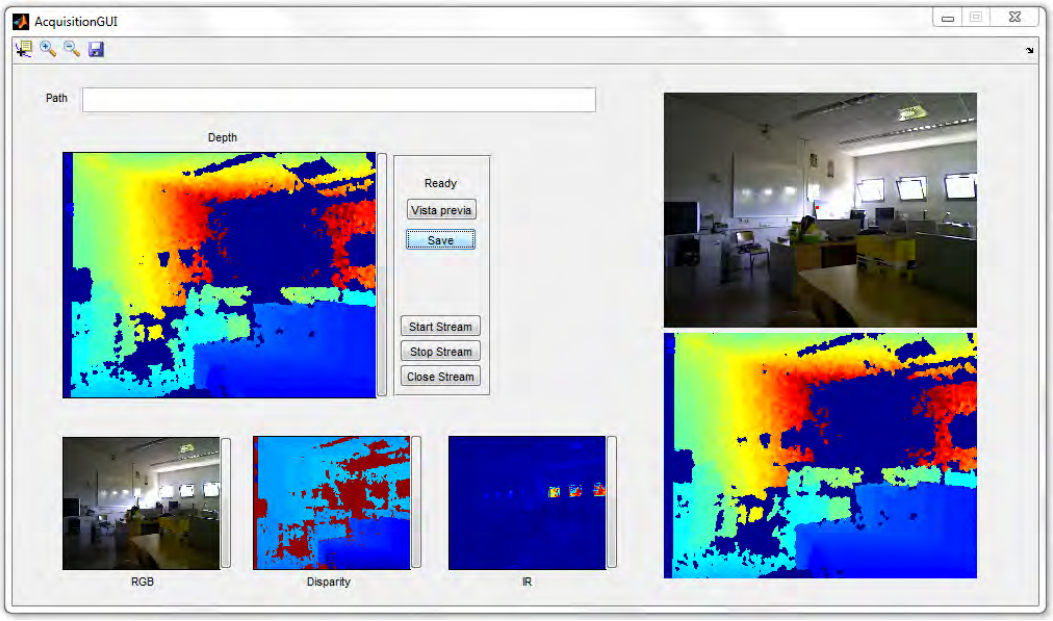


Figura 3.1: Herramienta de adquisición para OpenNI en Matlab

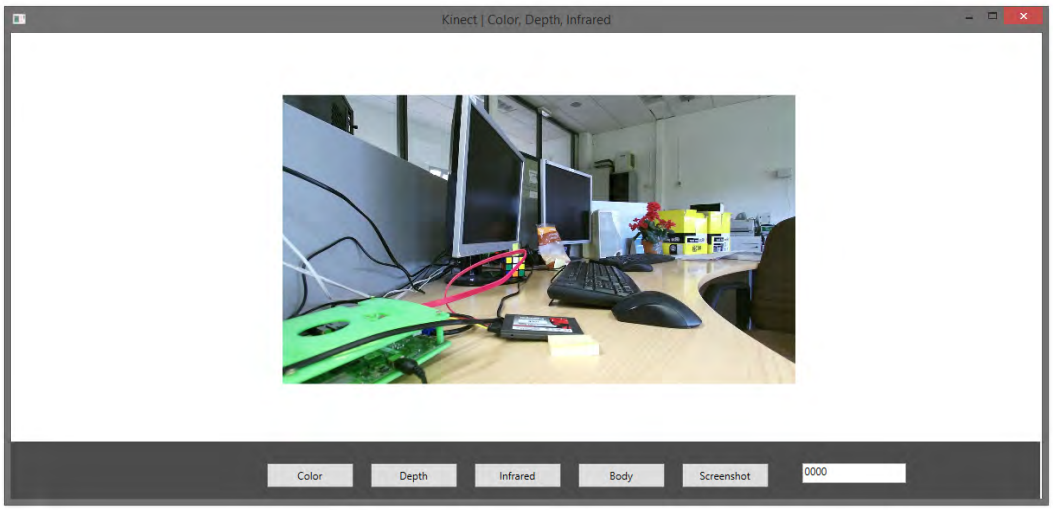


Figura 3.2: Herramienta de adquisición para Microsoft Kinect V2

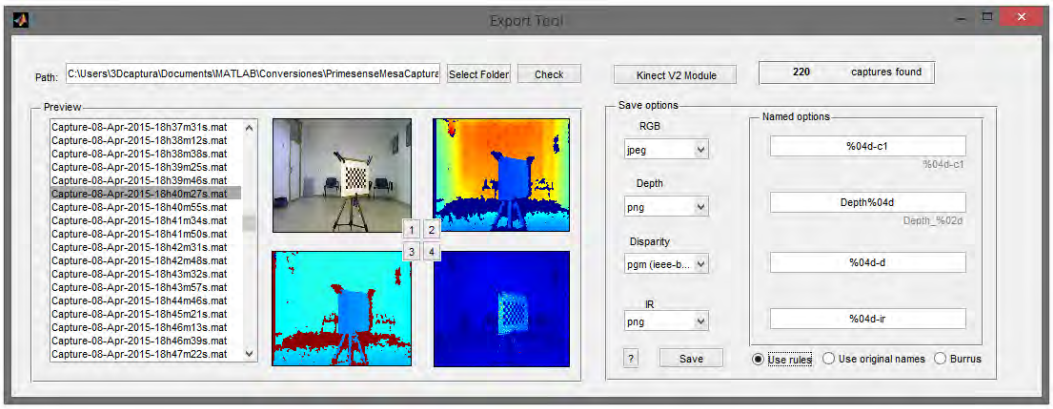


Figura 3.3: Herramienta para exportar imágenes adquiridas en Matlab

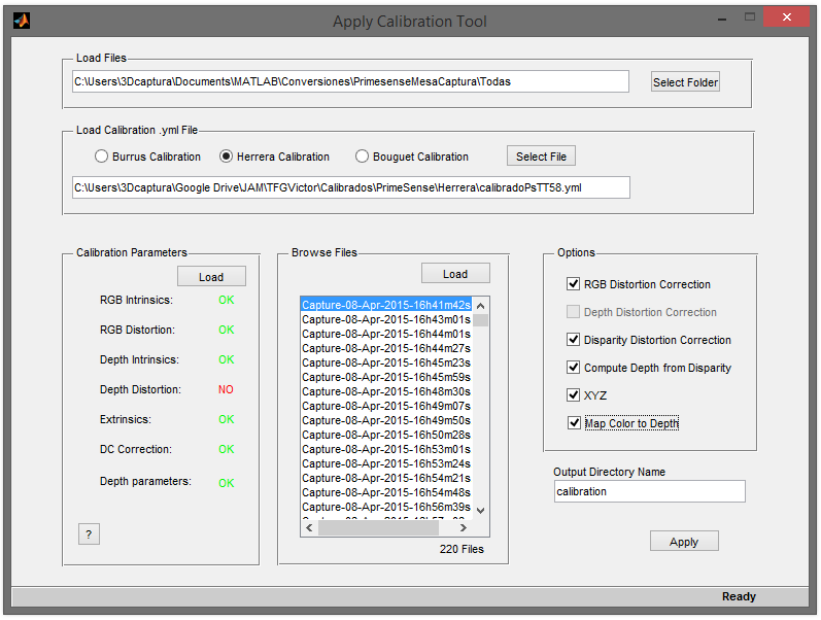


Figura 3.4: Herramienta para aplicar las correcciones del calibrado en Matlab

Los parámetros aportados por estos algoritmos se pueden almacenar en formato YML, que es un derivado de XML, para facilitar la modificación de futuras adquisiciones utilizando estos parámetros, otro programa (Figura 3.4) facilita esta tarea, permitiendo seleccionar las correcciones a aplicar, dado un archivo yml, a un conjunto de imágenes adquiridas previamente.

3.2 RESULTADOS DE CALIBRACIÓN

En esta sección se van a exponer los valores intrínsecos y extrínsecos para cada una de las cámaras, siendo la Microsoft Kinect, la Primesense Carmine y la Microsoft Kinect 2. Para cada una de las cámaras, se muestra el resultado de los métodos de calibración aplicados.

3.2.1 *Microsoft Kinect*

En este apartado se muestran los resultados obtenidos para el sensor Microsoft Kinect en las Tablas 3.1, 3.2 y 3.3, y en la Figura 3.5.

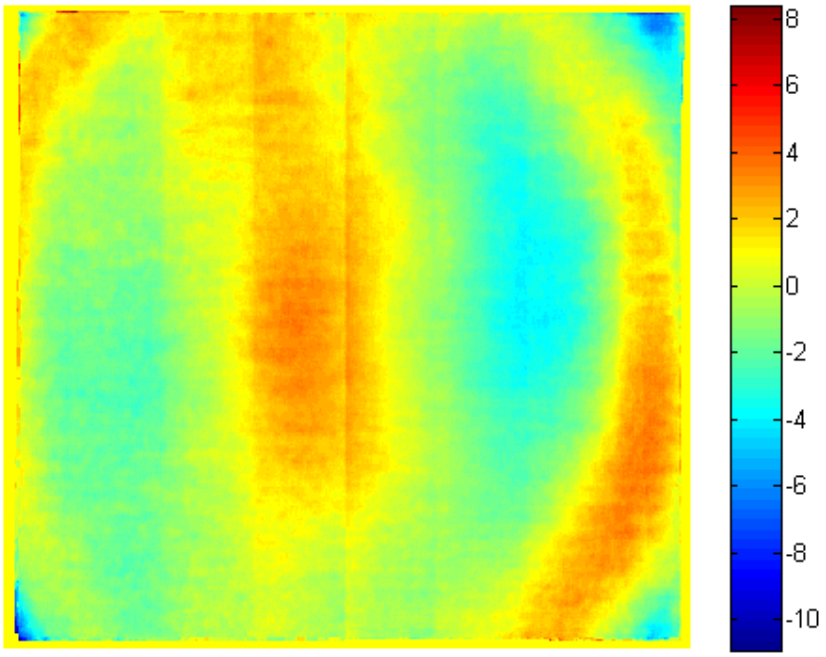


Figura 3.5: Patrón de distorsión espacial del algoritmo de Herrera para Microsoft Kinect

Parámetro		Cámara RGB	Cámara IR
Distancia focal	f_x	523,24	595,99
	f_y	521,68	592,44
Punto principal	c_x	328,65	314,43
	c_y	257,03	227,05
Distorsión radial	k_1	0,0215	−0,1567
	k_2	−0,6927	0,6467
	k_3	0,7170	−0,8859
Distorsión tangencial	p_1	−0,0007	0,0012
	p_2	−0,0005	0,0004
Transformaciones	R	$\begin{bmatrix} 0{,}9995 & 0{,}0082 & -0{,}0052 \\ -0{,}0081 & 0{,}9988 & 0{,}0125 \\ 0{,}0053 & -0{,}0125 & 0{,}9999 \end{bmatrix}$	
	T	$\begin{bmatrix} -0{,}0255 \\ 0{,}0026 \\ 0{,}0068 \end{bmatrix}$	

Tabla 3.1: Resultados del algoritmo de Burrus para Microsoft Kinect

Parámetro		Cámara RGB	Cámara IR
Distancia focal	f_x	$523{,}16 \pm 1{,}40$	$588{,}18 \pm 1{,}58$
	f_y	$521{,}32 \pm 1{,}35$	$586{,}00 \pm 1{,}52$
Punto principal	c_x	$330{,}14 \pm 1{,}04$	$315{,}83 \pm 1{,}21$
	c_y	$257{,}01 \pm 1{,}14$	$245{,}20 \pm 1{,}25$
Distorsión radial	k_1	$0{,}1475 \pm 0{,}00609$	$-0{,}0724 \pm 0{,}0052$
	k_2	$-0{,}2735 \pm 0{,}0116$	$0{,}1306 \pm 0{,}01$
	k_3	0	0
Distorsión tangencial	p_1	$-0{,}0014 \pm 0{,}00082$	$0{,}0009 \pm 0{,}0007$
	p_2	0	$-0{,}0013 \pm 0{,}00071$
Transformaciones	R	$\begin{bmatrix} -0{,}0076 \\ -0{,}0031 \\ 0{,}0078 \end{bmatrix} \pm \begin{bmatrix} 0{,}0012 \\ 0{,}0016 \\ 0{,}0004 \end{bmatrix}$	
	T	$\begin{bmatrix} 0{,}0250 \\ 0{,}0004 \\ -0{,}0003 \end{bmatrix} \pm \begin{bmatrix} 0{,}0001 \\ 0{,}0001 \\ 0{,}0004] \end{bmatrix}$	

Tabla 3.2: Resultados del algoritmo de Bouguet para Microsoft Kinect

Parámetro		Cámara RGB	Cámara IR
Distancia focal	f_x	$522{,}55 \pm 0{,}25$	$586{,}80 \pm 0{,}45$
	f_y	$520{,}24 \pm 0{,}25$	$577{,}70 \pm 0{,}59$
Punto principal	c_x	$329{,}76 \pm 0{,}33$	$318{,}92 \pm 0{,}35$
	c_y	$257{,}59 \pm 0{,}37$	$231{,}46 \pm 0{,}37$
Distorsión radial	k_1	$0{,}1930 \pm 0{,}0024$	0
	k_2	$-0{,}5651 \pm 0{,}012$	0
	k_3	$0{,}4843 \pm 0{,}0176$	0
Distorsión tangencial	p_1	$-0{,}0006 \pm -0{,}0004$	0
	p_2	$-0{,}0003 \pm 0{,}0002$	0
Relación disparidad-profundidad	c_0	$-$	$3{,}0946 \pm 0{,}0035$
	c_1	$-$	$-0{,}0028 \pm 3{,}7100e-06$
Distorsión disparidad	α_0	$-$	$1{,}2521 \pm 0{,}0510$
	α_1	$-$	$0{,}0022 \pm 7{,}4073e-05$
Transformaciones	R	$\begin{bmatrix} 1 & -0{,}0077 & -0{,}0047 \\ 0{,}0077 & 0{,}9999 & -0{,}0084 \\ 0{,}0048 & 0{,}0084 & 1 \end{bmatrix} \pm \begin{bmatrix} 6{,}9160e-04 \\ 5{,}9122e-04 \\ 3{,}4263e-04 \end{bmatrix}$	
	T	$\begin{bmatrix} 0{,}0269 \\ -0{,}0026 \\ -0{,}0024 \end{bmatrix} \pm \begin{bmatrix} 3{,}9870e-04 \\ 4{,}5291e-04 \\ 6{,}1674e-04] \end{bmatrix}$	

Tabla 3.3: Resultados del algoritmo de Herrera para Kinect

3.2.2 *PrimeSense Carmine 1.09*

A continuación se muestran los resultados para el sensor Carmine 1.09 en las Tablas 3.4, 3.5 y 3.6, y en la Figura 3.6.

Parámetro		Cámara RGB	Cámara IR
Distancia focal	f_x	540,84	580,04
	f_y	539,48	576,45
Punto principal	c_x	318,38	307
	c_y	237,82	232,75
Distorsión radial	k_1	0,0512	−0,0687
	k_2	−0,2236	0,2196
	k_3	0,1785	−0,4167
Distorsión tangencial	p_1	0,0010	−0,0007
	p_2	−0,0009	−0,004
Transformaciones	R	$\begin{bmatrix} 0{,}9999 & 0{,}0049 & 0{,}0089 \\ -0{,}005 & 9{,}9992 & -0{,}005 \\ -0{,}0089 & -0{,}0112 & 0{,}9989 \end{bmatrix}$	
	T	$\begin{bmatrix} -0{,}0257 \\ 0{,}0005 \\ 0{,}0037 \end{bmatrix}$	

Tabla 3.4: Resultados del algoritmo de Burrus para PrimeSense Carmine 1.09

Parámetro		Cámara RGB	Cámara IR
Distancia focal	f_x	$540{,}58 \pm 0{,}64$	$575{,}46 \pm 0{,}68$
	f_y	$538{,}95 \pm 0{,}62$	$573{,}98 \pm 0{,}65$
Punto principal	c_x	$318{,}62 \pm 0{,}99$	$318{,}79 \pm 1{,}05$
	c_y	$238{,}32 \pm 0{,}86$	$245{,}13 \pm 0{,}90$
Distorsión radial	k_1	$0{,}0232 \pm 0{,}0023$	$-0{,}0401 \pm 0{,}0029$
	k_2	$-0{,}0939 \pm 0{,}0059$	$0{,}0304 \pm 0{,}0061$
	k_3	0	0
Distorsión tangencial	p_1	$0{,}0012 \pm 0{,}00045$	$0{,}00011 \pm 0{,}00044$
	p_2	$-0{,}00064 \pm 0{,}00055$	$-0{,}00014 \pm 0{,}00054$
Transformaciones	R	$\begin{bmatrix} -0{,}00214 \\ 0{,}00201 \\ 0{,}00429 \end{bmatrix} \pm \begin{bmatrix} 0{,}00089 \\ 0{,}00121 \\ 0{,}0001 \end{bmatrix}$	
	T	$\begin{bmatrix} 0{,}0262 \\ 0{,}0001 \\ -0{,}0002 \end{bmatrix} \pm \begin{bmatrix} 0{,}00005 \\ 0{,}00005 \\ 0{,}00021] \end{bmatrix}$	

Tabla 3.5: Resultados del algoritmo de Bouguet para PrimeSense Carmine 1.09

Parámetro		Cámara RGB	Cámara IR
Distancia focal	f_x	$541{,}67 \pm 0{,}16$	$574{,}98 \pm 0{,}23$
	f_y	$539{,}48 \pm 0{,}16$	$570{,}58 \pm 0{,}31$
Punto principal	c_x	$316{,}87 \pm 0{,}27$	$323{,}97 \pm 0{,}23$
	c_y	$235{,}48 \pm 0{,}24$	$227{,}71 \pm 0{,}2$
Distorsión radial	k_1	$0{,}0578 \pm 0{,}0015$	0
	k_2	$-0{,}2610 \pm 0{,}0069$	0
	k_3	$0{,}2430 \pm 0{,}0098$	0
Distorsión tangencial	p_1	$0{,}0003 \pm 0{,}0001$	0
	p_2	$-0{,}0017 \pm 0{,}0001$	0
Relación disparidad-profundidad	c_0	–	$4{,}0054 \pm 0{,}0021$
	c_1	–	$-0{,}0029 \pm 1{,}68e-06$
Distorsión disparidad	α_0	–	$1{,}6229 \pm 0{,}0304$
	α_1	–	$0{,}0021 \pm 4{,}06e-05$
Transformaciones	R	$\begin{bmatrix} 1 & -0{,}0040 & 0{,}0086 \\ 0{,}0042 & 0{,}9998 & -0{,}0169 \\ -0{,}0086 & 0{,}0169 & 0{,}9998 \end{bmatrix} \pm \begin{bmatrix} 4{,}3043e-04 \\ 4{,}6892e-04 \\ 2{,}1396e-04 \end{bmatrix}$	
	T	$\begin{bmatrix} 0{,}0265 \\ -0{,}0007 \\ -0{,}0030 \end{bmatrix} \pm \begin{bmatrix} 1{,}7632e-04 \\ 1{,}3328e-04 \\ 2{,}0493e-04] \end{bmatrix}$	

Tabla 3.6: Resultados del algoritmo de Herrera para PrimeSense Carmine 1.09

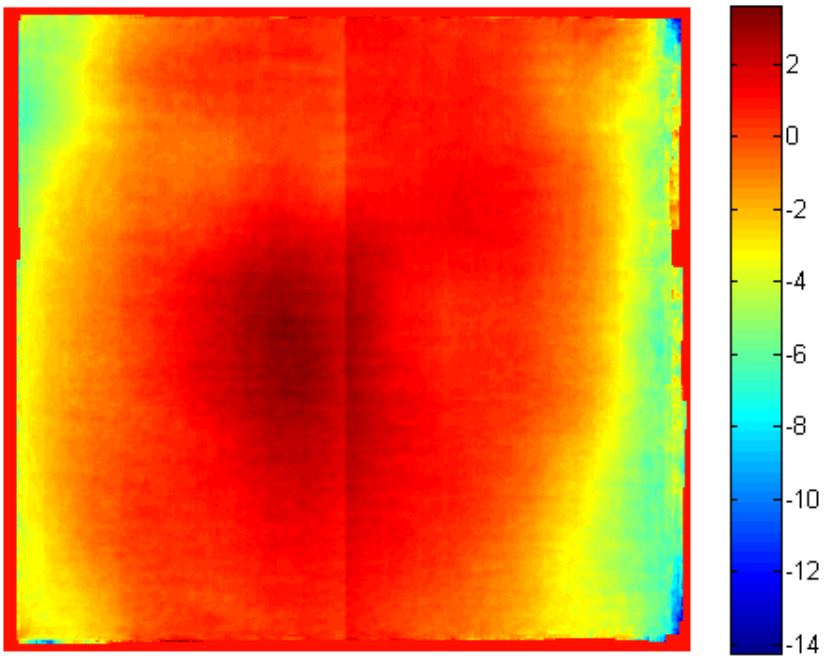


Figura 3.6: Patrón de distorsión espacial del algoritmo de Herrera para PrimeSense Carmine 1.09

3.2.3 *Microsoft Kinect v2*

Los resultados obtenidos para el sensor Microsoft Kinect v2 se muestran en las Tablas 3.7 y 3.8.

Parámetro		Cámara RGB	Cámara IR
Distancia focal	f_x	1669,54	364,92
	f_y	1588,27	364,29
Punto principal	c_x	345,85	256,67
	c_y	251,59	205,33
Distorsión radial	k_1	−0,0343	0,0934
	k_2	0,0697	−0,2748
	k_3	−0,0257	0,0963
Distorsión tangencial	p_1	−0,0209	−0,0004
	p_2	−0,0518	0,00004
Transformaciones	R	$\begin{bmatrix} 0{,}9261 & -0{,}0471 & -0{,}3740 \\ -0{,}0258 & 0{,}9818 & -0{,}1879 \\ 0{,}3761 & 0{,}1837 & 0{,}9081 \end{bmatrix}$	
	T	$\begin{bmatrix} -0{,}0468 \\ 0{,}0080 \\ -0{,}3432 \end{bmatrix}$	

Tabla 3.7: Resultados del algoritmo de Burrus para Microsoft Kinect v2

Los resultados obtenidos para el sensor Microsoft Kinect v2 con el algoritmo de Burrus (Tabla 3.7) se han descartado para pruebas posteriores dado el error que presenta para el punto principal de la cámara RGB ($c_x = 345{,}85$ $c_y = 251{,}59$), siendo el tamaño de la imagen

Parámetro		Cámara RGB	Cámara IR
Distancia focal	f_x	$1057{,}58 \pm 1{,}83$	$369{,}15 \pm 0{,}66$
	f_y	$1055{,}33 \pm 1{,}77$	$368{,}01 \pm 0{,}64$
Punto principal	c_x	$971{,}26 \pm 1{,}71$	$260{,}68 \pm 0{,}53$
	c_y	$538{,}00 \pm 1{,}68$	$205{,}92 \pm 0{,}60$
Distorsión radial	k_1	$0{,}0413 \pm 0{,}0023$	$0{,}0631 \pm 0{,}0034$
	k_2	$-0{,}0389 \pm 0{,}0019$	$-0{,}1758 \pm 0{,}0044$
	k_3	0	0
Distorsión tangencial	p_1	$-0{,}00107 \pm 0{,}00044$	$-0{,}00096 \pm 0{,}00037$
	p_2	$0{,}00001 \pm 0{,}00049$	$-0{,}00062 \pm 0{,}00032$
Transformaciones	R	$\begin{bmatrix} 0{,}00156 \\ 0{,}00402 \\ -0{,}00691 \end{bmatrix} \pm \begin{bmatrix} 0{,}00087 \\ 0{,}00112 \\ 0{,}00013 \end{bmatrix}$	
	T	$\begin{bmatrix} 0{,}05211 \\ -0{,}00061 \\ -0{,}00319 \end{bmatrix} \pm \begin{bmatrix} 0{,}00011 \\ 0{,}00011 \\ 0{,}00040] \end{bmatrix}$	

Tabla 3.8: Resultados del algoritmo de Bouguet para Microsoft Kinect v2

de color 1920×1980. Además, dista bastante del valor obtenido con el método de Bouguet (Tabla 3.8), ($c_x = 971{,}26$ $c_y = 538$).

3.2.4 *Modelos de distorsión*

La distorsión pude provocar que los píxeles de la imagen de profundidad no se sitúen en su posición correcta. Esto influye de manera notable en el calculo de la nube de puntos, provocando efectos como el que se muestra en la Figura 3.7, en la que las esquinas del plano sobresalen del resto de puntos.

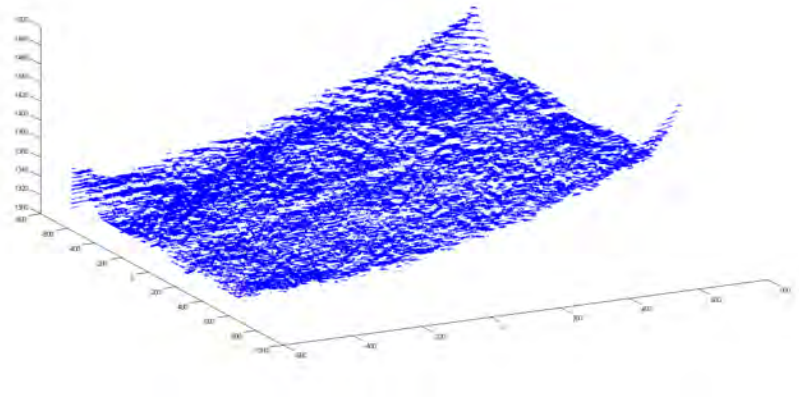
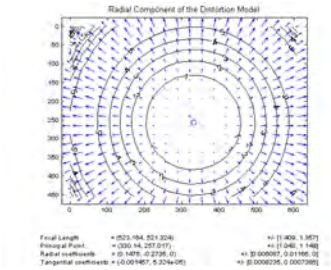
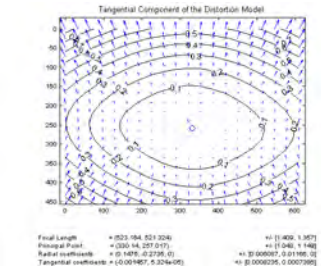


Figura 3.7: Deformación de un plano debido a la distorsión de la lente

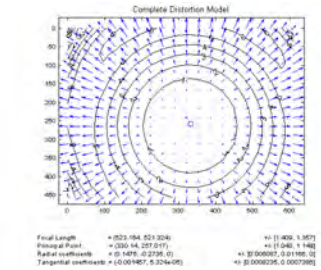
Estos defectos que se producen en la imagen se representan mediante modelos de distorsión, que indican de manera gráfica el desplazamiento que sufren los píxeles según su posición en la imagen. Estos modelos se pueden obtener a partir de la distancia focal, el punto principal y los coeficientes de distorsión. A continuación se muestran los modelos de distorsión obtenidos con cada uno de los métodos de calibración probados para cada cámara.



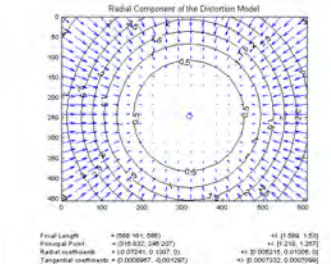
(a) Distorsión radial (Cámara RGB)



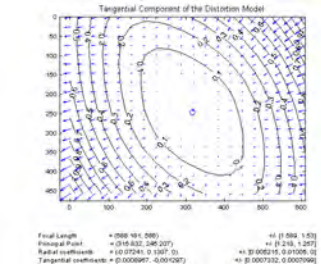
(b) Distorsión tangencial (Cámara RGB)



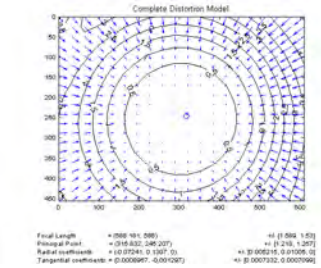
(c) Modelo completo de distorsión (Cámara RGB)



(d) Distorsión radial (Cámara Ir)

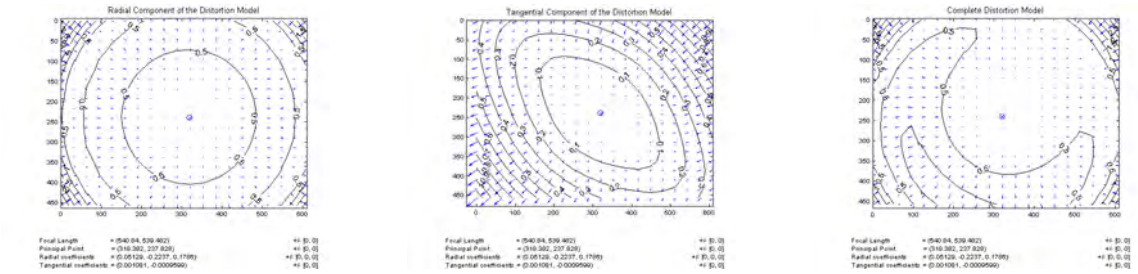


(e) Distorsión tangencial (Cámara Ir)



(f) Modelo completo de distorsión (Cámara Ir)

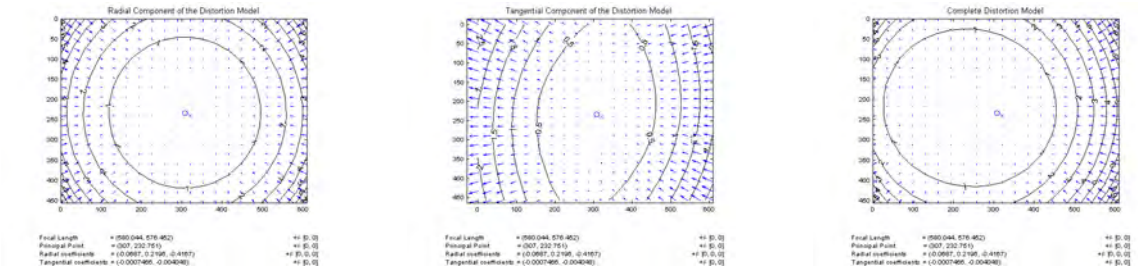
Figura 3.8: Modelos de distorsión del calibrado de Bouguet para Microsoft Kinect



(a) Distorsión radial (Cámara RGB)

(b) Distorsión tangencial (Cámara RGB)

(c) Modelo completo de distorsión (Cámara RGB)

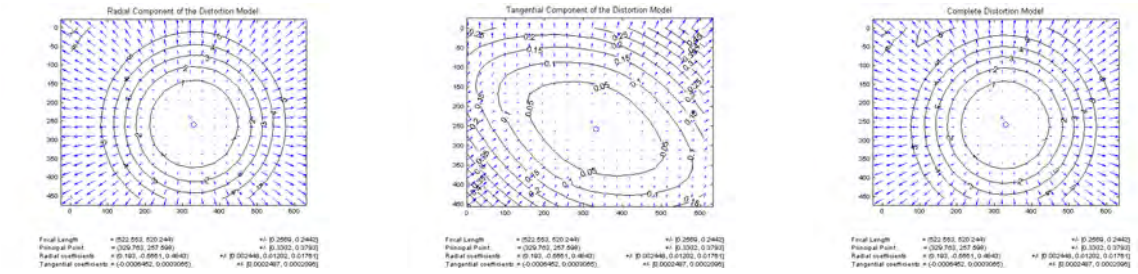


(d) Distorsión radial (Cámara Ir)

(e) Distorsión tangencial (Cámara Ir)

(f) Modelo completo de distorsión (Cámara Ir)

Figura 3.9: Modelos de distorsión del calibrado de Burrus para Microsoft Kinect

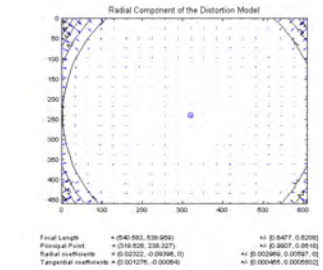


(a) Distorsión radial (Cámara RGB)

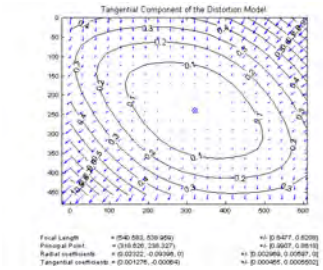
(b) Distorsión tangencial (Cámara RGB)

(c) Modelo completo de distorsión (Cámara RGB)

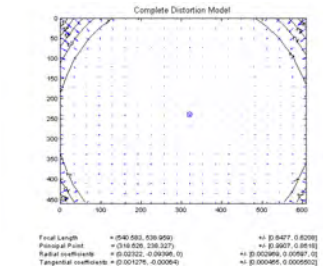
Figura 3.10: Modelos de distorsión del calibrado de Burrus para Microsoft Kinect



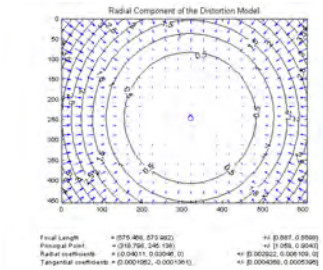
(a) Distorsión radial (Cámara RGB)



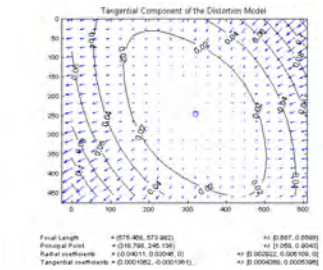
(b) Distorsión tangencial (Cámara RGB)



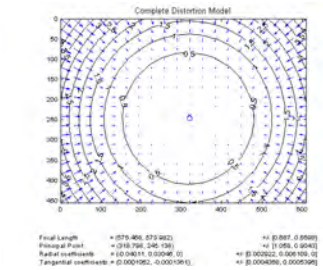
(c) Modelo completo de distorsión (Cámara RGB)



(d) Distorsión radial (Cámara Ir)

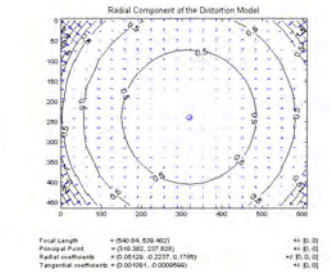


(e) Distorsión tangencial (Cámara Ir)

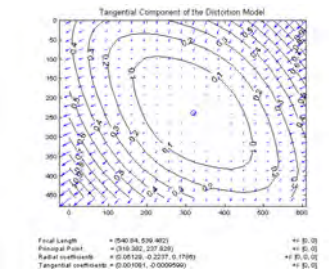


(f) Modelo completo de distorsión (Cámara Ir)

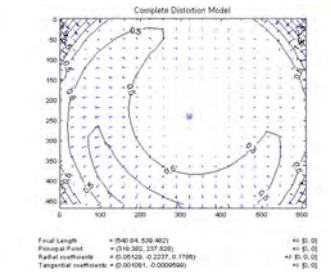
Figura 3.11: Modelos de distorsión del calibrado de Bouguet para PrimeSense Carmine 1.09



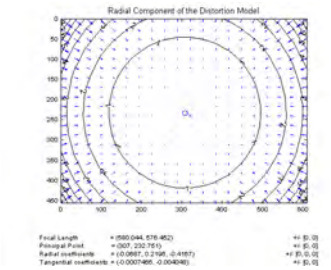
(a) Distorsión radial (Cámara RGB)



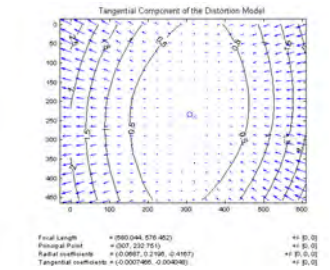
(b) Distorsión tangencial (Cámara RGB)



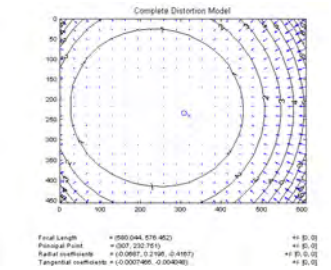
(c) Modelo completo de distorsión (Cámara RGB)



(d) Distorsión radial (Cámara Ir)

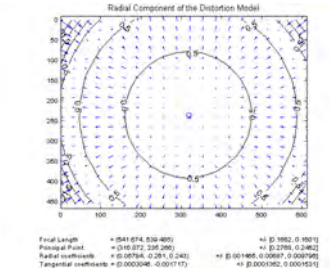


(e) Distorsión tangencial (Cámara Ir)

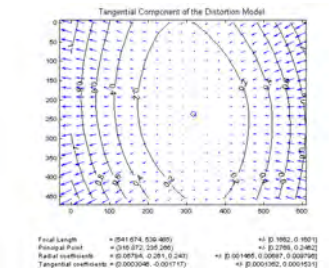


(f) Modelo completo de distorsión (Cámara Ir)

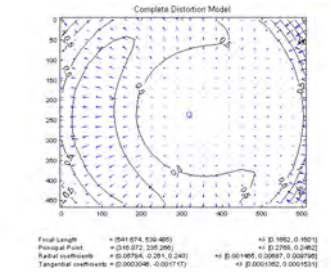
Figura 3.12: Modelos de distorsión del calibrado de Burrus para PrimeSense Carmine 1.09



(a) Distorsión radial (Cámara RGB)



(b) Distorsión tangencial (Cámara RGB)



(c) Modelo completo de distorsión (Cámara RGB)

Figura 3.13: Modelos de distorsión del calibrado de Burrus para PrimeSense Carmine 1.09

3.3 ANÁLISIS CUANTITATIVO DEL ERROR

En este apartado se realiza un análisis cuantitativo del error a través de varias pruebas, con el objetivo de determinar de forma analítica, el efecto en los datos de cada método de calibrado.

3.3.1 *Corrección del desplazamiento entre el infrarrojo y la profundidad*

En esta sección se pretende corregir el desplazamiento que existe entre las imágenes infrarrojas y de profundidad introducido en la Sección 2.2.1. Para ello se han utilizado técnicas de detección de bordes en ambas imágenes, y se ha medido, manualmente, la distancia entre dos puntos para cada eje. Esto ha permitido obtener un desplazamiento medio para cada sensor.

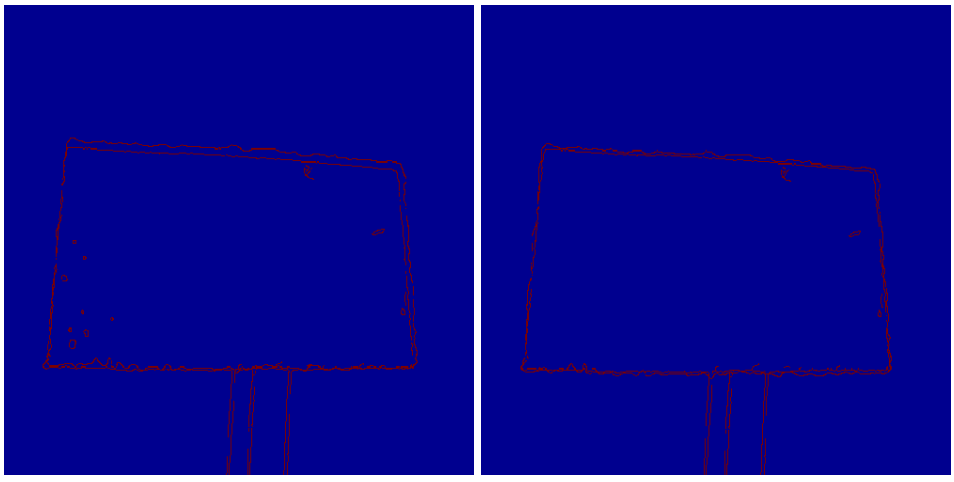
Según los datos obtenidos en las Tablas 3.9 y 3.10 se puede determinar el desplazamiento a aplicar a cada eje de la imagen infrarroja obtenida con Kinect, siendo de 3,5 píxeles en el eje X y de 3,7 en el eje Y. El resultado de aplicar esta corrección se puede observar en la Figura 3.14.

En el caso del sensor Carmine 1.09, se obtiene un desplazamiento menor, siendo la media en el eje X de 2,5 píxeles y en Y de 2,1. El resultado de aplicar este desplazamiento se puede observar en la Figura 3.15.

Este desplazamiento es necesario corregirlo cuando se utiliza la imagen infrarroja para realizar el proceso de calibrado, debido a que las correcciones que se aplican a la imagen de profundidad debido a la distorsión, o el alineamiento entre la de profundidad y el color, se han calculado utilizando imágenes infrarrojas y no de profundidad.

Punto origen	Punto destino	Desplazamiento (px)
493, 200	487, 200	5
498, 237	491, 237	6
300, 271	494, 271	5
503, 305	497, 305	5
506, 345	501, 345	4
76, 220	79, 220	2
72, 242	74, 242	1
69, 275	72, 275	2
64, 330	66, 330	1
61, 352	66, 352	4

Tabla 3.9: Desplazamiento en el eje X ente las imágenes infrarrojas y de profundidad para Microsoft Kinect

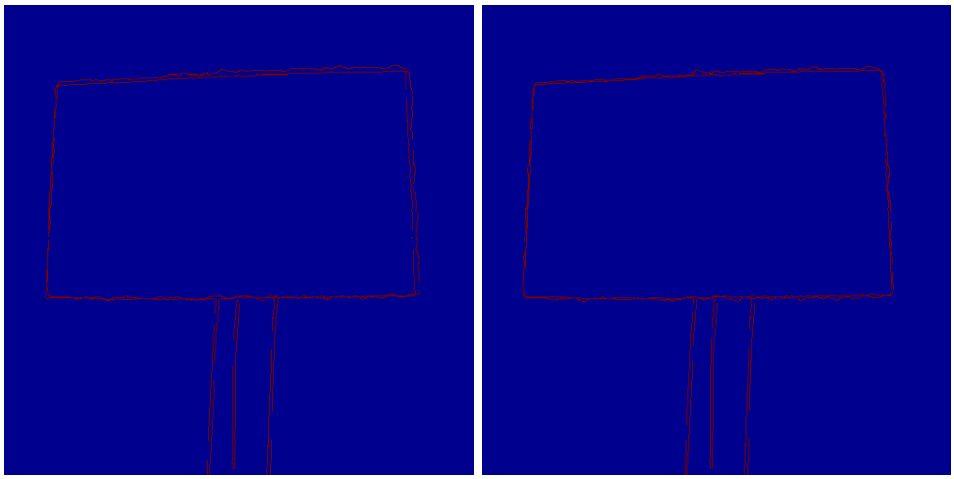


(a) Original (b) Modificada

Figura 3.14: Superposición de los bordes detectados en la imagen infrarroja y de profundidad para Microsoft Kinect

Punto origen	Punto destino	Desplazamiento (px)
99, 150	99, 157	6
185, 156	185, 161	4
254, 157	254, 164	6
346, 161	346, 169	7
455, 170	455, 177	6
110, 374	110, 370	3
190, 373	190, 375	1
326, 371	326, 375	3
423, 374	423, 376	1
476, 374	476, 375	0

Tabla 3.10: Desplazamiento en el eje Y ente las imágenes infrarrojas y de profundidad para Microsoft Kinect



(a) Original (b) Modificada

Figura 3.15: Superposición de los bordes detectados en la imagen infrarroja y de profundidad para Carmine 1.09

Punto origen	Punto destino	Desplazamiento (px)
503, 85	508, 85	4
504, 105	511, 105	6
507, 154	512, 154	4
506, 214	511, 214	4
514, 281	519, 281	4
74, 122	76, 122	1
72, 161	73, 161	1
66, 256	67, 286	1
65, 286	67, 286	0
68, 219	69, 219	0

Tabla 3.11: Desplazamiento en el eje X ente las imágenes infrarrojas y de profundidad para PrimeSense Carmine 1.09

Punto origen	Punto destino	Desplazamiento (px)
89, 90	89, 94	3
159, 87	159, 91	3
254, 83	254, 86	2
369, 77	369, 82	4
462, 76	462, 81	4
128, 302	128, 304	1
243, 304	243, 305	0
329, 303	329, 303	2
400, 301	400, 303	1
474, 301	474, 303	1

Tabla 3.12: Desplazamiento en el eje Y ente las imágenes infrarrojas y de profundidad para PrimeSense Carmine 1.09

3.3.2 *Análisis del error respecto a la profundidad*

El error en eje Z es la diferencia que existe entre la profundidad a un punto y la obtenida por la cámara. La situación ideal para obtener este error es situar un plano completamente paralelo a la cámara a una distancia lo más exacta posible y ocupando todo el campo de visión. Sin embargo, es muy difícil obtener esta situación por varias razones. En primer lugar, conseguir que tanto la cámara como el plano estén lo más paralelo posible, sin el entorno o las herramientas adecuadas, resulta prácticamente imposible. En segundo lugar, sólo se puede medir la distancia entre el sensor y el plano de forma aproximada, y no desde la lente de la cámara, que sería lo más adecuado.

Por esta razón, para analizar este error, se han obtenido capturas de una pared plana a varias distancias, situando las cámaras lo más paralelas posible. A continuación, se han aplicado las correcciones según los distintos métodos de calibrado a los datos. Posteriormente, se ha extraído un conjunto de 100×100 *píxeles* del centro del plano de la nube de puntos (al ser a priori los que menos error tienen), y con esos valores se ha obtenido el modelo de un plano utilizando RANSAC (*Random Sample Consensus*). Con ese plano, se han eliminado los puntos atípicos de la nube de puntos, valores a cero que no se detectan por la cámara. De los datos restantes se ha estimado la distancia del punto al plano para ver el error en la posición.

Analizando esta distancia para cada método de calibrado empleado se obtiene el gráfico de la Figura 3.16, en el cual se puede observar como los datos corregidos con el calibrado del algoritmo de Herrera son los que aportan un menor error, seguido por los de Bouguet y Burrus, entre los que hay muy poca diferencia, y finalmente los originales.

Observando este error según el método de calibrado para la cámara Microsoft Kinect (Figura 3.17), los datos obtenidos con el calibrado de Herrera es el que aporta menor error. Además, este calibrado es el que menor dispersión tiene de los datos referenciado por la

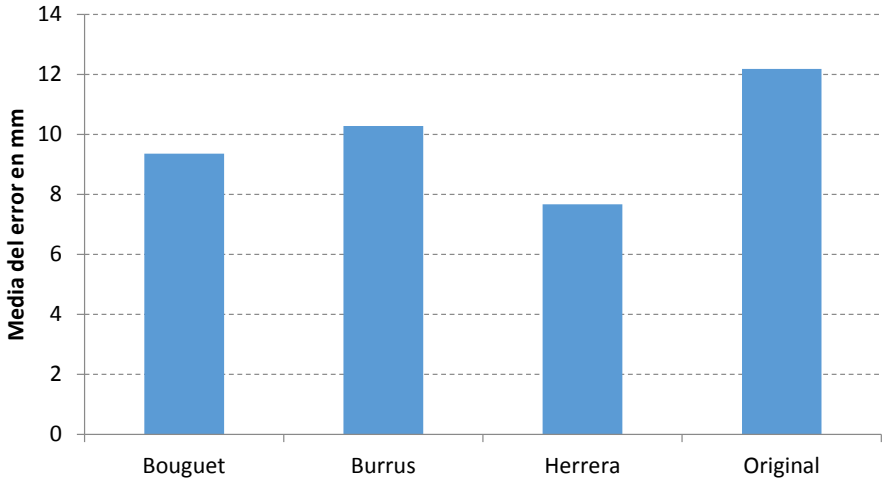


Figura 3.16: Error en la profundidad para cada método de calibrado

desviación típica (barras de error), que son las menores para todos los casos.

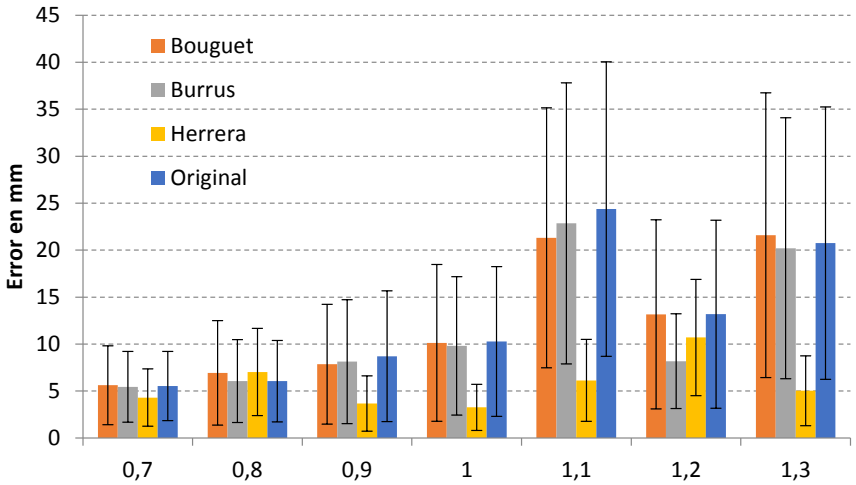


Figura 3.17: Error en la profundidad de los diferentes métodos de calibrado en Microsoft Kinect

Para el sensor de PrimeSense (Figura 3.18) sucede algo parecido, sin embargo, los datos del calibrado de Herrera tiene mayor error que en el caso anterior, pero en general son los mejores.

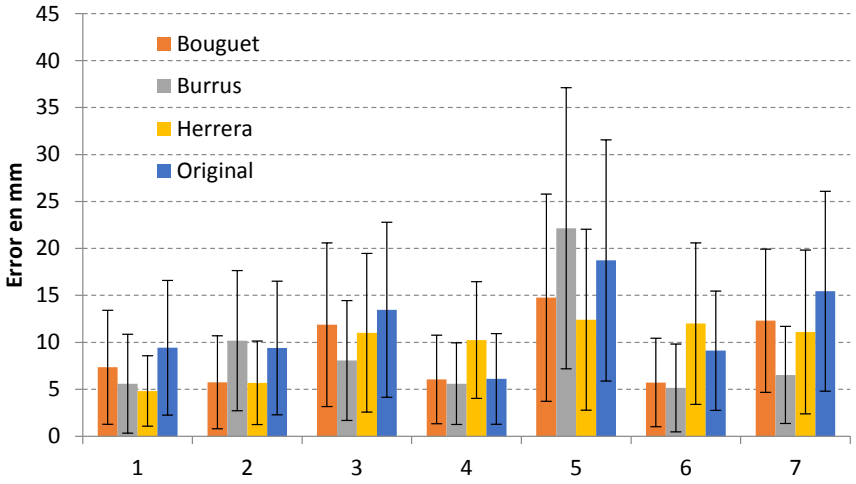


Figura 3.18: Error en la profundidad de los diferentes métodos de calibrado en PrimeSense Carmine 1.09

En el caso de Microsoft Kinect v2 (Figura 3.19), a diferencia de lo que ocurre con el resto de cámaras, los valores de error del calibrado de Bouguet están un poco por encima de los originales, pero sin una diferencia destacable.

3.3.3 *Error en las coordenadas XY de la nube de puntos*

El error en las coordenadas XY en la nube de puntos consiste en analizar el error que entre la posición de los puntos en el espacio real y la nube de puntos. Para ello se toma la distancia en la nube de puntos entre dos puntos conocidos y se compara con la distancia real (distancia conocida).

Para analizar este error se han realizado dos adquisiciones con cada cámara de cinco marcadores como el de la Figura 3.20 repartidos en el campo de visión de la cámara, como se muestra en la Figura 3.21, a dos distancias diferentes, 1,5m y 2m.

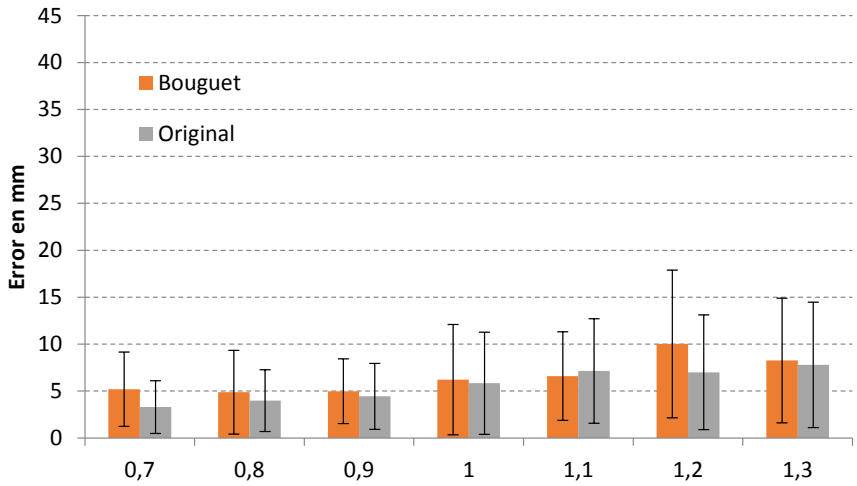


Figura 3.19: Error en la profundidad de los diferentes métodos de calibrado en Microsoft Kinect v2

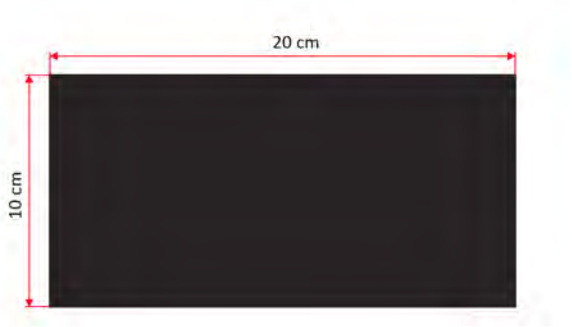
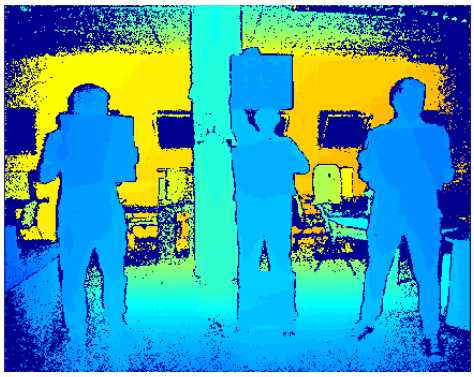


Figura 3.20: Marcador de 20cm × 10cm

A continuación, para cada calibrado de cada cámara se han aplicado las diferentes correcciones a las imágenes adquiridas y se han obtenido las medidas manualmente desde la nube de puntos.

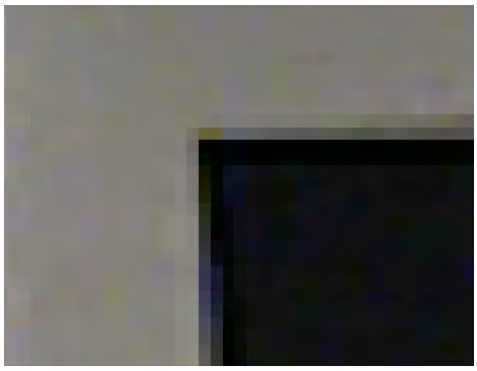
Hay que destacar que la selección de las esquinas de los marcadores para medir las distancias se ha realizado de forma manual sobre la imagen de color, por lo que a medida que incrementa la distancia, la definición de las esquinas en la imagen de color es peor (Figura 3.22), dificultando su selección

(a) Imagen de color

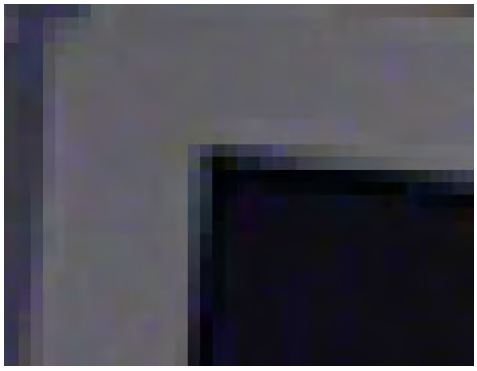


(b) Imagen de profundidad

Figura 3.21: Imágenes de color y profundidad con el marcador



(a) Esquina del marcador a 1,5m



(b) Esquina del marcador a 2m

Figura 3.22: Esquina del marcador a distintas profundidades

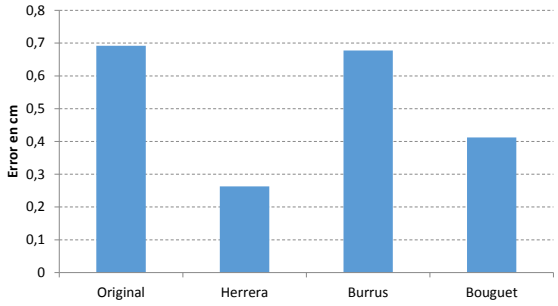


Figura 3.23: Valor del error para cada método de calibrado

La gráfica de la Figura 3.23 muestra el error medio de todas las mediciones para los tres métodos probados, comparándolos con el obtenido con los datos originales aportados por los *driver*. Como se puede observar, los mejores resultados se obtienen con el calibrado de Herrera, seguido por el de Bouguet, y finalmente el de Burrus, que mejora muy poco respecto a los resultados originales confirmándose los datos vistos en la sección donde el algoritmo de Herrera en términos generales provee los mejores datos.

Realizando un desglose del error según las tres cámaras utilizadas (Figura 3.24), es decir, mostrando para cada método el error medio de cada cámara, los tres algoritmos de calibrado mejoran el error respecto a los datos originales, a excepción del método de Burrus para Kinect, con unos errores superiores a los originales.

Según la distancia a la que se sitúan los marcadores (1,5m y 2m) y su posición en la imagen, el error medio para cada método se muestra en las Figuras 3.25 y 3.26. Se puede observar como a mayor distancia (Figura 3.26) el error es mayor, esto es debido a una mayor dispersión de los datos y a la selección manual de los puntos para determinar las distancias. Entendiendo la dispersión de los datos como el error en la posición en el espacio 3D respecto a la posición real, el error incrementa proporcionalmente a la distancia. Una diferencia de un píxel en la imagen de profundidad debido a la distorsión puede suponer una desviación de un centímetro a un metro de distancia en el espacio 3D. Sin embargo, a

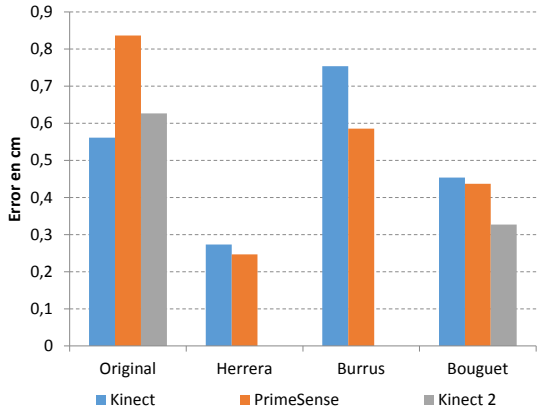


Figura 3.24: Valor del error para cada método de calibrado según el sensor utilizado

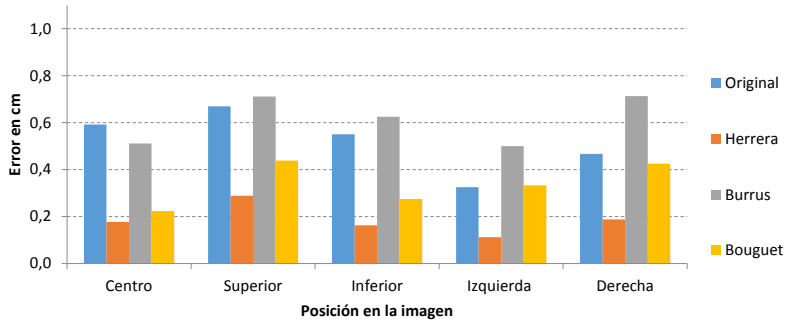


Figura 3.25: Valor del error según la posición en la imagen a una distancia de 1,5m

2 metros el error en ese mismo punto puede ser de cuatro centímetros.

En la gráfica de la Figura 3.25 el método de Herrera es el que produce menor error, el de Bouguet proporciona unos resultados en torno a 1mm peores en el peor de los casos, y con el de Burrus se obtiene más error que utilizando los datos originales.

En el caso de la gráfica de la Figura 3.26, obtenida con los marcadores situados a 2m se obtiene mayor error en casi todos los casos. Al igual que en la gráfica anterior, el de Herrera proporciona los mejores resultados, seguido por el de Bouguet. En este caso, los resultados

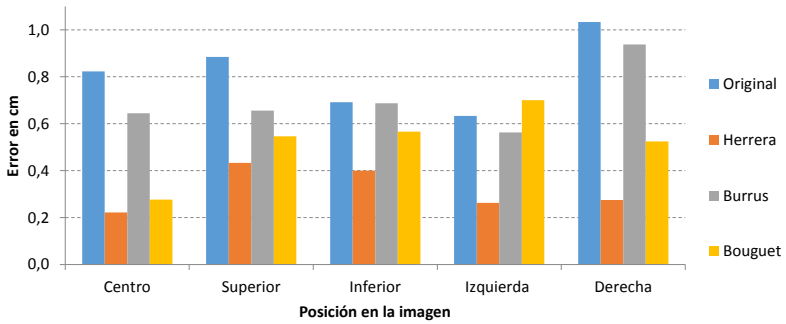


Figura 3.26: Valor del error según la posición en la imagen a una distancia de 2m

del método de Burrus son mejores que los originales.

3.4 ANÁLISIS CUALITATIVO DEL ERROR

En esta sección se ha llevado a cabo una experimentación cualitativa basada en técnicas de registro de puntos 3D, evaluando mediante inspección visual los resultados. Una de las aplicaciones más comunes de los sensores RGB-D es la reconstrucción 3D, para lo que es necesario un proceso de registro en el que se transforman las diferentes vistas a origen de coordenadas común. Para estos casos, los datos deben ser lo más precisos posibles y sin distorsiones.

Por esta razón, se han realizado pruebas de registro utilizando el método $\mu - MAR$ [Saval Calvo, 2015], que realiza un registro de diferentes vistas basándose en marcadores situados en la escena para obtener un modelo 3D de un objeto. Para los marcadores se han empleado cubos (Figura 3.27). El método de registro segmenta estos cubos en las imágenes de color y profundidad para posteriormente buscar los planos que los componen. Por lo tanto, es fundamental que los puntos describan correctamente las caras de los cubos. Una vez conocidos los cubos, se aplican las transformaciones necesarias a toda la nube de puntos

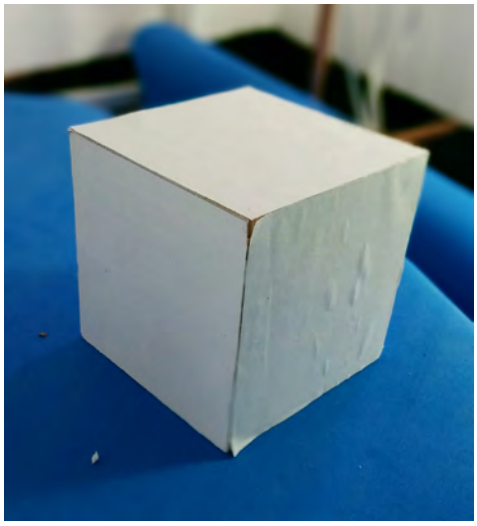


Figura 3.27: Marcador del método $\mu - MAR$

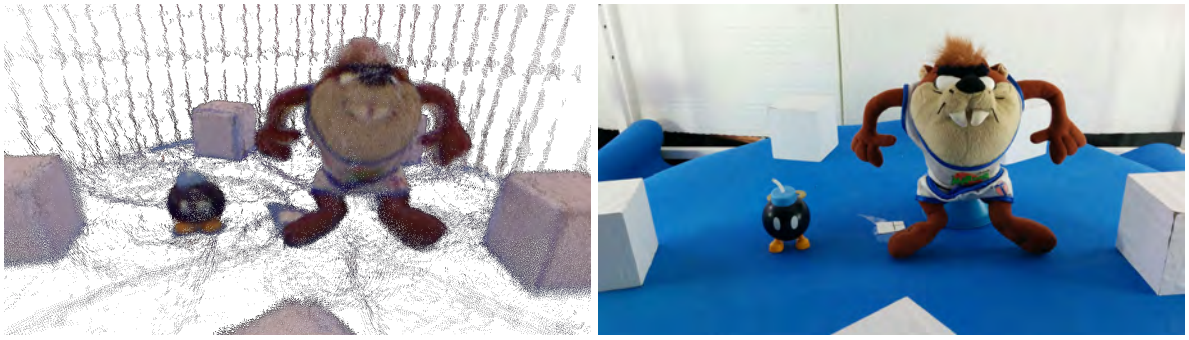
para alinear el objeto, además de los cubos.

Estas pruebas se han llevado a cabo en un entorno controlado, para minimizar el efecto de factores externos en la percepción de los datos. Este entorno se define con detalle en el Anexo A. Se han utilizado dos objetos mostrados en la Figura 3.28, situados simultáneamente en el entorno, de los cuales se han tomado 79 capturas con cada cámara y posteriormente se han aplicado los cálculos y correcciones de cada método de calibración. El objetivo es obtener una escena reconstruida en 3D a partir de varias vistas de una real, como se muestra en la Figura 3.29.

Para observar a rasgos generales como afecta el calibrado al registro, la Figura 3.30 muestra la vista frontal del registro de los dos objetos, utilizando el sensor Carmine 1.09 y los datos corregidos por los tres métodos de calibrado utilizados en comparación con los originales. Los datos originales son aquellos que devuelve la cámara por defecto y sobre los que no se ha aplicado ninguna corrección. El resultado del registro de estos datos se muestra en la Figura 3.30a, en él se puede observar como hay puntos mal registrados, por ejemplo, los que se encuentran en las manos se juntan con el resto del cuerpo. El registro con los

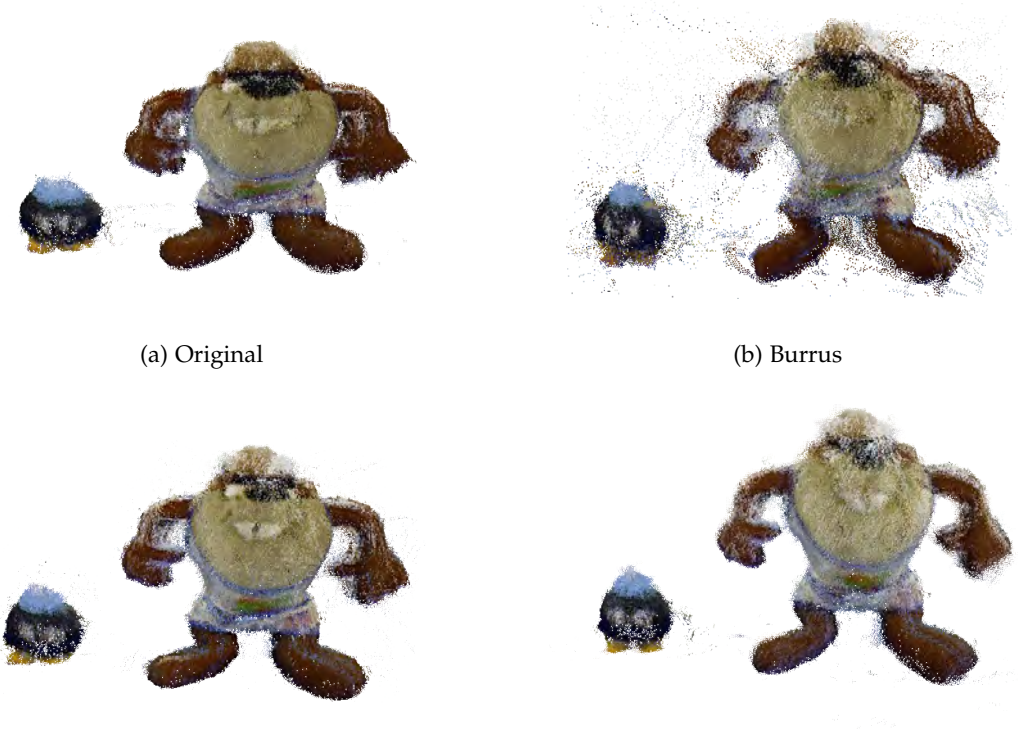
(a) Objeto 1 (*Taz*) (b) Objeto 2 (*Bob-omb*)

Figura 3.28: Objetos empleados en el registro



(a) Escena registrada (b) Escena real

Figura 3.29: Registro obtenido a partir de una escena real



(a) Original

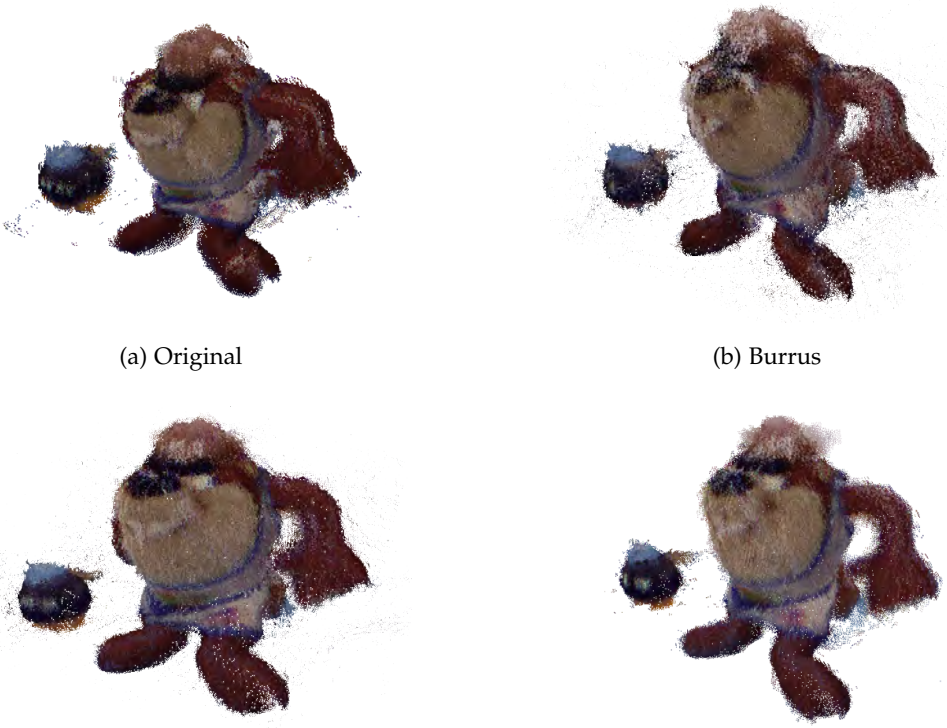
(b) Burrus

(c) Bouguet

(d) Herrera

Figura 3.30: Vista frontal de los diferentes registros obtenidos con el sensor Carmine 1.09

datos de Burrus (Figura 3.30b) mejora notablemente respecto al original, aunque presenta altos niveles de ruido (muchos puntos dispersos cerca del objeto) y continúa teniendo errores de registro, pero se reducen en gran medida, por ejemplo el brazo o pie derecho. El resultado de Bouguet (Figura 3.30c) mejora en el registro respecto al de Burrus, los errores son menos notable y además no tiene tanto ruido. Por último, el método de Herrera es con el que se obtiene los mejores resultados (Figura 3.30d), ya que el modelo final representa de forma adecuada el real. Aunque se pueden observar puntos de color en lugares donde no se corresponden (por ejemplo, en el pie derecho se observan puntos azules), se debe principalmente a reflejos del resto del entorno que se hacen visibles al registrar todas las vistas.



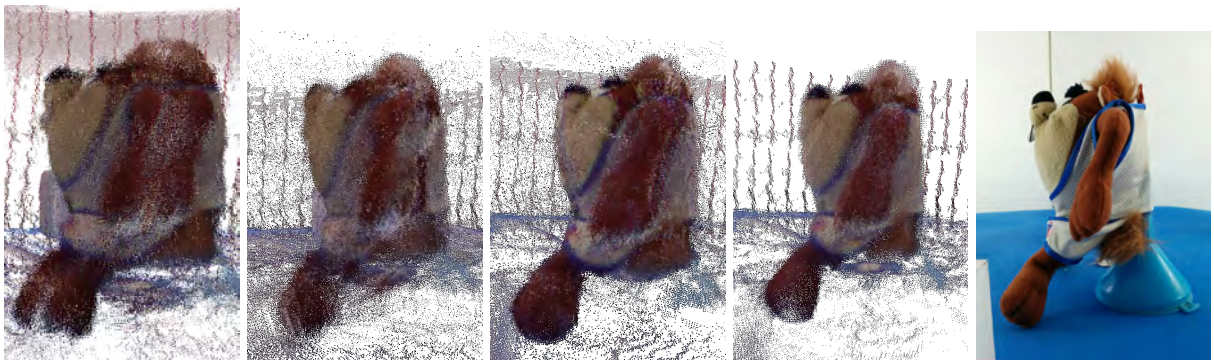
(a) Original (b) Burrus

(c) Bouguet (d) Herrera

Figura 3.31: Perspectiva de los diferentes registros obtenidos con el sensor Microsoft Kinect

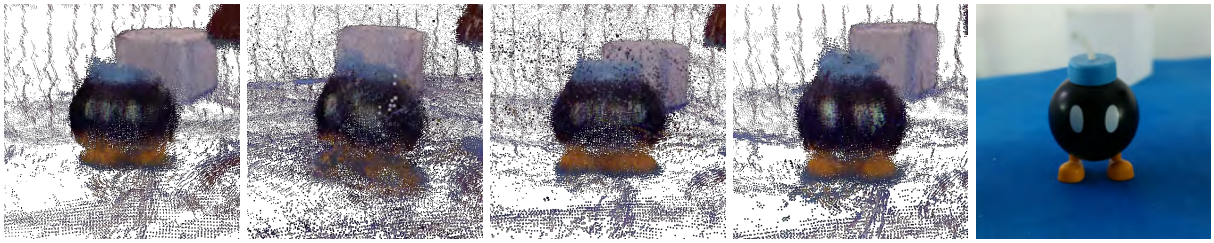
Unos resultados similares se aprecian en la Figura 3.31 para el sensor Microsoft Kinect, especialmente en el brazo derecho del Objeto 1, que se encuentra registrando con un alto nivel de error utilizando los datos originales (Figura 3.31a) y es difícilmente diferenciable. Este error mejora con los datos de Burrus y Bouguet (Figura 3.31b y 3.31c), pero se obtiene el mejor resultado con los datos corregidos mediante el algoritmo de Herrera, como se muestra en la Figura 3.31d.

La mejora de los datos calibrados respecto a los originales se puede observar desde una vista de perfil del Objeto 1 (Figura 3.32), viendo en detalle el brazo derecho. Destaca el resultado de Herrera (Figurar 3.32d), cuya representación 3D del modelo es el que más se aproxima al real, y el que contiene menos fallos de registro.



(a) Original (b) Burrus (c) Bouguet (d) Herrera (e) Real

Figura 3.32: Vista de perfil del registro del Objeto 1 con los datos de los diferentes métodos de calibración en comparación con el real para el sensor Microsoft Kinect



(a) Original (b) Burrus (c) Bouguet (d) Herrera (e) Real

Figura 3.33: Vista centrada del registro del Objeto 2 con los datos de los diferentes métodos de calibración en comparación con el real para el sensor Microsoft Kinect

El método de Herrera también es el que aporta mejores resultados para el Objeto 2 (Figura 3.33d). En comparación con el resto de métodos, es el que mejor representa la forma del objeto. Además, hay que tener en cuenta que se trata de un objeto de unos 5cm de alto, lo que dificulta tanto la adquisición como el registro.

El sensor Microsoft Kinect v2 utiliza una tecnología distinta al resto de los sensores utilizados en este trabajo, por lo que presenta ciertos errores que impiden el registro utilizando el método $\mu - MAR$. La Figura 3.34 muestra una escena, desde diferentes puntos de vista, adquirida con este sensor. Aunque la Figura 3.34a no parece presentar grandes errores, si se observa desde otro punto de vista (Figura 3.34b) se aprecia fácilmente una estela de

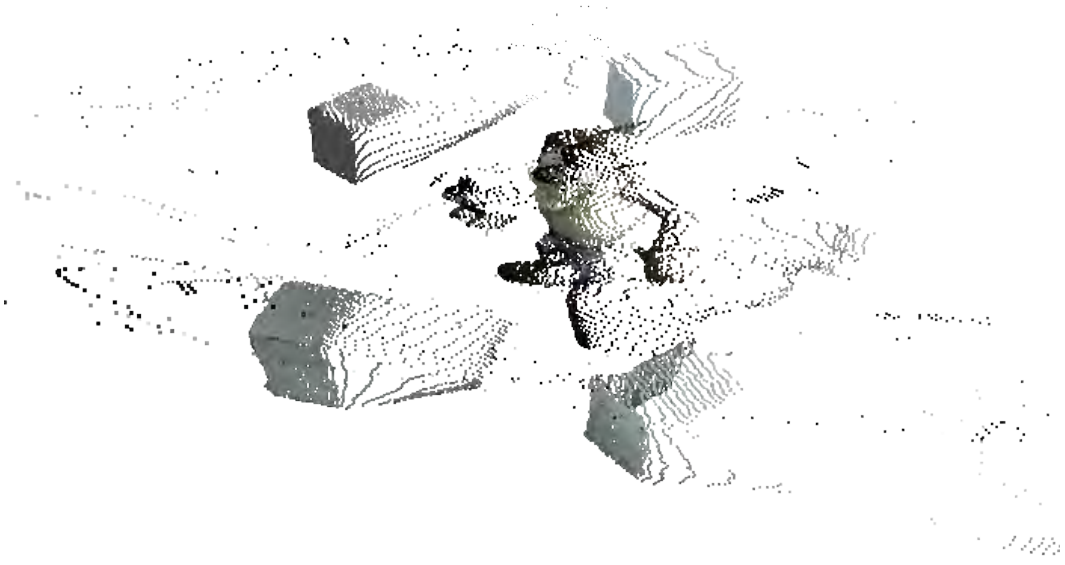
puntos erróneos en los cubos.

Además, no se trata de una estela uniforme respecto al cubo (Figura 3.35a), llegando en algunos a casos a ser incoherente como la que se muestra en la Figura 3.35b.

Por otra parte, los puntos que representan los planos de un cubo no forman un ángulo de 90 grados como se puede observar en la Figura 3.36b, y estos planos, en ocasiones, son curvos como el que se aprecia en la Figura 3.37.

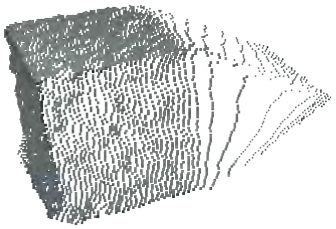
Dado que el método $\mu - MAR$ se basa en la detección de los marcadores para realizar la reconstrucción 3D, resulta imposible su aplicación a los datos obtenidos con el sensor Microsoft Kinect v2 por todas la dificultades que presenta, provocando resultados como el que se muestra en la Figura 3.38.

(a) Vista frontal



(b) Vista en perspectiva

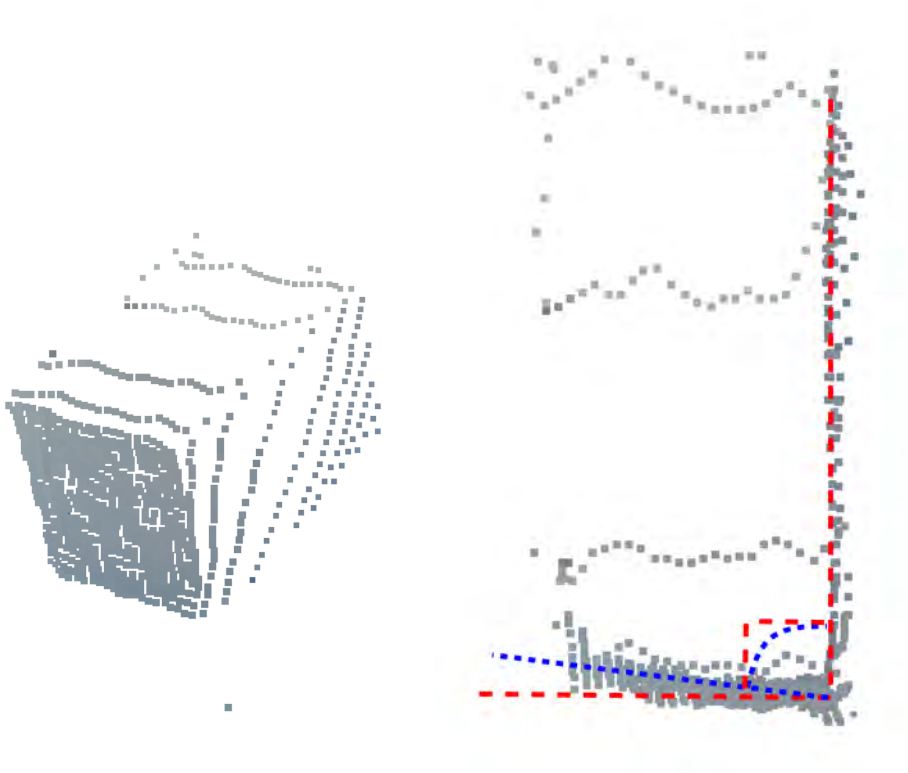
Figura 3.34: Escena adquirida con el sensor Microsoft Kinect v2 representada desde diferentes puntos de vista



(a)

(b)

Figura 3.35: Estela descrita por un cubo marcador



(a) Vista en perspectiva

(b) Vista cenital. Ángulo esperado (rojo) y obtenido (azul)

Figura 3.36: Angulo formado por los puntos de los planos que describen el cubo

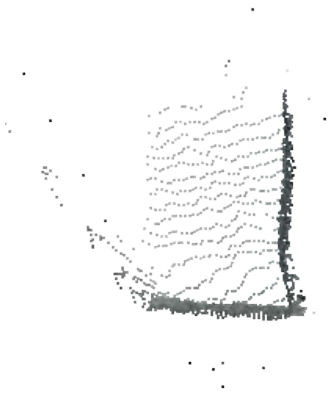


Figura 3.37: Puntos que definen el cubo marcador

Figura 3.38: Registro realizado con los datos adquiridos con Microsoft Kinect v2

4

CONCLUSIÓN

4.1 CONCLUSIONES

Viendo el amplio uso de los sensores de propósito general RGB-D en situaciones adversas, resulta imprescindible conocer los algoritmos de calibración existentes que permiten mejorar la calidad de los datos percibidos. Por ello, es necesario conocer qué parámetros intervienen en este proceso, además de las tecnologías disponibles más comunes (luz estructurada y tiempo de vuelo) que emplean estos sensores para obtener información de profundidad de la escena.

Por ello se ha hecho una revisión del estado del arte, mostrando los sensores que existen actualmente, prestando atención a los de propósito general RGB-D. Además, se han indicado algunos trabajos que hacen uso de ellos, lo que ha permitido observar como aquellos que lo utilizan en el límite de su sensibilidad realizan un proceso de calibrado previo.

Posteriormente, se han introducido tres algoritmos de calibrado conocidos y se ha realizado el proceso de calibrado para tres cámaras RGB-D diferentes: Microsoft Kinect, PrimeSense Carmine 1.09 y Microsoft Kinect v2, las dos primeras se basan en luz estructurada para determinar la profundidad, mientras que la tercera utiliza tiempo de vuelo. Los algoritmos utilizados han sido el de Burrus, Bouguet y Herrera. El algoritmo de Bouguet es genérico para estéreo, mientras que el de Burrus y Herrera están más centrados en sensores RGB-D.

Para analizar el comportamiento de cada algoritmo se han obtenido resultados para diferentes experimentos, realizando una serie de pruebas para analizar la mejora de los distintos calibrados. De estas pruebas se ha podido determinar que los datos calibrados con el método propuesto por [Daniel Herrera C et al., 2012] son los que han obtenido mejores resultados en términos generales. Tambén se ha demostrado que el resto de algoritmos probados obtienen mejoría respecto a los datos originales.

Finalmente, estos resultados se han verificado mediante inspecciones visuales para aplicaciones de registro 3D, que es uno de los usos más comunes para este tipo de sensores en investigación. Para ello, se ha utilizado el método $\mu - MAR$ [Saval Calvo, 2015], que se basa en la detección de marcadores para realizar una reconstrucción 3D. Este método se ha aplicado a los sensores Kinect y Carmine 1.09, analizando aquellos inconvenientes que impiden su aplicación con los datos aportados por Kinect v2. Esto ha permitido observar visualmente que los resultados de registro con los datos de Herrera son los que mejor representan el modelo real.

La principal aportación de este trabajo es un análisis entre distintos métodos de calibrado tanto de propósito general, como de otros más específicos para sensores RGBD.

Este trabajo se enmarca dentro del proyecto financiado por la Generalitat Valenciana, titulado “Adquisición y modelado tridimensional del crecimiento de plantas” (GV/2013/005), donde he intervenido con el calibrado de sensores RGBD, que también ha contribuido en la mejora del registro basado en planos del método $\mu - MAR$.

4.2 LÍNEAS FUTURAS

Como líneas futuras a corto plazo para este trabajo se plantea la comparativa con métodos más recientes a los utilizados, por ejemplo el propuesto en [Staranowicz et al., 2014]. Ade-

más, también sería interesante un sitio web que permita a la comunidad publicar el método empleado, los resultados y una evaluación de ellos, de manera que permita situarlos en un *ranking*.

Por otra parte, el algoritmo de Herrera es el que ha aportado los mejores resultados de los evaluados en este trabajo. Por ello, se plantea introducir el modelo de error de la tecnología de tiempo de vuelo en esté método, de manera que fuese posible su aplicación a un mayor rango de sensores, entre ellos Kinect v2.

También se considera interesante ampliar y depurar las herramientas desarrolladas en la Sección 3.1 con el objetivo de publicarlas a la comunidad, para que cualquier persona que las necesite o las encuentre útiles, pueda hacer uso de ellas. Uno de los inconvenientes que presentan algunas de estas herramientas es que necesitan un tiempo de computo elevado, en especial la que aplica las correcciones a un conjunto de capturas dado un calibrado. Dado que la mayoría de las operaciones que realiza son altamente paralelizables, se plantea mejorar su rendimiento utilizando para ello tecnologías como CUDA.

Finalmente, se plantea a largo plazo realizar un análisis del calibrado para determinar su efecto en el análisis del comportamiento basado en trayectorias tridimensionales utilizando *Activity Descrition Vector* ADV [Azorín-lópez et al., 2013] y mapas auto-organizados capaces de preservar la topología de los datos de entrada [Azorín-López et al., 2015]. Para esta tarea se realiza una discretización de la trayectoria, por lo que resulta fundamental que su posición en la nube de puntos sea lo más exacta posible.

ANEXOS

A

ENTORNO EXPERIMENTAL

Con el objetivo de realizar la adquisición 3D de objetos, se ha creado un entorno experimental compuesto por una cámara en una posición fija, un Arduino que controla una mesa giratoria, y un ordenador que obtiene las imágenes de la cámara. La mesa giratoria permite obtener imágenes alrededor de los 360 grados del motivo.

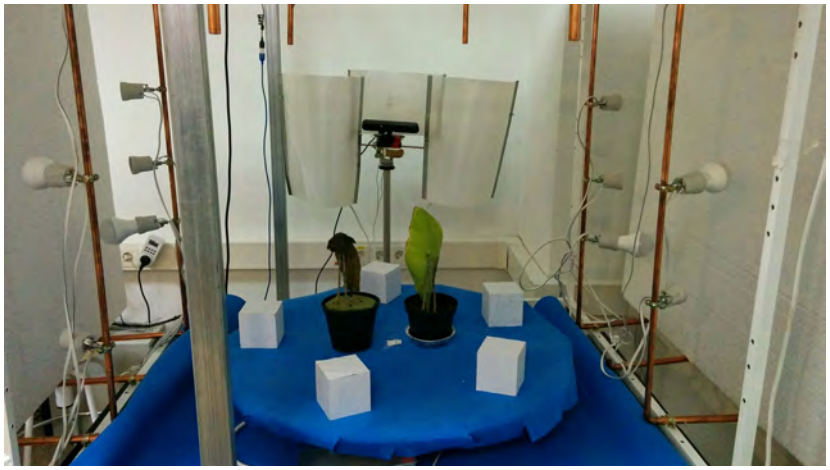


Figura A.1: Entorno experimental

El entorno está cubierto con paneles de poliestireno, evitando que luces externas al entorno afecten a la adquisición. Para la iluminación interna se han utilizado bombillas LED de luz blanca orientadas hacia los paneles (Figura A.2) para difuminar la luz y evitar que incidan directamente en la escena. Además, las bombillas LED no producen la misma radiación en el espectro infrarrojo que las bombillas incandescentes, evitando interferencias en el patrón de motas infrarrojo que utilizan los sensores basados en luz estructurada.

Figura A.2: Bombilla LED orientada al panel de poliestireno

La mesa giratoria está situada sobre un motor paso a paso controlado por una placa Arduino Uno, permitiendo obtener vistas alrededor de los 360 grados de la escena. También tiene un sensor magnético (Figura A.3) que permite conocer la posición inicial de la mesa y determinar cuando se ha completado un giro. El programa cargado en la placa Arduino permite el control del motor paso a paso utilizando comandos transmitidos a través de puerto serie RS232. Entre ellos se incluye: para comprobar o ir a la posición inicial, mover un paso, establecer la dirección de la rotación y el número de pasos para completar un giro, entre otros.

La placa Arduino Uno esta conectada a un ordenador con Matlab R2011a, el cual controla la cámara y los comandos de Arduino. La cámara RGB-D es una PrimeSense Carmine 1.09 (Figura A.4), utilizando el *driver* OpenNI para Windows en su versión 1.5. Además, se utiliza *Kinect for Matlab*[1] para controlar la cámara desde Matlab. Esta herramienta proporciona funciones MEX (*Matlab Executable*), permitiendo utilizar las funciones de la API de OpenNI en Matlab.

1 *http://sourceforge.net/projects/kinect-mex/* última visita: 1 de Abril, 2015

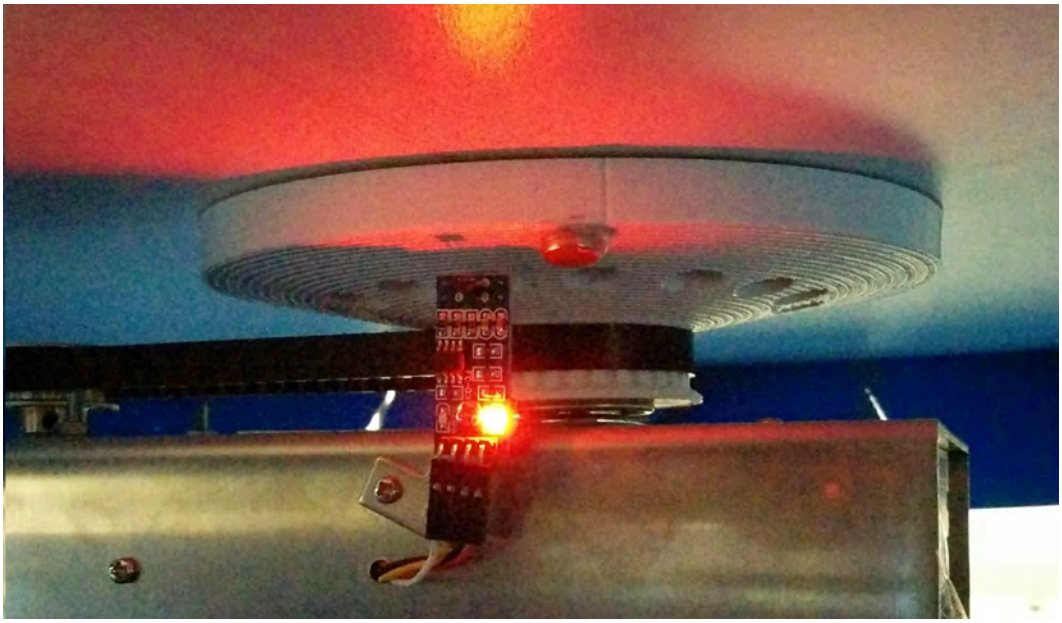


Figura A.3: Sensor magnético

Un *script* desarrollado en Matlab controla el proceso de adquisición de la escena. Para realizar una adquisición, la rotación se divide en n pasos. Para cada paso de la mesa, se obtiene una imagen de color, una de profundidad (*depth*) y una de disparidad. Además, entre cada paso del motor y la adquisición de las imágenes, se espera un intervalo de tiempo para que se estabilicen los objetos de la escena. También se almacena la nube de puntos.

En el caso de que sea necesario la adquisición de datos de plantas durante su periodo de crecimiento, el entorno dispone con luces especiales para simular la luz natural que necesitan las plantas para crecer. Para evitar interferencias provocadas por estas luces durante la adquisición, son apagadas previamente al proceso.

Figura A.4: PrimeSense Carmine 1.09

BIBLIOGRAFÍA

Azorín López, J. (2007). *Modelado de sistemas para visión de objetos especulares: inspección visual automática en producción industrial*. PhD thesis, Universidad de Alicante. (Citado en las páginas 2, 3, 4, 10 y 14.)

Azorín-lópez, J., Saval-calvo, M., Fuster-guilló, A., and García-rodríguez, J. (2013). Human Behaviour Recognition based on Trajectory Analysis using Neural Networks. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 99–105. (Citado en la página 85.)

Azorín-López, J., Saval-Calvo, M., Fuster-Guilló, A., Mora-Mora, H., and Villena-Martínez, V. (2015). Topology Preserving Self-Organizing Map of Features in Image Space for Trajectory Classification. In *6th International Work-Conference on the Interplay between Natural and Artificial Computation, 2015*. (Citado en la página 85.)

Bouguet, J.-Y. (2004). Camera calibration toolbox for matlab. (Citado en las páginas 10, 36, 38 y 41.)

Burrus, N. (2012). Rgbdemo. *Retrieved March*, 30:2012. (Citado en las páginas 9, 38 y 41.)

Cui, Y., Schuon, S., Chan, D., Thrun, S., and Theobalt, C. (2010). 3D shape scanning with a time-of-flight camera. (Citado en la página 5.)

Daniel Herrera C, Juho Kannala, and Janne Heikkilä (2012). Joint depth and color camera calibration with distortion correction. (Citado en las páginas 9, 10, 38, 41 y 84.)

Foix, S., Alenyà, G., and Torras, C. (2011). Lock-in time-of-flight (ToF) cameras: A survey. (Citado en la página 5.)

Freedman, B., Shpunt, A., Machline, M., and Arieli, Y. (2012). Depth mapping using projected patterns. US Patent 8,150,142. (Citado en la página 31.)

Fuster Guilló, A. (2003). *Modelado de sistemas para visión realista en condiciones adversas y escenas sin estructura*. PhD thesis, Universidad de Alicante. (Citado en las páginas 1, 3 y 10.)

Ghosh, S. K. (2005). *Fundamentals of computational photogrammetry*. Concept Publishing Company. (Citado en la página 22.)

Han, J., Shao, L., Xu, D., and Shotton, J. (2013). Enhanced computer vision with Microsoft Kinect sensor: a review. *IEEE transactions on cybernetics*, 43(5):1318–34. (Citado en la página 8.)

Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. (2012). RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. (Citado en la página 6.)

Herakleous, K. and Poullis, C. (2014). 3DUNDERWORLD-SLS: An Open-Source Structured-Light Scanning System for Rapid Geometry Acquisition. (Citado en la página 5.)

Herrera C., D., Kannala, J., and Heikkilä, J. (2011). Accurate and practical calibration of a depth and color camera pair. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6855 LNCS(PART 2):437–445. (Citado en la página 9.)

Izadi, S., Davison, A., Fitzgibbon, A., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., and Freeman, D. (2011). KinectFusion. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*, page 559, New York, New York, USA. ACM Press. (Citado en la página 8.)

Jedvert, M. (2013). 3D Head Scanner - Master of Science Thesis Chalmers University of Technology. (Citado en la página 8.)

Kasper, A., Xue, Z., and Dillmann, R. (2012). The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics. (Citado en la página 7.)

Khoshelham, K. and Elberink, S. O. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454. (Citado en las páginas 6 y 8.)

Lai, K., Bo, L., Ren, X., and Fox, D. (2013). Consumer Depth Cameras for Computer Vision. (Citado en la página 6.)

Lazaros, N., Sirakoulis, G. C., and Gasteratos, A. (2008). Review of Stereo Vision Algorithms: From Software to Hardware. (Citado en la página 5.)

Lovato, C., Bissolo, E., Lanza, N., Stella, A., and Giachetti, A. (2014). A low cost and easy to use setup for foot scanning. (Citado en la página 8.)

Morell-Gimenez, V., Saval-Calvo, M., Azorin-Lopez, J., Garcia-Rodriguez, J., Cazorla, M., Orts-Escolano, S., and Fuster-Guillo, A. (2014). A comparative study of registration methods for RGB-D video of static scenes. (Citado en la página 8.)

Paier, W. (2011). Acquisition of 3D-Head-Models using SLR-Cameras and RGBZ-Sensors. (Citado en la página 8.)

Raposo, C., Barreto, J. P., and Nunes, U. (2013). Fast and Accurate Calibration of a Kinect Sensor. (Citado en la página 9.)

Salvi, J., Fernandez, S., Pribanic, T., and Llado, X. (2010). A state of the art in structured light patterns for surface profilometry. (Citado en la página 5.)

Salvi, J., Pagès, J., and Batlle, J. (2004). Pattern codification strategies in structured light systems. *Pattern Recognition*, 37:827–849. (Citado en la página 5.)

Saval Calvo, M. (2015). *Methodology based on registration techniques for representing subjects and their deformations acquired from general purpose 3D sensors*. PhD thesis, Universidad de Alicante. (Citado en las páginas 2, 10, 14, 72 y 84.)

Schwarz, B. (2010). Lidar: Mapping the world in 3D. (Citado en la página 5.)

Shao, L., Han, J., Kohli, P., and Zhang, Z. (2014). *Computer Vision and Machine Learning with RGB-D Sensors*. Springer International Publishing. (Citado en la página 8.)

Smisek, J., Jancosek, M., and Pajdla, T. (2011). 3D with Kinect. (Citado en las páginas 8, 9 y 36.)

Staranowicz, A., Brown, G. R., Morbidi, F., and Mariottini, G. L. (2014). Easy-to-use and accurate calibration of RGB-D cameras from spheres. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8333 LNCS:265–278. (Citado en las páginas 9 y 84.)

Van Den Bergh, M. and Van Gool, L. (2011). Combining RGB and ToF cameras for real-time 3D hand gesture interaction. (Citado en la página 36.)

Weiss, A., Hirshberg, D., and Black, M. J. (2011). Home 3D body scans from noisy image and range data. (Citado en la página 8.)

Weng, J., Cohen, P., and Herniou, M. (1992). Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980. (Citado en la página 23.)

Zelinsky, a. (2009). Learning OpenCV—Computer Vision with the OpenCV Library (Bradski, G.R. et al.; 2008)[On the Shelf]. (Citado en las páginas 18 y 22.)

Zhang, C. and Zhang, Z. (2011). Calibration between depth and color sensors for commodity depth cameras. *Proceedings - IEEE International Conference on Multimedia and Expo*. (Citado en la página 9.)