

UNIVERSIDAD DE CANTABRIA

DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA,
INGENIERÍA DE SISTEMAS Y AUTOMÁTICA



TESIS DOCTORAL

CONTRIBUCIONES AL
ALINEAMIENTO DE NUBES DE PUNTOS 3D
PARA SU USO EN APLICACIONES DE
CAPTURA ROBOTIZADA DE OBJETOS

CARLOS TORRE FERRERO
INGENIERO INDUSTRIAL
2010

Agradecimientos

Quiero expresar mi sincero agradecimiento a:

- Dr. José Ramón Llata, mi director de tesis, por su apoyo, motivación y oportunos consejos durante la realización de este trabajo de investigación.
- Dr. Peter Sturm por darme la oportunidad de trabajar con el Perception Group del INRIA Rhône-Alpes en Grenoble (Francia) y por su guía al comienzo de esta tesis.
- El apoyo de los Proyectos PROFIT FIT-020100-2003-715 y CICYT DPI2006-15313.
- Mis compañeros del Grupo de Ingeniería de Control por su valiosa colaboración y por saber crear siempre un buen ambiente de trabajo.
- El personal de la Secretaría del Departamento por su inestimable labor.
- Y sobre todo, mi familia y amigos por su cariño, su continuo apoyo y su confianza en mí.

Índice general

1. Introducción	1
1.1. Descripción del problema y motivación	1
1.2. Organización de los capítulos de la tesis	3
2. Estado del Arte	5
2.1. Introducción	5
2.2. Adquisición de formas tridimensionales	6
2.2.1. Introducción	6
2.2.2. Método activo para medida de rango basado en triangulación	8
2.2.3. Método activo para medida de rango basado en tiempo de vuelo	15
2.3. Alineamiento 3D	18
2.3.1. Introducción	18
2.3.2. Alineamiento entre Pares de Vistas vs. Alineamiento Multivista	19
2.3.3. Clasificación y requisitos de los algoritmos de alineamiento 3D	20
2.4. Algoritmos de alineamiento aproximado	25
2.4.1. Basados en descriptores	25
2.4.2. Basados en correspondencia de curvas, estructuras y grafos	39
2.4.3. Otros métodos	43
2.5. Algoritmos de alineamiento fino	45
2.6. Selección de puntos de interés	46

2.7. Medidas de similitud	48
3. Descriptor CIRCON	51
3.1. Introducción	51
3.2. Notación	52
3.3. Construcción del Descriptor	54
3.3.1. Coordenadas Cilíndricas	54
3.3.2. Sistema de Coordenadas Local	55
3.3.3. Imagen Cíclica de Contornos Radiales	56
3.4. Área representada por los píxeles del descriptor	60
3.5. Aplicaciones	64
3.5.1. Compresión de los Datos 3D	65
3.5.2. Alineamiento 3D	66
3.5.3. Reconocimiento de Objetos	67
4. Sistema de Visión Basado en el descriptor CIRCON	69
4.1. Descripción del Sistema de Visión	69
4.2. Descriptor CIRCON Normalizado	73
5. Medida de Similitud	75
5.1. Clasificación de los píxeles	76
5.2. Distancia media en el Área Solapada	78
5.3. Distancias en el Área No Solapada	79
5.4. Distancia media entre imágenes	79
5.5. Medida basada en la distancia entre descriptores	80
5.6. Normalización de la Medida de Similitud	81
5.7. Cálculo de la Medida de Similitud	82
5.8. Matriz de rotación asociada a la medida de similitud	82
6. Búsqueda de Correspondencias	85
6.1. Introducción	85
6.2. Reducción de las nubes de puntos	85
6.3. Selección de Puntos de Interés	87
6.4. Algoritmo de Búsqueda de Correspondencias	90

6.4.1. Datos de entrada y parámetros	91
6.4.2. Descripción del algoritmo	92
6.5. Matriz de transformación asociada a una correspondencia	98
7. Algoritmo de Alineamiento 3D	103
7.1. Introducción	103
7.2. Datos de entrada y parámetros	103
7.3. Descripción del algoritmo	104
7.4. Evaluación del criterio de parada	108
7.5. Factor de Corrección del Solapamiento	113
8. Adaptación del Algoritmo ICP	115
9. Resultados experimentales	119
9.1. Introducción	119
9.2. Medida del error cometido en el alineamiento aproximado	120
9.3. Evaluación de la medida de similitud	121
9.4. Alineamiento de nubes de puntos	129
9.4.1. Nubes de puntos procedentes de bases de datos	129
9.4.2. Nubes de puntos de escáneres láser	138
9.4.3. Nubes de puntos de cámaras de tiempo de vuelo	141
9.5. Alineamiento 3D de nubes de puntos de dispositivos diferentes	144
9.6. Alineamiento 3D en escenas con mezcla de objetos	144
10. Algoritmo de Alineamiento 3D basado en Descomposición Wavelet	149
10.1. Introducción	149
10.2. Búsqueda de Correspondencias	149
10.3. Descomposición Wavelet	151
10.4. Estrategia multinivel	152
10.5. Detalles del algoritmo	154
10.5.1. Transformación de un Descriptor CIRCON	154
10.5.2. Medida de Similitud	156
10.5.3. Eliminación de las Correspondencias No Válidas	158

10.5.4. Verificación de las correspondencias	159
10.5.5. Alineamiento de las nubes de puntos	161
10.6. Resultados	166
11. Conclusiones, contribuciones y trabajo futuro	167
11.1. Conclusiones	167
11.2. Contribuciones	170
11.3. Trabajo futuro	172
A. Sistema de Coordenadas Local	175
B. Selección de parámetros de la medida de similitud	181
C. Implementación de la medida de similitud	187
C.1. Implementación para los n_s giros sobre la normal	187
C.2. Implementación para un número reducido de giros sobre la normal	190
C.3. Cálculo de la Medida de Similitud	191
C.4. Compatibilidad de dos correspondencias	192

Índice de figuras

2.1. Clasificación de los sistemas industriales de medida de rango 3D.	8
2.2. Sistema basado en la proyección de un punto luminoso.	9
2.3. Rejilla para proyección multipunto.	12
2.4. Sistema de visión 3D mediante proyección de una línea láser. . .	13
2.5. Ejemplo de patrones de proyección utilizando un código Gray de 3 bits.	14
2.6. Medida basada en tiempo de vuelo con láser pulsado.	16
2.7. Medida basada en desplazamiento de fase para sistemas de tiem- po de vuelo.	17
2.8. Correspondencia y alineamiento.	22
2.9. Construcción de la signatura de un punto.	29
2.10. Construcción de una <i>spin image</i>	31
2.11. Construcción de un histograma geométrico.	32
2.12. Construcción de una signatura de superficie.	33
2.13. Imágenes de forma armónica.	34
2.14. Conjunto de histogramas 2D propuesto por <i>Ankerst et al.</i> para descomponer el espacio 2D.	35
2.15. Concepto de representación <i>splash</i>	37
2.16. Descriptor basado en integración de volumen.	38
2.17. Curvas Bitangentes	39
2.18. Curvas intrínsecas.	40
2.19. Procedimiento de búsqueda de correspondencia y alineamiento de las superficies mediante el método RBD (RANSAC-based DARCES).	42

2.20. Emparejamiento global de puntos intrínsecos.	43
3.1. Coordenadas cilíndricas	54
3.2. Elección de la orientación de los ejes del sistema de coordenadas local q con respecto al sistema de coordenadas global W	56
3.3. Construcción del descriptor CIRCON.	57
3.4. Matriz CIRCON.	59
3.5. Ejemplo de imagen CIRCON.	60
3.6. Área representada por los píxeles de la imagen CIRCON.	61
3.7. Matriz de pesos del área de cada píxel de la imagen CIRCON.	63
3.8. Forma 3D simplificada.	66
4.1. Sistema de visión propuesto.	70
4.2. Estructura del sistema de visión 3D.	71
4.3. Imagen CIRCON de la carcasa de un alternador ($\rho_\theta = 2,5^0$, $\rho_r = \rho_z = 0,5mm$).	72
4.4. Normalización de la imagen CIRCON mostrada en la Figura 4.3.	73
5.1. Clasificación de los píxeles de las imágenes CIRCON en píxeles de <i>Objeto</i> y píxeles de <i>No-Objeto</i>	76
5.2. Conjuntos Intersección, OR-Exclusivo y Unión correspondientes a las dos imágenes mostradas en la Figura 5.1.	77
6.1. Reducción de la imagen de rango.	87
6.2. Reducción de la nube de puntos.	88
6.3. Selección de puntos de interés.	91
6.4. Ejemplo de punto de partida y punto objetivo del algoritmo de búsqueda de correspondencias.	93
6.5. Convergencia del algoritmo de búsqueda.	94
6.6. Mapas de similitud para las tres iteraciones de la Figura 6.5.	95
6.7. Alineamiento aproximado expresado en el s.d.c. local en el <i>punto objetivo</i>	99
6.8. Alineamiento aproximado expresado en el s.d.c. original de la nube 2.	101

7.1. Reducción del radio de búsqueda.	105
7.2. Ejemplo de refinamiento multirresolución.	107
7.3. Reducción del número de celdas de búsqueda.	109
7.4. Evaluación del criterio de parada.	110
7.5. Alineamiento aproximado obtenido por el algoritmo para el objeto 'ducky'.	112
8.1. Alineamiento fino obtenido mediante la adaptación del algoritmo ICP.	117
9.1. Imágenes de rango de cinco objetos.	122
9.2. Comportamiento del algoritmo para el objeto 'ducky' con tres medidas de similitud.	124
9.3. Comportamiento del algoritmo para el objeto 'bunny' con tres medidas de similitud.	125
9.4. Comportamiento del algoritmo para el objeto 'hip' con tres medidas de similitud.	126
9.5. Comportamiento del algoritmo para el objeto 'dragon' con tres medidas de similitud.	127
9.6. Comportamiento del algoritmo para el objeto 'porsche' con tres medidas de similitud.	128
9.7. Alineamiento para el objeto 'bunny'	130
9.8. Alineamiento para el objeto 'dragon'	131
9.9. Alineamiento para el objeto 'hip'	132
9.10. Alineamiento para el objeto 'porsche'	133
9.11. Alineamiento para el objeto 'frog'.	134
9.12. Imágenes de rango de diez objetos.	135
9.13. Alineamiento para el objeto 'foot'.	136
9.14. Resultados obtenidos para cuatro objetos del estudio comparativo de Salvi-Matabosch [89]: 'fractal', 'wave', 'frog' y 'owl'. Los errores cometidos se muestran en la Tabla 9.2.	137
9.15. Escáner láser utilizado para adquirir las imágenes de rango.	138

9.16. Alineamiento de dos imágenes de rango de un colector de escape adquiridas mediante un escáner láser.	139
9.17. Alineamiento de dos imágenes de rango de un cubo de rueda adquiridas mediante un escáner láser.	140
9.18. Cámara de tiempo de vuelo empleada en los experimentos. . . .	141
9.19. Imágenes de rango de una cámara TOF (colector de escape). . .	142
9.20. Descriptores CIRCON correspondientes y alineamiento aproximado para dos imágenes de rango de una cámara TOF (Ej.: colector de escape).	142
9.21. Imágenes de rango de una cámara TOF (cubo de rueda).	143
9.22. Descriptores CIRCON correspondientes y alineamiento aproximado para dos imágenes de rango de una cámara TOF (Ej.: cubo de rueda).	143
9.23. Alineamiento 3D de imágenes de rango de diferentes dispositivos.	145
9.24. Alineamiento 3D en escenas con mezcla de objetos (cubo de rueda).	146
9.25. Alineamiento 3D en escenas con mezcla de objetos (carcasa de alternador).	147
10.1. Diagrama de bloques básico del proceso de búsqueda de correspondencias.	150
10.2. Descomposición de una imagen utilizando la Transformación Wavelet Discreta (DWT).	151
10.3. Situación inicial de las nubes de puntos.	152
10.4. Imágenes CIRCON correspondientes a las dos nubes de puntos del mismo objeto mostradas en la parte superior de la figura. . .	153
10.5. Aproximaciones Wavelet para tres niveles de resolución de las dos imágenes CIRCON iniciales (A_0 y B_0).	154
10.6. Vista en planta de las reconstrucciones 3D obtenidas a partir de las aproximaciones mostradas en la Figura 10.5.	155
10.7. Mapas de los valores de similitud y de los índices de auto-rotación.	159

10.8. Las diez imágenes CIRCON con mayor valor de similitud para el ejemplo.	160
10.9. Refinamiento de la transformación euclídea.	163
10.10 Plano ajustado para calcular la corrección de la transformación euclídea.	164
10.11 Alineamiento Final.	165
10.12 Resultados obtenidos para diferentes objetos.	166
A.1. Obtención de los ejes del sistema de coordenadas local a partir del s.d.c. global.	177
A.2. Obtención del ángulo γ_q a partir de los triángulos que se pueden extraer de la Figura A.1.	178
B.1. Superficie descrita por M_G (Ejemplo 1).	184
B.2. Superficie descrita por M_G (Ejemplo 2).	185

ÍNDICE DE FIGURAS

Índice de tablas

9.1. Errores de alineamiento para 10 objetos diferentes.	135
9.2. Errores de rotación y traslación para los alineamientos de las Figuras 9.13 y 9.14.	137
10.1. Resultados de Verificación para el ejemplo.	161

ÍNDICE DE TABLAS

Capítulo 1

Introducción

1.1. Descripción del problema y motivación

Coger un objeto de una caja en la que hay muchos más objetos iguales colocados de forma aleatoria es una tarea que resulta simple para un humano desde sus primeros años de vida. Realizar la misma tarea con un manipulador robótico se convierte en una tarea bastante más complicada, pues implica determinar qué objeto se puede coger en primer lugar, determinar en qué orientación se encuentra, cuál es la mejor manera de agarrar el objeto en esa situación y finalmente asirlo con una fuerza adecuada pues, dependiendo del material con el que esté fabricado, el objeto podría sufrir daños si la fuerza fuese muy grande o caerse al suelo si ésta no fuese suficiente.

Si consideramos una tarea más compleja en la que todos los objetos presentes en la caja fuesen diferentes, un bebé necesitaría un cierto tiempo de aprendizaje para identificar, seleccionar, coger y clasificar cada objeto; aunque una vez superada esa etapa de aprendizaje no sería para él una tarea demasiado complicada. En cambio, para un sistema robótico es realmente difícil encontrar una solución a este problema, a no ser que éste sea simplificado o se le impongan una serie de restricciones que lo hagan viable. Todas las operaciones que conlleva localizar, reconocer y manipular un objeto de manera no supervisada en un entorno en el que diferentes objetos aparecen mezclados de manera aleatoria (clutter) es un problema que aún no ha sido totalmente

resuelto y que es conocido en la literatura como Bin-Picking Problem.

El principal inconveniente en estas tareas de captura robotizada que se han descrito se encuentra en determinar la posición y orientación de objetos de diferentes formas cuando solamente se puede extraer información de una parte de ellos. Aunque existen infinidad de propuestas para resolver este problema, aún no ha sido convenientemente resuelto.

Por esta razón, la principal motivación de esta tesis es contribuir a la búsqueda de una solución más general al problema de localización de los objetos en aplicaciones de captura robotizada utilizando únicamente métodos de adquisición que nos proporcionen información de la escena en forma de nubes de puntos 3D.

Los métodos de adquisición de información tridimensional de las superficies de los objetos han mejorado considerablemente en las últimas dos décadas, de manera que cada vez es más frecuente el uso de nubes de puntos obtenidas mediante escáneres láser o cámaras de tiempo de vuelo. Sin embargo, su utilización conlleva ciertos inconvenientes que se traducen en tener que disponer de un método para identificar las diferentes partes en las que puede dividirse la nube de puntos y en determinar la orientación de cada una de esas partes mediante la comparación con un modelo conocido. También se plantean problemas cuando se desea digitalizar la forma completa de un objeto (lo que se conoce como modelado 3D), ya que será necesario realizar varias pasadas con el escáner que, posteriormente, habrá que referir a un sistema de coordenadas común. En cualquiera de los dos casos, será necesario disponer de un algoritmo que permita encontrar correspondencias entre las dos nubes de puntos para poder calcular la transformación que permite alinearlas. En nuestro caso, vamos a considerar que los objetos son indeformables, de modo que dicha transformación va a ser rígida, es decir, va a constar solamente de rotación y traslación.

Se tratará de buscar una solución que pueda ser aplicada con el menor número de restricciones posible. Para ello enfocaremos nuestro trabajo en tratar de superar aquellos inconvenientes que hacen que otros métodos fallen, como son la falta de una información completa de los objetos (debido a oclusiones o a la mezcla de objetos) o la dependencia de una información densa y precisa

para que éstos funcionen.

Por consiguiente, nuestro objetivo es desarrollar un algoritmo de alineamiento que tenga carácter general (es decir, que no esté restringido a una determinada topología de los objetos), que no necesite una gran precisión y densidad de las nubes de puntos y que permita obtener una solución aceptable aun en condiciones de bajo solapamiento entre las nubes de puntos.

1.2. Organización de los capítulos de la tesis

Además de este capítulo de introducción, este documento tiene los siguientes contenidos:

- Capítulo 2: Una revisión del estado del arte centrada principalmente en la adquisición de información tridimensional así como en los diversos métodos de alineamiento de nubes de puntos 3D, haciendo especial hincapié en los basados en descriptores.
- Capítulo 3: Descriptor propuesto para alineamiento de nubes de puntos 3D. Descripción y aplicaciones.
- Capítulo 4: Un sistema de visión 3D basado en el descriptor presentado en el Capítulo 3.
- Capítulo 5: Medida de similitud propuesta. Descripción de los fundamentos de ésta y cómo su aplicación a la búsqueda de correspondencias.
- Capítulo 6: Algoritmo propuesto para la búsqueda de correspondencias entre dos nubes de puntos. Explicación de su funcionamiento mediante un ejemplo.
- Capítulo 7: Un algoritmo de alineamiento aproximado que combina el descriptor y la medida de similitud propuestos en los Capítulos 3 y 5 respectivamente.
- Capítulo 8: Una adaptación del algoritmo ICP que usa la distribución en celdas de los puntos de las nubes que es proporcionada por el algoritmo de alineamiento aproximado.

- Capítulo 9: Resultados de los experimentos realizados.
- Capítulo 10: Versión del algoritmo del Capítulo 7 basada en descomposición wavelet.
- Capítulo 11: Conclusiones y trabajo futuro.
- Apéndices A, B y C: Explicación detallada de ciertos aspectos de los Capítulos 3 y 5.

Capítulo 2

Estado del Arte

2.1. Introducción

Dentro del proceso de captura robotizada de objetos, ya estén éstos separados o apilados en un contenedor [17, 51, 57, 16], las partes que más dificultades plantean son el reconocimiento y localización de los objetos, por lo que será en ellas donde centremos nuestra atención.

Como ya se comentó en el Capítulo 1, cuando se utilizan dispositivos de adquisición de imágenes de rango esos dos problemas pueden reducirse a tratar de encontrar una transformación rígida (rotación y traslación) que permita alinear dos nubes de puntos: la correspondiente a la escena que ha sido escaneada y la del modelo del objeto que se pretende localizar en ella.

En la Sección 2.2 se hace una revisión de los diferentes métodos que permiten adquirir datos tridimensionales en forma de imágenes de rango, mientras que en la Sección 2.3 se analizan diferentes aspectos relacionados con el alineamiento de nubes de puntos y se establece una clasificación de los métodos más relevantes del estado del arte.

2.2. Adquisición de formas tridimensionales

2.2.1. Introducción

La mayoría de los sistemas comerciales de adquisición de información tridimensional proporcionan un conjunto de muestras discretas que corresponden a la profundidad con la que son vistos los puntos de una superficie dada desde un punto de visión concreto. Ese conjunto de muestras ordenadas recibe el nombre de *imagen de rango* pues, en vez de representarse un valor de intensidad o un color, se representa el valor de profundidad de los puntos.

Esa información de profundidad puede obtenerse de diferentes formas, desde simples sistemas de medida de distancia a complejos escáneres 3D para cualquier tamaño de objeto. Su principal atractivo es que son métodos de no contacto, por lo que pueden utilizarse en un amplio rango de situaciones adversas, como por ejemplo con objetos frágiles, pegajosos o que contengan sustancias químicas agresivas. Además son sistemas relativamente rápidos y económicos, aunque no funcionan bien con transparencias y reflexiones múltiples.

De forma general, los métodos de adquisición de datos de rango se suelen clasificar en dos grandes categorías: activos y pasivos [49].

Los métodos pasivos tienen la ventaja de no necesitar una fuente de luz adicional que no sea la luz ambiente que ilumina la escena, por lo que pueden utilizarse en ambientes exteriores. Por esta razón, hay un gran número de aplicaciones que utilizan este tipo de métodos. Aunque, debido a que solamente utilizan cámaras y éstas proporcionan información bidimensional, será necesario llevar a cabo algún tipo de procesamiento adicional para obtener información de profundidad. La técnica pasiva más común para obtener una imagen de rango es la visión estereoscópica. Aunque es cierto que existen otras técnicas pasivas como las basadas en sombras [108, 84], enfoque-desenfoco [34], estructura desde movimiento [31, 9] o estéreo fotométrica [10], entre otras [38], normalmente no están disponibles en aplicaciones industriales o no pueden utilizarse para adquisición en tiempo real.

El concepto básico de la visión estereoscópica es la triangulación; si las

cámaras están debidamente calibradas [109, 101, 45] y se conoce la distancia (línea de base) y el ángulo relativo entre ellas, es posible determinar la distancia al objeto mediante la diferencia de posición de sus proyecciones en las dos imágenes. Para ello, estos sistemas necesitan resolver el llamado problema de correspondencia [43, 42, 80], el cual consiste en determinar qué pares de puntos en las dos imágenes son proyecciones del mismo punto del espacio tridimensional. Solucionar este complejo problema requiere aplicar técnicas que son computacionalmente muy costosas. Otra posibilidad es emplear sistemas pasivos de triangulación con más de dos cámaras [43, 100], sin embargo muchas de estas soluciones tienen problemas de robustez y sensibilidad a las condiciones de iluminación [75].

A diferencia de los métodos anteriores, las técnicas activas utilizan una fuente de luz bien definida (por ejemplo luz láser) para simplificar el problema de correspondencia. Actualmente, estos sistemas activos son superiores, en cuanto a precisión, coste y robustez, a los sistemas estereoscópicos y a otros métodos industriales de medida.

Los métodos activos se basan principalmente en dos técnicas: triangulación y tiempo de vuelo (TOF-Time Of Flight). En la Figura 2.1 se muestra una clasificación de los diferentes sistemas industriales que proporcionan medidas de rango 3D.

En los sistemas de triangulación activa, la escena es iluminada por una fuente de luz láser desde una dirección determinada mientras que es observada por una cámara desde otra dirección. En cambio, en los sistemas TOF se mide el tiempo que tarda en reflejarse un pulso láser para así determinar la distancia a un objeto. Las ventajas de ambos métodos activos son la densidad de puntos obtenida, su gran robustez y precisión y, sobre todo, que se resuelve el problema de la correspondencia que presentan los métodos pasivos. Sin embargo, se deben añadir fuentes de luz adicionales, lo que hace que no sean de fácil aplicación en ambientes exteriores, además de verse afectada su precisión por la diferente manera en que las superficies de los objetos absorben y reflejan la luz.

Entre las dos metodologías activas la principal diferencia estriba en el rango de distancias que pueden llegar a medir; mientras que los basados en tiempo de vuelo pueden ser de largo alcance (pueden superar sin problemas varios

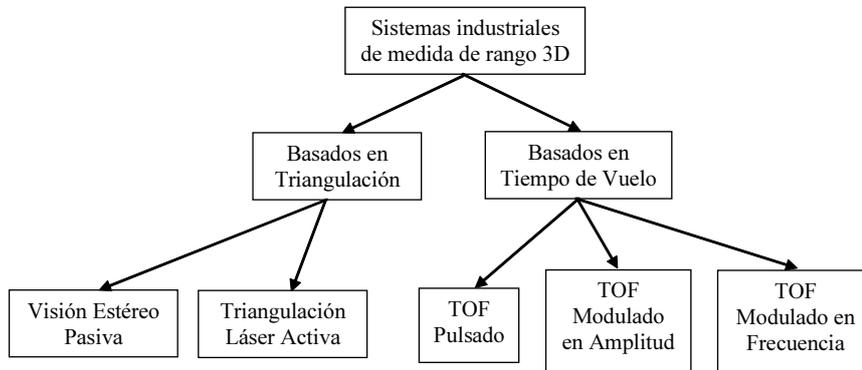


Figura 2.1: Clasificación de los sistemas industriales de medida de rango 3D.

metros de distancia) los métodos basados en triangulación son de corto alcance, debido a que pierden precisión cuando los objetos se alejan de la cámara, ya que necesitan ver con nitidez la luz láser sobre su superficie.

La mayoría de los escáneres de largo alcance se usan, entre otras aplicaciones, para catalogación y conservación del patrimonio cultural y arqueológico [85, 18], obtención de información geodésica o escaneo 3D de edificios. En cambio, los escáneres de corto alcance suelen utilizarse en aplicaciones de control de calidad, ingeniería inversa, visualización y modelado 3D, además de ser una de las principales fuentes de información tridimensional en sistemas robóticos industriales [54, 39, 16].

Aunque en esta tesis vamos a utilizar fundamentalmente conjuntos de datos obtenidos mediante escáneres láser de triangulación activa, en realidad, su procedencia no va a ser ningún obstáculo, siempre que su densidad sea suficiente para obtener una nube de puntos que describa convenientemente la superficie del objeto escaneado. A continuación vamos a hacer una breve descripción de los principales métodos activos para escáneres láser.

2.2.2. Método activo para medida de rango basado en triangulación

Como ya se expuso anteriormente, los sistemas de triangulación activa son aquellos que utilizan un fuente de luz, con posición y orientación conocidas,

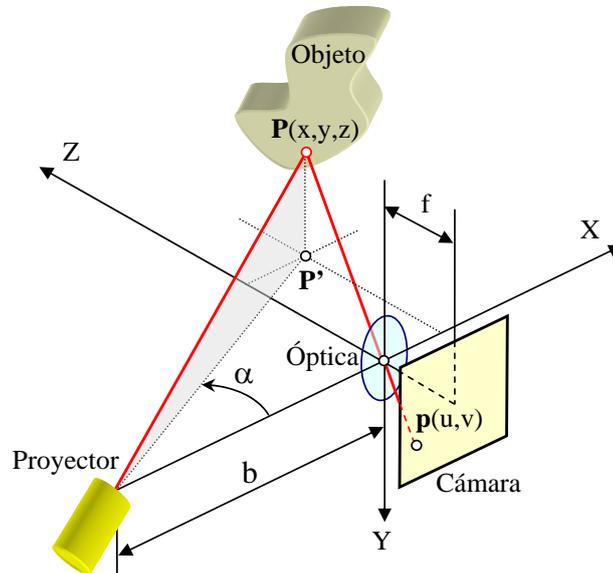


Figura 2.2: Sistema basado en la proyección de un punto luminoso.

para proporcionar una información adicional que contribuya a la reconstrucción de la escena.

Metodología

Para mayor claridad en la exposición, vamos a considerar el sistema mostrado en la Figura 2.2, que va a constar de una fuente de luz (láser) y una cámara CCD.

El láser va a emitir un haz de luz lo suficientemente fino como para que su proyección sobre el objeto se pueda aproximar al punto P . Esto nos va a permitir, cuando enfoquemos el objeto con la cámara, que dicho punto se pueda percibir con claridad debido a su mayor intensidad.

La posición de P se va a poder determinar por triangulación; por simplicidad vamos a elegir como sistema de coordenadas de referencia aquél que está asociado a la cámara:

1. El origen va a estar situado en el centro de la lente.
2. El eje Z va a coincidir con el eje óptico.

3. Los ejes X e Y van a ser paralelos respectivamente a los ejes u y v de coordenadas de imagen.

La fuente de luz va estar localizada sobre el eje X a una distancia b del origen de coordenadas.

Vamos a suponer, también por simplicidad, el punto P' que está en el plano X - Z . La distancia b y el ángulo α van a ser parámetros conocidos, puesto que pueden determinarse mediante calibración del sistema.

Entonces, aplicando propiedades trigonométricas y semejanza de triángulos se pueden hallar las expresiones que permiten calcular las coordenadas 3D del punto que es iluminado por el láser:

$$X_P = u \cdot \frac{b}{u - f \cdot \cot \alpha} \quad (2.1)$$

$$Y_P = v \cdot \frac{b}{u - f \cdot \cot \alpha} \quad (2.2)$$

$$Z_P = -f \cdot \frac{b}{u - f \cdot \cot \alpha} \quad (2.3)$$

De estas expresiones se deduce que las coordenadas del punto 3D se pueden calcular en base a las coordenadas de su proyección en el plano de imagen ($u, v, -f$) multiplicadas cada una de ellas por un factor que depende de la posición (b) y orientación (α) de la fuente de luz, de la coordenada horizontal de la proyección (u) y de la longitud focal de la cámara (f).

Este método, aunque sencillo, va a hacer que el proceso de reconstrucción sea muy lento debido a que el punto luminoso ha de recorrer toda la escena, vertical y horizontalmente. Por esta razón, van a existir diversos métodos de triangulación activa que difieren en la manera de proyectar y detectar la luz. De entre todos ellos vamos a centrarnos en describir los que son más utilizados (aunque existan otros métodos como, por ejemplo, los sincronizados [86] o los basados en la razón de intensidad [15]):

- Métodos de proyección multipunto
- Métodos de proyección de líneas

- Métodos de proyección de patrones

Los factores que hay que considerar para la elección de uno u otro método son los siguientes:

- Precisión con la que se quiere reconstruir la escena.
- Existencia de zonas muertas donde no es posible triangular.
- Velocidad de rastreo de la escena (que nos determinará la rapidez del método).

En cuanto a los requerimientos básicos de este tipo de sistemas, éstos deberán constar al menos de una cámara, una fuente de luz y el hardware necesario para la computación.

Métodos de proyección multipunto

Para acelerar el proceso de barrido de la escena se puede utilizar una rejilla de puntos, para lo cual necesitaremos un sistema de iluminación que sea capaz de proyectarla.

Desplazando la rejilla sobre una pequeña zona es posible cubrir toda el área de trabajo. Los rangos de barrido horizontal y vertical van a estar delimitados por las distancias D_x y D_y entre puntos de la rejilla (ver Figura 2.3).

Si se incrementa el número de puntos proyectados (reduciendo la distancia entre puntos) se puede reducir el rango de barrido y, de este modo, acelerar el proceso, pero se corre el riesgo de que puedan surgir ambigüedades en la correspondencia. Por lo tanto, el número de puntos de la rejilla va estar limitado por este inconveniente.

Aunque se tenga en cuenta esto último, es inevitable tener que recurrir a algún método para eliminar posibles ambigüedades que pudieran surgir. Uno de los más eficaces para este tipo de sistemas consiste en establecer la correspondencia al inicio del proceso de barrido y, a continuación, hacer un seguimiento de cada uno de los puntos después de cada desplazamiento horizontal Δx y vertical Δy . Considerando ciertas restricciones sobre las superficies a reconstruir, es posible a partir de Δx y Δy pronosticar con una cierta aproximación el entorno de la imagen donde se va a encontrar cada punto.

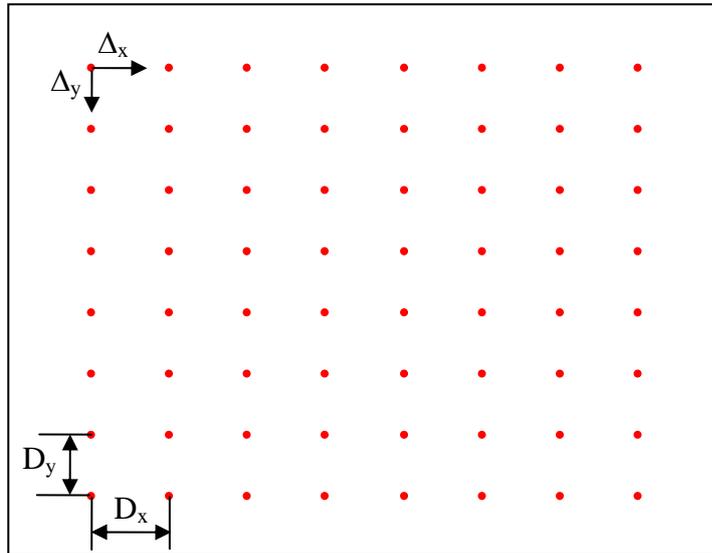


Figura 2.3: Rejilla para proyección multipunto.

Métodos de proyección de líneas

Los métodos activos parten del hecho de que un punto 3D va a ser resultado de la intersección de un rayo luminoso con la recta de proyección de su píxel correspondiente.

Si se utiliza, en cambio, un plano de luz con posición y orientación conocidas, los píxeles de la imagen que aparecerán brillantes serán aquellos puntos de la escena que corresponden a su intersección con dicho plano. Entonces podremos calcular las coordenadas 3D de los puntos (sin ambigüedad alguna) mediante la intersección de las rectas de proyección asociadas a dichos píxeles con el plano de luz cuya ecuación es conocida. De esta manera, en un solo intervalo de muestreo podemos reconstruir todos los puntos pertenecientes a una línea de barrido.

El hardware mínimo requerido para este método (Figura 2.4) va a constar de una cámara (necesariamente bidimensional), una fuente de luz que sea capaz de proyectar una línea sobre la escena y un sistema mecánico que desplace o bien la línea de barrido o bien el objeto a reconstruir. Estos elementos se suelen disponer de tal manera que las líneas de barrido se desplacen vertical u

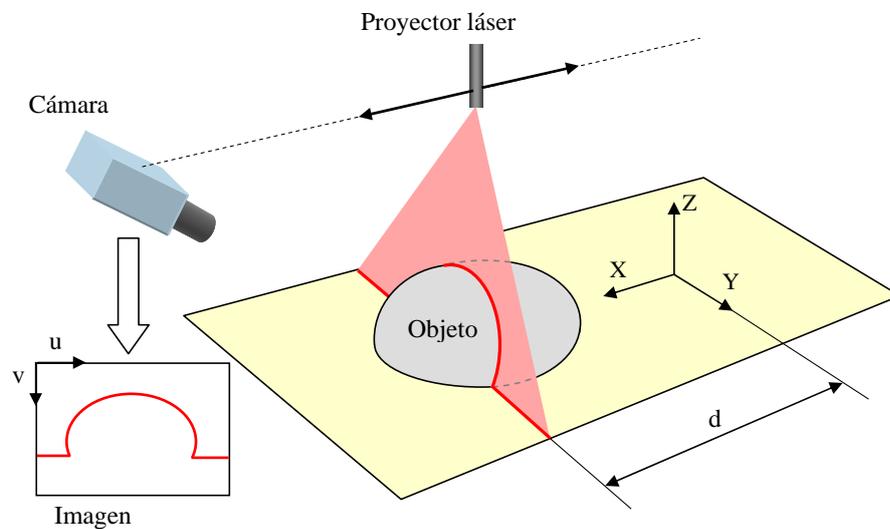


Figura 2.4: Sistema de visión 3D mediante proyección de una línea láser.

horizontalmente en la imagen.

El procedimiento a seguir consiste en desplazar el plano de luz sobre la escena mientras la cámara registra la secuencia. Entonces, para cada imagen adquirida, se deberán buscar aquellos píxeles que tengan una mayor intensidad para, a continuación, obtener mediante triangulación las coordenadas 3D correspondientes a cada uno de ellos. Aplicando esto a todas las imágenes de la secuencia podemos obtener un mapa de profundidad de la escena.

Una forma de agilizar el proceso de barrido consistiría en proyectar múltiples líneas tal y como se hacía en los sistemas multipunto. Sin embargo, como ya se comentó, el principal desafío de estos sistemas es la separación de las diferentes direcciones de proyección de una forma única y sin ambigüedades. Para ello, será necesario establecer ciertas restricciones a las superficies que se traten de reconstruir.

La aparición de ambigüedades en estos sistemas multilínea se puede evitar proyectando líneas de colores sobre la escena.

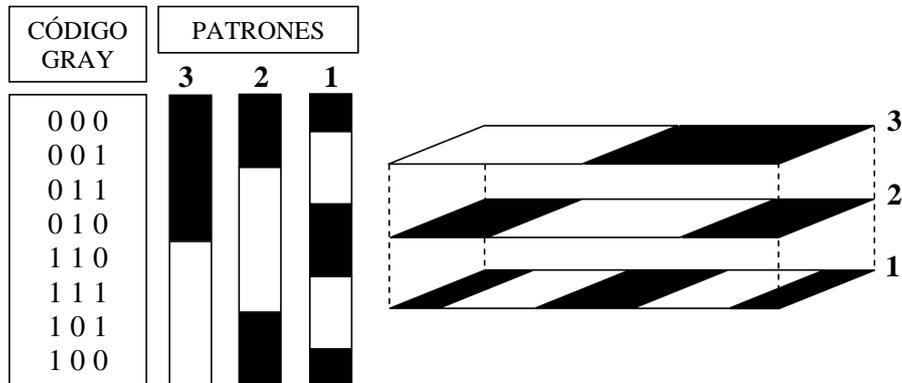


Figura 2.5: Ejemplo de patrones de proyección utilizando un código Gray de 3 bits.

Métodos de proyección de patrones

Según se ha visto en el apartado anterior, utilizar una línea de barrido implica que se necesitarán n imágenes para obtener la reconstrucción de n secciones de la escena. Por consiguiente, puede llegar a ocurrir que ese número n sea demasiado grande.

Este problema puede resolverse mediante la proyección de patrones codificados en binario, pues se requieren solamente $\log_2 n$ imágenes para obtener la misma información que en el caso anterior.

El código Gray es el más empleado en este tipo de métodos pues, a diferencia del código binario natural, sus números consecutivos difieren en un único dígito (ver Figura 2.5). De este modo, dado que es más probable que se produzca una decodificación errónea en aquellas localizaciones donde se cambia un bit, ésta sólo introduciría un desplazamiento de una unidad de resolución respecto a la posición correcta. Por otro lado, las líneas del patrón correspondiente a la máxima resolución son dos veces más anchas que las del código binario natural, lo que facilitaría el análisis en superficies abruptas en las que el código pudiera aparecer comprimido.

Por el contrario, una desventaja de esta técnica es que, debido a que cada dirección de proyección está asociada con una palabra del código, la resolución es más bien baja. Por ejemplo, con 9 imágenes podríamos discriminar 512

planos de la escena.

Afortunadamente, cuando este método se combina con una técnica de desplazamiento de fase [13] se puede aumentar considerablemente la resolución del sistema, que estará limitada por los errores debidos a la cuantización de los niveles de gris de la imagen y por la presencia de ruido electrónico.

2.2.3. Método activo para medida de rango basado en tiempo de vuelo

Los sistemas basados en tiempo de vuelo miden la distancia entre el objeto y la fuente de luz. Para ello emiten un rayo luminoso que será reflejado de manera difusa por la superficie del objeto, de modo que sólo una parte de la luz emitida llegará al receptor (ver Figura 2.6). Entonces se mide el tiempo que tarda la luz en realizar ese viaje de ida y vuelta entre el diodo láser y el objeto.

Para evitar efectos de sombra se sitúan muy cerca el emisor y el receptor, de modo que las direcciones de iluminación y observación sean aproximadamente colineales.

Los métodos existentes que se basan en el principio de tiempo de vuelo se pueden clasificar en tres categorías:

- TOF de láser pulsado
- Onda continua modulada en amplitud
- Onda continua modulada en frecuencia

En el caso de TOF de láser pulsado, se va a poner en marcha un cronómetro en el momento en que se emite el pulso de luz. Cuando el pulso de luz llega al receptor tras reflejarse en la superficie del objeto se mide el tiempo transcurrido. Entonces se calculará la distancia teniendo en cuenta la velocidad de la luz en el medio.

Teóricamente, la precisión de la medida de profundidad va a ser independiente de la distancia del objeto a la cámara y solamente dependerá de la precisión con la que se mida el tiempo de vuelo. Además, el rango máximo va a estar influenciado por la frecuencia de repetición de los pulsos, pues para

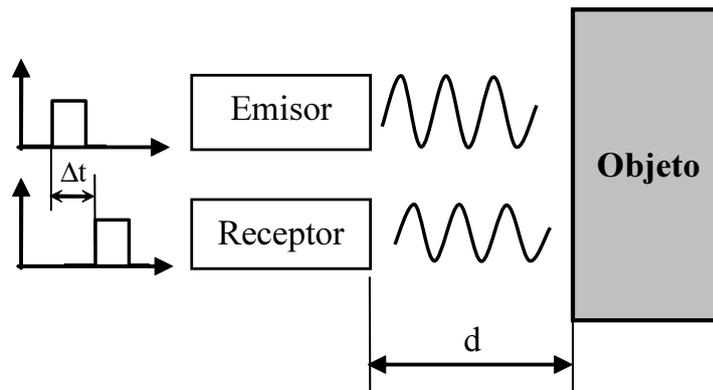


Figura 2.6: Medida basada en tiempo de vuelo con láser pulsado.

poder enviar un nuevo pulso se tiene que esperar a que llegue el último eco al receptor. Otros aspectos que van a limitar la máxima distancia que puede ser medida con estos sensores son la sensibilidad de la unidad receptora y la potencia del pulso emitido.

Otra forma de obtener una medida de distancia mediante el tiempo de vuelo consiste en medir el desplazamiento de fase (ver Figura 2.7). En este caso se va a utilizar una emisión continua de luz y se medirá la diferencia entre las fases de las señales emitida y recibida para extraer la información de distancia. La onda continua que se utiliza en este método puede modularse en amplitud o en frecuencia.

El rango de medida para este tipo de sensores depende de la longitud de onda de la señal modulada, por lo que se puede mejorar su resolución si se usan bajas longitudes de onda. Por otro lado el retardo de fase que puede detectarse sin ambigüedades está limitado a 360° , lo que significa que la máxima distancia medible equivale a la mitad de la distancia que viaja la luz en un periodo.

Aunque la medición de desplazamiento de fase tiene una mayor precisión que la del TOF de láser pulsado, normalmente se utiliza una combinación de los dos métodos. En cualquier caso, la onda reflejada debe llegar con una calidad apropiada para que pueda calcularse la distancia correcta. Algunas veces este problema conduce a resultados erróneos.

Dado que el principio del TOF se basa en una única medida de distancia mediante un rayo láser, para adquirir una superficie completa los escáneres

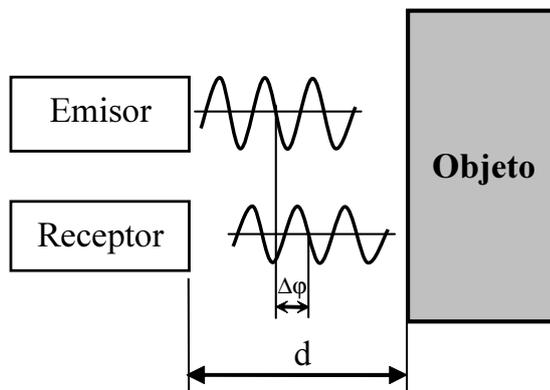


Figura 2.7: Medida basada en desplazamiento de fase para sistemas de tiempo de vuelo.

basados en TOF deben realizar un "barrido" de la superficie mediante desplazamientos definidos del rayo láser. Normalmente, éste se va a mover mediante incrementos angulares, de modo que los resultados consistirán en contornos formados por las distancias radiales correspondientes. Si el escáner se desplaza en una dirección perpendicular a la dirección de proyección del rayo láser, se puede repetir lo anterior de manera sucesiva hasta adquirir una superficie completa.

Otros sensores consisten en un array de unidades emisoras y receptoras, con el fin de adquirir una imagen de rango completa sin tener que realizar desplazamientos. En ocasiones, este tipo de sensores aparecen combinados con cámaras CCD, dando lugar a las cámaras de tiempo de vuelo [63].

Los sensores de tiempo de vuelo proporcionan buenos resultados en cuanto a precisión y resolución para rangos de medida de hasta 100m [12]. Otra de sus principales ventajas es que no necesitan resolver el problema de la correspondencia que tienen los métodos pasivos, ni tampoco presentan las ambigüedades de rango propias de los métodos de triangulación activa. Sin embargo son menos precisos para medidas de corto alcance. Su precisión varía desde unos pocos milímetros hasta dos o tres centímetros, dependiendo de cómo se lleve a cabo la medida del tiempo y de la distancia que haya entre el objeto y el escáner.

2.3. Alineamiento 3D

2.3.1. Introducción

Dado que la mayoría de los escáneres tienen un campo de visión limitado y que los objetos no son inmunes a las auto-oclusiones, será necesario realizar varios escaneos del mismo objeto desde diferentes puntos de visión para poder obtener de él un conjunto de muestras suficiente con el que poder construir un modelo 3D que lo represente de manera adecuada [29].

Cada vista del objeto estará referida al sistema de coordenadas del escáner, por lo que será necesario obtener la transformación que exprese sus puntos en el sistema de coordenadas de referencia. De esta manera será posible construir el modelo completo del objeto escaneado.

Dicha transformación se puede obtener fácilmente si se conocen los movimientos que realiza el objeto cuando se escanean sus diferentes vistas. Lo más habitual es colocar el objeto sobre una plataforma giratoria que esté calibrada, aunque otra posibilidad sería registrar el movimiento del sensor mientras se realiza un escaneado completo. Si se opta por la primera opción el tamaño de los objetos que pueden reconstruirse estará limitado por el tamaño de la plataforma y si, en cambio, se opta por la segunda, la complejidad del diseño del escáner va a hacer que éste se encarezca considerablemente, lo cual obligará prácticamente a descartarlo en muchas aplicaciones debido a su inviabilidad económica.

Una tercera opción es incluir un área común entre vistas consecutivas, de modo que, mediante estos solapamientos, se pueda obtener la transformación relativa entre cada par de vistas. Aplicando este proceso a todas las vistas del objeto se puede reconstruir su forma completa.

Mediante algoritmos de alineamiento 3D se pueden encontrar esas áreas comunes entre vistas para calcular una transformación rígida que las alinee y así poder referir todas las vistas del objeto a un sistema de coordenadas común.

Otro problema diferente al de construir un modelo 3D, pero que puede solucionarse mediante alineamiento de superficies, es determinar la orientación de

un objeto escaneado para poder capturarlo mediante un manipulador robótico. Para ello se buscará la mejor transformación que alinee la superficie escaneada con un modelo 3D del objeto que haya sido construido previamente.

2.3.2. Alineamiento entre Pares de Vistas vs. Alineamiento Multivista

En general, se puede hablar de dos tipos diferentes de alineamiento de superficies: el de dos vistas (en inglés "pairwise registration") y el multivista ("multiview registration"). El primero va a tratar de encontrar una transformación que alinee dos vistas parciales escaneadas o una vista parcial escaneada con un modelo 3D del objeto. En cambio, el segundo tipo va tener como objetivo crear el modelo tridimensional de un objeto procesando grupos de vistas (normalmente todos los escaneos).

El alineamiento *pairwise* parte de dos superficies A y B , normalmente llamadas "escena" y "modelo", para buscar una transformación rígida T que, aplicada a los puntos de A , minimice la distancia entre estos y los puntos de B . Debido a que normalmente las superficies escaneadas se van a solapar de manera parcial, esa distancia debería calcularse solamente sobre los puntos que pertenezcan a esas áreas comunes, las cuales serán inicialmente desconocidas. Por consiguiente, la calidad del alineamiento se va medir como la distancia residual después de aplicar la mejor transformación que se haya encontrado.

Sería posible además, aplicar un algoritmo de alineamiento *pairwise* para obtener la forma completa de un objeto a partir de varios escaneos. Tomando como referencia al primero de todos ellos, el resto de escaneos podría alinearse sucesivamente a éste o a cualquiera de los que ya se hayan ido ensamblando. Sin embargo, este procedimiento tiende a producir malos resultados debido a que los errores cometidos en cada alineamiento entre pares de vistas tienden a acumularse de tal manera que las vistas parciales primera y última no encajarán bien. Precisamente, para solucionar este problema, se desarrollaron los algoritmos de alineamiento multivista.

Dados N escaneos parciales de un objeto: S_1, S_2, \dots, S_N , el objetivo de

un algoritmo multivista consiste en encontrar N transformaciones, T_1, T_2, \dots, T_N , de tal modo que, si se aplican a sus escaneos correspondientes, la suma de distancias entre todos aquellos pares que puedan solaparse entre sí sea mínima. Esto quiere decir que aquellos pares de vistas que no compartan un área común no contribuirán al error de alineamiento. De ahí puede deducirse que el principal desafío de un algoritmo multivista será decidir qué pares de vistas van a considerarse en el cálculo del error.

La mayoría de los algoritmos multivista [50, 85, 60, 93] van a aplicar una estrategia *pairwise* para obtener un alineamiento previo de los escaneos y, a continuación, van a tratar de ajustar las transformaciones resultantes con el objetivo de minimizar el error total de alineamiento. Por esa razón, el alineamiento *pairwise* ha recibido una mayor atención por parte de la comunidad científica, ya que una mejora en el alineamiento entre pares de vistas va a proporcionar un mejor punto de partida para los algoritmos multivista.

En esta tesis vamos a centrar también nuestra atención en los algoritmos de alineamiento entre pares de vistas, ya que tienen una aplicación directa a los problemas de localización y reconocimiento de objetos.

2.3.3. Clasificación y requisitos de los algoritmos de alineamiento 3D

Los algoritmos que permiten obtener una solución a este problema de alineamiento cuando las nubes de puntos están situadas en posiciones arbitrarias y no se conoce ninguna estimación de la transformación euclídea están basados fundamentalmente en dos estrategias básicas: búsqueda en el espacio de parámetros de la transformación o búsqueda en el espacio de todas las correspondencias posibles entre los puntos de las nubes.

Los métodos que realizan una búsqueda directa de la transformación se basan en el hecho de que el espacio de parámetros que la definen es de baja dimensión, por lo que se va a realizar una búsqueda exhaustiva del pequeño conjunto de parámetros que permiten especificar la transformación óptima. La forma en la que se realiza dicha búsqueda da lugar a dos tipos de métodos: los

que exploran la superficie del error y los que se basan en votación.

Los primeros tratan de minimizar el error de alineamiento que resulta al aplicar cada transformación. Debido a la no linealidad de este problema la superficie del error presenta muchos mínimos locales, de modo que, para buscar la transformación óptima, se han aplicado muchos de los métodos que se conocen para escapar de los mínimos locales: búsqueda evolutiva [37], algoritmos genéticos [24, 94] y recocido simulado [58, 81]. Los principales inconvenientes de estos métodos es que pueden ser realmente lentos y que no garantizan la convergencia al alineamiento óptimo. Además, su funcionamiento depende considerablemente de la inicialización del algoritmo.

Por otro lado, los métodos de votación dividen el espacio de parámetros en una tabla de seis dimensiones en cuya celdas se acumulan los votos correspondientes a las transformaciones que resultan de alinear ternas de puntos de ambas nubes (escena y modelo). Ejemplos de este tipo de métodos son la transformada de Hough generalizada [44], el hashing geométrico [112] y el pose clustering [98, 92]. El principal inconveniente de estos métodos es su alto coste computacional, de modo que generalmente van precedidos por una etapa de extracción de características o por una simplificación de las nubes de puntos. Asimismo, debido a la gran necesidad de almacenamiento de información que presentan, estos métodos de votación solamente son útiles cuando el número de peticiones de alineamiento es lo suficientemente alto como para compensar el coste computacional del preprocesamiento, lo cual puede llegar a ocurrir en aplicaciones de reconocimiento de objetos o de recuperación de formas desde bases de datos 3D.

Si en vez de buscar en el espacio de parámetros de la transformación la búsqueda se lleva a cabo en el espacio de todas las posibles correspondencias que se pueden establecer entre las nubes de puntos, el problema se va a dividir en dos partes bien diferenciadas: correspondencia y alineamiento (ver Figura 2.8). La primera parte consiste en identificar regiones de las dos nubes de puntos que representen la misma región física del objeto escaneado; mientras que la segunda, el alineamiento, trata de encontrar la transformación rígida que minimice la distancia entre esas partes correspondientes.

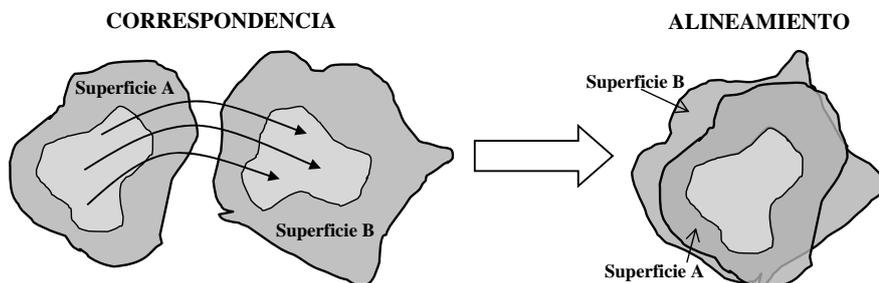


Figura 2.8: Establecimiento de un conjunto de correspondencias entre dos superficies A y B que permita un correcto alineamiento mediante el cálculo de una transformación rígida.

Aunque este problema se ha tratado de solucionar mediante optimización del conjunto de correspondencias sometido a restricciones geométricas [4], la mayoría de los algoritmos existentes tratan de analizar las superficies para encontrar un conjunto de puntos correspondientes que será empleado para calcular una transformación que las alinee. Por consiguiente, el grado de validez de esas hipótesis de correspondencia va a repercutir en la calidad del alineamiento.

Si en cambio, las correspondencias fuesen conocidas, encontrar el alineamiento sería algo prácticamente trivial. Bastaría con obtener la transformación rígida (R, t) que minimizase la suma de los cuadrados de las distancias entre cada punto de la superficie B y su correspondiente punto transformado en A [33].

Dado que el problema de alineamiento de superficies puede considerarse básicamente como un problema de optimización, los algoritmos se pueden dividir en dos categorías básicas dependiendo del tipo de optimización que realicen: global o local.

Así, se hablará de alineamiento global cuando los algoritmos traten de encontrar la mejor correspondencia entre las superficies sin ningún conocimiento a priori de sus posiciones iniciales; por lo tanto, se puede decir que estos algoritmos estarán destinados a buscar el mínimo global de la función de distancia. Para ello se recurre o bien a representaciones que consideran la forma completa

del objeto (p.e. cilindros generalizados [32], superficies deformables [30], imagen gaussiana extendida [48]), o bien a representaciones de forma de ámbito local. Las primeras tienen como principales limitaciones la falta de generalidad y una notable sensibilidad a las oclusiones, de modo que su principal aplicación es la recuperación de objetos de base de datos (shape retrieval [20]) mientras que las segundas tienen un ámbito de aplicación mucho más amplio y son más robustas ante oclusiones y mezcla de objetos. Por esta razón, se dedicará la Sección 2.4 a hacer una revisión de los algoritmos que utilizan este último tipo de representación de forma.

Por otro lado, se va a hablar de alineamiento local cuando los algoritmos partan de la suposición de que la transformación relativa entre las dos superficies es aproximadamente correcta, es decir, el mínimo local más próximo de la función de distancia es el alineamiento óptimo. Estos algoritmos van a tratar de mejorar el alineamiento de manera local, para lo cual suelen estar basados, casi exclusivamente, en optimización de funciones mediante búsqueda local.

De la misma manera que los algoritmos de alineamiento se pueden clasificar en función del ámbito de búsqueda de la solución, se puede establecer una clasificación diferente atendiendo a la manera en que éstos van a realizar la correspondencia entre las superficies. Aunque existen diferentes formas de llevarla a cabo, la más extendida consiste en emparejar puntos que tengan propiedades de superficie similares. Sin embargo, la estrategia a seguir va a venir condicionada por el tipo de propiedad de superficie que se considere en cada caso. De este modo se van a clasificar las propiedades en dos tipos fundamentales: las intrínsecas y las extrínsecas.

Las propiedades intrínsecas están relacionadas exclusivamente con la información local de la superficie, es decir, son propias de su particular geometría. En cambio, las propiedades extrínsecas van a relacionar la superficie (o la nube de puntos) con el sistema de coordenadas en el que está expresada y que, por lo tanto, se modificarán cuando éste cambie. Un ejemplo de propiedad intrínseca es la distancia entre dos puntos de superficie (que será invariante a transformaciones rígidas), mientras que el plano tangente a uno de esos puntos será un ejemplo de propiedad extrínseca (pues varía cuando la superficie se expresa

en un sistema de coordenadas diferente).

Los algoritmos que utilicen propiedades intrínsecas serán apropiados para determinar una correspondencia aproximada entre las superficies, pues los emparejamientos entre puntos solamente van a basarse en propiedades inherentes a ellas. Por esta razón, serán las que utilicen los algoritmos de alineamiento global para llevar a cabo la correspondencia. Por el contrario, los algoritmos que utilicen propiedades extrínsecas van a estar sometidos a una importante restricción: dado que van a emparejar propiedades que son relativas a un sistema de coordenadas, necesitarán que las superficies estén alineadas de manera aproximada para poder establecer la correspondencia de puntos. Por lo tanto, serán los algoritmos de alineamiento local los que utilicen este tipo de propiedades para refinar la transformación inicial y obtener una más precisa.

Teniendo en cuenta lo anterior se puede llegar a la conclusión de que la parte más desafiante del problema de alineamiento de superficies tridimensionales está en determinar esa correspondencia inicial aproximada, ya que la parte de refinamiento ya ha sido convenientemente resuelta [14, 22] (este tipo de algoritmos de alineamiento fino se explican con más detalle en la Sección 2.5).

Existe un gran número de algoritmos que han tratado de resolver el problema de encontrar una correspondencia de dos superficies sin ninguna suposición de sus orientaciones en el espacio ni de las partes que tienen en común, sin embargo ninguno de ellos ha logrado prevalecer como la solución más aceptada en términos de eficiencia y fiabilidad. Como consecuencia, han surgido en los últimos años nuevas propuestas que intentan corregir las debilidades de los métodos anteriores [70, 69, 41, 19]. En la Sección 2.4 se hace un repaso de los algoritmos más relevantes; para ello se han clasificado en base a las estrategias que siguen para lograr la correspondencia de las superficies.

En general, para que tanto la correspondencia como el alineamiento puedan realizarse de manera completamente automática y con ciertas garantías de fiabilidad, los algoritmos deberían cumplir los siguientes requisitos:

- No debe haber ningún tipo de interacción con el usuario. Es decir, no se van a utilizar marcadores sobre los objetos para simplificar la búsqueda de correspondencias ni tampoco un alineamiento inicial aproximado que

se haya suministrado de forma manual.

- La capacidad del algoritmo para obtener la transformación que alinee las superficies no debe depender de la posición inicial en que éstas se encuentren. Esto va a permitir una mayor flexibilidad a la hora de escanear los objetos.
- Debe ser capaz de alinear vistas parciales de los objetos. Normalmente sólo va haber un subconjunto de puntos correspondientes entre ambas superficies, por lo que uno de los principales objetivos del algoritmo de alineamiento debe ser encontrar esas áreas comunes.
- Se deberá intentar obtener el alineamiento de las superficies con un alto grado de fiabilidad y en el menor tiempo posible.

2.4. Algoritmos de alineamiento aproximado

En esta sección se hace un repaso de los métodos de alineamiento aproximado más relevantes, haciendo una clasificación de éstos en base a la estrategia que siguen para resolver el problema de la correspondencia. Esta clasificación va a dar lugar a dos grandes grupos: los basados en descriptores y los basados en curvas, grafos y estructuras. En un tercer grupo aglutinaremos aquellos métodos que no pueden ser clasificados en los dos primeros. Otras clasificaciones pueden encontrarse en [89, 83].

2.4.1. Basados en descriptores

Un problema clásico de la visión artificial 2D como es la búsqueda de correspondencias entre dos imágenes, es habitualmente resuelto mediante descriptores y seguimiento de características.

En visión tridimensional, estas estrategias 2D pueden ser igualmente válidas, de modo que, dado un punto de la superficie de un objeto, se puede extraer información del entorno de dicho punto que permita caracterizar de forma adecuada su geometría local. Normalmente, un descriptor de forma representa la

superficie local alrededor de cada punto mediante un vector de dimensiones fijas. De este modo, determinar la similitud entre dos puntos potencialmente correspondientes podría reducirse al cálculo de la distancia euclídea entre dos vectores; aunque en la práctica esto no siempre arroja resultados satisfactorios.

Para que los descriptores geométricos sean útiles para obtener correspondencias en algoritmos de alineamiento 3D, estos deberían poseer ciertas propiedades de invarianza y tener un carácter fundamentalmente local.

La invarianza es una característica muy importante en un descriptor, ya que éste no debería cambiar cuando a los puntos del objeto se les aplique una transformación. Dicha transformación puede ser rígida (translación y rotación) o puede someter al objeto a deformaciones afines o articuladas, para las cuales es más difícil encontrar descriptores invariantes.

Por otro lado, el descriptor debería ser local, ya que no es muy habitual en alineamiento 3D que las dos vistas se solapen completamente, sino sólo en parte, lo cual hace que el uso de propiedades globales, tales como el centro de masas o la orientación de los ejes principales, sea, en este caso, inapropiado. Por el contrario, los descriptores deberían basarse en la información del entorno local de cada punto.

Se puede establecer una clasificación de los descriptores de forma 3D más usuales en función de su tamaño. De este modo, se van a dividir en descriptores de baja dimensión y descriptores de alta dimensión.

Si la dimensión del descriptor es pequeña, como ocurre con los basados en algunas propiedades geométricas (p.e. curvatura [35, 59]) o los basados en operaciones de integración de características (p.e. volumen [41]), tanto su cálculo como su comparación se podrán realizar en menos tiempo, pero, en cambio, presentarán mayor sensibilidad al ruido y tendrán menor poder discriminante.

Por otro lado, cuando se utilizan descriptores de baja dimensión, la presencia de simetrías en los objetos y el carácter excesivamente local de dichos descriptores va a facilitar la aparición de muchas correspondencias erróneas al final del proceso de búsqueda. Esto puede verse aún más acentuado cuando las dos nubes de puntos que se quieren alinear tengan un área común pequeña (es decir, bajo solapamiento), de modo que el número de falsas correspondencias pueda llegar a ser considerable frente al de correspondencias buenas. Esto

obligará a realizar un postprocesado de las correspondencias halladas para verificar su validez (generalmente mediante algún tipo de comprobación de su consistencia geométrica), lo cual aumentará el tiempo de computación del algoritmo (haciendo menos atractiva la principal ventaja de estos descriptores) además de correrse el riesgo de que el alineamiento final no sea el apropiado.

Estos problemas podrían mitigarse aumentando el poder discriminante, lo cual se puede conseguir utilizando descriptores de mayor tamaño. Aunque la principal desventaja va a ser un mayor tiempo de computación para obtenerlos y compararlos, este tipo de descriptores van a caracterizar de una manera más precisa el entorno de los puntos de interés. Además, ese sacrificio en la velocidad de cálculo puede verse recompensado con una mayor eficacia en la búsqueda de correspondencias y, por consiguiente, con la obtención de un mejor alineamiento entre las superficies. Incluso parte de ese tiempo "perdido" podría recuperarse si se reduce considerablemente el tiempo invertido en evaluar la validez de las correspondencias obtenidas.

A continuación se presentan los descriptores utilizados por los algoritmos de alineamiento aproximado más relevantes.

Curvatura principal

Feldmar y Ayache desarrollaron un algoritmo [35] que utiliza información de la curvatura principal para encontrar una única correspondencia con la que calcular la transformación euclídea que alinea dos imágenes de rango.

Va a caracterizar cada punto por sus curvaturas principales, las cuales corresponden a la máxima y mínima curvatura de la superficie en el punto. Además también se considerarán la normal en el punto y la dirección de máxima curvatura.

Con el objetivo de facilitar la búsqueda, los puntos de la segunda imagen de rango se organizan en una tabla que es indexada por sus valores de curvatura. De esta forma, si se conocen los valores de curvatura en un punto de la primera vista, se podrá encontrar rápidamente el conjunto de puntos de la segunda vista que tienen unos valores de curvatura similares. Así se podrá evaluar la validez de cada uno de ellos como punto correspondiente. Para cada correspon-

dencia potencial se calcula la transformación euclídea que alinea las normales en los puntos y sus direcciones principales de curvatura. Entonces se aplica la transformación a la primera nube y se comprueba si el porcentaje de puntos correspondidos (cuya distancia está por debajo de cierto umbral) supera un valor determinado. Si no se satisface esta condición de parada el algoritmo se repetirá usando un punto inicial alternativo. En caso afirmativo, la matriz de transformación calculada se considerará una buena estimación.

El principal problema de este método es que no es robusto. Solamente se busca una correspondencia y cuando se encuentra un posible solución el algoritmo se para. Sin embargo, como no se realiza ninguna validación, podría tratarse de una falsa correspondencia, obteniendo de este modo un alineamiento inicial erróneo.

Signatura de puntos (Point Signature)

Las firmas de puntos [26] se obtienen mediante características angulares y de distancia que son extraídas de regiones construidas mediante restricciones de distancia en torno a un punto de interés.

Dado un punto p , cuando la curva de superficie se interseca con una esfera de radio r centrada en p da como resultado un contorno C (ver Figura 2.9). Los puntos que forman este contorno son entonces representados en un nuevo sistema de coordenadas local centrado en p . La orientación de los ejes viene dada por el vector normal en p (n_p), un vector de referencia (n_{ref}) y el vector obtenido mediante su producto vectorial. Todos los puntos de C se proyectan al plano tangente dando como resultado una curva C' . El vector n_{ref} se calcula como el vector unitario que apunta desde p al punto de C' que tiene la mayor distancia de proyección. De este modo, cada punto x de C se puede caracterizar por:

- La distancia signada d_x con su propia proyección en el plano tangente.
- Un ángulo de rotación en sentido horario θ_x desde el vector de referencia n_{ref} . Dependiendo de la resolución se pueden elegir diferentes incrementos $\Delta\theta$. Entonces, la firma del punto p puede expresarse como un conjunto de distancias en cada θ desde 0° hasta 360° .

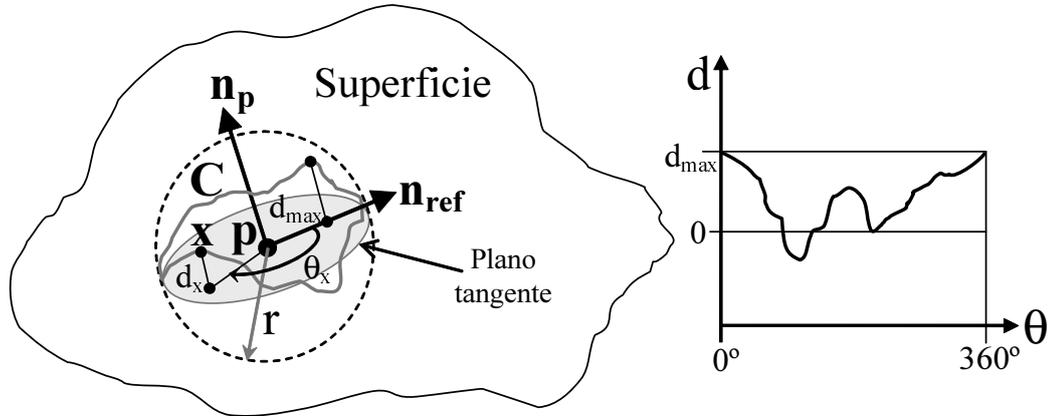


Figura 2.9: Construcción de la firma de un punto p (point signature). Se extrae un contorno C mediante la intersección de la superficie con una esfera de radio r . La distancia d de los puntos del contorno al plano tangente son representados en función del ángulo θ que usa como referencia el vector n_{ref} .

Cuando se deseen alinear dos superficies A y B se compararán las firmas de sus puntos. El proceso de emparejamiento es rápido y eficiente; sin embargo, el principal inconveniente del algoritmo es el proceso seguido para calcular el descriptor. La intersección de una esfera con la superficie no es muy fácil de obtener, sobre todo cuando la superficie viene dada en forma de nube de puntos o de malla triangular. En estos casos habrá que recurrir a interpolación de los datos, incrementando de este modo el tiempo de computación y empeorando la calidad del descriptor. Además el cálculo del vector de referencia es muy sensible al ruido, por lo que, cualquier error que se cometa va a afectar considerablemente al descriptor obtenido.

Además del cálculo del descriptor también se estima la curvatura local de cada punto de la superficie para utilizarla posteriormente en la etapa de emparejamiento global.

Tras evaluar los emparejamientos de los descriptores de ambas superficies, el algoritmo va a seleccionar solamente aquellos que satisfagan a unas condiciones prefijadas. Estas correspondencias se agruparán usando tests de consistencia geométrica y las características de curvatura que fueron extraídas previamente. A continuación se evalúa el alineamiento producido por cada grupo hasta que uno de ellos cumpla ciertas restricciones preestablecidas.

Footprints

El algoritmo basado en este descriptor selecciona, en primer lugar, todos los puntos de A y B como puntos de interés. Entonces se calculan los *footprints* [8] de cada uno de estos puntos como una cierta función vectorial o escalar (p.e. curvatura) de los puntos que componen su entorno más cercano. La particularidad del algoritmo basado en este descriptor es que no se limita a alinear las superficies utilizando un único tipo de representación de características, sino que dependiendo del tipo de superficies a alinear se van a utilizar diferentes definiciones y formas de calcular los *footprints*. En todo caso, el *footprint* elegido debe ser invariante a traslaciones y rotaciones y suficientemente descriptivo, pues el éxito de este método depende fundamentalmente de una buena definición del descriptor.

En la etapa de emparejamiento local, se seleccionan los *footprints* de A y B que contengan suficiente información y se comparan entre sí. A continuación, se seleccionan las mejores correspondencias obtenidas para la etapa de emparejamiento global, en la cual se van a asociar en diferentes grupos para determinar transformaciones euclídeas. Esas transformaciones se van a utilizar para modificar una tabla de votaciones que acumula las rotaciones y traslaciones más consistentes entre las dos superficies.

El principal inconveniente de método es que no tiene carácter general, ya que modifica el descriptor dependiendo del tipo de superficie.

Spin image

El algoritmo que usa este descriptor [52, 53] es quizás el más popular de todos los que emplean propiedades intrínsecas de la superficie; de hecho, suele considerarse como un estándar de comparación con el resto de algoritmos. Éste comienza con la selección de una serie de puntos de interés en ambas superficies. A continuación se obtienen las regiones en torno a cada uno de estos puntos utilizando como restricción el ángulo que forman las normales de los demás puntos de la superficie con la normal en dicho punto de interés. Para los puntos de cada una de estas regiones se calculan las distancias horizontales y perpendiculares que tienen con respecto a su correspondiente punto de interés.

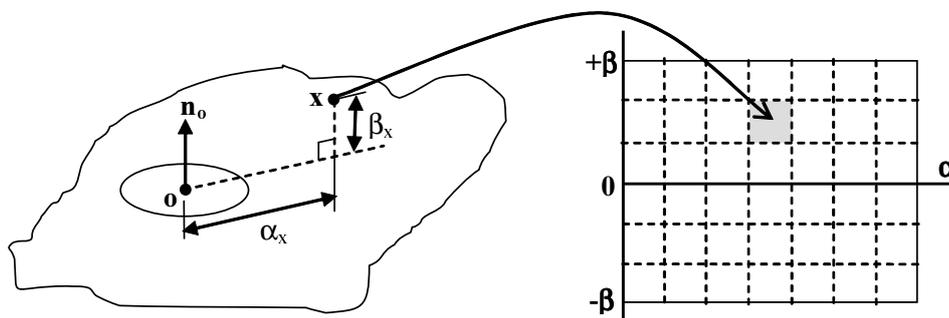


Figura 2.10: Construcción de una *spin image*. Se extraen, para cada punto \mathbf{x} de la región considerada, las distancias horizontales y perpendiculares a un punto de interés \mathbf{o} . A continuación, la celda del histograma 2D indexada por esas características es incrementada en una unidad.

A continuación se construye un histograma 2D cuyo tamaño depende del rango elegido para ambas distancias y de la resolución de las celdas (ver Figura 2.10). Cada una de sus celdas va a acumular el número de puntos de cada región que tienen los mismos valores para ambas distancias.

Dadas dos nubes de puntos A y B , se extraen las *spin-images* correspondientes a sus conjuntos de puntos de interés previamente seleccionados. A continuación se comparan todas las *spin-images* de A con todas las de B , de modo que los resultados de similitud obtenidos son almacenados en un nuevo histograma, donde los valores extremos representan las mejores correspondencias encontradas. Este nuevo conjunto es procesado en una etapa de emparejamiento global, el cual evalúa los alineamientos que se obtienen con grupos de correspondencias geoméricamente consistentes.

Histograma geométrico

El histograma geométrico [6, 7] sigue un esquema similar al de las *spin-images*. En este algoritmo se requiere un mallado previo de la nube de puntos para seleccionar como puntos de interés los centroides de las facetas de la malla. A continuación, se construyen las regiones alrededor de estos puntos usando restricciones basadas en la distancia perpendicular y en el ángulo entre normales. Estas mismas características de los puntos que forman la región se utilizan para indexar las celdas de un histograma 2D, tal y como se muestra

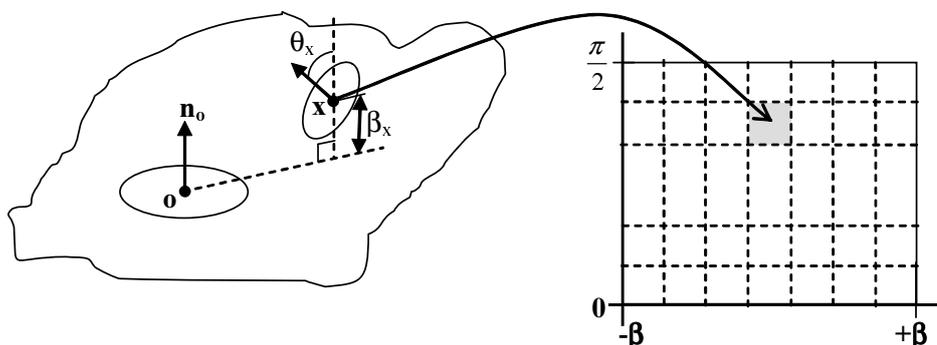


Figura 2.11: Construcción de un histograma geométrico. Se extraen, para cada punto \mathbf{x} de la región considerada, la distancia perpendicular al punto de interés \mathbf{o} y el ángulo formado por sus normales. A continuación, la celda del histograma 2D indexada por esas características es incrementada en una unidad.

en la Figura 2.11.

En la etapa de emparejamiento local del algoritmo, se comparan todos los histogramas geométricos de A con los de B . Entonces se considerará que, para cada punto de interés en A , su correspondencia en B viene determinada por el mejor resultado obtenido de todos los emparejamientos de ese punto de interés.

En una etapa posterior se evalúa la correspondencia global para determinar la mejor rotación y traslación entre A y B . Mediante dos procedimientos RAN-SAC se agrupan las correspondencias locales obtenidas en la etapa anterior y se evalúa el alineamiento que produce cada grupo para determinar si cumple unas condiciones prefijadas.

Este algoritmo se ha utilizado tanto para alineamiento 3D como para reconocimiento de objetos; sin embargo requiere superficies relativamente suaves para poder estimar de manera suficientemente los vectores normales que permitan definir las regiones y extraer sus características.

Signaturas de superficie

La signatura de superficie (*surface signature* [115]) es un descriptor que obtiene una representación 2D mediante características angulares y de distancia (ver Figura 2.12). El algoritmo de alineamiento correspondiente usa la superficie completa para crear la región alrededor del punto de interés. De este

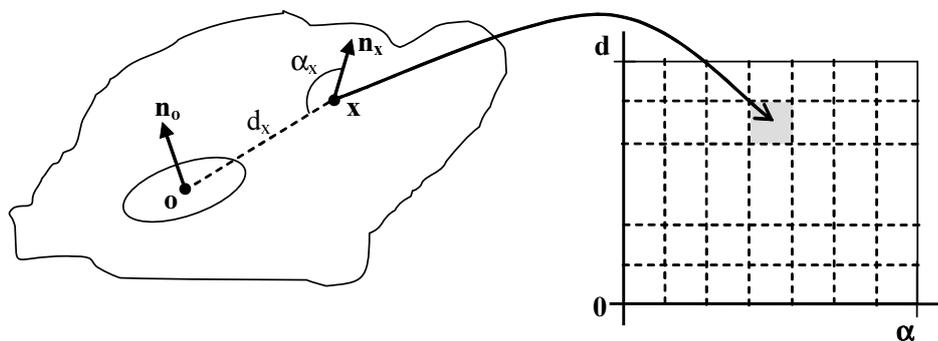


Figura 2.12: Construcción de una signatura de superficie (*surface signature*). Se extraen, para cada punto \mathbf{x} de la superficie, la distancia euclídea al punto de interés \mathbf{o} y el ángulo formado por el vector que une ambos puntos y la normal de ese punto \mathbf{x} . A continuación, la celda del histograma 2D indexada por esas características es incrementada en una unidad.

modo se emparejan las distintas representaciones de la superficie A con las de la superficie B . Las signaturas de superficie que satisfagan una condición prefijada se seleccionan como correspondencias locales. Posteriormente, en la etapa de emparejamiento global se agrupan las correspondencias locales que sean geoméricamente consistentes y se verifican los alineamientos que produce cada grupo. La transformación rígida vendrá determinada por el mejor de todos ellos.

Este algoritmo va a ser dependiente del tipo de superficie, ya que selecciona puntos de interés de alta curvatura donde se calculan los descriptores. Esto hace que sea más difícil encontrar correspondencias para superficies con pocas características topológicas.

Imágenes de Forma Armónica (HSI)

Esta representación [116] está basada en la teoría matemática de mapas armónicos. Partiendo de superficies representadas mediante mallas triangulares, se extraen parches de superficie creados mediante restricciones de distancia en torno a puntos seleccionados aleatoriamente. Para cada uno de estos parches con topología de disco, y dado un dominio planar 2D, se crea un mapa armónico tanto para los bordes como para el interior. Por consiguiente, mediante

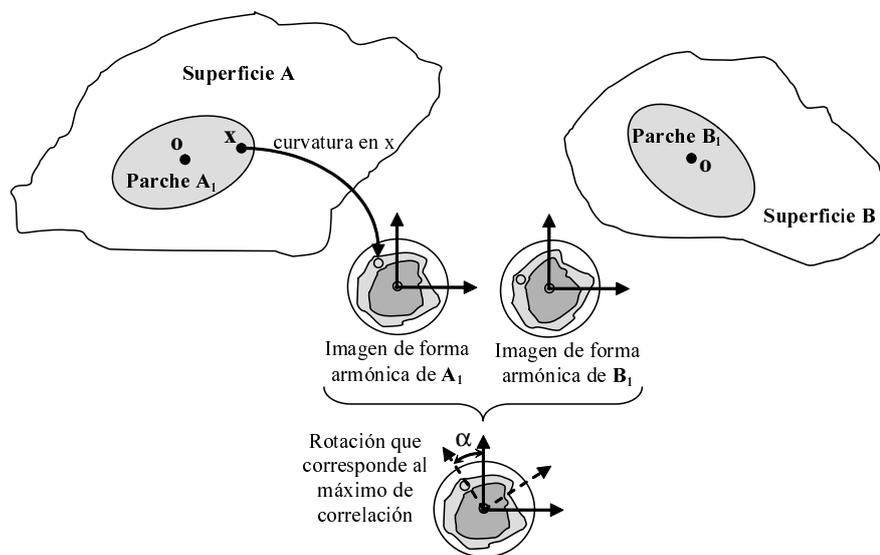


Figura 2.13: Emparejamiento de dos imágenes de forma armónica. El máximo de correlación determina la mejor correspondencia y la rotación que alinea los parches.

este mapa, habrá una correspondencia uno a uno entre los puntos del parche de superficie 3D y la imagen armónica resultante. Usando esta relación de correspondencia se pueden crear las HSI (*Harmonic Shape Images*) asociando características de forma calculadas en cada punto del parche de superficie con su punto correspondiente de la imagen armónica. Las características que se mapean en este caso están basadas en curvatura.

Dadas dos superficies A y B de un mismo objeto, se van a establecer emparejamientos entre sus respectivas HSI, de modo que el que proporcione el mayor valor de correlación determinará la mejor correspondencia entre las dos superficies. Además, se podrá obtener a partir de esta información, la rotación y traslación que permite alinear A y B . Girando una de las dos HSI sobre la otra, se puede calcular la mejor rotación entre ellas (ver Figura 2.13). Y la mejor traslación puede calcularse desplazando el centroide del parche de A al centroide del de B .

La principal ventaja de este algoritmo es que no necesita una etapa de emparejamiento global. Por contra, sus inconvenientes principales son el laborioso proceso de obtención de las imágenes armónicas (que implica además un ma-

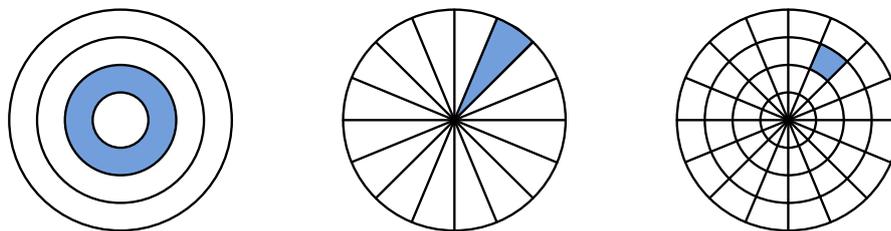


Figura 2.14: Conjunto de histogramas 2D propuesto por *Ankerst et al.* para descomponer el espacio 2D.

llado de la nube de puntos) así como su elevado tamaño, lo cual, hace que su aplicación a problemas de alineamiento de superficies con bajo solapamiento no sea muy aconsejable.

Contextos de forma 3D

Los contextos de forma se definieron en un principio por *Belongie et al.* [11] para hallar la similitud entre imágenes 2D. Los contextos de forma se crean en torno a puntos de referencia de la imagen capturando la distribución de los puntos restantes. Posteriormente, *Körtgen* extendió estos contextos de forma a una descripción de forma 3D para la que definió un conjunto de descriptores llamados contextos de forma 3D [62].

Para calcular estos descriptores, se muestrean N puntos de los bordes del objeto y, a continuación, se calculan los vectores que van desde uno de esos puntos a todos los demás puntos de la superficie. Considerando la distribución sobre las posiciones relativas, se calcula para el punto un histograma aproximado de las coordenadas relativas de los $N-1$ puntos restantes. Este histograma es una adaptación de los histogramas de forma de *Ankerst* [5] centrados en el punto de muestra (ver Figura 2.14).

Aunque este descriptor ha sido utilizado fundamentalmente para búsqueda de formas 3D en bases de datos, lo hemos incluido en esta revisión debido a que utiliza una distribución del espacio similar a la que utilizaremos para definir nuestro descriptor.

Imágenes locales de rango log-polar

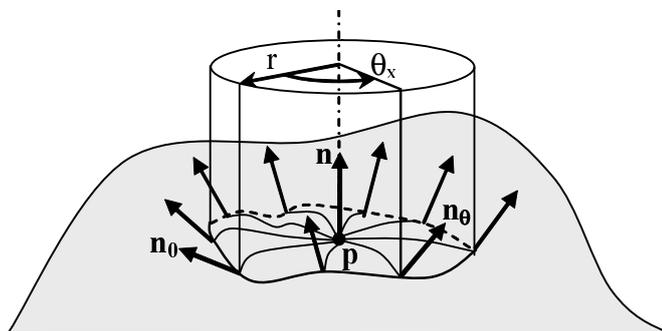
Este método empareja vectores de características invariantes generados a partir de imágenes locales de rango obtenidas mediante la transformación log-polar [72].

Las imágenes locales de rango (LRI) se van a obtener a partir de las muestras generadas por un algoritmo de muestreo basado en SDF (Signed Distance Field [73]) que reducirá la resolución de las imágenes de rango originales. A continuación se aplicará a las LRI la transformación log-polar y se extraerá de la imágenes resultantes un vector de características invariante a rotación. Para obtener estos vectores invariantes se utilizan dos métodos diferentes: el espectro de potencia de Fourier (FPS) y la autocorrelación local de orden más alto (HLAC). Finalmente, estos vectores son comprimidos para reducir su dimensión.

Por consiguiente, la correspondencia de un punto se va a llevar a cabo encontrando el vector de características más cercano a su vector asociado. A continuación se aplica un algoritmo RANSAC para extraer los pares de puntos potencialmente válidos de entre todos los que se obtuvieron en la fase de correspondencia. Mediante este conjunto de correspondencias se calculará la transformación rígida que alinee de forma aproximada las superficies.

Los principales inconvenientes de este método es que al utilizar la transformación log-polar la descripción de la superficie varía de forma logarítmica con la distancia al punto de interés, por lo que falla cuando el radio del entorno del punto es grande. Si para evitar este problema se considera un entorno más pequeño, la densidad de puntos debería ser más grande, ya que, de lo contrario, habría píxeles de la LRI que no serían "rellenados". Pero como el algoritmo genera LRI's en todos los puntos de la imagen de rango, el tiempo de computación crecería considerablemente.

Aunque tanto la transformación de la imagen para convertirla en invariante a rotaciones, como la compresión final de los datos para reducir su dimensión, contribuyen a reducir el tiempo de computación de la etapa de correspondencia, los problemas de pérdida de poder descriptivo con el radio del entorno se van a ver acrecentados, lo cual restará eficacia al algoritmo.

Figura 2.15: Concepto de representación *splash*.

Splash

La representación *splash* [97] utiliza pequeños parches de superficie donde se puedan calcular de manera fiable propiedades diferenciales. Se basa en una descomposición radial de las normales de superficie para obtener un mapa gaussiano local que describe la distribución de la orientación de la superficie a lo largo de un círculo geodésico de radio r (ver Figura 2.15). La representación de este descriptor se obtiene mediante tres características angulares que definen la normal de cada punto del contorno geodésico y que pueden combinarse para formar una curva tridimensional compacta. Esta curva 3D es aproximada por un conjunto de segmentos conectados que serán posteriormente codificados mediante el radio del entorno considerado y un conjunto de valores de curvatura y de torsión (tantos como el número de segmentos de la aproximación poligonal).

El algoritmo de alineamiento va a seleccionar puntos de interés en A y B basándose en criterios de estabilidad de la normal y va a crear regiones en torno a esos puntos basándose en restricciones de distancia. Las representaciones *splash* de A son codificadas y almacenadas en una tabla. El código de cada *splash* de B es entonces usado para indexar la tabla y devolver su correspondencia hipotética. Posteriormente se aplicará un algoritmo de clustering para agrupar correspondencias locales. Cada grupo se verifica mediante restricciones intrínsecas prefijadas para, finalmente, quedarse con el que produce la mejor correspondencia global. De esta manera no va a ser necesaria una evaluación del alineamiento que producen los grupos.

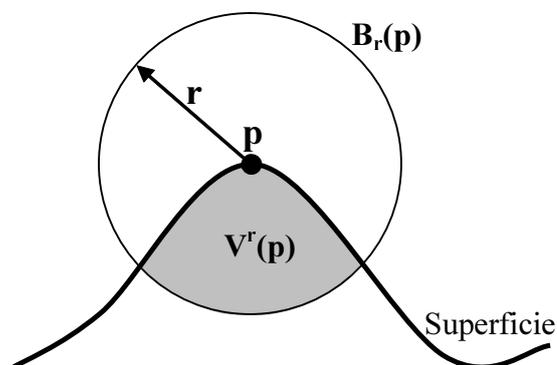


Figura 2.16: Obtención del descriptor de integral de volumen. Se calcula el volumen encerrado por una bola de radio r centrada en el punto p y el interior de la superficie.

Debido a que este algoritmo realiza un suavizado previo de las superficies originales para calcular las características de curvatura, será necesario establecer en cada caso una serie de parámetros tales como los niveles de cuantización de las características representadas o las tolerancias utilizadas para las aproximaciones poligonales.

Integral de volumen

Este descriptor [41], invariante a transformaciones rígidas, está basado en operaciones de integración del entorno del punto de interés. Para cada punto, se va a obtener el volumen de la intersección de la bola B_r con el interior del objeto definido por el conjunto de datos de entrada (ver Figura 2.16).

El descriptor se utiliza para identificar zonas características en una de las dos nubes de puntos (si se debe alinear con un modelo se hace sobre la nube de puntos parcial) y así seleccionar puntos de interés que puedan ser potencialmente correspondidos. La búsqueda de correspondencias en la otra nube se va a realizar sobre todos sus puntos, de manera que así se intenta solucionar el problema de que estos sean incompatibles con los seleccionados en la otra nube. El algoritmo va a usar una medida de calidad de las correspondencias basada solamente en el cálculo de características intrínsecas de la superficie.

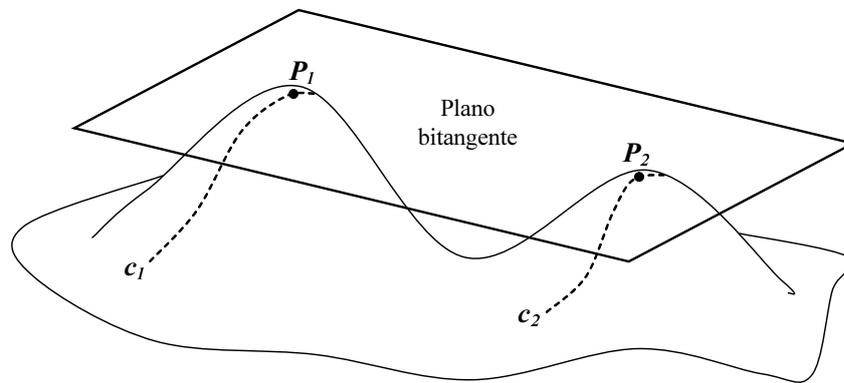


Figura 2.17: Curvas bitangentes. Los puntos P_1 y P_2 son puntos bitangentes pues comparten el mismo plano tangente.

2.4.2. Basados en correspondencia de curvas, estructuras y grafos

Curvas bitangentes

Este algoritmo [114] usa información de la curvatura para determinar la correspondencia entre dos superficies. Se extraen pares de curvas bitangentes de cada una de las superficies que se desea alinear (un ejemplo de este tipo de curvas se muestra en la Figura 2.17). A continuación se calcula la representación de cada par mediante características locales; éstas van a ser las longitudes de los arcos entre cada par de curvas bitangentes. Entonces se emparejarán las representaciones obtenidas de la superficie A con las de la superficie B, de modo que los pares que satisfagan ciertas restricciones serán seleccionados para verificar el alineamiento que producen. La transformación que logre cumplir las condiciones prefijadas será la elegida.

El principal inconveniente de este método es que necesita que las superficies sean relativamente suaves para poder extraer las curvas bitangentes. Además, el algoritmo no funcionará bien en aquellas superficies que no presenten variaciones de curvatura significativas.

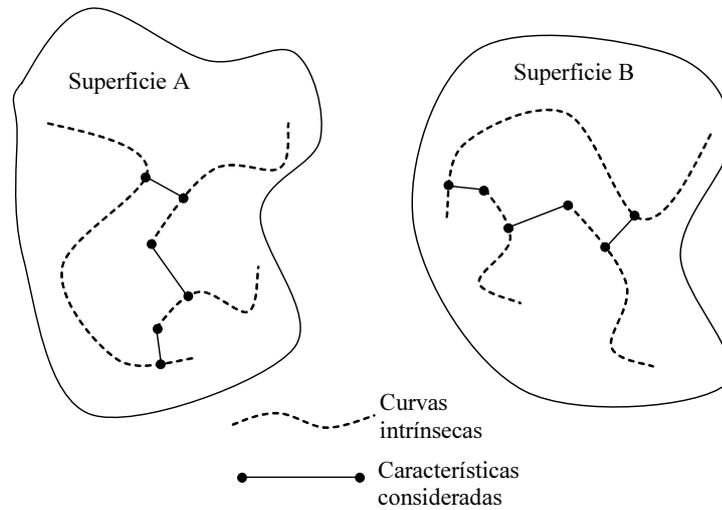


Figura 2.18: Las características que se utilizan para emparejar las superficies son las distancias mínimas entre pares de curvas intrínsecas (curvas de curvatura gaussiana de media cero).

Curvas intrínsecas

Este algoritmo [61] extrae curvas de curvatura gaussiana de media cero en ambas superficies. Las características que se va a utilizar para buscar correspondencias son las distancias mínimas entre todos los pares de curvas que pueda obtenerse en cada superficie (ver ejemplo en la Figura 2.18).

Estas distancias van a utilizarse para establecer emparejamientos entre las dos superficies. Aquellos que satisfagan ciertas condiciones prefijadas se utilizarán para la fase de emparejamiento global, la cual se lleva a cabo mediante el agrupamiento de esas correspondencias y la verificación de las diferentes transformaciones que producen. Por consiguiente, se elegirá la que produzca el mejor alineamiento entre las superficies.

Al igual que el algoritmo de curvas bitangentes, no funciona bien en superficies en las que no haya una variación significativa de la curvatura.

DARCES basada en RANSAC (RBD)

DARCES [96] es un acrónimo de Data-Aligned Rigidity Constrained Exhaustive Search (búsqueda exhaustiva restringida por la rigidez de los datos

alineados). Dicha búsqueda está basada en el método RANSAC que a su vez es otro acrónimo de RANdom SAmple Consensus (consenso para muestreo aleatorio).

El algoritmo va a seleccionar un conjunto de puntos de interés en A y todos los puntos de B . Entonces se aplica un procedimiento RANSAC para seleccionar aleatoriamente un *punto semilla* de la lista de puntos de interés de A . A continuación se construye en torno a ese punto una estructura, utilizando para ello un conjunto de puntos de interés que cumplan ciertas restricciones geométricas.

El siguiente paso consiste en intentar construir, para cada punto de B , la misma estructura que en A (ver Figura 2.19). Para los grupos que sean consistentes se alinearán ambas estructuras para así verificar el alineamiento global de las superficies. Este proceso de emparejamiento y evaluación se va a repetir hasta que sean satisfechas ciertas condiciones prefijadas.

Este método RBD es bastante flexible, ya que no requiere que las superficies estén representadas por mallas o que tengan una determinada topología. Por el contrario, necesita que la resolución de ambas superficies sea alta para poder calcular con precisión las estructuras que han de emparejarse.

Emparejamiento de Puntos intrínsecos

Este algoritmo [25] es una versión más antigua del basado en firmas de puntos; sin embargo éste no está basado en una estrategia de emparejamiento local de características. Aunque se van a calcular características de curvatura principal para todos los puntos de ambas superficies, esta información se va a utilizar para el procedimiento de emparejamiento global.

Las correspondencias se agrupan usando tests de consistencia geométrica y características de curvatura de puntos (ver Figura 2.20). A continuación se evalúa el alineamiento que resulta de cada grupo hasta que uno de ellos cumpla unas condiciones prefijadas.

Este algoritmo requiere superficies relativamente suaves para poder extraer información de curvatura, y también resoluciones altas que permitan obtener emparejamientos precisos de los grupos de correspondencias.

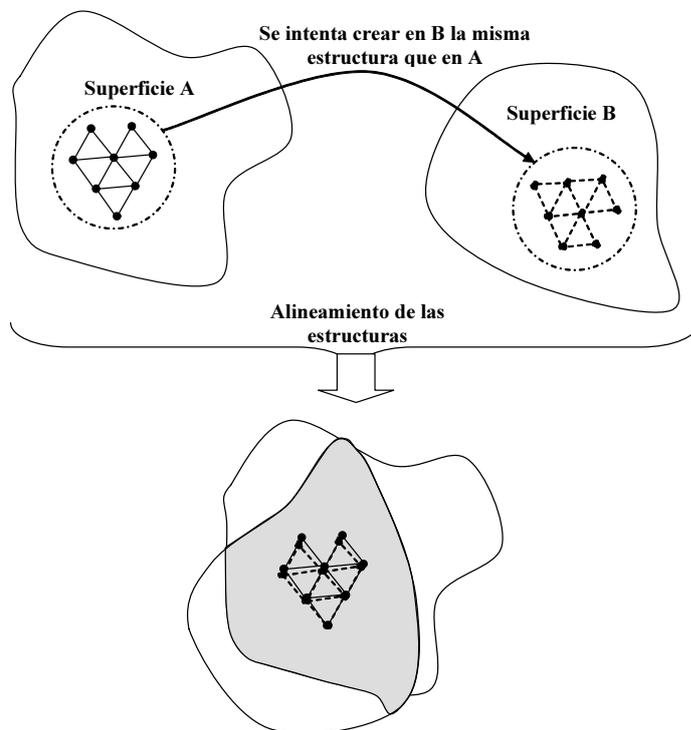


Figura 2.19: Procedimiento de búsqueda de correspondencia y alineamiento de las superficies mediante el método RBD (RANSAC-based DARCES).

Emparejamiento de grafos

En este método [23], cada punto de la superficie A está contenido dentro de un grafo que describe la superficie. Todos los puntos de B van a considerarse como puntos de interés. El procedimiento de emparejamiento global es similar al RBD, pues va a intentar encontrar en B el mismo grafo que en A , aplicando restricciones geométricas entre los nodos del grafo. La diferencia entre los dos métodos es que, en este caso, no se va a llevar el alineamiento de las superficies para probar la validez de los grupos. Lo que se va a hacer es utilizar el error acumulado en las diferencias geométricas entre los nodos del grafo como única medida para evaluar los grupos.

Como ocurría en el método RBD, la resolución de las superficies debe ser relativamente alta para obtener de manera precisa los grafos que se van a emparejar.

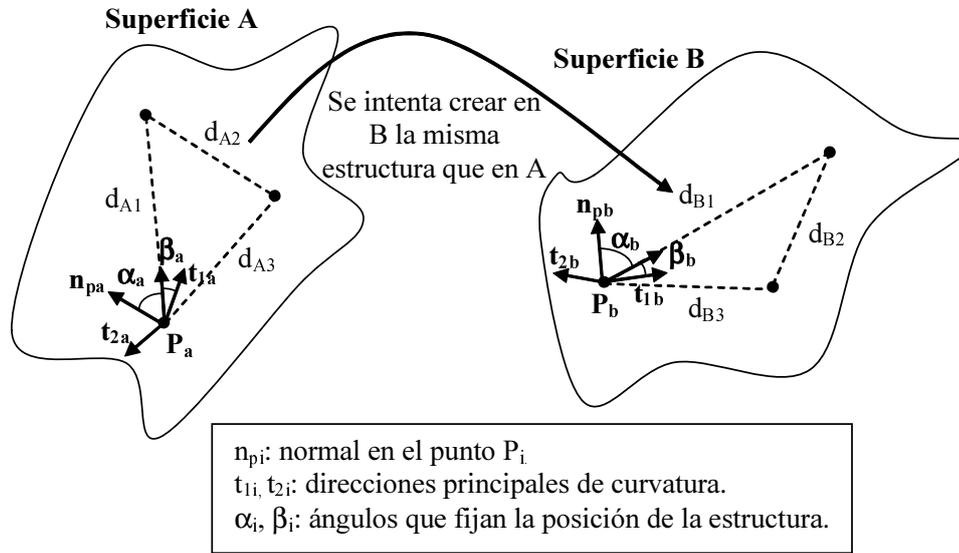


Figura 2.20: Emparejamiento global de puntos intrínsecos. Se aplican restricciones geométricas (los ángulos α_i y β_i que fijan la posición de la estructura se deben conservar, así como las distancias entre sus puntos) y restricciones de curvatura (t_{1b} y t_{2b} deben ser similares a t_{1a} y t_{2a}).

2.4.3. Otros métodos

Correspondencia basada en la teoría de campo medio

Este método [3] extrae pares de puntos de ambas superficies. Cada uno de estos pares tiene atributos que lo identifican (por ejemplo, la distancia entre los puntos). Utilizando un código de atributos para cada par de puntos de la superficie A se rellena una tabla. Entonces, los códigos de atributos de los pares de B se emplean para indexar la tabla anterior. A cada uno de estos emparejamientos se les va a asignar una probabilidad relacionada con su grado de validez. Estas probabilidades serán actualizadas mediante la teoría de campo medio (MFT). Las probabilidades MFT finales se usan para extraer el mejor grupo de pares correspondientes entre las dos superficies. Mediante este grupo se podrá calcular la transformación rígida que permite alinear A con B .

Aunque este método no requiere que las superficies tengan una topología determinada, sí necesitará que la resolución con la que han sido adquiridas sea alta, para así poder extraer con precisión los atributos de los pares de puntos.

Imagen de atributo esférico (SAI)

Cada una de las superficies se va a codificar mediante una única representación. El algoritmo va a comenzar construyendo mallas regulares para ambas superficies. Entonces, cada vértice de la malla y sus tres puntos adyacentes en la malla se usan para calcular la característica de curvatura de ese punto: el ángulo simplex. Este ángulo se va a almacenar en una representación esférica conocida como imagen de atributo esférico (Spherical Attribute Image [46]).

Por lo tanto, si se desean alinear dos superficies, A y B , se rotará la representación SAI de A hasta hacerla coincidir con la SAI de la superficie B . La transformación resultante entre las imágenes esféricas se traducirá al espacio euclídeo para obtener la transformación rígida que alinea ambas superficies.

El principal inconveniente de este método es que necesita que las superficies tengan una topología adecuada para poder ser transformadas a estas imágenes esféricas.

Correspondencia basada en algoritmos genéticos

Este método [55] usa un algoritmo genético para resolver el problema de la correspondencia. La clave está en definir el vector que contiene los n índices de las correspondencias entre ambas superficies. Para medir la calidad de cada solución potencial se ha de determinar una función *fitness* basada en cuatro invariantes entre dos pares de correspondencias. Esos invariantes van a calcularse mediante los vectores de posición y los vectores normales de dos puntos de la misma superficie. Utilizando estos invariantes se va a determinar la calidad de la correspondencia mediante un análisis del error de distancia y el error en los parámetros de las normales.

Una vez que se ha definido la función *fitness*, se fijan las probabilidades de cruce y mutación.

Finalmente se debe definir el criterio de parada. El autor utiliza tres estrategias diferentes: (a) colocar un porcentaje de correspondencias buenas; (b) supervisar la función *fitness* para que no se incremente después de un cierto número de iteraciones; (c) contar iteraciones hasta que sobrepasen un cierto número.

Cuando el algoritmo termina, la transformación euclídea puede calcularse porque el cromosoma que maximiza la función *fitness* contiene las correspondencias entre puntos.

Como en la mayoría de los métodos basados en algoritmos genéticos, los resultados que se obtienen son bastante buenos pero el tiempo de computación es excesivo, especialmente cuando las superficies tienen muchos puntos, pues existirá una gran cantidad de correspondencias potenciales.

2.5. Algoritmos de alineamiento fino

Se van a llamar algoritmos de alineamiento fino aquellos en los que se conoce una estimación de la transformación que alinea las dos nubes de puntos y se utiliza como punto de partida para llevar a cabo una serie de iteraciones que converjan hacia una solución más precisa. Esa estimación inicial puede venir de un algoritmo de alineamiento aproximado, del seguimiento de la posición del escáner o de una entrada de usuario.

Con el objetivo de refinar la transformación euclídea, se va a minimizar una función de distancia. Algunos autores utilizan la distancia entre puntos correspondientes, mientras que otros utilizan la distancia desde un punto de una nube al plano tangente a su punto más próximo en la otra nube.

La mayoría de los algoritmos de alineamiento fino están basados en el algoritmo ICP (Iterative Closest Point, es decir, algoritmo iterativo de punto más próximo) desarrollado por *Besl* y *McKay* [14] y modificado por *Chen* y *Medioni* [22]. Otras variantes del ICP pueden encontrarse en [88]. Debido a que vamos a utilizar y mencionar este algoritmo en algunas partes de esta tesis, haremos una descripción básica de su funcionamiento.

Todos los algoritmos basados en el ICP siguen básicamente los siguientes pasos:

1. **Encontrar correspondencias:**

Para cada punto p_i de la primera nube, se va a buscar el punto más próximo q_i en la segunda nube. Normalmente, se suele aplicar algún umbral a la distancia entre p_i y q_i , de modo que los puntos que estén

demasiado lejos de sus correspondencias serán descartados como si se tratasen de puntos que se encuentren fuera del área de solapamiento.

2. Calcular los parámetros de transformación:

Dado el conjunto de pares de puntos correspondientes $\{p_i, q_i\}$ se calcula la transformación (R,t) que minimiza la suma de las distancias cuadráticas entre puntos correspondientes. Si la distancia medida es la distancia euclídea entre puntos, entonces la transformación rígida puede hallarse en forma cerrada [33]; en caso contrario se puede utilizar una aproximación de la transformación [79] o una búsqueda local [22].

3. Mover la primera nube de puntos e iterar:

Se aplica la transformación calculada a los puntos de la primera nube y se realiza una nueva iteración mientras no se satisfaga el criterio de parada. El algoritmo va a finalizar cuando el cambio en el residuo de distancias cuadráticas sea suficientemente pequeño.

El algoritmo ICP converge a un mínimo local de la medida de distancia, de ahí que sea necesario disponer de una estimación de la transformación que esté suficientemente próxima al alineamiento correcto.

Se ha demostrado tanto teórica [79] como experimentalmente [88] que la velocidad y el conjunto de transformaciones iniciales que aseguran la convergencia del ICP dependen fuertemente de la elección de los puntos correspondientes y de la medida de distancia que se minimiza cuando se calcula la transformación.

2.6. Selección de puntos de interés

En la mayoría de las aplicaciones que utilizan nubes de puntos obtenidas mediante escáneres el tamaño de éstas es considerablemente grande, por lo que la búsqueda de puntos correspondientes entre dos de estos inmensos conjuntos de datos puede llegar a ser realmente costosa en términos computacionales. Por lo tanto, sería más conveniente utilizar un subconjunto de puntos en el que poder simplificar dicha búsqueda, de modo que las correspondencias que

se encuentren puedan utilizarse para el alineamiento de las superficies (tal y como se hace con pares de imágenes [66]). Lo más usual es escoger puntos que tengan alguna característica que los distinga del resto [47, 110], por lo que normalmente reciben el nombre de "puntos de interés".

La selección de puntos de interés conlleva un análisis de ciertas propiedades locales del entorno de los puntos de las superficies a alinear, aunque también podría ser necesario llevar a cabo un análisis más global. En todo caso, esos puntos característicos se van a tener que seleccionar de manera independiente en ambas superficies, pues lo contrario implicaría conocer la correspondencia entre sus puntos. Por lo tanto, la principal dificultad va a ser encontrar por separado dos conjuntos de puntos de interés que sean consistentes entre sí, ya que, si no se diera tal consistencia, sería imposible encontrar correspondencias entre ellos.

Como consecuencia de esto, los algoritmos suelen utilizar fundamentalmente dos estrategias: seleccionar más puntos de los necesarios para garantizar que habrá puntos que sean comunes o llevar a cabo complicados análisis que detecten puntos con características salientes [40] bajo la suposición de que algunos de esos puntos se encontrarán en ambas superficies (lo cual puede no producirse cuando se trata de nubes de puntos parciales). Otros algoritmos siguen una estrategia diferente que consiste en analizar las dos superficies al mismo tiempo, lo cual, como ya se ha dicho, significa llegar a resolver en cierto modo el problema de correspondencia, haciendo que el método sea bastante costoso en términos de computación.

En cualquiera de los casos, asegurar la consistencia de los puntos de interés es un proceso que consume tiempo, por lo que, en nuestro caso, vamos a tratar de diseñar un algoritmo de búsqueda de correspondencias (Sección 6.4) que no dependa tanto de esa selección de los puntos de interés. De este modo, la selección de puntos de interés se va llevar a cabo mediante un algoritmo sencillo que obtiene una serie de puntos de interés en base a la potencial estabilidad de su vector normal y a su cercanía a puntos característicos de la imagen de rango. Aunque con este proceso de selección no se va a asegurar, en principio, la consistencia de los dos conjuntos de puntos de interés, esto no va a restar demasiada eficacia a la búsqueda de correspondencias, ya que, como se expli-

cará en la Sección 6.4, estos dos conjuntos de puntos no van a ser más que "puntos de partida" y "puntos objetivo" del algoritmo.

2.7. Medidas de similitud

Cuando se utilizan descriptores basados en características del entorno de un punto de interés, será necesario compararlos para decidir si los puntos son correspondientes o no. Por lo tanto, se deberá disponer de una medida que proporcione el grado de similitud entre dos descriptores.

El uso de una medida de similitud u otra debería depender de la naturaleza del descriptor que se utiliza para realizar la correspondencia.

Cuando los descriptores corresponden a un vector de características es posible utilizar medidas como la distancia euclídea, *Manhattan*, *Chebyshev* o *Mahalanobis* [68]. En cambio, van a ser menos útiles cuando se consideren descriptores que representan la geometría del entorno del punto de interés mediante una representación 2D (como por ejemplo las *spin images* o el descriptor que se propone en esta tesis). Cuando sea éste el caso y se traten de alinear dos nubes de puntos parciales, los descriptores que se obtengan sólo van a tener ciertas partes comunes, de modo que va a haber elementos en ambas matrices que no tengan correspondencia en la otra matriz. Lo que suele hacerse en estos casos es comparar solamente las áreas que se solapan en ambos descriptores, de modo que, para esos elementos comunes, se aplicará una medida de similitud. En la mayoría de los algoritmos que utilizan descriptores 2D esta medida suele basarse en la correlación cruzada normalizada (*CCN*) en el área solapada. Dados dos descriptores bidimensionales, A y B , que se solapan en una región S , esta medida se define como:

$$CCN = \frac{\sum_{(i,j) \in S} (A(i,j) - \bar{A}) \cdot (B(i,j) - \bar{B})}{\sqrt{\sum_{(i,j) \in S} (A(i,j) - \bar{A})^2} \cdot \sqrt{\sum_{(i,j) \in S} (B(i,j) - \bar{B})^2}} \quad (2.4)$$

Siendo \bar{A} y \bar{B} los valores medios de los descriptores en la región solapada

S.

Algunos algoritmos [52, 116] también consideran en su medida de similitud basada en *CCN* el número de puntos utilizados para evaluar la Ecuación 2.4.

Otras medidas de similitud son la información mutua [111, 67], la entropía conjunta [27, 99] o el criterio de *Woods* [113], que son utilizadas con éxito en alineamiento de imágenes médicas multimodales (se puede consultar una revisión más amplia en [95, 82]). Sin embargo, debido a que se basan en funciones de distribución de probabilidad de los píxeles y los descriptores 2D se pueden considerar como imágenes sintéticas de baja resolución, que incluso pueden presentar bajo solapamiento, su aplicación, en este caso, no es muy recomendable (ver [87] para un análisis más detallado).

En este trabajo de tesis se va a utilizar un nuevo descriptor 2D para alinear nubes de puntos 3D, pero en vez de emplear una medida basada en el coeficiente de correlación de las áreas solapadas de dos descriptores se va a definir una nueva medida que aproveche la propia estructura de éstos para determinar su similitud.

Capítulo 3

Descriptor CIRCÓN

3.1. Introducción

Uno de los principales inconvenientes que hemos observado tras el análisis de los descriptores más utilizados en el estado del arte, es que, aunque muchos de ellos están basados en propiedades geométricas del entorno del punto de interés, la evaluación de su similitud no tiene una relación directa con la distancia entre las nubes de puntos. Y dado que es en base a dicha distancia como determinaremos qué transformación euclídea alinea mejor dos nubes de puntos, sería más conveniente, en nuestra opinión, utilizar un descriptor que representase mejor la geometría del entorno. Además, con esos descriptores analizados, es necesario disponer de al menos tres correspondencias buenas para determinar una transformación euclídea aproximada.

Otro inconveniente asociado a algunos descriptores es que, al término de la etapa de emparejamiento local, se presente un porcentaje considerable de falsas correspondencias. Esto suele producirse por un bajo poder discriminante del descriptor debido a su baja dimensión y a una elección de una medida de similitud que no es lo suficientemente apropiada.

Por consiguiente, las principales características que vamos a demandar al descriptor para una correcta búsqueda de correspondencias son las siguientes:

- Debe describir la geometría del entorno de los puntos de interés.

- Debe tener un alto poder descriptivo, de manera que se puedan discriminar de manera adecuada los entornos de los puntos de interés y no aparezcan falsas correspondencias.
- Debe permitir el uso de una medida de similitud basada en distancias entre puntos de la nube. Esta medida debe favorecer una correcta diferenciación de las correspondencias evaluadas, de manera que puedan clasificarse en función del alineamiento que éstas producen. Esto deberá evitar la presencia de falsas correspondencias.
- Debe permitir el cálculo de una transformación euclídea en base a una sola correspondencia.
- Debe ser útil tanto para modelado 3D (alineamiento de dos escaneos desde dos puntos de vista diferentes del objeto) como para reconocimiento de objetos 3D (alineamiento de la nube de puntos de la escena y la del modelo).

3.2. Notación

Para evitar posibles confusiones en las secciones de este capítulo y en los capítulos posteriores, vamos a describir la notación que se va a utilizar para representar los puntos, vectores, matrices o correspondencias entre puntos que van a aparecer en las expresiones matemáticas. Esta notación es muy similar a la utilizada por *Craig* en [28] así como por otros autores para representar transformaciones entre sistemas de coordenadas.

- ${}^w p_q$: Se trata de un punto de una nube $\{p\}$ al cual le corresponde un índice q que le identifica dentro de dicha nube. El superíndice w indica que el punto está expresado en un sistema de coordenadas global (mundo). Sin embargo, se prescindirá de este superíndice cuando se trate de variables de un algoritmo, ya que éstas pueden ser modificadas para representar a los puntos de una nube en sistemas de coordenadas diferentes. También se prescindirá de él cuando se necesite hacer referencia a los puntos sin importar el sistema de coordenadas en el que estén expresados.

- \vec{n}_q : Vector normal en el punto ${}^w p_q$.
- ${}^q p_x$: Con esta notación se indica que un punto de la nube $\{p\}$, al cual le corresponde un índice x , está expresado en el sistema de coordenadas local centrado en el punto de la nube con índice q . Por lo tanto, para referirnos a este sistema de coordenadas utilizaremos el superíndice q .
- ${}^w R_q$: Matriz de rotación formada por los vectores unitarios del sistema de coordenadas local q expresados en el sistema de coordenadas global (W). Su inversa (que es igual a su traspuesta) la expresamos como ${}^q R_W$.
- ${}^q T_w$: Matriz que permite transformar (mediante rotación y traslación) un punto expresado en el sistema de coordenadas global W al sistema de coordenadas local q . Su matriz traspuesta, ${}^w T_q$, realiza la transformación inversa.
- $R_X(\gamma)$, $R_Y(\beta)$ y $R_Z(\alpha)$: Matrices que representan rotaciones con ángulos γ , β , α sobre los ejes X, Y, Z de un sistema de coordenadas.
- Cuando nos refiramos a las dos nubes de puntos utilizaremos subíndices 1 y 2: $\{p_1\}$ y $\{p_2\}$. La transformación euclídea que se obtenga será aplicada sobre la nube 1 para alinearla con la nube 2.
- $a \leftrightarrow b$: Ésta será la notación utilizada para referirse a una correspondencia entre un punto de la nube 1 con índice a , ${}^w p_{1a}$, y un punto de la nube 2 con índice b , ${}^w p_{2b}$. Se supondrá que las normales de ambos puntos, \vec{n}_{1a} y \vec{n}_{2b} , se han calculado previamente y por lo tanto son conocidas.
- ${}^{w_2} T_{w_1}$: Matriz de transformación que relaciona los sistemas de coordenadas globales, W_1 y W_2 en el que estaban expresadas originalmente las nubes de puntos. Cuando ésta matriz de transformación se haya calculado a partir de una correspondencia $a \leftrightarrow b$ será denotada por ${}^{w_2}_{b \leftrightarrow a} T_{w_1}$.
- ${}^{w_2}_{b \leftrightarrow a} p_{w_1,org}$: Vector de traslación de la matriz ${}^{w_2} T_{w_1}$ que expresa el origen del sistema de coordenadas W_1 en el s.d.c. W_2 .

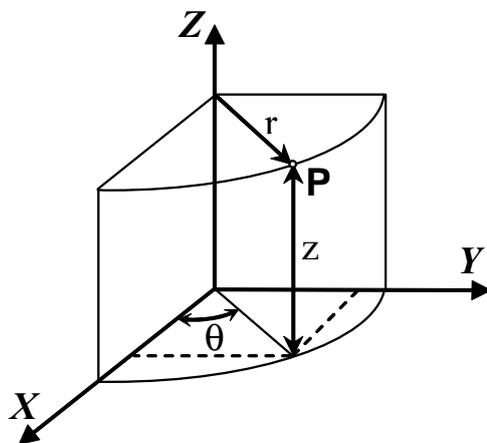


Figura 3.1: Coordenadas cilíndricas

3.3. Construcción del Descriptor

3.3.1. Coordenadas Cilíndricas

Como se vio en el Estado del Arte, los descriptores empleados habitualmente para alineamiento de nubes de puntos utilizan diferentes propiedades geométricas para describir el entorno local de los puntos de interés seleccionados.

El descriptor propuesto en esta tesis describe directamente la geometría de dicho entorno basándose en coordenadas cilíndricas pues, como se verá a continuación, éstas permiten clasificar los puntos en función de su radio, su ángulo y su altura.

Como puede deducirse de la Figura 3.1, las coordenadas cilíndricas (r, θ, z) están relacionadas con las coordenadas cartesianas (x, y, z) mediante las siguientes expresiones,

$$r = \sqrt{x^2 + y^2} \quad (3.1)$$

$$\theta = \tan^{-1} \left(\frac{y}{x} \right) \quad (3.2)$$

$$z = z \quad (3.3)$$

Notar que la Ecuación (3.2) queda indefinida para $r=0$. Esta ambigüedad en el ángulo se considerará posteriormente cuando se genere el descriptor.

3.3.2. Sistema de Coordenadas Local

Para obtener el descriptor asociado a un determinado punto de interés ${}^w p_q$, se han de transformar los puntos de la nube a un sistema de coordenadas local centrado en ${}^w p_q$ que tiene como eje Z_q su vector normal asociado \vec{n}_q . El eje X_q se va a elegir de modo que sea perpendicular al eje Y_w del sistema de coordenadas de referencia y a la normal en ${}^w p_q$. De este modo el eje Y_q queda determinado por el producto vectorial de los vectores unitarios de X_q y Z_q . Este criterio fija una referencia única para los ángulos de giro sobre el eje Z_q (es decir, sobre la normal \vec{n}_q), lo cual va a facilitar posteriormente el cálculo de la transformación euclídea asociada a esa correspondencia.

Por lo tanto, la matriz de rotación entre el s.d.c. local y el s.d.c. global se construirá como sigue:

$${}^w R_q = \begin{bmatrix} \bar{x}_q & \bar{n}_q \times \bar{x}_q & \bar{n}_q \end{bmatrix} \quad (3.4)$$

Siendo \bar{x}_q determinado por la siguiente expresión:

$$\bar{x}_q = \frac{\bar{y}_w \times \bar{n}_q}{\|\bar{y}_w \times \bar{n}_q\|} \quad (3.5)$$

De modo que la matriz de transformación que se aplicará a los puntos de la nube será:

$${}^q T_w = \begin{bmatrix} {}^w R_q^T & -{}^w R_q^T \cdot {}^w p_q \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (3.6)$$

Notar que, la notación utilizada para la matriz ${}^w R_q$ indicaba una rotación del s.d.c. local q al s.d.c. global W , mientras que ${}^q T_w$ indica una transformación

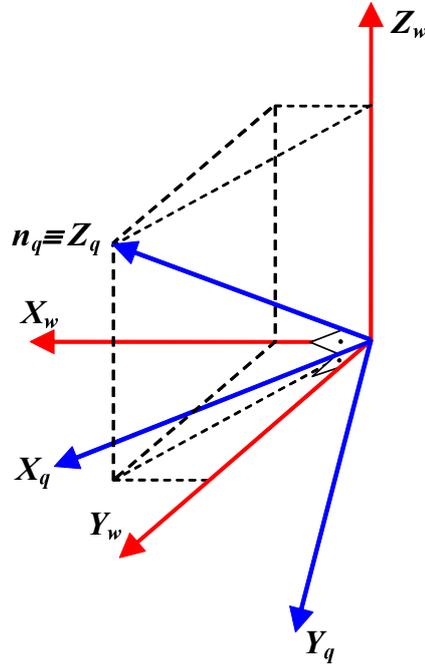


Figura 3.2: Elección de la orientación de los ejes del sistema de coordenadas local q con respecto al sistema de coordenadas global W .

(rotación y traslación) en sentido inverso.

Para un análisis más exhaustivo de las implicaciones de esta elección de los ejes del s.d.c. local, consultar el Apéndice A.

3.3.3. Imagen Cíclica de Contornos Radiales

Una vez transformados los puntos de la nube al s.d.c. local, se va a considerar que el entorno del punto de interés está dividido en n_s sectores de ángulo ρ_θ radianes que están a su vez divididos radialmente en celdas de longitud ρ_r milímetros (excepto la celda más cercana al centro, "celda 0", que será un sector de radio $0,5 \cdot \rho_r$). Los sectores se van a numerar en sentido horario comenzando por el sector que está centrado sobre el eje X_q ($\theta_1 = 0$). En la Figura 3.3 se muestra, en torno al eje Z_q , este sentido de numeración y cómo quedarían las celdas para el sector i -ésimo.

Atendiendo a esta división de la nube de puntos en sectores y celdas, se

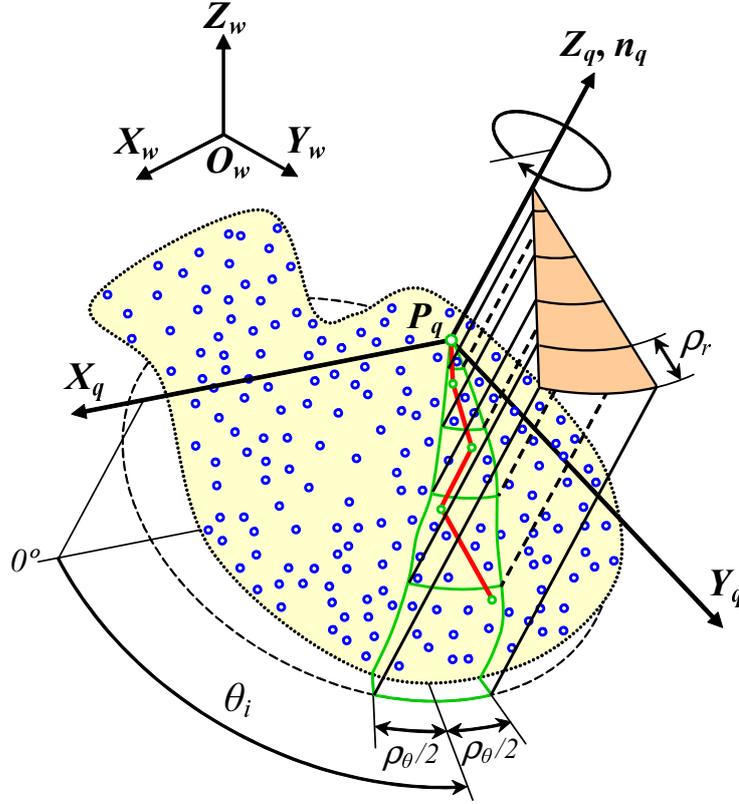


Figura 3.3: Construcción del descriptor CIRCON. En color verde se muestra la división en celdas del sector i ; en rojo el contorno formado por los puntos de esas celdas con mayor coordenada z .

aplicará una transformación basada en coordenadas cilíndricas que permite obtener, para cualquier punto ${}^q p_d$ con coordenadas (x_d, y_d, z_d) en el sistema de coordenadas con origen en p_q , el índice i que corresponde al número de sector al que pertenece, el índice j que indica el número de celda dentro de ese sector, y el valor de altura asociado a su coordenada z_d .

$$i = \left(\left[n_s - \frac{\tan^{-1} \left(\frac{y_d}{x_d} \right)}{\rho_\theta} \right] \bmod n_s \right) + 1 \quad (3.7)$$

$$j = \left\lfloor \frac{\sqrt{x_d^2 + y_d^2}}{\rho_r} \right\rfloor \quad (3.8)$$

$$c_{i,j} = \left\lfloor \frac{z_d}{\rho_z} \right\rfloor \quad (3.9)$$

Siendo ρ_θ la resolución angular, ρ_r la resolución radial y ρ_z la resolución en altura.

Notar que la Ecuación 3.7 utiliza el operador *mod* (resto de la división entre números enteros) para asignar valores al índice i que se encuentren entre 1 y n_s . Esta asignación, como se ha dicho, se lleva a cabo siguiendo un sentido horario de numeración de los sectores, de manera que el primer sector ($i = 1$) forma un ángulo $\theta_1 = 0$ (ó 2π) radianes con el eje X_q , el segundo sector un ángulo de $2\pi - \rho_\theta$ radianes, el tercero uno de $2\pi - 2 \cdot \rho_\theta$ y así va decreciendo sucesivamente hasta el último sector ($i = n_s$) al que le corresponde un ángulo de $2\pi - (n_s - 1) \cdot \rho_\theta$. Por consiguiente, un sector genérico i formará con el eje X_q un ángulo de $2\pi - (i - 1) \cdot \rho_\theta$ radianes.

Notar también que, según la Ecuación 3.8, existirá una "celda 0" ($j = 0$) que corresponde a un sector de radio $0,5 \cdot \rho_r$ (como ya se dijo anteriormente). Esto es debido a la operación de "redondeo al entero más cercano" (operador " $\lfloor \cdot \cdot \rfloor$ "), que hace que los puntos que tienen una distancia al punto de interés menor que $0,5 \cdot \rho_r$ tengan un índice j igual a cero. Para todos estos puntos pertenecientes a la "celda 0" se considerará un valor $c_{i,0}$ igual al del punto de interés, es decir, $c_{i,0} = 0$. Por consiguiente, las "celdas 0" de todos los sectores van a tener siempre asignado un valor nulo, de modo que nunca va a haber diferencias en estas posiciones cuando comparemos dos descriptores cualesquiera. Y teniendo en cuenta que, como ya se apuntó, la Ecuación 3.2 es ambigua para el punto de interés, estas celdas no se van a considerar cuando se evalúe la similitud entre dos descriptores.

Aplicando estas tres ecuaciones a todos los puntos de la nube y quedándonos con los valores $c_{i,j}$ que sean máximos para cada celda (que es equivalente a decir que su coordenada z es máxima) obtendremos un conjunto de ternas $(i, j, c_{i,j})$ que corresponden de manera unívoca a cada una de las celdas en las que ha sido dividido el entorno del punto de interés.

Por consiguiente, el descriptor estará formado por un conjunto de valores

	<u>radio</u> →					
ángulo ↓	$c_{1,1}$	$c_{1,2}$...	$c_{1,j}$...	c_{1,n_r}
	$c_{2,1}$	$c_{2,2}$...	$c_{2,j}$...	c_{2,n_r}
	⋮	⋮	⋮	⋮	⋮	⋮
	$c_{i,1}$	$c_{i,2}$...	$c_{i,j}$...	c_{i,n_r}
	⋮	⋮	⋮	⋮	⋮	⋮
	$c_{n_s,1}$	$c_{n_s,2}$...	$c_{n_s,j}$...	c_{n_s,n_r}

Figura 3.4: Matriz CIRCON. Las flechas muestran las direcciones de decrecimiento para el ángulo y de crecimiento para el radio.

codificados que representan los contornos descritos por los puntos con la máxima coordenada z de cada una de las celdas en las que está dividido cada sector. En la Figura 3.3 se muestra, para el sector i , el contorno que describen los puntos con mayor ' z ' de sus celdas.

Como a cada pareja de índices (i, j) le corresponde un único valor de $c_{i,j}$, se puede construir una matriz C que tenga como índice de fila el número de sector i y como columna el número de celda j dentro de ese sector. De este modo, la matriz tendrá n_s filas, tantas como el número de sectores en los que se haya dividido el entorno del punto de interés, y n_r columnas, que corresponde al radio máximo considerado (es decir, $n_r = \left\lceil \frac{R_{max}}{\rho_r} \right\rceil$). En la Figura 3.4 se muestra esta matriz y las direcciones de crecimiento del radio y de decrecimiento del ángulo (recordar que la numeración de los sectores se había tomado en sentido horario y que, por lo tanto, los ángulos se hacen cada vez más negativos).

Esta disposición matricial permite preservar las relaciones de adyacencia de las celdas, es decir, los elementos de la matriz correspondientes a dos celdas contiguas de un mismo sector estarán dentro de la misma fila y en columnas adyacentes. Del mismo modo, los de dos celdas contiguas de diferentes sectores estarán dentro de la misma columna y en filas adyacentes de la matriz. Por consiguiente, y dado que la secuencia debe ser cerrada, pues describe el entorno del punto de interés, las filas primera y última deben considerarse adyacentes, ya que sus elementos con mismo índice de columna corresponden a celdas

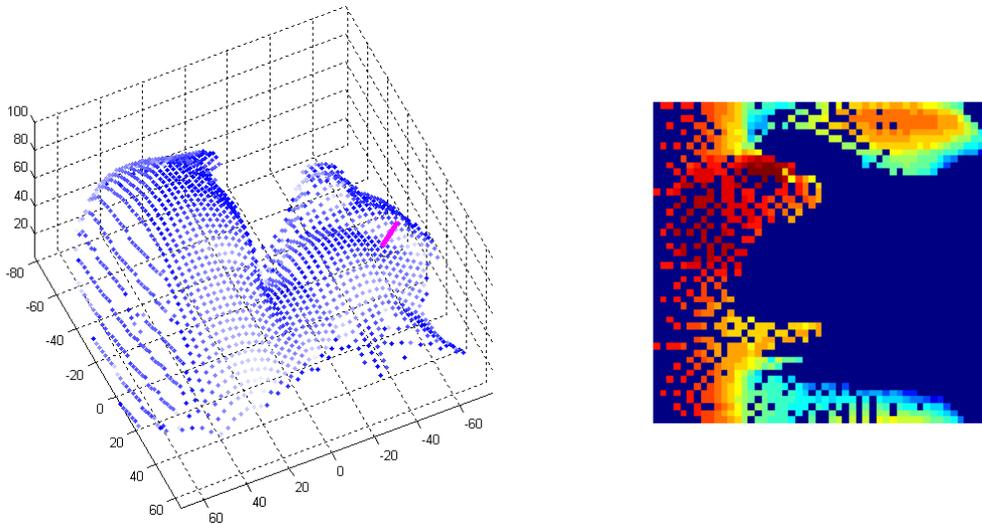


Figura 3.5: Ejemplo de imagen CIRCON. Nube de puntos con la normal de un punto en magenta (izquierda). Imagen CIRCON que corresponde a ese punto con $n_s = 48$ (derecha).

contiguas.

Esta matriz se puede también visualizar como una imagen (ver Figura 3.5) cuyos píxeles representan los valores $c_{i,j}$, de manera que cada uno de ellos tendrá asociado un color (o un nivel de gris) . De ahí que se pueda considerar al descriptor propuesto como una imagen cíclica del entorno del punto de interés. Y puesto que cada fila representa una codificación en sentido radial del contorno descrito por las alturas máximas de cada una de las celdas de un sector, vamos a denominar al descriptor resultante "Imagen Cíclica de Contornos Radiales" (en inglés "Cyclic Image of Radial CONtours": CIRCON).

3.4. Área representada por los píxeles del descriptor

Para construir un descriptor CIRCON es necesario definir, en primer lugar, la resolución radial ρ_r y la resolución angular ρ_θ con la que vamos a muestrear el entorno de un determinado punto de interés. El valor de estos parámetros

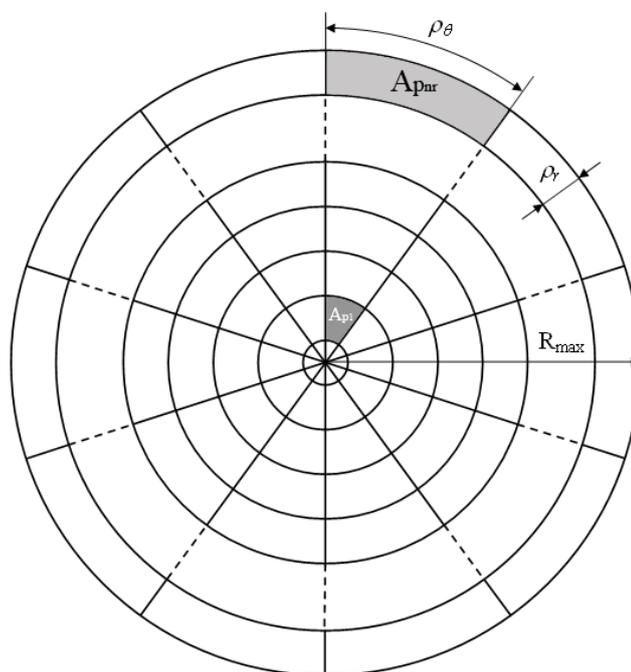


Figura 3.6: Área representada por los píxeles de la imagen CIRCON.

determinará el número de filas n_r y el número de columnas n_s que va a tener la matriz del descriptor.

Puesto que, debido al redondeo, se asignarán índices de columna de valor cero a puntos que estén a una distancia radial menor que $0,5 \cdot \rho_r$, el radio máximo que alcanzará el descriptor será $r_{\text{máx}} = \rho_r \cdot (n_r + 0,5)$. Por lo tanto el área barrida por éste puede calcularse como:

$$A_{\text{total}} = \pi \cdot r_{\text{máx}}^2 = \pi \cdot \rho_r^2 \cdot (n_r + 0,5)^2 \quad (3.10)$$

Mientras que, para el conjunto de celdas que contienen puntos cercanos al punto de interés, el área barrida tendrá como expresión:

$$A_{\text{central}} = \pi \cdot (0,5 \cdot \rho_r)^2 = \frac{\pi \cdot \rho_r^2}{4} \quad (3.11)$$

Por otro lado, el área del anillo comprendido entre radios a $0,5 \cdot \rho_r$ y a $1,5 \cdot \rho_r$ (es decir, su radio medio es ρ_r) puede obtenerse mediante la siguiente expresión:

$$A_{anillo1} = \pi \cdot (1,5 \cdot \rho_r)^2 - \pi \cdot (0,5 \cdot \rho_r)^2 = 2\pi \cdot \rho_r^2 \quad (3.12)$$

Y para una sola celda de ese anillo (un píxel en la primera columna de la matriz del descriptor):

$$A_{P_1} = \frac{A_{anillo1}}{n_s} = \frac{\pi \cdot (1,5 \cdot \rho_r)^2 - \pi \cdot (0,5 \cdot \rho_r)^2}{n_s} = \frac{2\pi \cdot \rho_r^2}{n_s} \quad (3.13)$$

Si se generaliza esta última expresión para una celda comprendida entre radios de valor $(j - 0,5) \cdot \rho_r$ y $(j + 0,5) \cdot \rho_r$ (correspondiente a un píxel en la columna j de la matriz), el área podrá calcularse como:

$$A_{P_j} = \frac{\pi}{n_s} \cdot ((\rho_r \cdot (j + 0,5))^2 - (\rho_r \cdot (j - 0,5))^2) \quad (3.14)$$

Simplificando,

$$A_{P_j} = \frac{2\pi \cdot \rho_r^2}{n_s} \cdot j \quad (3.15)$$

Si consideramos la expresión del área A_{P_1} como referencia:

$$A_{P_j} = j \cdot A_{P_1} \quad (3.16)$$

Es decir, el área que representa un píxel de la columna j de la matriz del descriptor es j veces mayor que el área de un píxel de la primera columna.

De igual modo, si se divide el área central en n_s partes iguales, cada una de ellas tendrá un área:

$$A_{P_0} = \frac{A_{central}}{n_s} = \frac{\pi \cdot \rho_r^2}{4 \cdot n_s} = \frac{A_{P_1}}{8} \quad (3.17)$$

Para calcular el área total bastará con sumar el área de todas las celdas:

$$A_{total} = n_s \cdot (A_{P_0} + A_{P_1} + A_{P_2} + \dots + A_{P_{n_r}}) = n_s \cdot A_{P_1} \cdot \left(\frac{1}{8} + 1 + 2 + \dots + n_r \right) \quad (3.18)$$

Por consiguiente, el área total barrida por el descriptor CIRCÓN será igual

$$W = \begin{array}{c} \begin{array}{|c|c|c|c|c|c|} \hline \triangleleft \frac{1}{8} & 1 & 2 & 3 & 4 & \dots & n_r \\ \hline \triangleleft \frac{1}{8} & 1 & 2 & 3 & 4 & \dots & n_r \\ \hline \triangleleft \frac{1}{8} & 1 & 2 & 3 & 4 & \dots & n_r \\ \hline \triangleleft \frac{1}{8} & 1 & 2 & 3 & 4 & \dots & n_r \\ \hline \end{array} \end{array}$$

Figura 3.7: Matriz de pesos del área de cada píxel de la imagen CIRCON.

al área de una celda del primer anillo multiplicada por la suma de los elementos de la matriz de pesos mostrada en la Figura (3.7).

La columna "cero" de la matriz se ha representado mediante triángulos para indicar que corresponde a los puntos que tienen una distancia radial al punto de interés menor que $0,5 \cdot \rho_r$ y por lo tanto se encuentran dentro de lo que hemos denominado área central. Como ya se comentó en la Sección (3.3.3), los píxeles del descriptor que correspondan a esta área tienen asignado un valor cero, ya que se encuentran muy cercanos al punto de interés, que es origen del sistema de coordenadas local. Por lo tanto, cuando midamos la similitud entre dos descriptores CIRCON, las diferencias entre los píxeles del área central van a ser siempre nulas, de modo que la principal contribución a esta medida vendrá dada por las diferencias entre los valores de los píxeles del resto de la matriz.

Como se puede observar en la Ecuación (3.19), la suma de los elementos de una fila de la matriz de pesos (exceptuando el píxel central) se puede calcular teniendo en cuenta que estos constituyen una serie aritmética. Por consiguiente:

$$A_{total} = A_{P_1} \cdot n_s \cdot \left(\frac{1}{8} + \sum_{j=1}^{n_r} j \right) = A_{P_1} \cdot n_s \cdot \left(\frac{1}{8} + \frac{n_r \cdot (n_r + 1)}{2} \right) \quad (3.19)$$

Reemplazando (3.13) y desarrollando la ecuación anterior, llegamos a la expresión que habíamos obtenido en (3.10) considerando el área del círculo barrido por el radio máximo R_{max} .

$$A_{total} = \frac{2\pi \cdot \rho_r^2}{n_s} \cdot n_s \cdot \left(\frac{1}{8} + \frac{n_r^2}{2} + \frac{n_r}{2} \right) = 2\pi \cdot \rho_r^2 \cdot \frac{(n_r + 0,5)^2}{2} = \pi \cdot r_{max}^2$$

Como ha quedado demostrado, la Ecuación (3.22) nos permite calcular el área total barrida por el descriptor sumando la contribución individual de cada elemento de la matriz de pesos, representando dicha contribución el número de veces que su celda correspondiente es mayor que el área de una celda del primer anillo.

Mientras que la Ecuación (3.10) sólo es válida para calcular el área correspondiente al conjunto de todas las celdas, la Ecuación (3.15) podría emplearse para obtener la suma total de áreas que correspondan solamente a celdas que contienen algún punto.

Esto resultará muy útil cuando tratemos con nubes de puntos parciales; en este caso habrá elementos de la matriz del descriptor que no tengan asignado un valor de altura (tendrán asignado un valor "not-a-number", NaN), de modo que será posible calcular el área total ocupada sumando solamente los elementos de la matriz de pesos que correspondan a píxeles del descriptor que no tengan un valor 'NaN'. Esto simplificará notablemente el cálculo ponderado de distancias entre dos descriptores CIRCON haciendo que su implementación computacional sea mucho más eficiente.

3.5. Aplicaciones

El descriptor que hemos presentado en este capítulo puede utilizarse para diferentes aplicaciones en el ámbito de la visión artificial 3D que detallamos a continuación.

3.5.1. Compresión de los Datos 3D

La imagen CIRCON se asemeja a un abanico recogido, de modo que el pliegue superior corresponde a la primera fila de la matriz, y el resto de pliegues sigue una numeración correlativa. Cuando el abanico se despliega los 360 grados muestra aproximadamente la vista en planta del objeto representado por la nube de puntos.

Se puede observar también que, debido a la elección de un sentido horario de numeración de los sectores, la imagen mostrada preserva el mismo orden visual que la vista en planta. Esto es fácil de ver si se considera que la imagen CIRCON tiene la 'columna cero' (que corresponde al punto central) a la izquierda de la matriz, mientras que las demás columnas corresponden al resto de puntos ordenados (de izquierda a derecha) en función de su distancia al punto central. Por lo tanto, si comenzamos por la fila 1 (correspondiente al sector 1), la siguiente fila corresponde al sector 2 que está, en sentido horario, debajo del primero, siguiendo ambos, de este modo, el mismo orden visual.

Si el sentido de numeración hubiese sido antihorario y quisiéramos seguir manteniendo que la primera columna de la imagen corresponda al primer anillo (el que es contiguo al punto central), no se preservaría el mismo orden visual pues, en este caso, el sector 2 estaría por encima del sector 1 mientras que, en la matriz CIRCON, la fila 2 estará, obviamente, debajo de la fila 1.

Se puede observar asimismo que, cuando se transforman los puntos de la nube para poder almacenarse en la matriz CIRCON, puede decirse que se lleva a cabo una compresión de los datos de partida, pues, aunque con pérdidas, la forma tridimensional de la nube se va a preservar. Es decir, la transformación que se aplicó a los puntos hasta llegar al descriptor CIRCON puede deshacerse y recuperar una aproximación de la forma 3D original (ver Figura 3.8). Esta propiedad, que será muy útil cuando se comparen dos descriptores, no la poseen ni las *spin images* ni otros descriptores intrínsecos similares.

Dado un elemento de la matriz CIRCON, $c_{i,j}$, con índice de fila i e índice de columna j , bastará con aplicar las siguientes expresiones para recuperar su punto 3D correspondiente, es decir, para realizar la transformación "inversa":

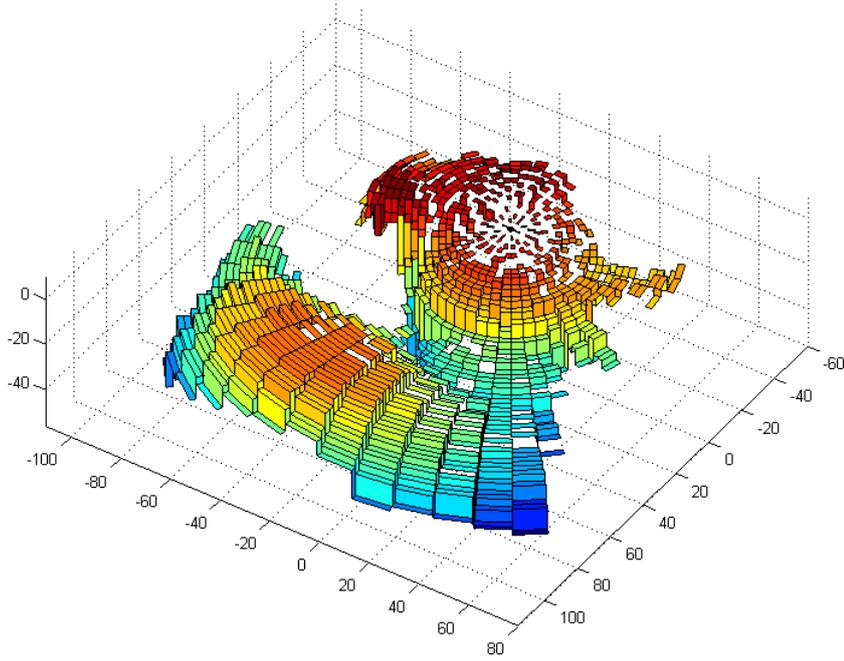


Figura 3.8: Recuperación de la forma 3D a partir de la imagen CIRCON mostrada en la Figura 3.5.

$$x_k = (j \cdot \rho_r) \cdot \cos(2\pi - (i - 1) \cdot \rho_\theta) \quad (3.20)$$

$$y_k = (j \cdot \rho_r) \cdot \sin(2\pi - (i - 1) \cdot \rho_\theta) \quad (3.21)$$

$$z_k = c_{i,j} \cdot \rho_z \quad (3.22)$$

3.5.2. Alineamiento 3D

Como ya se comentó en el Estado del Arte, el proceso de alineamiento 3D mediante descriptores intrínsecos busca correspondencias locales entre las dos nubes de puntos para calcular una transformación que las alinee en un sistema de coordenadas común. Esas correspondencias pueden obtenerse empleando descriptores CIRCON en ambas superficies tal y como se mostrará en el Capítulo

6.

3.5.3. Reconocimiento de Objetos

El descriptor CIRCON puede usarse también para tareas que impliquen reconocimiento de objetos y estimación de la posición y orientación de estos. Además, será útil en tareas de "Bin-Picking" ya que sus propias características hacen que pueda emplearse, en combinación con la medida de similitud presentada en el Capítulo 5, en situaciones en las que la nube de puntos representa solamente una pequeña parte del objeto.

Debido a que la distancia entre dos elementos adyacentes que pertenezcan a una misma columna de la matriz depende del radio, aquellos puntos que se encuentren lejos del punto de interés serán representados de manera menos precisa que aquellos que estén más cerca. Esto proporciona una ventaja a la hora de combatir problemas de oclusiones y mezcla de objetos ("clutter"), ya que la información será más densa en el entorno más próximo al punto de interés, mientras que los puntos más lejanos (los que tienen más probabilidad de pertenecer a otro objeto en caso de que estén mezclados) tendrán menos importancia en el descriptor.

Capítulo 4

Sistema de Visión Basado en el descriptor CIRCÓN

4.1. Descripción del Sistema de Visión

El sistema de visión 3D que se propone permite obtener una reconstrucción tridimensional de un objeto colocado sobre una plataforma giratoria y genera, además, un descriptor CIRCÓN que facilitará el alineamiento de la nube de puntos con la procedente de otro escaneo o con el modelo completo del objeto [103].

Este sistema de visión usa una variante del método de triangulación activa. Tal y como se muestra en las Figuras 4.1 y 4.2, su estructura se compone de dos cámaras situadas a ambos lados de la escena que van a permanecer fijas mientras se gira el objeto colocado sobre la plataforma. Para cada ángulo predeterminado, el proyector láser ilumina un contorno del objeto. A continuación, las dos imágenes adquiridas por las cámaras son procesadas por un computador para obtener una reconstrucción tridimensional de ese contorno justo antes de que la plataforma se mueva a la siguiente posición.

Debido a que tanto el plano láser como las cámaras están calibrados (es decir, se conoce la ecuación que define el plano así como los parámetros intrínsecos y extrínsecos de ambas cámaras), aquellos puntos que solamente son vistos por una de las dos cámaras pueden obtenerse mediante triangulación

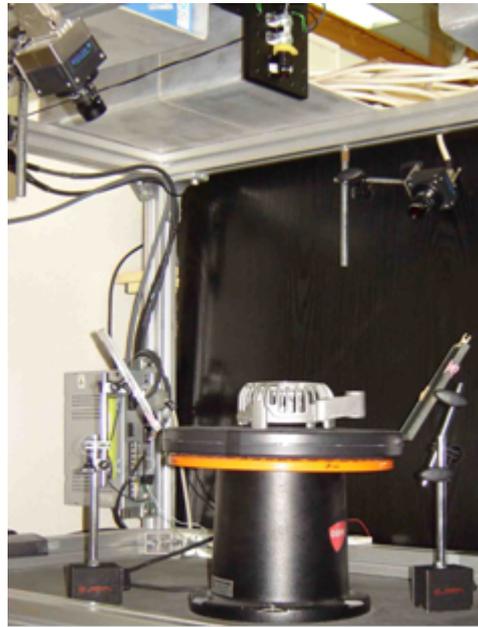


Figura 4.1: Sistema de visión propuesto.

activa.

Notar que si se hubiese utilizado una única cámara, habría que girar la plataforma una vuelta completa para poder adquirir todos los contornos del objeto; en cambio al utilizar dos cámaras se acelera la reconstrucción 3D, ya que solamente será necesario realizar un giro de 180 grados. Además, con el fin de evitar que algunas partes no puedan ser iluminadas debido a oclusiones producidas por el propio objeto, se han colocado dos espejos para re proyectar el láser sobre dichas regiones de modo que sí aparezcan iluminadas.

Como el sistema de coordenadas de referencia en el que se expresarán los contornos reconstruidos es solidario a la plataforma (ver Figura 4.2), la ecuación del plano láser va a modificarse para cada ángulo de giro; de manera que, si el eje de rotación está contenido en el plano láser, las ecuaciones de todos esos planos van a tener al vector unitario Z como vector director, o lo que es lo mismo, el eje de rotación va a ser la intersección de todos ellos. Tener en cuenta que en la práctica, debido a imprecisiones en el ajuste mecánico del sistema, va a producirse una cierta desviación con respecto a esta situación ideal. Por consiguiente, con el fin de minimizar los errores de reconstrucción de

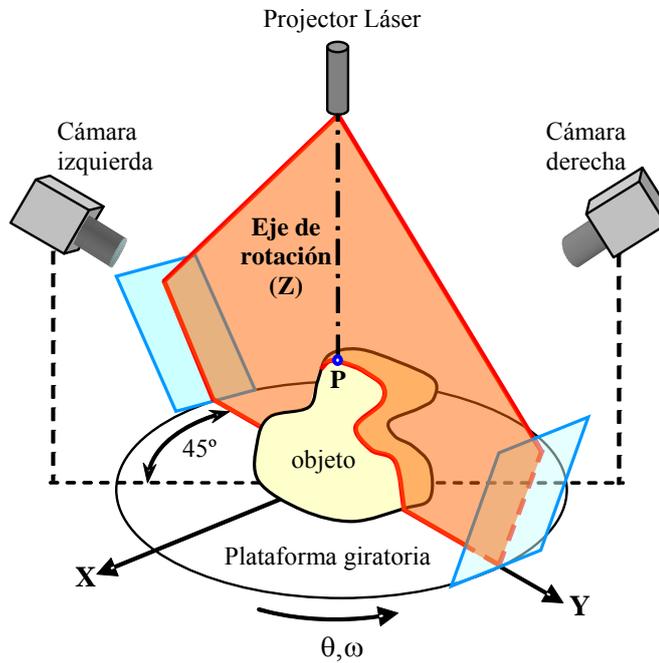


Figura 4.2: Estructura del sistema de visión 3D.

los contornos habrá que determinar con precisión la ecuación del plano láser para cada ángulo de giro.

Una vez que se hayan determinado las ecuaciones de los planos y las cámaras estén debidamente calibradas, el sistema ya estará en condiciones de obtener las coordenadas de los puntos iluminados mediante triangulación activa. Si se transforman las coordenadas de esos puntos mediante las Ecuaciones 3.7, 3.8 y 3.9, se pueden ir almacenando los resultados siguiendo la estructura matricial mostrada en la Figura 3.4, de modo que, al final del proceso de adquisición, se habrán completado todas las filas del descriptor CIRCON correspondiente. Notar que, debido a que la imagen CIRCON codifica "contornos radiales", se van a rellenar dos filas de la matriz, la fila i y la fila $i + N$, por cada contorno reconstruido i (siendo N el número de ángulos de giro prefijados e igual a $n_s/2$).

El descriptor obtenido puede utilizarse directamente para tareas que impliquen tanto el reconocimiento del objeto situado sobre la plataforma como la estimación de su posición y orientación. De hecho, algunas nubes de puntos ob-



Figura 4.3: Imagen CIRCON de la carcasa de un alternador ($\rho_\theta = 2,5^\circ$, $\rho_r = 0,5mm$).

tenidas mediante este sistema de visión se utilizaron para probar el algoritmo de visión 3D mostrado en el Capítulo 10.

Además el sistema de visión presentado podría integrarse en un brazo manipulador tan sólo montando las cámaras y el láser de manera que se aprovechase la rotación de la muñeca para hacer que estos girasen al unísono mientras los objetos permanecen inmóviles sobre una superficie plana o apilados en un contenedor. Esta configuración puede ser útil para tareas de captura automática de objetos, de modo que, si el eje de rotación apunta hacia uno de los puntos del objeto, el descriptor CIRCON resultante estará centrado en dicho punto y podrá compararse con un modelo para determinar su posición y orientación.

Un ejemplo del descriptor CIRCON obtenido inicialmente por el sistema de visión se muestra en la Figura 4.3, el cual corresponde al objeto situado sobre la plataforma en la Figura 4.1. En este caso, con las resoluciones que se han utilizado y que aparecen indicadas en la nota al pie, la imagen mostrada tiene $n_s = 144$ filas. Puede observarse además que los niveles de gris de la última y la primera filas guardan continuidad, pues como ya se había dicho los contornos representados siguen una secuencia cerrada, es decir, corresponden a un giro de 360° .

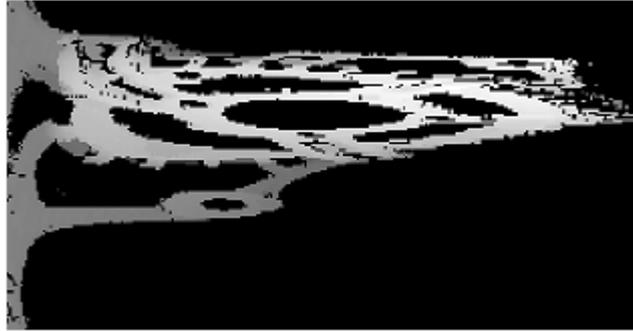


Figura 4.4: Normalización de la imagen CIRCON mostrada en la Figura 4.3.

4.2. Descriptor CIRCON Normalizado

El descriptor CIRCON obtenido al concluir el proceso de adquisición corresponde al punto del objeto que es intersecado por el eje de rotación, pero el sistema de coordenadas al que está referido no es el definido en la Sección 3.3.2. Para llevar a cabo esta normalización del descriptor habrá que llevar a cabo una transformación de los elementos de la matriz CIRCON o de los puntos 3D obtenidos por el sistema de visión.

En el primer caso habría que aplicar las Ecuaciones 3.20, 3.21 y 3.22, mientras que en el segundo caso esto no sería necesario. A continuación, los puntos 3D serán transformados mediante la matriz obtenida de la Ecuación (3.6). Si el ángulo entre el vector normal y el eje de rotación fuera pequeño el descriptor apenas sufriría cambios, tal y como se muestra en la Figura (4.4). Como puede observarse, debido a la rotación realizada, la imagen resultante es ligeramente más brillante en la parte superior que la imagen mostrada en la Figura (4.3). En cambio, si la variación de la normal fuese grande, el descriptor podría sufrir cambios muy significativos, aunque esto no quiere decir que no fuese posible emplearlo para las aplicaciones descritas en la Sección (3.5).

Se ha de tener en cuenta, por otro lado, que la normalización de la imagen CIRCON obtenida por el sistema de visión no sería necesaria si el eje de rotación coincidiera con la normal en el punto de interés. Para evitar este inconveniente, se podría utilizar un sistema de visión de similares características pero montado sobre un manipulador robótico, de manera que los giros de la

articulación de la "muñeca" se aprovecharían para girar tanto el plano láser como las cámaras, las cuales se montarían sobre una estructura solidaria a dicha articulación. En esta configuración los objetos permanecerían inmóviles sobre una superficie cualquiera, eliminando de este modo la necesidad de una plataforma giratoria. De esta forma el sistema de visión podría orientarse mediante el manipulador robótico para hacer que el eje de rotación y una estimación de la normal permanezcan alineados durante la adquisición.

Capítulo 5

Medida de Similitud

Como ya se apuntó en la Sección 3.1, una de las principales características que se demandaban al descriptor era su adecuación para utilizar una medida de similitud basada en distancias entre puntos de la nube. Otro requisito era que ésta favoreciese una correcta diferenciación de las correspondencias evaluadas con el fin de evitar la aparición de falsas correspondencias. Este fenómeno suele producirse en mayor grado cuando, en presencia de bajo solapamiento entre las nubes de puntos, se utilizan medidas de similitud tales como correlación cruzada, información mutua [111, 67, 99] y entropía conjunta [27, 21] (por nombrar algunas de las más ampliamente usadas en alineamiento 3D). Dado que estas medidas se aplican solamente al área solapada, es más probable que haya más puntos con entornos similares si esta área común es pequeña. Con el fin de evitar esas falsas correspondencias, sería más apropiado tener también en cuenta las áreas no solapadas entre ambos descriptores.

Atendiendo, pues, a los requisitos anteriores, hemos diseñado una medida de similitud [102] que se basa en las diferencias entre los valores de los píxeles de las imágenes CIRCON, considerando tanto las áreas solapadas como las no solapadas.

Además, para tratar de superar los inconvenientes de la selección de falsas correspondencias, se va a considerar que las imágenes CIRCON representan la extensión completa de las nubes de puntos, incrementando de este modo su poder descriptivo. Aunque en la práctica se podría reducir el "alcance" del

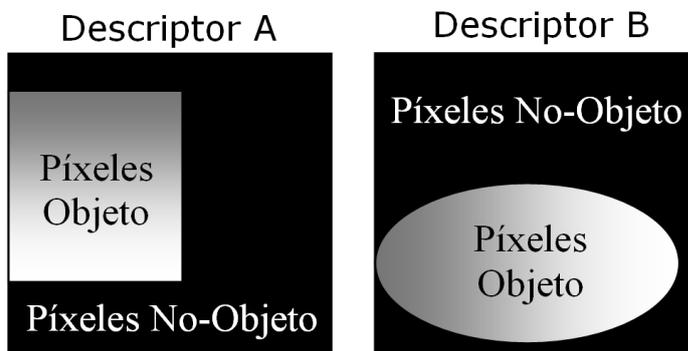


Figura 5.1: Clasificación de los píxeles de las imágenes CIRCON en píxeles de *Objeto* y píxeles de *No-Objeto*.

descriptor, se quiere demostrar con esto que la medida de similitud es capaz, aun en presencia de bajo solapamiento entre ambos descriptores, de encontrar una correspondencia que produzca un alineamiento aceptable de las nubes de puntos.

5.1. Clasificación de los píxeles

Cuando comparemos dos imágenes CIRCON para evaluar su similitud vamos a hacer una distinción entre los que llamaremos píxeles "*Objeto*" y píxeles "*No-Objeto*".

Se van a considerar píxeles "*Objeto*" aquellos que correspondan a celdas que contengan al menos un punto de la nube. Por consiguiente, al resto de píxeles de la imagen, cuyas celdas correspondientes no contienen ningún punto, se les va a denominar "*No-Objeto*". Estos píxeles serán considerados computacionalmente como un valor "*not-a-number*" (*NaN*) ya que no es posible calcular para ellos un valor de c_{ij} .

Esta clasificación de los píxeles nos va a permitir introducir la definición de *solapamiento*. Dadas dos matrices A y B correspondientes a dos imágenes CIRCON; si un píxel perteneciente a A , a_{ij} , y un píxel perteneciente a B , b_{ij} , son ambos píxeles "*Objeto*", se considerará que existe un *solapamiento* para ese par de píxeles con índices (i, j) .



Figura 5.2: Conjuntos Intersección, OR-Exclusivo y Unión correspondientes a las dos imágenes mostradas en la Figura 5.1.

Atendiendo, pues, a las definiciones anteriores, los pares de píxeles de dos descriptores CIRCON se pueden clasificar en los siguientes conjuntos:

- Conjunto Intersección (píxeles solapados):

$$I_{AB} = \{ (i, j) \mid (a_{ij} \neq NaN) \text{ and } (b_{ij} \neq NaN) \} \quad (5.1)$$

- Conjunto Exclusive-OR (píxeles no solapados):

$$X_{AB} = \{ (i, j) \mid (a_{ij} \neq NaN) \text{ xor } (b_{ij} \neq NaN) \} \quad (5.2)$$

- Conjunto Unión (píxeles solapados y no solapados):

$$U_{AB} = \{ (i, j) \mid (a_{ij} \neq NaN) \text{ or } (b_{ij} \neq NaN) \} \quad (5.3)$$

Debido a que los dos primeros conjuntos, I_{AB} y X_{AB} , son mutuamente excluyentes, se va a cumplir lo siguiente:

$$U_{AB} = I_{AB} \cup X_{AB} \quad (5.4)$$

En la Figura 5.2 se muestra un ejemplo de cómo serían los conjuntos para las imágenes CIRCON de la Figura 5.1.

5.2. Distancia media en el Área Solapada

Cuando dos nubes de puntos están expresadas, respectivamente, en los sistemas de coordenadas de dos puntos de interés, la distancia entre los puntos está directamente relacionada con las diferencias entre los valores de los píxeles de sus dos imágenes CIRCON correspondientes. De modo que podemos basar la medida de similitud en estas diferencias y calcular, por consiguiente, una distancia promedio entre los píxeles que corresponden al objeto representado. Sin embargo, para calcular esta distancia promedio, deberemos considerar por separado los píxeles del área solapada y los del área no solapada.

Atendiendo a lo visto en la Sección 3.4, la distancia de cada par de píxeles solapados va a ser ponderada por el área teórica que representan (Ecuación 3.16). De modo que vamos a definir la distancia promedio en el área solapada mediante la siguiente expresión:

$$D_{ov} = \frac{\sum_{(i,j) \in I_{AB}} (j \cdot A_{P1}) \cdot |a_{ij} - b_{ij}|}{\sum_{(i,j) \in I_{AB}} (j \cdot A_{P1})} \quad (5.5)$$

Que si es simplificada resulta:

$$D_{ov} = \frac{\sum_{(i,j) \in I_{AB}} j \cdot |a_{ij} - b_{ij}|}{\sum_{(i,j) \in I_{AB}} j} \quad (5.6)$$

Para obtener la contribución del área solapada a la distancia promedio del conjunto de píxeles, se deberá distribuir la suma de distancias de los píxeles solapados entre todos los píxeles del conjunto unión (U_{AB}):

$$\Delta_{ov} = \frac{\sum_{(i,j) \in I_{AB}} j \cdot D_{ov}}{\sum_{(i,j) \in U_{AB}} j} \quad (5.7)$$

5.3. Distancias en el Área No Solapada

Debido a que en los píxeles de las áreas no solapadas no es posible hallar una distancia ya que uno de los dos píxeles no pertenece al objeto, se asignará a dicho par de píxeles una distancia $D_{ov} + \lambda$ (con $\lambda > 0$). De este modo la contribución del área no solapada, Δ_{nov} , va a ser igual a la suma de las distancias asignadas a sus píxeles promediada entre el número de píxeles del área total (solapada más no solapada):

$$\Delta_{nov} = \frac{\sum_{(i,j) \in X_{AB}} j \cdot (D_{ov} + \lambda)}{\sum_{(i,j) \in U_{AB}} j} \quad (5.8)$$

Y dado que:

$$U_{AB} = I_{AB} \cup X_{AB} \implies \sum_{(i,j) \in X_{AB}} j = \sum_{(i,j) \in U_{AB}} j - \sum_{(i,j) \in I_{AB}} j \quad (5.9)$$

Se puede también expresar la Ecuación 5.8 de la siguiente forma:

$$\Delta_{nov} = \frac{\left(\sum_{(i,j) \in U_{AB}} j - \sum_{(i,j) \in I_{AB}} j \right) \cdot (D_{ov} + \lambda)}{\sum_{(i,j) \in U_{AB}} j} = \left(1 - \frac{\sum_{(i,j) \in I_{AB}} j}{\sum_{(i,j) \in U_{AB}} j} \right) \cdot (D_{ov} + \lambda) \quad (5.10)$$

Notar que el valor que se asigne al parámetro λ va a permitir regular esta contribución a la distancia media.

5.4. Distancia media entre imágenes

Por consiguiente, la distancia promedio será la suma de la contribución del área solapada y la contribución asignada al área no solapada:

$$D_m = \Delta_{ov} + \Delta_{nov} \quad (5.11)$$

Si el porcentaje de solapamiento se define como:

$$\sigma_{ov} = \frac{\sum_{(i,j) \in I_{AB}} j}{\sum_{(i,j) \in U_{AB}} j} \quad (5.12)$$

las Ecuaciones 5.7 y 5.10 pueden reescribirse como,

$$\Delta_{ov} = D_{ov} \cdot \sigma_{ov} \quad (5.13)$$

y

$$\Delta_{nov} = (1 - \sigma_{ov}) \cdot (D_{ov} + \lambda) \quad (5.14)$$

Sustituyendo estas expresiones en la Ecuación 5.11 y simplificando, resulta:

$$D_m = D_{ov} + \lambda \cdot (1 - \sigma_{ov}) \quad (5.15)$$

5.5. Medida basada en la distancia entre descriptores

El inconveniente de utilizar solamente la distancia media es que va a proporcionar un valor cercano a $D_{ov} + \lambda$ cuando el porcentaje de solapamiento sea muy bajo. Si la distancia en esa pequeña parte solapada fuese casi nula ($D_{ov} \simeq 0$), D_m sería aproximadamente igual a λ ; esto supondría un valor de λ muy grande para poder penalizar un solapamiento muy bajo, lo cual perjudicaría excesivamente a las medidas con un solapamiento mayor. Por el contrario, utilizar un valor bajo de λ podría dar lugar a falsas correspondencias, ya que los pares de descriptores con bajo solapamiento se verían favorecidos.

Para poder evitar estos inconvenientes vamos a distribuir la suma total de distancias solamente entre los píxeles del solapamiento:

$$D_S = \frac{\sum_{(i,j) \in U_{AB}} j \cdot D_m}{\sum_{(i,j) \in I_{AB}} j} = \frac{D_{ov} + \lambda \cdot (1 - \sigma_{ov})}{\sigma_{ov}} = \frac{D_{ov}}{\sigma_{ov}} + \lambda \cdot \left(\frac{1}{\sigma_{ov}} - 1 \right) \quad (5.16)$$

De esta manera, un solapamiento muy bajo produciría un valor alto de D_S independientemente del valor de λ , lo cual va a facilitar la penalización de correspondencias con un solapamiento muy bajo. Por otro lado, el parámetro λ va a permitir regular la contribución de los píxeles no solapados cuando $D_{ov} = 0$.

Dicho de otro modo, si hubiese dos pares de descriptores con una distancia pequeña en el solapamiento, la Ecuación 5.16 debería asignar una distancia estimada D_S más pequeña al par que tuviese mayor solapamiento.

5.6. Normalización de la Medida de Similitud

Aunque podría usarse directamente la medida de distancia estimada D_S , suele ser más conveniente disponer de una medida normalizada que proporcione valores próximos a la unidad para valores de D_S casi nulos y valores cercanos a cero cuando D_S sea muy grande.

Para normalizar la medida de similitud a valores comprendidos entre 0 y 1, utilizaremos la siguiente expresión:

$$M_S = \frac{1}{\rho \cdot D_S + 1} \quad (5.17)$$

siendo ρ un parámetro que va modificar la relación entre D_S y M_S .

Sustituyendo la expresión de D_S (Ecuación 5.16),

$$M_S = \frac{1}{\rho \cdot \left(\frac{D_{ov} + \lambda \cdot (1 - \sigma_{ov})}{\sigma_{ov}} \right) + 1} = \frac{\sigma_{ov}}{(\rho \cdot D_{ov} + \rho \cdot \lambda) + \sigma_{ov} (1 - \rho \cdot \lambda)}$$

Si definimos el parámetro λ' como $\lambda' = \rho \cdot \lambda$

$$M_S = \frac{\sigma_{ov}}{(\rho \cdot D_{ov} + \lambda') + \sigma_{ov} \cdot (1 - \lambda')} \quad (5.18)$$

En el Apéndice B se proporciona un sencillo criterio para seleccionar los valores de los parámetros ρ y λ .

5.7. Cálculo de la Medida de Similitud

Como ya se vio en la Sección 3.3.3, desplazar todas las filas de una imagen CIRCON una posición hacia abajo (de manera que la última fila pasa a ser la primera) implica girar la nube de puntos un ángulo de ρ_θ radianes en sentido horario sobre la normal en el punto de interés. Por consiguiente, cuando se evalúa la similitud de dos imágenes CIRCON, A y B , va a haber n_s combinaciones posibles, tantas como el máximo número de "rotaciones" que pueden aplicarse a una de las dos imágenes (por ejemplo A) mientras la otra permanece fija. De ahí que, para determinar la medida de similitud, habrá que realizar, en principio, n_s evaluaciones de la Ecuación 5.18 y, de todos los valores resultantes, hallar su máximo, $M_{S_{max}}$, al cual le corresponden k_i "rotaciones" de las filas de la imagen A .

En el Apéndice C se explica de forma detallada la implementación de la medida de similitud para que el cálculo del valor máximo $M_{S_{max}}$ sea más eficiente y pueda incluso reducirse el número de evaluaciones de la Ecuación 5.18.

5.8. Matriz de rotación asociada a la medida de similitud

Como se ha visto en la sección anterior, cuando se obtiene el valor $M_{S_{max}}$ de similitud entre dos descriptores asociados a una correspondencia $a \leftrightarrow b$, se determina también el índice de rotación k_a sobre la normal. Por consiguiente, a partir de este índice se puede obtener la matriz de rotación correspondiente, que va a representar un giro de $-k_a \cdot \rho_\theta$ radianes sobre el eje Z_a del sistema de

coordenadas local:

$$R_{Z_a}(-k_a \cdot \rho_\theta) = \begin{bmatrix} \cos(k_a \cdot \rho_\theta) & \sin(k_a \cdot \rho_\theta) & 0 \\ -\sin(k_a \cdot \rho_\theta) & \cos(k_a \cdot \rho_\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.19)$$

Siendo su matriz de transformación asociada:

$$T_{Z_a}(-k_a \cdot \rho_\theta) = \begin{bmatrix} R_{Z_a}(-k_a \cdot \rho_\theta) & 0_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (5.20)$$

Capítulo 6

Búsqueda de Correspondencias

6.1. Introducción

Como ya se apuntó en el Capítulo 2, la búsqueda de correspondencias entre las dos nubes de puntos es la parte del algoritmo de alineamiento que más dificultades entraña. Utilizando el descriptor presentado en el Capítulo 3 y la medida similitud del Capítulo 5, se ha diseñado un algoritmo que busca la correspondencia aproximada de un punto de interés de una de las dos nubes de puntos que se desea alinear. Para ello se evalúa la similitud entre el descriptor asociado a ese punto y los descriptores que pueden extraerse alrededor de un punto de interés de la otra nube.

Antes de aplicar el algoritmo de búsqueda de correspondencias será conveniente reducir la cantidad de puntos (si esta fuese excesiva) y seleccionar un conjunto de puntos de interés de la nube resultante. Estas dos operaciones permitirán reducir considerablemente el tiempo de computación sin apenas restar eficacia al algoritmo.

6.2. Reducción de las nubes de puntos

Uno de los inconvenientes que se plantean a la hora de buscar correspondencias entre dos nubes de puntos es que la cantidad de datos que se manejan es muy grande, lo cual puede hacer que cualquier operación o transformación

que se realice sobre ellos lleve asociado un tiempo de computación elevado. Para evitar estos problemas vamos a proceder a reducir las nubes de puntos originales de modo que se preserven las características fundamentales de su forma.

El algoritmo utilizado para reducir la cantidad de información de la nube de puntos es bastante sencillo, aunque, como se demostrará más adelante, suficiente para los requisitos del algoritmo de alineamiento que se ha diseñado, el cual, cabe recordar, tiene como objetivo encontrar una transformación euclídea que alinee de forma aproximada las dos nubes de puntos.

El algoritmo de reducción parte de la imagen de rango en la que está expresada la nube de puntos. A continuación se calculan los valores mínimos y máximos de las coordenadas X e Y que correspondan a puntos de la nube. Estos valores se utilizarán para calcular la resolución en las direcciones X e Y para la nueva imagen de rango:

$$\rho_X = \frac{\max(X) - \min(X)}{N_c - 1} \quad (6.1)$$

$$\rho_Y = \frac{\max(Y) - \min(Y)}{N_f - 1} \quad (6.2)$$

Siendo N_f y N_c el número de filas y columnas que se desean para dicha imagen.

A continuación se construye la rejilla con N_c valores para las coordenadas X (desde $\min(X)$ hasta $\max(X)$) y N_f valores para las coordenadas Y (desde $\min(Y)$ hasta $\max(Y)$).

Utilizando las coordenadas de los puntos de la imagen de rango original se calcula, mediante interpolación con el vecino más cercano, la coordenada Z correspondiente a los puntos de la rejilla. Dicha interpolación sólo se aplicará a aquellas posiciones de la rejilla a las que les deberían corresponder puntos de la imagen de rango original, de modo que el objeto en la imagen de rango reducida siempre estará contenido en el de la imagen original. En la Figura 6.1 se muestra un ejemplo del resultado obtenido al aplicar este método de interpolación de la imagen de rango. Se puede observar en la Figura 6.2 cómo se reduce el número de puntos de la nube (de 49570 a 984) mientras se preserva

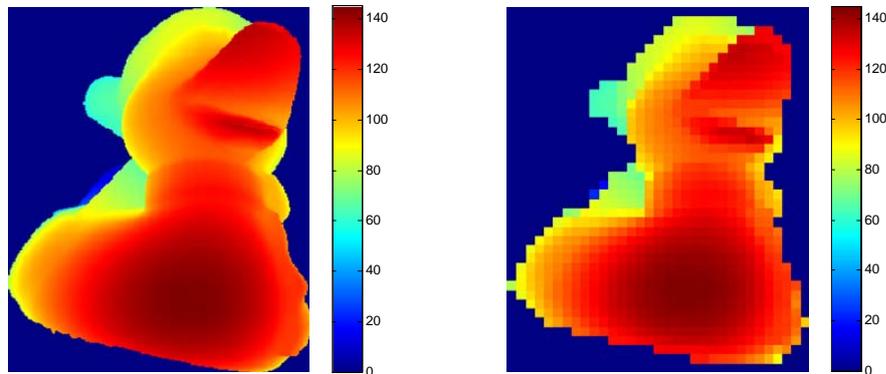


Figura 6.1: Reducción de la nube de puntos mediante interpolación de la imagen de rango. Dimensiones de la imagen de rango original (izquierda): 300×247 . Dimensiones de la imagen de rango reducida (derecha): 40×33 .

la forma básica original.

También es posible, utilizando un ajuste bicúbico de los datos X , Y , Z de la imagen de rango reducida, estimar los vectores normales en cada punto de la nube. Para ello se calculan vectores diagonales para cada celda de la imagen de rango y, a continuación, se obtiene la normal mediante su producto vectorial. Aquellos puntos para los que no haya sido posible obtener una estimación de su vector normal serán eliminados de la imagen de rango y, por consiguiente, no se considerarán para la búsqueda de correspondencias.

La propia sencillez de este algoritmo de reducción de las nubes de puntos nos va a indicar, a la vista de los resultados que se mostrarán más adelante, que el funcionamiento del algoritmo de alineamiento 3D no depende fuertemente de una medida precisa de la superficie de los objetos.

Sin embargo, y dado que ésta es una parte perfectamente separable del resto, se podría sustituir este algoritmo de reducción de datos por otro más complejo que ayudara a obtener resultados más precisos.

6.3. Selección de Puntos de Interés

Como se expuso en el Capítulo 2, una etapa clave en el proceso de búsqueda de correspondencias es la selección de los puntos de interés, es decir, aquellos

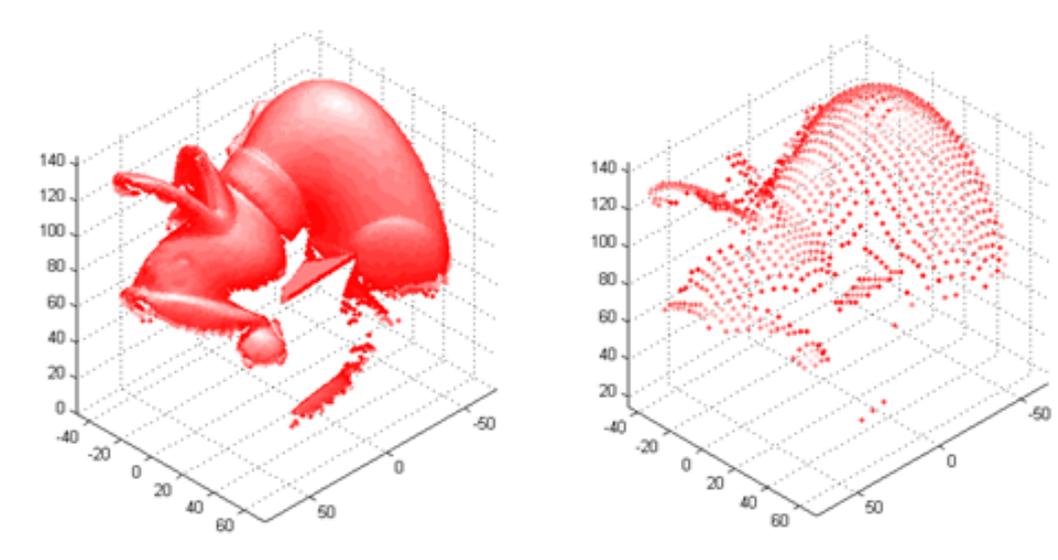


Figura 6.2: Reducción de la nube de puntos. Número original de puntos: 49570. Número reducido de puntos: 984.

puntos que tienen una especial relevancia dentro de cada nube de puntos y que van a permitir simplificar dicha búsqueda.

Aunque existen diferentes métodos para seleccionar puntos de interés de la nube, en nuestro caso se ha optado por uno que tenga en cuenta las características particulares del descriptor CIRCON. Mediante una serie de operaciones sobre la imagen de rango reducida (obtenida mediante el algoritmo de la Sección 6.2) se van a seleccionar puntos en los que su descriptor CIRCON sea potencialmente más estable y con mayor poder discriminante.

Dado que el descriptor CIRCON es una representación del entorno de un punto de interés y ésta depende a su vez de la normal en dicho punto, deberíamos seleccionar puntos de la nube que tuviesen un vector normal lo más estable posible. Por otro lado, dado que la descripción de ese entorno es más precisa en las primeras columnas de la matriz CIRCON, se deberían seleccionar puntos de interés que fuesen cercanos a zonas de la nube de puntos donde haya mayor variación de la normal, pues dichas zonas proporcionarán al descriptor mayor poder discriminante. En resumen, se deberán seleccionar puntos de interés que pertenezcan a regiones de la nube que tengan una normal estable y

que al mismo tiempo sean cercanos a otras con gran variación de la normal.

Partiendo de la imagen de rango reducida se va a llevar a cabo el siguiente procedimiento para seleccionar los puntos de interés:

- Se aplica un detector de bordes a la imagen de rango para obtener los puntos de la nube donde haya una mayor variación de la profundidad (Figura 6.3.b). A continuación se conectan píxeles cercanos y se rellenan huecos que tengan un solo píxel para evitar los problemas generados por una mala detección (Figura 6.3.c). Llamamos B a esa imagen binaria.
- Creamos otra imagen binaria con los puntos de la imagen de rango que pertenecen al objeto y la erosionamos con una máscara 5×5 . Aplicando una operación lógica 'AND' a la imagen resultante y al complemento de la imagen de bordes, eliminamos el contorno exterior y solamente nos quedamos con los puntos interiores que no pertenecen a los bordes (ver ejemplo en la Figura 6.3.d). Llamamos H a la matriz resultante.
- A continuación se va a aplicar la laplaciana discreta a la matriz Z , pues proporciona información de la curvatura media en cada punto. Como resultado se obtiene una matriz L que contiene las estimaciones de la laplaciana en los puntos de la nube reducida (ver Figura 6.3.e).
- Se calcula la mediana, med_L , de todos los elementos de la matriz L .
- Ponemos a cero todos los píxeles de la máscara binaria H cuyo píxel equivalente en la matriz L tenga un valor mayor que med_L . Por lo tanto, sólo se realizará la búsqueda de puntos de interés en aquellos píxeles de H que permanezcan a uno (Figura 6.3.f).
- A continuación se llevará a cabo la siguiente secuencia de operaciones:
 1. Se crea una lista pdi donde se almacenarán los puntos de interés.
 2. Se crea un elemento estructurante en forma de disco de radio r píxeles, tal que $\pi \cdot r^2$ sea aproximadamente el 4% del número total de píxeles (de esta forma el número de puntos de interés no será excesivo). En la Figura 6.3.g se ha tomado $r=3$.

3. De todos los píxeles que valgan '1' en H , se obtiene el píxel p_{ij} que tenga el mínimo valor en la matriz L (pues será el que tenga la normal potencialmente más estable).
4. Se extrae el entorno de p_{ij} mediante el elemento creado en el paso 2.
5. Si entre los píxeles del entorno de p_{ij} hay al menos un píxel de borde,
 - a) Se guarda p_{ij} en la lista pdi .
 - b) Se elimina el entorno del punto p_{ij} de la matriz H (de esta manera los píxeles que se seleccionen tendrán entre ellos al menos una distancia R).
6. Si no, se elimina solamente el punto p_{ij} de la matriz H (por lo tanto, ya no se considerará en el paso 3).
7. Si aún existen puntos en la matriz H , se vuelve al paso 3.

Este sencillo algoritmo permite obtener un conjunto de puntos de interés distribuidos por la nube de puntos en zonas donde el vector normal cumple ciertas condiciones de estabilidad y de cercanía a puntos característicos de la imagen de rango (ver Figura 6.3.g). Estos puntos de interés se van a utilizar como puntos de partida (en la nube 1) y puntos objetivo (en la nube 2) en el algoritmo de búsquedas de correspondencias .

6.4. Algoritmo de Búsqueda de Correspondencias

Esta algoritmo de búsqueda constituye el núcleo del algoritmo de alineamiento aproximado (*coarse alignment*). Se basa en una búsqueda iterativa del mayor valor de la medida de similitud utilizando para ello las celdas en las que es dividido el entorno de un punto de interés de partida. El criterio de parada seleccionado asegura que ésta sea una búsqueda convergente, ya que siempre va a converger dentro de un determinado entorno de la nube de puntos.

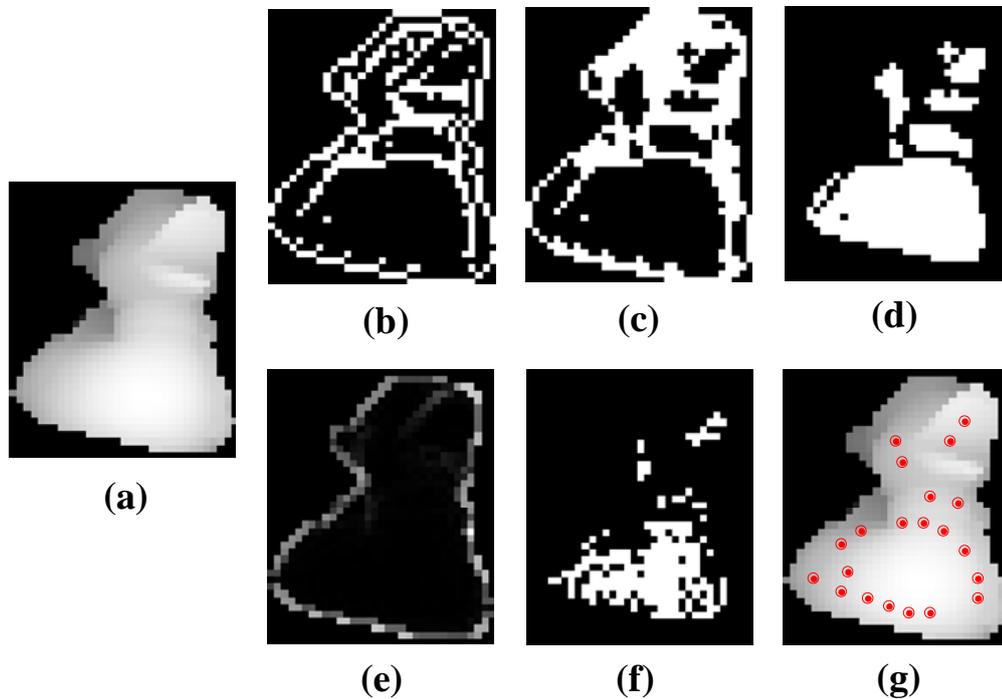


Figura 6.3: Selección de puntos de interés a partir de la imagen de rango. (a) Imagen de rango. (b) Imagen de bordes. (c) Imagen de bordes después de conectar píxeles cercanos y rellenar huecos. (d) Máscara obtenida tras complementar solamente los píxeles de la imagen (c) que corresponden al objeto. (e) Imagen obtenida tras aplicar el operador laplaciano a la imagen de rango. (f) Píxeles de la imagen (d) cuyo valor en la imagen (e) es menor que la mediana. (g) Puntos de interés seleccionados.

6.4.1. Datos de entrada y parámetros

El algoritmo de búsqueda de correspondencias va a necesitar los siguientes datos de entrada y parámetros:

$\{p_1\}$: puntos de la nube 1.

$\{n_1\}$: vectores normales en los puntos de la nube 1.

I_1 : imagen CIRCON en el punto de interés donde comenzará el algoritmo.

C_1 : distribución de los puntos de la nube 1 en celdas correspondientes a los píxeles de I_1 .

I_2 : imagen CIRCON objetivo (perteneciente a la nube 2).

α_n : ángulo de soporte. La imagen CIRCON se construirá solamente con los puntos cuya normal forme con la normal del punto central un ángulo menor que α_n .

δ : distancia mínima admitida entre los puntos que componen la lista de las mejores correspondencias de cada iteración. Este valor va a fijar el criterio de parada del algoritmo y debe ser menor que la resolución elegida para construir la imagen CIRCON.

τ_{nv} : máximo valor admitido para el ratio de puntos no válidos asociados a una celda. En la siguiente sección se explica cómo obtener dicho ratio, el cual permite descartar aquellas celdas donde va a ser muy poco probable que se encuentre la correspondencia buscada y acelerar, de este modo, el algoritmo.

6.4.2. Descripción del algoritmo

El algoritmo va a evaluar correspondencias entre diferentes puntos de la nube 1 y un punto de interés previamente seleccionado de la nube 2. El grado de validez de cada correspondencia se va a determinar en base a la similitud entre sus imágenes *circon* correspondientes: I_{1x} (imagen del punto de la nube 1 que se desea evaluar), e I_2 (imagen objetivo).

Si el valor de similitud obtenido M_{Sx} es inferior en un cierto porcentaje al valor máximo de similitud obtenido en la iteración, es decir, $M_{Sx} < \eta \cdot M_{Sit}$ ($\eta = 0,5$ en nuestros experimentos), entonces ese punto será considerado "no válido" (pasando de este modo a figurar en la *lista de índices no válidos: ind_{nv}*) y no se tendrá en cuenta para posteriores evaluaciones. Al comienzo del algoritmo esa lista (ind_{nv}) se encuentra vacía y se va actualizando en cada iteración. Apoyándonos en esta información podemos determinar en qué celdas de C_1 es más probable encontrar la correspondencia buscada. Por lo tanto, antes de seleccionar un punto de una celda para evaluar la similitud de su descriptor correspondiente, comprobaremos si va a ser probable que la correspondencia buscada se encuentre en dicha celda. Para ello utilizaremos el *ratio de puntos no válidos en la celda*, r_{nv} , que es el número de puntos de la celda que están en la lista ind_{nv} con respecto al número total de puntos de la celda. Cuando r_{nv} supere el máximo valor admitido τ_{nv} , esa celda no será evaluada. De esta

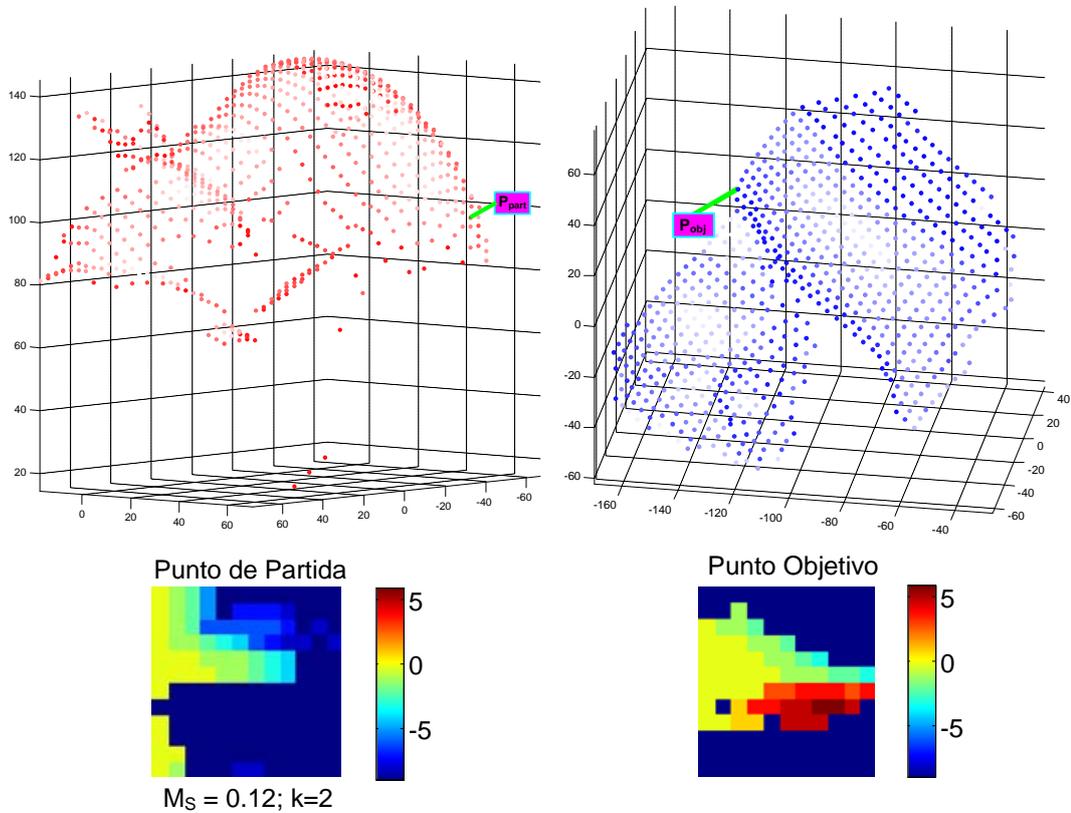


Figura 6.4: Para encontrar la correspondencia de un *punto objetivo* de la nube 2 (arriba a la derecha) el algoritmo de búsqueda comenzará en un punto de interés de la nube 1 (arriba a la izquierda). Las imágenes CIRCON que corresponden a ambos puntos se muestran abajo junto con su valor de similitud e índice k .

forma se consigue acelerar sensiblemente el algoritmo.

En la Figura 6.4 se muestra un *punto de partida* en la nube 1 (arriba a la izquierda), que es donde comenzará la búsqueda, y un *punto objetivo* en la nube 2 (arriba a la derecha) del que se quiere conocer su punto correspondiente en la nube 1. La máxima similitud entre ellos (con un valor de 0.12) se produce, en este caso, cuando a la imagen *circon* del *punto de partida* se le rotan sus filas dos posiciones en sentido horario. Como en este ejemplo el número de filas, n_s , de los descriptores es 12, un índice $k = 2$ implica un giro sobre la normal

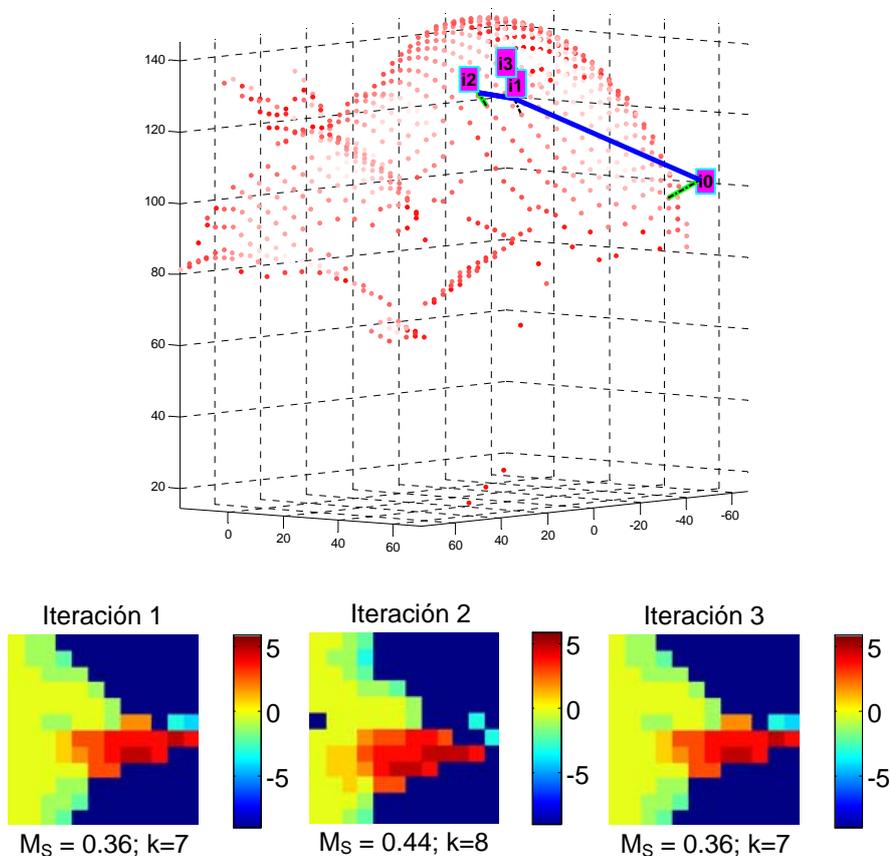


Figura 6.5: Ejemplo del camino descrito por las correspondencias con la máxima similitud de cada iteración (arriba) hasta que se produce la convergencia del algoritmo. Las imágenes CIRCON de las tres iteraciones se muestran abajo junto con sus valores de similitud e índices k .

en el *punto de partida* de 60° en sentido horario. Para que la comparación sea más fácil a los ojos del lector, a las imágenes CIRCON mostradas en las Figuras 6.4 y 6.5 se les han rotado sus filas tantas posiciones como indica el índice k .

En la Figura 6.5 se muestran los resultados de las tres iteraciones que siguió el algoritmo de búsqueda hasta encontrar la condición de parada. En este caso, la parada se produce porque el resultado obtenido en la iteración 3 coincide con el resultado de la iteración 1, lo cual implica que, al ser idénticas las condiciones de ambas iteraciones, se volvería a repetir de nuevo la misma

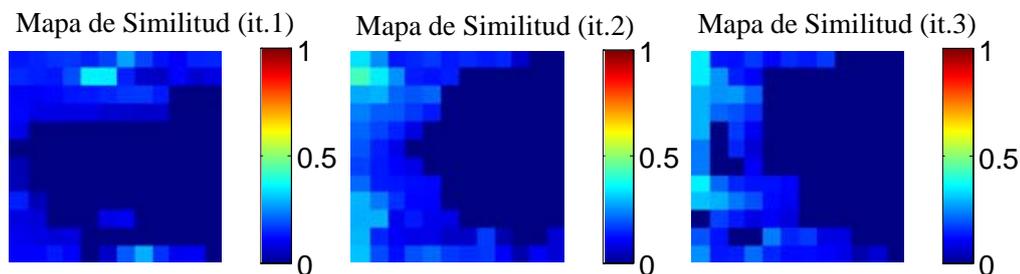


Figura 6.6: Mapas de similitud para las tres iteraciones de la Figura 6.5.

secuencia de resultados. Es decir, la iteración 4 sería igual a la iteración 2.

El valor de similitud que obtuvo el algoritmo en este caso fue $M_{Sc} = 0,44$ (con un índice $k_{1c} = 8$), que es el máximo obtenido en las tres iteraciones. Por lo tanto, el punto obtenido en la segunda iteración será la correspondencia aproximada, p_{1c} , del *punto objetivo* de la nube 2 (ver Figura 6.4).

En la Figura 6.6 se muestran los mapas de similitud de las tres iteraciones de la Figura 6.5. El mapa de similitud muestra, mediante una escala de colores, la medida de similitud que se obtuvo en cada una de las celdas evaluadas en una iteración. La distribución de celdas mostrada en el mapa corresponde al punto que obtuvo la mayor similitud en la iteración anterior (excepto para la primera iteración). En este ejemplo, la iteración 1 tiene la distribución del *punto de partida* del algoritmo, la iteración 2 tiene la correspondiente al punto obtenido en la iteración 1, y la tercera iteración tiene la del punto resultante de la iteración 2.

Como el mapa de similitud de la primera iteración tiene la distribución de celdas correspondiente al *punto de partida* y su valor de similitud con el *punto objetivo* es de solamente 0.12, hay varias celdas a las que le corresponde un valor de similitud mayor (en la Figura 6.6 aparecen con un color azul más brillante). Por consiguiente, el punto con el que comienza la siguiente iteración estará en la celda que da lugar al descriptor con mayor similitud. Como esa celda está en la segunda fila y quinta columna (radio medio igual a $5 \cdot \rho_r$), ese nuevo punto estará a una distancia radial comprendida entre $4,5 \cdot \rho_r$ y $5,5 \cdot \rho_r$ y tendrá una coordenada angular entre 15° y 45° (es importante destacar que los mapas de similitud mostrados son los originales y que, por lo tanto, no se

les ha aplicado ninguna rotación de sus filas).

Como puede observarse, a medida que aumentan las iteraciones la celda con el mayor valor de similitud (azul más brillante) se desplaza hacia las primeras columnas, lo cual significa que el punto elegido para obtener la distribución de celdas en la siguiente iteración se va acercando al de la iteración actual. Esto quiere decir que el algoritmo va convergiendo hacia un entorno pequeño de un punto de la nube, tal y como muestra el camino descrito por las iteraciones del ejemplo de la Figura 6.5.

Como nuestro descriptor no considera la columna "cero", el punto de comienzo de una iteración no se va a incluir en el entorno de búsqueda, lo cual va a obligar al algoritmo a elegir otro punto para la siguiente iteración. Ya que solamente se evalúa un descriptor por celda y ésta engloba a más puntos, esto va a permitir analizar mejor el entorno más cercano de ese nuevo punto y, de esta forma, poder confirmar si el punto anterior era efectivamente el que tenía el máximo de similitud o hay uno nuevo en la cercanía del nuevo punto de comienzo que tenga un valor mayor.

El algoritmo se detendrá cuando una iteración utilice un punto de comienzo cuya distancia a uno de los utilizados anteriormente sea menor que un valor prefijado δ (en nuestra implementación $\delta = \rho_r/16$).

La secuencia de todas las operaciones necesarias para llevar a cabo esta búsqueda iterativa de correspondencias se recoge en el Algoritmo 6.1. Éste parte de un punto seleccionado en la nube 1, p_{1x} , con el objetivo de encontrar la correspondencia de un punto de la nube 2 (*punto objetivo*) del que se conoce su imagen CIRCON I_2 . Tras sucesivas iteraciones en las que se va modificando el punto p_{1x} y una vez satisfecha la condición de parada, se obtiene como resultado el punto de la nube 1 (p_{1c}) que obtuvo el mayor valor de similitud (M_{Sc}) de todas las iteraciones. Asimismo se obtiene k_{1c} (giro sobre la normal de p_{1c} para alinear las dos nubes de puntos).

Algoritmo 6.1 Búsqueda de correspondencias.

1. Inicialización de variables y parámetros.
2. **while** no se satisfaga el criterio de parada:
3. Se obtienen las celdas de C_1 que contienen puntos de la nube.
4. **for** cada celda con puntos:
5. **if** $r_{nv} < \tau_{nv}$
 - a) Se escoge el punto p_{1x} (índice ind_{1x}) con normal n_{1x} más próximo al centro de la celda.
 - b) Centrada en ese punto p_{1x} , obtenemos la imagen CIRCON I_{1x} y C_{1x} .
 - c) Se evalúa la similitud (M_{Sx}) entre I_{1x} e I_2 (imagen CIRCON buscada). Obtenemos también el índice de giro (k_{1x}) sobre la normal n_{1x} .
 - d) **if** $M_{Sx} > M_{Sit}$, Se salva la información de esa correspondencia:
 $M_{Sit} = M_{Sx}$; $I_{1it} = I_{1x}$; $C_{1it} = C_{1x}$; $ind_{1it} = ind_{1x}$; $k_{1it} = k_{1x}$
 - e) **end if**
6. **end if**
7. **end for**
8. $I_1 = I_{1it}$ and $C_1 = C_{1it}$ (actualización con los resultados de la iteración).
9. Los índices de los puntos evaluados con peor similitud se añaden a la lista ind_{nv} .
10. Se comprueba la distancia mínima D_{min} del punto p_{1it} a los puntos que han sido almacenados anteriormente en la variable $indexes$.
11. Se evalúa el criterio de parada:
 - a) **if** $D_{min} \leq \delta$, Se finaliza la búsqueda.
 - b) **else**, Se almacenan ind_{1it} , M_{Sit} y k_{1it} en la variable $indexes$.
 - c) **end if**
12. **end while**
13. Se obtiene la máxima similitud almacenada en $indexes$ (M_{Sc}) y sus índices asociados (ind_{1c} y k_{1c}).

6.5. Matriz de transformación asociada a una correspondencia

Cuando el algoritmo de búsqueda encuentra una correspondencia $a \leftrightarrow b$ entre las dos nubes de puntos, se puede calcular, en base a sus puntos ${}^{w_1}p_{1a}$ y ${}^{w_2}p_{2b}$, a sus vectores normales \vec{n}_{1a} y \vec{n}_{2b} y al índice de rotación k_a proporcionado por la medida de similitud, una matriz de transformación entre los dos sistemas de coordenadas globales.

Vamos a llamar a esa matriz ${}_{b \leftrightarrow a}^{w_2}T_{w_1}$, ya que relaciona los sistemas de coordenadas globales, W_1 y W_2 , mediante una correspondencia $a \leftrightarrow b$ (notar que en la matriz se escribe en sentido inverso, $b \leftrightarrow a$, para indicar el orden de las transformaciones intermedias). Para obtener su expresión se han de considerar las transformaciones efectuadas sobre los puntos de ambas nubes para construir las imágenes CIRCON asociadas a esa correspondencia.

En primer lugar, aplicando la Ecuación 3.6, un punto cualquiera ${}^{w_1}p_{1u}$ y su punto potencialmente correspondiente ${}^{w_2}p_{2v}$, se expresaron en los sistemas de coordenadas locales centrados respectivamente en ${}^{w_1}p_{1a}$ y ${}^{w_2}p_{2b}$:

$${}^a p_{1u} = {}^a T_{w_1} \cdot {}^{w_1} p_{1u} \quad (6.3)$$

$${}^b p_{2v} = {}^b T_{w_2} \cdot {}^{w_2} p_{2v} \quad (6.4)$$

A continuación, tras construir las imágenes CIRCON asociadas a la correspondencia $a \leftrightarrow b$, I_{1a} y I_{2b} , se obtuvo el índice k_a equivalente al número de filas que hubo de "rotarse" la imagen para obtener el máximo valor de similitud, M_{Smax} , entre ambas imágenes. Por lo tanto, para alinear el punto ${}^a p_{1u}$ con su correspondiente ${}^b p_{2v}$ bastará con girarlo un ángulo $-k_a \cdot \rho_\theta$ sobre el eje Z_a :

$${}^b p_{1u} = T_{Z_a} \cdot {}^a p_{1u} \quad (6.5)$$

Notar que, por simplicidad, se ha omitido en la expresión el ángulo $-k_a \cdot \rho_\theta$.

Una vez realizada esta última operación, si la correspondencia $a \leftrightarrow b$ es válida, los puntos de ambas nubes estarán aproximadamente alineados, es decir:

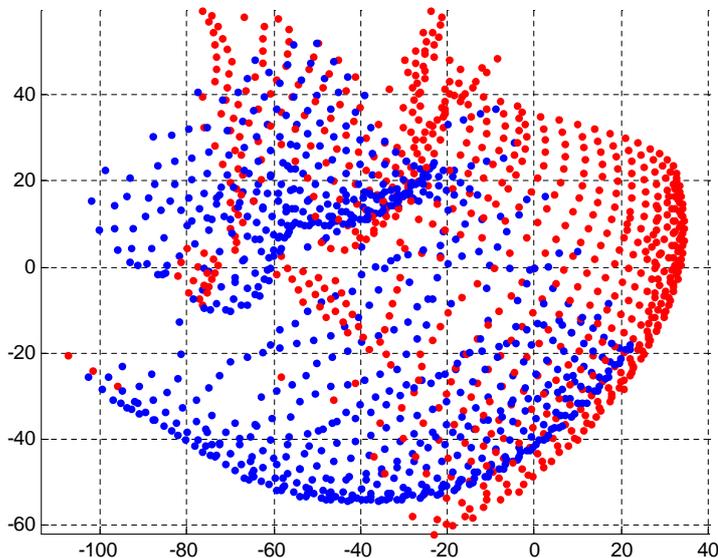


Figura 6.7: Alineamiento aproximado obtenido a partir de la correspondencia encontrada en el ejemplo de la Figura 6.5. Ambas nubes de puntos están expresadas en el sistema de coordenadas local del *punto objetivo*.

$${}^b p_{1u} \approx {}^b p_{2v} \quad (6.6)$$

En la Figura 6.7 se muestra el alineamiento de las dos nubes de puntos de la Figura 6.4 mediante la correspondencia encontrada en el ejemplo de la Figura 6.5. Ambas nubes están expresadas en el sistema de coordenadas local del *punto objetivo*, de modo que los dos puntos correspondidos están en el origen. La rotación que se ha aplicado sobre el eje Z de este s.d.c. viene dada por un índice $k = 8$ (con $n_s = 12$), es decir 240° en sentido horario.

Para referir el alineamiento de la Figura 6.7 al sistema de coordenadas en el que estaba expresada originalmente la nube de puntos 2, habrá que considerar algunas transformaciones adicionales.

Sustituyendo las Ecuaciones 6.3, 6.4 y 6.5 en la Ecuación 6.6, se obtiene:

$$T_{Z_a} \cdot {}^a T_{w_1} \cdot {}^{w_1} p_{1u} \approx {}^b T_{w_2} \cdot {}^{w_2} p_{2v} \quad (6.7)$$

Y despejando ${}^{w_2}p_{2v}$:

$${}^bT_{w_2}^{-1} \cdot T_{Z_a} \cdot {}^aT_{w_1} \cdot {}^{w_1}p_{1u} \approx {}^{w_2}p_{2v} \quad (6.8)$$

Por consiguiente, la transformación euclídea asociada a la correspondencia $a \leftrightarrow b$ que alinea aproximadamente ambas nubes de puntos tendrá como expresión:

$${}_{b \leftrightarrow a}^{w_2}T_{w_1} = {}^bT_{w_2}^{-1} \cdot T_{Z_a} \cdot {}^aT_{w_1} \quad (6.9)$$

Que también puede escribirse en función de sus submatrices:

$${}_{b \leftrightarrow a}^{w_2}T_{w_1} = \begin{bmatrix} {}_{b \leftrightarrow a}^{w_2}R_{w_1} & {}_{b \leftrightarrow a}^{w_2}p_{w_1org} \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (6.10)$$

Siendo,

$${}_{b \leftrightarrow a}^{w_2}R_{w_1} = {}^{w_2}R_b \cdot R_{Z_a} \cdot {}^{w_1}R_a^T \quad (6.11)$$

y

$${}_{b \leftrightarrow a}^{w_2}p_{w_1org} = {}^{w_2}p_b - {}_{b \leftrightarrow a}^{w_2}R_{w_1} \cdot {}^{w_1}p_a \quad (6.12)$$

Como puede observarse, la matriz de transformación asociada a una correspondencia $a \leftrightarrow b$ depende solamente de los puntos correspondidos ${}^{w_1}p_{1a}$ y ${}^{w_2}p_{2b}$, de sus normales (pues ${}^{w_1}R_a$ y ${}^{w_2}R_b$ se construyeron exclusivamente con sus componentes) y del índice k_a que expresa la rotación sobre la normal en ${}^{w_1}p_{1a}$.

Si aplicamos la Ecuación 6.9 a la correspondencia hallada en el ejemplo de la Figura 6.5, se obtiene el alineamiento aproximado que se muestra en la Figura 6.8.

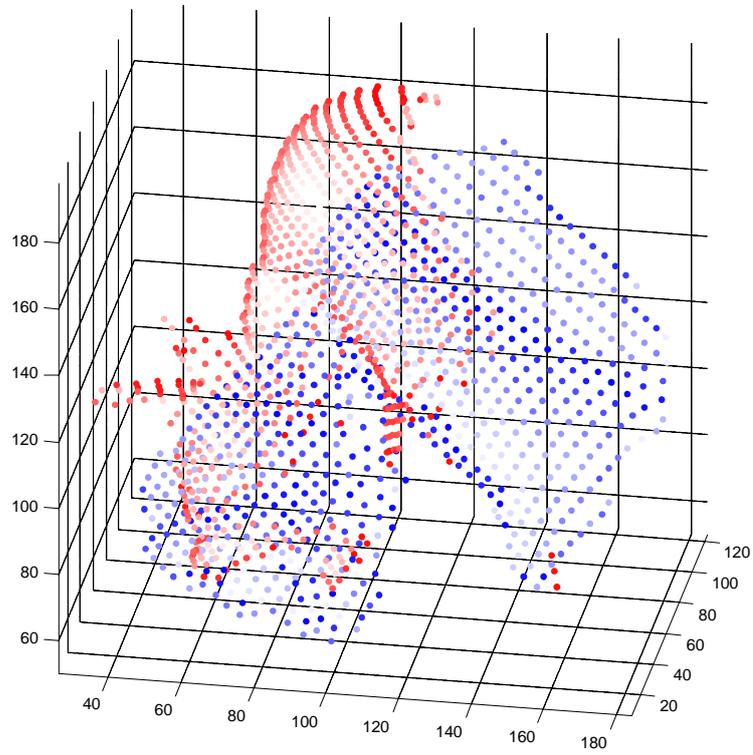


Figura 6.8: Alineamiento aproximado obtenido a partir de la correspondencia encontrada en el ejemplo de la Figura 6.5. Ambas nubes están expresadas en el sistema de coordenadas original de la nube 2.

Capítulo 7

Algoritmo de Alineamiento 3D

7.1. Introducción

El algoritmo presentado en la Sección 6.4 parte de un único punto de interés de la nube 1 que se va modificando iterativamente hasta encontrar la correspondencia aproximada de un punto de interés de la nube 2. Sin embargo, cuando se desean alinear dos nubes de puntos con bajo solapamiento entre sí, puede ocurrir que, si ambas son parciales, un punto seleccionado en la nube 2 no tenga ningún punto correspondiente en la nube 1. Por lo tanto, el algoritmo de alineamiento debería seleccionar varios puntos de interés en la nube 2, $\{p_{2i}\}$, para que haya una mayor probabilidad de encontrar correspondencias válidas. Para cada uno de esos puntos, a los que denominamos "*puntos objetivo*", se llevarán a cabo varias búsquedas de su punto correspondiente en la nube 1 (mediante el Algoritmo 6.1) usando diferentes niveles de resolución. Esas búsquedas deben comenzar, para el primer nivel, en un conjunto de puntos de interés, $\{p_{1i}\}$, previamente seleccionados en la nube 1.

7.2. Datos de entrada y parámetros

El algoritmo de alineamiento 3D que se ha diseñado requiere un conjunto de datos de entrada así como los valores de una serie de parámetros que permiten ajustar su funcionamiento a las necesidades de la aplicación. Las variables que

aparecen en el Algoritmo 7.1 asociadas a dicha información son las siguientes:

$\{p_1\}$: puntos de la nube 1.

$\{n_1\}$: vectores normales en los puntos de la nube 1.

$\{p_2\}$: puntos de la nube 2.

$\{n_2\}$: vectores normales en los puntos de la nube 2.

n_v : número de niveles de resolución.

n_{s1} : número de filas (divisiones angulares) para el primer nivel de resolución.

Partiendo de este valor, los siguientes niveles lo irán aumentando al doble de manera sucesiva. De este modo, para un nivel de resolución i el número de filas será $n_{s1} \cdot (2^{i-1})$.

n_{c1} : número de columnas de búsqueda para el nivel de resolución inicial. Este número se irá reduciendo a la mitad a medida que se aumente la resolución.

τ_{M_S} : vector que contiene los n_v umbrales de similitud (uno para cada nivel de resolución) para que el criterio de parada pueda ser evaluado. Inicialmente valen cero y son actualizados cada vez que se supera su valor.

τ_R : error máximo admisible para los ángulos de Euler ZYX.

τ_t : error máximo admisible para los vectores de traslación.

7.3. Descripción del algoritmo

La estrategia principal del algoritmo de alineamiento consiste en acelerar la búsqueda de correspondencias mediante la utilización de varios niveles de resolución, lo cual va a permitir el sucesivo descarte de zonas donde no va a ser probable que se encuentre la correspondencia del *punto objetivo* p_{2y} . De esta manera, el primer nivel va a tener un número de zonas de búsqueda pequeño, el cual viene limitado por el tamaño del descriptor utilizado para baja resolución, lo que permitirá una rápida convergencia del Algoritmo 6.1. Como las correspondencias se van a buscar solamente en celdas que contengan puntos de la nube, el número de éstas siempre será menor o igual que el número de píxeles de la imagen CIRCON para este primer nivel, o lo que es lo mismo, que el producto del número de divisiones angulares, n_{s1} , por el número de divi-

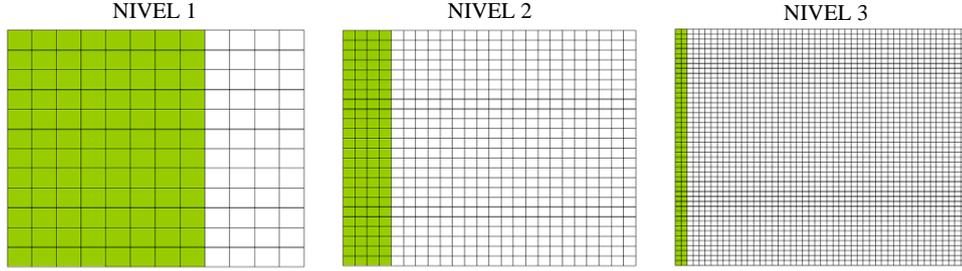


Figura 7.1: Reducción del radio de búsqueda, r_b , para cada nivel de resolución. El número de celdas de búsqueda, n_b , se va a mantener. Para el nivel 1: $n_{s1} = 12$; $n_{c1} = 8$; $n_{b1} = 12 \cdot 8 = 96$; $r_{b1} = 8 \cdot \rho_r$. Para el nivel 2: $n_{s2} = 24$; $n_{c2} = 4$; $n_{b2} = 24 \cdot 4 = 96$; $r_{b2} = 4 \cdot \rho_r$. Para el nivel 3: $n_{s3} = 48$; $n_{c3} = 2$; $n_{b3} = 48 \cdot 2 = 96$; $r_{b3} = 2 \cdot \rho_r$.

siones radiales, n_{r1} . Sin embargo, para reducir el tiempo de computación del algoritmo, el número máximo de celdas de búsqueda para el primer nivel va a ser $8 \cdot n_{s1}$ (ver Figura 7.1).

Una vez que el Algoritmo 6.1 ha encontrado una correspondencia aproximada para el primer nivel, se pasará al siguiente nivel estableciendo una nueva búsqueda en torno a ese nuevo *punto de partida* pero con celdas de menor tamaño. En nuestra implementación, al reducir a la mitad tanto ρ_θ como ρ_r se aumentan al doble el número de divisiones angulares y el número de divisiones radiales, lo cual va a implicar que, para cada nuevo nivel de resolución, las imágenes CIRCON tengan un número de píxeles cuatro veces mayor. Sin embargo, vamos a reducir el radio de búsqueda en torno al *punto de partida* p_{1x} , realizando de esta manera un refinamiento progresivo de la correspondencia con el *punto objetivo* p_{2y} .

Tal y como se muestra en la Figura 7.1, la cantidad de celdas donde se realizará la búsqueda se va a mantener para todos los niveles, de modo que el número de columnas consideradas, n_c , se va a ir reduciendo a la mitad. Y dado que el tamaño de las celdas se reduce cuando se incrementa el nivel de resolución, el radio de búsqueda, r_b , en torno al *punto de partida* va a ser cada vez menor.

Algoritmo 7.1 Alineamiento de dos nubes de puntos.

1. Se seleccionan puntos de interés en la nube 1 $\{p_{1i}\}$ y en la nube 2 $\{p_{2i}\}$.
 2. **while** no se hayan comprobado todos los *puntos objetivo* $\{p_{2i}\}$:
 3. **while** no se hayan comprobado todos los *puntos de partida* $\{p_{1i}\}$:
 4. **for** los n_v niveles de resolución de las imágenes CIRCON:
 5. Se aplica el Algoritmo 6.1 para buscar en la nube 1 (comenzando con p_{1x}) la correspondencia de un punto de interés p_{2y} . Devuelve la correspondencia: p_{1c} , el array de celdas C_{1c} , el índice de giro sobre la normal, k_{1c} , y la medida de similitud, M_{Sc} .
 6. **if** $M_{Sc} < \tau_{Ms}(l)$, Se salta al paso 12; **else** $\tau_{Ms}(l) = M_{Sc}$, **end if**;
 7. **if** el nivel l es el último:
 - a) Se calcula la transformación euclídea ${}_{y \leftrightarrow c}^{w_2} T_{w_1}$ asociada a la correspondencia.
 - b) Se eligen dos celdas de C_{1c} que cumplan ciertas condiciones de ángulo entre normales y distancia con respecto a otras dos celdas de la nube 2.
 - c) Se establecen dos correspondencias entre esas celdas: $m \leftrightarrow n$, $r \leftrightarrow s$
 - d) Con los puntos p_{1c} , p_{1m} , p_{1r} de la nube 1 y los puntos p_{2y} , p_{2n} y p_{2s} de la nube 2 se crea una correspondencia ficticia $t_1 \leftrightarrow t_2$ y se calcula su transformación euclídea asociada: ${}_{t_1 \leftrightarrow t_2}^{w_2} T_{w_1}$.
 - e) Se calculan las distancias d_R y d_t entre las transformaciones ${}_{y \leftrightarrow c}^{w_2} T_{w_1}$ y ${}_{t_1 \leftrightarrow t_2}^{w_2} T_{w_1}$.
 - f) **if** $(d_R < \tau_R)$ **and** $(d_t < \tau_t)$, Se finaliza el algoritmo. **end if**
 8. **end if**
 9. $p_{1x} = p_{1c}$
 10. Se prepara el siguiente nivel de resolución: $n_s = 2 \cdot n_s$; $n_c = n_c/2$; $k_{1c} = 2 \cdot k_{1c}$
 11. **end for**
 12. p_{1x} =Siguiente punto de $\{p_{1i}\}$ (si no ha sido ya evaluado)
 13. **end while**
 14. p_{2y} =Siguiente punto de $\{p_{2i}\}$
 15. **end while**
-

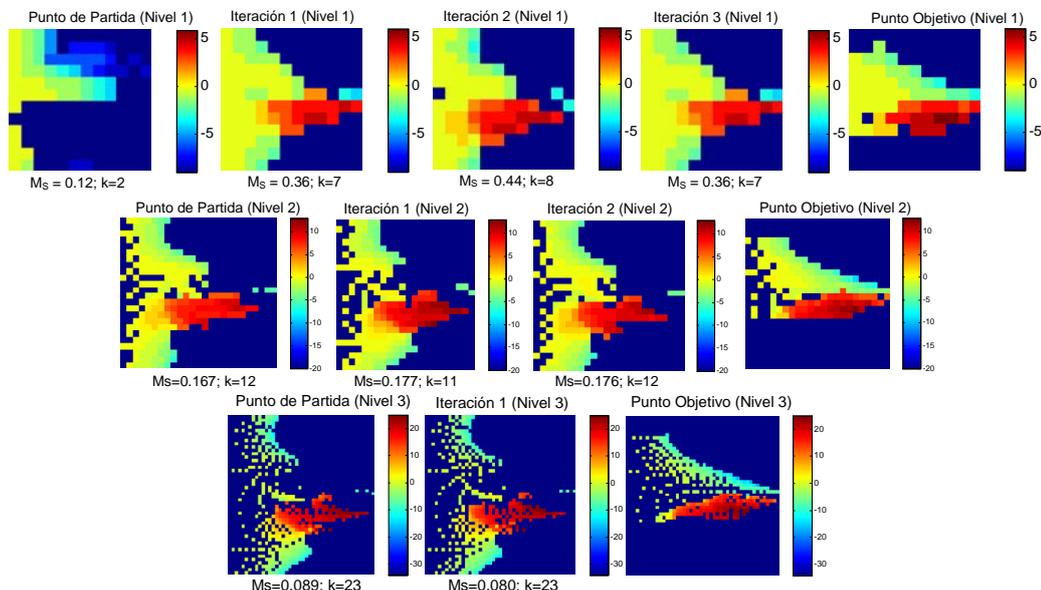


Figura 7.2: Ejemplo de refinamiento de la correspondencia mediante tres niveles de resolución. Las imágenes objetivo son las mostradas en la parte derecha.

Si, cuando se alcanza la convergencia de la búsqueda asociada al último nivel de resolución, el valor de similitud de la correspondencia resultante es mayor que un valor $\tau_{M_S}(n_v)$, se calculará su transformación euclídea correspondiente (mediante la Ecuación 6.9) y se realizará una comprobación de su validez. Si el resultado de esta comprobación es positivo, el alineamiento aproximado de las dos nubes de puntos vendrá determinado por dicha transformación.

En la Figura 7.2 se muestra cómo se obtiene la correspondencia del mismo *punto objetivo* de la Figura 6.4 mediante un refinamiento progresivo con tres niveles y 12 divisiones angulares para el nivel más bajo de resolución. Se eligieron estos valores para la implementación de nuestro algoritmo tras probar diferentes combinaciones para alinear nubes de puntos sintéticas. Se observó que con menos de 10 divisiones angulares el algoritmo era más rápido en ese primer nivel pero favorecía la aparición de más falsas correspondencias que, además de aumentar el tiempo de computación de los siguientes niveles, puede provocar un alineamiento final incorrecto.

Por otro lado, se observó que 48 divisiones angulares para el nivel más alto

de resolución son suficientes para obtener un alineamiento aproximado aceptable. Aunque el máximo error de rotación sobre la normal que podría cometerse es de 3.75° , en la práctica se evalúa la similitud de tantas correspondencias con diferentes orientaciones de la normal que generalmente ese error no suele alcanzarse (tal y como se muestra en los resultados del Capítulo 9). En cualquier caso, si se necesitase más precisión en el alineamiento esta aproximación podría refinarse con la adaptación del algoritmo ICP que se presenta en el Capítulo 8.

En cuanto a la máxima resolución radial, ésta se elige de modo que el número máximo de columnas de la matriz CIRCON sea igual al número de divisiones angulares del nivel más alto de resolución. Para calcular este valor se considera como radio máximo la distancia del punto de la nube más alejado de su centro de masas.

Finalmente, la máxima resolución en altura se fija de modo que los valores de los píxeles de las imágenes CIRCON sean múltiplos del máximo error admisible.

Con esa elección de parámetros puede observarse en la Figura 7.2 cómo se afina la búsqueda y se reduce el número de iteraciones a medida que se incrementa la resolución. Esta reducción del entorno de búsqueda se muestra en la Figura 7.3, donde el número de celdas que deben evaluarse disminuye progresivamente. También se puede observar cómo la celda que tiene el máximo valor se desplaza dentro del mapa de similitud hacia las primeras columnas; esto significa que la búsqueda va convergiendo y que, por lo tanto, el número de columnas a evaluar puede reducirse.

7.4. Evaluación del criterio de parada

Cuando el algoritmo ha llegado al último nivel de resolución y ha superado su valor correspondiente de $\tau_{MS}(n_v)$ (que es el máximo de similitud encontrado hasta ese momento), se llevará a cabo la evaluación del criterio de parada del algoritmo.

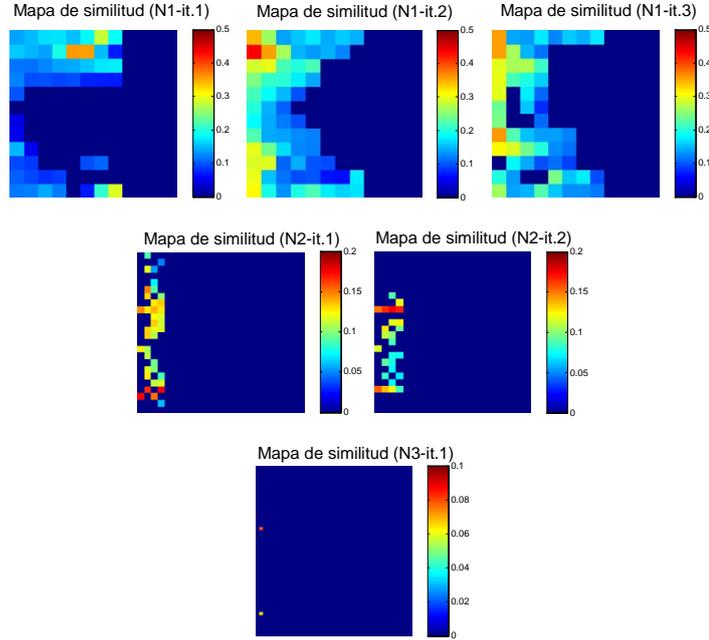


Figura 7.3: Reducción del número de celdas de búsqueda en los mapas de similitud al aumentar el nivel de resolución. El número de columnas de búsqueda es el indicado en la Figura 7.1.

Este criterio utiliza dos correspondencias adicionales, $m \leftrightarrow n$ y $r \leftrightarrow s$, que son seleccionadas mediante la distribución de celdas en torno a la correspondencia $a \leftrightarrow b$ encontrada por el algoritmo. La condición que deben satisfacer los dos puntos seleccionados en la nube 1 es que los ángulos que formen sus vectores normales con la normal del punto que se está evaluando deben ser muy similares a los que formen sus puntos correspondientes.

Estas dos correspondencias adicionales van a permitir calcular una nueva transformación euclídea que no depende de los vectores normales de los puntos de la nube. Para ello se va a crear una correspondencia ficticia $t_1 \leftrightarrow t_2$ cuyos puntos p_{t1} y p_{t2} estarán situados en el centroide de los dos triángulos que forman los puntos p_{1a} , p_{1m} y p_{1r} en la nube 1 y los puntos p_{2b} , p_{2n} y p_{2s} en la nube 2 (ver Figura 7.4). Los vectores normales \vec{n}_{t1} y \vec{n}_{t2} que corresponden a los puntos ficticios serán perpendiculares a dichos triángulos.

Para que fuese válida la transformación euclídea obtenida por el algoritmo, la correspondencia $t_1 \leftrightarrow t_2$ debería cumplir las Ecuaciones C.22 y C.23. Sin

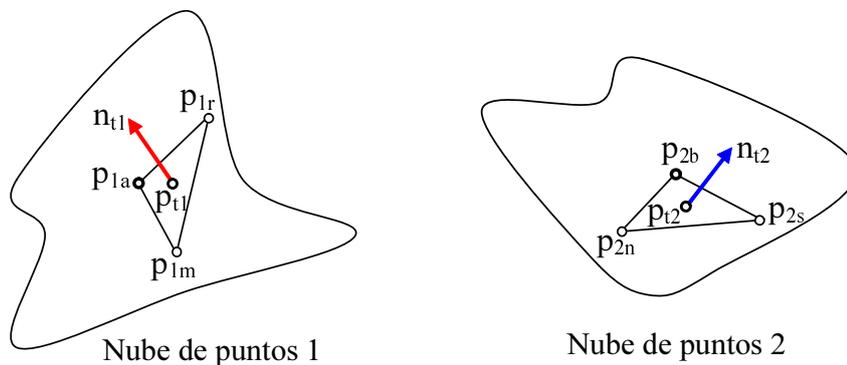


Figura 7.4: Evaluación del criterio de parada. Los puntos p_{t_1} y p_{t_2} , y sus respectivos vectores normales, se obtienen a partir de la correspondencia encontrada por el algoritmo de alineamiento ($a \leftrightarrow b$) y dos correspondencias adicionales ($m \leftrightarrow n$ y $r \leftrightarrow s$). La nueva correspondencia $t_1 \leftrightarrow t_2$ se utilizará para calcular una transformación euclídea más robusta.

embargo estas condiciones dependen de los vectores normales de la correspondencia $a \leftrightarrow b$, por lo que vamos a calcular una transformación que no dependa directamente de las normales en los puntos de la nube sino de las normales \vec{n}_{t_1} y \vec{n}_{t_2} que se han calculado usando tres puntos distantes entre sí (en nuestra implementación deben estar separados al menos una distancia $5 \cdot \rho_r$). Cuanto mayor sea esa distancia entre los puntos de esos triángulos menos vulnerables van a ser \vec{n}_{t_1} y \vec{n}_{t_2} a pequeños cambios en la posición de los puntos.

Por lo tanto, se puede calcular para esa correspondencia ficticia $t_1 \leftrightarrow t_2$ una transformación ${}_{t_2 \leftrightarrow t_1}^{w_2} T_{w_1}$ mediante la Ecuación 6.9. Para ello será necesario construir las imágenes CIRCON en p_{t_1} y p_{t_2} y evaluar su similitud para obtener el índice de rotación k_{t_1} sobre la normal \vec{n}_{t_1} . De esta manera podemos también comprobar que la transformación ${}_{t_2 \leftrightarrow t_1}^{w_2} T_{w_1}$ está asociada a imágenes CIRCON con un valor de similitud alto, lo cual nos da información de la validez de esta transformación. Aunque podríamos utilizar este indicador para decidir si la correspondencia $a \leftrightarrow b$ es suficientemente buena o, por el contrario, el algoritmo debería seguir buscando una mejor, vamos a basar, en cambio, esa decisión en la diferencia existente entre las transformaciones ${}_{b \leftrightarrow a}^{w_2} T_{w_1}$ y ${}_{t_2 \leftrightarrow t_1}^{w_2} T_{w_1}$. Para ello vamos a utilizar una medida para la rotación y otra para la traslación.

Si las matrices de rotación de ambas transformaciones fuesen iguales, se

debería cumplir:

$${}_{t_2 \leftrightarrow t_1}^{w_2} R_{w_1}^T \cdot {}_{b \leftrightarrow a}^{w_2} R_{w_1} = I_{3 \times 3} \quad (7.1)$$

Dado que estas matrices no van a ser exactamente iguales, la ecuación anterior dará como resultado una matriz E_R que se aproximará más a la matriz identidad cuanto más parecidas sean las matrices de rotación. Como la matriz $I_{3 \times 3}$ se puede considerar una matriz de rotación cuyos ángulos de Euler ZYX son todos nulos, podemos obtener para la matriz E_R sus ángulos correspondientes, $(\alpha_R, \beta_R, \gamma_R)$, y compararlos con el caso ideal. Por lo tanto, para obtener una medida relacionada con la diferencia entre las matrices de rotación utilizaremos la siguiente expresión:

$$d_R = \sqrt{\frac{1}{3}(\alpha_R^2 + \beta_R^2 + \gamma_R^2)} \quad (7.2)$$

Para obtener la medida relacionada con la traslación se podría pensar que basta con obtener la desviación *rms* entre los vectores de traslación. Sin embargo, puede suceder que las nubes de puntos estén aceptablemente alineadas en ambos casos y que la diferencia entre los vectores de traslación sea muy grande. Esto se debe a que una ligera diferencia en la rotación puede producir un gran cambio en el vector de traslación cuando las nubes de puntos están inicialmente muy distanciadas entre sí. Para evitar que esto suceda vamos a calcular un vector $\delta_T = (\delta_x, \delta_y, \delta_z)^T$ mediante la siguiente expresión:

$$\delta_T = ({}_{t_2 \leftrightarrow t_1}^{w_2} p_{w_1 org} - {}_{b \leftrightarrow a}^{w_2} p_{w_1 org}) + ({}_{t_2 \leftrightarrow t_1}^{w_2} R_{w_1} - {}_{b \leftrightarrow a}^{w_2} R_{w_1}) \cdot \bar{p}_1 \quad (7.3)$$

Siendo ${}_{b \leftrightarrow a}^{w_2} p_{w_1 org}$ el vector de traslación para la correspondencia $a \leftrightarrow b$, ${}_{t_2 \leftrightarrow t_1}^{w_2} p_{w_1 org}$ el obtenido para la correspondencia ficticia y \bar{p}_1 la media de los puntos de la nube 1. Si inicialmente se restó la media a todos los puntos de la nube, el último sumando de la expresión se anularía y δ_T sería simplemente la diferencia de los vectores de traslación. Sin embargo, habría que tener en cuenta los vectores \bar{p}_1 y \bar{p}_2 para obtener la transformación euclídea que realmente alinea las nubes de puntos originales.

Por lo tanto, para obtener la medida relativa a la traslación se utilizará:

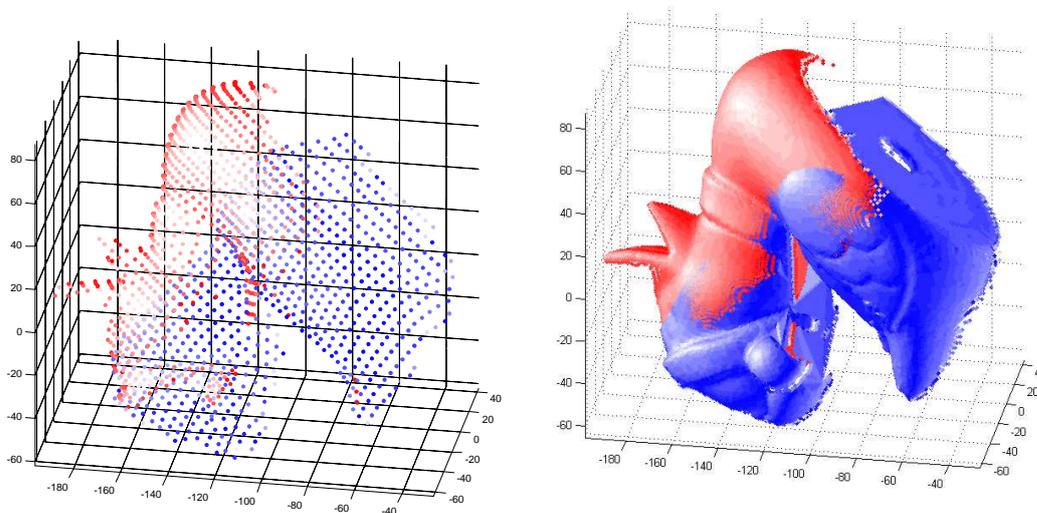


Figura 7.5: Alineamiento aproximado de las nubes de puntos reducidas mediante la transformación obtenida por el algoritmo (izquierda). Alineamiento aproximado de las nubes de puntos originales (derecha).

$$d_t = \sqrt{\frac{1}{3}(\delta_x^2 + \delta_y^2 + \delta_z^2)} \quad (7.4)$$

Para que la correspondencia sea considerada válida se deben cumplir simultáneamente dos condiciones: d_R debe ser menor que la distancia máxima admisible para rotación, τ_R , y d_t , a su vez, debe ser menor que τ_t , que es el umbral para traslación.

En la Figura 7.5 se muestra el alineamiento aproximado obtenido por el algoritmo para la correspondencia encontrada en el ejemplo de la Figura 7.2 tras verificar su validez con $\tau_R = 4^\circ$ y $\tau_t = 2\text{mm}$. Asimismo se muestra el resultado de aplicar la misma transformación euclídea a las nubes de puntos originales.

Si el algoritmo no fuese capaz de encontrar un par de correspondencias adicionales que permitan ratificar la validez de al menos una correspondencia, el algoritmo finalizaría tras acabar la comprobación de todos los *puntos de partida* de la nube 1 y todos los *puntos objetivo* de la nube 2. En este caso, la transformación euclídea sería la resultante de la correspondencia con mayor valor de similitud.

7.5. Factor de Corrección del Solapamiento

Cuando se pretende alinear dos nubes de puntos con bajo solapamiento es más probable que aparezcan falsas correspondencias, sobre todo si el nivel de resolución de los descriptores CIRCON es bajo. Esto se debe a que la probabilidad de que el entorno de un punto de interés se parezca al entorno del *punto objetivo* es mayor cuando ambos están representados por un número pequeño de celdas, lo cual es más probable que ocurra en el primer nivel de resolución.

Si el porcentaje de solapamiento entre los dos descriptores de una falsa correspondencia fuese sensiblemente mayor que el de cualquiera de las correspondencias buenas (lo cual puede llegar a ocurrir en algunas situaciones) y la distancia entre las áreas solapadas no fuese un factor claramente diferenciador debido a la baja resolución, podría ocurrir que el valor de similitud de alguna falsa correspondencia fuese del orden de los de las buenas.

Si el bajo solapamiento entre las nubes de puntos impidiese encontrar un par de correspondencias adicionales que permitan ratificar la validez de una correspondencia buena (tal y como se explica en la Sección 7.4), el algoritmo finalizaría tras acabar la comprobación de todos los *puntos de partida* de la nube 1 y todos los *puntos objetivo* de la nube 2. En este caso, las correspondencias evaluadas deberían estar ordenadas correctamente en función de su valor de similitud. Para que, en este caso, se consigan rechazar todas las falsas correspondencias, se aplicará un factor de corrección al porcentaje de solapamiento en el último nivel de resolución.

Este factor de corrección va a depender de la variación media de D_S (Ecuación 5.16) en torno al índice de rotación k correspondiente al mayor valor de similitud:

$$\Delta D_{Smed} = \frac{D_S(k-1) - 2 \cdot D_S(k) + D_S(k+1)}{2} \quad (7.5)$$

Y el factor de corrección Ψ va a tener la siguiente expresión:

$$\Psi = e^{(1 - \frac{\max\{D_S(k-1), D_S(k), D_S(k+1)\}}{\Delta D_{Smed}})} \quad (7.6)$$

Éste será igual a la unidad cuando el valor ΔD_{Smed} coincida con el máximo de los tres valores de D_S , lo cual se produce en el caso ideal, cuando ambos descriptores son iguales y se cumple que $D_S(k) = 0$ y $D_S(k-1) = D_S(k+1)$. Por el contrario, dicho factor será nulo cuando ΔD_{Smed} valga cero, es decir, cuando los tres valores sean iguales y no haya variación de la medida de similitud con el índice k .

Por consiguiente, cuando el algoritmo haya encontrado una correspondencia para el último nivel de resolución habrá que recalcular la medida de similitud (Ecuación 5.18) con un porcentaje de solapamiento corregido, tal que:

$$\sigma_{ov} = \sigma_{ov} \cdot \Psi \quad (7.7)$$

Capítulo 8

Adaptación del Algoritmo ICP

Como se comentó en la Sección 2.5, los algoritmos de alineamiento fino necesitan partir de una transformación euclídea que alinee de forma aproximada las dos nubes de puntos. Utilizando esta posición de partida, se establecen correspondencias entre puntos (basándose fundamentalmente en su proximidad) que se utilizarán para calcular una nueva matriz de transformación que se irá refinando de manera iterativa.

En este caso, el algoritmo ICP se va a adaptar para aprovechar el conjunto de datos que proporciona el algoritmo de alineamiento 3D descrito en el Capítulo 7. La transformación euclídea hallada se utilizará para inicializar el algoritmo ICP, mientras que los arrays, M_1 y M_2 , que contienen los índices de los puntos de cada nube distribuidos en celdas en torno a un punto central, se utilizarán para establecer las correspondencias.

Por consiguiente, la adaptación del algoritmo ICP que se utiliza en esta tesis es la siguiente:

1. Se aplica a la nube 1 la transformación euclídea obtenida por el algoritmo de alineamiento 3D.
2. **for** cada celda solapada (i, j) de los arrays de celdas M_1 y M_2 :
 - a) Se eligen aleatoriamente un 10 % de los puntos de la celda (i, j) de $M_1 \Rightarrow \{p_{1ij}\}$

- b) Para cada punto seleccionado p_{1ij_k} se busca el punto más cercano en la celda (i, j) de M_2 . Este punto, p_{2ij_k} , debe cumplir lo siguiente:
- El ángulo que forma su normal, n_{2ij_k} , con la del punto p_{1ij_k} debe ser menor que un valor prefijado (en nuestra implementación 5°).
 - Su distancia al punto p_{1ij_k} debe ser menor que la distancia máxima entre los puntos correspondidos del paso anterior (para la primera iteración utilizamos una distancia máxima de 10mm, ya que la transformación inicial es una aproximación).
- c) Se asigna un peso $w_{ij_k} = j$ a cada correspondencia (p_{1ij_k}, p_{2ij_k}) .
- d) Se añaden las correspondencias y los pesos a los conjuntos $\{p_{1C}, p_{2C}\}$ y $\{w\}$.

3. end for

4. Se calcula la distancia *rms* para el conjunto de puntos correspondidos:

$$D_C = \sqrt{\frac{1}{N_C} \cdot \sum_{k=1}^{N_C} w_k \cdot \|p_{1Ck} - p_{2Ck}\|^2}, \text{ siendo } N_C \text{ el número de correspondencias encontradas.}$$

5. Con el conjunto de correspondencias $\{p_{1C}, p_{2C}\}$ y sus pesos asociados, $\{w\}$, se halla la transformación rígida correspondiente utilizando la solución propuesta en [76].
6. Se aplica la nueva transformación calculada a la nube de puntos 1 y a sus vectores normales.
7. Se busca entre los puntos del centro y de la primera columna del array de celdas M_1 cuál es el más cercano al punto de la nube 2 en el que está centrado el array M_2 .
8. Se obtiene el array M_1 para el nuevo punto.
9. Se vuelve al paso 2 si la diferencia entre los dos últimos valores de D_C es mayor que un valor prefijado. Si no, el algoritmo finaliza.

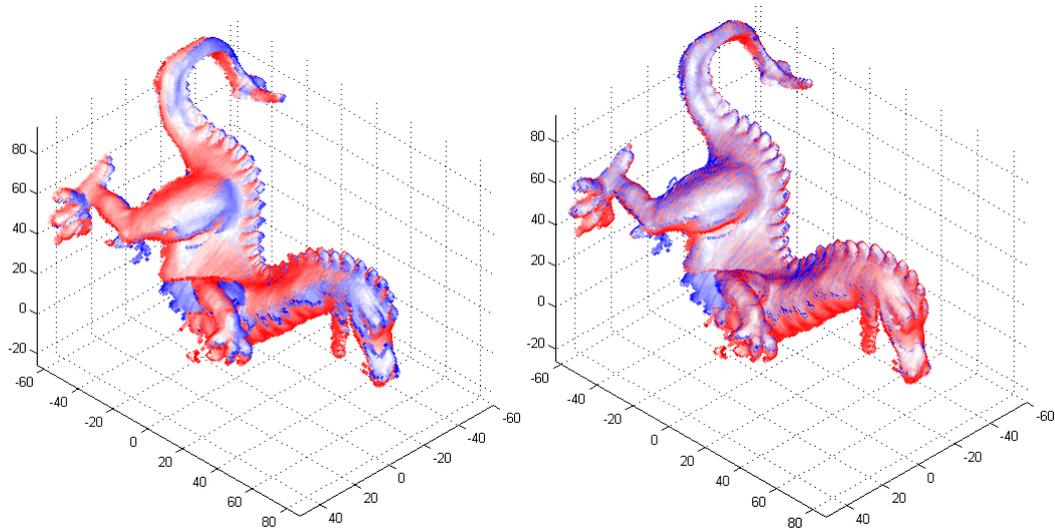


Figura 8.1: Alineamiento aproximado obtenido mediante el Algoritmo 7.1 (izquierda). Alineamiento obtenido mediante la adaptación del algoritmo ICP: $d_{RMS} = 0,5529$ (derecha).

Al asignar a cada correspondencia un peso w_k que depende de su distancia al punto central de la imagen CIRCON, se van a tener más en cuenta en el cálculo de la transformación las correspondencias más alejadas; de este modo se trata de evitar que el error de alineamiento sea mayor en estos puntos. Utilizando estos pesos se aplicará la solución en forma cerrada presentada por *Maurer* en [76] para calcular la matriz de transformación.

Esta adaptación del algoritmo ICP se utiliza en el Capítulo 9 para obtener la transformación euclídea con la que comparar los resultados del algoritmo de alineamiento propuesto. Esta se considerará como un patrón de comparación, ya que la distancia d_{RMS} entre puntos correspondientes no superó en ningún caso los 0.7 mm, siendo incluso mucho menor para algunos objetos.

Como se muestra en la Figura 8.1, si se parte de un buen alineamiento inicial (obtenido con el Algoritmo 7.1), esta variante del ICP va a permitir refinar la transformación euclídea hasta llegar a un resultado lo suficientemente preciso como para considerarlo un modelo comparativo válido.

Capítulo 9

Resultados experimentales

9.1. Introducción

Los experimentos que se han llevado a cabo y que se presentan en este capítulo tienen dos objetivos fundamentales:

- Demostrar que la medida de similitud es capaz de seleccionar las mejores correspondencias para diferentes grados de solapamiento cuando los descriptores se construyen utilizando los puntos de las nubes parciales reducidas (Sección 9.3).
- Aplicar el algoritmo de alineamiento a nubes de puntos de diferente procedencia y mostrar su robustez para encontrar una transformación rígida aproximada (Sección 9.4).

Para obtener los resultados presentados en este capítulo, el Algoritmo 7.1 se implementó en *Matlab*[®]. En todos los casos se utilizaron 3 niveles de resolución con 12, 24 y 48 divisiones angulares respectivamente. Esta elección es resultado de un compromiso entre el poder discriminante a bajo nivel de resolución y el tiempo de procesamiento en ese nivel, mientras que la elección de 48 divisiones angulares para el máximo nivel aseguran, como se verá en los resultados, un alineamiento de las nubes de puntos suficientemente bueno para aplicaciones de modelado 3D y de localización de objetos para su captura con un manipulador robótico.

Dado el carácter experimental de la implementación del algoritmo, se ha primado fundamentalmente la funcionalidad del programa; por esta razón, aunque también se ha tratado de aprovechar al máximo las posibilidades que ofrece el lenguaje de programación de *Matlab*[®], el código no está suficientemente optimizado. Al igual que en el estudio comparativo de *Salvi et al.*[89] no se va a llevar a cabo un análisis de eficiencia del algoritmo ya que, como ahí se apunta, ésta es muy dependiente de la implementación. Sin embargo, indicaremos a modo de referencia que, cuando las nubes de puntos tienen un porcentaje de solapamiento superior al 70 %, el tiempo invertido por el algoritmo de alineamiento no supera en la gran mayoría de los casos los 5 segundos, mientras que para valores inferiores de solapamiento el tiempo puede llegar a las decenas de segundos. En este caso el incremento de tiempo se debe, por una parte, a que se tienen que utilizar más puntos de partida del algoritmo para evitar falsas correspondencias y por otra parte a que la elección de dichos puntos no sea la adecuada (dada la sencillez del algoritmo de selección de puntos de interés), lo cual implica, en ambos casos, un número adicional de iteraciones.

En definitiva, los resultados se analizarán en términos de eficacia del algoritmo, que vendrá caracterizada por una medida del error cometido en la matriz de transformación que alinea de forma aproximada las dos nubes de puntos.

9.2. Medida del error cometido en el alineamiento aproximado

Si llamamos R_c a la matriz de rotación aproximada obtenida por el algoritmo de alineamiento 3D y R_{ICP} a la transformación obtenida mediante el algoritmo ICP, se debería cumplir en el caso ideal:

$$R_{ICP}^T \cdot R_c = I_{3 \times 3} \quad (9.1)$$

Como la matriz R_c va a ser una aproximación, la ecuación anterior dará como resultado una matriz E_R que debería ser aproximadamente igual a la

matriz identidad. Como la matriz $I_{3 \times 3}$ se puede considerar una matriz de rotación cuyos ángulos de Euler ZYX son todos nulos, podemos obtener para la matriz E_R sus ángulos correspondientes, $(\alpha_R, \beta_R, \gamma_R)$, y compararlos con la situación ideal. Por lo tanto, podemos calcular un error de rotación con la siguiente expresión:

$$e_R = \sqrt{\frac{1}{3}(\alpha_R^2 + \beta_R^2 + \gamma_R^2)} \quad (9.2)$$

Y para obtener el error de traslación se tendrán en cuenta las mismas consideraciones que en el criterio de parada (Sección 7.4). De modo que se va a calcular un vector $\varepsilon_t = (\varepsilon_x, \varepsilon_y, \varepsilon_z)^T$ mediante la siguiente expresión:

$$\varepsilon_t = (t_c - t_{ICP}) + (R_c - R_{ICP}) \cdot \bar{p}_1 \quad (9.3)$$

Siendo t_c el vector de traslación de la transformación euclídea aproximada, t_{ICP} el obtenido mediante el algoritmo ICP y \bar{p}_1 la media de los puntos de la nube 1.

Por lo tanto, para obtener la medida relativa a la traslación se utilizará:

$$e_t = \sqrt{\frac{1}{3}(\varepsilon_x^2 + \varepsilon_y^2 + \varepsilon_z^2)} \quad (9.4)$$

9.3. Evaluación de la medida de similitud

En este experimento vamos a comparar los resultados de alineamiento 3D obtenidos con la medida de similitud presentada en el Capítulo 5 con los obtenidos mediante dos medidas de similitud comúnmente usadas por los algoritmos de alineamiento 3D: la correlación cruzada normalizada y la medida de información mutua.

El objetivo de este experimento es caracterizar cómo se comporta el algoritmo descrito en el Capítulo 7 con cada una de estas tres medidas de similitud cuando se modifica el porcentaje de solapamiento de las dos nubes de puntos que se quieren alinear.

Para poder realizar esta comparación se van a utilizar imágenes de rango

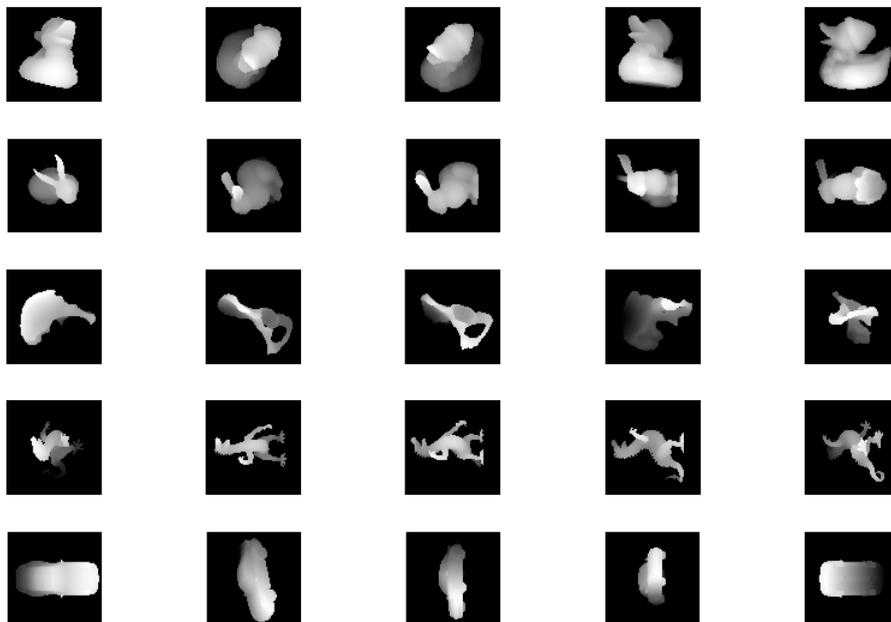


Figure 9.1: Imágenes de rango de cinco objetos ('ducky', 'bunny', 'hip', 'dragon', 'porsche') de la base de datos de la Universidad de Stuttgart [56] que se utilizarán para evaluar la medida de similitud del Capítulo 5.

sintéticas obtenidas de la base de datos de la Universidad de Stuttgart [56] (aunque la mayoría se han obtenido a partir de modelos poligonales de alta resolución que están disponibles en bases de datos de internet como el *Stanford 3D Scanning Repository* [2] y el *Georgia Tech Large Geometric Models Archive* [1]). Estas imágenes de rango se van modificar para obtener dos nubes de puntos parciales no alineadas y con un determinado porcentaje de solapamiento.

Para ello, para cada una de las imágenes de rango mostradas en la Figura 9.1 se calcularán los vectores normales en sus puntos y se crearán dos nuevas imágenes de rango. Para obtener la primera se eliminarán de la imagen de rango los N_1 primeros píxeles de objeto mientras que para la segunda los N_2 últimos. Si la nube original tenía N puntos, entonces el número de puntos solapados, N_S , se calcula como: $N_S = N - (N_1 + N_2)$. Por lo tanto, el porcentaje de puntos solapados entre las dos nubes se obtiene mediante la siguiente expresión:

$$\sigma_{nubes} = \frac{N_S}{N} \cdot 100 = \left(1 - \frac{N_1 + N_2}{N}\right) \cdot 100 \quad (9.5)$$

A continuación se obtienen las nubes de puntos correspondientes a esas dos imágenes de rango. Como en principio van a estar alineadas, se va a aplicar una transformación rígida a la segunda nube de puntos. Para ello se obtiene, de forma aleatoria, un vector de rotación (los tres ángulos de Euler ZYX) y un vector de traslación. Una vez aplicada esta transformación, se va a modificar ligeramente la posición de los puntos de esta segunda nube añadiendo ruido pseudoaleatorio con desviación estándar de 1 mm. A continuación, se aplicará el algoritmo descrito en el Capítulo 7 para alinear las dos nubes de puntos.

Mediante este procedimiento que se ha descrito se va a poder conocer de antemano la transformación euclídea ideal, qué puntos son realmente correspondientes y qué partes de la nubes van a estar o no solapadas. De esta forma se podrá calcular el error cometido en el vector de rotación y en el de traslación, además de poder comprobar la influencia que tiene en el alineamiento el grado de solapamiento entre las nubes de puntos.

Los descriptores CIRCON se van a generar en los puntos de interés utilizando como región circundante la totalidad de las nubes de puntos; de esta manera se va a demostrar que, aun habiendo un bajo solapamiento entre ellas, la medida de similitud es capaz de encontrar una correspondencia que permite alinear las dos nubes de forma aproximada. Esto significa una gran diferencia con respecto a otros algoritmos que utilizan descriptores 2D, pues, en presencia de bajo solapamiento, debe reducirse el tamaño de éstos para facilitar a la medida de similitud la clasificación de las correspondencias. Sin embargo, esa reducción de tamaño implica una reducción del poder discriminante, lo cual puede provocar la aparición de un alto porcentaje de falsas correspondencias y, por consiguiente, la necesidad de un costoso post-procesamiento para poder descartarlas.

Las Figuras 9.2, 9.3, 9.4, 9.5 y 9.6 muestran los resultados del experimento de evaluación del algoritmo con tres medidas de similitud. En el eje de abscisas aparece reflejado el grado de solapamiento de las nubes de puntos, mientras el eje de ordenadas se corresponde con el número de puntos de la nube reducida que se utilizó para crear dos nubes con diferentes grados de solapamiento. Es decir, 1500 puntos de la nube reducida para un 30 % de solapamiento signi-

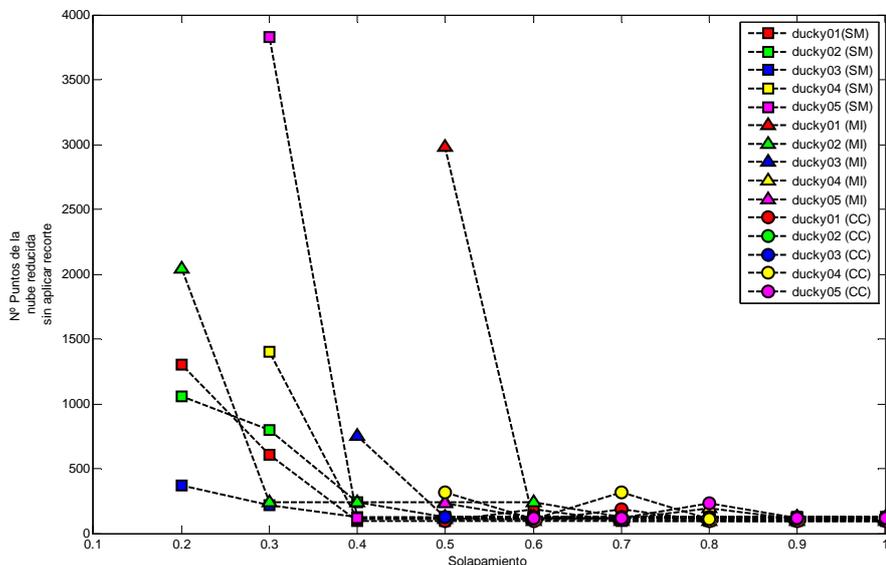


Figura 9.2: Comportamiento del algoritmo de alineamiento con tres medidas de similitud y cinco imágenes de rango del objeto 'ducky': la medida propuesta (SM), información mutua (MI) y coeficiente de correlación cruzada (CC). Se varía el grado de solapamiento y se incrementa el número de puntos de la nube (antes del 'recorte') hasta que el error de rotación sea inferior a 5 grados y el de traslación menor de 5 mm.

fica que cada nube tiene (haciendo $N_{nube} = N_1 = N_2$ en la Ecuación 9.5 y despejando): $N_{nube} = \left(\frac{1-0.3}{2}\right) \cdot 1500 = 525$ puntos.

La Figura 9.2 muestra, para el objeto 'ducky', cómo la medida de similitud es capaz de encontrar un alineamiento apropiado para las imágenes de rango 'ducky01', 'ducky02' y 'ducky03' con un porcentaje de solapamiento del 20%. Sin embargo, para ese mismo porcentaje y usando la información mutua como medida de similitud, el algoritmo solamente encuentra una solución buena para la imagen de rango 'ducky03', que además tiene que utilizar una nube con mayor densidad de puntos que en el caso de nuestra medida de similitud. Por otro lado, usando el coeficiente de correlación cruzada, el algoritmo comienza a fallar cuando el porcentaje de solapamiento es menor del 50%.

Notar que, para un porcentaje de solapamiento del 30% y utilizando la medida de similitud propuesta, se encontró un alineamiento adecuado para las

cinco imágenes de rango, mientras que solamente se encontró con la información mutua para la imagen 'ducky02' y falló en todos los casos para el coeficiente de correlación cruzada.

Notar también que, con la medida propuesta, solamente en un caso ('ducky05' con 30 % de solapamiento) se necesitó una nube reducida con más de 1500 puntos. De hecho, con solapamientos mayores que el 30 % ese número no superó los 400 puntos. Sin embargo, para 'ducky05', las otras dos medidas de similitud, MI y CC, solamente encontraron una solución buena para un solapamiento del 50 % y 60 % respectivamente.

Por último, se puede observar en la Figura 9.2 que el algoritmo falla en todos los casos (con las condiciones que hemos fijado) cuando el solapamiento es menor del 20 %.

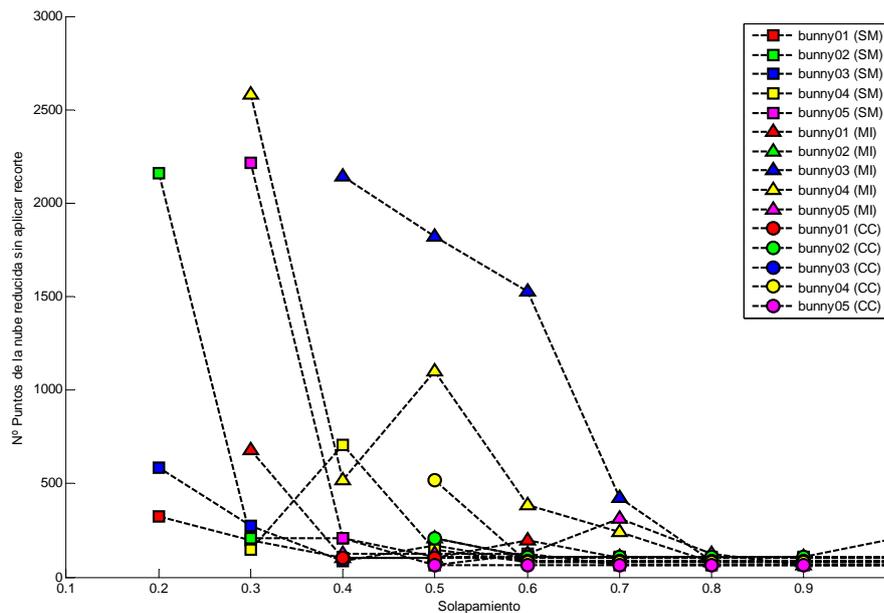


Figura 9.3: Comportamiento del algoritmo de alineamiento con tres medidas de similitud y cinco imágenes de rango del objeto 'bunny': la medida propuesta (SM), información mutua (MI) y coeficiente de correlación cruzada (CC). Se varía el grado de solapamiento y se incrementa el número de puntos de la nube (antes del 'recorte') hasta que el error de rotación sea inferior a 5 grados y el de traslación menor de 5 mm.

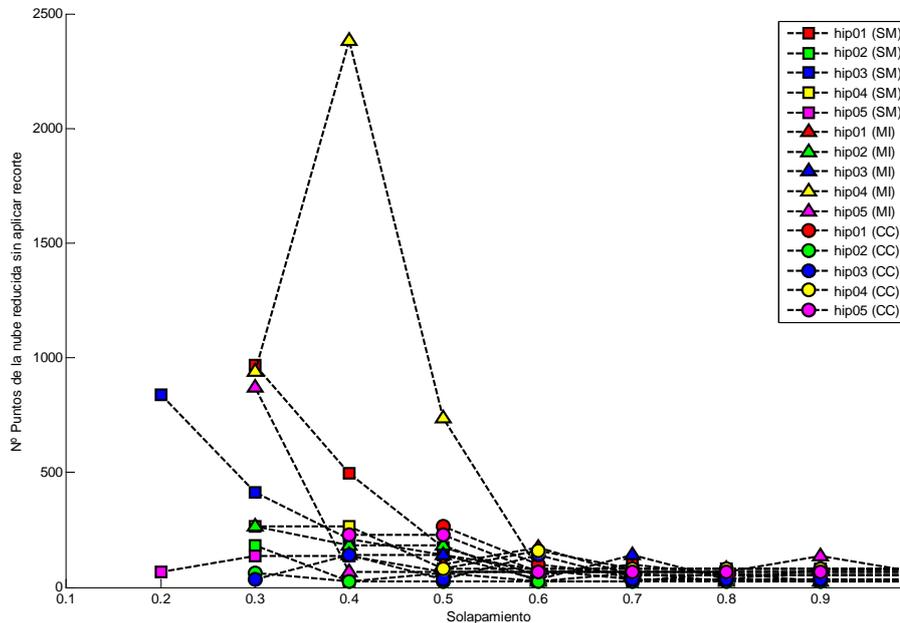


Figura 9.4: Comportamiento del algoritmo de alineamiento con tres medidas de similitud y cinco imágenes de rango del objeto 'hip': la medida propuesta (SM), información mutua (MI) y coeficiente de correlación cruzada (CC). Se varía el grado de solapamiento y se incrementa el número de puntos de la nube (antes del 'recorte') hasta que el error de rotación sea inferior a 5 grados y el de traslación menor de 5 mm.

Para el resto de objetos se produjeron resultados similares. La medida de similitud propuesta permitió encontrar un buen alineamiento para las cinco imágenes de rango (excepto para el objeto 'porsche') cuando el solapamiento es igual al 30% y para algunas de ellas cuando es del 20%.

Por el contrario, la medida de información mutua obtuvo algún buen resultado para esos porcentajes pero solamente en casos aislados. Además mostró una cierta inestabilidad en el número de puntos necesario para llegar a un buen resultado (como ocurre por ejemplo en la Figura 9.4 para la imagen de rango 'hip04').

En cuanto al coeficiente de correlación cruzada, éste no logró obtener alineamientos buenos para porcentajes de solapamiento inferiores al 40% excepto

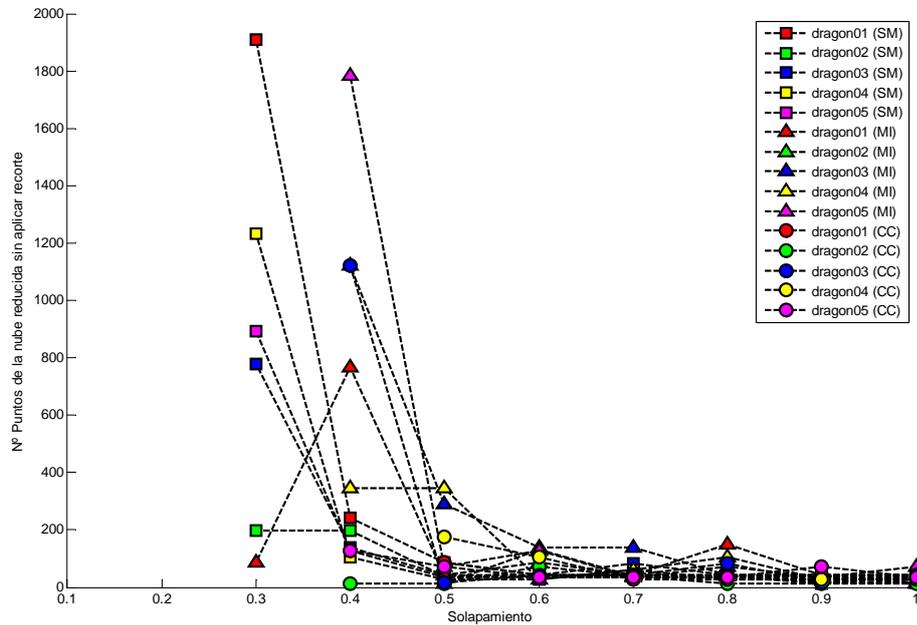


Figura 9.5: Comportamiento del algoritmo de alineamiento con tres medidas de similitud y cinco imágenes de rango del objeto 'dragon': la medida propuesta (SM), información mutua (MI) y coeficiente de correlación cruzada (CC). Se varía el grado de solapamiento y se incrementa el número de puntos de la nube (antes del 'recorte') hasta que el error de rotación sea inferior a 5 grados y el de traslación menor de 5 mm.

para el objeto 'hip' que llegó al 30 % con 'hip02' y 'hip03' (Figura 9.4). Además, en la mayoría de los casos falló con porcentajes inferiores al 50 %.

Como ya se ha comentado, en el caso del objeto 'porsche' (Figura 9.6) solamente se obtuvo un buen alineamiento para la medida de similitud propuesta y solapamiento del 30 % para 'porsche02', 'porsche03' y 'porsche04'. En el caso de 'porsche01' no se obtuvo un buen resultado para un solapamiento inferior al 40 % y para 'porsche05' este límite se fijó en el 50 %. Sin embargo, con la información mutua y la correlación cruzada los resultados fueron aun peores: para 'porsche01' no se pudo bajar del 50 % y para 'porsche05' ni siquiera del 70 %.

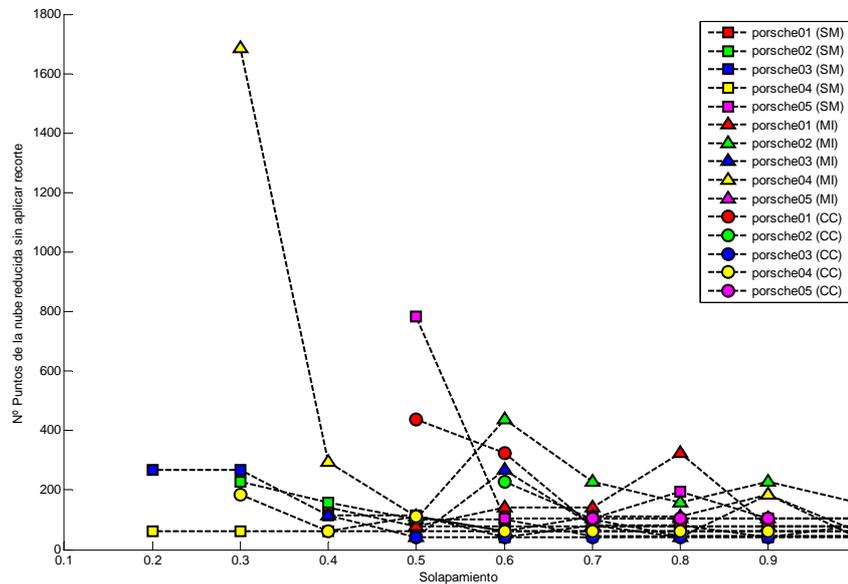


Figura 9.6: Comportamiento del algoritmo de alineamiento con tres medidas de similitud y cinco imágenes de rango del objeto 'porsche': la medida propuesta (SM), información mutua (MI) y coeficiente de correlación cruzada (CC). Se varía el grado de solapamiento y se incrementa el número de puntos de la nube (antes del 'recorte') hasta que el error de rotación sea inferior a 5 grados y el de traslación menor de 5 mm.

9.4. Alineamiento de nubes de puntos

El algoritmo de alineamiento 3D presentado en el Capítulo 7 se probó con tres conjuntos de imágenes de rango de diferente procedencia: bases de datos públicas, un escáner láser y una cámara de tiempo de vuelo.

Con el primer conjunto de datos (que comprende tanto imágenes de rango sintéticas como reales) se pretende obtener resultados de alineamiento que permitan una comparación cualitativa y cuantitativa del algoritmo propuesto con otros métodos del estado del arte. En cambio, los otros dos conjuntos de datos mostrarán cómo el algoritmo es capaz de alinear nubes de puntos procedentes de dispositivos reales.

9.4.1. Nubes de puntos procedentes de bases de datos

Con el fin de comparar el algoritmo de alineamiento propuesto con algunos de los ya existentes, utilizaremos imágenes de rango de objetos que han sido empleados por algunos autores para evaluar sus métodos así como en diversos estudios comparativos [89, 117, 88, 78]).

Los datos utilizados corresponden a nubes parciales de diferentes objetos procedentes de la base de datos de imágenes de rango de la Universidad de Stuttgart [56].

Se van a elegir pares de imágenes de rango para objetos de diferente topología. En las Figuras 9.7, 9.8, 9.9, 9.10 y 9.11, se muestran los resultados obtenidos para los objetos 'bunny', 'hip', 'dragon', 'porsche' y 'frog'. Además de las nubes de puntos alineadas aproximadamente se muestran las imágenes CIRCON de la correspondencia que obtuvo el mayor valor de similitud. Los errores que aparecen en las notas al pie de las figuras corresponden a los errores de rotación y traslación de la transformación aproximada con respecto a una transformación euclídea obtenida de manera supervisada con la adaptación del algoritmo ICP (ya que éste proporciona un alineamiento fino de las dos nubes de puntos).

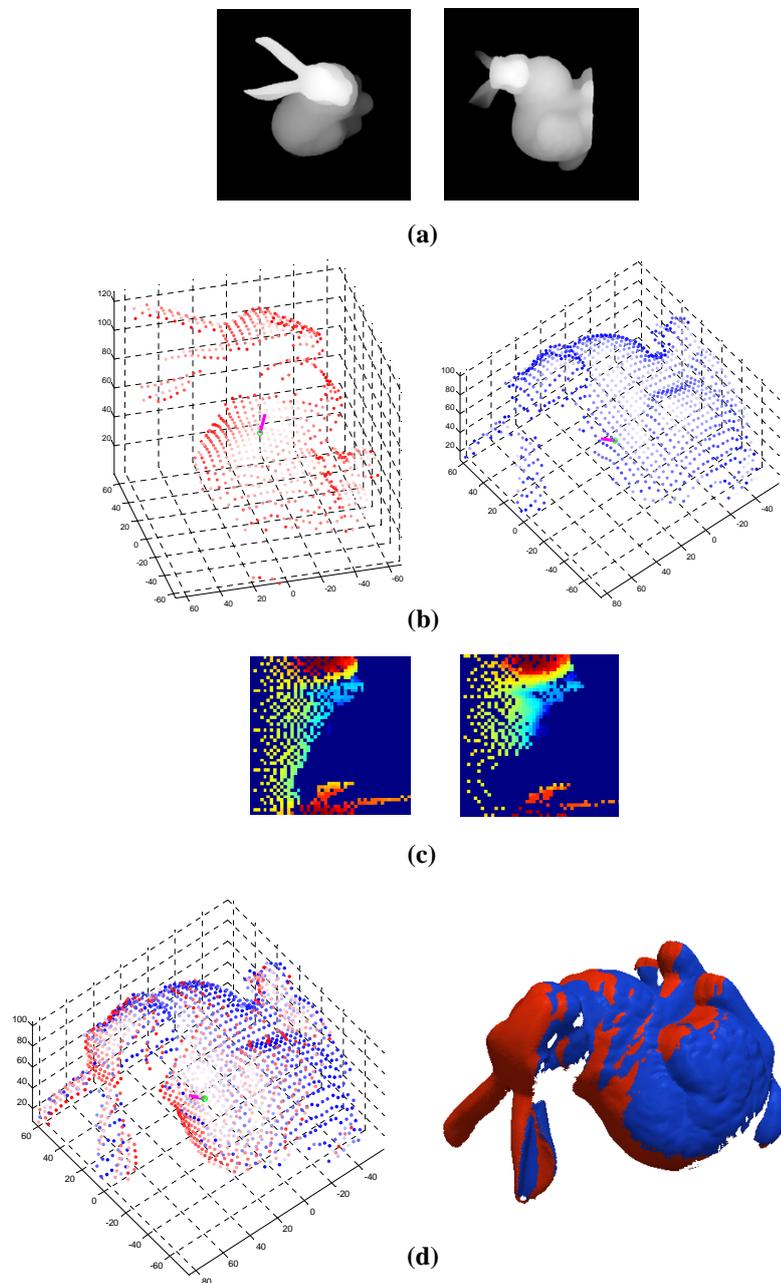


Figura 9.7: Alineamiento de dos imágenes de rango del objeto 'bunny'. (a) Imágenes de rango. (b) Puntos correspondientes (en magenta) que obtuvieron la máxima similitud. (c) Imágenes CIRCON asociadas a esos puntos. (d) Alineamiento obtenido: nube reducida (izquierda) y representación 3D con las nubes de puntos originales (derecha). Error de rotación: 1.6514° . Error de traslación: 1.2208 mm.

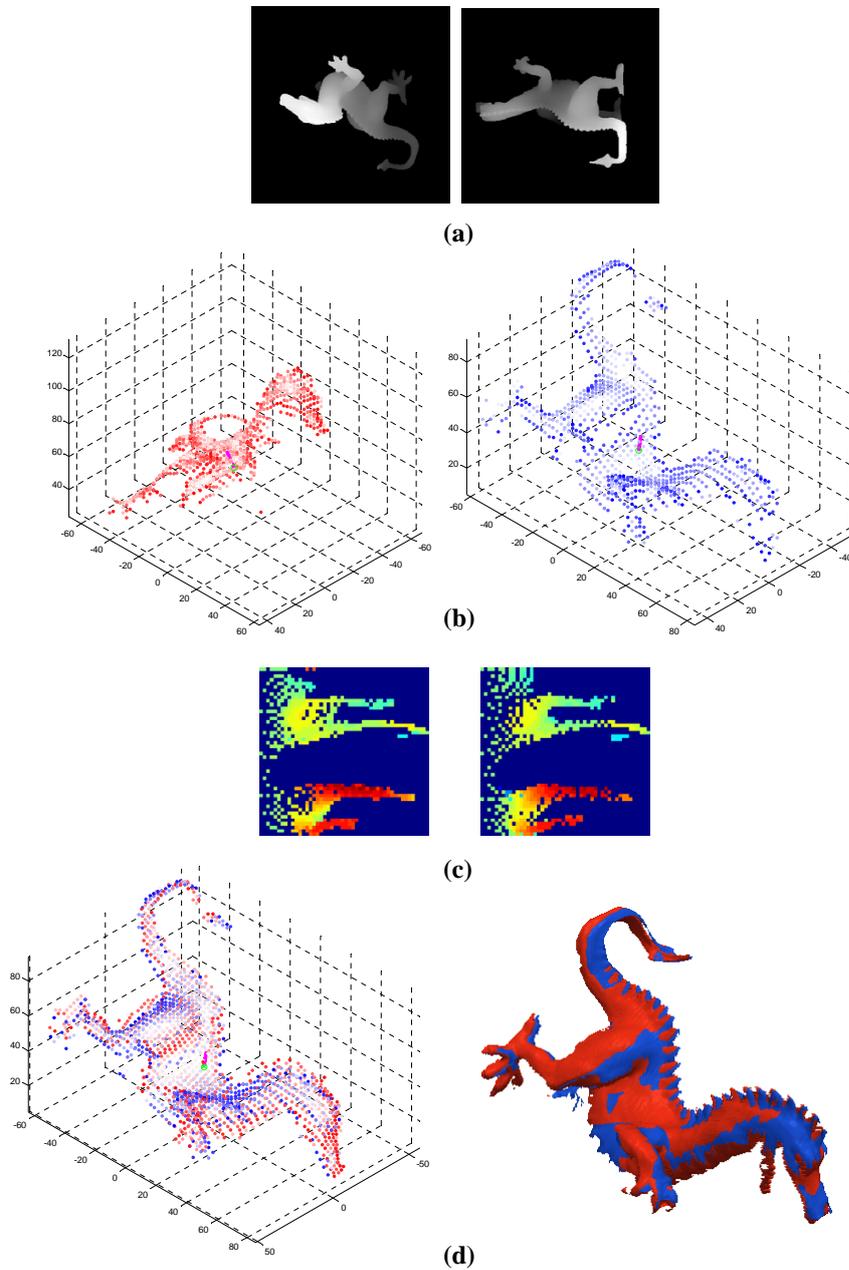


Figura 9.8: Alineamiento de dos imágenes de rango del objeto 'dragon'. (a) Imágenes de rango. (b) Puntos correspondientes (en magenta) que obtuvieron la máxima similitud. (c) Imágenes CIRCONE asociadas a esos puntos. (d) Alineamiento obtenido: nube reducida (izquierda) y representación 3D con las nubes de puntos originales (derecha). Error de rotación: 0.9213° . Error de traslación: 1.0734 mm.

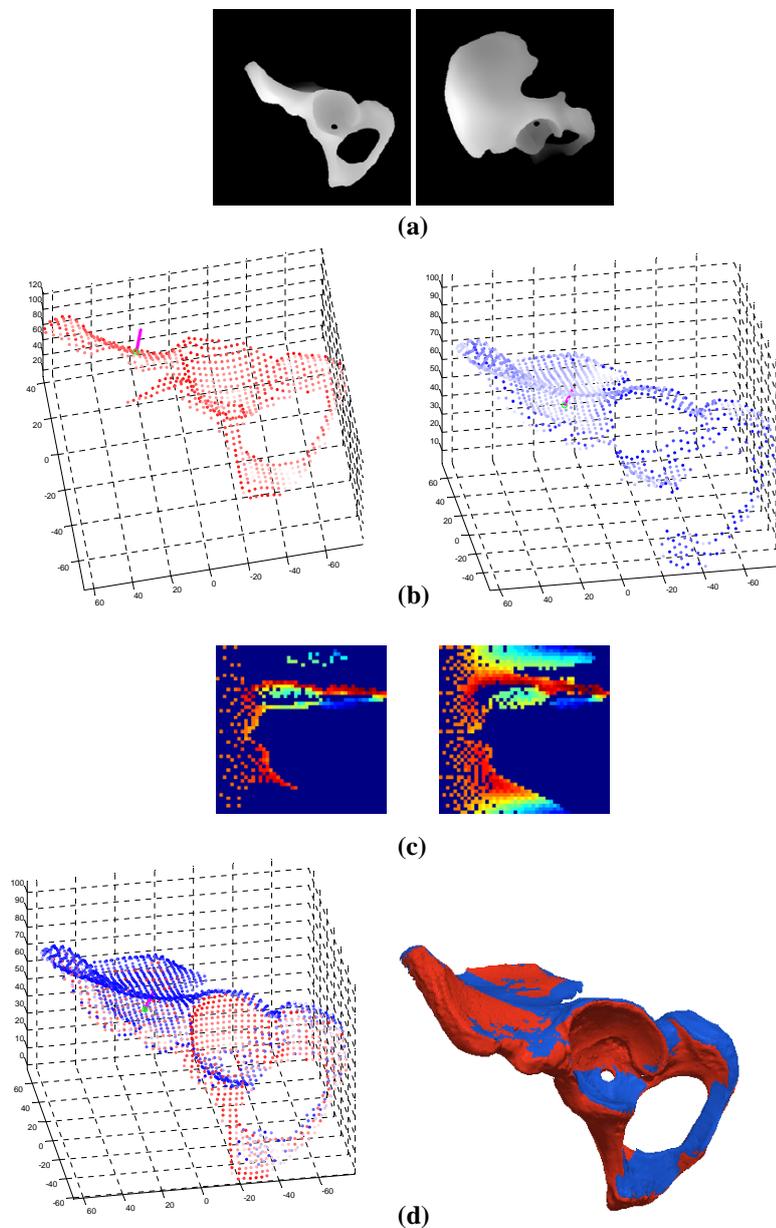


Figura 9.9: Alineamiento de dos imágenes de rango del objeto 'hip'. (a) Imágenes de rango. (b) Puntos correspondientes (en magenta) que obtuvieron la máxima similitud. (c) Imágenes CIRCÓN asociadas a esos puntos. (d) Alineamiento obtenido: nube reducida (izquierda) y representación 3D con las nubes de puntos originales (derecha). Error de rotación: 1.1909° . Error de traslación: 0.7436 mm.

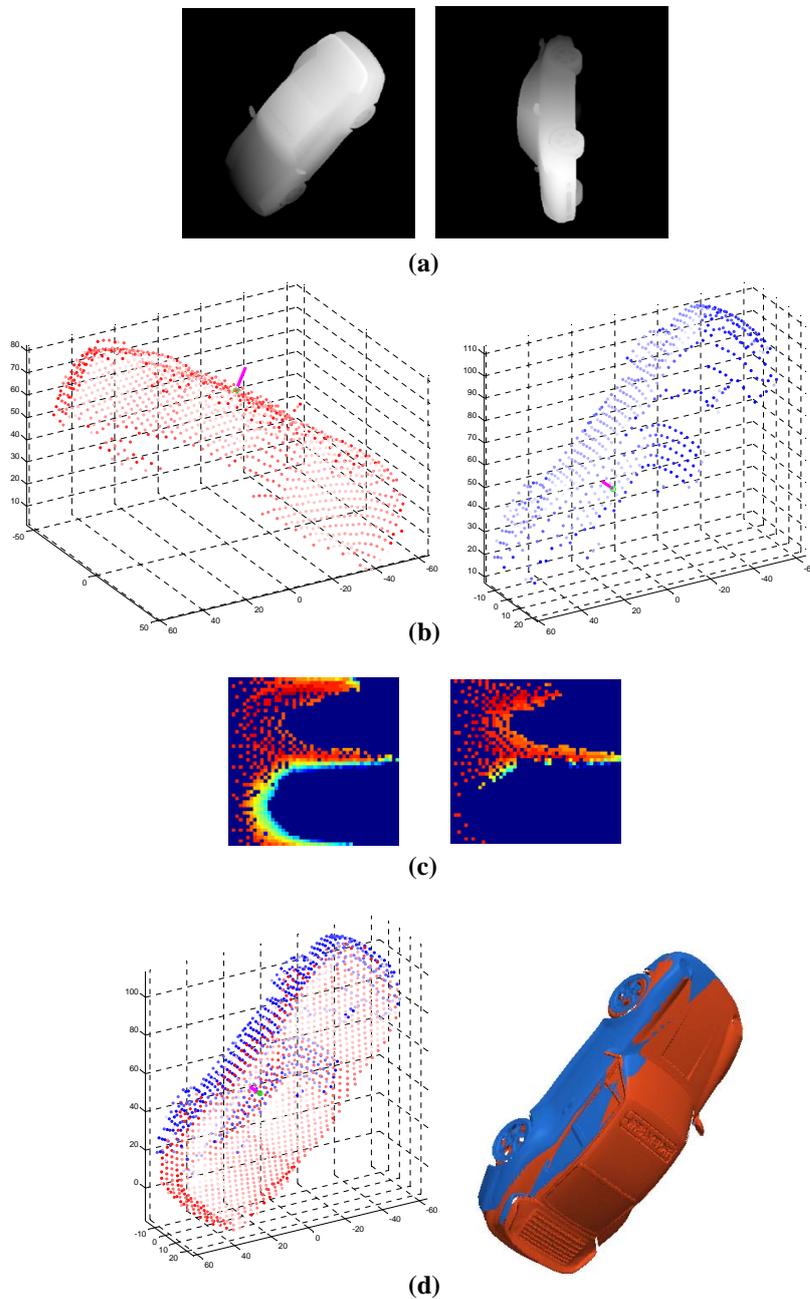


Figura 9.10: Alineamiento de dos imágenes de rango del objeto 'porsche'. (a) Imágenes de rango. (b) Puntos correspondientes (en magenta) que obtuvieron la máxima similitud. (c) Imágenes CIRCON asociadas a esos puntos. (d) Alineamiento obtenido: nube reducida (izquierda) y representación 3D con las nubes de puntos originales (derecha). Error de rotación: 2.7810° . Error de traslación: 1.8755 mm.

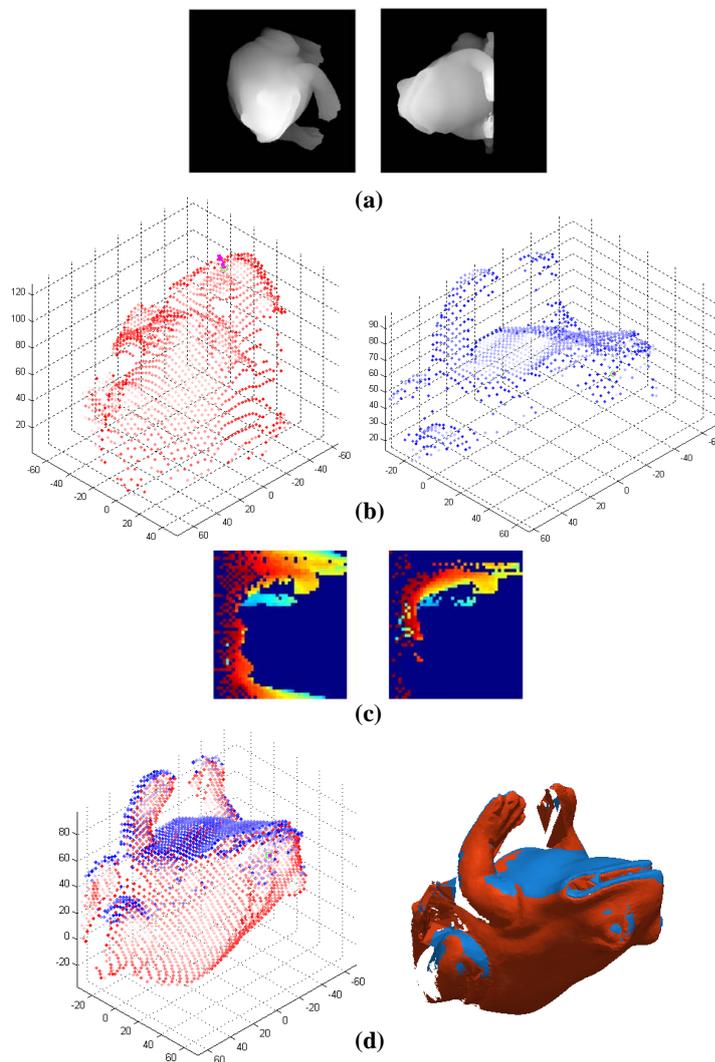


Figura 9.11: Alineamiento de dos imágenes de rango del objeto 'frog'. (a) Imágenes de rango. (b) Puntos correspondientes (en magenta) que obtuvieron la máxima similitud. (c) Imágenes CIRCON asociadas a esos puntos. (d) Alineamiento obtenido: nube reducida (izquierda) y representación 3D con las nubes de puntos originales (derecha). Error de rotación: 1.5563° . Error de traslación: 0.69191 mm.

Además de los ejemplos mostrados en las Figuras 9.7, 9.8, 9.9, 9.10 y 9.11, en la Tabla 9.1 se recogen los resultados obtenidos con el algoritmo de alineamiento para más objetos de diferente topología (ver Figura 9.12).

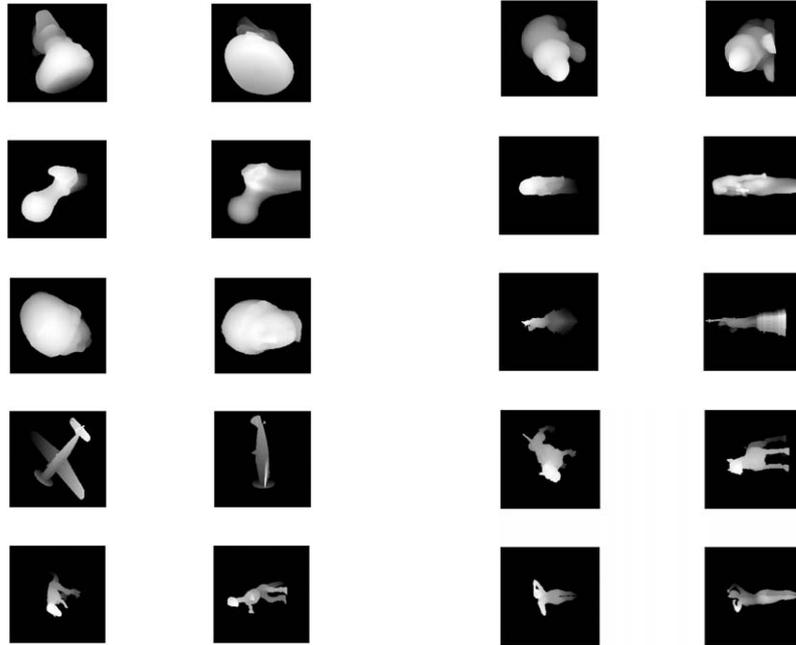


Figura 9.12: Imágenes de rango de diez objetos de la base de datos de la Universidad de Stuttgart: 'ducky', 'femur', 'igea', 'fighter', 'dino', 'mole', 'isis', 'liberty', 'pitbull', 'woman'.

	Error Rot. (grados)	Error Trasl. (mm)	Nº Puntos Nube reducida 1 (% Total ptos.)	Nº Puntos Nube Reducida 2 (% Total ptos.)
Ducky	2,9921	1,1464	859 (1,61%)	1142 (1,90%)
Femur	3,7737	1,9364	453 (1,54%)	497 (1,25%)
Igea	2,4891	1,8897	544 (0,93%)	419 (0,73%)
Fighter	1,5039	0,4712	635 (2,13%)	401 (2,67%)
Dino	4,0881	0,9955	528 (4,92%)	314 (2,45%)
Mole	2,3435	1,0631	673 (1,45%)	652 (1,65%)
Isis	3,6157	2,0581	543 (3,50%)	381 (1,47%)
Liberty	4,8414	0,6614	715 (5,27%)	428 (2,47%)
Pitbull	2,3053	0,8867	153 (0,83%)	108 (0,58%)
Woman	3,5380	0,6183	549 (5,58%)	244(1,67%)

Tabla 9.1: Errores cometidos en la transformación obtenida por el Algoritmo 7.1 para 10 objetos diferentes. Números de puntos de las nubes reducidas utilizadas por el algoritmo y porcentaje sobre el total de puntos de las nubes originales.

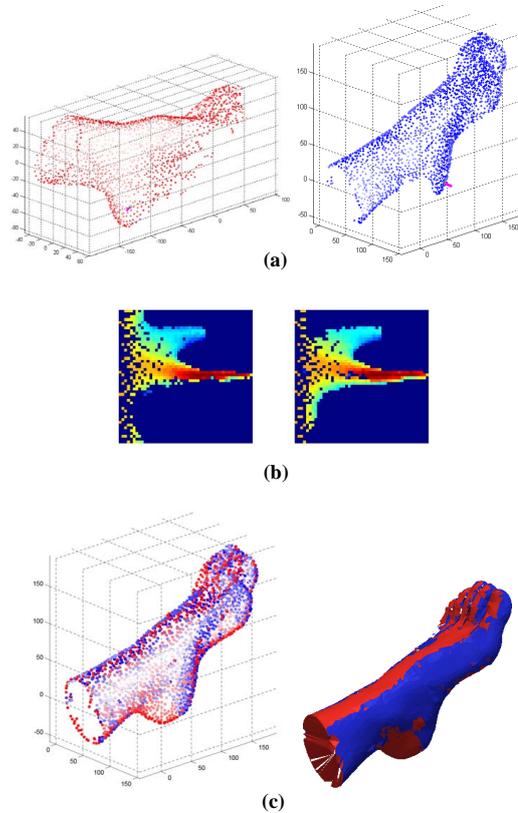


Figure 9.13: Alineamiento obtenido por el algoritmo para el objeto 'foot'. (a) Puntos correspondientes (vector normal en magenta) que obtuvieron el máximo valor de similitud. (b) Imágenes CIRCONE asociadas a esos puntos. (c) Alineamiento: Nube reducida (izquierda) y representación 3D utilizando las nubes de puntos originales (derecha). Error de rotación: 0.9136° . Error de traslación: 0.5224 mm.

También mostramos los resultados obtenidos con los cinco objetos (Figuras 9.13 y 9.14 y Tabla 9.2) que se utilizaron en [89] para comparar los resultados obtenidos con cuatro algoritmos de alineamiento aproximado: Spin images, RANSAC based-DARCES, análisis de componentes principales (PCA) y el algoritmo genético propuesto por Brännstrom. Las nubes de puntos correspondientes a esos objetos de test se encuentran a disposición pública en la página web del autor [74].

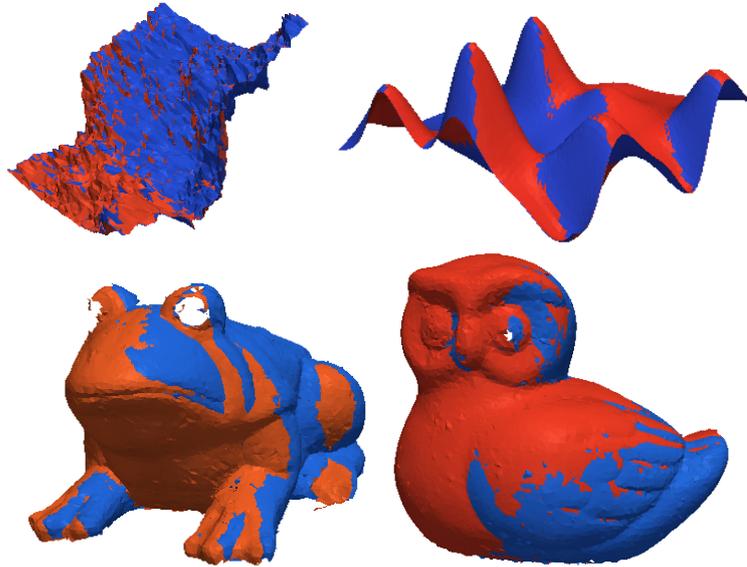


Figura 9.14: Resultados obtenidos para cuatro objetos del estudio comparativo de Salvi-Matabosch [89]: 'fractal', 'wave', 'frog' y 'owl'. Los errores cometidos se muestran en la Tabla 9.2.

	Error Rot. (grados)	Error Trasl. (mm)
Foot	0,9136	0,5224
Fractal	0,0516	0,0001
Waves	0,4619	0,0019
Frog	1,2714	0,7502
Owl	1,3312	0,2547

Tabla 9.2: Errores de rotación y traslación para los alineamientos de las Figuras 9.13 y 9.14.

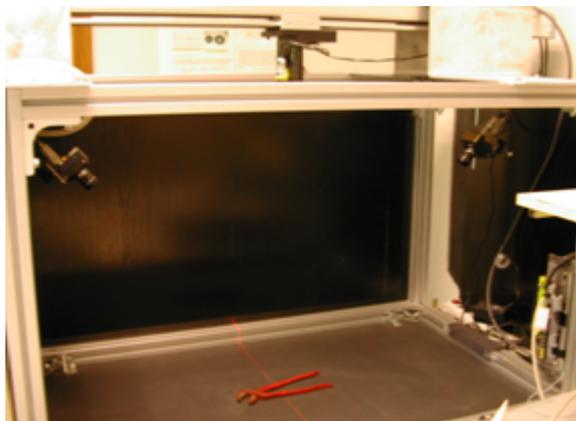


Figure 9.15: Escáner láser utilizado para adquirir las imágenes de rango.

9.4.2. Nubes de puntos de escáneres láser

El algoritmo de alineamiento se probó con nubes de puntos tomadas mediante el escáner de triangulación láser mostrado en la Figura 9.15. Este sistema [106] consta de dos cámaras CCD, un proyector láser y una guía lineal, controlada por computador, que facilita el desplazamiento de la línea láser sobre la escena. La adquisición de las imágenes y el movimiento de la guía están sincronizados mediante una señal de comunicación entre el *frame-grabber* de las cámaras y el controlador de la guía.

Este sistema combina triangulación estéreo con triangulación activa para obtener una reconstrucción más completa y precisa de los objetos [107]. Sin embargo, es inevitable que aparezcan oclusiones debido a la forma en la que el láser se proyecta sobre los objetos y a la propia topología de éstos (p.e. debido a concavidades). Por esta razón, sigue siendo necesario realizar varios escaneos del objeto desde puntos de vista diferentes para obtener una información tridimensional más completa.

La Figura 9.16 muestra un ejemplo del alineamiento obtenido para dos imágenes de rango de un colector de escape adquiridas con este escáner láser. Como puede observarse se han tomado desde dos puntos de vista diferentes, de modo que algunas partes de las nubes de puntos solamente aparecen en una de las dos nubes de puntos. El algoritmo de alineamiento es capaz de encontrar dos descriptores CIRCON similares que permiten obtener un alineamiento aproxi-

mado de las dos nubes de puntos con pequeños errores de rotación y traslación respecto al alineamiento fino obtenido con la adaptación del algoritmo ICP.

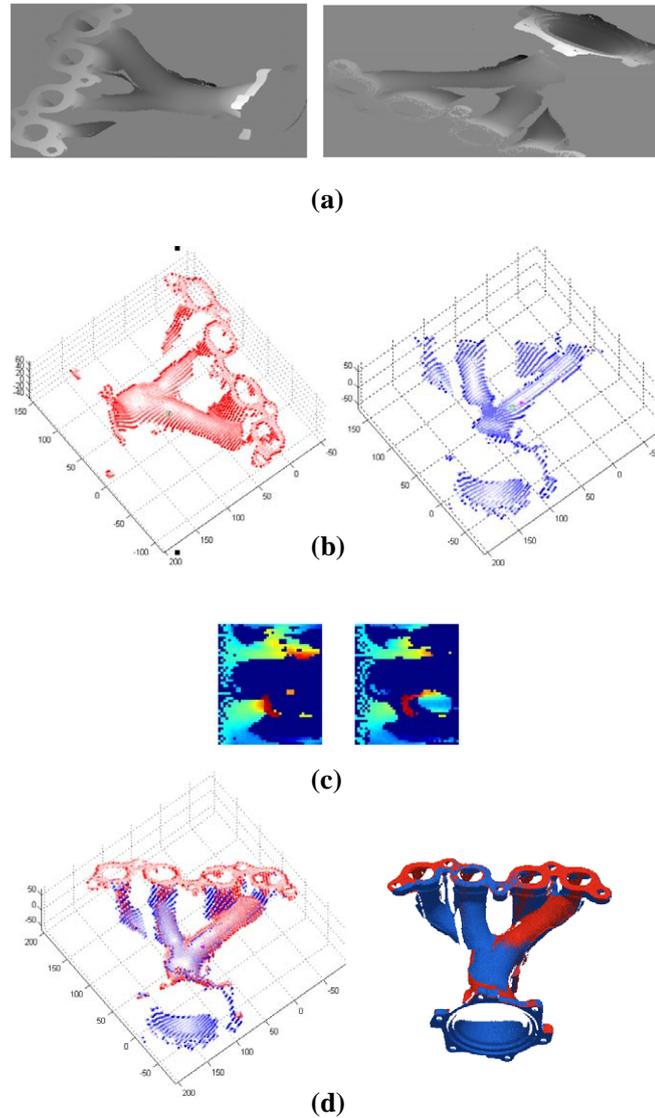


Figure 9.16: Alineamiento de dos imágenes de rango de un colector de escape adquiridas mediante un escáner láser. **(b)** Puntos correspondientes (vector normal en magenta) que obtuvieron el máximo valor de similitud. **(c)** Imágenes CIRCON asociadas con esos puntos. **(d)** Alineamiento: nube reducida (izquierda) y representación 3D usando las nubes de puntos originales (derecha). Error de rotación: 1.7427° . Error de traslación: 0.7638 mm.

En la Figura 9.17 se muestra otro ejemplo de alineamiento con imágenes del escáner láser de la Figura 9.15. A pesar de que el objeto presenta simetría axial, el algoritmo es capaz de alinear correctamente sus cinco taladros.

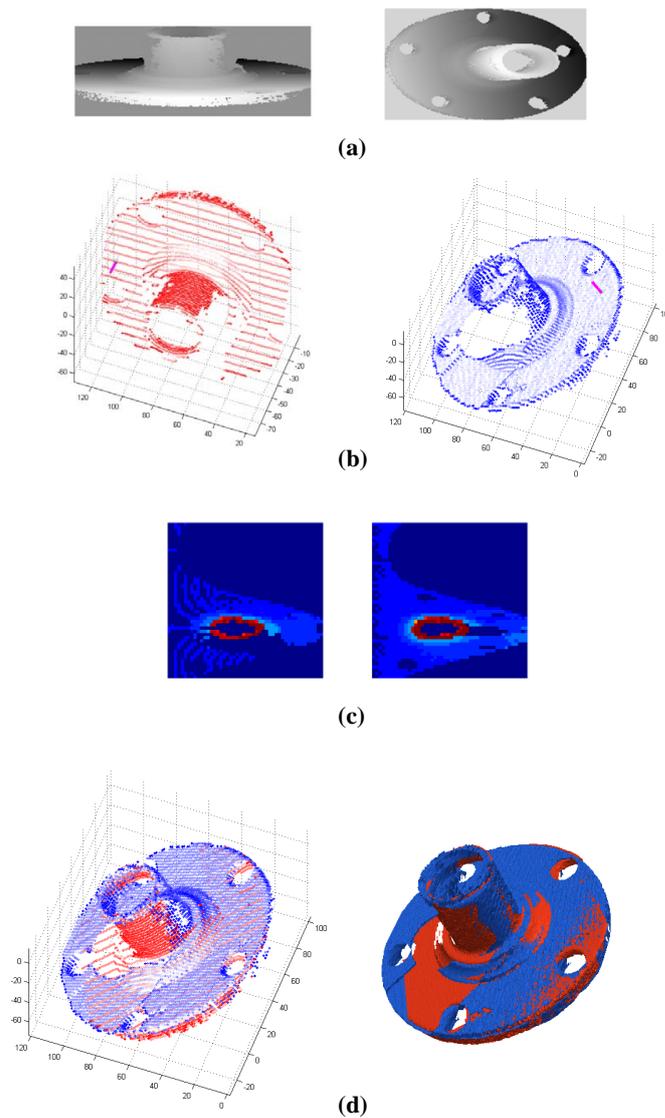


Figure 9.17: Alineamiento de dos imágenes de rango de un cubo de rueda adquiridas mediante un escáner láser. (b) Puntos correspondientes (vector normal en magenta) que obtuvieron el máximo valor de similitud. (c) Imágenes CIRC-ON asociadas con esos puntos. (d) Alineamiento: nube reducida (izquierda) y representación 3D usando las nubes de puntos originales (derecha). Error de rotación: 1.0691° . Error de traslación: 0.8114 mm.

9.4.3. Nubes de puntos de cámaras de tiempo de vuelo

Otra forma de obtener información tridimensional es mediante cámaras de tiempo de vuelo. En este caso se utilizó el modelo SR4000 de MESA Imaging AG, que proporciona imágenes de rango de 176×144 píxeles casi en tiempo real. Esta cámara tiene un rango de medida que va desde 0.8m hasta 5m con una precisión absoluta de $\pm 10\text{mm}$, lo que puede tener como consecuencia una variación sustancial de la posición de los puntos de la superficie de los objetos. Por consiguiente se deberían utilizar objetos con unas dimensiones que superen los 20cm de largo, para que la carencia de resolución y precisión de la cámara se vean compensadas por un número suficiente de puntos de superficie.



Figura 9.18: Cámara de tiempo de vuelo empleada en los experimentos.

Las Figuras 9.19 y 9.20 muestran los resultados obtenidos para dos imágenes de rango de un colector de escape adquiridas con la cámara de tiempo de vuelo. A pesar de la alta dispersión y la baja densidad que tienen ambas nubes de puntos y que, como se puede observar en las imágenes de rango, éstas no poseen una gran superficie común, el algoritmo es capaz de encontrar dos descriptores CIRCON similares y alinear correctamente ambas nubes de puntos. Otro ejemplo para dos imágenes de rango de un cubo de rueda se muestra en las Figuras 9.21 y 9.22.

Estos resultados vuelven a mostrar (como en la Sección 9.3) que el algoritmo no necesita una gran precisión ni densidad de las nubes de puntos, por lo que podrían utilizarse dispositivos rápidos de adquisición 3D, aunque sean menos precisos, para operaciones de captura robotizada de objetos, incluso aunque el modelo utilizado para la comparación tampoco fuese demasiado preciso. De esta manera, las cámaras de tiempo de vuelo podrían utilizarse para obtener un modelo 3D aproximado del objeto tomando imágenes desde diferentes puntos

de vista sin tener que utilizar un sistema de posicionado, es decir, tan sólo llevando la cámara en mano. Dado que la cámara de tiempo de vuelo otorga un valor de "confianza" a cada punto reconstruido, esta información podría utilizarse para refinar la forma final del modelo 3D.

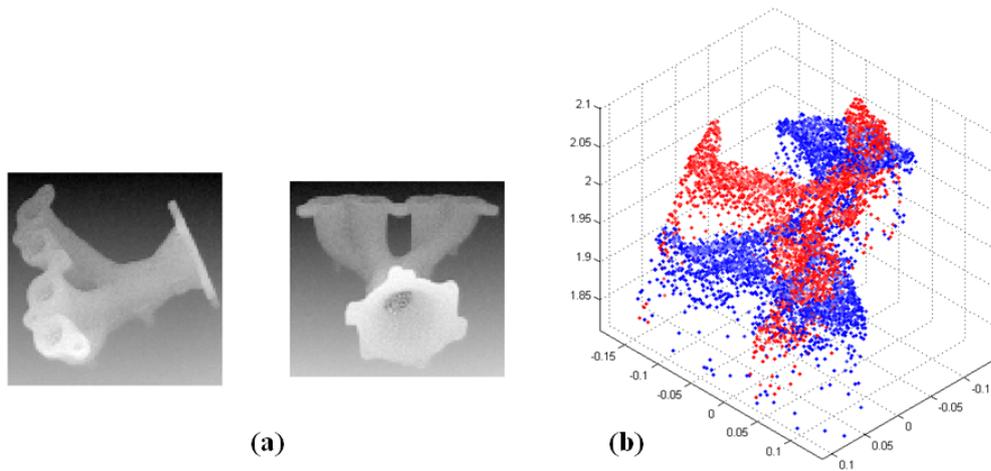


Figura 9.19: (a) Imágenes de rango de un colector de escape adquiridas por la cámara TOF mostrada en la Figura 9.18. (b) Desalineamiento inicial entre las dos nubes de puntos correspondientes.

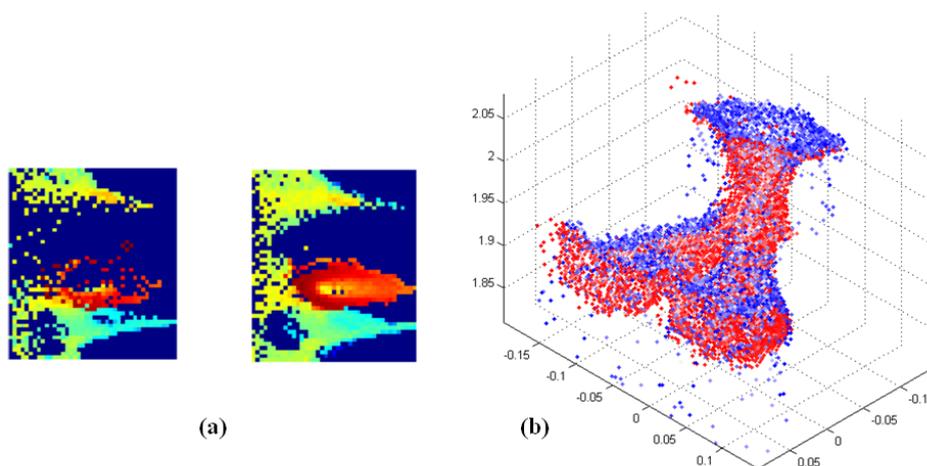


Figura 9.20: (a) Imágenes CIRCON similares encontradas por el algoritmo para las nubes de puntos mostradas en la Figura 9.19. (b) Alineamiento aproximado obtenido por el algoritmo.

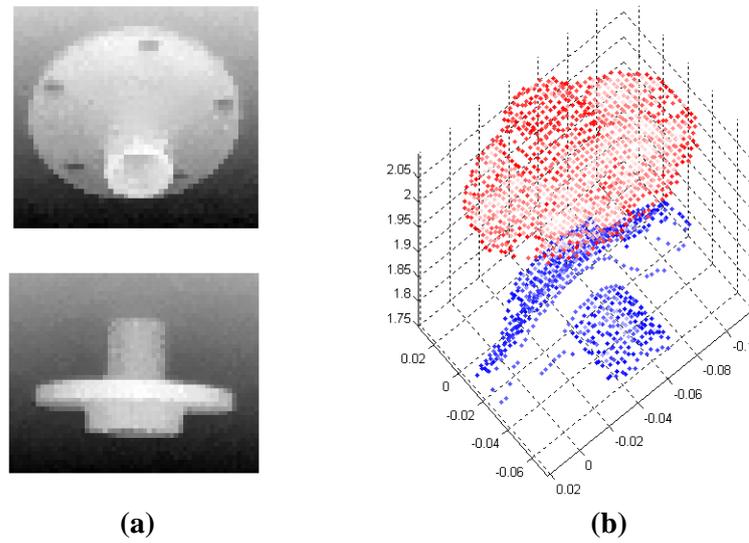


Figura 9.21: (a) Imágenes de rango de un cubo de rueda adquiridas por la cámara TOF mostrada en la Figura 9.18. (b) Desalineamiento inicial entre las dos nubes de puntos correspondientes.

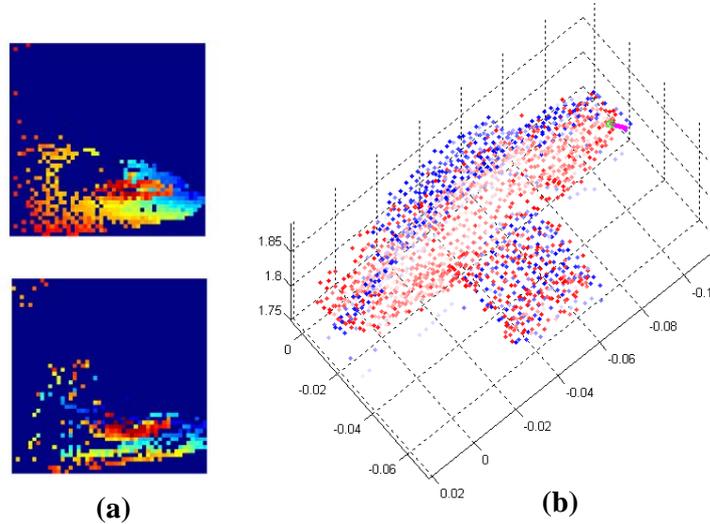


Figura 9.22: (a) Imágenes CIRCON similares encontradas por el algoritmo para las nubes de puntos mostradas en la Figura 9.21. (b) Alineamiento aproximado obtenido por el algoritmo.

9.5. Alineamiento 3D de nubes de puntos de dispositivos diferentes

También se ha probado el algoritmo para alinear imágenes de rango obtenidas de dispositivos diferentes: la cámara TOF de la Figura 9.18 y el escáner láser de la Figura 9.15. A pesar de la baja densidad y la escasa precisión de la nube de puntos de la cámara de tiempo de vuelo (ver Figura 9.23) se obtiene un alineamiento bastante aceptable, ya que el algoritmo es capaz de encontrar una correspondencia con imágenes CIRCON similares. En este caso se ha elegido como modelo la nube de puntos del escáner láser por ser más densa y precisa.

9.6. Alineamiento 3D en escenas con mezcla de objetos

Las Figuras 9.24 y 9.25 muestran cómo el algoritmo de alineamiento aproximado puede usarse también para localizar objetos en una escena en la que éstos se encuentren mezclados. Como en este caso la transformación obtenida se aplica a la nube de puntos de la escena para poder alinearla con el modelo, será la transformación inversa la que realmente proporciona la posición y orientación del objeto en dicha escena. Esta información podría utilizarse directamente para capturar el objeto si tanto el escáner láser como el manipulador robótico estuviesen referenciados a un sistema de coordenadas común [36] y la posición y orientación del modelo con respecto a este s.d.c. fuese conocida.

Notar que la única diferencia en comparación con los resultados obtenidos en las secciones anteriores es que el radio del entorno del punto de interés se ha limitado a un valor acorde a los objetos que se quieren localizar en la escena (en estos ejemplos, este límite se fijó en 120 mm).

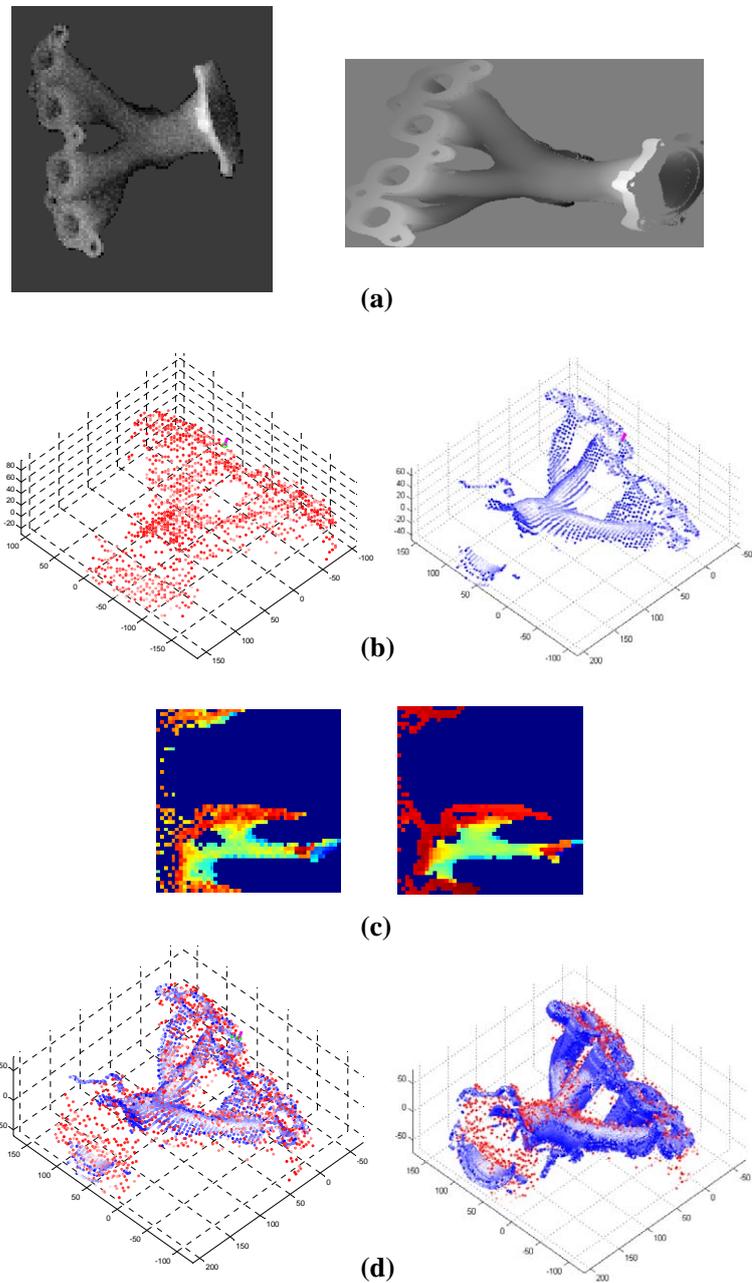


Figure 9.23: Alineamiento 3D de imágenes de rango de diferentes dispositivos: una procedente de una cámara TOF (izquierda) y otra de un escáner láser de triangulación (derecha). **(a)** Imágenes de rango. **(b)** Puntos correspondientes (vector normal en magenta) que obtuvieron el máximo valor de similitud. **(c)** Imágenes CIRCON asociadas a esos puntos. **(d)** Alineamiento: nube reducida (izquierda) y representación 3D con las nubes de puntos originales (derecha).

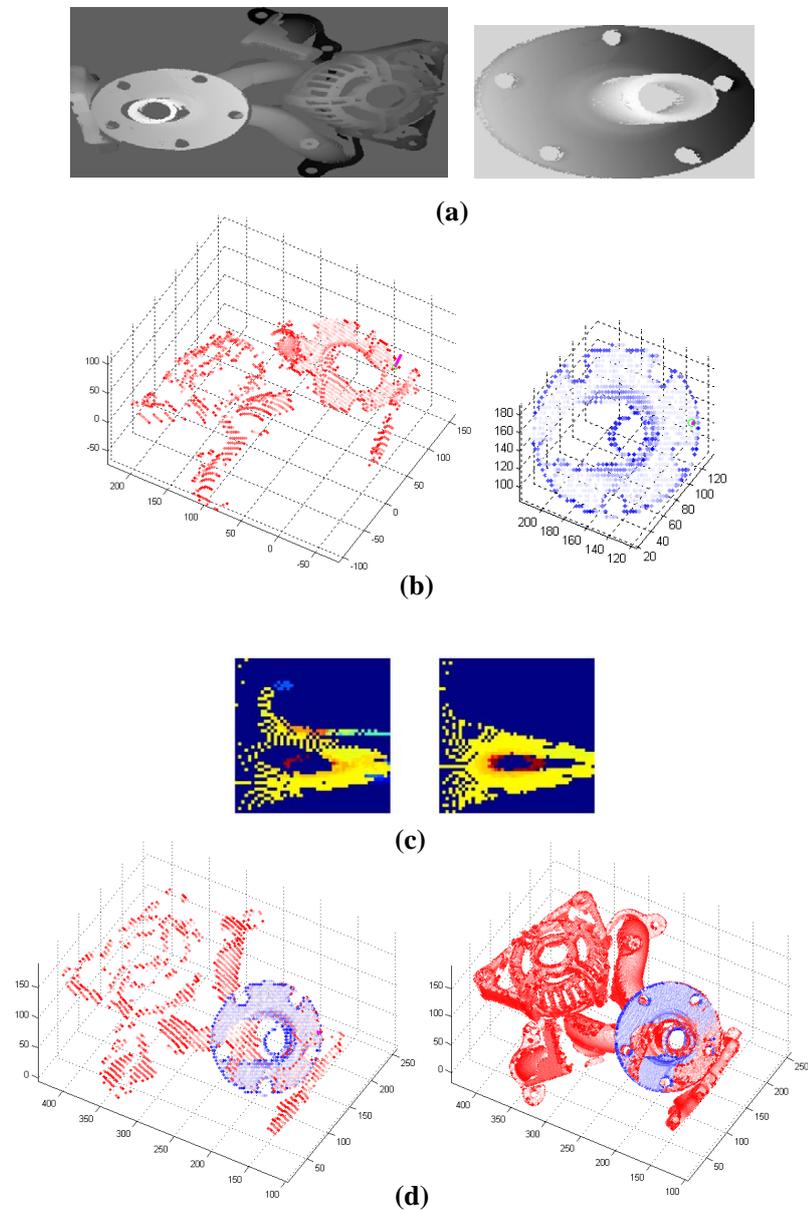


Figure 9.24: Alineamiento 3D en escenas con mezcla de objetos (el radio del entorno del punto de interés es igual a 120 mm). **(a)** Imágenes de rango: escena (izquierda) e imagen de rango del modelo "cubo de rueda" (derecha). **(b)** Puntos correspondientes (vector normal vector en magenta) que obtuvieron el máximo valor de similitud. **(c)** Imágenes CIRCONE asociadas con esos puntos. **(d)** Alineamiento: nube reducida (izquierda) y representación 3D utilizando las nubes originales (derecha).

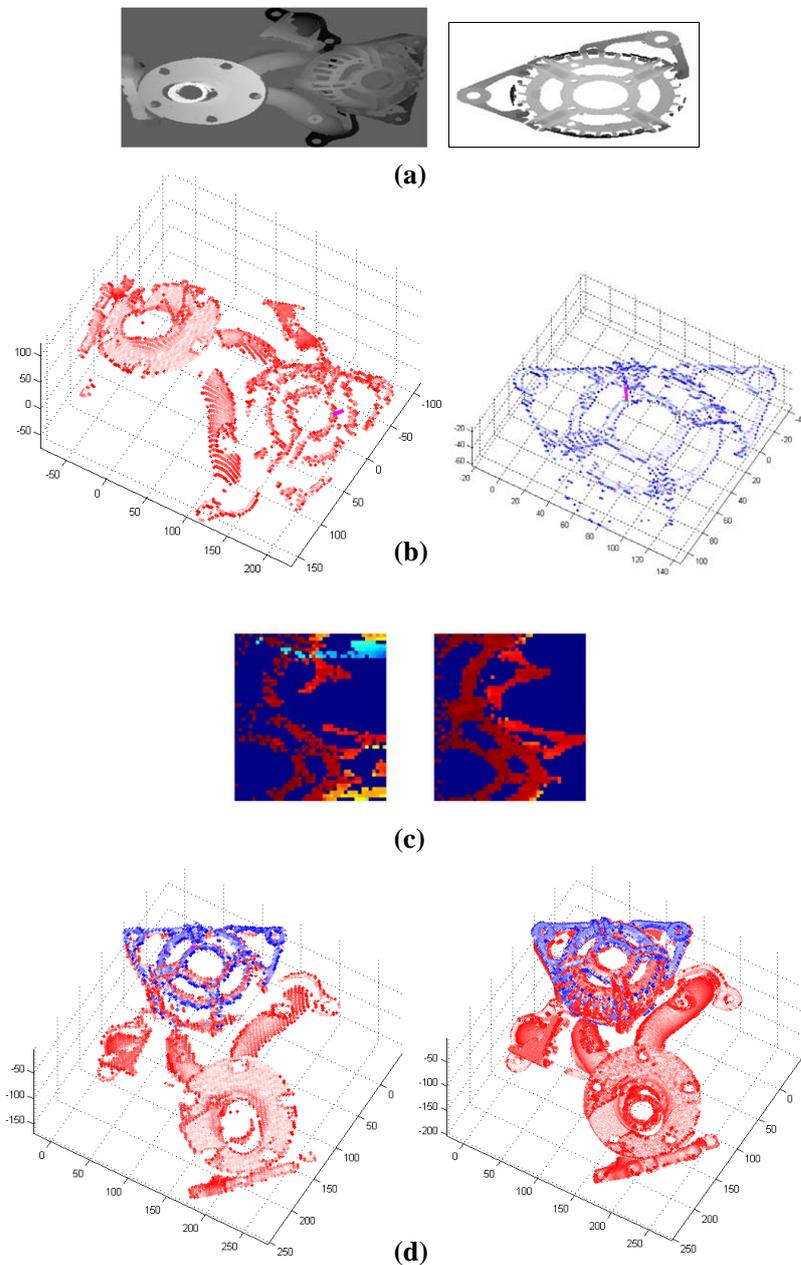


Figure 9.25: Alineamiento 3D en escenas con mezcla de objetos (el radio del entorno del punto de interés es igual a 120 mm). **(a)** Imágenes de rango: escena (izquierda) e imagen de rango del modelo "carcasa de alternador" (derecha). **(b)** Puntos correspondientes (vector normal vector en magenta) que obtuvieron el máximo valor de similitud. **(c)** Imágenes CIRCON asociadas con esos puntos. **(d)** Alineamiento: nube reducida (izquierda) y representación 3D utilizando las nubes originales (derecha).

Capítulo 10

Algoritmo de Alineamiento 3D basado en Descomposición Wavelet

10.1. Introducción

En este capítulo describimos un algoritmo para alineamiento 3D [105] que utiliza una estrategia similar a la presentada en el Capítulo 7; sin embargo, en este caso, los descriptores se generan a partir de la forma 3D simplificada (ver Sección 3.5.1) y la búsqueda de correspondencias se basa en una descomposición wavelet de las imágenes CIRCON. Esto permite acelerar la búsqueda mediante la reducción, para cada nivel de resolución, de la cantidad de descriptores que han de ser evaluados. Finalmente, la información obtenida por este proceso se utiliza para el cálculo de una transformación euclídea (traslación y rotación) que permite alinear de forma aproximada las dos nubes de puntos.

10.2. Búsqueda de Correspondencias

Dado que el objetivo de este proceso es obtener el mejor alineamiento entre dos nubes de puntos correspondiente a dos puntos de vista diferentes del mismo objeto, las imágenes CIRCON se van a generar utilizando todos los puntos de

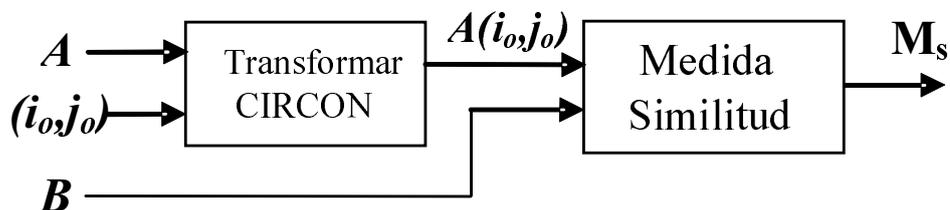


Figura 10.1: Diagrama de bloques básico del proceso de búsqueda de correspondencias.

ambas nubes.

En este caso se va a seguir una estrategia que simplifica el algoritmo de búsqueda explicado en el Capítulo 6. Este nuevo algoritmo parte de un esquema básico de comparación de correspondencias 3D que se presentó en [104] y que se muestra en la Figura 10.1. En este diagrama de bloques la principal entrada son dos únicos descriptores, A y B , correspondientes a dos puntos de interés ${}^{w_1}p_{1a}$ y ${}^{w_2}p_{2b}$. Entonces, utilizando la forma 3D simplificada que puede recuperarse a partir del descriptor (ver Sección 3.5.1), se pueden obtener nuevos descriptores. Tan sólo habrá que desplazar el sistema de coordenadas local a uno de los puntos definidos para los píxeles del descriptor A y aplicar las Ecuaciones 3.7, 3.8 y 3.9. Es decir, si escogemos un píxel de la matriz A , cuya fila es la i_0 y su columna es la j_0 , calculamos su punto 3D correspondiente y creamos su descriptor CIRCON asociado con el resto de puntos de la forma 3D simplificada. A continuación, compararemos este nuevo descriptor, que denotaremos como $A(i_0, j_0)$, con el descriptor B utilizando para ello una medida de similitud.

Para obtener una transformación aproximada que sea aceptable, la resolución de los descriptores debería ser relativamente alta, lo que obligaría a comprobar un elevado número de píxeles haciendo que el tiempo de computación pueda llegar a ser excesivo. Por lo tanto, el principal inconveniente de esta estrategia está en la dificultad para reducir la cantidad de índices (i_0, j_0) que tienen que ser evaluados.

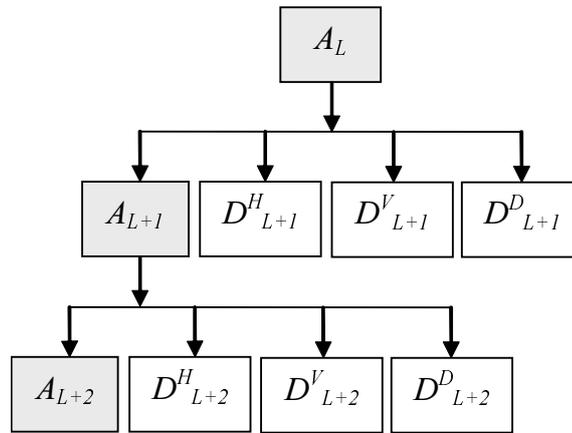


Figura 10.2: Descomposición de una imagen utilizando la Transformación Wavelet Discreta (DWT).

10.3. Descomposición Wavelet

El enfoque presentado en esta sección utiliza la descomposición wavelet con el fin de reducir progresivamente, "coarse-to-fine", la cantidad de índices que deben ser verificados.

Este método consiste básicamente en la descomposición de las dos imágenes CIRCON mediante la Transformación Wavelet Discreta (DWT) [71]. Para poder aplicar esta transformación a los descriptores CIRCON solamente habrá que efectuar un único cambio en su definición. En vez de colocar un valor 'NaN' en aquellos elementos de la matriz que no tengan asociados puntos de la superficie del objeto, se les asignará un valor igual al mínimo de la matriz menos una unidad.

La DWT es una transformación bidimensional que conduce a una descomposición de los coeficientes de aproximación para el nivel L en cuatro componentes: la aproximación a nivel $L + 1$, y los detalles en tres orientaciones (horizontal, vertical y diagonal). A su vez, la aproximación A_{L+1} se puede descomponer en otros cuatro componentes, como se muestra en la Figura 10.2.

La familia de wavelets que ha sido elegida para nuestros experimentos son las wavelets tipo *Haar* [77], ya que, debido a su morfología y a que se trata de descomponer imágenes sintéticas, produce los mejores resultados para el

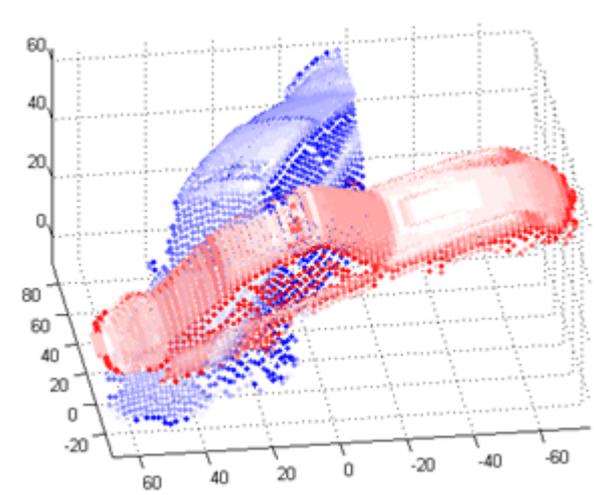


Figura 10.3: Situación inicial de las nubes de puntos.

algoritmo de alineamiento 3D que se explica en el párrafo siguiente.

10.4. Estrategia multinivel

Dadas dos nubes de puntos obtenidas desde puntos de vista diferentes de un mismo objeto, cuya situación inicial se muestra en la Figura 10.3, este algoritmo "coarse-to-fine" reduce, mediante un procesamiento multi-nivel, la cantidad de información que se evalúa.

En primer lugar, se obtienen los puntos más cercanos al centro de gravedad de las nubes de puntos, $w^1 p_{1a}$ y $w^2 p_{2b}$, y se generan sus descriptores CIRCON correspondientes A y B (ver Figura 10.4). A continuación, las imágenes asociadas a esos dos descriptores *básicos* se descomponen en N niveles y se toman las N aproximaciones A_L y B_L que serán procesadas (ver Figura 10.5). Las vistas en planta de sus formas 3D correspondientes se muestran en la Figura 10.6 (notar que la recuperación de la forma 3D es una de las propiedades de las imágenes CIRCON).

A continuación, se evalúan las correspondencias para las imágenes del nivel N usando el diagrama de bloques mostrado en la Figura 10.1 para cada píxel de la imagen A_N , que pertenece al objeto representado. Los valores de simili-

tud resultantes permitirán descartar las correspondencias (y por lo tanto, sus transformaciones euclídeas asociadas) que no alcancen un mínimo de similitud.

A su vez, la misma evaluación se repetirá para las imágenes del nivel $N-1$, pero sólo será aplicado a los índices (i_0, j_0) que correspondan a los píxeles que no han sido eliminados en el paso anterior. De tal manera que, en cada paso, las correspondencias, se reducen progresivamente (ver Subsección 10.5.3). Finalmente, el pequeño grupo de correspondencias que se ha obtenido para el último nivel será verificado para determinar cuál de ellas produce el mejor alineamiento entre las dos nubes de puntos (ver Subsección 10.5.4).

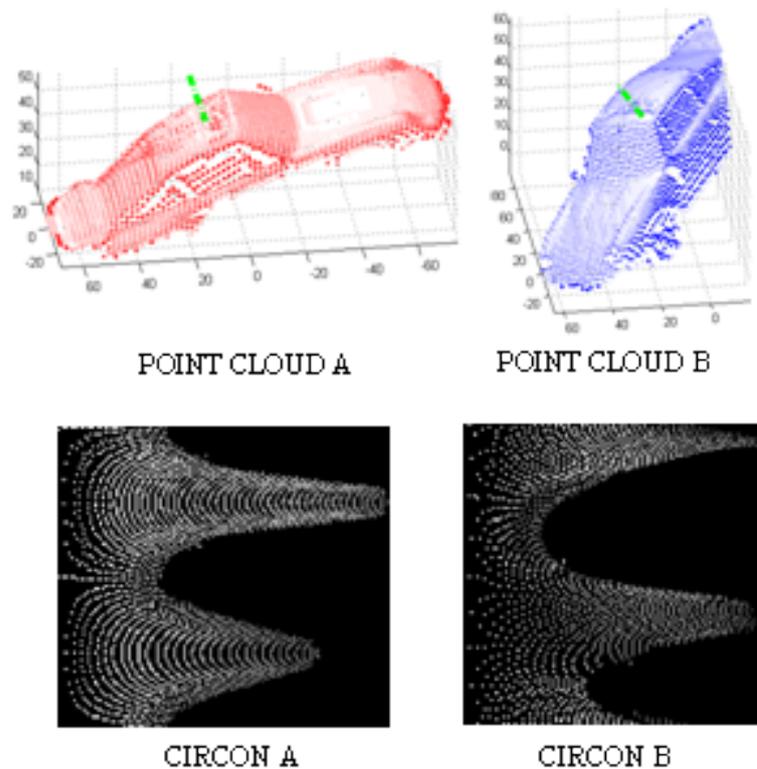


Figura 10.4: Imágenes CIRCON correspondientes a las dos nubes de puntos del mismo objeto mostradas en la parte superior de la figura.

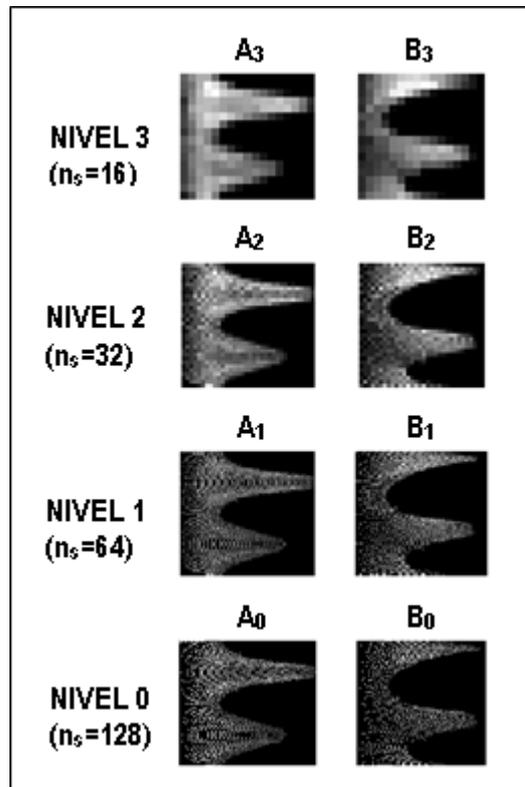


Figura 10.5: Aproximaciones Wavelet para tres niveles de resolución de las dos imágenes CIRCON iniciales (A_0 y B_0).

10.5. Detalles del algoritmo

10.5.1. Transformación de un Descriptor CIRCON

Con el fin de encontrar correspondencias entre las nubes de puntos, se realizarán transformaciones sobre una de las dos imágenes CIRCON correspondientes (A y B). Estas transformaciones modifican la imagen CIRCON original, produciendo una nueva que será comparada con la imagen B .

Supongamos que, como se muestra en la Figura 10.1, la imagen CIRCON A es a la que se aplican las transformaciones. Dados dos índices (i_0, j_0) , esta transformación modifica todos los elementos de la matriz y sus posiciones originales, centrando la imagen CIRCON en el punto 3D correspondiente al píxel con índice de fila i_0 e índice de columna j_0 .

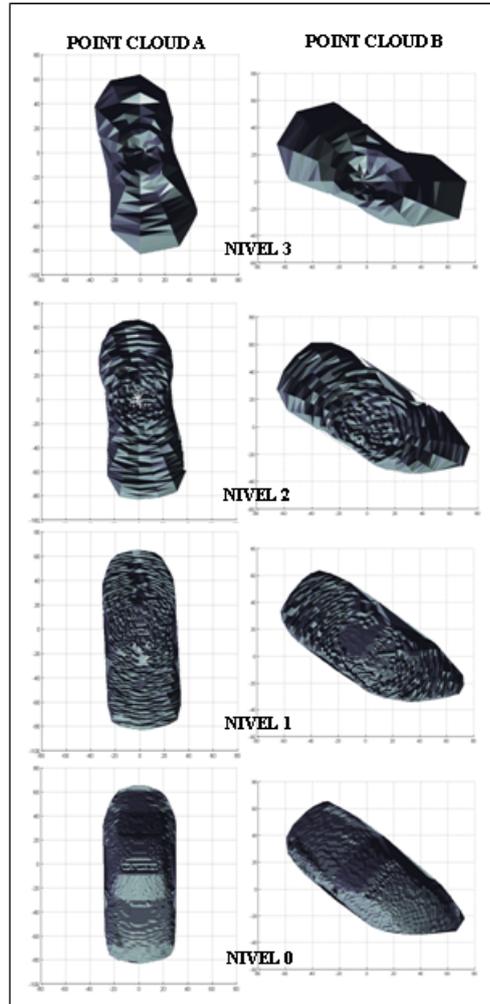


Figura 10.6: Vista en planta de las reconstrucciones 3D obtenidas a partir de las aproximaciones mostradas en la Figura 10.5.

La transformación se compone de tres pasos. En primer lugar, las expresiones siguientes se aplican a todos los pares (i, j) de la matriz:

$$x'_{ij} = \rho_r \cdot (j \cdot \cos(\rho_\theta \cdot (i - 1)) - j_0 \cdot \cos(\rho_\theta \cdot (i_0 - 1))) \quad (10.1)$$

$$y'_{ij} = \rho_r \cdot (j \cdot \sin(\rho_\theta \cdot (i - 1)) - j_0 \cdot \sin(\rho_\theta \cdot (i_0 - 1))) \quad (10.2)$$

La nueva posición (i', j') del píxel a_{ij} será calculada mediante las Ecuaciones (5.1) y (3.2), mientras que el valor del píxel por,

$$c_{i'j'} = c_{ij} - c_{i_0j_0} \quad (10.3)$$

El siguiente paso será referir la nueva imagen CIRCON a la normal en el punto 3D en el que está centrada ahora. La normal se calcula ajustando un plano a los puntos 3D obtenidos a partir de los píxeles vecinos. Por lo tanto, los datos se rotarán aplicando la matriz de la Ecuación A.4 con la normal calculada.

La matriz resultante de la aplicación de estos pasos se denota por $A(i_0, j_0)$.

10.5.2. Medida de Similitud

Antes de realizar cualquier cálculo, los valores mínimos de las dos matrices $A(i_0, j_0)$ y B tienen que ser iguales.

Se va a definir el mínimo de ambas matrices como

$$m = \min(A(i_0, j_0), B) \quad (10.4)$$

Por lo tanto, si $\min(A) > m$ se cambiarán todos los píxeles a_{ij} que valgan $\min(A)$ al nuevo valor m . La misma operación se debería realizar para la matriz B si fuera necesario.

Debido a que se van a utilizar las aproximaciones resultantes de la descomposición wavelet de los descriptores, algunos píxeles pueden sufrir cambios significativos en su valor, por lo que, en vez de utilizar la medida de similitud descrita en el Capítulo 5, que está basada en diferencias entre los valores de los píxeles, se va emplear una medida de similitud basada en el coeficiente de correlación cruzada. Sin embargo, se van a seguir ponderando de manera diferente los píxeles solapados y los no solapados. Por consiguiente, la medida de similitud se va a definir mediante la siguiente expresión:

$$M_s = \frac{\sum_{i=1}^{n_s} \sum_{j=1}^{n_r} (a_{ij} - \bar{A}) \cdot (b_{ij} - \bar{B}) \cdot w_{ij}}{\sqrt{\left(\sum_{i=1}^{n_s} \sum_{j=1}^{n_r} (a_{ij} - \bar{A})^2 \cdot w_{ij} \right) \cdot \left(\sum_{i=1}^{n_s} \sum_{j=1}^{n_r} (b_{ij} - \bar{B})^2 \cdot w_{ij} \right)}} \quad (10.5)$$

donde los valores medios vienen dados por

$$\bar{A} = \frac{1}{n_s \cdot n_r} \sum_{i=1}^{n_s} \sum_{j=1}^{n_r} a_{ij} \quad (10.6)$$

y

$$\bar{B} = \frac{1}{n_s \cdot n_r} \sum_{i=1}^{n_s} \sum_{j=1}^{n_r} b_{ij} \quad (10.7)$$

Se trata de un coeficiente de correlación ponderado. El peso de cada par de píxeles que se evalúa en ambas imágenes equivale a su índice de columna. Por consiguiente, los píxeles a_{ij} y b_{ij} tendrán un peso $w_{ij} = j$. De este modo, los píxeles que se encuentren lejos del punto central serán mejor ponderados, ya que representan un área mayor del objeto. Sin embargo, si $a_{ij} = m$ y $b_{ij} = m$ entonces $w_{ij} = 0$. Además, si $a_{ij} = m$ y $b_{ij} \neq m$ (o viceversa), entonces

$$w_{ij} = j \cdot \left(1 - \frac{\sum_{(i,j) \in I} w_{ij}}{\sum_{(i,j) \in U} w_{ij}} \right) \quad (10.8)$$

donde I y U son, respectivamente, el conjunto intersección y el conjunto unión de los píxeles que cumplen las siguientes condiciones

$$I = \{(i, j) \mid (a_{ij} \neq m) \text{ and } (b_{ij} \neq m)\} \quad (10.9)$$

$$U = \{(i, j) \mid (a_{ij} \neq m) \text{ or } (b_{ij} \neq m)\} \quad (10.10)$$

10.5.3. Eliminación de las Correspondencias No Válidas

Una vez que todos los índices de objeto han sido evaluados para el nivel más bajo de resolución, habrá un valor de similitud M_s y un parámetro k de auto-rotación para cada posible correspondencia. Por lo tanto, estos resultados se pueden visualizar como imágenes, como se muestra en la figura.

Los índices de similitud cuyo valor no supera un cierto umbral τ_L no serán evaluados para el siguiente nivel resolución.

Dado que se utilizan aproximaciones de las imágenes originales, este umbral tendrá un valor diferente para cada nivel L de acuerdo con la siguiente expresión:

$$\tau_L = (\eta_1)^L \cdot M_{s(max)} \quad (10.11)$$

donde η_1 es el porcentaje de $M_{s(max)}$ correspondiente al nivel 1 (en nuestros experimentos $\eta_1 = 0,95$).

Sin embargo, cuando la resolución es baja, es posible que se eliminen algunas correspondencias buenas. Con el fin de preservar sus índices correspondientes, se aplica un umbral más bajo τ'_L a los vecinos de los píxeles que han superado el umbral anterior. Si su valor de similitud es más grande que este nuevo umbral, estos píxeles serán también considerados para el siguiente nivel de resolución.

Para este caso se utiliza también la expresión (10.11) pero con un valor η'_1 menor (en nuestros experimentos, $\eta'_1 = 0,8$).

Por último, los índices que han sido elegidos se modifican para el siguiente nivel de resolución. Notar que el número de filas y columnas depende del nivel de resolución.

En la Figura (10.7), se muestran los valores de similitud y los índices de auto-rotación para los píxeles válidos para cada nivel de resolución. Puede apreciarse que la cantidad de índices que deben comprobarse se reduce progresivamente, de manera que se consigue acelerar la búsqueda de correspondencias.

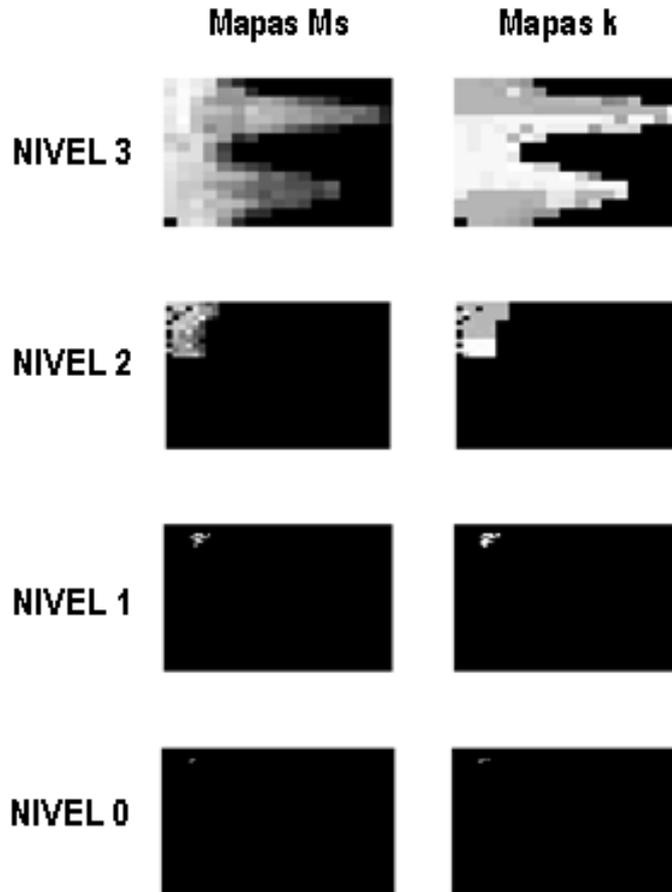


Figura 10.7: Mapas de los valores de similitud y de los índices de auto-rotación.

10.5.4. Verificación de las correspondencias

Una vez que los índices correspondientes al último nivel de resolución han sido evaluados, se eligen, para su verificación, las diez imágenes CIRCON transformadas que tengan los valores de similitud más altos. Un ejemplo se muestra en la Figura (10.8). La primera imagen corresponde a la imagen CIRCON original, mientras que las diez siguientes son sus imágenes transformadas $A(i_0, j_0)$ con los valores de M_s más altos. La última corresponde a la imagen CIRCON B , es decir, la imagen con la que fueron comparadas.

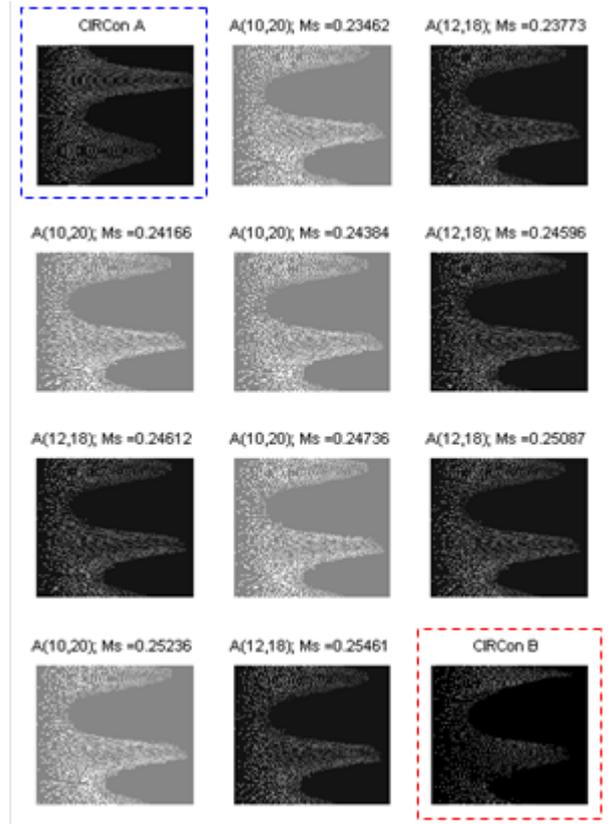


Figura 10.8: Las diez imágenes CIRCON con mayor valor de similitud para el ejemplo.

La siguiente expresión se evalúa para cada una de estas diez imágenes:

$$M_d = \frac{tol}{\sum_{i=1}^{n_s} \sum_{j=1}^{n_r} w_{ij}} \left(\sum_{i=1}^{n_s} \sum_{j=1}^{n_r} \frac{w_{ij}}{\max\{|a_{ij} - b_{ij}| - d_w, tol\}} \right) \quad (10.12)$$

Esto permite verificar si la distancia entre las dos nubes de puntos es suficientemente pequeña y determinar cuál de las imágenes CIRCON proporciona el mejor alineamiento.

Los valores de M_d están comprendidos entre 0 y 1. Si ambas imágenes son exactamente iguales, $M_d = 1$.

Imagen CIRCON transformada	M_s	M_d
A(10,20,43)	0.2346	0.0027
A(12,18,43)	0.2377	0.0484
A(10,20,46)	0.2417	0.0027
A(10,20,47)	0.2438	0.0027
A(12,18,47)	0.2460	0.0573
A(12,18,44)	0.2461	0.0510
A(10,20,45)	0.2474	0.0027
A(12,18,45)	0.2509	0.0491
A(10,20,44)	0.2524	0.0027
A(12,18,46)	0.2546	0.0550

Tabla 10.1: Resultados de Verificación para el ejemplo.

Notar que los pares de píxeles cuyas distancias son menores que $d_w + tol$ tendrán un peso igual a $1/tol$, mientras que para el resto los pesos serán inversamente proporcionales a la distancia entre los dos píxeles. Por consiguiente, esta fórmula da una mayor importancia a los píxeles cuya distancia sea muy pequeña. En los resultados de la verificación de correspondencias que se muestran en la Tabla 1, $d_w = 1$ y $tol = 0,1$.

La imagen CIRCON con el valor M_d más alto se considerará que produce la mejor alineación entre las dos nubes de puntos S_A y S_B . En la Tabla 10.1, la imagen transformada $A(12, 18, 47)$ tiene un valor $M_d = 0,0573$, es decir, el mejor alineamiento se produce cuando los índices de la matriz son: $i_S = 12$, $j_S = 18$, y el índice de auto-rotación es $k_S = 47$.

10.5.5. Alineamiento de las nubes de puntos

Una característica interesante de esta representación 3D es que, una vez que se determina la mejor correspondencia, la transformación euclídea que alinea las dos nubes de puntos depende de los índices i_S , j_S y el número de filas k_S que fueron desplazadas para calcular el máximo valor de similitud.

Transformación Euclídea

En primer lugar, se puede obtener un vector de traslación y una matriz de rotación que transformen el sistema de coordenadas local donde se definió el *descriptor básico* A en el sistema de coordenadas local del descriptor B (el cual ha de coincidir con el del descriptor $A(i_S, j_S, k_S)$).

Vector de Traslación:

$${}^a t_b = \begin{bmatrix} \rho_r \cdot j_S \cdot \cos(-\rho_\theta \cdot (i_S - 1)) \\ \rho_r \cdot j_S \cdot \sin(-\rho_\theta \cdot (i_S - 1)) \\ \rho_z \cdot a_{ij} \end{bmatrix} \quad (10.13)$$

Matriz de Rotación:

$${}^b R_a = R_Z(k_S) \cdot {}^a R_s^T \quad (10.14)$$

donde ${}^a R_s$ es la matriz obtenida al aplicar la Ecuación A.4 y,

$$R_Z(k_S) = \begin{bmatrix} \cos(\rho_\theta \cdot k_S) & \sin(\rho_\theta \cdot k_S) & 0 \\ -\sin(\rho_\theta \cdot k_S) & \cos(\rho_\theta \cdot k_S) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10.15)$$

La matriz de transformación correspondiente sería:

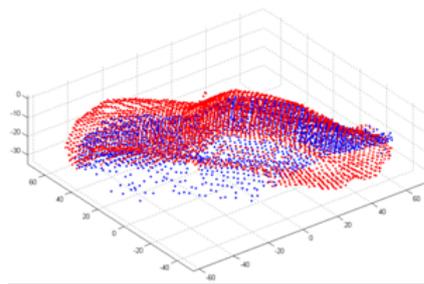
$${}^b T_a = \begin{bmatrix} {}^b R_a & -{}^b R_a \cdot {}^a t_b \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (10.16)$$

Refinamiento de la Transformación Euclídea

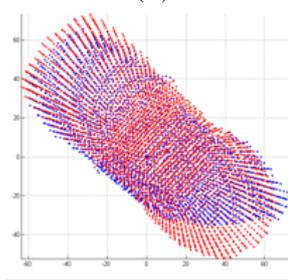
Como consecuencia del cálculo aproximado de los vectores normales y también debido a la descomposición wavelet de las imágenes CIRCON, después de aplicar la transformación euclídea calculada, las nubes de puntos 3D podrían no estar lo suficientemente cerca (ver Figura 10.9a). Por esa razón, será necesario refinar la transformación euclídea obtenida.

Sin embargo, las nubes de puntos 3D son bastante similares, si son vistas desde arriba (Figura 10.9b), ya que las imágenes CIRCON encontradas por las transformaciones tienen un valor de similitud alto y por lo tanto, el porcentaje

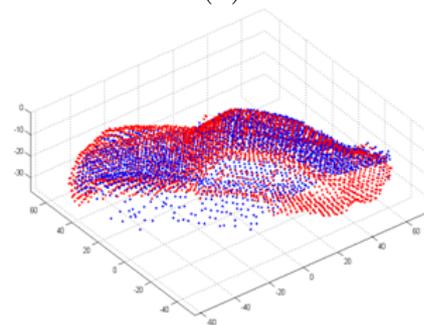
de coincidencia será también elevado. Por lo tanto, habrá un número suficiente de correspondencias para calcular la corrección de la transformación euclídea. El vector de traslación y la matriz de rotación se obtienen mediante el ajuste de un plano a la nube de puntos 3D cuyas coordenadas-z son las diferencias entre las coordenadas-z correspondientes a los píxeles superpuestos en ambas imágenes CIRCON (ver Figura 10.10).



(a)



(b)



(c)

Figura 10.9: Refinamiento de la transformación euclídea.(a) Alineamiento obtenido antes del refinamiento de la transformación.(b) Vista en planta de las nubes de puntos antes del refinamiento.(c) Alineamiento final de las nubes de puntos.

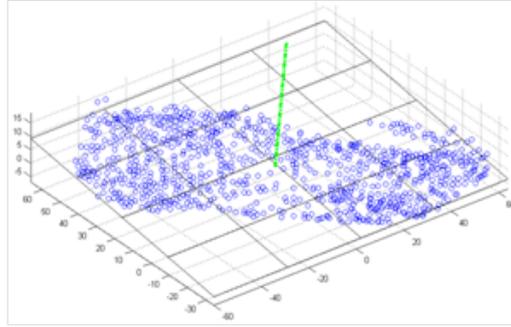


Figura 10.10: Plano ajustado para calcular la corrección de la transformación euclídea.

La ecuación del plano será calculado por análisis de componentes principales; de este modo, se obtienen el vector normal y la distancia al origen d . Por lo tanto, la corrección del vector de traslación Δt tendrá la siguiente expresión:

$$\Delta t = \begin{bmatrix} 0 \\ 0 \\ d \end{bmatrix} \quad (10.17)$$

Y la corrección de la matriz de rotación ΔR se calculará aplicando la Ecuación A.4 con el vector normal obtenido.

En consecuencia, el vector de traslación corregido es el siguiente:

$${}^a t_b = {}^a t_b + {}^b R_a^T \cdot \Delta t \quad (10.18)$$

Y la matriz de rotación:

$${}^b R_a = \Delta R \cdot {}^b R_a \quad (10.19)$$

Los resultados de estas correcciones se muestran en la Figura 10.9c.

Sin embargo, todos los puntos 3D que se muestran en la Figura 10.9 están expresados en el sistema de coordenadas centrado en los puntos de interés utilizados para generar las imágenes CIRCON. Por lo tanto, es necesario tener en cuenta una transformación adicional con el fin de obtener el alineamiento final entre las nubes de puntos iniciales mostradas en la Figura 10.3.

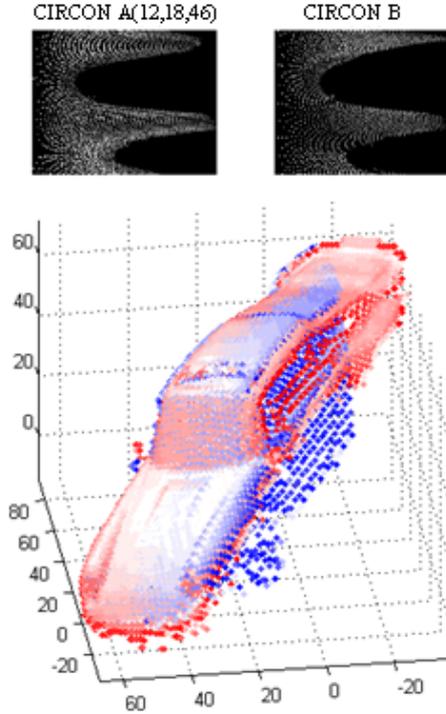


Figura 10.11: Alineamiento Final.

Sean ${}^aT_{w_1}$ y ${}^bT_{w_2}$ las matrices de transformación que relacionan el s.d.c. original de cada nube de puntos con el s.d.c. local en sus respectivos puntos de interés ${}^{w_1}p_{1a}$ y ${}^{w_2}p_{2b}$. Entonces, la transformación euclídea ${}^{w_2}T_{w_1}$ que alinea de manera aproximada las dos nubes de puntos se obtiene mediante la siguiente expresión:

$${}^{w_2}T_{w_1} = {}^bT_{w_2}^{-1} \cdot {}^bT_a \cdot {}^aT_{w_1} = \quad (10.20)$$

$$\begin{bmatrix} {}^{w_2}R_b & {}^{w_2}p_{2b} \\ 0_{1 \times 3} & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^bR_a & -{}^bR_a \cdot {}^a t_b \\ 0_{1 \times 3} & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^{w_1}R_a^T & -{}^{w_1}R_a^T \cdot {}^{w_1}p_{1a} \\ 0_{1 \times 3} & 1 \end{bmatrix}$$

Como se muestra en la Figura 10.11, la transformación aproximada obtenida es lo suficientemente buena para ser utilizada como estimación inicial por el algoritmo ICP [14, 22], si se requiriera un alineamiento más fino.

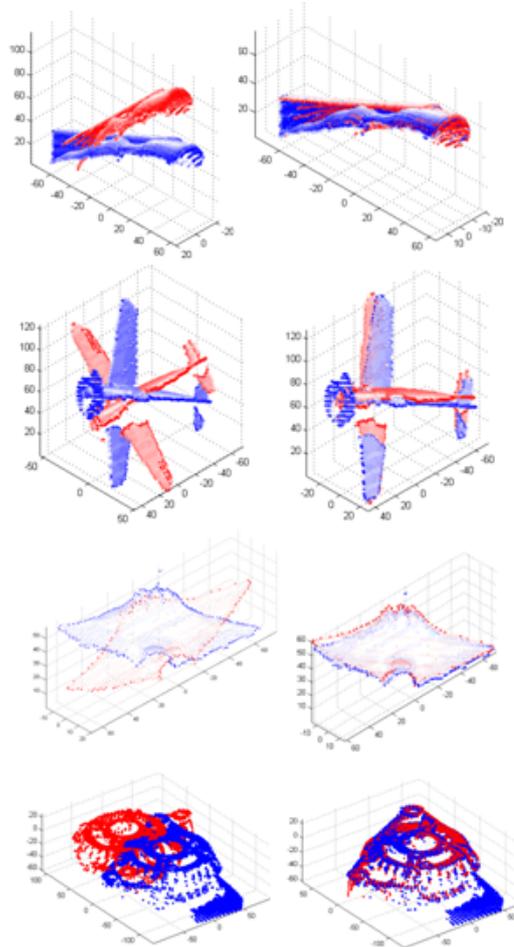


Figura 10.12: Resultados obtenidos para diferentes objetos.

10.6. Resultados

Otros resultados obtenidos para diferentes objetos se muestran en la Figura 10.12. Los tres primeros objetos proceden de la base de datos de imágenes de rango de la Universidad de Stuttgart [56], mientras que el último objeto fue escaneado mediante el sistema de visión 3D descrito en el Capítulo 4.

Como se muestra en la Figura 10.12, el alineamiento obtenido para cada uno de estos objetos puede considerarse aceptable, ya que su transformación euclídea correspondiente se puede utilizar para inicializar el algoritmo ICP y de este modo obtener un alineamiento más fino.

Capítulo 11

Conclusiones, contribuciones y trabajo futuro

11.1. Conclusiones

Dentro del problema de la captura robotizada de objetos, nos habíamos planteado como objetivo obtener la posición y orientación de éstos a partir de la información proporcionada por dispositivos de adquisición 3D tales como escáneres láser o cámaras de tiempo de vuelo. Si se dispone de un modelo de los objetos, este problema puede reducirse a encontrar una transformación rígida que alinee dos nubes de puntos (escena y modelo) cuya posición y orientación relativa es desconocida. Además, el alineamiento de nubes de puntos puede utilizarse para la construcción de modelos 3D e incluso en complejas tareas de reconocimiento de objetos tales como el problema de "Bin Picking".

Tras revisar el estado del arte, se encontraron algunos inconvenientes que presentaban los algoritmos de alineamiento más relevantes: falta de generalidad (funcionan bien con objetos de una determinada topología), tiempo de computación excesivo, problemas con las simetrías, mal comportamiento cuando las nubes de puntos tienen baja densidad y cuando presentan bajo solapamiento entre sí, necesidad de un método para descartar falsas correspondencias y de un grupo de correspondencias válidas para poder obtener la transformación euclídea. Todos los métodos revisados presentaban en mayor o menor medida

al menos dos de estos inconvenientes.

Las tres características que hemos primado a la hora de diseñar nuestro algoritmo de alineamiento es que no presente restricciones respecto al tipo de objetos que pueden usarse, que se comporte bien ante la presencia de simetrías (algo que es bastante habitual en piezas industriales) y que mantenga un buen comportamiento cuando el solapamiento y la densidad de las nubes de puntos son bajos. Sin embargo, también se han tenido en cuenta el resto de inconvenientes que podían presentarse.

Atendiendo a la característica de generalidad, el algoritmo basado en *spin images* es quizás el método que tiene un mejor comportamiento. Sin embargo, no funciona bien con objetos simétricos o cuando hay partes repetidas en los objetos (esto último se debe a que tiene un ámbito demasiado local). Además este método (como también ocurre con otros estudiados) necesita obtener un grupo de correspondencias para poder calcular una transformación euclídea que alinee de forma aproximada las nubes de puntos. Y debido a que el proceso de búsqueda de correspondencias no descarta de forma adecuada las que son falsas, necesita llevar a cabo un proceso de evaluación que establezca grupos de correspondencias que tengan consistencia geométrica.

Para evitar los problemas enumerados, se propone un descriptor que representa la geometría local de los puntos de interés de las nubes de puntos.

A diferencia de las *spin images* o de los contextos de forma 3D (que siguen una estructura similar a nuestro descriptor), el descriptor propuesto no utiliza histogramas, sino que los valores de los píxeles representan valores de altura de las celdas en las que se divide el entorno de un punto de interés. Esto va a favorecer su utilización con nubes de puntos de baja densidad y con bajo solapamiento, ya que el valor de los píxeles de la matriz no dependen de la cantidad de puntos por celda, sino que basta con que haya un solo punto en una celda para que su píxel correspondiente tenga asignado un valor.

Por otro lado, se diferencia de las imágenes log-polar de rango local en que no utiliza una división radial logarítmica, sino lineal; de esta forma, la descripción del entorno del punto de interés va a ser más detallada, de manera que se pueda valorar mejor la similitud entre descriptores. En nuestro caso se ha aprovechado su estructura matricial y sus propiedades geométricas, para

diseñar una medida de similitud basada en las distancias entre sus píxeles, ya que están íntimamente relacionadas con la distancia entre los puntos de la nube. Además de permitirnos clasificar las correspondencias en base a distancias y al grado de solapamiento de los descriptores, la evaluación de la medida de similitud nos va a proporcionar una información extra. En vez de tratar de hacer invariante a rotación el descriptor, nosotros aprovechamos que es cíclico para determinar qué desplazamiento de filas produce la máxima similitud y así calcular una matriz de rotación sobre la normal. Esto permite calcular una matriz de transformación euclídea con tan sólo una correspondencia.

Además, al haber diseñado una medida de similitud que aprovecha las características específicas del descriptor y que tiene en cuenta el porcentaje de solapamiento (y no el número absoluto de píxeles solapados, como en las *spin images*), vamos a poder incrementar el radio del entorno de los puntos de interés (y de este modo el poder discriminante) y así evitar el inconveniente de las simetrías de los objetos.

Otra importante ventaja de nuestro algoritmo es que puede finalizar en cualquier momento, no necesita evaluar obligatoriamente todas las combinaciones entre los puntos de interés de ambas nubes. Cuando se obtiene una correspondencia que tiene un valor de similitud que supera al máximo alcanzado hasta ese momento, se evalúa un criterio de parada. Este criterio está basado en dos medidas de distancia (relativas a rotación y traslación) entre las transformaciones euclídeas asociadas a la correspondencia actual y a una correspondencia ficticia creada a partir de la actual y otras dos que se obtienen de dos pares de celdas correspondientes de la distribución en torno al punto encontrado por el algoritmo. En otras palabras, el algoritmo propuesto no necesita llevar a cabo un proceso de evaluación de las correspondencias para crear grupos geoméricamente consistentes.

Por otro lado, los resultados muestran que la medida de similitud soporta bajas densidades cuando el solapamiento es alto y que, a medida que éste disminuye, la densidad de las nubes de puntos debe aumentar para mantener los errores de rotación y traslación dentro de unos límites prefijados. También ha podido observarse que el algoritmo se comporta razonablemente bien con objetos de diferente topología, en condiciones de bajo solapamiento y con nubes

de puntos de baja densidad, además de poder alinear nubes de puntos de diferente procedencia y diferente densidad. Aunque cuando el solapamiento es muy bajo (menor que el 20 %) el algoritmo de alineamiento tiende a seleccionar falsas correspondencias; lo mismo ocurre cuando la densidad de las nubes de puntos es muy baja (en este caso la obtención de un buen alineamiento depende más de la topología del objeto).

Para acelerar la búsqueda de correspondencias se propuso también una versión que utiliza una descomposición wavelet multinivel de la imagen CIRCON en el punto más cercano al centroide de cada nube. Este algoritmo utiliza la forma 3D simplificada de la nube 1 (que se puede obtener a partir de las aproximaciones wavelet de ese *descriptor básico*) para construir nuevos descriptores y evaluar su correspondencia con el *descriptor básico* de la nube 2. Aunque esto reduce notablemente el tiempo de ejecución del algoritmo (ya que utiliza nubes de puntos de muy escasa densidad), también es cierto que su éxito depende en gran medida de la topología de los objetos. Debido a que todos los descriptores evaluados se construyen a partir de la información del *descriptor básico*, el vector normal en un gran número de puntos de la nube no debe ser muy diferente al del punto de interés donde se creó dicho descriptor. Por esa razón esta versión simplificada del algoritmo original se ha de utilizar solamente cuando los objetos cumplan estos requisitos, ya que, en caso contrario, la posibilidad de obtener un buen alineamiento se reduce notablemente.

11.2. Contribuciones

Consideramos que este trabajo de tesis aporta las siguientes contribuciones al alineamiento de nubes de puntos 3D:

- Se ha propuesto un nuevo descriptor (CIRCON) que representa, mediante una imagen cíclica, la geometría del entorno de un punto de interés de la nube. Para construir la matriz de imagen se utiliza una distribución de los puntos en sectores que, a su vez, son divididos en celdas que tienen la misma longitud radial. Los valores de los elementos de la matriz representan la máxima coordenada z de los puntos contenidos en sus celdas

correspondientes.

- Se ha diseñado una medida de similitud que tiene en cuenta tanto las distancias entre los píxeles de los descriptores como su porcentaje de solapamiento. Además se aprovecha el carácter cíclico del descriptor para obtener, además del valor de similitud, un índice que representa la rotación sobre la normal en el punto de interés. Dado que los descriptores se van a poder comparar sin tener que restringir el entorno del punto de interés, podría aumentarse su poder discriminante para alinear objetos que presenten simetrías o partes repetidas.
- Con el descriptor y la medida de similitud propuestos se puede calcular una matriz de transformación euclídea determinando una única correspondencia.
- Se ha diseñado un algoritmo de alineamiento aproximado que combina el descriptor y la medida de similitud propuestos para clasificar las correspondencias evaluadas en función de su medida de similitud y así atenuar la aparición de falsas correspondencias. Una de las principales ventajas de este algoritmo es que tiene la posibilidad de finalizar cuando encuentra una correspondencia que supera el máximo valor de similitud alcanzado hasta ese momento.
- Se propone un criterio de parada para el algoritmo de alineamiento que también utiliza las características propias del descriptor. Utiliza la correspondencia encontrada y las celdas en las que se distribuyen los puntos para crear una correspondencia ficticia con la que se calcula otra matriz de transformación. Entonces ésta se compara con la obtenida por el algoritmo. Éste finalizará su ejecución si una medida de distancia relativa a la rotación y otra relativa a la traslación no sobrepasan sus respectivos umbrales.
- Se ha propuesto un algoritmo basado en una descomposición wavelet de los descriptores y en la utilización de la forma 3D simplificada para construir nuevos descriptores con los que evaluar correspondencias. Este

algoritmo simplificado sólo se recomienda para el alineamiento de objetos en los que predomine una determinada dirección de la normal.

- Se ha propuesto un sistema de visión basado en el descriptor CIRCON que utiliza triangulación activa láser para obtener contornos radiales de los objetos. Se propone también cómo adaptar este sistema de visión para poder montarlo sobre un manipulador robótico.
- Se propone una adaptación del algoritmo ICP que utiliza la distribución de celdas en torno a los puntos de interés que obtuvieron el mayor valor de similitud en el algoritmo de alineamiento para establecer nuevas correspondencias entre las nubes de puntos. Estas correspondencias permitirán refinar de forma iterativa la transformación euclídea que obtuvo el algoritmo de alineamiento aproximado.

11.3. Trabajo futuro

Aunque los objetivos principales que nos habíamos fijado en esta tesis han quedado cubiertos, aún quedan varios aspectos que podrían ser mejorados. Así, consideramos que la futura línea de trabajo debería tratar de abordar las siguientes cuestiones:

- Aunque los resultados muestran que el algoritmo propuesto es capaz de encontrar un alineamiento adecuado a pesar utilizar unos sencillos criterios de selección de los puntos de interés, en algunas ocasiones esos puntos de partida no son los más adecuados y el algoritmo ha de realizar más iteraciones de las que serían necesarias. Como una de las ventajas del algoritmo que hemos propuesto es que puede finalizar en el momento que encuentra una correspondencia con una similitud alta y que cumple los criterios de parada, si seleccionásemos los puntos de interés de una manera más adecuada es muy probable que el algoritmo pudiese finalizar en las primeras iteraciones la gran mayoría de las veces.
- También se puede prestar atención a la reducción de las nubes de puntos. El sencillo algoritmo de reducción que hemos utilizado muestra, a tenor

de los resultados, que nuestro algoritmo es suficientemente robusto para encontrar un alineamiento aproximado adecuado. Sin embargo, podría ser interesante utilizar un algoritmo de reducción que preserve mejor la información de la nube de puntos original.

- Aunque se propone una forma de seleccionar los parámetros de la medida de similitud, creemos que su eficacia podría verse aumentada si seleccionamos estos parámetros mediante un algoritmo genético que encuentre el par óptimo que clasifique correctamente las correspondencias entre nubes de puntos pertenecientes a objetos de diferente topología y con diferente grado de solapamiento.
- Otro aspecto a mejorar es el de la implementación. El algoritmo se ha implementado en el lenguaje de programación de *Matlab*[®], por lo que su eficiencia podría mejorarse si se programa en otro lenguaje de alto nivel.
- Utilizando la experiencia previa de nuestro grupo en visión ultrasónica [64, 65, 90, 91] trataremos de extender el algoritmo del Capítulo 7 al alineamiento de datos ultrasónicos e imágenes de rango procedentes de escáneres láser y cámara de tiempo de vuelo.
- Como ya se ha comentado, además de poder utilizar el algoritmo de alineamiento para determinar la posición y orientación de los objetos dado su modelo correspondiente, se podría emplear también para reconocerlos en situaciones complejas en las que éstos aparecen apilados y mezclados con otros objetos diferentes ("Bin Picking problem"). Ésta será la línea preferente de investigación que seguiremos en el futuro debido a la gran dificultad y al desafío que supone.

Apéndice A

Sistema de Coordenadas Local

La elección de los ejes del sistema de coordenadas local se introdujo en la Sección 3.3.3, donde se presentaron las expresiones necesarias para calcular la matriz de rotación (Ecuaciones 3.4, 3.5 y 3.6).

Si desarrollamos la Ecuación 3.5, podemos obtener las componentes del vector \hat{x}_q en función de las componentes de la normal en el punto de interés ${}^w p_q$. Para ello vamos a calcular, en primer lugar, el producto escalar del vector unitario del eje Y_w y el vector normal \vec{n}_q .

$$\hat{y}_w \times \vec{n}_q = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0 & 1 & 0 \\ n_{qx} & n_{qy} & n_{qz} \end{vmatrix} = n_{qz} \cdot \mathbf{i} - n_{qx} \cdot \mathbf{k} \quad (\text{A.1})$$

Luego el vector unitario \hat{x}_q será:

$$\hat{x}_q = \frac{1}{\sqrt{n_{qx}^2 + n_{qz}^2}} \cdot \begin{bmatrix} n_{qz} \\ 0 \\ -n_{qx} \end{bmatrix} \quad (\text{A.2})$$

Notar que este vector se encuentra siempre en el plano $X_w Z_w$ independientemente de la normal \vec{n}_q . Además tiene la misma dirección que otro vector del mismo plano cuya coordenada x es igual a la coordenada z de la normal y su coordenada z es la coordenada x de la normal cambiada de signo.

Para determinar el vector unitario del eje restante, Y_q , hay que calcular el producto escalar de los otros dos vectores unitarios del sistema de coordenadas local:

$$\begin{aligned}\hat{y}_q &= \hat{z}_q \times \hat{x}_q = \vec{n}_q \times \hat{x}_q = \frac{1}{\sqrt{n_{qx}^2 + n_{qz}^2}} \cdot \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ n_{qx} & n_{qy} & n_{qz} \\ n_{qz} & 0 & -n_{qx} \end{vmatrix} = \\ &= \frac{1}{\sqrt{n_{qx}^2 + n_{qz}^2}} \cdot (-n_{qx} \cdot n_{qy} \cdot \mathbf{i} + (n_{qx}^2 + n_{qz}^2) \cdot \mathbf{j} - n_{qy} \cdot n_{qz} \cdot \mathbf{k}) \quad (\text{A.3})\end{aligned}$$

Sustituyendo en la Ecuación 3.4 los vectores hallados, se obtiene la matriz de rotación expresada solamente en función de las componentes del vector normal en el punto de interés.

$${}^w R_q = \begin{bmatrix} \frac{n_{qz}}{\sqrt{n_{qx}^2 + n_{qz}^2}} & \frac{-n_{qx} \cdot n_{qy}}{\sqrt{n_{qx}^2 + n_{qz}^2}} & n_{qx} \\ 0 & \frac{n_{qx}^2 + n_{qz}^2}{\sqrt{n_{qx}^2 + n_{qz}^2}} & n_{qy} \\ \frac{-n_{qx}}{\sqrt{n_{qx}^2 + n_{qz}^2}} & \frac{-n_{qy} \cdot n_{qz}}{\sqrt{n_{qx}^2 + n_{qz}^2}} & n_{qz} \end{bmatrix} \quad (\text{A.4})$$

Una propiedad que tiene esta elección de los ejes del s.d.c. local es que los vectores \hat{y}_q , \hat{y}_w y \vec{n}_q son coplanares. Esto se puede demostrar comprobando que el determinante de la matriz formada por los tres vectores es nulo:

$$\begin{vmatrix} \frac{-n_{qx} \cdot n_{qy}}{\sqrt{n_{qx}^2 + n_{qz}^2}} & \frac{n_{qx}^2 + n_{qz}^2}{\sqrt{n_{qx}^2 + n_{qz}^2}} & \frac{-n_{qy} \cdot n_{qz}}{\sqrt{n_{qx}^2 + n_{qz}^2}} \\ 0 & 1 & 0 \\ n_{qx} & n_{qy} & n_{qz} \end{vmatrix} = -\frac{n_{qx} \cdot n_{qy} \cdot n_{qz}}{\sqrt{n_{qx}^2 + n_{qz}^2}} + \frac{n_{qx} \cdot n_{qy} \cdot n_{qz}}{\sqrt{n_{qx}^2 + n_{qz}^2}} = 0 \quad (\text{A.5})$$

Otra forma de obtener la matriz de rotación ${}^q R_w$ es mediante los ángulos de Euler α , β , γ , que equivalen a las sucesivas rotaciones que han de realizarse sobre los ejes Z, Y, X para transformar el s.d.c. global en el s.d.c. local. Notar que, una vez efectuada una rotación, la siguiente se aplica sobre el s.d.c. resultante.

En la Figura A.1 se muestran los ángulos β y γ que han de girar los ejes

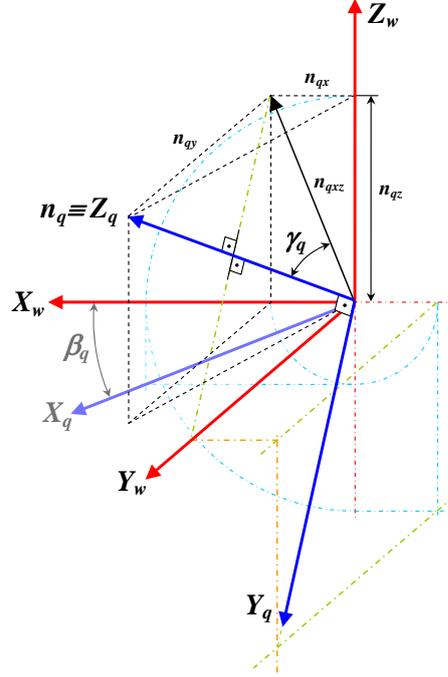


Figura A.1: Obtención de los ejes del sistema de coordenadas local a partir del s.d.c. global.

(siendo $\alpha = 0$). En primer lugar, se girará sobre el eje Y_w un ángulo β y a continuación, sobre el eje X_q del s.d.c. resultante, se girará un ángulo γ en sentido horario. Por consiguiente, la matriz de rotación que describe la transformación entre ambos sistemas de coordenadas será la siguiente:

$${}^w R_q = R_Y(\beta_q) \cdot R_X(-\gamma_q) \quad (\text{A.6})$$

El ángulo β_q es el que forma el vector unitario \hat{x}_q (Ecuación A.2) con el eje X_w :

$$\beta_q = \arctan\left(\frac{n_{qx}}{n_{qz}}\right) \quad (\text{A.7})$$

Para calcular el ángulo γ_q habrá que considerar el triángulo mostrado en la Figura A.2. De ahí se puede fácilmente deducir que:

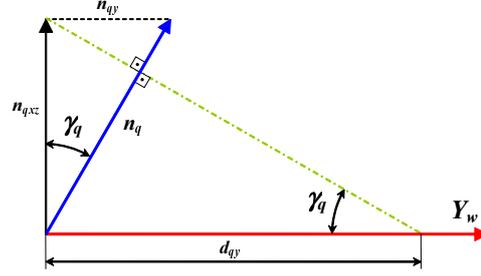


Figura A.2: Obtención del ángulo γ_q a partir de los triángulos que se pueden extraer de la Figura A.1.

$$\gamma_q = \arctan\left(\frac{n_{qy}}{n_{qxz}}\right) = \arctan\left(\frac{n_{qy}}{\sqrt{n_{qx}^2 + n_{qz}^2}}\right) \quad (\text{A.8})$$

Una vez obtenidas las expresiones de β_q y γ_q ya podemos calcular la matriz ${}^w R_q$ mediante la Ecuación A.6.

$$\begin{aligned} {}^w R_q &= R_Y(\beta_q) \cdot R_X(-\gamma_q) \\ &= \begin{bmatrix} \cos \beta_q & 0 & \sin \beta_q \\ 0 & 1 & 0 \\ -\sin \beta_q & 0 & \cos \beta_q \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma_q & \sin \gamma_q \\ 0 & -\sin \gamma_q & \cos \gamma_q \end{bmatrix} \\ &= \begin{bmatrix} \cos \beta_q & -\sin \beta_q \cdot \sin \gamma_q & \sin \beta_q \cdot \cos \gamma_q \\ 0 & \cos \gamma_q & \sin \gamma_q \\ -\sin \beta_q & -\cos \beta_q \cdot \sin \gamma_q & \cos \beta_q \cdot \cos \gamma_q \end{bmatrix} \\ &= \begin{bmatrix} \frac{n_{qz}}{\sqrt{n_{qx}^2 + n_{qz}^2}} & -\frac{n_{qx}}{\sqrt{n_{qx}^2 + n_{qz}^2}} \cdot n_{qy} & \frac{n_{qx}}{\sqrt{n_{qx}^2 + n_{qz}^2}} \cdot \frac{n_{qx}^2 + n_{qz}^2}{\sqrt{n_{qx}^2 + n_{qz}^2}} \\ 0 & \frac{n_{qx}^2 + n_{qz}^2}{\sqrt{n_{qx}^2 + n_{qz}^2}} & n_{qy} \\ -\frac{n_{qx}}{\sqrt{n_{qx}^2 + n_{qz}^2}} & -\frac{n_{qz}}{\sqrt{n_{qx}^2 + n_{qz}^2}} \cdot n_{qy} & \frac{n_{qz}}{\sqrt{n_{qx}^2 + n_{qz}^2}} \cdot \frac{n_{qx}^2 + n_{qz}^2}{\sqrt{n_{qx}^2 + n_{qz}^2}} \end{bmatrix} \\ &= \begin{bmatrix} \frac{n_{qz}}{\sqrt{n_{qx}^2 + n_{qz}^2}} & \frac{-n_{qx} \cdot n_{qy}}{\sqrt{n_{qx}^2 + n_{qz}^2}} & n_{qx} \\ 0 & \frac{n_{qx}^2 + n_{qz}^2}{\sqrt{n_{qx}^2 + n_{qz}^2}} & n_{qy} \\ \frac{-n_{qx}}{\sqrt{n_{qx}^2 + n_{qz}^2}} & \frac{-n_{qy} \cdot n_{qz}}{\sqrt{n_{qx}^2 + n_{qz}^2}} & n_{qz} \end{bmatrix} \quad (\text{A.9}) \end{aligned}$$

Como era de esperar, este resultado es igual al obtenido en la Ecuación A.4.

De la Figura A.2 puede también obtenerse la coordenada y de un vector que tenga la misma dirección que el vector \hat{y}_q . Para ello vamos a expresar este último de una manera diferente a la mostrada en la Ecuación A.3:

$$\hat{y}_q = \frac{1}{\sqrt{n_{qx}^2 + n_{qz}^2}} \begin{bmatrix} -n_{qx} \cdot n_{qy} \\ n_{qx}^2 + n_{qz}^2 \\ -n_{qy} \cdot n_{qz} \end{bmatrix} = \frac{n_{qy}}{\sqrt{n_{qx}^2 + n_{qz}^2}} \begin{bmatrix} -n_{qx} \\ \sqrt{n_{qx}^2 + n_{qz}^2} \cdot \frac{\sqrt{n_{qx}^2 + n_{qz}^2}}{n_{qy}} \\ -n_{qz} \end{bmatrix} \quad (\text{A.10})$$

Sustituyendo la Ecuación A.8 en la ecuación anterior, se obtiene:

$$\hat{y}_q = \tan \gamma_p \cdot \begin{bmatrix} -n_{qx} \\ \frac{n_{qxz}}{\tan \gamma_p} \\ -n_{qz} \end{bmatrix} = \tan \gamma_p \cdot \vec{d}_q \quad (\text{A.11})$$

Siendo \vec{d}_q un vector que tiene la misma dirección que el vector \hat{y}_q . Sus coordenadas x y z son iguales a las de la normal \vec{n}_q pero cambiadas de signo, mientras que su coordenada d_{qy} se puede obtener gráficamente sobre el eje Y_w tal y como se muestra en la Figura A.2.

Apéndice B

Selección de parámetros de la medida de similitud

Mediante una adecuada selección de los parámetros ρ y λ' se va a poder modificar el valor que tomará la medida de similitud para los diferentes valores de D_{ov} y σ_{ov} .

Si la distancia promedio en el solapamiento fuese nula ($D_{ov} = 0$), la medida de similitud dependería del porcentaje de solapamiento y del parámetro λ' .

$$M_S = \frac{\sigma_{ov}}{\lambda' + \sigma_{ov} \cdot (1 - \lambda')} = \frac{1}{1 + \lambda' \cdot \left(\frac{1}{\sigma_{ov}} - 1\right)} \quad (\text{B.1})$$

Notar que, si $\lambda' = 1 \Rightarrow M_S = \sigma_{ov}$

Por el contrario, si $\sigma_{ov} = 1$, la medida de similitud sólo va a depender de D_{ov} y del parámetro ρ :

$$M_S = \frac{1}{(\rho \cdot D_{ov} + \rho \cdot \lambda) + 1 \cdot (1 - \rho \cdot \lambda)} = \frac{1}{\rho \cdot D_{ov} + 1} \quad (\text{B.2})$$

De las Ecuaciones B.1 y B.2 se puede deducir que, variando el parámetro λ' , se modifica el grado de importancia que va a tener el solapamiento en la medida de similitud, mientras que, con el parámetro ρ , se actúa sobre la influencia de D_{ov} .

Para poder observar esa influencia vamos a obtener las expresiones de ρ y λ' en función de M_S .

Despejando el parámetro ρ de la Ecuación (5.18), resulta:

$$\rho = \frac{\sigma_{ov}}{D_{ov}} \cdot \left(\frac{1}{M_S} - 1 + \lambda' \right) - \frac{\lambda'}{D_{ov}} \quad (\text{B.3})$$

Y despejando el parámetro λ' :

$$\lambda' = \frac{\sigma_{ov} \cdot \left(\frac{1}{M_S} - 1 \right) - \rho \cdot D_{ov}}{1 - \sigma_{ov}} \quad (\text{B.4})$$

Si consideramos el caso ideal (haciendo $M_S = 1$ en la expresión de ρ), entonces:

$$\rho = \frac{\sigma_{ov}}{D_{ov}} \cdot \left(\frac{1}{1} - 1 + \lambda' \right) - \frac{\lambda'}{D_{ov}} = -\frac{\lambda'}{D_{ov}} \cdot (1 - \sigma_{ov}) \quad (\text{B.5})$$

Despejando λ' se llega a la condición que debe cumplir λ para que M_S valga la unidad:

$$\frac{\lambda'}{\rho} = -\frac{D_{ov}}{1 - \sigma_{ov}} = \lambda \quad (\text{B.6})$$

Si se hubiese empleado la Ecuación (B.4) el resultado habría sido el mismo.

Dado que $D_{ov} \geq 0$ y que $\sigma_{ov} \leq 1$, el valor de λ debería ser negativo para que M_S fuera la unidad. Como ya se dijo anteriormente, λ debe ser positivo para considerar el efecto desfavorable que tiene el área no solapada en la medida de similitud. Por lo tanto la única posibilidad de que M_S sea la unidad es que D_{ov} sea cero y σ_{ov} valga uno, lo cual va a hacer que el valor de λ quede indeterminado. De ahí que, para calcular los valores de ρ y λ' , se van a considerar por separado esas dos condiciones:

- Con $D_{ov} = 0$ en la Ecuación (B.4), el parámetro λ' sólo depende de M_S y σ_{ov} :

$$\lambda' = \frac{\frac{1}{M_S} - 1}{\frac{1}{\sigma_{ov}} - 1} \quad (\text{B.7})$$

- Con $\sigma_{ov} = 1$ en la Ecuación (B.3), el parámetro ρ sólo depende de M_S y D_{ov} :

$$\rho = \frac{\frac{1}{M_S} - 1}{D_{ov}} \quad (\text{B.8})$$

Ejemplo 1

Vamos a considerar el caso especial en el que la distancia promedio en el área solapada sea nula ($D_{ov} = 0$) y el porcentaje de solapamiento sea $\sigma_{ov} = 0,5$. Si para este caso especial consideramos que la medida de similitud debería ser 0,5, entonces, sustituyendo en la Ecuación (B.7):

$$\lambda' = \frac{\frac{1}{0,5} - 1}{\frac{1}{0,5} - 1} = 1$$

Si σ_{ov} fuese la unidad, vamos a suponer que una distancia unidad en el área solapada ($D_{ov} = 1$) debería proporcionar el mismo valor $M_S = 0,5$. Para que esto suceda, el parámetro ρ debe valer, según la Ecuación (B.8):

$$\rho = \frac{\frac{1}{0,5} - 1}{1} = 1$$

Notar que, para este valor de ρ , $\lambda = 1$.

Por consiguiente, la medida de similitud tendría, en este caso, la siguiente expresión:

$$M_S = \frac{\sigma_{ov}}{(1 \cdot D_{ov} + 1) + \sigma_{ov} \cdot (1 - 1)} = \frac{\sigma_{ov}}{D_{ov} + 1} \quad (\text{B.9})$$

La superficie descrita por M_S al variar D_{ov} y σ_{ov} se muestra en la Figura B.1.

Ejemplo 2

Si consideramos que la medida de similitud debería ser 0,5 cuando $D_{ov} = 0$ y el porcentaje de solapamiento es $1/3$, entonces:

$$\lambda' = \frac{\frac{1}{0,5} - 1}{3 - 1} = \frac{1}{2}$$

Para el caso en el que σ_{ov} fuese la unidad, vamos a suponer que un valor de D_{ov} igual a $1/3$ debería proporcionar el mismo valor de $M_S = 0,5$. Para que esto suceda, el parámetro ρ debe valer:

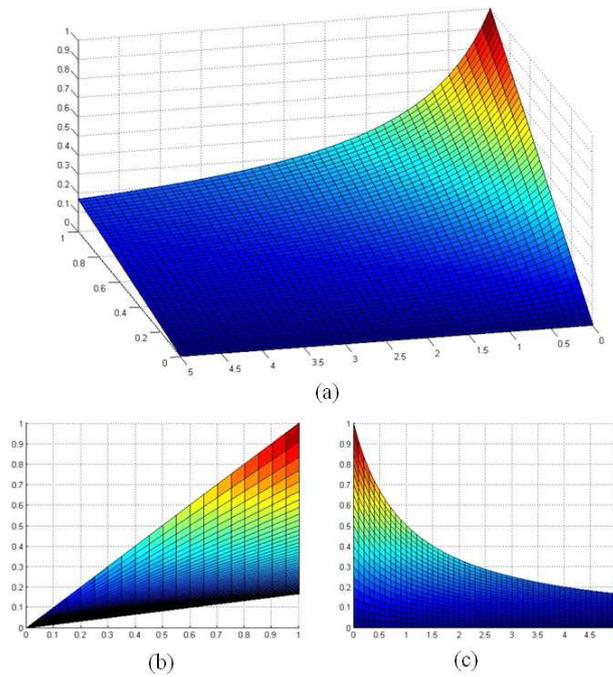


Figura B.1: (a) Superficie descrita por M_S para las distintas combinaciones de D_{ov} y σ_{ov} cuando $\lambda=1$ y $\rho=1$. (b) Variación de M_S con σ_{ov} . (c) Variación de M_S con D_{ov} .

$$\rho = \frac{\frac{1}{0,5} - 1}{1/3} = 3$$

Para este valor de ρ , $\lambda = 1/6$.

Por consiguiente, la medida de similitud tendría la siguiente expresión:

$$M_S = \frac{\sigma_{ov}}{(3 \cdot D_{ov} + 1/2) + \sigma_{ov} \cdot (1 - 1/2)} = \frac{2 \cdot \sigma_{ov}}{6 \cdot D_{ov} + \sigma_{ov} + 1} \quad (\text{B.10})$$

En la Figura B.2 se muestra el cambio que se produce en la superficie descrita por M_S , con respecto a la mostrada en la Figura B.1, cuando se modifican los parámetros λ y ρ con el fin de satisfacer los nuevos requisitos impuestos.

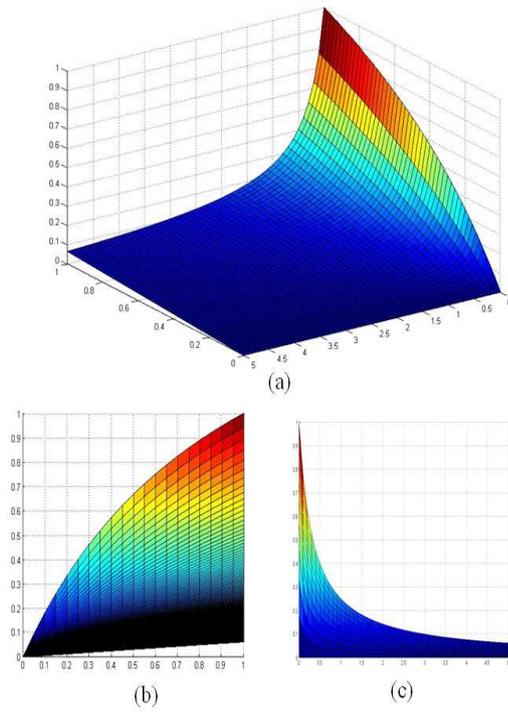


Figura B.2: (a) Superficie descrita por M_S para las distintas combinaciones de D_{ov} y σ_{ov} cuando $\lambda=1/6$ y $\rho=3$. (b) Variación de M_S con σ_{ov} . (c) Variación de M_S con D_{ov} .

Apéndice C

Implementación de la medida de similitud

C.1. Implementación para los n_s giros sobre la normal

Para poder realizar las n_s combinaciones posibles de una manera eficiente, vamos a implementarlas en forma de matrices de bloques, de manera que todas ellas puedan evaluarse de una sola vez y sin tener que utilizar bucles 'for'.

Como el descriptor que se va a rotar es A , se va a construir una matriz C_A que tenga n_s filas (una por cada posible giro) y $n_s \cdot n_r$ columnas, de modo que la fila f -ésima de la matriz va a contener, en orden ascendente, las filas del descriptor CIRCON que corresponda a un giro de $-f \cdot \rho_\theta$ radianes. Por consiguiente, si llamamos c_{Af} a la fila f -ésima de C_A y a_1, a_2, \dots, a_{n_s} a cada una de las filas del descriptor A , entonces:

$$c_{Af} = \left[a_{(n_s-f+1)} \quad a_{(n_s-f+2)} \quad \cdots \quad a_{n_s} \quad a_1 \quad a_2 \quad \cdots \quad a_{(n_s-f)} \right] \quad (\text{C.1})$$

De modo que la matriz total sería:

$$C_A = \begin{bmatrix} a_{n_s} & a_1 & a_2 & \cdots & a_{(n_s-4)} & a_{(n_s-3)} & a_{(n_s-2)} & a_{(n_s-1)} \\ a_{(n_s-1)} & a_{n_s} & a_1 & a_2 & \cdots & a_{(n_s-4)} & a_{(n_s-3)} & a_{(n_s-2)} \\ a_{(n_s-2)} & a_{(n_s-1)} & a_{n_s} & a_1 & a_2 & \cdots & a_{(n_s-4)} & a_{(n_s-3)} \\ \vdots & \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ a_{(n_s-f+1)} & a_{(n_s-f+2)} & \cdots & a_{n_s} & a_1 & a_2 & \cdots & a_{(n_s-f)} \\ \vdots & \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ a_2 & a_3 & a_4 & \cdots & \cdots & a_{(n_s-1)} & a_{n_s} & a_1 \\ a_1 & a_2 & a_3 & \cdots & \cdots & a_{(n_s-2)} & a_{(n_s-1)} & a_{n_s} \end{bmatrix} \quad (C.2)$$

Notar que si se rota la matriz A un número de filas n_s , la matriz resultante es ella misma (ver última fila de C_A).

Para obtener esta matriz C_A vamos a dividirla en dos submatrices que se construirán por separado.

$$C_A = \begin{bmatrix} C_{A1} & C_{A2} \end{bmatrix} \quad (C.3)$$

La primera submatriz tiene una dimensión $n_s \times n_r$ y está formada por las n_s filas de la matriz A colocadas en orden inverso:

$$C_{A1} = \begin{bmatrix} a_{n_s} \\ a_{(n_s-1)} \\ a_{(n_s-2)} \\ \vdots \\ a_{(n_s-f+1)} \\ \vdots \\ a_2 \\ a_1 \end{bmatrix} \quad (C.4)$$

Mientras que la segunda submatriz tiene una dimensión $n_s \times ((n_s - 1) \cdot n_r)$, de modo que, para construirla, se necesitan $n_s \cdot (n_s - 1)$ filas de la matriz A .

$$C_{A2} = \begin{bmatrix} a_1 & a_2 & \cdots & a_{(n_s-4)} & a_{(n_s-3)} & a_{(n_s-2)} & a_{(n_s-1)} \\ a_{n_s} & a_1 & a_2 & \cdots & a_{(n_s-4)} & a_{(n_s-3)} & a_{(n_s-2)} \\ a_{(n_s-1)} & a_{n_s} & a_1 & a_2 & \cdots & a_{(n_s-4)} & a_{(n_s-3)} \\ \vdots & \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ a_{(n_s-f+2)} & \cdots & a_{n_s} & a_1 & a_2 & \cdots & a_{(n_s-f)} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ a_3 & a_4 & \cdots & \cdots & a_{(n_s-1)} & a_{n_s} & a_1 \\ a_2 & a_3 & \cdots & \cdots & a_{(n_s-2)} & a_{(n_s-1)} & a_{n_s} \end{bmatrix} \quad (C.5)$$

Notar que, si se sigue un orden de lectura de izquierda a derecha y de arriba a abajo, los índices de las filas de A en la matriz C_{A2} siguen una secuencia ordenada que comienza en 1 y termina en n_s para volver a comenzar de nuevo. Esa secuencia se va a repetir $n_s - 1$ veces hasta completar la matriz. La construcción de C_{A2} es fácilmente implementable, ya que la secuencia que siguen las filas es la misma que sigue la matriz A ; luego bastará con colocar sus n_s filas de forma consecutiva $n_s - 1$ veces, de manera que la matriz resultante tenga una dimensión $n_s \times ((n_s - 1) \cdot n_r)$.

Como ya se dijo anteriormente el descriptor B no se va a rotar, de modo que la matriz C_B que comparemos con C_A deberá tener todas sus filas iguales. Cada una de esas n_s filas de C_B (pues ha de tener la misma dimensión que C_A) se puede construir disponiendo las filas del descriptor B de forma consecutiva y en orden ascendente. Por consiguiente, si llamamos c_{Bf} a la fila f -ésima de C_B y b_1, b_2, \dots, b_{n_s} a cada una de las filas del descriptor B , entonces:

$$c_{Bf} = \left[a_1 \quad a_2 \quad a_3 \quad \cdots \quad a_{n_s} \right] \quad (C.6)$$

Esta expresión es válida para $1 \leq f \leq n_s$.

C.2. Implementación para un número reducido de giros sobre la normal

Si se dispone de una estimación k_S del índice asociado al giro sobre la normal y se quiere evaluar la medida de similitud para un intervalo de índices $k_S \pm \Delta k$ (en vez de hacerlo para el número total de giros, n_s) habrá que determinar, en primer lugar, el conjunto de índices de fila del descriptor A con los que hay que construir la matriz C_A .

En este caso, la matriz I va a tener $2 \cdot \Delta k + 1$ filas y $n_s \cdot n_r$ columnas. Por lo tanto, la submatriz C_{A1} tendrá una dimensión $(2 \cdot \Delta k + 1) \times n_r$, mientras que la de la submatriz C_{A2} será $(2 \cdot \Delta k + 1) \times ((n_s - 1) \cdot n_r)$. Será necesario, pues, determinar los N_{I_2} índices de fila que se necesitan para formar C_{A2} , tal que:

$$N_{I_2} = (2 \cdot \Delta k + 1) \cdot (n_s - 1) \quad (\text{C.7})$$

Estos índices se van a obtener a partir de una secuencia que comienza en un número entero n_0 y se va incrementando de uno en uno hasta completar N_{I_2} números. El valor de n_0 se obtiene mediante la siguiente expresión:

$$n_0 = n_s - (k_S - \Delta k) + 2 \quad (\text{C.8})$$

Ésta resulta de sustituir $f = k_S - \Delta k$ en el primer índice de la fila f -ésima de la matriz C_{A2} (Ecuación C.5).

Cada número n_g de dicha secuencia estará dentro del intervalo $[n_0, n_0 + N_{I_2} - 1]$ y tendrá, por lo tanto, que normalizarse al intervalo $[1, n_s]$ para obtener su índice g correspondiente:

$$g = \text{mod}(n_g - 1, n_s) + 1 \quad (\text{C.9})$$

Por otro lado, los $2 \cdot \Delta k + 1$ índices de C_{A1} se obtienen a partir de una secuencia que comienza en $n_0 - 1$ y decrece de uno en uno hasta $(n_0 - 1) - 2 \cdot \Delta k$. Posteriormente, los números n_g de esa secuencia tendrán también que normalizarse mediante la Ecuación C.9.

Una vez que se ha determinado C_A , se construirá la matriz C_B (correspondiente al descriptor B) con $2 \cdot \Delta k + 1$ filas iguales obtenidas mediante la Ecuación C.6.

C.3. Cálculo de la Medida de Similitud

Una de las ventajas de la utilización de matrices que expresan las rotaciones sobre la normal, es que los conjuntos de píxeles solapados y no solapados (I_{AB} , X_{AB} , U_{AB}) para cada giro, se pueden obtener directamente a partir de C_A y C_B siguiendo las mismas reglas de la Sección 5.1 sin tener que utilizar en la implementación ningún bucle "for". Igualmente ocurre con la obtención de la matriz de pesos (Figura 3.7), pues también es de fácil implementación una matriz de pesos total W_{AB} que tenga la misma dimensión que C_A y C_B , de modo que todas sus filas (de $n_s \cdot n_r$ elementos) sean iguales y de la forma:

$$w_{ABf} = [1 \ 2 \ \dots \ n_r \ 1 \ 2 \ \dots \ n_r \ \dots \dots \ 1 \ 2 \ \dots \ n_r] \quad (\text{C.10})$$

De esta manera, las distancias píxel a píxel se pueden obtener, para cualquier giro, con tan sólo calcular el valor absoluto de la resta de las matrices C_A y C_B , es decir:

$$D_{AB} = |C_A - C_B| \quad (\text{C.11})$$

Una vez calculada esta matriz de distancias D_{AB} , obtener para cada giro el porcentaje de solapamiento σ_{ov} y la distancia media D_{ov} en el área solapada es sencillo si se utiliza un lenguaje de programación que proporcione una amplia variedad de operaciones matriciales. Por ejemplo, una sola línea de código de *Matlab* u *Octave* es suficiente para calcular tanto σ_{ov} como D_{ov} cuando se conocen D_{AB} , W_{AB} y los conjuntos intersección y unión (I_{AB} y U_{AB}) de todos los giros. Esto simplifica enormemente el código y evita la utilización de bucles "for" que aumentan el tiempo de computación.

Por consiguiente, el resultado de evaluar la Ecuación 5.18 de forma matricial

es un vector M_S de n_s elementos (ó $2 \cdot \Delta k + 1$), de modo que cada uno de ellos corresponde a la medida de similitud para un determinado giro. El máximo de todos esos valores, $M_{S_{max}}$, será el escogido para la correspondencia $a \leftrightarrow b$.

Para determinar el giro asociado a la medida de similitud, habrá que considerar dos casos: si se evalúan los n_s giros o un número reducido de ellos.

Si estamos en el primer caso y el f -ésimo elemento de M_S es el que contiene el máximo, el giro asociado será de $-f \cdot \rho_\theta$ radianes, es decir $k_a = f$. Si, en cambio, estamos evaluando un número reducido de giros ($2 \cdot \Delta k + 1$) en torno a un índice estimado k_S , la equivalencia anterior ya no es válida, de modo que habrá que calcular el índice k_a que corresponde a esa fila f -ésima. Para ello habrá que utilizar la siguiente expresión:

$$k_a = \text{mod}((f - 1) + (k_S - \Delta k), n_s) \quad (\text{C.12})$$

Por lo tanto, el giro asociado a la correspondencia $a \leftrightarrow b$ será de $-k_a \cdot \rho_\theta$ radianes.

El ángulo obtenido, en cualquiera de los dos casos, va a permitir determinar la matriz de rotación asociada a la medida de similitud, la cual será fundamental para calcular posteriormente la transformación euclídea que alinea de forma aproximada las dos nubes de puntos.

C.4. Compatibilidad de dos correspondencias

Una vez deducida la expresión de la matriz de transformación ${}^{w_2}_{b \leftrightarrow a} T_{w_1}$ asociada a una correspondencia $a \leftrightarrow b$, vamos a estudiar su compatibilidad con otra posible correspondencia $m \leftrightarrow n$.

Si las dos correspondencias fuesen válidas, sus transformaciones euclídeas asociadas deberían ser muy similares, es decir:

$${}^{w_2}_{b \leftrightarrow a} T_{w_1} \approx {}^{w_2}_{n \leftrightarrow m} T_{w_1} \quad (\text{C.13})$$

Como ya se vio en la sección anterior (Ecuación 6.9), ambas transformaciones se pueden expresar en función de tres matrices, que corresponden a las transformaciones entre el s.d.c. global W_2 y el s.d.c. local (${}^b T_{w_2}$ y ${}^n T_{w_2}$), a la ro-

tación sobre la normal (T_{Z_a} y T_{Z_m}) y a la transformación entre el s.d.c. global W_1 y el s.d.c. local (${}^aT_{w_1}$ y ${}^mT_{w_1}$). Por consiguiente:

$${}^bT_{w_2}^{-1} \cdot T_{Z_a} \cdot {}^aT_{w_1} \approx {}^nT_{w_2}^{-1} \cdot T_{Z_m} \cdot {}^mT_{w_1} \quad (C.14)$$

Como la suposición de partida es que la matriz ${}^{w_2}_{b \leftrightarrow a}T_{w_1}$ es conocida, la única matriz por determinar de la Ecuación C.14 es T_{Z_m} , que corresponde a la rotación sobre la normal \vec{n}_m ,

$$T_{Z_m} \approx {}^nT_{w_2} \cdot {}^{w_2}_{b \leftrightarrow a}T_{w_1} \cdot {}^{w_1}T_m \quad (C.15)$$

Expresando cada una de las matrices en función de sus submatrices,

$$\begin{bmatrix} R_{Z_m} & 0_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix} \approx \begin{bmatrix} {}^nR_{w_2} & -{}^nR_{w_2} \cdot {}^{w_2}p_{2n} \\ 0_{1 \times 3} & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^{w_2}_{b \leftrightarrow a}R_{w_1} & {}^{w_2}_{b \leftrightarrow a}p_{w_1org} \\ 0_{1 \times 3} & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^{w_1}R_m & {}^{w_1}p_{1m} \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (C.16)$$

Y efectuando las multiplicaciones, teniendo en cuenta la expresión de ${}^{w_2}_{b \leftrightarrow a}p_{w_1org}$, resulta:

$$\begin{bmatrix} R_{Z_m} & 0_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix} \approx \begin{bmatrix} {}^nR_m & -{}^nR_{w_2} \cdot \delta_{mn} \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (C.17)$$

Siendo,

$${}^nR_m = {}^nR_{w_2} \cdot {}^{w_2}_{b \leftrightarrow a}R_{w_1} \cdot {}^{w_1}R_m \quad (C.18)$$

y

$$\delta_{mn} = {}^{w_2}_{b \leftrightarrow a}R_{w_1} \cdot ({}^{w_1}p_{1m} - {}^{w_1}p_{1a}) - ({}^{w_2}p_{2n} - {}^{w_2}p_{2b}) \quad (C.19)$$

Por consiguiente, se ha de cumplir:

$${}^n R_m = R_{Z_m} \quad (C.20)$$

$$\delta_{mn} = 0_{3 \times 1} \quad (C.21)$$

Estas ecuaciones establecen las dos condiciones que han de cumplir dos correspondencias para que puedan considerarse compatibles:

- La matriz de rotación ${}^n R_m$ debe corresponder solamente a una rotación sobre el eje Z. Es decir, los ángulos de Euler ZYX correspondientes a esta matriz deberán cumplir lo siguiente:

$$\left(\alpha_{mn}, \beta_{mn}, \gamma_{mn} \right) \approx \left(-\hat{k}_m \cdot \rho_\theta, 0, 0 \right) \quad (C.22)$$

- El vector que une los puntos ${}^{w_1}p_{1a}$ y ${}^{w_1}p_{1m}$, expresado en el s.d.c. global W_2 , debe ser aproximadamente igual al que une sus puntos correspondientes ${}^{w_2}p_{2b}$ y ${}^{w_2}p_{2n}$:

$${}^{w_2}_{b \leftrightarrow a} R_{w_1} \cdot ({}^{w_1}p_{1m} - {}^{w_1}p_{1a}) - ({}^{w_2}p_{2n} - {}^{w_2}p_{2b}) \approx 0_{3 \times 1} \quad (C.23)$$

Bibliografía

- [1] The Georgia Tech Large Geometric Models Archive. http://www.cc.gatech.edu/projects/large_models/.
- [2] The Stanford 3D Scanning Repository. <http://www-graphics.stanford.edu/data/3Dscanrep/>.
- [3] K. Brunnström. A. J. Stoddart. Free-form surface matching using mean field theory. In *In British Machine Vision Conference*, pages 33–42, 1996.
- [4] Dragomir Anguelov, Praveen Srinivasan, Hoi cheung Pang, and Daphne Koller. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *NIPS*, pages 33–40, 2004.
- [5] Mihael Ankerst, Gabi Kastenmüller, Hans-Peter Kriegel, and Thomas Seidl. 3d shape histograms for similarity search and classification in spatial databases. In *SSD '99: Proceedings of the 6th International Symposium on Advances in Spatial Databases*, pages 207–226, London, UK, 1999. Springer-Verlag.
- [6] A. P. Ashbrook, R. B. Fisher, C. Robertson, and N. Werghi. Aligning arbitrary surfaces using pairwise geometric histograms. In *Proc. NM-BIA98*, pages 103–108, 1998.
- [7] A. P. Ashbrook, R. B. Fisher, C. Robertson, and N. Werghi. Finding surface correspondence for object recognition and registration using pairwise geometric histograms. In *in Computer Vision-ECCV'98*, pages 674–686, 1998.

- [8] Gill Barequet and Micha Sharir. Partial surface matching by using directed footprints. In *In Proc. 12th Annual Symp. on Computational Geometry*, pages 409–410, 1996.
- [9] Adrien Bartoli and Peter Sturm. Structure-from-motion using lines: representation, triangulation, and bundle adjustment. *Comput. Vis. Image Underst.*, 100(3):416–441, 2005.
- [10] Ronen Basri, David Jacobs, and Ira Kemelmacher. Photometric stereo with general, unknown lighting. *Int. J. Comput. Vision*, 72(3):239–257, 2007.
- [11] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.
- [12] J.-Angelo Beraldin. Integration of laser scanning and close-range photogrammetry – the last decade and beyond. In *IN: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science Congress, Istanbul, Turkey, Commission VII*, pages 972–983, 2004.
- [13] D. Bergmann. New approach for automatic surface reconstruction with coded light. In *Proc. Remote Sensing and Reconstruction for Three-Dimensional Objects and Scenes, SPIE*, volume vol. 2572, pages pp. 2–9, 1995.
- [14] P. J. Besl and H. D. McKay. A method for registration of 3-d shapes. *IEEE Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb. 1992.
- [15] P.J. Besl. Active optical range imaging sensors. *Machine Vision and Applications*, Vol. 1, No. 2:pp. 127–152, 1988.
- [16] Kay Boehnke and Marius Ottesteanu. Progressive mesh based iterative closest points for robotic bin picking. In *ICINCO-RA (2)*, pages 121–124, 2008.

- [17] Faysal Boughorbel, Yan Zhang, Sangkyu Kang, Umayal Chidambaram, Andreas Koschan, and Mongi Abidi. Laser ranging and video imaging for bin picking. *Assembly Automation*, 23:53–59, 2003.
- [18] Benedict J. Brown. *Registration and Matching of Large Geometric Datasets for Cultural Heritage Applications*. PhD thesis, Princeton University, June 2008.
- [19] Nicola Brusco, Marco Andreetto, Andrea Giorgi, and Guido M. Cortelazzo. 3d registration by textured spin-images. In *3DIM '05: Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*, pages 262–269, Washington, DC, USA, 2005. IEEE Computer Society.
- [20] Benjamin Bustos, Daniel A. Keim, Dietmar Saupe, Tobias Schreck, and Dejan V. Vranić. Feature-based similarity search in 3d object databases. *ACM Comput. Surv.*, 37(4):345–387, 2005.
- [21] D. L. Hill C. Studholme and Y. Bizais D. J. Hawkes. Multiresolution voxel similarity measures for mr-pet registration. In *Proc. Information Processing in Medical Imaging*, pages 287–298, 1995.
- [22] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, 1992.
- [23] J.C. Cheng and H.S. Don. A graph matching approach to 3-d point correspondences. *Pattern Recognition and Artificial Intelligence*, 5:399–412, 1991.
- [24] Chi Kin Chow, Hung Tat Tsui, and Tong Lee. Surface registration using a dynamic genetic algorithm. *Pattern Recognition*, 37(1):105 – 117, 2004.
- [25] Chin S. Chua and Ray Jarvis. 3d free-form surface registration and object recognition. *International Journal of Computer Vision*, 17(1):77–99, January 1996.
- [26] Chin S. Chua and Ray Jarvis. Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision*, 25(1):63–85, October 1997.

- [27] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multi-modality image registration based on information theory. *Information Processing in Medical Imaging*, pages 263–274, 1995.
- [28] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
- [29] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, New York, NY, USA, 1996. ACM.
- [30] H. Delingette, Martial Hebert, and Katsushi Ikeuchi. Shape representation and image segmentation using deformable surfaces. *Image and Vision Computing*, pages 132 – 145, 1992.
- [31] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun. Structure from motion without correspondence. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 2000.
- [32] D. Dion, Jr., D. Laurendeau, and R. Bergevin. Generalized cylinders extraction in a range image. In *NRC '97: Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, page 141, Washington, DC, USA, 1997. IEEE Computer Society.
- [33] D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Mach. Vision Appl.*, 9(5-6):272–290, 1997.
- [34] J. Ens and P. Lawrence. An investigation of methods for determining depth from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:97–108, 1993.
- [35] Jacques Feldmar and Nicholas Ayache. Rigid, affine and locally affine registration of free-form surfaces. *Int. J. Comput. Vision*, 18(2):99–119, 1996.

- [36] Brecht Fevery, Bart Wyns, Luc Boullart, J LLATA GARCIA, and C TORRE FERRERO. Active security system for an industrial robot based on artificial vision and fuzzy logic principles. In *ICINCO 2008, fifth international conference on informatics in control, automation and robotics*, 2008.
- [37] Daniel Fischer, Peter Kohlhepp, and Frank Bulling. An evolutionary algorithm for the registration of 3-d surface representations. *Pattern Recognition*, 32(1):53 – 69, 1999.
- [38] Robert B. Fisher and Kurt Konolige. Range sensors. In *Handbook of Robotics*, pages 521–542. Springer Verlag, 2008.
- [39] Andrew W. Fitzgibbon. Robust registration of 2d and 3d point sets. In *In British Machine Vision Conference*, pages 411–420, 2001.
- [40] Ran Gal and Daniel Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph.*, 25(1):130–150, 2006.
- [41] Natasha Gelfand, Niloy J. Mitra, Leonidas J. Guibas, and Helmut Pottmann. Robust global registration. In *Symposium on Geometry Processing*, 2005.
- [42] Javier González-Jiménez. *Visión por computador*. Paraninfo.
- [43] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [44] Y.C. Hecker and R.M. Bolle. On geometric hashing and the generalized hough transform. *IEEE Transactions on Systems, Man and Cybernetics*, 24(9):1328–1338, September 1994.
- [45] Janne Heikkila and Olli Silven. A four-step camera calibration procedure with implicit image correction. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1106, 1997.

- [46] K. Higuchi, Martial Hebert, and Katsushi Ikeuchi. Building 3d models from unregistered range images. In *Proceedings of IEEE Conference on Robotics and Automation (ICRA '94)*, volume 3, pages 2248 – 2253, May 1994.
- [47] Donald D. Hoffman and Manish Singh. Saliency of visual parts. *Cognition*, 63(1):29 – 78, 1997.
- [48] Berthold K. P. Horn. Extended gaussian images. *Proceedings of the IEEE*, 72(2):1671–1686, 1984.
- [49] B.K.P. Horn. *Robot Vision*. MIT Press, 1986.
- [50] Daniel F. Huber and Martial Hebert. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing*, 21:637–650, 2001.
- [51] Katsushi Ikeuchi, Berthold K.P. Horn, Shigemi Nagata, Tom, Tom Callahan, and Oded Feingold. Picking up an object from a pile of objects. In *Proceedings of the First International Symposium on Robotics Research*, pages 139–166. MIT Press, 1983.
- [52] A. E. Johnson and M. Hebert. Surface registration by matching oriented points. In *Proc. International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pages 121–128, 12–15 May 1997.
- [53] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Pattern Analysis and Machine Intelligence*, 21(5):433–449, May 1999.
- [54] Andrew Johnson and Sing Bing Kang. Registration and integration of textured 3-d data. In *Image and Vision Computing*, pages 234–241, 1996.
- [55] A. Stoddart. K. Brunnstrom. Genetic algorithms for free-form surface matching. In *Proceedings of the International Conference on Pattern Recognition (ICPR '96) Volume IV-Volume 7472, p.689, August 25-29, 1996*, pages 689–693, 1996.

- [56] G. Hetzel. K. Eisele. Range Image Database, University of Stuttgart. <http://range.informatik.uni-stuttgart.de/htdocs/html/>.
- [57] Avinash C. Kak and L. Edwards. Experimental state of the art in 3d object recognition and localization using range data. In *Proceedings of the workshop on vision for robots in IROS1995*, pages 45–54, 1995.
- [58] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [59] Jan J. Koenderink and Andrea J. van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10(8):557–564, October 1992.
- [60] Shankar Krishnan, Pei Yean Lee, John B. Moore, and Suresh Venkatasubramanian. Global registration of multiple 3d point sets via optimization-on-a-manifold. In *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*, pages 187–196, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.
- [61] Pavel Krsek, Tomáš Pajdla, Václav Hlavác, and Ralph Martin. Range image registration driven by a hierarchy of surfaces. In *22nd Workshop of the Austrian Association for Pattern Recognition*, pages 175–183, 1998.
- [62] Marcel Körtgen, Marcin Novotni, and Reinhard Klein. 3d shape matching with 3d shape contexts. In *In The 7th Central European Seminar on Computer Graphics*, 2003.
- [63] M. Lindner, A. Kolb, and T. Ringbeck. New insights into the calibration of tof sensors. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Workshop on ToF Camera based Computer Vision (TOF-CV)*, 2008.
- [64] J. R. Llata, E. G. Sarabia, and J. P. Oria. Three-dimensional robotic vision using ultrasonic sensors. *J. Intell. Robotics Syst.*, 33(3):267–284, 2002.

- [65] Jose R. Llata, Carlos Torre-Ferrero, Esther G. Sarabia, and Sandra Robla. Optical-ultrasound data processing for hybrid 3d vision on air. In *ASMCCS'09: Proceedings of the 3rd International Conference on Applied Mathematics, Simulation, Modelling, Circuits, Systems and Signals*, pages 181–186, Stevens Point, Wisconsin, USA, 2009. World Scientific and Engineering Academy and Society (WSEAS).
- [66] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.
- [67] Frederik Maes, André Collignon, Dirk V, Guy Marchal, and Paul Suetens. Multimodality image registration by maximization of mutual information. *IEEE transactions on Medical Imaging*, 16:187–198, 1997.
- [68] P. C. Mahalanobis. On the generalised distance in statistics. In *Proceedings National Institute of Science, India*, volume 2, pages 49–55, April 1936.
- [69] Ameesh Makadia, Er Patterson Iv, and Kostas Daniilidis. Fully automatic registration of 3d point clouds. In *in CVPR'06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1297–1304, 2006.
- [70] S. Malassiotis and M. G. Strintzis. Snapshots: A novel local surface descriptor and matching algorithm for robust 3d surface alignment. *IEEE Pattern Analysis and Machine Intelligence*, 29(7):1285–1290, July 2007.
- [71] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.
- [72] T. Masuda. Automatic registration of multiple range images by the local log-polar range images. In *Proc. Third International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 216–223, 14–16 June 2006.

- [73] Takeshi Masuda. Registration and integration of multiple range images by matching signed distance fields for object shape modeling. *Comput. Vis. Image Underst.*, 87(1-3):51–65, 2002.
- [74] Carles Matabosch. Range Image Registration Toolbox. "<http://eia.udg.es/%7Ecatabos/ToolboxRegistration.rar>".
- [75] L. Matthies and Steven Shafer. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, RA-3(1):239 – 250, June 1987.
- [76] C.R. Maurer, G.B. Aboutanos, B.M. Dawant, R.J. Maciunas, and J.M. Fitzpatrick. Registration of 3-d images using weighted geometrical features. 15(6):836–849, December 1996.
- [77] Yves Meyer. *Wavelets and operators*, volume 37 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1992. Translated from the 1990 French original by D. H. Salinger.
- [78] A. S. Mian, M. Bennamoun, and R. A. Owens. A novel representation and feature matching algorithm for automatic pairwise registration of range images. *International Journal of Computer Vision*, 66:19–40, 2006.
- [79] Niloy J. Mitra, Natasha Gelfand, Helmut Pottmann, and Leonidas Guibas. Registration of point cloud data from a geometric optimization perspective. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIG-GRAPH symposium on Geometry processing*, pages 22–31, New York, NY, USA, 2004. ACM.
- [80] Gonzalo Pajares and Jesús M. de la Cruz. *Visión por Computador. Imágenes digitales y aplicaciones*. RA-MA Editorial, 2001.
- [81] Georgios Papaioannou, Evaggelia-Aggeliki Karabassi, and Theoharis Theoharis. Virtual archaeologist: Assembling the past. *IEEE Computer Graphics and Applications*, 21:53–59, 2001.

- [82] G.P. Penney, J. Weese, J.A. Little, P. Desmedt, D.L.G. Hill, and D.J. Hawkes. A comparison of similarity measures for use in 2-D-3-D medical image registration. *IEEE Transactions on Medical Imaging*, 17(4):586–595, 1998.
- [83] B. M. Planitz, A. J. Maeder, and J. A. Williams. The correspondence framework for 3d surface matching algorithms. *Comput. Vis. Image Underst.*, 97(3):347–383, 2005.
- [84] E. Prados, F. Camilli, and O.D. Faugeras. A unifying and rigorous shape from shading method adapted to realistic data and applications. *Journal of Mathematical Imaging and Vision*, 25(3):307–328, October 2006.
- [85] Kari Pulli. Multiview registration for large data sets. pages 160–168, Los Alamitos, CA, USA, 1999. IEEE Computer Society.
- [86] M. Rioux. Laser range finder based on synchronized scanners. *Applied Optics*, 23(21):3837–3855, November 1984.
- [87] Alexis Roche, Grégoire Malandain, Xavier Pennec, and Nicholas Ayache. The correlation ratio as a new similarity measure for multimodal image registration. pages 1115–1124. Springer Verlag, 1998.
- [88] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pages 145–152, 2001.
- [89] Joaquim Salvi, Carles Matabosch, David Fofi, and Josep Forest. A review of recent range image registration methods with accuracy evaluation. *Image Vision Comput.*, 25(5):578–596, 2007.
- [90] E. G. Sarabia, J. R. Llata, C. Torre, S. Robla, and J. P. Oria. Pose recognition by using neural networks and ultrasounds for a robotic picking up application. In *Proc. CIMSA Computational Intelligence for Measurement Systems and Applications 2005 IEEE International Conference on*, pages 128–133, 20–22 July 2005.

- [91] E.G. Sarabia, J.R. Llata, J. Arce, and J.P. Oria. Shape recognition and orientation detection for industrial applications using ultrasonic sensors. *IEEE International Joint Symposia on Intelligence and Systems*, 0:301, 1998.
- [92] Anshul Sehgal and U. B. Desai. 3d object recognition using bayesian geometric hashing and pose clustering. *Pattern Recognition*, 36(3):765 – 780, 2003.
- [93] G.C. Sharp, S.W. Lee, and D.K. Wehe. Multiview registration of 3d scenes by minimizing error between coordinate frames. *PAMI*, 26(8):1037–1050, August 2004.
- [94] Luciano Silva, Olga R.P. Bellon, and Kim L. Boyer. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:762–776, 2005.
- [95] D. Skerl, B. Likar, and F. Pernus. A protocol for evaluation of similarity measures for rigid registration. *IEEE Transactions on Medical Imaging*, 25(6):779–791, June 2006.
- [96] Chu song Chen, Yi ping Hung, and Jen bo Cheng. Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:1229–1234, 1999.
- [97] Fridtjof Stein and Gérard Medioni. Structural indexing: Efficient 2d object recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(12):1198–1204, 1992.
- [98] G.C. Stockman. Object recognition and localization via pose clustering. 40(3):361–387, December 1987.
- [99] C. Studholme, D.L.G. Hill, and D.J. Hawkes. An overlap invariant entropy measure of 3d medical image alignment. *Pattern Recognition*, 32(1):71–86, January 1999.

- [100] P. Sturm, S. Ramalingam, and S. Lodha. On calibration, structure from motion and multi-view geometry for generic camera models. In Kostas Daniilidis and Reinhard Klette, editors, *Imaging Beyond the Pinhole Camera*, pages 87–105. Springer Netherlands, 2006.
- [101] P.F. Sturm and S. Ramalingam. A generic concept for camera calibration. pages Vol II: 1–13, 2004.
- [102] C. Torre Ferrero, J.R. Llata, S. Robla, and E.G. Sarabia. A similarity measure for 3d rigid registration of point clouds using image-based descriptors with low overlap. In *S3DV09, IEEE 12th International Conference on Computer Vision, ICCV Workshops 2009*, pages 71–78, 2009.
- [103] C. Torre-Ferrero, S. Robla, E. G. Sarabia, and J. R. Llata. Circon: An alternative 3d shape representation for surface matching. In *Proc. IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, pages 3320–3323, 6–10 Nov. 2006.
- [104] C. Torre-Ferrero, S. Robla, E. G. Sarabia, and J. R. Llata. 3d registration by using an alternative 3d shape representation. In *ISCGAV'07: Proceedings of the 7th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision*, pages 101–106, Stevens Point, Wisconsin, USA, 2007. World Scientific and Engineering Academy and Society (WSEAS).
- [105] C. Torre-Ferrero, S. Robla, E. G. Sarabia, and J. R. Llata. A coarse-to-fine algorithm for 3d registration based on wavelet decomposition. *WTOS*, 7(7):655–664, 2008.
- [106] Carlos Torre-Ferrero, J. R. Llata, A. G. Saro, and E. G. Sarabia. Fusión de datos para reconstrucción 3D mediante un sistema híbrido de visión. In *Proceedings de las XXV Jornadas de Automática de CEA-IFAC*, 2004.
- [107] Carlos Torre-Ferrero and J.R. Llata. Improving precision of a hybrid 3D vision system by using ICP registration. In *Proceedings of the IADAT-*

- aci2005 International Conference on Automation, Control and Instrumentation (Bilbao, SPAIN)*, 2005.
- [108] P. S. Tsai and M. Shah. Shape from shading using linear-approximation. *IVC*, 12(8):487–498, October 1994.
- [109] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. 3(4):323–344, August 1987.
- [110] Ranjith Unnikrishnan and Martial Hebert. Multi-scale interest regions from unorganized point clouds. In *Workshop on Search in 3D (S3D), IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2008.
- [111] William M. Wells-III, Paul Viola, Hideki Atsumi, Shin Nakajima, and Ron Kikinis. Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1):35 – 51, 1996.
- [112] Haim Wolfson and Isidore Rigoutsos. Geometric hashing: An overview. *IEEE Comput. Sci. Eng.*, 4(4):10–21, 1997.
- [113] R. P Woods, J. C. Mazziotta, and S. R. Cherry. Mri-pet registration with automated algorithm. *Journal of Computer Assisted Tomography*, 17(4):536–546, 1993.
- [114] Joris Vanden Wyngaerd, Reinhard Koch, Marc Proesmans, and Luc Van Gool. Invariant-based registration of surface patches. *IEEE International Conference on Computer Vision*,, 1:301, 1999.
- [115] S. M. Yamany and A. A. Farag. Surface signatures: an orientation independent free-form surface representation scheme for the purpose of objects registration and matching. *IEEE Pattern Analysis and Machine Intelligence*, 24(8):1105–1120, Aug. 2002.
- [116] Dongmei Zhang. Harmonic shape images: A 3-d free-form surface representation and its application in surface matching. In *Ph.D. dissertation, Carnegie Mellon Univ*, 1999.

BIBLIOGRAFÍA

- [117] Timo Zinsser, Jochen Schmidt, and Heinrich Niemann. A refined icp algorithm for robust 3-d correspondence estimation. In *Proceedings of the International Conference on Image Processing*, pages 695–698, 2003.

