

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/304073891>

# Reconocimiento de Objetos 3D con Descriptores de Superficie

Chapter · June 2016

CITATIONS

0

READS

899

5 authors, including:



**Pablo Gil**

University of Alicante

136 PUBLICATIONS 1,140 CITATIONS

[SEE PROFILE](#)



**Carlos Mateo**

University of Burgundy

25 PUBLICATIONS 115 CITATIONS

[SEE PROFILE](#)



**Jorge Pomares**

University of Alicante

142 PUBLICATIONS 1,388 CITATIONS

[SEE PROFILE](#)



**Gabriel J. Garcia**

University of Alicante

26 PUBLICATIONS 209 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



MOMUE: MOBILE Manipulation for Unstructured outdoor Environments (funded by Ministry of Science, Innovation and Universities (MICINN)-RTI2018-094279-B-I00)

[View project](#)



COMMANDIA: COLlaborative robotic Mobile MANipulation of Deformable objects in Industrial Applications (funded by European Commission, Programa Interreg V- B Sudoe, Project Reference: SOE2/P1/F0638) [View project](#)

# Conceptos y Métodos en **VISIÓN POR COMPUTADOR**

---

Editado por Enrique Alegre,  
Gonzalo Pajares y Arturo de la Escalera



*Grupo de Visión del  
Comité Español de Automática (CEA)*



# **Conceptos y Métodos en Visión por Computador**

Editado por Enrique Alegre, Gonzalo Pajares y Arturo de la Escalera

Junio 2016

Conceptos y métodos en Visión por Computador

Varios autores

Editado por: Alegre Gutiérrez, Enrique; Pajares Martinsanz, Gonzalo; de la Escalera Hueso, Arturo

ISBN: 978-84-608-8933-5

Materias: Ingeniería de instrumentos e instrumentación, Ingeniería de instrumentos e instrumentación, Ingeniería de control automático, Procesamiento de imágenes

España, Junio 2016

11.3.2. Modelos de curvas y superficies implícitas .....	218
11.3.3. Modelos de curvas y superficies explícitas .....	221
11.3.4. Modelos CSG .....	223
11.3.5. Modelos de partición espacial .....	224
11.4. Aplicaciones .....	225
11.5. Bibliografía.....	228
<b>CAPÍTULO 12.....</b>	<b>229</b>
12.1. Introducción.....	230
12.1.1 ¿Cómo inferir información 3D a partir de una imagen 2D? .....	230
12.1.2 Sistemas comerciales y aplicaciones .....	230
12.2 Modelos de distorsión de la lente .....	232
12.3 Modelo proyectivo de la cámara .....	233
12.3.1. Parámetros extrínsecos del modelo de cámara (matriz T).....	234
12.3.2. Proyección de perspectiva (matriz P) .....	235
12.3.3. Parámetros intrínsecos del modelo de cámara (matriz K) .....	236
12.3.4 Calibración de cámaras.....	238
12.4 Homografía.....	241
12.4.1 Definición de homografías .....	241
12.4.2 Estimación de homografías .....	243
12.4. Bibliografía.....	244
<b>CAPÍTULO 13.....</b>	<b>247</b>
13.1. Introducción.....	247
13.2. Geometría de los sistemas estéreo .....	248
13.2.1. Matriz Fundamental .....	250
13.2.2. Matriz Esencial .....	255
13.2.3. Reconstrucción de la escena.....	256
13.2.4. Configuración de cámaras en paralelo .....	257
13.2.5. Rectificación de imágenes .....	258
13.3. Establecimiento de correspondencias .....	259
13.4. Mapas densos de disparidad.....	260
13.4.1. Métodos basados en correlación.....	260
13.4.2. Restricciones en los emparejamientos.....	261
13.4.3. Métodos basados en programación dinámica.....	262
13.5. Bibliografía .....	264
<b>CAPÍTULO 14.....</b>	<b>265</b>
14.1. Representación de una escena. La nube de puntos.....	265
14.1.1. El concepto de entorno de vecindad y similitud entre puntos de un entorno .....	267
14.1.2. Métricas de distancia entre puntos de una superficie .....	268
14.2. Procesamiento y extracción de características geométricas de superficie.....	269
14.2.1. Filtrado del ruido y puntos atípicos .....	269
14.2.2. Remuestreo de una nube de puntos .....	270
14.2.3. Vectores normales y curvatura de una superficie .....	271
14.2.4. Marco de Darboux .....	273
14.3. Descriptores geométricos de superficie.....	274
14.3.1. Histograma de características de tipo punto: PFH y FPFH .....	275
14.3.2. Histogramas de características de punto de vista: VFH y CVFH.....	277
14.3.3. Histogramas de orientación de superficies: SHOT.....	279
14.4. Aplicación al reconocimiento y clasificación de objetos 3D .....	281
14.5. Bibliografía.....	283
<b>CAPÍTULO 15.....</b>	<b>285</b>
15.1. Ajuste por mínimos cuadrados .....	285
15.1.1. Solución algebraica general .....	286
15.1.2. La regresión ortogonal de la recta .....	288
15.1.3. Ajuste a una circunferencia .....	289

# CAPÍTULO 14

## RECONOCIMIENTO DE OBJETOS 3D CON DESCRIPTORES DE SUPERFICIE

Pablo GIL<sup>1</sup>, Carlos M. MATEO<sup>2</sup>, Jorge POMARES<sup>1</sup>, Gabriel J. GARCIA<sup>1</sup>, Fernando TORRES<sup>1</sup>

<sup>1</sup> Depto. Física, Ingeniería de Sistemas y Teoría de la Señal, Universidad de Alicante, Alicante, España

<sup>2</sup> Instituto Universitario de Investigación Informática, Universidad de Alicante, Alicante, España

---

Este capítulo presenta un conjunto de descriptores basados en información de superficie para reconocer objetos asumiendo que se dispone de un modelo previo de éstos. Las superficies son representadas como nubes de puntos tridimensionales. De éstas se estudian características geométricas como la variación en dirección de los vectores normales y las curvaturas a lo largo de la nube de puntos que las definen, para así construir un descriptor matemático que identifique el tipo de objeto según su superficie. En este capítulo se proporcionan fundamentos geométricos de superficie, metodologías de filtrado para eliminación de ruido y puntos atípicos, características para describir una superficie, métricas de clasificación por distancia. Además de un conjunto de técnicas y métodos de procesamiento básicos que constituirán las etapas necesarias que se precisan implementar para identificar objetos 3D, sin la necesidad de considerar parámetros como la textura o el color.

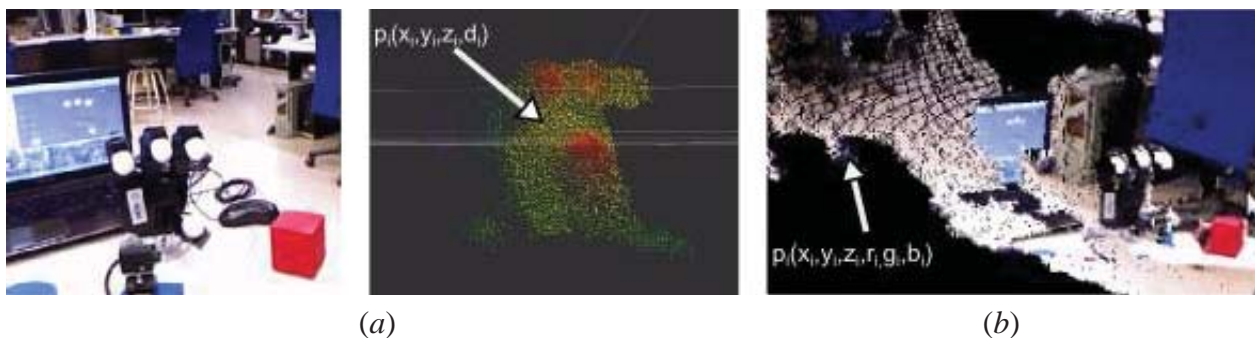
---

### 14.1. Representación de una escena. La nube de puntos

La estructura básica para representar una superficie en el Espacio Euclídeo se conoce como nube de puntos 3D. En la actualidad, existen una gran cantidad de sensores visuales que son capaces de adquirir información de profundidad de una escena real. Entre estos sensores han adquirido gran popularidad, las cámaras RGB-D y las cámaras de tiempo de vuelo (*'ToF: Time of Flight'* en el ámbito anglosajón) (Gil y col. 2014). Estos sensores se basan en principios físicos y tecnologías distintas, pero en ambos tipos es posible almacenar la información adquirida en forma de nubes de puntos, figura 14.1.



En su definición más sencilla, una nube de puntos  $\mathbf{P}$  es un conjunto de puntos 3D  $p_i(x_i, y_i, z_i)$  (Klette y col. 1998). El número de puntos que forman  $\mathbf{P}$  depende de la resolución y la precisión con la que la cámara es capaz de adquirir información del entorno. Generalmente, cada uno de los puntos  $p_i \in \mathbf{P}$  tiene un orden, ya que está referenciado respecto a un sistema de coordenadas fijo situado en el mismo sensor con el que se ha adquirido la información. Así, las coordenadas  $(x_i, y_i, z_i)$  de un punto  $p_i$  representan la posición que ocupa éste respecto al sensor con el que fue adquirido. Para que la cámara obtenga de una manera fiable las coordenadas de cada punto  $p_i$  requiere haber sido calibrada previamente. El proceso de calibración (Cyganek y Siebert 2009) permite obtener los parámetros intrínsecos y extrínsecos de un sensor.



**Figura 14.1. Ejemplo de nube de puntos adquirida desde una cámara: (a) ToF (b) RGBD**

Es habitual que las estructuras como nubes de puntos permitan almacenar información adicional además de la posición tridimensional de los puntos. Esta información adicional, frecuentemente, corresponde a las propiedades que los puntos tienen en la escena real. En estos casos, cada  $p_i$  tiene más de 3 dimensiones. Así, por ejemplo, si la nube de puntos está generada a partir de un sensor RGBD cada punto consta de 6 dimensiones  $(x_i, y_i, z_i, r_i, g_i, b_i)$  si se opta por almacenar la información de color. Esto se puede generalizar, ya que en una nube de puntos podrían almacenarse otras muchas propiedades de un punto de la escena como por ejemplo posición, color, geometría, etc. De modo que se podría considerar que cada punto  $p_i \in \mathbf{P}$  se puede representar como  $p_i(f_1, \dots, f_n)$  donde  $f_i$  es cada una de las características del punto y  $n$  el número de estas.

Las nubes de puntos son estructuras que por su tamaño y por las dimensiones de cada uno de sus elementos requieren, en muchos casos, de una gran cantidad de memoria. Además, su volumen de datos y su manera de organización incrementa los tiempos y costes computacionales, sobre todo cuando se aplican etapas de procesamiento para tratar y analizar los datos que contiene. Un ejemplo para ilustrar este problema sería la búsqueda del punto  $p_j$  en la escena más próximo a otro punto  $p_i$ , representados ambos en la nube de puntos  $\mathbf{P}$  como  $p_j$  y  $p_i$ , respectivamente. En la escena, la proximidad o lejanía entre dos puntos 3D cualesquiera se puede calcular con la distancia Euclídea, y además la geometría de la escena nos garantiza que, en el mundo, el punto más próximo  $P_j$  a un punto  $P_i$  escogido arbitrariamente siempre se encontrará en el entorno de vecindad de  $P_i$ . Este hecho no es trasladable a la nube de puntos, es decir, en  $\mathbf{P}$  estos dos elementos  $p_i$  y  $p_j$  pueden estar separados. De modo, que los puntos que están situados como vecinos  $p_i$  no tienen por qué ser los más próximos en distancia Euclídea en la escena. Esto hace que, por lo general, no se trabaje directamente sobre la nube de puntos y se opte por almacenar ésta como una estructura de datos más eficiente para hacer operaciones sobre ella. Las estructuras de datos más empleadas para trabajar con nubes son los árboles.



Es frecuente emplear estructuras de datos como ‘*kd-tree*’, ‘*octree*’ (Eberhard y col. 2010) y otras variantes para organizar y descomponer espacialmente  $\mathbf{P}$ .

Por otro lado, en ocasiones la nube de puntos  $\mathbf{P}$  determina la superficie de un objeto en vez de una escena más compleja, en estos casos es habitual que se empleen técnicas de reconstrucción de superficies. Por ejemplo se pueden emplear métodos de triangulación para construir una malla de triángulos que constituyan una aproximación poligonal de toda la superficie del objeto, u otros polígonos planos o formas geométricas.

#### 14.1.1. El concepto de entorno de vecindad y similitud entre puntos de un entorno

En una nube de puntos  $\mathbf{P}$  se puede definir un entorno de vecindad de un punto  $p_q \in \mathbf{P}$  con radio  $r$  como el conjunto de todos los puntos  $p_i \in \mathbf{P}$  tal que  $i \neq q$ :

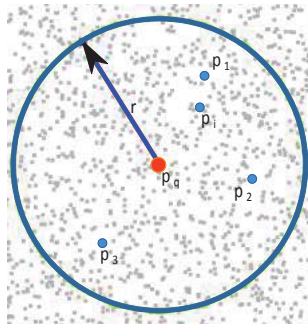
$$\|p_q \neq p_i\| \leq d_r \quad (14.1)$$

donde  $d_r$  es la distancia máxima que limita el radio especificado, figura 14.2. Para medir esa distancia, puede utilizarse distintas métricas, la más popular es la métrica  $L_2$  también conocida como distancia Euclídea.

Cuando se emplea una nube de puntos  $\mathbf{P}$  para representar un objeto o escena, que se quiere reconocer no basta con considerar  $\mathbf{P}$  como un conjunto de puntos con coordenadas Cartesianas  $(x, y, z)$ . En estos casos, es necesario tener en cuenta la relación existente entre los puntos que componen la nube. Así, si se toma un punto  $p_q \in \mathbf{P}$  y se fija un radio  $r$  que delimita su entorno de vecindad  $P^r$ , la relación entre él y sus vecinos  $p_i$  constituye un descriptor local en  $P^r$  que describe cómo varía la geometría y la forma de la superficie en  $p_q$  respecto a los puntos de su entorno  $P^r$ . La representación de esta información geométrica local se puede definir como un vector:

$$F_q(p_q, P^r) \neq \{f_1, f_2, \dots, f_n\} \quad (14.2)$$

donde  $f_i$  es la característica geométrica que relaciona  $p_q$  con cada  $p_i$ , siendo  $n$  el número de puntos en  $P^r$ .



**Figura 14.2. Ejemplo de entorno de vecindad de radio  $r$  en una nube de puntos**

Si se desea comparar dos puntos cualesquiera  $p_i$  y  $p_j$  de una nube de puntos  $\mathbf{P}$ , se procede tomando la información geométrica local de cada uno de ellos  $F_i$  y  $F_j$  en un entorno de vecindad  $P^r$  y comparando ésta. Generalmente, para llevar a cabo esta comparación se emplean métricas de distancia. Por consiguiente, es imprescindible escoger un entorno de vecindad adecuado para obtener una

representación del entorno del punto lo más significativa posible. La idea principal subyace en escoger el radio de modo que a partir de  $P^r$  sea posible obtener un vector de información geométrica local que no cambie (sea invariante) en presencia de transformaciones de cuerpos rígidos sólidos (rotaciones y traslaciones en el espacio Euclídeo, es decir en 6 grados de libertad), variación en la densidad de  $P^r$  (número de puntos que contiene) y ruido en los datos debido al proceso de adquisición (cierta falta de precisión a la hora de determinar las coordenadas Cartesianas de cada punto desde el sensor).

Por un lado, como ya se ha comentado con anterioridad, no existe una relación directa entre el concepto de proximidad entre dos puntos en la escena y entre esos dos puntos tal y como han sido registrados en  $\mathbf{P}$ . En otras palabras, puntos cercanos en la escena no se garantiza que se hayan registrado cercanos en la nube de puntos. Esto implica que, si se quieren obtener los puntos cercanos a un punto  $p_q$  en  $\mathbf{P}$ , se requiere calcular una métrica de distancia entre él y todos los puntos de la nube, además este proceso es costoso computacionalmente. Por otro lado, el proceso de registro de  $\mathbf{P}$  puede introducir ruido, y hacer que el vecino más cercano a  $p_q$  no sea aquel que minimice la distancia. Esto a veces es más frecuente cuanto más densa es  $\mathbf{P}$ . En estos casos, se aplican mecanismos de estimación para la aproximación en la búsqueda de los puntos  $p_i$  que pertenezcan al conjunto de vecinos más cercanos a  $p_q$ . Este mecanismo trabaja en función de un parámetro de error permitido  $\varepsilon$  que determina la calidad de la solución proporcionada, de la siguiente manera:

$$\|p_q \neq p_i\| \leq (1 \neq \varepsilon) \|p_q \neq p^*\| \quad (14.3)$$

donde  $p^*$  sería el punto real más cercano. Así, la ecuación 14.3 determina para cualquier punto  $p_i$  en un entorno de vecindad  $P^r$ , que la relación entre su distancia a  $p_q$  y la distancia de  $p_q$  al vecino real más cercano denotado por  $p^*$  debe ser de al menos  $\varepsilon + 1$ .

Como consecuencia de todo esto, queda patente que hay dos parámetros que el usuario requiere definir cuando se trata de trabajar y procesar nubes de puntos: el radio  $r$  (define el tamaño a considerar como entorno de vecindad  $P^r$ ) y el número  $n$  de puntos en ese radio  $r$ . Así, es posible fijar el radio en función del número mínimo de puntos que se desea que contenga  $P^r$ . Más adelante se mostrará cómo influyen valores grandes o pequeños de  $r$  o  $n$  a la hora de calcular el vector normal en un punto en función del tamaño y densidad de su entorno de vecindad.

### 14.1.2. Métricas de distancia entre puntos de una superficie

En particular, en un entorno de vecindad  $P^r$ , y en general, en una nube de puntos  $\mathbf{P}$ , se hace imprescindible definir una métrica para medir la cercanía o lejanía de un punto de la superficie a otro cualquiera. Entre las métricas de distancia más comunes se encuentran: *Manhattan*, Euclídea,  $\chi$ -Cuadrado y Núcleo de Intersección de Histograma (*HIK: Histogram Intersection Kernel* en el ámbito anglosajón).

Sean  $\bar{v}$  y  $\bar{u}$  dos vectores de  $n$  dimensiones en un espacio vectorial, donde ambos están referenciados respecto al mismo sistema de coordenadas fijo.

Se define la distancia *Manhattan* o distancia L1, como:

$$d_{L1} \neq \|\bar{v} \neq \bar{u}\|_{L1} \neq \sum_{i=1}^n |v_i \neq u_i| \quad (14.4)$$

La distancia Euclídea o distancia L2, se calcula como:

$$d_{L2} \neq \|\bar{v} \neq \bar{u}\|_{L2} \neq \sqrt{\sum_{i=1}^n (v_i \neq u_i)^2} \quad (14.5)$$

Se define la distancia  $\chi$ -Cuadrado, como:

$$d_{\chi^2} \neq \|\bar{v} \neq \bar{u}\|_{\chi^2} \neq \sum_{i=1}^n \frac{(v_i \neq u_i)^2}{v_i \neq u_i} \quad (14.6)$$

Y finalmente, la distancia *HIK*, como:

$$d_{HIK} \neq \|\bar{v} \neq \bar{u}\|_{HIK} \neq \sum_{i=1}^n \min\{v_i, u_i\} \quad (14.7)$$

Los puntos en una nube **P** se pueden considerar vectores de 3 o más dimensiones, tal y como se mostró previamente, dependiendo de si tienen únicamente información de posición espacial o incorporan información adicional como color, geometría, etc.

## 14.2. Procesamiento y extracción de características geométricas de superficie

### 14.2.1. Filtrado del ruido y puntos atípicos

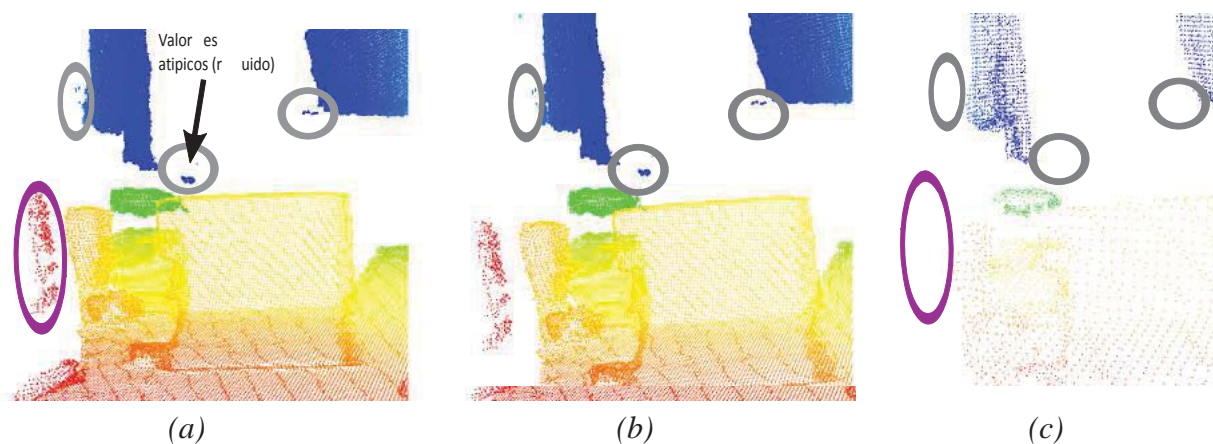
Cuando las nubes de puntos **P** han sido registradas a partir de un proceso de adquisición de una cámara RGBD o ToF, frecuentemente, incorporan puntos con ruido. El ruido de un punto dado  $p_q$  es la variación aleatoria de su posición  $(x_q, y_q, z_q)$ . Este concepto es extensible a cada una de las  $n$  dimensiones del punto  $p_q(f_1, \dots, f_n)$  de **P**. Así, si el punto es representado por un vector de 6 dimensiones  $(x_q, y_q, z_q, r_q, g_q, b_q)$ , el ruido sería la variación aleatoria en posición y color. Generalmente, se asume que si la nube de puntos se genera de manera sintética a partir de programas de modelado CAD, el problema del ruido desaparece. Principalmente, el ruido en cámaras RGBD o ToF se genera cuando la escena tiene alta reflectancia (superficies metálicas pulidas) o cuando la escena es compleja y hay varios objetos, en cuyo caso se producen transiciones entre superficies de objetos distintos que están situadas a distintas profundidades. Las transiciones se dan en los bordes donde hay salto de superficie o zona de oclusión por solapamiento y se caracterizan por estar representadas en **P** con bajo nivel de densidad de puntos. Una primera etapa de procesamiento puede consistir en eliminar de **P** todos aquellos puntos atípicos que están en zonas de transición o salto de superficie. De este modo, sólo se considerarían los puntos de la nube que están influenciados en menor

medida por el ruido del proceso de adquisición. Otra aproximación para eliminar puntos atípicos de la nube, consiste en llevar a cabo un análisis estadístico de puntos en sus entornos de vecindad. Primero, para cada punto  $p_q$  de  $\mathbf{P}$ , considerar un entorno de vecindad  $P^r$  y calcular la distancia media  $\mu_d$  y la varianza en distancia  $\sigma_d$  a sus  $n$  vecinos dentro del entorno de vecindad de radio  $r$ . Segundo, eliminar todos aquellos puntos cuya distancia está alejada de la distancia media del conjunto de puntos de  $P^r$ . Al igual que en la búsqueda del vecino más cercano, ecuación (14.3), es necesario definir un parámetro de tolerancia, similar a  $\varepsilon$ . Este parámetro delimitará el nivel de densidad en  $P^r$ , eliminando todos aquellos puntos que no cumplan la restricción de distancia media. Por lo tanto, el ruido siempre influirá negativamente si dado un punto  $p_q$  se desea calcular la relación existente con sus vecinos en un entorno  $P^r$ . Consecuentemente, el ruido afectará a la representación geométrica de la superficie local  $F_q(p_q, P^r)$ .

### 14.2.2. Remuestreo de una nube de puntos

El proceso de remuestreo de una nube de puntos se suele emplear cuando después de filtrar una nube de puntos  $\mathbf{P}$ , el resultado es otra nube de puntos  $\mathbf{P}'$  cuya densidad global de puntos es prácticamente la misma a la que tenía la nube sin filtrar. El proceso de remuestreo tiene como objetivo eliminar las imperfecciones, suavizando la superficie y descartando todos aquellos puntos que generan anomalías como picos o similares (concavidades abruptas), o rellenando pequeños agujeros (generados por falta de puntos debido a convexidades abruptas) en la superficie, figura 14.3.

El remuestreo se basa en técnicas de interpolación mediante optimización matemática y se emplea comúnmente para eliminar todos aquellos puntos atípicos de la nube. La técnica de optimización más sencilla es la conocida como Mínimos Cuadrados Promedio (*LMS: Least Mean Square*) (Trucco, 1998). Esta técnica busca una función continua que aproxime un conjunto de datos ordenados y minimice el error cuadrático medio. El método *LMS* requiere de un gran número de iteraciones para converger y obtener la función que aproxima y minimiza el error. Además, en este método se asume que el ruido de cada punto debe estar distribuido de forma aleatoria para que la aproximación sea correcta.



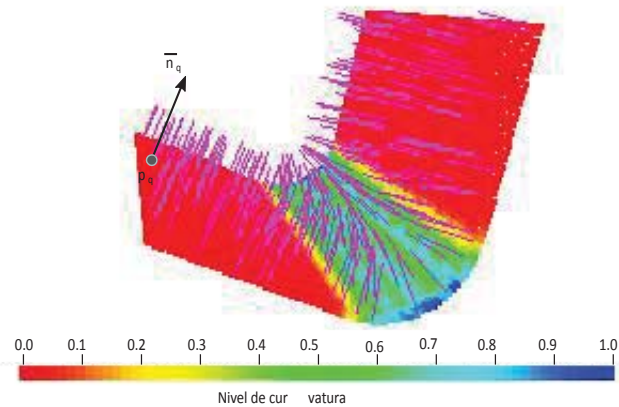
**Figura 14.3. Ejemplo de filtrado y remuestreo. (a) Original (b) Nube de puntos eliminando ruido (c) Remuestreada eliminando puntos por suavizado de la superficie**

Una variación de este tipo de ajuste es el método de Mínimos Cuadrados Ponderados o Mínimos Cuadrados Móviles (*'MLS: Moving Least Square'*) (Lancaster y Salkanskas 1981), que consiste en dar un peso a cada punto de la nube en función de que se considere que éste tenga más o menos error de medida. El método de MLS permite ajustar una superficie desde un conjunto de puntos de muestra no organizados o estructurados, asegurando que la superficie pasa por los puntos de la muestra. Para ello, mediante la asignación de pesos se da mayor o menor importancia a cada uno de los puntos de la muestra. Así, se pondera con mayor peso aquellos puntos de la muestra que son más representativos de la superficie.

En ocasiones, la nube de puntos  $\mathbf{P}$  contiene superficies que no son completamente suaves (es decir con cambios no bruscos de dirección entre sus puntos), y que no son debidos a errores o anomalías en la etapa de registro, sino que son puntos que caracterizan el tipo de superficie, por ejemplo objetos que tienen regiones angulares e interesa preservar estas zonas angulosas sin que se consideren valores atípicos. Un objeto con forma de esfera se representa con una nube de puntos de superficie suave, mientras que un cubo presenta en la superficie cambios bruscos en la zona de aristas. En estos casos, suele emplearse para ajustar superficies planas RANSAC (*'RANSAC: RANdom SAmple Consensus'*) (Trucco y Verri, 1998) o alguna de sus variantes como RMSAC (*'Randomized M-Estimator SAmple Consensus'*). Los métodos RANSAC y sus variantes proporcionan métodos de ajuste más robustos que mínimos cuadrados cuando hay muchos puntos atípicos, es decir datos con ruido, que pueden interferir en el ajuste. Además, puede interesar rellenar pequeños huecos, añadiendo puntos en aquellas áreas de la superficie donde hay ausencia de estos. La manera de hacerlo escapa al objetivo de este capítulo.

### 14.2.3. Vectores normales y curvatura de una superficie

Los conceptos de vector normal  $\bar{n}$  y de curvatura  $c$  de una superficie han sido ampliamente usados, en aplicaciones de visión por computador, para proporcionar datos de medida de la superficie y definir propiedades geométricas de ésta (Klette y col. 1998). El término de curvatura, menos conocido que el de vector normal, tiene su origen en la geometría diferencial de curvas. En geometría diferencial, una curva es una región de puntos donde existe una misma variación de superficie. Y la manera más sencilla de medir esta variación es calculando los cambios de orientación de los vectores normales en dicha superficie. La medida de esa variación se conoce con el término de parámetro de curvatura en visión por computador, figura 14.4.



**Figura 14.4. Ejemplo de cálculo de vectores normales y curvaturas. El color indica el valor de curvatura y las flechas los vectores normales**

El cálculo de estas propiedades geométricas,  $\bar{n}$  y  $c$  en un punto  $p_q$  de  $\mathbf{P}$  es muy dependiente del entorno de vecindad  $P^r$  que se considera para dicho punto. Es decir, en el cálculo del vector normal y de la curvatura en  $p_q$  influyen tanto el radio como el número de puntos que contiene  $P^r$ . Así, si hay pocos puntos dentro del entorno de vecindad o si el radio es muy pequeño, el cálculo de estas propiedades geométricas se llevará a cabo con inexactitud. Por el contrario, muchos puntos o un radio muy amplio distorsionan los valores obtenidos dificultando la detección de cambios de orientación de los vectores normales y, como consecuencia, se genera falta de precisión en el cálculo de la curvatura. En este último caso, el valor de curvatura tiende a igualarse como si se realizará un suavizado.

Tal y como se ha comentado en la sección 14.1, si dada una nube de puntos  $\mathbf{P}$  se desea almacenar información geométrica además de la posición cartesiana, el punto se denotaría por  $p_q(f_1, f_2, f_3)$  y se requeriría trabajar en un entorno de vecindad  $P^r$  definido por el usuario. En el caso que nos ocupa,  $f_1$  es una tupla que representa la posición cartesiana de cada punto  $(x_i, y_i, z_i)$ ,  $f_2$  representa el vector normal a la superficie en dicho punto, y  $f_3$  el parámetro de variación en la superficie o curvatura en  $P^r$ .

Una manera geométrica sencilla de calcular el vector normal a una superficie  $P^r$  en un punto  $p_q(x_q, y_q, z_q)$  consiste en estimar el vector normal  $\bar{n}$  a un plano tangente a la superficie en dicho punto. El plano tangente a  $P^r$  en  $p_q$  se define matemáticamente como  $\Pi_q(x, y, z) = a(x - x_q) + b(y - y_q) + c(z - z_q) = 0$  y su vector normal es el vector gradiente al plano denotado por  $\bar{n} = \nabla \Pi(x, y, z) = \{a, b, c\}$ . El problema reside en que hay infinitos planos que pasan por  $p_q$  y se desea estimar el plano tangente a  $p_q$  que mejor aproxima todos los puntos de  $P^r$ , es decir, que se ajusta minimizando la distancia entre él y cada uno de los puntos de  $P^r$ . Por lo tanto, el problema se reduce a un problema de ajuste por mínimos cuadrados del sistema  $d_i = 0$  donde  $d_i$  representa la distancia entre  $p_i \in P^r$  y  $\Pi_q$ , y ésta se define como  $d_i = (p_i - p_q) \cdot \bar{n}$ .

Para obtener el vector normal  $\bar{n}$  en  $p_q$  se procede realizando un análisis de valores y vectores propios de la matriz de covarianza  $C_q$  de todos los puntos  $p_i \in P^r$  (Mateo y col 2015). La matriz de covarianza  $C_q$  se define aplicando Análisis de Componentes Principales ('PCA: Principle Component Analysis') (Cyganek y Siebert 2009) del siguiente modo:



$$C_q \neq PP^T \neq \begin{bmatrix} p_1 \neq \bar{p} \\ \dots \\ p_k \neq \bar{p} \end{bmatrix} \begin{bmatrix} p_1 \neq \bar{p} \\ \dots \\ p_k \neq \bar{p} \end{bmatrix}^T \quad (14.8)$$

donde  $\bar{p}$  es el centroide de  $P^r$  y  $k$  su número de puntos. Y resolviendo la ecuación (14.8) por descomposición en valores singulares ('SVD: Singular Value Descomposition') (Trucco y Verri 1998) se obtienen los valores propios  $\lambda_j$  y vectores propios  $\bar{v}_j$  de  $C_q$ :

$$C \cdot \bar{v}_j \neq \lambda_j \cdot \bar{v}_j \quad (14.9)$$

La suma  $\lambda_0 \neq \lambda_1 \neq \lambda_2$  de los valores propios proporciona un parámetro que describe la variación local de superficie que se detecta entre los puntos  $p_i \in P^r$  y su centroide  $\bar{p}$ . El valor propio más pequeño  $\lambda_0$  proporciona información sobre la variación a lo largo del vector normal  $\bar{n}$ , y su vector propio  $\bar{v}_0$  corresponde a una estimación de  $\bar{n} \neq \{a, b, c\} \neq \{n_x, n_y, n_z\}$ , siendo desconocido si el sentido del vector normal es positivo o negativo. Para resolver esta ambigüedad se requiere conocer el punto de vista con el que fue adquirida la nube de puntos  $P$ . El vector normal se puede representar en coordenadas esféricas como:

$$\phi \neq \text{atan} \left( \frac{n_z}{n_y} \right), \theta \neq \text{atan} \left( \frac{\sqrt{(n_z^2 + n_y^2)}}{n_x} \right) \quad (14.10)$$

Hay varios estudios que permiten modelar el parámetro de curvatura cuando la superficie se modela con estructuras poligonales triangulares y no como nubes de puntos. No obstante, la gran mayoría de ellos calculan la curvatura a partir de puntos de vértice. Este es el caso de la curvatura Gaussiana. Para el caso que nos ocupa, una manera de estimar la curvatura máxima local a  $P^r$  en un punto  $p_q$  es el método de Pauly (Pauly y col. 2002) que tiene la ventaja de que es invariante a escala y donde la curvatura puede ser calculada a partir de los valores propios de la matriz de covarianza  $C_q$  del siguiente modo:

$$c_p \neq \frac{\lambda_0}{\lambda_0 \neq \lambda_1 \neq \lambda_2} \quad (14.11)$$

Un valor pequeño de  $c_p$  indica que no hay variación en la superficie y por lo tanto, el punto yace sobre el plano tangente a la superficie.

#### 14.2.4. Marco de Darboux

Cuando se trabaja con nubes de puntos, conviene calcular la relación geométrica que existe entre cada uno de los puntos de la nube y sus vecinos. En estos casos, el marco de Darboux constituye una herramienta matemática que permite calcular esta relación. Así, el marco de Darboux es una herramienta matemática de la geometría diferencial de superficies orientadas, que define tres vectores para representar matemáticamente una curva de la superficie. Se entiende como curva al lugar geométrico en la superficie de las distintas posiciones que ocupan los puntos en el espacio de acuerdo a



una trayectoria dada. Los tres vectores que definen la curva según Darboux son el vector normal, el vector tangente y el producto vectorial de éstos.

Si se traslada este concepto a un punto  $p_q$  de un entorno de vecindad de la superficie representada por  $P^r$  y se considera una curva sobre ella, el marco de Darboux se define como  $\{\bar{u}, \bar{v}, \bar{w}\} \neq \{\bar{n}_q, \bar{t}_q, \bar{n}_q \wedge \bar{t}_q\}$  donde  $\bar{n}_q$  es el vector normal unitario a la superficie en  $p_q$  y  $\bar{t}_q$  es el vector tangente a la curva en  $p_q$ , figura 14.5. Así, el marco de Darboux define un sistema de referencia, como una base ortonormal en cada punto de la curva que representa un marco móvil natural para medir variaciones a lo largo de una curva. Este sistema de referencia está relacionado con la regla de la mano derecha, donde  $\bar{u}$  representa el vector que apunta hacia nosotros,  $\bar{w}$  está girado  $90^\circ$  en sentido horario desde el vector  $\bar{v}$ .

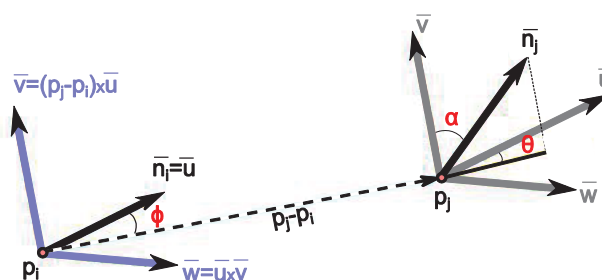


Figura 14.5. Variación geométrica entre dos puntos en función del marco de Darboux

### 14.3. Descriptores geométricos de superficie

En ocasiones, una nube de puntos puede representar un objeto 3D aislado. En procesos de reconocimiento de objetos 3D, es común crear descriptores como conjuntos de propiedades que los identifiquen, sección 14.4. Estos descriptores permiten realizar procesos de comparación de dos o más objetos mediante un análisis de semejanza de las propiedades de los objetos. En este apartado se pretende ilustrar algunos descriptores que ayudan a representar propiedades geométricas de la superficie de un objeto.

Los valores de los descriptores geométricos de un objeto 3D dependen de cómo se encuentran representadas la superficie o volumen de éste después de ser registrado por el sistema sensorial de adquisición. Las representaciones más comunes para representar un objeto 3D son: una nube de puntos de su superficie; una malla ‘*mesh*’ de su superficie calculada a partir de la aproximación poligonal de la nube de puntos; y un volumen compuesto por ‘*voxels*’ (cada voxel representa un pequeño volumen cúbico que envuelve un punto de la nube) (Cyganek y Siebert 2009). En este capítulo, los descriptores que se presentan se obtendrán siempre a partir de la representación de una nube de puntos, y éstos se pueden agrupar en tres grandes tipos:

- Basados en características de puntos de la nube y su entorno de vecindad.
- Basados en características de punto de vista.
- Basados en orientación de superficie.

Estos descriptores hacen uso de conceptos básicos anteriormente comentados, tales como entorno de vecindad, vector normal, curvatura o marco de Darboux, para describir la geometría de la superficie de un objeto representada como una nube de puntos.

### 14.3.1. Histograma de características de tipo punto: PFH y FPFH

El histograma de características de tipo punto (*PFH: Point Feature Histogram*) es un descriptor que hace uso de las características de puntos escogidos  $p_q$  de la superficie  $\mathbf{P}$  a describir (Rusu, 2013). Los puntos escogidos representan su geometría en función de la relación a sus puntos vecinos  $p_i$  en cierto entorno de vecindad  $P^r$ . La información geométrica de  $p_q$  se obtiene midiendo la variación entre las características de  $p_q$  y los  $p_i$  en ese entorno de vecindad.

En primer lugar, para poder calcular PFH conviene definir el concepto de entorno de vecindad de doble radio. Así, para un punto cualquiera  $p_q \in \mathbf{P}$  se pueden definir dos radios  $r_1$  y  $r_2$  con  $r_1 \leq r_2$  que determinan dos entornos de vecindad concéntricos  $P^{r_1}$  y  $P^{r_2}$ . Cada uno de los dos entornos de vecindad proporcionan una representación distinta de  $p_q$ . De este modo, PFH permite codificar dos tipos de información por cada punto candidato. Por un lado, se calcula el vector normal a la superficie en  $p_q$  haciendo uso de PCA (ver sección 14.2.3) en el entorno  $P^{r_1}$ ; para ello se emplea la ecuación (14.10). Por otro lado, PFH calcula el valor de curvatura media alrededor de  $p_q$  como una codificación de las propiedades geométricas en el entorno  $P^{r_2}$  según la ecuación (14.11). Después, PFH codifica las posibles variaciones geométricas de superficie considerando los cambios de dirección de los vectores normales estimados en  $P^{r_2}$ .

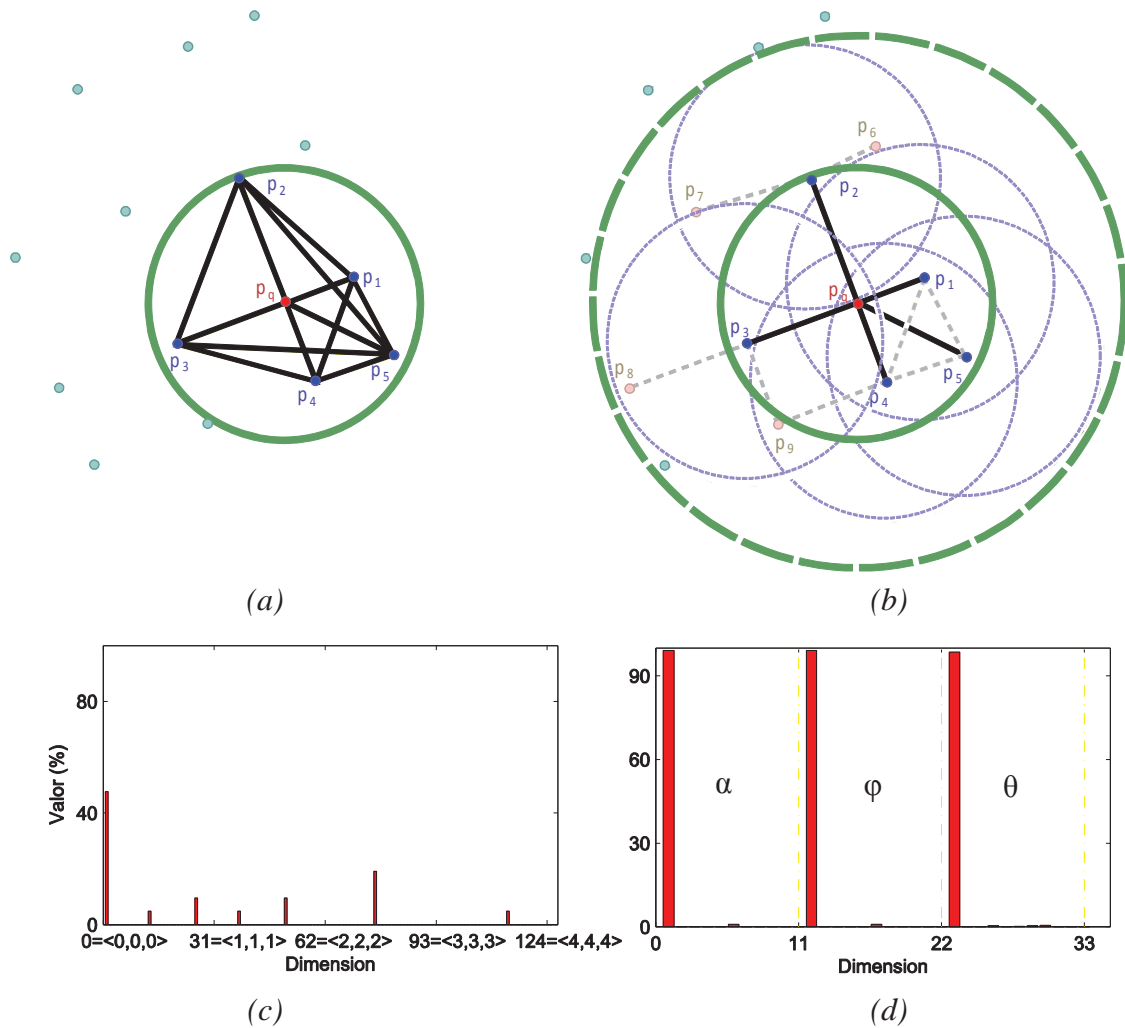
A continuación, se ilustra cómo se lleva a cabo el cálculo de la variación geométrica entre dos puntos cualesquiera  $p_i$  y  $p_j$  de  $P^{r_2}$  que se encuentran separados una distancia Euclídea  $d_{L_2}$ . Para ello, se parte de que previamente ya se han calculado todos los vectores normales para todos los puntos de  $P^{r_2}$  y que se ha definido un sistema de referencia fijo (Marco de Darboux) en uno de los puntos, por ejemplo  $p_i$ . Este marco de referencia fijado en  $p_i$  queda definido por  $\{\bar{u}, \bar{v}, \bar{w}\} \neq \{\bar{n}_i, \bar{t}_i, \bar{n}_i \wedge \bar{t}_i\}$  tal como se presentó en la Sección 14.2.4, de la siguiente manera:

$$\bar{u} \neq \bar{n}_i, \quad \bar{v} = \bar{t}_i = \bar{u} \wedge \frac{(p_j \neq p_i)}{d_{L_2}}, \quad \bar{w} \neq \bar{u} \wedge \bar{v} \neq \bar{n}_i \wedge \bar{t}_i \quad (14.12)$$

De modo que la variación geométrica entre ambos puntos se puede expresar como la diferencia relativa entre la dirección de los vectores normales  $\bar{n}_i$  y  $\bar{n}_j$  de dichos puntos (ver Figura 14.5), y se calcula como:

$$\alpha \neq \text{acos}(\bar{v} \cdot \bar{n}_j), \quad \phi \neq \text{acos}\left(\bar{u} \cdot \frac{(p_j \neq p_i)}{d_{L_2}}\right), \quad \theta \neq \text{atan}(\bar{w} \cdot \bar{n}_j, \bar{u} \cdot \bar{n}_j) \quad (14.13)$$

En el algoritmo de cálculo del descriptor PFH se calcula la tupla  $(\alpha, \phi, \theta, d_{L_2})$  para cualquier par de puntos  $p_i$  y  $p_j$  de  $P^{r_2}$ . Como consecuencia, si el entorno de vecindad  $P^{r_2}$  contiene  $k$  número de puntos, entonces la representación PFH de ese entorno estará formada por  $k^2$  tuplas  $(\alpha, \phi, \theta, d_{L_2})$ , figura 14.6a. Por simplicidad, de ahora en adelante  $d_{L_2}$  será llamado  $d$ .



**Figura 14.6.** (a) Entorno de vecindad  $P^r^2$  para un punto candidato  $p_q$  y relación con sus vecinos, (b) Influencia de vecinos en FPFH para  $p_q$ , (c) Firma del descriptor PFH de  $p_q$  como una conjunto de tuplas  $(\alpha, \phi, \theta)$  con 125 codificaciones, (d) Firma del descriptor FPFH de  $p_q$  con 33 codificaciones.

El descriptor PFH de un punto candidato  $p_q$  se puede representar como un histograma multidimensional invariante a transformaciones espaciales (6 grados de libertad: traslación y rotación) de la superficie que describe. Por lo general, un histograma unidimensional representa el número de veces que se repite un valor. Por lo tanto, aplicando el mismo concepto, un histograma como PFH representa el número de veces que se repiten los valores de una tupla  $(\alpha, \phi, \theta)$ , figura 14.6c. Además, como cualquier histograma unidimensional, es necesario dividirlo en un número determinado de subdivisiones. Cada una de esas subdivisiones representa un rango de valores de cada elemento de la tupla y gráficamente, indican el número de ocurrencias en cada intervalo de valores, figura 14.6. Es conveniente que haya correlación entre valores de la tupla y, para conseguirlo, conviene crear el mismo número de subdivisiones para cada elemento de la tupla. Puesto que tres de los valores son angulares, el criterio más usado es emplear la máxima diferencia angular para determinar el máximo número de subdivisiones a crear. El descriptor PFH se comporta mejor que el descriptor FPFH frente a la presencia de ruido y con nubes de puntos cuyos entornos de vecindad varían sus densidades. Además, es posible reducir la complejidad de PFH despreciando la componente  $d$ , quedando la tupla tal que  $(\alpha, \phi, \theta)$ . Esto es frecuente cuando se trabaja con nubes de puntos que no representan distancia

en la escena real, si no que la información almacenada en distancia es una distancia relativa entre objetos en la nube y no en la escena. En este caso se dice que la nube de puntos contiene información  $2^{1/2}$ D y no 3D.

El principal problema de PFH es el coste computacional que supone trabajar con todos los puntos pertenecientes al entorno de vecindad. Una solución para reducir la complejidad de PFH manteniendo su poder de descripción, que nos va a permitir discriminar superficies distintas, es la variante FPFH (*Fast PFH*). El cálculo del descriptor FPFH (Rusu, 2013) se hace en dos pasos.

Inicialmente, se procede de modo similar a como se ha hecho en FPH. Así, para cada punto candidato  $p_q$  con  $P^{r2}$  que contiene  $k$  puntos  $p_i$ , se calcula el conjunto de todas las tuplas  $(\alpha, \phi, \theta)$  de  $p_q$  a  $p_i$  según la ecuación (14.13). El histograma multidimensional resultante se llamará SPFH (*Simply PFH*). Posteriormente, para cada punto  $p_i$  vecino de  $p_q$  también se considera su propio entorno de vecindad  $P^{r2}$ , figura 14.6b, y se vuelve a calcular SPFH, pero esta vez, los valores se ponderan por un peso en función de la distancia de cada  $p_q$  a  $p_i$ , de acuerdo a:

$$FPFH(p_q) \neq SPFH(p_q) \neq \frac{1}{k} \sum_{i=1}^k \frac{1}{\omega_i} SPFH(p_i) \quad (14.14)$$

De este modo, habrá puntos vecinos  $p_i$  que contribuyen más a la geometría de  $p_q$  que otros de sus vecinos, como se observa en la figura 14.6b. A diferencia de PFH, FPFH no considera todas las relaciones entre puntos dentro de  $P^{r2}$  para determinar la geometría, sólo las relaciones entre  $p_q$  y  $p_i$ . Además, sí que considera puntos vecinos fuera de  $P^{r2}$ , aunque la influencia de estos será menor debido a la ponderación por proximidad a  $p_q$ .

En general, PFH trabaja con un histograma codificado de los valores angulares de la tupla  $(\alpha, \phi, \theta)$ , el número de posibles combinaciones es 125. Esto es debido a que cada uno de los valores angulares de la tupla se encuentra clasificado en 5 subdivisiones (una cada  $72^\circ$ ). En la figura 14.6c, la 46ª subdivisión indica que  $(\alpha, \phi, \theta) \neq (1,4,1) \neq 1 \times 1 \neq 4 \times 5 \neq 1 \times 25 \neq 46$ , o lo que es lo mismo que los valores angulares son  $\alpha \neq 72^\circ, \phi \neq 288^\circ, \theta \neq 72^\circ$ . FPFH trabaja con 11 subdivisiones (cada  $32^\circ$ ) por elemento de la tupla. No obstante, para eliminar subdivisiones donde no hay ocurrencias (número de veces que se repite el valor de la tupla es 0), se emplean 3 histogramas independientes concatenados, uno para cada elemento de la tupla, de modo que se dan 33 posibles combinaciones en vez de 1331, figura 14.6d.

### 14.3.2. Histogramas de características de punto de vista: VFH y CVFH

El histograma VFH (*VFH: View Feature Histogram*) (Rusu y col. 2010; Rusu, 2013) se crea como una extensión de FPFH, ya que éste es robusto al ruido y a la escala pero no es invariante al punto de vista de la escena. Si se recuerda, tanto PFH como FPFH codifican los ángulos de variación (cabeceo, alabeo y guiño) entre los vectores normales de dos puntos cualesquiera en una región de la superficie mediante un marco de Darboux (Figura 14.5), y éste se emplea para determinar la forma geométrica de la superficie. Esta misma idea se vuelve a emplear en VFH que, además, incorpora información para codificar la dirección del punto de vista. De este modo es posible emplear VFH tanto para procesos de

reconocimiento de objetos, como para procesos en los que es necesario calcular su posición (*'pose'* en el ámbito anglosajón). Así, VFH consta de un descriptor de punto de vista y un descriptor de forma. El descriptor de forma se calcula como en FPFH, añadiendo la componente  $d$ , de modo que en esta sección sólo se comenta cómo se calcula la componente de punto de vista de VFH.

Para definir el punto de vista, se calcula el centroide de la superficie  $p_c$  para fijar un sistema de referencia en ese punto. Así, la dirección de referencia se escoge como un vector normal a la superficie  $\bar{n}_c$  en el centroide  $p_c$ . De modo que el punto de vista se codifica como un histograma que representa la desviación angular entre  $\bar{n}_c$  y el vector normal  $\bar{n}_j$  en cada punto  $p_j$ , de modo similar a como se hizo en FPH y FPFH. Es decir, se asume que  $p_i \neq p_c$  en la ecuación (14.12) para definir el marco de Darboux. Y posteriormente, se mide la variación angular de cabeceo, alabeo y guiño entre el punto de referencia del objeto  $p_c$  y cada uno de los puntos  $p_j$  de la superficie, como la desviación angular entre los vectores normales  $\bar{n}_j$  a la superficie y la dirección de referencia de punto de vista definida por  $\bar{n}_c$ , de acuerdo a la ecuación (14.13).

El descriptor CVFH (*'CVFH: Clustered Viewpoint Feature Histogram'*) (Aldoma y col. 2012) se crea como una extensión de VFH para solventar algunos de los problemas de éste último. Entre los inconvenientes que presenta VFH cabe mencionar que no permite determinar la localización de un objeto cuando hay giros de la escena sobre el eje de punto de vista de la cámara (eje z). Este hecho genera ambigüedad si se desea estimar la *'pose'* del objeto. Además, VFH presenta deficiencias en el reconocimiento de objetos reales porque es sensible a la pérdida de información. Es decir, la estimación de centroides y normales se ve afectada por tres factores: la presencia de oclusiones en la superficie del objeto (el objeto se observa parcialmente y no se registran todos los puntos de la superficie); los procesos de segmentación para identificar sólo los puntos de objetos (eliminar otros puntos de la escena); y los puntos que yacen en aristas del objeto que suelen ser registrados con mucho ruido por el sensor RGBD o ToF. CVFH tiene una gran ventaja frente a VFH, y es que no considera aquellas partes de la nube de puntos que se sabe que no se han registrado de manera robusta por el sensor, como son las aristas del objeto.

CVFH divide la nube de puntos  $\mathbf{P}$  que representa el objeto en regiones estables *'clusters'*  $\mathbf{P} \neq \{P_1, P_2, \dots, P_n\}$ . Cada subconjunto (*'cluster'*)  $P_k$  representa una porción del objeto cuyos puntos han sido registrados con poco ruido. Para ello, se eliminan todos aquellos puntos que tienen un valor de curvatura elevado. Los puntos con curvatura grande suelen corresponder con puntos atípicos producidos por ruido o con puntos en las proximidades de aristas del objeto. Para dividir  $\mathbf{P}$  en regiones  $P_k$  estables, se busca agrupar puntos  $p_i$  en función de la dirección de sus vectores normales. Es decir, se busca construir agrupaciones de puntos  $p_{ki}$  en regiones  $P_k$  que haga máxima la variación entre los valores medios de la dirección del vector normal de todos los *clusters*  $\{P_1, P_2, \dots, P_n\}$  y a su vez hagan mínima la variación en dirección de todos los puntos  $p_{ki}$  de un mismo conjunto  $P_k$ . Así, a modo de ejemplo, un  $p_i$  se incluirá como un punto de  $P_k$  y pasaría a etiquetarse como  $p_{ki}$ , si dado un punto  $p_{kj} \in P_k$  se cumpliera que ambos son próximos, es decir por debajo de un umbral  $\|p_i - p_{kj}\| < d$ , además de que las orientaciones de sus vectores normales son parecidas, es decir  $\bar{n}_i \cdot \bar{n}_{kj} > n$ . Una vez se tienen las regiones estables  $P_k$  se procede a calcular los descriptores VFH para cada una de ellas. Primero, se estima el centroide y la normal de éste para definir el punto de vista de cada región. Y

después, se calculan las desviaciones angulares para el conjunto de cada punto  $p_{ki}$  con respecto al centroide.

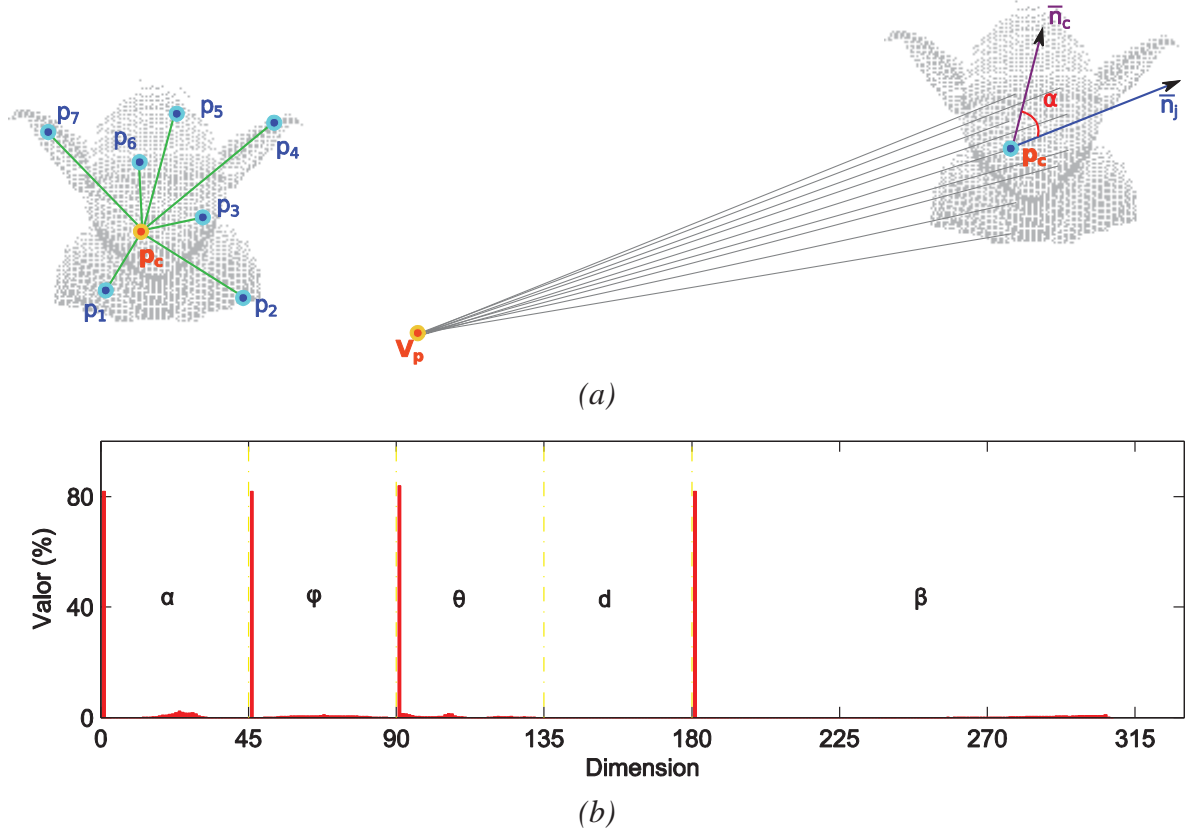


Figura 14.7. *a)* Descriptor de punto de vista VFH, *(b)* Firma del descriptor VFH/CVFH con las dos componentes encadenadas: forma y punto de vista  $\beta$ .

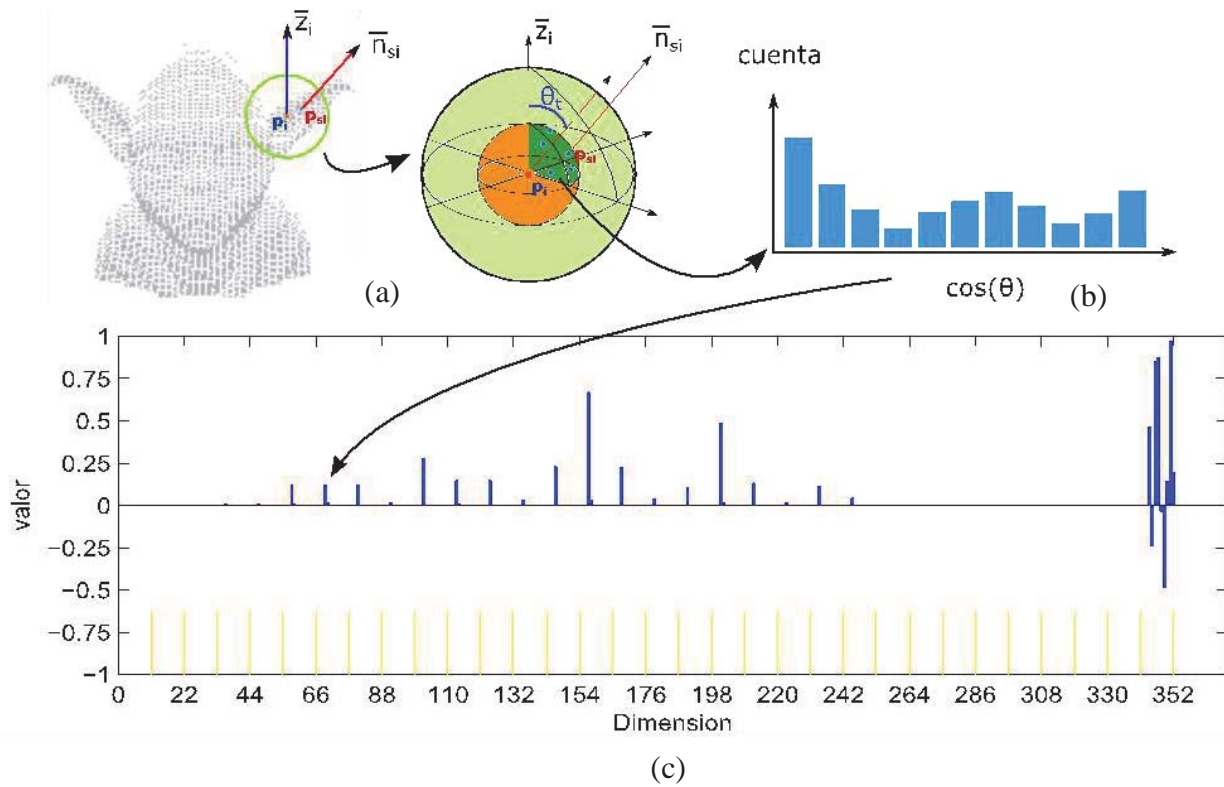
La representación del descriptor VFH trabaja con 45 subdivisiones para cada uno de los elementos de la tupla  $(\alpha, \phi, \theta, d)$ , en vez de 11 como FPFH y 128 codificaciones para el punto de vista  $\beta$ . En total 308 codificaciones para 5 variables distintas  $(\alpha, \phi, \theta, d, \beta)$ . La distribución de forma  $(\alpha, \phi, \theta, d)$  permite diferenciar superficies similares en tamaño y densidad pero cuyos puntos guardan otra distribución.

### 14.3.3. Histogramas de orientación de superficies: SHOT

El descriptor 3D, SHOT (*Signature of Histograms of Orientation*) (Salti y col. 2014) codifica dos tipos de información. Por un lado, codifica en forma de histograma la información geométrica de los puntos de la superficie. Estos histogramas almacenan el número de veces que se repiten cada uno de los valores geométricos de la superficie, tal y como se ha hecho en los descriptores anteriores, aunque en este caso la tupla de información geométrica es distinta y no es calculada usando el marco de referencia de Darboux. Por otro lado, y de manera conjunta, se codifica una firma que registra para cada valor geométrico, cuáles son las coordenadas 3D de los puntos de la nube que contribuyen con la misma variación geométrica. De este modo, y a diferencia del resto de descriptores, SHOT registra no sólo información de variación geométrica sino que también aporta información espacial de donde se presenta ésta en la nube de puntos.



El nuevo marco de referencia se define a partir de una rejilla esférica isotrópica que se divide en sectores mediante paralelos (divisiones en ángulo de acimut), meridianos (divisiones en ángulo de elevación) y radios. Lo más habitual, es usar una rejilla esférica que consta de 32 sectores formados a partir de 8 divisiones en acimut, 2 divisiones en elevación y 2 a lo largo del radio, figura 14.8a.



**Figura 14.8.** (a) Marco de referencia de SHOT en  $p_i$  (por simplicidad solo se han representado 4 secciones sobre acimut) (b) Histograma local de SHOT para una sección (c) Descriptor de SHOT en  $p_i$  para todas las regiones

Para un punto  $p_i$  cualquiera de  $\mathbf{P}$  sobre el que se desea calcular la geometría, se construye una esfera cuyo centroide se hace coincidir con  $p_i$ . Después, la esfera se orienta de modo que su ‘polo norte’ coincida con la dirección del vector normal a  $\bar{n}_i$  en  $p_i$ . De esta manera se ha fijado un nuevo marco de referencia donde el eje z de la esfera coincide con  $\bar{n}_i$  denotándose ahora como  $\bar{z}_i$ . Posteriormente, y haciendo uso de ese marco de referencia, se construye un histograma local, figura 14.8b, que define la relación de variación geométrica entre  $p_i$  y cada uno de los sectores en los que ha quedado dividida la rejilla esférica isotrópica, figura 14.8a. En total 32 histogramas locales muestran la variación geométrica entre el punto  $p_i$  y la rejilla esférica. Cada uno de los 32 histogramas locales determina el número de veces que se repite un cierto valor angular  $\theta_i$ . El ángulo  $\theta_i$  se calcula para cada punto  $p_{si}$  dentro de la sección de la esfera y representa la variación angular entre el vector normal al sector  $\bar{n}_{si}$  en ese punto, dentro del sector del marco de referencia y el eje z en el punto  $p_i$  de  $\mathbf{P}$ . Así, el ángulo  $\theta_i$  que permite construir el histograma local para cada sector de la rejilla se obtiene como:

$$\theta_i = \arccos(\bar{z}_i, \bar{n}_{si}) \quad (14.15)$$

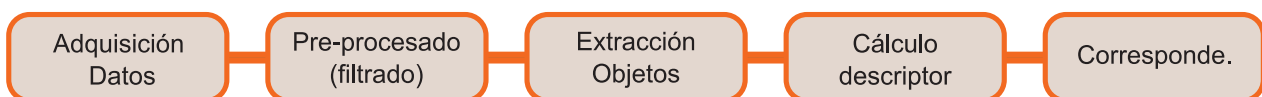


Y el histograma local representa el número de veces que se repite cada posible valor de  $\theta_i$ . La concatenación de los 32 histogramas locales de  $p_i$ , representa su descriptor SHOT, figura 14.8c.

#### 14.4. Aplicación al reconocimiento y clasificación de objetos 3D

En esta sección se presenta un esquema general para un método básico de reconocimiento y clasificación de objetos rígidos tridimensionales, como en Aldoma y col. (2014). El método propuesto asume que la escena ha sido capturada como una nube de puntos (ver sección 14.1). La escena además puede ser compleja o simple. Si es compleja por lo general representa varios objetos adquiridos desde un sensor RGBD o ToF, y si es simple, la escena representa un único objeto que se apoya sobre una mesa de trabajo.

El método de reconocimiento propuesto consta de cinco etapas de procesamiento básicas, figura 14.9. En la primera etapa ‘*Adquisición Datos*’ el sensor adquiere información de una escena que es almacenada como una nube de puntos. Tanto si la escena es simple como si es compleja, la nube de puntos que la representa debe ser filtrada. Por ejemplo, en una escena simple, se requiere aislar los puntos que representan el objeto del resto de puntos (mesa de trabajo, fondo, etc.). Si la escena es compleja, además, hay que dividir la nube de puntos en regiones, una para cada objeto. En este ejemplo, se asume que la escena es simple. Para detectar el objeto presente en la escena, primero se realiza una etapa de ‘*Pre-procesado*’, esta etapa consiste en aplicar filtros como los vistos en las secciones 14.2.1 y 14.2.2, de este modo se aíslan los puntos que determinan el objeto del resto de puntos de la escena. Así, se elimina el ruido y todos aquellos puntos demasiado cercanos o alejados del sensor. En un tercer paso, ‘*Extracción de objetos*’ se emplea el algoritmo de Trevor (Trevor y col. 2013) para encontrar el plano que ajusta todos aquellos puntos de la nube que forman parte de la mesa de trabajo (plano dominante). En el caso de que la nube represente objetos aislados sin fondo (por ejemplo, una representación CAD), algunas de estas etapas pueden no ser necesarias.



**Figura 14.9. Etapas generales del método de reconocimiento.**

Una vez que se dispone de una nube de puntos que representan un único objeto rígido aislado  $O_i$ , se calculan los vectores normales, como se ha comentado en la sección 14.2.3., y partir de éstos y empleando el marco de Darboux mostrado en la sección 14.2.4, se puede ‘*Calcular un descriptor*’ escogiendo cualquiera de los comentados en la sección 14.3. Además, si el objeto  $O_i$  quiere ser identificado y clasificado se requiere disponer de un modelo  $M_j$  con el que se desea comparar. Para que objeto y modelo puedan ser comparados, se requiere que la representación de ambos sea similar, de ahí que también  $M_j$  estará representado por una nube de puntos. Más aún, si la clasificación se quiere realizar correctamente, se necesita disponer de un conjunto amplio de representaciones de  $M_j$ , esto es, obtener diversas nubes de puntos que representen vistas de dicho modelo. Por ejemplo, si  $M_j$

es un modelo CAD de una forma geométrica se pueden aplicar distintas rotaciones, traslaciones y escalados para construir un conjunto de nubes de puntos que simulen los  $n$  distintos puntos de vista  $\mathbf{M}_j = \{m_{j1}, \dots, m_{jn}\}$ . Y para cada una de estas vistas se calcula un descriptor, de entre los vistos.

Generalmente, en cualquier proceso de reconocimiento se almacenan en una base de datos un conjunto de descriptores para cada uno de las distintas clases de objetos que se desea reconocer. Cada clase de objeto viene representada por un modelo  $\mathbf{M}_j$ . La figura 14.10 representa los objetos empleados en este ejemplo de aplicación.

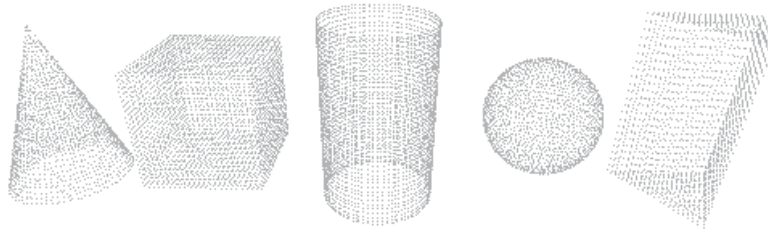


Figura 14.10. Modelos almacenados en la base de datos.

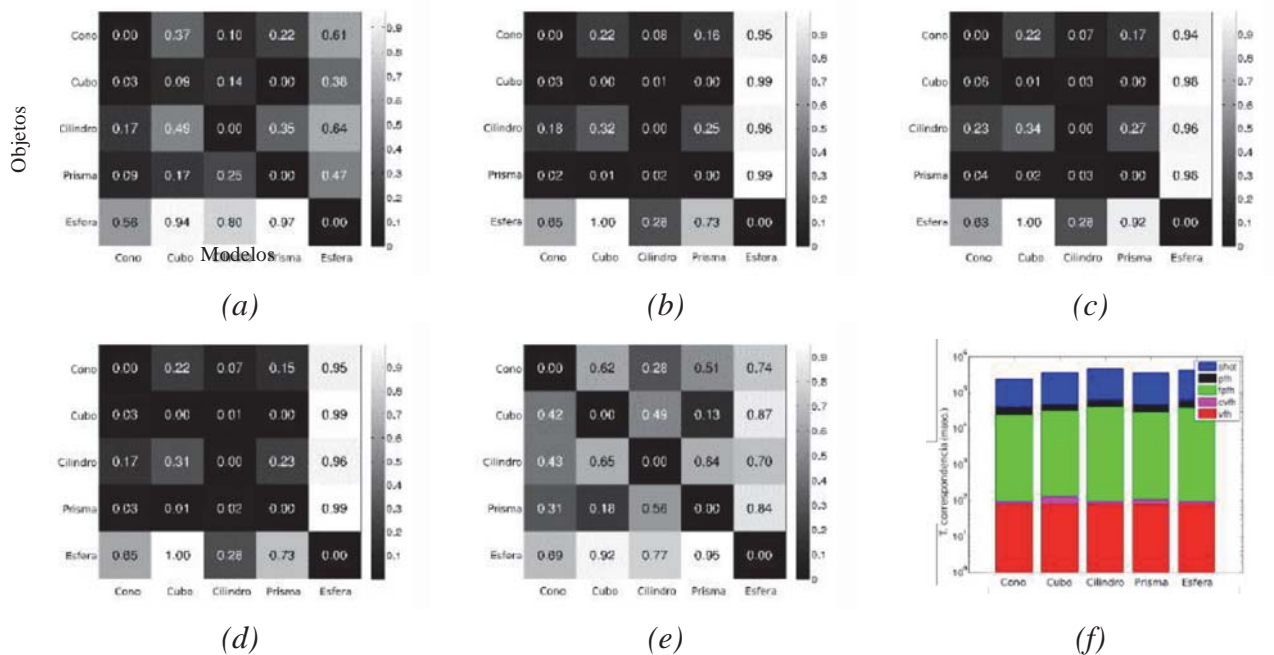


Figura 14.11. Distancias métricas calculadas en el proceso de correspondencia entre objeto y modelo para cada uno de los descriptores (a) PFH (b) FPFH (c) VFH (d) CVFH (e) SHOT (f) Comparativa de tiempos.

Finalmente, la clasificación se lleva a cabo mediante una etapa de ‘Correspondencia’ donde se compara el descriptor de cada una de las vistas  $\{m_{j1}, \dots, m_{jn}\}$  del modelo  $\mathbf{M}_j$  con la nube de puntos del objeto. Si desde el sensor se ha adquirido más de una vista del objeto, por ejemplo  $m$  vistas, se tendrá que comparar cada  $\mathbf{O}_i = \{o_{i1}, \dots, o_{im}\}$  con las vistas de los modelos  $\mathbf{M}_j$ , similar a lo que sucede en Mateo y col. (2015). Para compararlos, se emplean las distancias métricas vistas en 14.1.2 de acuerdo a la ecuación (14.15). Así, aplicando la ecuación (14.15) para cada uno de los descriptores, se obtienen las matrices de distancia, figura 14.11.

$$d(\mathbf{O}_i, \mathbf{M}_j) \neq \min\{d(o_k, m_r)\} \text{ donde } 1 \leq r \leq n \text{ y } 1 \leq k \leq m \quad (14.16)$$

## 14.5. Bibliografía

- Aldoma, A.; Marton, Z.C.; Tombari, F.; Wohlking, W.; Potthast, C.; Zeisl, B.; Rusu, R.B.; Gedikli, S.; Vincze, M. (2014) Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation. *IEEE Robotics & Automation Magazine*, vol. 1070(9932/12), pp. 80-91.
- Aldoma, A.; Tombari, F.; Rusu, R.; Vincze, M. (2012) OUR-CVFH – Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram for Object Recognition and 6DOF Pose Estimation. En *Pattern Recognition, Lecture Notes in Computer Science*; Springer-Verlag: Germany; Vol. 7476, pp.113-122.
- Cyganek, B.; Siebert, J.P.; (2009) An introduction to 3D Computer vision techniques and algorithms. *Editorial John Wiley & Sons*. ISBN: 978-0-470-01704-3.
- Eberhard, H.; Klumpp, V.; Hanebeck, U.B.; (2010) Density trees for efficient nonlinear state estimation. *13<sup>th</sup> Conference on Information Fusion (FUSION)*, pp. 1-8, ISBN: 978-0-9824438-1-1
- Gil, P.; Mateo, C.M.; Torres, F. (2014). 3D Visual Sensing of Human Hand for Remote Operation of a Robotic Hand. *International Journal of Advanced Robotics System*, vol. 11(26), pp. 1-13.
- Klette, R.; Schlüns, K.; Koschan, A.; (1998) Computer Vision: Three-dimensional data from images. *Springer*. ISBN: 981-3083-71-9.
- Lancaster, P.; Salkanskas K.; (1981) Surface Generated by Moving Least Squares Methods. *Mathematics of computation*, Vol. 37, No. 155, pp. 141-158.
- Mateo, C.M.; Gil, P., Torres, F. (2015) Analysis of Shapes to Measure Surface: an approach for detection of deformations. En *Actas de 12th Conference on Informatics in Control Automation and Robotics (ICINCO)*, Colmar, Francia, 21-23 Julio 2015, vol. 2, pp. 60-65.
- Mateo, C.M.; Gil, P.; Torres, F. (2015) Visual perception for the 3D recognition of geometric pieces in robotic manipulation. *The International Journal of Advanced Manufacturing*, doi: 10.1007/s00170-015-7708-8, pp. 1-15.
- Pauly, M.; Gross, M., Kobbelt, L.P.; (2002) Efficient Simplification of Point-Sampled Surfaces. En *Actas de IEEE Conference on Visualization* pp. 163-170
- Rusu, R. (2013) *Semantic 3D object maps for everyday robot manipulation*; STAR 85, Springer Series-Verlag, Germany, pp. 154–196.
- Rusu, R.; Bradski, G., Thibaux, R.; Hsu, J. (2010) Fast 3D recognition and pose using the Viewpoint Feature Histogram. En *Actas de IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 18-22 October 2010; pp. 2155-2161.
- Salti, S.; Tombari, F.; Di-Stefano, L. (2014) SHOT: Unique signatures of Surface and texture description. *Computer Vision and Image Understanding*, vol. 125, pp. 251-264.
- Trevor A, Rusu RB, Christensen HI (2013) Efficient organized point cloud segmentation with connected components. En *Actas de 3rd Workshop on Semantic Perception*

Trucco, E.; Verri, A.; (1998) Introductory techniques for 3D computer vision. *Prentice Hall*. ISBN: 0-13-261108-2.



*Grupo de Visión del  
Comité Español de Automática (CEA)*