

Anexo A.

El dispositivo Kinect

Kinect es un controlador de juego y entretenimiento desarrollado por Microsoft para la videoconsola Xbox 360 y para PC, lanzado al mercado en Noviembre de 2010. En la Figura A.1 se muestra una fotografía del dispositivo y sus componentes.

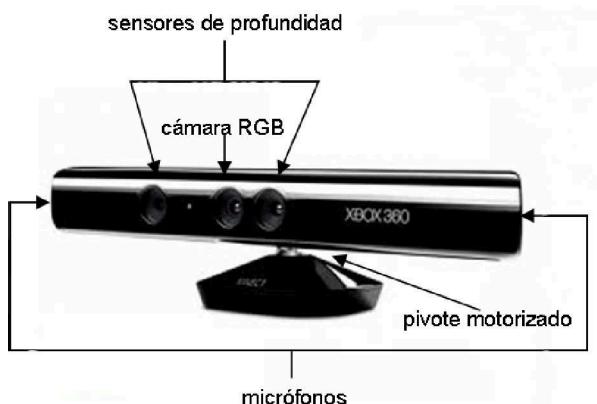


Figura A.1: Dispositivo *Kinect* y sus componentes.

El sensor *Kinect* es una barra horizontal conectada a una pequeña base con un pivote motorizado y está diseñado para posicionarse encima o debajo de una pantalla de vídeo. El dispositivo cuenta con una cámara RGB, un sensor de profundidad y varios micrófonos. También integra un software propietario que permite la captura de movimientos en 3D del cuerpo, reconocimiento facial y reconocimiento por voz. No obstante, en este anexo solo se van a describir las características relacionadas con visión.

El sensor de profundidad consiste en un proyector de láser infrarrojo combinado con un sensor CMOS monocromático, lo que captura datos de vídeo en 3D bajo cualquier condición de luz ambiental.

La cámara RGB del sensor proporciona vídeo a una frecuencia de 30 Hz, con una resolución de 640x480 píxeles y una profundidad de 8 bits, mientras que el sensor de profundidad cuenta con una resolución de 640x480 píxeles con una profundidad de 11 bits, lo que provee hasta 2048 niveles de sensibilidad. El sensor tiene un rango de trabajo de aproximadamente 0.7–6 metros. El sensor tiene un campo de visión angular de 57° horizontalmente y de 43° verticalmente, mientras que el pivote motorizado puede inclinar el sensor hasta 27°, hacia arriba o hacia abajo.

A. El dispositivo Kinect

Desde su lanzamiento han aparecido varios controladores de código abierto que permiten integrarlo en un ordenador bajo cualquier sistema operativo, lo cual ha permitido su explotación para fines de investigación. Actualmente este sensor es la herramienta de trabajo de multitud de investigaciones, pues sus características abren nuevas posibilidades a áreas de investigación relacionadas con la visión por computador, como pueden ser la reconstrucción de escenas 3D, la detección y reconocimiento de objetos, captura de gestos...

Anexo B.

Gestión del proyecto

En este anexo se detalla como se ha gestionado el tiempo para la realización de este proyecto. El periodo durante el cual se ha desarrollado el proyecto abarca desde principios de Octubre de 2011 hasta mediados de Abril de 2012. El diagrama de Gantt representado en la Figura B.1 detalla las tareas que se han llevado a cabo y el tiempo empleado en ellas. Se puede apreciar dos bloques consecutivos, que son el estudio de literatura, diseño, implementación y evaluación de cada sistema de reconocimiento desarrollado. En el diagrama se observa que el sistema final ha llevado más tiempo para ser desarrollado, aun utilizando parte del reconocimiento 2D implementado en el sistema inicial. Las principales causas son el fuerte estudio de documentación llevado a cabo y el aprendizaje de librerías para manejar y procesar información 3D. También se observa un tiempo importante en la integración del dispositivo *kinect* con el ordenador. Esto se debió a que en aquel momento los drivers y librerías para su manejo no eran del todo estables, y su instalación solía ser dificultosa.

Durante el desarrollo del proyecto se mantuvo una reunión semanal con la directora que podía ser presencial o vía Internet. Estas reuniones fueron de utilidad para llevar un control de lo que se había hecho hasta el momento, discutir las posibles vías de trabajo y para recibir ayuda en las cuestiones que surgieron durante el desarrollo del proyecto.

B. Gestión del proyecto



Figura B.1: Diagrama de Gantt que muestra las tareas realizadas y el tiempo empleado en cada una de ellas.

Anexo C.

Experimentos sobre el sistema de inicial de reconocimiento 2D

En este anexo se presentan con detalle todos los experimentos llevados a cabo sobre el sistema de reconocimiento 2D inicial con objeto de evaluar su rendimiento. Este sistema mantiene una pequeña base de datos de 13 objetos, con 8 fotos de cada objeto, lo que hace un total de 104 fotos de referencia.

C.0.1. Experimentos con el algoritmo de búsqueda de vecino más cercano

Todos los algoritmos explicados en esta sección utilizan alguna variante del algoritmo de búsqueda del vecino más cercano, exhaustiva o aproximada, y en algunos casos también incorporan la estimación robusta RANSAC.

Experimento 1: SURF vs ORB

Este experimento se ha realizado probando 30 fotos de test de los objetos de la base de datos, un objeto por foto. Las fotos están tomadas sobre fondo blanco para eliminar ruido y que el reconocimiento sea más sencillo. El objetivo es comparar el rendimiento que ofrecen los descriptores SURF y ORB, así como la búsqueda exhaustiva y la aproximada. En este caso no se utiliza el algoritmo RANSAC.

Para comenzar, dadas dos imágenes Im_i y Im_j se probaron las siguientes medidas de similitud:

$$Sim_1(Im_i, Im_j) = Matches_{ij}$$

$$Sim_2(Im_i, Im_j) = Matches_{ij} / \text{Max}(NumDesc_i, NumDesc_j)$$

Siendo $Matches_{ij}$ el número de correspondencias entre las imágenes i y j y $NumDesc_i$ el número de descriptores extraídos de la imagen i . Las Tablas C.1 y C.2 muestran los resultados obtenidos de utilizar descriptores SURF y las alternativas de búsqueda exhaustiva y aproximada respectivamente. La columna “Prof. de búsqueda” representa el nivel máximo explorado durante la búsqueda en árbol.

Las Tablas C.3 y C.4 reflejan el mismo experimento que las anteriores, pero en este caso se utilizan los descriptores ORB.

C. Experimentos sobre el sistema de reconocimiento 2D

Similitud	Tiempo de ejecución (seg)	% Aciertos
Sim_1	11.43	90.0
Sim_2	11.59	90.0

Tabla C.1: Resultados obtenidos utilizando descriptores SURF y búsqueda exhaustiva.

Similitud	Prof. de búsqueda	Tiempo de ejecución (seg)	% Aciertos
Sim_1	64	30.25	53.33
	32	11.02	53.33
	16	10.23	50.00
Sim_2	64	30.75	80.00
	32	11.12	76.67
	16	10.55	73.33

Tabla C.2: Resultados obtenidos utilizando descriptores SURF y búsqueda aproximada.

A la vista de los resultados, se decidió trabajar con los descriptores SURF, pues aunque son más lentos, muestran unos resultados mucho mejores que los descriptores ORB en estos experimentos iniciales.

Observando los resultados, también se aprecia que aunque la alternativa de búsqueda aproximada puede ser un poco más rápida que la búsqueda exhaustiva, en este caso la diferencia es mínima ya que el tamaño de la base de datos es pequeño y la ganancia en el tiempo de búsqueda se pierde construyendo el árbol de búsqueda, los resultados son significativamente peores. Con una base de datos superior la ganancia sería mayor, pero por otro lado los resultados empeorarían. Así pues, hasta que no se conozca con más detalle cómo va a integrarse el reconocimiento 2D dentro del sistema final, se decidió dejar la puerta abierta a los dos algoritmos.

Como última decisión, a partir de ese momento en los experimentos solo se utilizó Sim_2 como medida de similitud, pues ofrece un rendimiento mejor a la otra alternativa.

Experimento 2: RANSAC

El objetivo de este experimento es comprobar cómo afecta el algoritmo RANSAC al funcionamiento del sistema en entornos más reales que los del experimento anterior, es decir, donde el objeto a reconocer no está sobre fondo blanco, sino sobre una mesa o una estantería y rodeado de más objetos en la escena. Para ello el experimento se ha realizado probando 30 fotos de test de los distintos objetos de la base de datos, un objeto por foto, en escenarios con mucho ruido.

Las Tablas C.5 y C.6 muestran los resultados obtenidos utilizando descriptores SURF, la medida de similitud Sim_2 y las alternativas de búsqueda exhaustiva y aproximada respectivamente.

Observando estos resultados, resulta obvio que es necesaria una etapa de segmentación cuando se trabaja en entornos más reales, pues ninguna de las alternativas arroja unos resultados aceptables. Se aprecia también que ahora la ganancia de tiempo de cómputo de la búsqueda aproximada respecto a la exhaustiva es mayor. Esto es así porque ahora el

C. Experimentos sobre el sistema de reconocimiento 2D

Similitud	Tiempo de ejecución (seg)	% Aciertos
Sim_1	8.59	70.00
Sim_2	8.63	76.67

Tabla C.3: Resultados obtenidos utilizando descriptores ORB y búsqueda exhaustiva.

Similitud	Prof. de búsqueda	Tiempo de ejecución (seg)	% Aciertos
Sim_1	16	11.26	40.00
	8	8.33	40.00
	4	5.97	36.67
Sim_2	16	11.17	50.00
	8	8.02	50.00
	4	5.84	50.00

Tabla C.4: Resultados obtenidos utilizando descriptores ORB y búsqueda aproximada.

número de puntos SURF que encuentra el algoritmo en las imágenes de test es mayor, y por tanto comparar todos los descriptores entre sí conlleva una carga pesada. Otro aspecto que salta a la vista es la notable mejora de los resultados cuando se aplica el algoritmo RANSAC, aunque el tiempo de cómputo también aumenta considerablemente.

Teniendo todo esto en cuenta, se decidió que era necesario implementar una etapa de segmentación de la imagen en regiones de interés que parezcan tener un único objeto antes de intentar el reconocimiento del objeto. Esto permitirá evaluar si finalmente es necesario utilizar RANSAC o el hecho de segmentar regiones de la imagen en posibles objetos permite una robustez suficiente.

C.0.2. Experimentos utilizando bolsa de palabras

En los experimentos explicados en esta sección se utiliza como algoritmo de similitud la bolsa de palabras, previamente explicado. Las imágenes de entrada son las mismas que las utilizadas en el experimento 1 de la sección anterior con el algoritmo de búsqueda del vecino más cercano, es decir, 30 fotos de los objetos de la base de datos sobre fondo blanco. Las medidas de similitud utilizadas en los siguientes experimentos están basadas en el trabajo desarrollado en [15]

Experimento 1: comparación de histogramas

En este experimento se ha utilizado una medida de similitud basada en la comparación de los histogramas de palabras (Sección 2.3.2) de cada imagen. Dadas dos imágenes Im_i y Im_j

$$Sim(Im_i, Im_j) = |H_i - H_j|$$

Siendo H_i el histograma de palabras asociado a Im_i . En la Tabla C.7 se muestran los resultados obtenidos La columna “Número de clusters” indica cuantos clusters se emplearon en la creación del vocabulario.

C. Experimentos sobre el sistema de reconocimiento 2D

RANSAC	Tiempo de ejecución (seg)	% Aciertos
NO	33.25	33.33
SÍ	78.93	50.00

Tabla C.5: Resultados obtenidos utilizando descriptores SURF y búsqueda exhaustiva.

RANSAC	Prof. de búsqueda	Tiempo de ejecución (seg)	% Aciertos
NO	64	32.95	23.33
	32	20.06	20.00
	16	14.27	20.00
SÍ	64	33.95	36.67
	32	21.57	33.33
	16	16.60	30.00

Tabla C.6: Resultados obtenidos utilizando descriptores SURF y búsqueda aproximada.

Los resultados reflejan una gran disminución en el tiempo de cómputo respecto al algoritmo del vecino más cercano, pero un porcentaje de acierto muy por debajo de lo aceptable. La causa de esta pobre actuación es debida a que este algoritmo está orientado a trabajar con grandes bases de datos, donde trabajar un algoritmo como el de búsqueda del vecino cercano no sería posible, debido al coste computacional. Además, para poder obtener un vocabulario de palabras visuales representativo es necesario tener más información de referencia donde calcularlo. Por tanto, se decidió eliminar esta opción como alternativa para el sistema final.

Experimento 2: frecuencia de aparición

En este caso, la medida de similitud está basada en la frecuencia con la que aparece cada palabra w en cada imagen de referencia Im_i . Cada palabra que aparece en la imagen de consulta Im_c vota a aquellas imágenes de referencia donde aparecía con una frecuencia más similar. La imagen más similar será aquella con más votos. Más formalmente:

$$Votos(Im_i, w_j) = \begin{cases} H_c(j) & \text{si } [H_c(j) - H_i(j)] = \min([H_c(j) - H_1(j)], \dots, [H_c(j) - H_n(j)]) \\ 0 & \text{en cualquier otro caso} \end{cases}$$

$$Votos_i = \sum_{j=1}^n Votos(Im_i, w_j)$$

Donde $H_c(j)$ es el número de votos a la palabra j existente en la imagen de consulta. $H_i(j)$ es el número de votos a la palabra j existente en la imagen de referencia i . En la Tabla C.8 se muestran los resultados obtenidos.

Se observa que en este caso los resultados tampoco son suficientemente buenos. Aunque el porcentaje sea un poco mejor que en el experimento anterior, sigue estando muy por debajo de lo aceptable. Por tanto, esta medida de similitud también quedó eliminada, eliminando por extensión la opción del algoritmo de la bolsa de palabras como opción a integrar en el sistema final.

C. Experimentos sobre el sistema de reconocimiento 2D

Número de Clusters	Tiempo de ejecución (seg)	% Aciertos
200	2.88	20.00
300	2.99	23.33
500	3.06	23.33
750	3.55	33.33
1000	3.82	26.67

Tabla C.7: Resultados obtenidos mediante el algoritmo de bolsa de palabras.

Número de Clusters	Tiempo de ejecución (seg)	% Aciertos
200	2.93	23.33
300	3.01	26.67
500	3.25	26.67
750	3.58	33.33
1000	3.69	30.00

Tabla C.8: Resultados obtenidos mediante el algoritmo de bolsa de palabras.

Esto dejaba como única opción de algoritmo de similitud la búsqueda del vecino cercano, que fue la que finalmente se implantó en el sistema de reconocimiento con información 3D.

C. Experimentos sobre el sistema de reconocimiento 2D

Anexo D.

Experimentos sobre el sistema de final de reconocimiento 3D

En este anexo se presentan los experimentos adicionales llevados a cabo para identificar el rendimiento del sistema y que no aparecen en la Sección 4.3. Todos los experimentos se han ejecutado sobre una base de datos de 42 objetos, la mayoría de ellos fotografiados personalmente, aunque también se incluye algún modelo obtenido del *dataset* [23]. Como medida del rendimiento, en todos los experimentos se han utilizado los coeficientes *precision* y *recall*, definidos en la Ecuación 4.1

Experimento 1: objetos con forma diferenciable

El objetivo de este experimento es comprobar como se comporta el sistema cuando trata de identificar objetos de forma fácilmente diferenciables, es decir, no hay ningún objeto en la base de datos que tenga una forma parecida al objeto a reconocer. Para ello el experimento se ha dividido en dos partes, una para objetos con textura y otra para los objetos sin textura, ya que todas las combinaciones de descriptores no son válidas para los objetos sin textura. Para la parte con textura, se han reconocido 56 objetos repartidos en 20 fotos, mientras que para la parte sin textura se han reconocido 42 objetos repartidos en 15 fotos. En las Tablas D.1 y D.2 se muestran los resultados obtenidos.

Descriptores	Tiempo de ejecución (seg)	Precision	Recall
forma	132.407	0.678	1.0
forma+color	132.956	0.411	1.0
forma+surf	135.842	0.786	1.0
forma+surf+color	135.928	0.750	1.0

Tabla D.1: Resultados obtenidos con objetos diferenciables con textura.

Los resultados muestran que el añadir reconocimiento por SURF y por color apenas incrementa el tiempo de ejecución respecto al reconocimiento básico con el descriptor de información 3D, gracias al uso de la lista de candidatos. También se observa que en el caso de objetos diferenciables, añadir el descriptor de color perjudica el rendimiento. Esto último se debe a que estos objetos poseen varios colores, que dificultan notablemente su

D. Experimentos sobre el sistema de reconocimiento 3D

Descriptores	Tiempo de ejecución (seg)	Precision	Recall
forma	101.084	0.730	0.844
forma+color	101.106	0.541	0.800

Tabla D.2: Resultados obtenidos con objetos diferenciables sin textura.

reconocimiento por color.

Experimento 2: objetos con forma parecida

En este caso, se quiere comprobar el rendimiento del sistema cuando se trata de reconocer objetos que tienen forma parecida a otros objetos almacenados en la base de datos, es decir, casos en los que el descriptor de forma no es del todo discriminante. Estos experimentos también se han dividido en objetos con y sin textura, reconociendo para el primer caso 55 objetos repartidos en 20 fotos y 39 objetos repartidos en 15 fotos para el segundo. Las Tablas D.3 y D.4 muestran los resultados.

Descriptores	Tiempo de ejecución (seg)	Precision	Recall
forma	115.930	0.489	1.0
forma+color	119.791	0.511	1.0
forma+surf	118.681	0.709	1.0
forma+surf+color	120.120	0.727	1.0

Tabla D.3: Resultados obtenidos con objetos de forma parecida con textura.

Descriptores	Tiempo de ejecución (seg)	Precision	Recall
forma	75.583	0.821	1.0
forma+color	75.823	0.744	1.0

Tabla D.4: Resultados obtenidos con objetos de forma parecida sin textura.

A raíz de los resultados, se observa que en este caso el descriptor de color ya no supone una pérdida tan abultada como con los objetos de forma diferenciable, incluso mejora el rendimiento en los objetos con textura. También se aprecia la importante ganancia de aplicar el reconocimiento por SURF en objetos con textura, donde esto mejora notablemente el rendimiento.

Experimento 3: objetos con forma igual

Este experimento pretende comprobar el rendimiento del sistema en casos donde se quieren reconocer objetos cuya forma es igual, o extremadamente parecida, a otros objetos de la base de datos. Esto implica que el descriptor de forma es nada o muy poco discriminante, por lo que los demás descriptores deberán hacer la mayor parte del

D. Experimentos sobre el sistema de reconocimiento 3D

trabajo. También se ha dividido el experimento en objetos con y sin textura, reconociendo 53 objetos repartidos en 20 fotos en el primer caso y 55 objetos repartidos en 20 fotos en el segundo. Las Tablas D.5 y D.6 muestran los resultados.

Descriptores	Tiempo de ejecución (seg)	Precision	Recall
forma	154.171	0.377	1.0
forma+color	154.755	0.340	1.0
forma+surf	161.148	0.717	1.0
forma+surf+color	161.814	0.717	1.0

Tabla D.5: Resultados obtenidos con objetos de forma igual con textura.

Descriptores	Tiempo de ejecución (seg)	Precision	Recall
forma	64.638	0.363	1.0
forma+color	64.805	0.927	1.0

Tabla D.6: Resultados obtenidos con objetos de forma igual sin textura.

Estos resultados prueban que el empleo de descriptores de diferente naturaleza es una idea acertada, pues en este caso y como era de esperar los resultados obtenidos solamente con el descriptor de forma son muy bajos, mientras que aplicando también descriptores SURF en los objetos con textura y color en los objetos sin textura se logra un rendimiento mucho mayor. También se aprecia en este caso un aumento del tiempo de ejecución respecto a los demás experimentos en el caso de objetos con textura. Esto se debe a que la lista de candidatos formada para cada objeto a reconocer es más abultada, pues el descriptor de forma no discriminará entre todos los objetos de la base de datos que tengan la misma forma.

Experimento 4: objetos con oclusiones

En este caso, se quiere comprobar el rendimiento del sistema cuando se trata de reconocer objetos que presentan oclusiones, es decir, no se puede ver el objeto completo, bien porque está tapado por otro objeto o porque está cortado por los límites de la escena. Los objetos a reconocer son variados, por lo que solo se ha utilizado la combinación de descriptores “forma+surf+color” que es la que mejores resultados arrojaba, véase la Sección 4.3. Se han reconocido 53 objetos repartidos en 12 fotos. La Tabla D.7 muestra el resultado.

Descriptores	Tiempo de ejecución (seg)	Precision	Recall
forma+surf+color	80.685	0.745	0.950

Tabla D.7: Resultados obtenidos con objetos con oclusiones.

Los resultados muestran que el sistema mantiene unos resultados aceptables incluso con objetos con oclusión, algo muy deseable en sistemas de reconocimiento, pues no

D. Experimentos sobre el sistema de reconocimiento 3D

siempre se puede asegurar que los objetos que se quieran reconocer vayan a aparecer completos en entornos reales de ejecución.

Anexo E.

Diagrama de módulos del sistema de reconocimiento

La Figura E.1 muestra el diagrama de los módulos del sistema de reconocimiento desarrollado. Dichos módulos se describen a continuación.

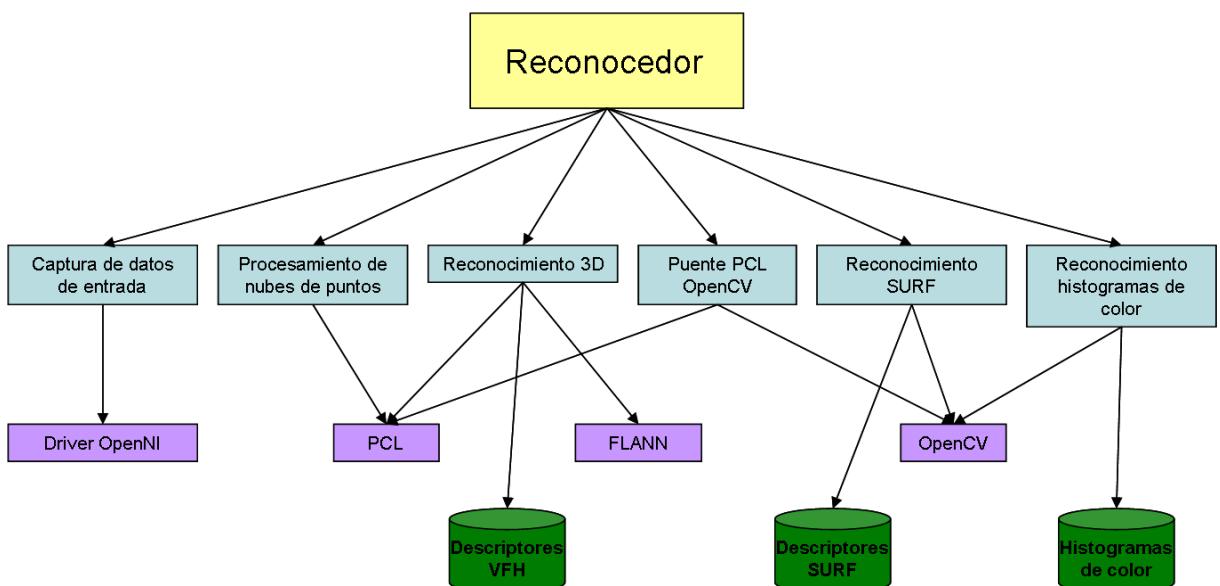


Figura E.1: Diagrama de módulos del sistema de reconocimiento desarrollado.

Reconocedor. Este es el módulo principal del sistema. Se encarga de realizar las llamadas a las funciones de todos los módulos implementados y de decidir con ayuda de las diferentes medidas de similitud calculadas por los módulos cuáles son los objetos a reconocer.

Captura de datos de entrada. La función de este módulo es conseguir un flujo de entrada de datos en forma de nube de puntos a través del sensor *kinect*. El módulo utiliza el driver OpenNI para comunicarse con el sensor.

Procesamiento de nubes de puntos. Este módulo se encarga de segmentar las nubes de puntos de entrada. Entre sus funciones están la de recortar zonas de la nube

E. Diagrama de módulos del sistema de reconocimiento

de puntos que no son de interés, por ser zonas demasiado lejanas, eliminar planos que pertenezcan a paredes, suelo, mesas...y agrupar los puntos en *clusters* que representan los puntos de la nube que pertenecen a un mismo objeto. El módulo utiliza funciones implementadas en la librería PCL.

Reconocimiento 3D. Este módulo es el encargado de todo lo relacionado con el reconocimiento utilizando información 3D. Sus funciones son extraer los descriptores VFH de los *clusters*, compararlos con los descriptores almacenados en la base de datos de objetos a través de una búsqueda en un *kd-tree*, crear una lista de objetos candidatos y calcular una medida de similitud para cada objeto de la lista. El módulo utiliza funciones de la librería PCL para extraer los descriptores VFH y la librería FLANN para realizar la búsqueda.

Puente PCL OpenCV. La función de este módulo es transformar la información 3D de las nubes de puntos, el tipo de datos con el que trabaja la librería PCL, en imágenes convencionales adecuadas para aplicar las técnicas de reconocimiento 2D, el tipo de datos con el que trabaja la librería OpenCV.

Reconocimiento SURF. Ese módulo se encarga de todo lo relacionado con el reconocimiento por descriptores SURF. Sus funciones son extraer los descriptores SURF de la imagen mínima que engloba el *cluster* a reconocer, compararlos con los descriptores SURF de los objetos de la lista de candidatos y calcular una medida de similitud con cada objeto de la lista. El módulo utiliza funciones de la librería OPENCV para extraer los descriptores SURF.

Reconocimiento histogramas de color. Este módulo se encarga de todo lo relacionado con el reconocimiento por histogramas de color. Sus funciones son calcular el histograma de color del *cluster* a reconocer, compararlo con los histogramas de los objetos de la lista de candidatos y calcular una medida de similitud con cada objeto de la lista. El módulo utiliza funciones de la librería OpenCV para calcular el espacio de color HSV a partir de la información RGB y para calcular la distancia EMD entre dos histogramas.

Anexo F.

Información de los objetos de la base de datos

La Tabla F.1 muestra los nombres de todos los objetos utilizados para realizar los experimentos que denotan el rendimiento del sistema. La mayoría de ellos se capturaron durante el desarrollo del proyecto, pero algunos pertenecen al *dataset* [23]. En la Tabla F.2 aparecen imágenes de un subconjunto de dichos objetos, solamente de los capturados durante la fase de adquisición de datos del proyecto. En la Tabla F.3 se muestran los nombres de todos los objetos del citado *dataset* utilizados en el experimento realizado para comparar el sistema desarrollado con otro de la literatura relacionada, mientras que en la Tabla F.4 aparece un subconjunto de imágenes de dichos objetos.

champú alisado	champú deliplus	champú fructis	champú squash	champú stylusc
champú azul	tazón	jarra	salero	manzana roja
banana	taza café 1	taza café 2	taza café 3	libro histories
libro delusion	coche	pelota amarilla	pelota azul	pelota roja
pelota verde	bota	bote tenis	bote azul	bote blanco
bote zaragoza	cinta vhs	cinta americana	caja zeppelin	caja gold
caja mmore	caja powerball	caja verbatim	caja comida	lata tomate
lata pepsi	lata seven up	lata comida	esponja	pasta dientes
linterna	gillette			

Tabla F.1: Nombres de todos los objetos capturados durante el desarrollo del proyecto. Para cada objeto capturado, se tomaron 24 fotografías.

F. Información de los objetos de la base de datos



Tabla F.2: Imágenes de objetos fotografiadas en la parte de adquisición de datos de este proyecto.

manzana roja	manzana amarilla	pelota fútbol	pelota rugby	banana
pimiento rojo	pimiento verde	bol 1	bol 2	bol 3
banana	taza café 1	taza café 2	taza café 3	taza café 4
gorra 1	gorra 2	cereales 1	cereales 2	esponja 1
esponja 2	linterna 1	linterna 2	caja comida 1	caja comida 2
caja comida 3	lata comida 1	lata comida 2	lata comida 3	tarrina comida
bote comida 1	bote comida 2	verduras 1	verduras 2	fideos instantáneos
pañuelos	limón	bombilla 1	bombilla 2	rotulador 1
rotulador 2	naranja	melocotón	pera	jarra 1
jarra 2	jarra 3	patata	lata pepsi	lata seven up
tomate 1	tomate 2	pasta dientes 1	pasta dientes 2	champú azul
grapadora	botella agua 1	botella agua 2	botella agua 3	plato

Tabla F.3: Nombres de todos los objetos del *dataset* [23] empleados. Para cada objeto empleado, se utilizaron 24 fotografías.

F. Información de los objetos de la base de datos

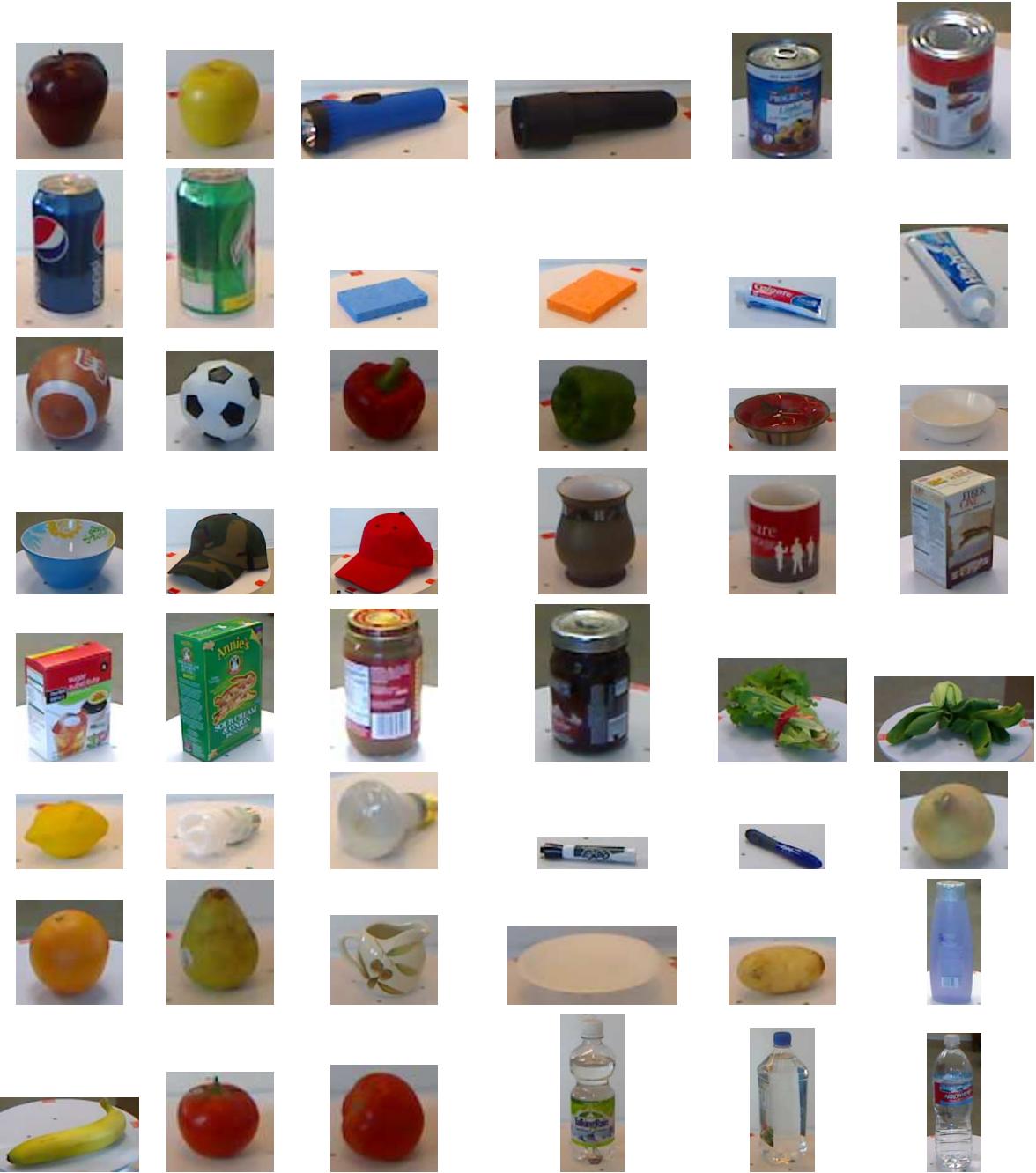


Tabla F.4: Imágenes de objetos obtenidas del *dataset* [23].