

Universitat Jaume I

Departamento de Informática



Tesis Doctoral

***Razonamiento Geométrico Basado en
Visión para la Determinación y Ejecución
del Agarre en Robots Manipuladores***

Pedro José Sanz Valero

Castellón, 1996

Directores: Ángel Pasqual del Pobil y Ferré & José Manuel Iñesta Quereda

A Victoria

A Carlos, Marta y Jorge

RESUMEN

(Consejo de Universidades)

El objetivo central de la tesis aquí presentada ha sido demostrar la viabilidad de un sistema completo para la determinación y ejecución del agarre en robots manipuladores mediante razonamiento geométrico basado en visión. Dicho sistema actúa en un entorno real, demostrando robustez, eficiencia y economía de medios, haciendo uso del principio "percepción-razonamiento-acción" inherente a la investigación en el campo de la robótica inteligente.

En síntesis, se presenta un sistema capaz de coordinar una pinza de dedos plano-paralelos y un sistema de visión con el objetivo de determinar puntos de agarre, con existencia de rozamiento, bajo condiciones de estabilidad, y la subsiguiente ejecución del mismo. La elección de puntos de agarre se realiza mediante una nueva heurística que trata de manejar todo tipo de objetos planos con centroide exterior o interior, permitiendo agujeros, concavidades, etc. Sólo se requieren dos características basadas en los momentos, centroide y dirección del eje principal, obtenidos a partir de una representación del código de cadena de Freeman del contorno. Los resultados son parámetros visuales exclusivamente, expresados en coordenadas visuales (píxeles).

Los algoritmos desarrollados se han implementado en un robot manipulador con 4 GDL y fijando una cámara CCD en su muñeca. El sistema de control utiliza realimentación visual, consiguiendo incrementar la precisión hasta el límite del sistema de visión utilizado.

Resumen

Se presenta un sistema capaz de coordinar una pinza de dedos plano-paralelos y un sistema de visión con el objetivo de determinar puntos de agarre bajo condiciones de estabilidad.

La elección de puntos de agarre se realiza mediante una nueva heurística que trata de manejar todo tipo de objetos planos con centroide exterior o interior, permitiendo agujeros. Sólo se requieren dos características basadas en los momentos, centroide y dirección del eje principal, obtenidos a partir de una representación del código de cadena de Freeman del contorno. Los resultados son parámetros visuales exclusivamente, expresados en coordenadas visuales (píxeles).

Dichos algoritmos se han implementado en un robot manipulador con 4 GDL y fijando una cámara CCD en su muñeca. El sistema de control utiliza realimentación visual, consiguiendo incrementar la precisión hasta el límite del sistema de visión utilizado.

Abstract

A system is presented which coordinates a parallel-jaw gripper and a vision system with the aim of choosing grasping points taking into account stability conditions.

The choice of grasping points is done by a new heuristic approach that tries to manage all kind of planar parts with or without internal centre of gravity, possibly containing holes. It needs only two moment-based features, centroid and the direction of the main axis, easily obtained from a Freeman chain-code representation of the boundary. The results are exclusively visual parameters, expressed in visual coordinates (pixels).

These algorithms have been implemented in a 4DOF robot arm with one camera fixed over the hand. Visual feedback was used in the control system to improve precision up to the limits of the available visual system.

Agradecimientos

Quisiera, en primer lugar, reconocer a mis codirectores: Ángel Pasqual del Pobil y José Manuel Iñesta la dedicación y aliento prestados en todo momento, cuya guía constante ha hecho posible que este proyecto llegara a término en un plazo razonable de tiempo, con garantías de éxito.

Agradecer la "savia nueva" aportada por investigadores de prestigio como Radu Horaud, Emmanuel Mazer, Kamal Gupta y, sobre todo Juan Manuel Ahuactzin por su ayuda incondicional en cualquier circunstancia, en el centro de investigación perteneciente al "Institut National de Recherche en Informatique et Automatique" (INRIA): LIFIA-IMAG (Grenoble, Francia), durante una estancia realizada en 1994.

Agradecer así mismo, la colaboración prestada por el Instituto de Robótica de Valencia, y muy especialmente a Juan de Mata Domingo, quien en todo momento se brindó a facilitar todos los recursos necesarios para acometer la implementación final de los algoritmos desarrollados.

A mis antiguos compañeros del grupo de investigación en robótica y visión del Departamento de Ingeniería de Sistemas y Automática (DISCA), perteneciente a la Universidad Politécnica de Valencia (UPV), y muy especialmente a Ángel Rodas con quien inicié mi andadura en el campo del tratamiento digital de imágenes.

Finalmente, un reconocimiento entrañable a todos mis colegas del Departamento de Informática por su inestimable apoyo moral y técnico, así como a las personas de administración y servicios del mismo.

Esta tesis doctoral ha sido desarrollada en el ámbito del Laboratorio de Robótica Inteligente de la Universidad Jaume-I de Castellón. Este laboratorio está financiado en parte por los proyectos de investigación de la Comisión Interministerial de Ciencia y Tecnología (CICYT): TAP92-0391-C02-0 y TAP95-0710; de la Fundació Caixa-Castelló: B-41-IN, A-36-IN, y P1A94-22; y de la Generalitat Valenciana: GV-2214/94.

Índice

Capítulo I. Introducción	1
1.1 Planteamiento del problema: motivación	1
1.2 Ubicación en el contexto de la robótica	8
1.3 Objetivos de la tesis	14
1.4 Organización de la tesis	16
Capítulo II. Antecedentes	17
2.1 El agarre y la percepción visual	17
2.1.1 Introducción	17
2.1.2 Hacia la manipulación inteligente	20
2.1.3 Perspectiva actual	25
2.2 La generación del agarre	28
2.3 La ejecución del agarre	30
Capítulo III. Caracterización del Agarre	35
3.1 Introducción	35
3.2 Condiciones iniciales	37
3.2.1 Herramienta	37
3.2.2 Sensorización	38
3.2.3 Universo de trabajo	42
3.3 Modelo de agarre elegido.....	44
3.3.1 El contacto entre dos cuerpos	45
3.3.2 Acerca de la estabilidad	52
3.3.3 Configuración del agarre	57
3.4 Conclusiones preliminares	59
Capítulo IV. Información Visual : Adquisición y Análisis	60
4.1 Introducción	60
4.2 Adquisición de la imagen	62
4.2.1 El esquema de iluminación empleado	63
4.2.2 Preprocesado	66
4.3 Segmentación de la imagen	68
4.3.1 Determinación del umbral óptimo	69
4.3.2 Obtención de una imagen pseudo-binaria	70
4.4 Extracción de contornos	74
4.5 Representación eficiente de los contornos	79

4.5.1 Justificación de la elección del 3-muestreo	79
4.5.2 Impacto del 3-muestreo en el rendimiento de los algoritmos	84
4.6 Computación eficiente de los momentos	85
4.7 Extracción de características basadas en los momentos	90
4.8 Especificaciones del algoritmo	94
4.9 Conclusiones preliminares	95

Capítulo V. Determinación Automática de Puntos de Agarre97

5.1 Introducción	97
5.2 Descripción del algoritmo	98
5.3 El algoritmo a bajo nivel	100
5.3.1 Obtención de candidatos a puntos de agarre	100
5.3.2 Mecanismo supervisor	104
5.3.2.1 Acerca de la simetría	105
5.3.2.2 Análisis global vs análisis local	109
5.3.2.3 Umbral de curvatura	118
5.3.2.4 Umbral angular	119
5.3.2.5 Umbral de distancia	120
5.3.2.6 Estrategia adoptada	121
5.3.3 Condiciones de seguridad	126
5.4 Resultados	127
5.4.1 Evaluación de la simetría	127
5.4.2 Ejemplos con solución	128
5.4.3 Ejemplos sin solución	132
5.4.4 Invariancia del agarre obtenido	138
5.5 Conclusiones preliminares	144

Capítulo VI. La Ejecución del Agarre145

6.1 Introducción	145
6.2 Estrategia de control empleada	147
6.3 Descripción del sistema	149
6.3.1 Proceso "off-line"	152
6.3.2 Proceso "on-line"	154
6.4 Resultados	156
6.5 Conclusiones preliminares	161

Capítulo VII. Conclusiones y Trabajo Futuro162

7.1 Aportaciones del presente trabajo	162
7.2 Líneas de investigación futura	163

Referencias166

Apéndices185

Apéndice A : La función <i>atan2</i>	185
Apéndice B : La pinza del Robot "UMI RT 100"	187
Apéndice C : Una muestra del universo de trabajo	189
Apéndice D : Acerca de los vectores dinámicos	191

Motivación

La presente tesis doctoral ha sido desarrollada en el ámbito del Laboratorio de Robótica Inteligente de la Universidad Jaume-I de Castellón. Sus objetivos, por tanto, se enmarcan dentro de las líneas y proyectos de investigación de este laboratorio que se ocupan de los diversos aspectos que se refieren al desarrollo de sistemas dotados de inteligencia robótica.

Brady (Oxford Univ.) define, en 1985, la Robótica inteligente como: "la conexión inteligente entre percepción y acción"; en 1992 Winston (director del Laboratorio de Inteligencia Artificial del MIT) cambia su definición de Inteligencia Artificial (IA) en la última edición de su libro de texto y dice que es: "el estudio de las computaciones que permiten percibir, razonar y actuar". Ambas definiciones coinciden en su raíz: asociar la inteligencia a procesos que incorporan en sus mecanismos la cadena percepción-acción. Dentro de este esquema general, este trabajo se planteó inicialmente con un objetivo ambicioso: el desarrollo de un sistema que cubriera las tres partes de la definición anterior, es decir la percepción visual, el razonamiento geométrico y la acción de agarre.

Este objetivo inicial se ha satisfecho mediante el trabajo realizado, dentro naturalmente de los límites que impone el desarrollo de una tesis doctoral. Resumiendo al máximo se puede decir que el sistema resultante es capaz de responder a órdenes del tipo "coge ese objeto", de manera que el robot percibe mediante visión artificial un objeto desconocido cuya posición ignora, analiza su forma y decide cómo agarrarlo, evaluando y corrigiendo las posiciones de los contactos de la pinza sobre el contorno del objeto, mediante un mecanismo supervisor que analiza en tiempo de ejecución las condiciones necesarias para garantizar la estabilidad de dicho agarre, para finalmente ejecutar la acción de cogerlo.

Todo ello además, minimizando tanto los costes computacionales como económicos. La filosofía de trabajo planteada inicialmente fue la de reducir al mínimo los tiempos de computación, así como el coste de los equipos utilizados. De esta forma se garantiza la viabilidad de futuras aplicaciones reales con costes razonables. Esta ha sido una preocupación constante a lo largo del desarrollo de la tesis cuyos resultados en este sentido han sido más que satisfactorios: el sistema ofrece unos tiempos de cálculo del orden de 3 segundos usando un PC 486 convencional y un robot de bajo coste. Los algoritmos desarrollados están, así pues, inducidos por estos requerimientos de tiempo mínimo, obligando a implantar estrategias de reducción de información para aliviar la carga de los mismos, en aquellas etapas que lo han permitido, primando la eficiencia y la robustez sobre cualquier otra consideración.

Teniendo en cuenta además, los objetivos adicionales de preservar las características de que el sistema sea: completo, es decir, contemple todo el

procesamiento de información necesario para incluir la cadena "percepción-razonamiento-acción", definida anteriormente; trabaje en un entorno real, con todas las dificultades inherentes al mismo (existencia de ruido, incertidumbre, etc.); sea robusto en cuanto a su funcionamiento relativo a cambios de diferentes propiedades en su entorno (pequeños cambios de iluminación, variabilidad en la forma de los objetos y su localización en la escena, etc.); sea eficiente, primando las estrategias que mejor se aproximen a las necesidades de tiempo real del sistema; y finalmente, simple, favoreciendo el principio general de economía de medios para resolver un problema, tanto a nivel computacional como económico.

En síntesis, se presenta un sistema capaz de coordinar una pinza de dedos plano-paralelos y un sistema de visión con el objetivo de determinar puntos de agarre, con existencia de rozamiento, bajo condiciones de estabilidad, y la subsiguiente ejecución del mismo. La elección de puntos de agarre se realiza mediante una nueva heurística que trata de manejar todo tipo de objetos planos con centroide exterior o interior, permitiendo agujeros, concavidades, etc. Sólo se requieren dos características basadas en los momentos, centroide y dirección del eje principal, obtenidos a partir de una representación del código de cadena de Freeman del contorno. Los resultados son parámetros visuales exclusivamente, expresados en coordenadas visuales (píxeles).

Cabe destacar, por último, que existen pocos sistemas integrados similares al presentado en esta tesis que aborden el problema del agarre en todas sus etapas y, prácticamente ninguno (conocido por el autor de esta tesis) que lo haga con el principio de economía de medios aquí empleado; ya que la mayoría de los trabajos de investigación en estos temas ofrecen una visión parcial (sesgada, a veces) del problema, centrándose, exclusivamente, en un aspecto concreto del mismo: sea la percepción visual, la caracterización del agarre, el control con realimentación visual, etc.

Capítulo I

Introducción

En este capítulo se ofrece una visión global del problema desarrollado: motivación, contexto, objetivos y organización general de la tesis.

1.1. Planteamiento del problema: motivación

¿Cuáles son las ventajas y limitaciones asociadas a los sistemas robotizados dotados de visión en cuanto a tareas de manipulación se refiere?. Durante el curso 89-90, me enfrenté por primera vez a éste problema, mientras realizaba la tesina del Master de CAD/CAM, en la Célula de Robótica perteneciente al Departamento de Ingeniería de Sistemas, Computadores y Automática (DISCA), de la Univ. Politécnica de Valencia [Sanz, 90]. El problema a resolver consistía en la instalación, puesta a punto y aplicación de un sistema de visión industrial, integrado en el robot IRB 1000 de ASEA.

En el agarre "ciego" se requiere un conocimiento absoluto previo a la acción, es decir, se trata de un sistema completamente determinista. El más mínimo error de posicionamiento, tanto de la pieza como del robot, llevará al fracaso en la acción del agarre. Por contra, dotando al sistema de una cámara y un software para el tratamiento de imágenes, se consigue capturar (en tiempo de ejecución) la localización (posición más orientación) de cualquier objeto presente en el área de trabajo. En nuestro caso además, el sistema de visión disponía de un sistema de reconocimiento automático de formas, de tal modo que habiendo almacenado pre-

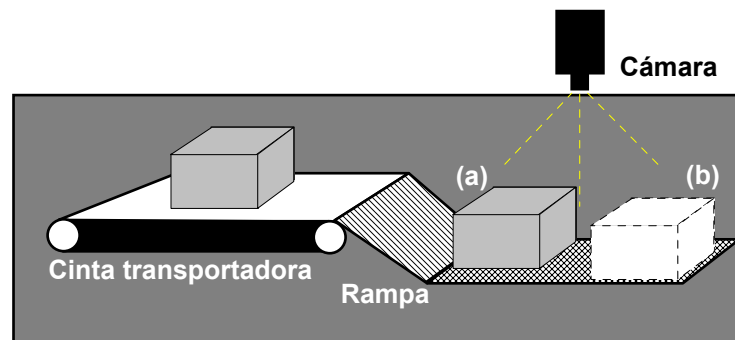


Figura 1.1. La alimentación de piezas a partir de una cinta transportadora se realiza mediante un plano inclinado (Rampa), que provoca a veces ligeros deslizamientos de (a) a (b). Esto puede inducir a capturar la imagen en (a) y disparar la ejecución del agarre hacia una localización inexistente, puesto que el objeto estará en (b) realmente.

viamente en memoria ciertos descriptores (compacidad, momentos de primer orden, etc.) asociados a dichas formas, creando una base de modelos, el sistema era capaz de reconocer los objetos que aparecieran en su área de trabajo, siempre y cuando correspondieran a alguno de los modelos previamente creados al efecto. Dichos modelos incorporaban además, información sobre la matriz de agarre asociada. En teoría, el sistema de visión empleado permitía, para una colección reducida de objetos fijada de antemano, el agarre y posterior procesado de cada pieza.

En dicho sistema, el determinismo se restringía al universo de trabajo, pero podemos hablar ya de una respuesta dinámica y flexible (pseudo-inteligente), ante un determinado estímulo: la entrada del objeto en la escena. El sistema será capaz de coger el objeto, con independencia de su localización dentro de la escena, lo que implica infinitas orientaciones y posiciones posibles (al menos en teoría). Podemos hablar así de una cierta percepción visual, dotando al robot del sentido artificial de la visión.

¿Qué limitaciones subsisten en un sistema de visión como el anteriormente descrito?. En primer lugar, se requiere una fase previa (de entrenamiento), para la creación de la base de modelos. Esta etapa es bastante laboriosa, pues incluye, además de la extracción de descriptores asociados a los modelos para el reconocimiento posterior, el cálculo de la matriz de agarre asociada a los mismos, lo cual requiere llevar físicamente la garra del robot a una serie de posiciones preestablecidas en torno al modelo, que facilitan su cálculo de modo automático.

La primera etapa del procesado de imagen será un preprocesado previo controlado de forma manual por el usuario en cada caso, dirigido a la binarización de la imagen con el objetivo de extraer los contornos de los objetos del modo más

eficiente posible. Este procedimiento requiere la segmentación mediante umbralización global de la imagen, y la única forma de llevar esto a cabo garantizando su funcionamiento automático es controlando las condiciones iniciales de iluminación. Lo más efectivo: una mesa de metacrilato iluminada interiormente, que ofrece imágenes de los objetos prácticamente binarizados en tiempo real.

En resumen, se tiene una base de modelos a priori, y cualquier objeto que aparezca en escena que no se corresponda con ellos será intratable. Por supuesto, se consideran únicamente escenas estáticas, sin solapamientos ni contacto entre objetos, etc.

Otro tipo de limitación a tener en cuenta en este sistema proviene de la ausencia total de realimentación visual en el sistema de control, lo cual se traduce en que un ligero desplazamiento entre la fase de captura de imagen y la acción correspondiente podría ser un fracaso, pues la mano del robot se posicionaría en una localización inexistente del objeto a manipular. Lo que en realidad ocurre es lo siguiente: el programa de control del manipulador incorpora la información visual en una estructura de bucle de manera que mientras no se identifique un determinado objeto de la escena no se abandona dicho bucle y prosigue 1º) {capturando-localizando-comparando}; pero en cuanto el sistema reconoce al objeto (identificación satisfactoria con la base de modelos creada previamente), se abandona dicho bucle y se pasa a 2º) {ejecutar el agarre}, haciendo uso de la caracterización del mismo incorporada al modelo previamente. La cuestión de qué ocurre si el objeto se desplaza al salir de 1º) y antes de resolverse 2º) queda explicada en la figura 1.1.

Hay que hacer notar que los algoritmos subyacentes de procesamiento de imágenes del sistema de visión comentado, al igual que muchos otros de incipiente comercialización en el entorno de la robótica industrial, tienen su inspiración en los desarrollados en Stanford en la década de los setenta, conocidos como "*SRI vision system*" [Agin & Duda, 75]. Volveremos a incidir sobre ello en el apartado 2.1.

¿En qué medida se pueden relajar algunas de las restricciones anteriores?. Desde mi punto de vista, podemos fijar la atención sobre los siguientes objetivos alcanzados, en mayor o menor medida, por el presente trabajo:

- Relajamos las condiciones de iluminación. No empleamos iluminación posterior, sino luz difusa, típica de ambientes interiores, donde pueden presentarse tanto objetos oscuros sobre fondo claro como al contrario. Es decir, la hipótesis se relaja a tener imágenes cuya distribución en niveles de gris sea aproximadamente bimodal.
- Se permite trabajar sobre objetos desconocidos. No nos preocupa la fase de reconocimiento citada anteriormente. La idea central es poder controlar el agarre sin conocimiento previo acerca de la geometría del

objeto a manipular. Todo lo necesario para dicho agarre será deducido en tiempo de ejecución, a partir de la forma (contorno) de dicho objeto.

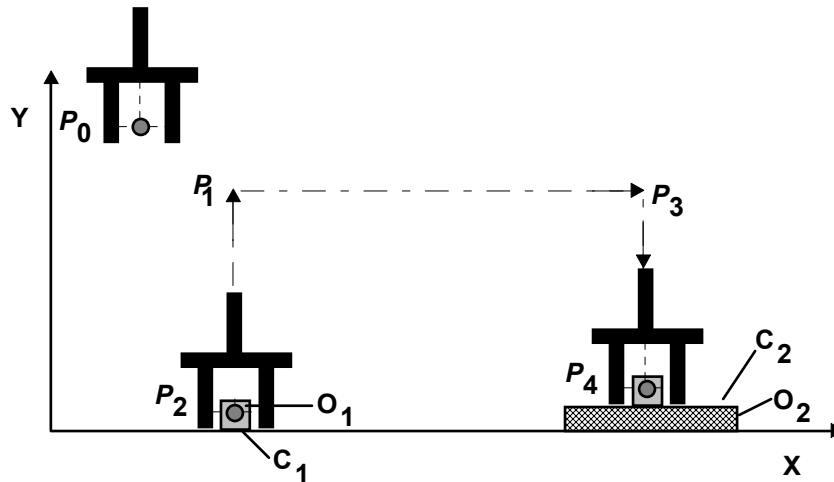
- La respuesta conjunta del sistema de visión y el módulo de determinación de puntos de agarre (coordenadas en píxeles de los puntos de agarre del objeto) sirve de entrada a un módulo de control que, utilizando ésta como realimentación visual, permitirá llevar la mano del robot al objetivo con muy buena aproximación y sin invertir esfuerzo en construir modelos cinemáticos precisos, ni laboriosos calibrados previos. El uso de la realimentación visual en el sistema de control será la clave para alcanzar dicho objetivo.

En líneas generales, la filosofía de fondo del presente trabajo es la de avanzar en la integración sensorial con vistas a facilitar el agarre por un manipulador industrial suficientemente genérico: nos referimos básicamente a los que poseen una o varias pinzas de dos dedos plano-paralelos, dedicados a tareas de ensamblado y manipulación en entornos industriales, principalmente. Este incremento en la flexibilidad de las tareas de manipulación intersecta directamente con los intereses de los planificadores de ensamblado [Sanz, 94], y en general con las necesidades de los denominados lenguajes de programación a nivel tarea¹ [Wolter et al., 85].

Siguiendo trabajos previos [Groover & Mikell, 86], podemos decir que el concepto de robot inteligente incluye la capacidad de recibir instrucciones de alto nivel expresadas como "comandos" para realizar una tarea general, trasladando dichas instrucciones a un conjunto de acciones que deben ejecutarse para llevar a cabo dicha tarea. Será consciente de su entorno y capaz de tomar decisiones acerca de sus acciones basadas, en parte, en la interpretación de dicho entorno.

Un ejemplo sencillo que pone de manifiesto el significado de lo dicho, puede observarse en la figura 1.2, donde se representan dos alternativas para programar una tarea sencilla de coge y deja. La primera, a la izquierda, representa la programación usual, denominada de "nivel robot", que requiere especificar cada una de las acciones del mismo (programación explícita), e.g. cada posición distinta a alcanzar es una acción diferente a tener en cuenta. Si se requiere encapsular diferentes movimientos en unas pocas acciones, estamos obligados a integrar información sensorial que permita disparar de forma automática dichas acciones, en nuestro caso, de manipulación (parte derecha de la figura), lo cual necesita, además, de una representación del modelo del mundo (programación orientada al modelo o implícita). El tipo de declaraciones simbólicas:

¹ Equivalente al *niveau objet* según [Latombe, 79]. Véase [Koutsou, 81] para un estudio bibliográfico sobre el tema.



Programación explícita. (nivel de manipulador)	Programación implícita. (nivel tarea)
ir a P_1	
ir a P_2	
cerrar pinza	coger O_1
ir a P_1	
ir a P_3	mover O_1, C_1 CONTRA O_2, C_2
ir a P_4	
abrir pinza	dejar O_1

Figura 1.2. Adaptada de [Koutsou, 81]. Alternativas en la programación de una tarea simple de "coge y deja". En la parte izquierda se observan las instrucciones necesarias para ejecutar la tarea a nivel de manipulador, orientada a las acciones. En la parte derecha, su equivalente en versión de nivel tarea, orientada al objeto (modelo del mundo).

mover objeto 1, característica 1 *contra* objeto 2, característica 2

donde aparecen relaciones que implican restricciones espaciales (e.g. *contra*, *coplanar*, etc.) es típico de los lenguajes de programación a nivel tarea como RAPT ([Popplestone et al., 78], [Popplestone et al., 80]) o LM-GEO ([Mazer, 83]), una de cuyas metas es facilitar la especificación de tareas al usuario. La idea es que a partir de, únicamente, ciertas declaraciones simbólicas que definen unos estados objetivos se pueda desencadenar automáticamente la correspondiente secuencia de operaciones a nivel del manipulador [Ambler & Popplestone, 75], [Lieberman & Wesley, 75], [Lozano-Pérez, 82].

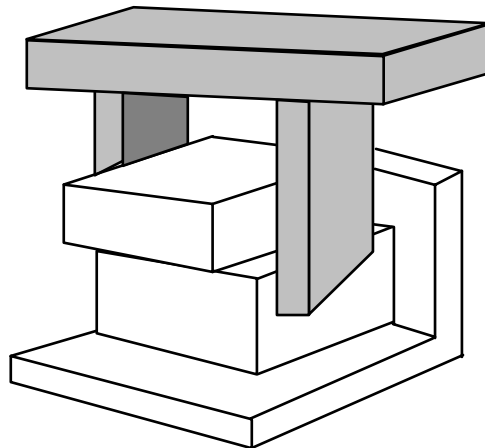


Figura 1.3. Adaptada de [Mason & Salisbury, 85]. Alineación de dos objetos (e.g. una caja y su tapadera).

En relación con los lenguajes de programación de robots y las operaciones propias de manipulación, existen cinco operaciones básicas generalmente aceptadas [Mason & Salisbury, 85]:

- mover a (*move to*). Existen dos versiones, según la mano del manipulador sostenga o no un objeto. El objetivo de dicha operación será mover el objeto o el robot desde su posición/orientación actual a la especificada.
- agarrar (*grasp*). Objetivo: restringir la posición/orientación del objeto relativo a la mano al cerrar los dedos (pinzas).
- dejar (*ungrasp*). Asume que el objeto es sostenido por la mano pero que la posición del objeto es estable, incluso sin la mano, por tanto, abriendo la mano el objeto no se moverá.
- mover-hasta-tocar (*move-until-touch*). El robot mueve la mano en una dirección establecida hasta que se activan una o más señales provenientes de sensores de contacto.
- mover-acomodar (*compliant-move*). El robot ejecuta un movimiento de acomodación a las restricciones de posición debidas al contacto con un objeto.

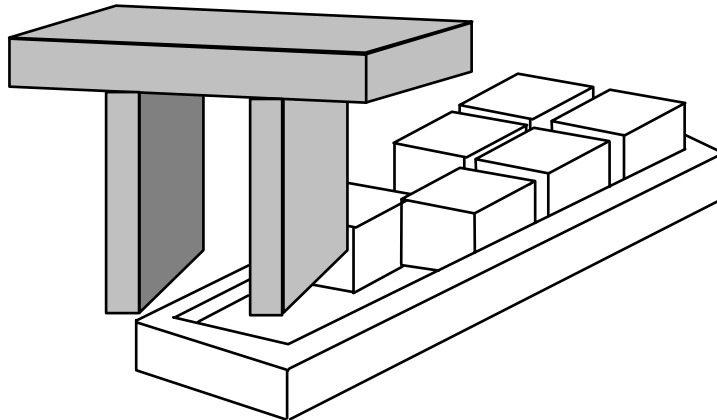


Figura 1.4. Adaptada de [Mason & Salisbury, 85]. Paletización de un bloque.

Las dos últimas operaciones incorporan la posibilidad de existencia de incertidumbre, a considerar en condiciones reales, mientras que las tres primeras representan las primitivas básicas de cualquier programa orientado a manipulación.

Es interesante puntualizar la existencia de ciertas aplicaciones de manipulación, desarrolladas con pinzas convencionales (dos dedos plano-paralelos) no dirigidas al agarre propiamente dicho:

- Alineamiento de objetos (object-alignment) [Albus & Evans, 76] (véase la figura 1.3).
- Empuje (Pushing) [Mason & Salisbury, 85] (véase la figura 1.4).

Normalmente estas operaciones requieren el uso intensivo de sensores de fuerza para poder resolver las incertidumbres de posicionamiento que se plantean. Observando las figuras 1.3-4 podemos apreciar la existencia de "paredes" que ayudan a computar la dinámica de fuerzas resultante necesaria para alcanzar los objetivos propuestos en cada caso.

La tendencia actual en la programación de robots es la de hacer uso extensivo de información sensorial. En este sentido podemos decir que estamos asistiendo al nacimiento de nuevos paradigmas en programación, como el de "la programación de robots basada en visión" [Inaba & Inoue, 90], con especial interés hacia aplicaciones de ensamblado [Kuniyoshi & Inoue, 93]. En el apartado 2.1, volveremos a incidir sobre este aspecto.

La correspondencia directa con nuestro problema: "la planificación del agarre basado en visión", es clara observando la figura 1.2; es decir, a partir de la información proporcionada por nuestro sistema de visión podremos disparar la

ejecución del agarre, encapsulando en una sola instrucción (e.g. llamada a un procedimiento) las diferentes posiciones intermedias y el manejo de pinza necesario.

Finalmente, puntualizar que el presente trabajo representa una aportación directa al proyecto CICYT: "Movimiento inteligente de un brazo robot con incertidumbre basado en visión artificial y sensores", del cual forma parte el doctorando. En dicho trabajo, referente al agarre, podemos establecer dos etapas: precontacto y contacto. La primera interseca totalmente con el trabajo aquí desarrollado, y la segunda, desarrollada por otros miembros del equipo de investigación [Cervera & del Pobil, 95a], [Cervera & del Pobil, 95b], requiere la entrada en funcionamiento de un planificador de movimiento fino, el cual utilizará redes neuronales con realimentación de la información (fuerzas y momentos), obtenida a través de sensores de fuerza, insertando o extrayendo la herramienta de sus correspondientes lugares a pesar de la incertidumbre en la posición.

1.2. Ubicación del problema en el contexto de la robótica

Un problema clave en esta tesis será la generación del agarre, el cual representa un componente básico de un sistema más general, conocido en el campo de la robótica como "planificador del agarre". Dicho sistema se considera de enorme complejidad, en tanto en cuanto contiene, a menor escala, toda la problemática inherente a los sistemas robotizados en su conjunto; es decir, el movimiento grueso (control de aproximación de la mano en la fase de precontacto), planificación de trayectorias libres de colisiones (condición de no intersección en la trayectoria del agarre propiamente dicho), movimiento fino (fase de contacto), etc.

El problema de la planificación del agarre forma parte de otro mucho más general: la planificación del ensamblado. En general los trabajos de investigación desarrollados en dicho campo siguen enfoques que pueden ubicarse en algún punto del espectro representado en la figura 1.5.

A lo largo de estos últimos años, los investigadores que se han enfrentado a un problema tan complejo como el de la implantación real de un sistema completo de planificación de ensamblado, a partir de un manipulador dotado de información sensorial (táctil, visual, etc.), se han apoyado en uno de los dos enfoques siguientes [Malcolm, 90]:

- Enfoque clásico (simbólico).
- Enfoque reactivo (reflejo).

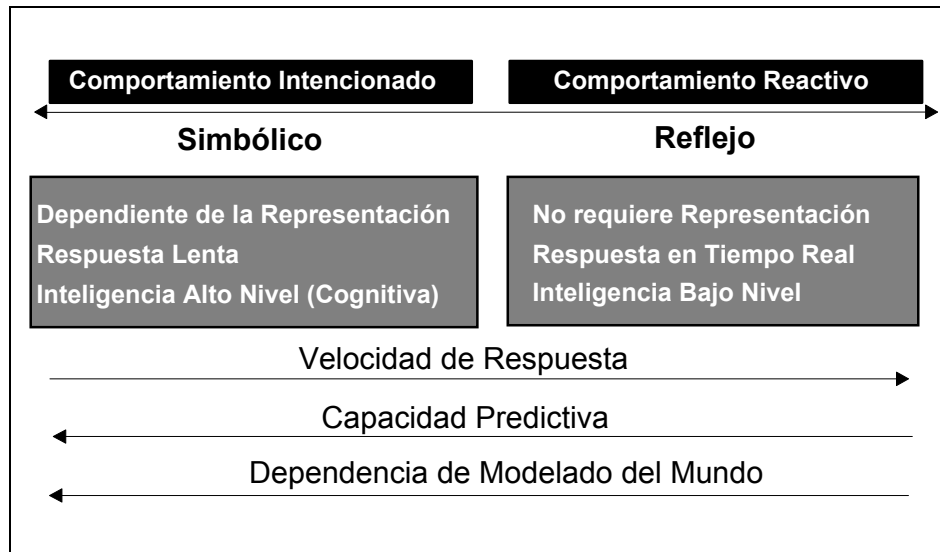


Figura 1.5. Adaptado de [Arkin & Grupen, 93]. Panorámica general de los diferentes enfoques seguidos en el desarrollo de sistemas robotizados (e.g. la planificación de ensamblado).

El primero (clásico) se basa en la estrategia, *top-down*, de descomposición de un problema complicado en los subproblemas necesarios para hacerlo abordable con la tecnología y los conocimientos disponibles. Una vez que estos subproblemas queden resueltos, el paso siguiente será el de su integración. Este primer enfoque se relaciona con la parte izquierda de la figura 1.5. Como ejemplo más representativo podemos citar "HANDEY" (*Hand + Eye*) [Lozano-Pérez et al., 87], [Lozano-Pérez et al., 92], desarrollado en el MIT.

Por otra parte, el otro enfoque considerado (reactivo) parte de la estrategia, *bottom-up*, en la que inicialmente se toma el sistema completo pero simplificado (e.g. procesamiento de imágenes 2D en lugar de 3D), de manera que sea abordable con el conocimiento disponible. Una vez superada esta etapa, o bien se procede a tratar de incrementar la complejidad hasta alcanzar la asociada con el dominio del problema original, o bien se supone alcanzado el suficiente grado de conocimiento para atacarlo directamente con garantías de éxito. Como ejemplo representativo de este segundo enfoque, relacionado con la parte derecha de la figura 1.5 (reflejo), tenemos la denominada: "*Subsumption Architecture*" de Brooks (MIT) [Brooks, 86].

Los dos enfoques anteriores son análogos a los utilizados en diseño de *software* y, como allí, son los polos extremos del espectro de estrategias representado en la figura 1.5.

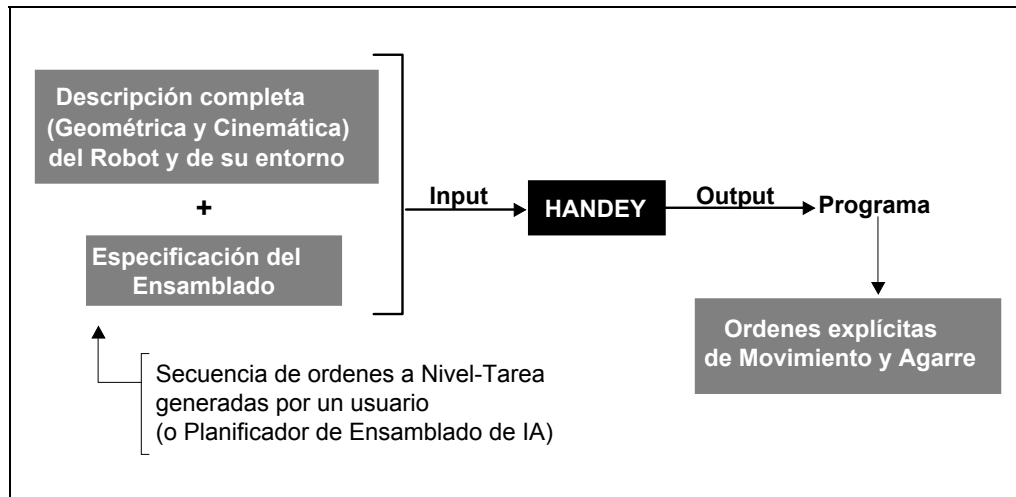


Figura 1.6. Diagrama de bloques del sistema HANDEY, donde se aprecia la información de entrada y la subsiguiente generación del programa final a nivel manipulador.

Veamos un poco más detenidamente cada uno de los ejemplos que se acaban de comentar. "HANDEY" tuvo su origen en un intento de conectar los dos sistemas siguientes:

- Un sistema de reconocimiento de objetos mediante *mapas de profundidad* en escenas tridimensionales [Grimson & Lozano-Pérez, 87], y
- Un programa de *planificación del movimiento* [Lozano-Pérez, 87].

Una representación gráfica del *modus operandi* de este sistema puede verse en la figura 1.6. El diseño de este planificador es modular, su núcleo lo forman cuatro planificadores independientes: el de movimiento grueso (*gross-motion*), el de agarre (*grasp*), el de re-agarre (*regrasp*²) y el de coordinación de múltiples manipuladores (*multi-arm coordinator*). Dichos planificadores operan con cinco clases principales de estructuras de datos: los objetos, el entorno, el robot, el objetivo y el plan. Donde el objetivo puede venir determinado tanto por un conjunto de valores del espacio de articulaciones asociado a un manipulador dado, como por la posición final deseada de un objeto. Por otra parte, un plan describe los movimientos elementales del manipulador y su pinza producidos por un planificador para alcanzar el objetivo deseado, el cual también contendrá el estado del mundo producido como resultado de cada operación. Sobre el problema del movimiento grueso puede encontrarse un estudio en profundidad en el libro [del Pobil & Serna, 95].

² Concepto definido posteriormente en el apartado 3.1

Digamos, para finalizar, que este sistema está considerado por muchos investigadores como el más completo en su género. Sin embargo, requiere de un conocimiento geométrico completo del entorno en todo momento y, como era de esperar, sólo es capaz de manipular objetos previamente aproximados o modelados como poliedros perfectos. Uno de sus puntos débiles es la especificación del ensamblado, tarea pendiente de resolver, sugiriéndose la creación de una interfaz con algún planificador convencional de IA. Además, en lo tocante a la generación del agarre, no tiene en cuenta condiciones de estabilidad en el mismo, nosotros sí lo haremos.

El siguiente ejemplo a comentar es: "*Subsumption Architecture*" de Brooks [Brooks, 86], surgido en relación con el programa de investigación dirigido por Brooks sobre robots insectoides, desarrollado en el MIT. Dicha arquitectura representa un buen ejemplo del denominado control reactivo. Aquí la típica estructura jerárquica de los modelos convencionales, como el descrito anteriormente, es sustituida por un modelo de capas. Los comportamientos de alto nivel (capas más externas) **subsumen** aquellos que representan actividades a bajo nivel. Está inspirado en la biología; en concreto, a partir de la acción coordinada de comportamientos, observada en organismos simples como los insectos (comportamiento insectoide). No se utilizan modelos del mundo y su viabilidad ha sido probada en diferentes robots del laboratorio de IA en el MIT, inicialmente, y en otros laboratorios de prestigio, posteriormente. El esquema básico de trabajo de esta arquitectura puede seguirse en la figura 1.7.

Podemos hablar pues, de la existencia de un paradigma clásico (escuela jerárquica) *versus* un paradigma comportamental (escuela reactiva). Siendo los ejemplos que acabamos de comentar, representativos de cada uno de ellos. Y como casi siempre en el punto medio está la virtud, habrá que encontrar un compromiso entre ambos enfoques, de manera que se eviten los defectos de ambos y se pongan de manifiesto sus virtudes. En el fondo, tanto el control reactivo como la planificación jerárquica, conforman partes diferentes de un mismo problema (filosofía híbrida):

- Ejecución del plan control reactivo, y
- Formulación del plan sistemas jerárquicos

Podemos destacar algunas características notables de los planificadores jerárquicos: poseen una alta dependencia de los modelos del mundo, pueden integrar directamente conocimiento que los usuarios tienen del mismo y, en general, facilitan la perspectiva global del problema. Por contra, será característico de los sistemas reactivos: garantizar robustez en su comportamiento, el desarrollo modular y el crecimiento incremental, y su predisposición para la interfaz sensorial. El objetivo de los sistemas híbridos será la ejecución robusta, mediante comportamientos, de un plan que sea capaz de explotar (sacar partido) del conocimiento de alto nivel del entorno (mundo) del robot.

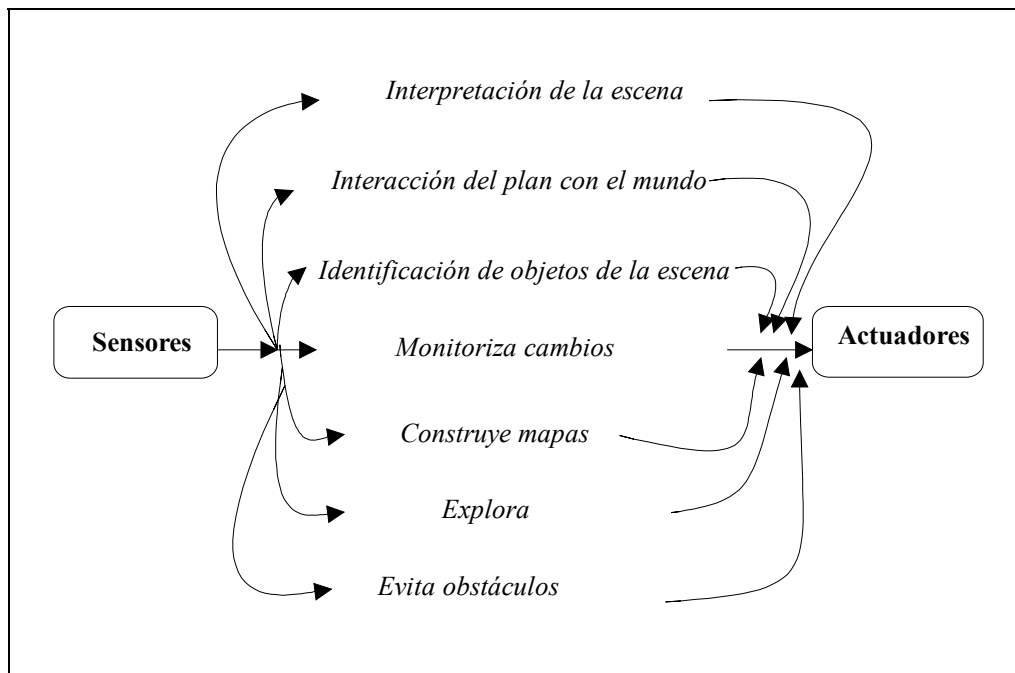


Figura 1.7. Representación del esquema de trabajo (por capas) de Brooks.

El control reactivo y la planificación jerárquica son completamente compatibles y existen evidencias psicológicas y neurocientíficas que demuestran que este enfoque (híbrido) es plausible [Arkin & Grupen, 93]. Esta cuestión ha sido tema de estudio en un *workshop* reciente [Musliner, 96].

Un ejemplo de sistema híbrido, de gran influencia en el contexto de aprendizaje de robots [Van de Velde, 93], lo constituye el sistema "TheoAgent" desarrollado por Mitchell [Mitchell, 87], cuyas características más destacadas son:

Reacciona cuando puede, planifica cuando debe.

Usa reglas estímulo-respuesta para reaccionar.

El método de elección de reglas es un componente de la planificación.

Capaz de aprender nuevas reglas.

Otro ejemplo de sistema híbrido más próximo a nuestros intereses es: "The SOMASS³ System" [Malcolm, 90]. El dominio de soma en su forma original, consiste en construir todas las formas irregulares diferentes posibles poniendo como máximo cuatro cubos juntos, cara con cara. En síntesis la estrategia utilizada consiste en mantener al planificador del ensamblado operando al mayor nivel de abstracción posible, para lo cual se descarga del manejo de sensores, incertidumbres, etc., haciendo que el sistema encargado de la ejecución del plan sea tan inteligente como sea posible. El planificador sigue el enfoque de los sistemas simbólicos tradicionales de IA, mientras que la ejecución del ensamblado se lleva a cabo por un sistema basado en el comportamiento (*behaviour-based system*). Veamos algunas definiciones al respecto [Malcolm, 90]:

- Comportamiento (*behaviour*). Operación o secuencia de operaciones con un determinado fin. Son modulares y jerárquicos.
- Módulo comportamental (*behavioural module*). Unidad encapsulada de cierta combinación de "*hardware + software + entorno*" que implementa un comportamiento.
- Operador comportamental (*behavioural operator*). Función abstracta necesaria para el manejo a nivel computacional de los módulos comportamentales. El ejemplo más simple relacionado con dicha manipulación computacional lo constituyen las llamadas a procedimientos.
- Sistema basado en el comportamiento (*behaviour-based system*). Formado en su totalidad por módulos comportamentales.
- Sistema simbólico (*symbolic system*). Programa tradicional de IA que utiliza programación simbólica, i.e. utiliza una representación simbólica sobre la que razonar, hacer búsquedas, etc. Una característica esencial de estos sistemas es que poseen una representación general del mundo, espacio del problema, o similar, y que los procesos pertinentes se controlan de modo central. A este respecto son la antítesis de los sistemas basados en el comportamiento.

Con las definiciones introducidas podemos retomar el concepto de sistema híbrido, surgido de la necesidad de construir un planificador, para abordar la fase donde se concentra la mayor complejidad lógica del problema, mediante un sistema simbólico tradicional de IA, dejando la actuación para los sistemas basados en el comportamiento. Es decir, se requiere que el sistema de ensamblado (en el ejemplo que nos ocupa) basado en el comportamiento sea susceptible de ser controlado por un plan de ensamblado. Los únicos términos en los que se expresará dicho plan será a través de operadores comportamentales.

³ SOMASS = "soma assembly". El dominio de soma es bien conocido en matemáticas como un puzzle caracterizado por un espacio de búsqueda combinatoriamente explosivo [Gardner, 61].

Algunas características típicas de los sistemas basados en el comportamiento son:

- Manejan las incertidumbres típicas de la tarea.
- Integración a bajo nivel de la información sensorial y las acciones.
- Se mantienen tan ignorantes del resto del problema como sea posible, utilizando al mínimo la representación simbólica.

Algunas características típicas de la integración de los módulos comportamentales en un sistema son:

- Se ejecutan, a veces, en paralelo con otros módulos comportamentales, salvo que se inhiban entre sí.
- Se prueban sobre sistemas completos operando en entornos reales (por oposición a los simulados).

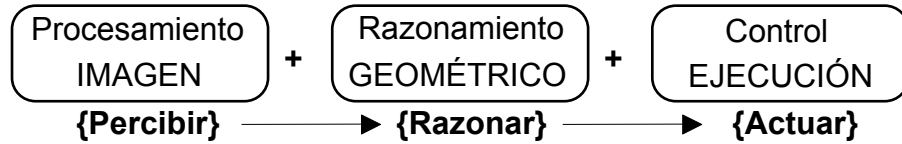
Por tanto, de lo dicho hasta ahora se deduce que lo más prometedor parece estar en el enfoque híbrido, así pues, esto motivará nuestra elección para tratar nuestro problema: la generación del agarre y su ejecución, de una manera modular, integrando el sistema final mediante dos módulos independientes: 1) generación del agarre, y 2) la ejecución del agarre propiamente dicho. 1) sería un módulo integrado en un planificador del agarre que a su vez formaría parte de un planificador de ensamblado, el cual trataría todos los aspectos elementales del mismo y al que, como hemos visto, interesa dotar de una estructura jerárquica, mientras que a 2) conviene aproximarse mediante un enfoque reactivo. Todo lo expuesto en este punto representa en esencia el contexto del trabajo aquí desarrollado, desde una perspectiva global en el campo de la robótica.

1.3. Objetivos de la Tesis

El objetivo central de la tesis aquí presentada es demostrar la viabilidad de un sistema completo de agarre que, actuando en un entorno real, fuera robusto, eficiente y simple, en la medida de lo posible.

Para conseguir lo anterior han sido necesarios toda una serie de objetivos parciales relacionados con las palabras clave (subrayadas) de la definición anterior:

Completo Se ha llevado a cabo la integración de la cadena "percepción-razonamiento-acción" [Winston, 93], mediante tres módulos básicos:



Entorno real No trataremos con objetos poliédricos ni, por supuesto, sintéticos, etc., nuestro universo de trabajo será el típico de un ambiente industrial (herramientas, tuercas, tornillos, etc.). Esto conlleva asumir la existencia de ruido, incertidumbre, etc., (entorno "no estructurado") a diferencia de los sistemas que trabajan con simulación (entorno "estructurado").

Robusto El sistema será capaz de funcionar con cierto margen de tolerancia relativa a variabilidad en la forma de los objetos, iluminación de la escena, posición de los objetos en la misma, etc.

Eficiente De todas las estrategias posibles en cada paso dado, hemos buscado la menos "cara", computacionalmente hablando que simultáneamente fuera robusta para alcanzar el objetivo requerido en cada caso.

Simple Ha primado en todo la economía de medios, tanto en la construcción de los algoritmos como en los medios "hardware" y "software" empleados.

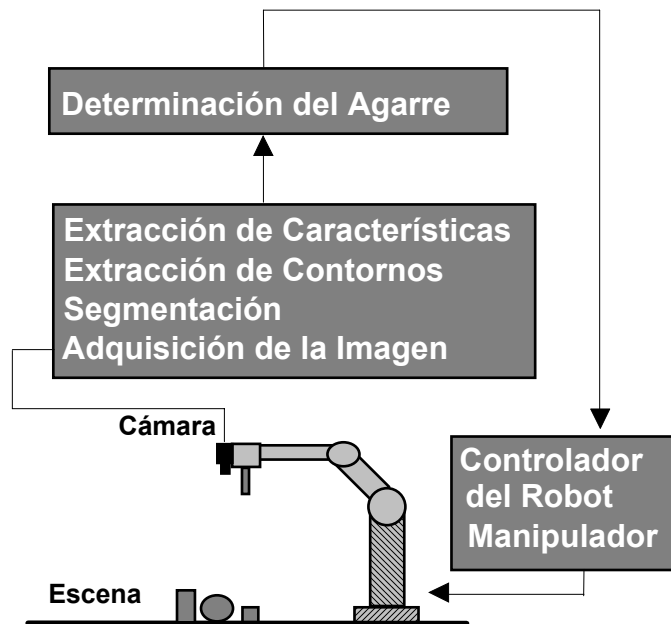


Figura 1.8. Adaptada de [Jarvis, 83]. Jerarquía establecida para el sistema completo de visión de robot planteado en la Tesis.

En la figura 1.8 se observa una visión de conjunto (jerárquica) del sistema completo que se plantea como objetivo, desglosado en los módulos más relevantes, mostrando la relación directa con el planteamiento expuesto anteriormente de "percepción-razonamiento-acción".

1.4. Organización de la Tesis

El estado de la investigación sobre el problema que nos ocupa se desarrolla en el capítulo II, donde se hará especial hincapié en la génesis de la percepción visual en su intersección con la problemática de la manipulación en el contexto de la robótica. Se intenta dar una perspectiva, no exhaustiva pero sí reveladora, de cuál ha sido el camino recorrido para llegar al dominio actual de éstas técnicas. En el capítulo III se abordan todas las hipótesis de partida necesarias para la caracterización del modelo de agarre elegido, así como su justificación. La problemática del sistema de visión utilizado, incluyendo todas las etapas desarrolladas desde la fase de adquisición hasta la extracción final de descriptores, es tratada en el capítulo IV. El algoritmo de detección automática de puntos de agarre y cada una de sus etapas elementales que, constituye una etapa fundamental en este trabajo, se desarrolla en el capítulo V, analizándose los resultados obtenidos a partir del mismo. La problemática relativa a la ejecución del agarre se presenta en el capítulo VI, que ha contado con la colaboración del Instituto de Robótica de Valencia [Sanz et al., 96]. Finalmente, en el capítulo VII y último se ofrecen, a modo de conclusión, las aportaciones realizadas así como algunas líneas de interés de futuras investigaciones.

Capítulo II

Antecedentes

En éste capítulo se pretende dar una breve exposición del estado actual de investigación referente al tema que nos ocupa, dando una perspectiva histórica y actual de los trabajos relacionados con el nuestro.

2.1. El Agarre y la percepción visual

2.1.1. Introducción

Los avances en una ciencia tan pluridisciplinar como la robótica, requieren de la confluencia de áreas de conocimiento muy diferentes, con orígenes y motivaciones dispares. De todas las perspectivas posibles, centraremos la nuestra en la intersección entre el agarre y la percepción visual.

Para facilitar la lectura de lo que sigue, haremos una serie de consideraciones preliminares:

- 1^a Denominaremos "mano" (también conocida como: efector final, elemento terminal, garra, pinza, etc.) a la interfaz del brazo mecánico del robot con su entorno (todos aquellos objetos que interactúan con él). Únicamente estamos interesados en elementos terminales con capacidad de aprehensión (dirigido a operaciones de ensamblado, fundamentalmente).

2ª La caracterización del "agarre" estará directamente relacionada con el diseño de dicha mano y, este diseño, a su vez, estará mediatizado por el tipo de aplicación a que se destine. Una primera clasificación, en cuanto al diseño de la mano se refiere, podría ser:

- ◆ Manos antropomorfas¹ (o también: diestras, de múltiples dedos, etc.), formadas por tres o más dedos poliarticulados, e inspiradas, habitualmente, en la mano humana.
- ◆ Manos simples (o también: pinzas, multipinzas, etc.), formadas por uno o varios pares de dedos plano-paralelos.

3ª Por otra parte, como se ha comentado, el diseño de la mano siempre está motivado por la tarea a desarrollar. En entornos industriales podemos considerar dos tipos básicos de objetos a manipular: rígidos y no rígidos (flexibles). Los primeros son los habituales en la industria de automoción y afines, plantas embotelladoras, industria azulejera y ensamblados en general, aquí predomina el uso de la mano simple (pinzas/multipinzas, de dedos plano-paralelos). Los segundos, menos automatizados que los anteriores, tienen que ver con la industria textil [Monkman, 95] y agroalimentaria, fundamentalmente, favoreciendo el uso de elementos terminales muy diferentes a las pinzas convencionales (desde pernos a campanas de vacío, pasando por elementos adhesivos, etc.) [Groover & Mikell, 86].

4ª En el contexto de la robótica existen básicamente dos tipos de información sensorial:

Propioceptiva. Proveniente de sensores internos al robot. e.g. sensores de fuerza (en la mano del robot).

Exteroceptiva. Provenientes de sensores externos. Estos se clasifican a su vez en activos, e.g. sensores táctiles, y pasivos, e.g. visión.

Dejemos a un lado estas primeras consideraciones y remontémonos a uno de los primeros hitos conocidos en la historia de las manos artificiales. La necesidad, de suplir un miembro tan valioso como la mano, perdido por accidente, en la guerra, etc., motivó la creación de dispositivos, de diseño antropomorfo, como el utilizado con dicho objetivo en 1509 por el caballero von Berlichingen, quien perdió su mano en una batalla a temprana edad [Childress, 72]. Durante muchos años este fue el único acicate de los investigadores para mejorar la mano (con vistas a servir de prótesis). Ya en nuestros días, en la década de los ochenta y con todo el impulso proveniente de la intersección de dos áreas en auge: la Robótica y

¹ De apariencia humana.

la Inteligencia Artificial, se diseñan en los laboratorios del JPL² (Stanford) y del MIT³ (Massachusetts), ambos en Estados Unidos, las denominadas manos de Salisbury (o Stanford / JPL *hand*) [Salisbury & Craig, 82], [Mason & Salisbury, 85]; y de Utah / MIT [Jacobsen et al., 84], respectivamente (véase la figura 2.1). Estas manos son, sin duda, las más citadas en la bibliografía al respecto y constituyen referencias obligadas en la investigación del agarre inteligente.

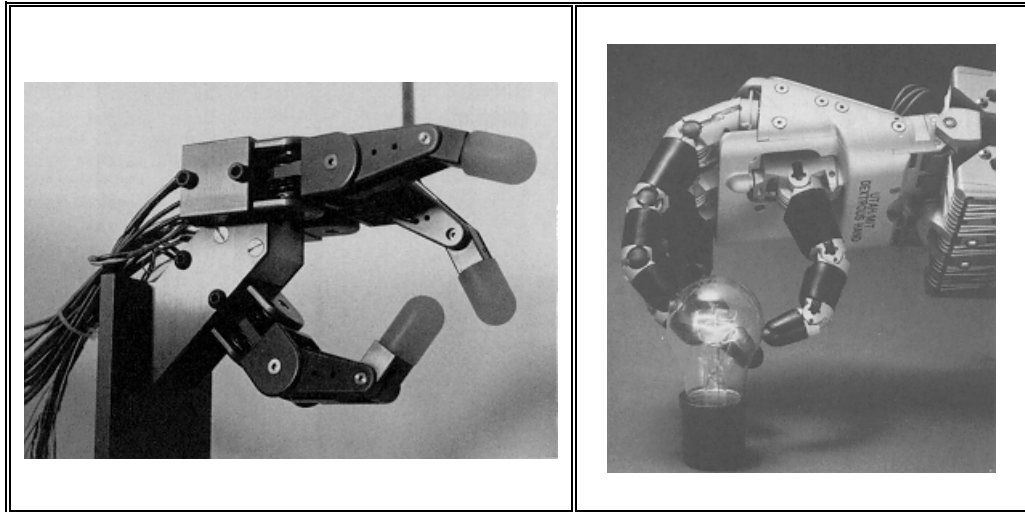


Figura 2.1. Adaptada de [Murray et al., 94]. (Izquierda) mano de Salisbury, con sus tres dedos característicos. (Derecha) mano de Utah, cuyo diseño de cuatro dedos es más antropomórfico que el anterior.

Existen trabajos muy recientes que utilizan estas manos antropomorfas, dotadas de sensores de fuerza, utilizando dicha información propioceptiva exclusivamente [Huber & Grupen, 94]; o junto con información visual para mejorar las estrategias de agarre a seguir [Bard et al., 95], [Fuentes et al., 94].

Por otra parte, la evolución de la mano simple (pinza convencional), ha seguido un camino muy distinto; desde sus comienzos estuvo ligada a las vicisitudes en el desarrollo de los robots mismos, y al igual que éstos, han derivado hacia diseños no antropomorfos, adaptados a su entorno de trabajo. Que nadie espere encontrar en las factorías de automóviles de hoy en día, ninguna mano diestra (de Salisbury o de Utah) trabajando en una línea de ensamblado o similar. Hay varias razones para explicar esto: 1º) el coste desorbitado de una de estas manos, y 2º) lo costoso, a nivel computacional, que resulta su control;

² JPL Jet Propulsion Laboratory.

³ MIT Massachusetts Institute of Technology

resumiendo, no existe la suficiente robustez, eficiencia y economía de medios para su implantación a corto o a medio plazo en un entorno real tan competitivo como el industrial.

No obstante lo dicho, conviene observar las desventajas de la pinza/multipinza convencional (de dedos plano-paralelos) frente a la mano antropomorfa [Murray et al., 94]:

- 1ª Carencia de destreza. Se pueden alcanzar agarres sólidos (seguros), pero no está permitida la manipulación del objeto agarrado (sin soltarlo). Esta es la explicación de porqué la mano antropomorfa se denomina a veces mano diestra.
- 2ª Número limitado de agarres posibles. Tareas diferentes exigirán elementos terminales diferentes.
- 3ª Movimientos prolongados del brazo. Determinados desplazamientos de la mano exigen extensos movimientos del brazo robot completo.
- 4ª Ausencia de control de fuerza fino, salvo raras excepciones.

Tengamos en cuenta que los problemas a superar por la mano diestra retrasarán su implantación en aplicaciones reales, al menos los años necesarios para lograr el progreso que requieren los mecanismos de control, fusión de información sensorial, etc. Pero un robot "inteligente", de propósito general, está abocado a poseer manos diestras. Los laboratorios de investigación más prestigiosos del mundo trabajan en ello actualmente.

Hemos dejado claro que el diseño que abre las mayores posibilidades es sin duda el antropomorfo, pero también somos conscientes de sus limitaciones actuales. En el lapsus de tiempo intermedio sería aconsejable preparar el terreno potenciando al máximo las posibilidades de la mano simple (la pinza convencional), que ya está funcionando desde hace varios años en múltiples entornos industriales. Y aquí es donde entra en juego la percepción artificial.

2.1.2. Hacia la manipulación inteligente

Investigadores como Dario [Dario, 89] plantearon en su día un dilema que sigue siendo actual en la automatización industrial de nuestros días: ¿cuándo se debería incorporar información sensorial a los sistemas robotizados reduciendo, en consecuencia, el grado de estructuración del entorno de trabajo?, y ¿cuándo se debería prescindir de dicha información, incrementando la estructuración del campo de operaciones?.

Dario responde sugiriendo que podemos distinguir dos extremos en el campo de aplicación o dominio de los robots industriales:

- A.** Tareas cuyas prioridades son: alta velocidad y precisión absoluta en la posición. Características típicas de las cadenas de producción o ensamblado a gran escala.
- B.** Tareas más sofisticadas, necesitadas de información sensorial. Típica de la producción o ensamblado de pequeños lotes.

En **A** interesa estructurar al máximo el entorno, evitando el uso de manos sofisticadas y sensores. Por contra, **B** justifica el coste adicional que supone el uso de sensores y la pérdida de estructuración. En conclusión, el uso de sensores estará justificado en aquellas aplicaciones en las que la calidad prime sobre la velocidad como único parámetro a contemplar.

Al margen de justificaciones como la anterior para el empleo o no de sensores, digamos que éstos tienen interés *per se*, en el sentido de que son ingredientes intrínsecos del proceso de percepción, siendo ésta la vía obligada para alcanzar la conexión inteligente entre percepción y acción, que es como se ha definido la denominada "robótica inteligente" [Brady, 84].

En general, la percepción artificial, según Mackerrow [Mackerrow, 93], intenta capacitar al robot para realizar tareas complejas, adaptarse a posibles variaciones de su entorno (ambiente no estructurado) y afrontar situaciones imprevistas. Se trata de un problema difícil, cuya solución requiere la aportación de diferentes campos del conocimiento y, en su forma general, exigirá la utilización de herramientas tecnológicas aún no disponibles.

Para comprender mejor donde residen las dificultades podemos buscar un paralelismo con la percepción visual humana. Para una revisión bibliográfica sobre el tema ver el trabajo de Zucker [Zucker, 81]. El mecanismo de percepción visual humano funciona como una cadena en la cual, a partir de un cierto estímulo (la luz), capturado por un receptor (la retina), se produce una determinada respuesta. Dicha respuesta, generalmente, requiere una interpretación previa de la escena iluminada que provocó el estímulo, y ésta a su vez puede depender del contexto, de la experiencia cognoscitiva anterior y de los objetivos que se persigan. Por tanto, la interpretación de una sensación no puede basarse únicamente en la información contenida en la misma. Requiere la cooperación de otras fuentes de información (experiencia, contexto, objetivos, etc.).

Llegado este punto interesa recordar algunas reflexiones de Russell [Russell, 48] al hilo de la percepción y el ordenador. Este plantea preguntas como: ¿podría un ordenador, con suficiente conocimiento de la estructura de determinado cerebro, predecir la respuesta muscular a un estímulo dado utilizando las leyes de la física y la química?; ¿O la intervención de la mente es un eslabón esencial para conectar un antecedente físico (el estímulo) con una consecuencia física (un movimiento corporal)? La experiencia demuestra que existen reflejos en los que la respuesta es automática y no controlada por la volición.

Subyacente en las observaciones de Russell está la idea del movimiento de acomodación *versus* el balístico. En el ser humano, el primero hace uso de cierto mecanismo de realimentación sensorial, mientras que el segundo se relaciona con los movimientos reflejos. El origen de la traslación de estos conceptos, en especial el de realimentación, a los sistemas de control de los robots, se debe a Wiener [Wiener, 48]. La realimentación sensorial en los sistemas robotizados, permite cerrar el bucle de control, dando lugar a los denominados sistemas servocontrolados, si la información sensorial utilizada es propioceptiva. Si además, se utiliza realimentación de información exteroceptiva (e.g. visual o táctil), se conseguirá una adaptabilidad respecto a la posición y una potencial capacidad de reacción, logrando que el sistema robotizado sea mucho más robusto⁴ e inteligente⁵.

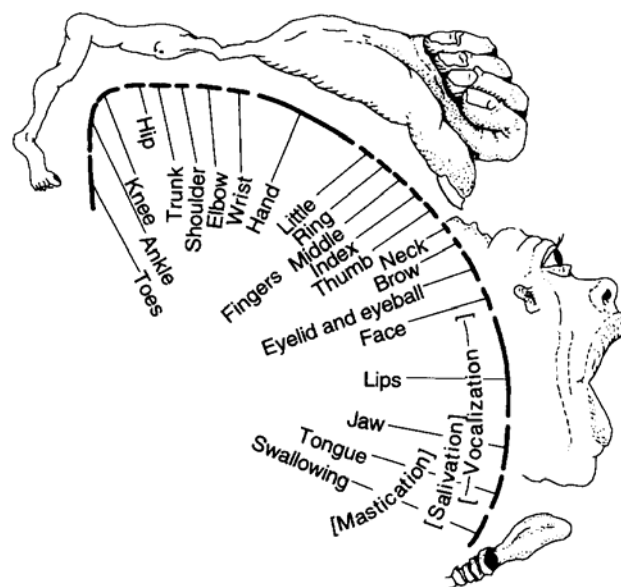


Figura 2.2. Adaptada de [Murray et al., 94]. Diagrama del *homunculus* del córtex motor. Destaca el gran porcentaje del córtex asociado a la mano completa (*hand, little...thumb*).

Los movimientos de acomodación en el hombre están relacionados con el uso de realimentación de la información sensorial (vista, tacto, etc.) para realizar movimientos precisos y tienen un paralelismo con el llamado movimiento

⁴ Es decir, menos vulnerable a posibles errores de posicionamiento, derivados del calibrado, modelo cinemático, etc.

⁵ Suponiendo que la capacidad de adaptación sea un indicador de un cierto grado de inteligencia.

diferencial en robótica, e.g. cuando se utiliza un sistema de visión para monitorizar la localización del elemento terminal. Para usar dicha información, debemos de ser capaces de controlar el movimiento diferencial del robot, esta información estará contenida, habitualmente, en una matriz: el Jacobiano del manipulador. Los aspectos relacionados con el control se relegan al apartado 2.3 y al capítulo VI, relacionados ambos con la ejecución del agarre.

Es interesante tomar conciencia de la conexión existente entre la capacidad de manipular algo por un ser vivo y su grado de inteligencia. Los neurofisiólogos miden dicho grado en los mamíferos estableciendo una relación entre un área determinada del cuerpo y el córtex motor en el cerebro. En la figura 2.2, se representa esta idea a partir del trabajo pionero de [Macmillan, 50], formando el denominado *homunculus*. Para seres humanos la mayor fracción del córtex motor (30-40 %), se dedica al control de las manos, en comparación con el 20-30 % para la mayoría de monos y menos del 10% para mamíferos como el gato o el perro. En general las denominadas neurociencias del comportamiento están dando un gran impulso a la investigación en robótica actualmente, sugiriendo nuevas estrategias de control o nuevos puntos de vista en los mecanismos de interacción del robot con el entorno, permitiendo progresar en la implantación de aspectos básicos asociados con la percepción. Una excelente fuente de información actualizada al respecto se encuentra en [Churchland & Sejnowski, 92].

No conviene perder de vista a la mayor fuente de modelos existente en el campo de los robots manipuladores: el ser humano. En concreto, la mano humana dotada en síntesis de dos funciones [Smith et al., 81]: abrazamiento esférico y pinza inteligente (véase la figura 2.3), constituye sin duda la herramienta más perfecta jamás diseñada, a imitar por manos cada vez más antropomorfas, como hemos visto anteriormente. En palabras de algunos investigadores en el campo de la biomecánica de los miembros:

"Mano y cerebro están indisolublemente unidos en el desarrollo de la capacidad intelectual humana, y esta dependencia del desarrollo cerebral y del desarrollo evolutivo de la mano, hace que se pueda considerar ésta como la expresión de la inteligencia" [Smith et al., 81].

Por otro lado, tengamos en cuenta que los huesos y articulaciones de la mano humana permiten 22 grados de libertad principales [McCloy & Harris, 93], lo cuál hace que sea impensable el poder construir, sólo desde el punto de vista del control (dejando de lado otros aspectos), nada parecido con la tecnología actualmente en uso.

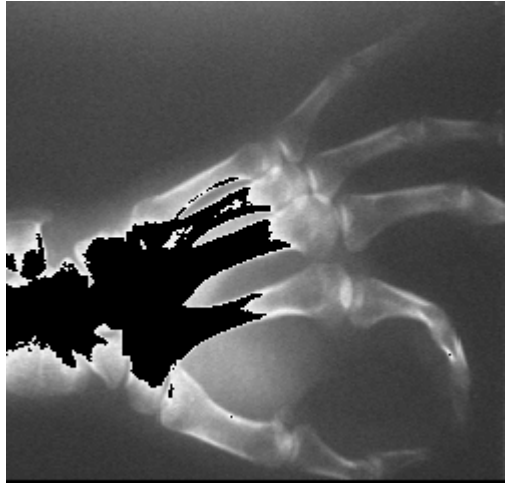


Figura 2.3. A partir de la radiografía de una mano humana. Se aprecia la posición de pinza formada por los dedos pulgar e índice, denominada "pinza inteligente" por los especialistas en biomecánica de los miembros.

Un dilema interesante en el campo de intersección de la robótica con la inteligencia artificial (IA), de plena actualidad, es el relacionado con la simulación versus la implantación real de un sistema robotizado dotado de sensores. Dreyfus [Dreyfus, 92] plantea la cuestión del papel jugado por el "cuerpo", como interfaz de un organismo con su entorno, en el denominado comportamiento inteligente. La cuestión que subyace es: ¿se puede simular completamente el comportamiento de un robot en interacción sensorial con su entorno?. El asunto está en que en IA, problemas como jugar al ajedrez se resuelven completamente en un ordenador, mientras que, por ejemplo, la interpretación de escenas reales se resiste a ello. Hoy por hoy, la incertidumbre asociada con el proceso de percepción (e.g., el ruido inherente a la captura de la señal por los sensores, etc.) no se puede modelar de forma que sustituya a los procesos reales de manipulación de la información sensorial. Esta es la razón fundamental por la que las implantaciones reales en robótica ofrecen una mayor credibilidad que las simuladas.

En la necesidad de buscar nuevos enfoques que permitan al robot desenvolverse en entornos no estructurados ha surgido, como expusimos en el capítulo I, el paradigma comportamental, y en esta línea aparecen aportaciones muy interesantes, como [Hallam & Hayes, 92] que ofrecen un estudio comparativo con la idea de incorporar nuevas posibilidades al diseño de los robots que se aproximen a las capacidades reactivas de los sistemas biológicos.

2.1.3. Perspectiva Actual

Si echamos un vistazo a lo hecho en robótica en la línea de investigación relativa a la problemática del agarre, una referencia obligada será [Pertín-Troccaz, 89]. La cual presenta una extensa revisión bibliográfica sobre *grasping*. Puntualicemos que de los 70 trabajos allí referenciados, únicamente tres de ellos intersectan con la visión y su problemática de procesamiento de imágenes; cuatro utilizan sensores de rango/profundidad (láser, etc.); y, finalmente, sólo tres de ellos emplean sensores táctiles. Así pues, únicamente el 10 % de los trabajos presentan algún tipo de información sensorial. Tengamos en cuenta, además, que el trabajo más actual de los referenciados data del 87.

Como ejemplo de uno de los trabajos más avanzados de esta etapa de finales de los ochenta se encuentra el desarrollado por Horn en los laboratorios del MIT [Horn, 86]. El objetivo fundamental aquí fue relajar las tres condiciones en que normalmente se basan los sistemas de manipulación con visión, es decir:

- 1ª Existe un fuerte contraste entre los objetos y el fondo.
- 2ª Los objetos deben estar separados entre sí.
- 3ª Los cambios en la localización (posición más orientación) de los objetos deben confinarse a un plano paralelo al plano de la imagen.

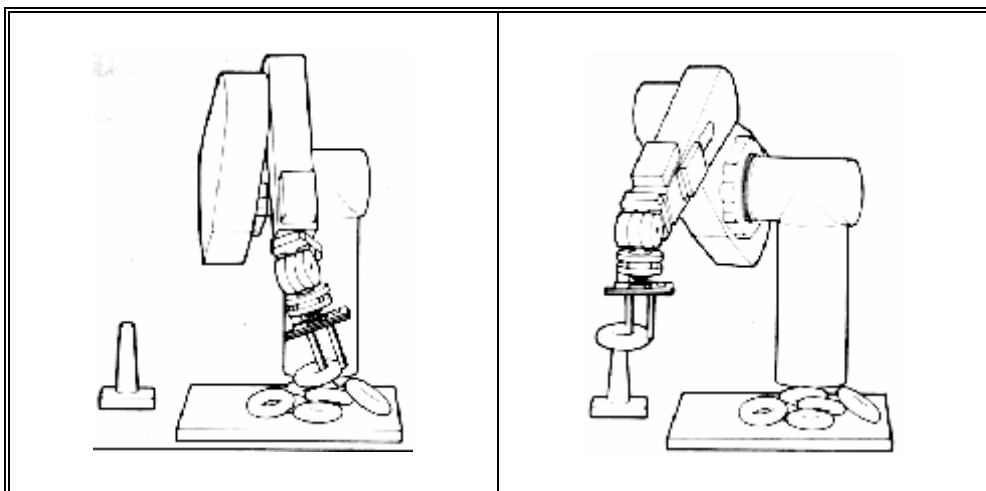


Figura 2.4. Adaptada de [Horn, 86]. Problema clásico de "coge y deja" a partir de una pila de objetos amontonados (*picking parts out of a bin*).

Un detalle de una posible aplicación directa de este trabajo se puede observar en el ejemplo de la figura 2.4. Aquí se ataca el problema tradicional de "coge y deja" a partir de una pila de objetos amontonados (*picking parts out of a bin*).

Un sistema, ya comentado en el capítulo I, que utiliza un modelo para la geometría de la pinza análogo al nuestro, pero sobre objetos modelados como poliedros, es HANDEY de [Lozano-Pérez et al., 92]. No trabaja, por tanto, sobre entornos no estructurados. Una de sus grandes aportaciones la constituyen los algoritmos denominados de "regrasping", pero sobre todo la capacidad de integrar en un solo sistema módulos de agarre o de planificación de trayectorias libres de colisiones, entre otros. Para obtener modelos del entorno combina información sensorial de una cámara CCD con técnicas de barrido láser, con vistas a obtener información sobre la profundidad de la escena.

También, en la misma línea que el anterior, iniciado hace ya algunos años pero en completa actividad, está el proyecto SHARP, dirigido actualmente por Christian Laugier, miembro del INRIA⁶, en Grenoble (Francia). El interés inicial de este proyecto se centra en el desarrollo de técnicas dirigidas a la programación a nivel tarea y de autonomía que permitan la aplicación de robots a nuevas tareas, parcialmente especificadas y no repetitivas. El esfuerzo principal se dirige a estrategias de planificación y ejecución de movimientos complejos, manipulación diestra y reactividad. En concreto, el equipo dedicado a la investigación sobre la planificación del agarre diestro, se centra en el agarre basado en visión, utilizando técnicas de estereoscopia, con mano diestra, la mano de Salisbury (ver figura 2.5). Una de sus últimas aportaciones [Bard et al., 95] pone de manifiesto las dificultades extremas en operar con dicha mano en la implementación final, teniendo que utilizarse de momento una pinza de dos dedos plano-paralelos convencional. Esto refuerza nuestra opinión, sugerida anteriormente, de que en el *impasse* de mejora efectiva sobre el diseño de las manos diestras, es necesario realizar aportaciones al agarre con la pinza convencional de dedos plano-paralelos.

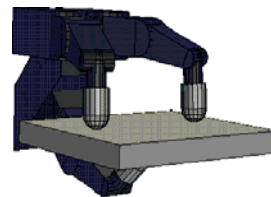


Figura 2.5. Modelado sólido utilizado en el proyecto SHARP para la mano de Salisbury.

Estos dos últimos proyectos comentados (HANDEY y SHARP) ponen de manifiesto la relación existente entre el esfuerzo por desarrollar un lenguaje efectivo de programación a nivel tarea y la necesidad a todos los niveles del uso

⁶ INRIA ≡ Institut National de Recherche en Informatique et Automatique.

de percepción sensorial; en concreto: la planificación del agarre basado en visión. Como ya se comentó en el capítulo I, esto ha derivado en el surgimiento de un nuevo paradigma en la programación de robots: basada en visión [Inaba & Inoue, 90] (véase la figura 2.6). En este marco de programación, se emplean tanto la visión humana como la de robot para resolver una tarea. La visión del robot muestra la imagen del entorno y los objetos que puede reconocer el usuario. La visión del usuario humano se utiliza para evaluar la habilidad del sistema de visión del robot y especificar la tarea de acuerdo a las posibilidades del sistema robotizado. Mediante la interfaz visual de usuario, el usuario puede a su vez especificar donde debería mirar el robot. Esta arquitectura representa una mejora significativa en la interfaz hombre-robot, facilitando la integración de características como visión, manipulación, inteligencia e interfaz de usuario en un mismo sistema al modo de HANDEY o SHARP, vistos anteriormente.

Una última observación, relativa al impacto del uso de visión en la programación de robots, deriva del hecho, no ya de que determinados entornos de programación extiendan sus posibilidades para admitir el empleo de dichos sistemas (esto es lo habitual), sino que, incluso determinados sistemas de visión acaban constituyendo el núcleo sobre el cual se construye el lenguaje.

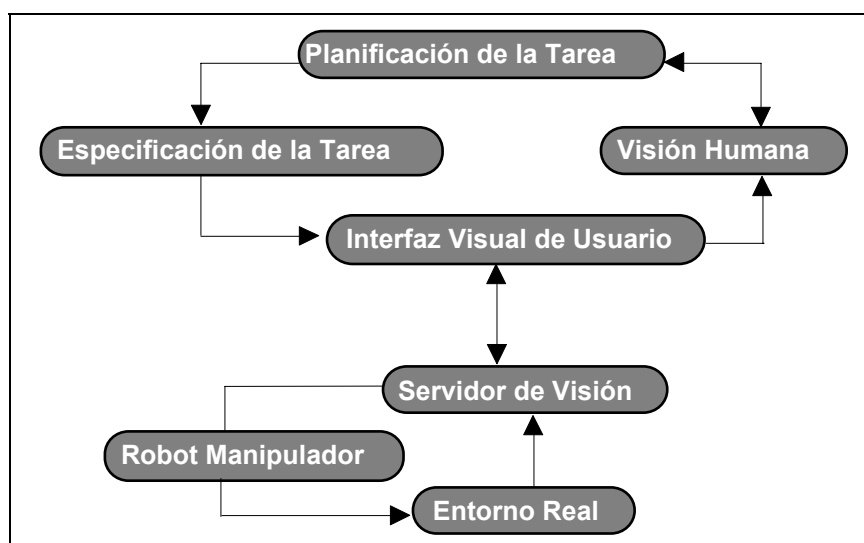


Figura 2.6. Adaptada de [Inaba & Inoue, 90]. Esquema de bloques básicos para la programación de robots basada en visión.

Este último es el caso del lenguaje RAIL, desarrollado por la empresa Automatix Inc. (USA) a partir del sistema de visión AUTOVISION, también de la misma empresa que deriva de los algoritmos SRI, desarrollados como su nombre indica en el Stanford Research Institute, a partir de trabajos pioneros en visión aplicada al campo de la robótica ([Agin & Duda, 75], [Gleason & Agin, 79] y

[Agin, 80], fundamentalmente). Digamos, por ejemplo, que RAIL posee capacidad para identificar hasta 45 características visuales diferentes [Mackerrow, 93], estando especialmente preparado para tareas donde intervengan: manipulación y clasificación de objetos, o la inspección de los mismos. Una visión actualizada de las características más significativas de los algoritmos SRI, puede seguirse en [Klafter et al., 89].

2.2. Generación del agarre

Como ya comentamos en el punto 1.2, la problemática del agarre en el contexto de la robótica se engloba en el problema denominado "planificación del agarre", el cual se matizará en el capítulo III, pero adelantamos que encierra entre otros tópicos el de la generación y/o análisis del agarre que nos ocupa, y la ejecución del mismo, a tratar en el punto siguiente 2.3. La generación del agarre consiste en determinar el mismo a partir de unos criterios específicos: preadaptación de la forma del objeto a la pinza y viceversa, estabilidad, etc.

Gran parte de la investigación desarrollada en la determinación del agarre ha seguido alguno de los enfoques siguientes [Bard et al., 95]:

- A1.** Basados en aspectos mecánicos del sistema "mano-objeto".
- A2.** Inspirados en estudios sobre el agarre humano.

Una consecuencia práctica del enfoque **A1** ha sido la necesidad de modelos geométricos precisos (CAD, sintéticos, etc.), y generalmente simples (modelos poliédricos en 3D o poligonales en 2D), frente a descripciones simbólicas que combinan información cualitativa (forma global de los objetos, protuberancias, etc.) con una cierta dosis de información cuantitativa (centro de masas y medidas globales), para el enfoque **A2**.

Asociados al primer enfoque **A1** existen importantes contribuciones, gran parte de las cuales se examinarán en el capítulo III. Entre ellas sobresalen las de Nguyen [Nguyen, 88], quien introduce algoritmos para la construcción del agarre cumpliendo la denominada clausura de fuerza⁷ (*force-closure*), a partir de la forma de los objetos a manipular (polígonos o poliedros). Un resultado a destacar será el uso del modelo de rozamiento de Coulomb (representado por un cono) para la determinación del agarre conocido el coeficiente de rozamiento entre la pinza del robot y el objeto. El agarre óptimo de un polígono convexo puede reducirse al caso del triángulo y éste es el caso estudiado en [Markenscoff & Papadimitriou, 89], pero suponiendo la existencia de una pinza con tres dedos; es

⁷ Un agarre posee clausura de fuerza si cualquier fuerza y par aplicado externamente al objeto puede cancelarse por algún conjunto de fuerzas positivas en los dedos de la pinza [Blake, 95].

decir, un mínimo de tres dedos son los requeridos para asegurar la clausura de fuerza en el plano. Para contornos de curvas suaves [Chen & Burdick, 92] demostraron que dos dedos son suficientes. Otro trabajo interesante es el de Peshkin [Peshkin & Sanderson, 85], que introduce un algoritmo de coste lineal para la búsqueda de vértices visibles en un polígono de aplicación directa a la generación del agarre, supuesta una morfología poligonal. Otros trabajos hacen referencia al problema del agarre en el plano considerando que no existe rozamiento [Trinkle et al., 88], o teniendo en cuenta su existencia [Faverjon & Ponce, 91] y además sobre objetos planos modelados por curvas paramétricas polinomiales. Por último, destacar la importancia de la estabilidad en relación al agarre que, desde un punto de vista cinemático, es tratado en diferentes trabajos [Wolter et al., 85], [Montana, 91].

Por otra parte, en el enfoque **A2** las principales aportaciones provienen de trabajos como [Iberall, 87], [Cutkosky, 89] o [Hershkovitz et al., 95] y, en general procuran un nivel de razonamiento más abstracto que el anterior.

Los enfoques que acabamos de presentar no hacen mención explícita al uso de la percepción. Nuestro interés se centra precisamente en la incorporación de la visión para poder atacar el problema del agarre desde una perspectiva real. El objetivo final será validar nuestros algoritmos de generación del agarre en un sistema real que lo integrará como un módulo más del mismo.

Con esta idea en mente podemos hablar de dos enfoques [Sanz et al., 96]:

- B1.** Sistemas basados en reconocimiento de formas.
- B2.** Sistemas que no utilizan dichas técnicas (no intentan reconocimiento).

Entre los sistemas que siguen el enfoque **B1** está el propuesto por Stansfield [Stansfield, 88]. En él se utiliza una base de modelos construida previamente, entre la información asociada al modelo se encuentra la relativa al agarre óptimo (a definir) en cada caso. Por otra parte, el enfoque **B2** no cuenta con información a priori sobre los modelos y aquí existen, a nivel práctico, dos líneas de ataque del problema: una **B2.1** que utiliza representaciones de los contornos completos de los objetos [Kamon et al., 96] y otra **B2.2** que trabaja sin dicha representación [Jarvis, 88]:

"Con vistas al agarre de un objeto no es necesario su reconocimiento previo; sólo se necesita cogerlo" [Stansfield, 91].

Este último enfoque **B2.2** es el más indicado, en teoría, para ambientes no estructurados, donde los objetos pueden aparecer defectuosos, en contacto entre sí o parcialmente visibles (existe solapamiento) [Jarvis, 88]. Una ventaja de **B2.2** sobre **B2.1** es un menor coste computacional asociado; sin embargo el enfoque **B2.1** será potencialmente más interesante en general, debido a que la

representación del objeto deja una vía abierta para el posterior reconocimiento de supuesto interés en etapas posteriores. Dicho interés radica en la conexión que habrá de existir entre la tarea que nos ocupa (la generación del agarre) y un planificador de tareas de alto nivel. Es decir, la generación del agarre constituye sólo una etapa subsumida en la planificación del agarre que, a su vez forma parte de planificadores más generales como el de ensamblado, donde será esencial la interpretación de la escena mediante técnicas de reconocimiento de formas, indispensable en la fase de modelado del mundo que utiliza como entrada el planificador para disparar de forma automática las secuencias de ensamblado programadas.

Nuestro enfoque seguirá esta última tendencia **B2.1**, siendo el trabajo de Kamon ([Kamon et al., 94] y [Kamon et al., 96]) el más próximo al nuestro, en cuanto a la estrategia seguida para la determinación de los puntos de agarre. En líneas muy generales podemos decir que en este trabajo prevalece el enfoque comentado anteriormente, inspirado en el agarre humano, teniéndolo en cuenta para incorporar condiciones de estabilidad en la determinación del agarre.

2.3. Ejecución del agarre

Hemos dejado claro, a partir de lo expuesto en el capítulo anterior, que nuestra estrategia de agarre requiere dos ingredientes básicos:

- (1) Información visual, como entrada a nuestro módulo de control, y
- (2) Enfoque reactivo, inspirado en un sistema biológico, para la construcción de los algoritmos de control.

Esta problemática está directamente relacionada con el denominado: "problema del control en la coordinación ojo-mano" (véase la figura 2.7). En realidad, el problema de la coordinación ojo-mano, en general, es una vastísima área de investigación en robótica, cubriendo diversos aspectos del mismo, desde métodos de calibrado [Zhuang et al., 95], hasta las estrategias de control propiamente dichas [Wijesoma et al., 93].

Una revisión sobre todos los aspectos elementales que intervienen en el control basado en visión de robots manipuladores, puede encontrarse en [Corke, 93].

Puntualicemos que estamos interesados, exclusivamente, en los sistemas de implantación real, y no en aquéllos que hacen uso de la simulación. Por tanto, aparecen problemas añadidos: el ruido (inherente a la captura de datos por los sensores y al propio entorno), y la incertidumbre (debido a imprecisiones en el modelo cinemático, calibrados previos, etc.).

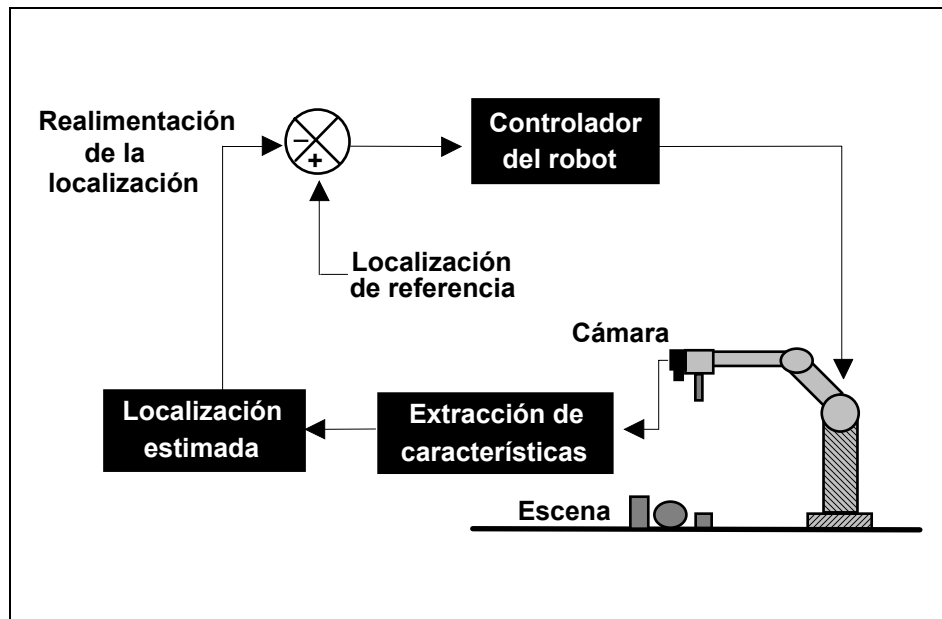


Figura 2.7. Representación gráfica del modelo genérico de control del manipulador basado en visión: coordinación ojo-mano. Nótese que la localización estimada puede darse tanto en coordenadas de un sistema externo (cartesiano) como en coordenadas propias del robot (espacio visual).

Otra observación a tener en cuenta será la posición de la cámara relativa al robot manipulador. En principio nosotros disponemos de una sola cámara; por tanto, las posibilidades son: **C1** cámara fija (estática) versus **C2** cámara en el elemento terminal (móvil). En este sentido, existen defensores de las dos estrategias, por ejemplo en [Wijesoma et al., 93], se defiende **C1** diciendo que en este caso se aúnan los objetivos de capturar información de la escena en conjunto, y guiar la mano del robot, integrado en su estructura de control. Además, requiere (según los autores) un menor esfuerzo computacional. Por otro lado, en [Corke, 93], se muestran los beneficios de **C2**, afirmando que entre éstos se encuentra la habilidad de evitar oclusiones, resolver ambigüedades e incrementar la precisión. Es indiscutible que ambos enfoques, **C1** y **C2**, poseen ventajas e inconvenientes, y una ventaja clara del enfoque **C2** será que, hasta el preciso momento del agarre, la cámara puede obtener información de la escena sin que el manipulador oculte la misma, parcialmente siquiera, debido a sus desplazamientos, pudiendo alternar, además, la visión global o de conjunto con la local.

Nosotros trabajaremos con la cámara situada en el elemento terminal, pero tomando imágenes desde únicamente dos alturas: una *máxima*, para facilitar el movimiento de aproximación al objeto de interés a partir de la panorámica de la

escena completa y una *media*, que iterativamente (máximo tres iteraciones) posicionará la mano del manipulador justo sobre la pieza a manipular (a detallar en el capítulo VI).

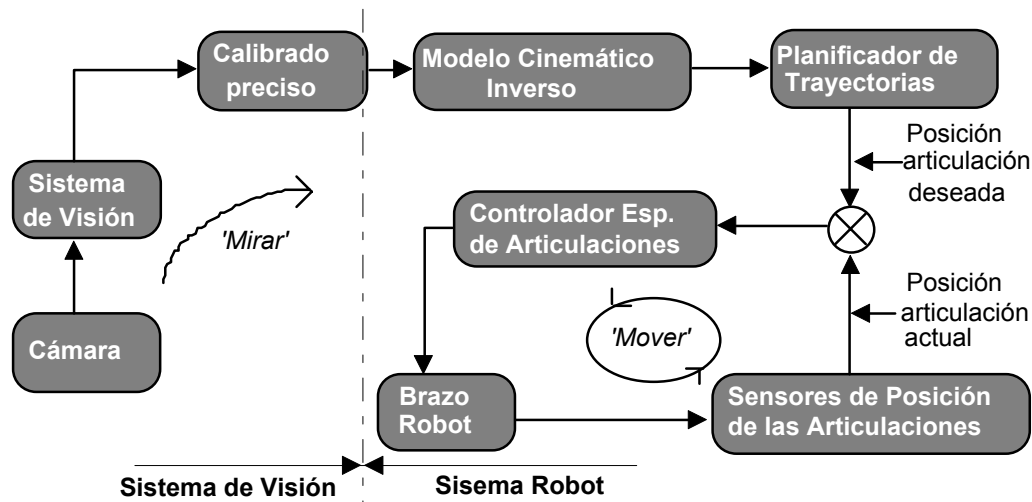


Figura 2.8. Adaptado de [Wijesoma et al., 93]. Representación gráfica de la estrategia de control: "mirar y mover" (*look-and-move*).

Acerca del calibrado, si procede, una buena fuente de información relacionada con el proceso de calibración de cámara podemos encontrarla en el trabajo de Tsai [Tsai, 86].

Es interesante darnos cuenta de las implicaciones que el uso de la realimentación visual provoca en los algoritmos de control. Para ello consideremos un enfoque anterior: "mirar y mover" (*look-and-move*) [Weiss, 84], el cuál se visualiza en la figura 2.8. En este caso no existe realimentación visual, y la señal de error del controlador del robot se construye a partir de sensores montados en cada una de las articulaciones del robot, de manera que se recoge en todo instante la configuración actual del robot a contrastar con la señal de referencia, capturada por el sistema de visión. En resumen, deberá existir servocontrol y, además será altamente dependiente del modelo cinemático del robot y de los calibrados previos (de cámara, etc.).

Por contra, cuando se tiene en cuenta la realimentación visual, integrada en el bucle de control del movimiento (*direct vision feedback*) [Wijesoma et al., 93] (véase la figura 2.9) se tienen dos ventajas fundamentales sobre el esquema anterior: 1ª) se eliminan los errores causados por el calibrado del sistema de visión ya que ahora controlamos con el mismo sensor (la cámara) las localizaciones del elemento terminal (TCP ó Tool Center Point) y del objeto (objetivo); y 2ª) se reducen los errores debidos a imprecisiones en el modelo cinemático, ya que

ahora el modelo cinemático inverso se utiliza no sólo para evaluar los ángulos de las articulaciones en el estado objetivo, sino también para estimar los mismos durante el movimiento, que corresponden a la posición del TCP medido por la cámara [Hashimoto, 93].

Un aspecto clave en el posicionamiento mediante visión de un manipulador, recae en la habilidad del programador para encontrar una correspondencia adecuada entre las coordenadas visuales y motoras, o bien, proveer al sistema de la capacidad de construir ésta de forma automática. El enfoque adoptado para resolver este problema (sistemas de posicionamiento del brazo del manipulador a partir de información visual) contempla dos categorías:

Aquéllos basados en el conocimiento de una relación analítica aproximada de la transformación entre un sistema de referencia cartesiano externo y el sistema de referencia de la cámara; y

Aquéllos basados en la adquisición del conocimiento sobre la transformación directa cámara-robot (píxeles-articulaciones).

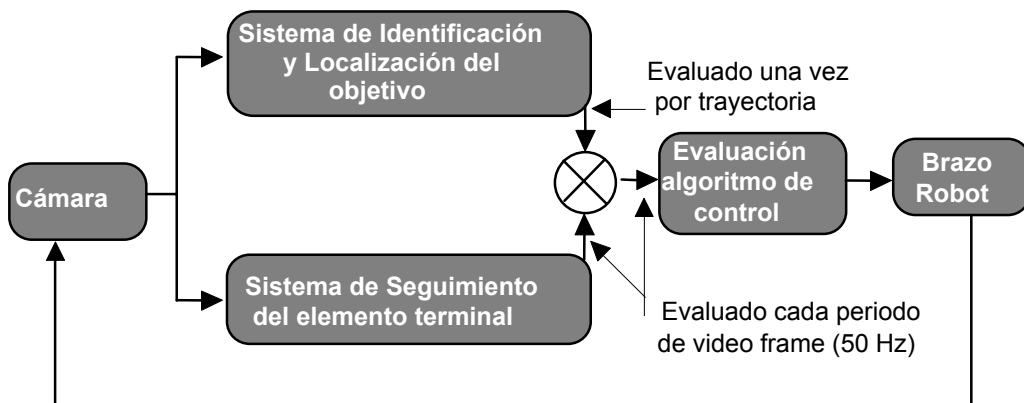


Figura 2.9. Adaptado de [Wijesoma et al., 93]. Representación gráfica de la estrategia de control: "realimentación visual directa" (*Direct vision feedback*).

Entre los primeros podemos citar [Lozano-Pérez et al., 92]; los cuales están basados en procedimientos de calibrado de cámara, como [Tsai, 87], y necesitan el conocimiento de la cinemática inversa del manipulador. El otro enfoque recoge una gran cantidad de pares válidos {(puntos visuales), (posiciones motoras)} que son procesados por algoritmos de aprendizaje para mejorar la calidad de la aproximación. Estos usan redes neuronales [Kroese et al., 93] o el modelo del

computador cerebelar aritmético (CMAC)⁸ [Albus, 81], [Tham & Prager, 92] y otros usan métodos de regresión [Atkeson, 91]. Casi todos (excepto [Clocksin & Moore, 89] y [Conkie & Chongstitvatana, 90]) han sido probados únicamente en robots simulados. El enfoque seguido, adaptado de [Domingo, 93] y [Domingo & Pelechano, 96], cae dentro de este grupo, pero la diferencia está en que el conocimiento no es adquirido, sino introducido a través de un proceso de medida de los valores de la transformación y sus derivadas parciales en un conjunto de puntos seleccionados, debiendo aplicarse realimentación para mejorar la precisión. Este método será explicado en detalle en el capítulo VI.

Resumiendo:

- La aplicación del paradigma comportamental a este problema intenta prescindir en lo posible del uso de un sistema de referencia externo, para establecer la correspondencia necesaria entre coordenadas naturales del sistema de visión (píxeles) y coordenadas de robot (ángulos de las articulaciones).
- El uso de la realimentación visual es absolutamente imprescindible si se quiere alcanzar la robustez necesaria para evitar la dependencia, en cuanto a precisión se refiere, de calibrados y modelos cinemáticos previos, así como dotar al sistema de capacidad reactiva.

⁸ Cerebellar Model Arithmetic Computer

Capítulo III

Caracterización del Agarre

En este capítulo se abordarán, en síntesis, los fundamentos matemáticos que modelan el agarre desde sus dos perspectivas tradicionales: La geométrica y la mecanicista, así como las directrices básicas que sustentan el enfoque seguido en esta tesis.

3.1. Introducción

La acción de agarrar un objeto puede definirse como el emplazamiento de la pinza relativo al objeto y se caracteriza por un conjunto de contactos llamados agarre (*grasp*).

Es interesante tener en cuenta qué posibilidades existen en el agarre de un objeto mediante una pinza convencional de dos dedos plano-paralelos, en cuanto a la aproximación de dicha pinza se refiere (véase la figura 3.1). En principio de las tres posibilidades que se muestran en la figura 3.1, elegiremos la denominada *grip* (empuñar). En el capítulo VI se muestra al manipulador en acción siguiendo la aproximación del agarre aquí propuesta. Puntualicemos que debido a que la microcámara empleada se ensambla en la muñeca, con la aproximación propuesta anteriormente se consigue que el eje óptico sea ortogonal a la escena todo el tiempo, lo cuál simplificará todo el razonamiento geométrico empleado (véase el capítulo V).

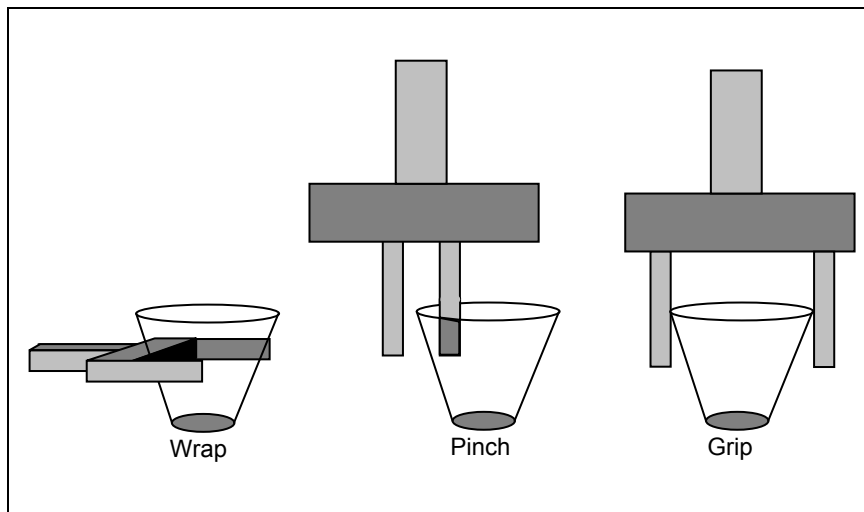


Figura 3.1. Adaptado de [Tan & Schlimler, 93]. Modelos de agarre diferentes según la vía elegida de aproximación al objeto: Envolver (Wrap), Pinzar (Pinch), y Empuñar (Grip).

Dos enfoques claves de la investigación sobre el agarre (*grasping*), siguiendo uno de los trabajos más citados en la bibliografía actual al respecto [Pertin-Troccaz, 89], son:

Geometría del Agarre. Tiene que ver, fundamentalmente, con el razonamiento espacial y morfológico.

Mecánica del Agarre. Incluye estudios sobre estabilidad, equilibrio, resistencia al deslizamiento, giro, etc., según el tipo de contacto definido con los dedos de la garra, y otros aspectos cinemáticos del agarre.

La generación y/o análisis del agarre forma parte de una problemática más general, conocida en la literatura de investigación como "Planificación del Agarre" (*Grasp Planning*). El cual engloba, a su vez otros dos tópicos, además del anterior:

Reaching Estudia como conviene alcanzar el agarre definido, generando trayectorias libres de colisiones.

Regrasping Se trata de desencadenar una serie de agarres intermedios consecutivos, hasta encontrar el agarre final, con vistas a superar una posible incompatibilidad entre el agarre inicial y la tarea a desarrollar (e.g. posible ensamblado final).

Además, en el caso de que la garra a utilizar sea una "mano diestra" o "de múltiples dedos" (*dextrous hand* o *multifingered hand*), aparece otro tópico a tener en cuenta:

Preshaping Preadaptar la mano de acuerdo a la morfología del objeto y/o la tarea a desarrollar en la manipulación.

El trabajo aquí desarrollado se centra exclusivamente en la generación del agarre. El cual consiste, en esencia, en generar dicho agarre de acuerdo con unos criterios específicos, comparación de la forma entre el objeto y la pinza, estabilidad, etc.

En cada uno de los apartados que siguen a continuación para perfilar la configuración del modelo de agarre elegido, se dará una breve comparativa para dejar constancia de otras alternativas referentes a trabajos de investigación que intersectan en alguna medida con el aquí presentado. Dichos trabajos serán fundamentalmente dos: "HANDEY" [Lozano-Pérez et al., 92] y "Learning to Grasp Using Visual Information" [Kamon et al., 96]¹. La razón es clara, el primer trabajo constituye seguramente una implementación única en su especie como planificador de tareas más completo que existe en la actualidad [Pertin-Troccaz, 92], constituyendo una referencia ampliamente conocida y conveniente, por tanto, para contrastar el modelo de agarre aquí desarrollado. Siendo el segundo trabajo la fuente de investigación más cercana a nuestros objetivos, en cuanto a la determinación de puntos de agarre se refiere.

3.2. Condiciones Iniciales.

3.2.1. Herramienta.

En el presente trabajo nos centraremos, exclusivamente, en la pinza elemental de dos dedos plano paralelos. La más comunmente utilizada en los robots manipuladores industriales, de investigación y educación, de nuestros días. Aquí queda contemplada también la posibilidad de la multipinza formada por varios pares de pinzas elementales del tipo anterior.

Supondremos conocida, a priori, la geometría de dicha pinza². Un modelo simplificado de la misma puede verse en la figura 3.2.

¹ En lo que sigue abreviaremos este trabajo mediante LGUVI.

² Véase el apéndice B, donde se describe la pinza real empleada en la implementación final.

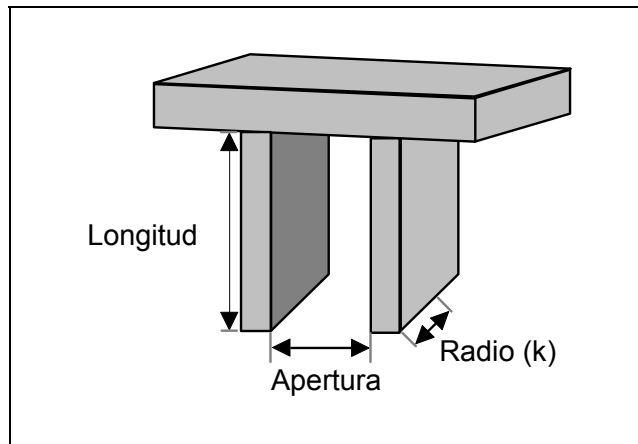


Figura 3.2. Modelo geométrico simplificado de la geometría de la pinza de dos dedos plano-paralelos, empleado para la generación del agarre.

Este modelo de pinza elegido por nosotros será muy similar al propuesto por otros investigadores [Kamon et al., 96], [Lozano-Pérez et al., 92] (véase la figura 3.3).

Puntualizemos que todo el trabajo desarrollado estará mediatizado por el tipo de pinza empleada por el robot. En nuestro caso, el modelo considerado (véase la figura 3.2.) tiene la ventaja, frente a modelos más sofisticados, de un menor coste: 1º) computacional, relativo al razonamiento geométrico necesario para la determinación del agarre y posterior ejecución (existencia de requerimientos mínimos de control), y 2º) económico, en la actualidad manos diestras como las de "Salisbury" o de "Utah" (véase el capítulo II) sólo están al alcance de unos pocos centros de investigación en el mundo. Por contra existirá el inconveniente, como veremos en el apartado 3.2.3, de la necesidad de que los objetos sean poco complicados, geométrica y topológicamente hablando; e.g. el agarre de un tetraédro o similar sería una tarea imposible de realizar con nuestro modelo de pinza.

3.2.2. Sensorización.

Es interesante recordar que la habilidad de un robot para desenvolverse en un entorno no estructurado y cambiante depende en gran medida de su capacidad de percibir éste mediante información sensorial. En realidad "sentir" no es suficiente, hace falta transformar dicha información sensorial en información acerca del entorno. Se puede definir la percepción como [Khatib et al., 92]:

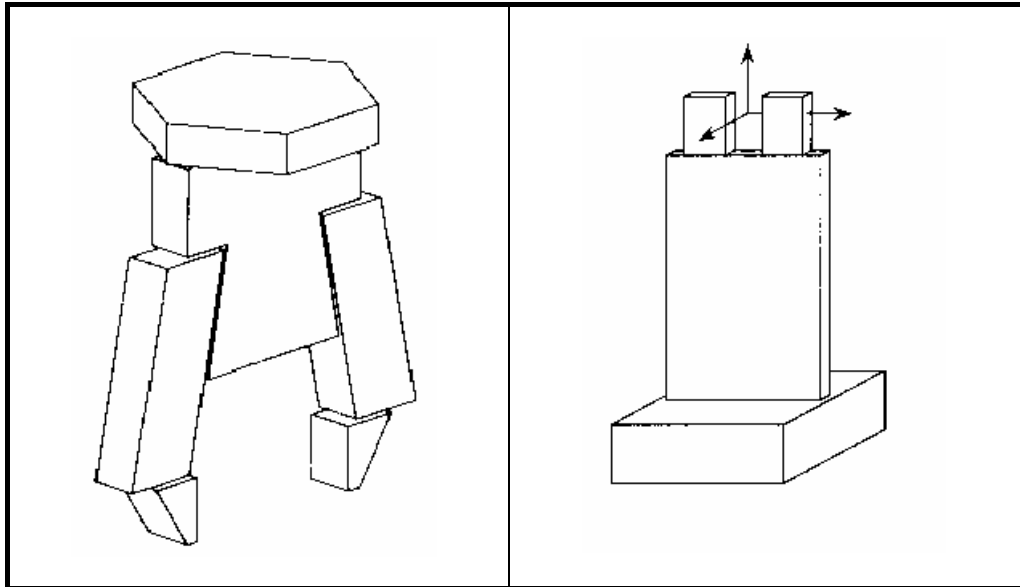


Figura 3.3. Adaptado de [Lozano-Pérez et al., 92]. Modelos de pinzas de dedos plano-parallelas: (Izquierda) un modelo preciso, y (Derecha) modelo simplificado utilizado en dicho planificador de agarre (HANDEY).

"La conexión inteligente entre la información sensorial y el entorno (modelos 3D)".

Algunos investigadores [Khatib et al., 92] clasifican la información sensorial (IS) en dos grandes líneas:

IS-1 Contacto (*Contact Sensing*). Sensores táctiles y de fuerza, fundamentalmente.

IS-2 No-Contacto (*Non-contact Sensing*). Información visual, principalmente. A dividir en:

IS-2.1 Descripción de Imágenes (*Image Understanding*). Proceso que establece relaciones con objetos 3D a partir de una o más imágenes 2D.

IS-2.2 Interpretación de Escenas (*Scene Understanding*). Proceso de obtención de una descripción coherente de una escena a partir de medidas geométricas diversas, posiblemente con la ayuda de modelos geométricos de los objetos.

Sensores de uso frecuente en el problema del agarre (*grasping*) que nos ocupa, serían los sensores de profundidad (*range sensor*) [Francois, 91], los sistemas de visión estéreo [Bard et al., 95], o los sensores táctiles y de fuerza [Howe & Cutkosky, 92]. Los dos primeros orientados al problema del acercamiento "grueso" en 3D y, los últimos al del movimiento fino.

En cuanto a su aplicación industrial, es en el tópico IS-2.1 donde se están realizando los principales avances en la actualidad. Recordemos, no obstante, la necesidad al aplicar estas técnicas al campo de la robótica de la restricción de operar en tiempo real. Esto ha conllevado una dificultad añadida en el desarrollo de dichos sistemas para aplicaciones robóticas; e.g. hardware específico, sensores inteligentes, etc.

La pregunta que subyace es porqué los sensores de contacto (IS-1) han encontrado hasta ahora poca aceptación en aplicaciones robóticas industriales [Howe & Cutkosky, 92]. Una posible razón es el uso inicial dado a los sensores táctiles como captadores de información geométrica. En este sentido las técnicas basadas en la descripción de imágenes (IS-2.1) se han mostrado superiores en la captura de información relativa a la morfología de los objetos y a su localización. Esto ha decantado el tipo de sensorización dominante en la industria hacia aquellos sistemas basados en visión.

Puntualicemos que al igual que los seres humanos utilizamos el uso del tacto para capturar propiedades mecánicas del entorno, imposibles para el sentido de la vista, sería conveniente extrapolar este enfoque al robot. Así se trabaja actualmente en la percepción de propiedades mecánicas tales como conformidad (*compliance*), rozamiento (*friction*), texturas (*texture*), etc. En cualquier caso, ambos tipos de información sensorial: IS-1 e IS-2, serán necesarios en tanto en cuanto representan aspectos complementarios de la percepción. En una tarea completa de manipulación la fase de precontacto (justo antes del agarre) será cubierta de modo preferente por información visual, mientras que será en la fase de contacto (los dedos de la pinza están tocando el objeto) cuando los sensores IS-1 muestren toda su potencialidad, debiendo controlar pequeñas fuerzas ejercidas y desplazamientos infinitesimales.

En el presente trabajo, con el fin de manipular los posibles objetos situados en el área de trabajo, necesitaremos establecer una cadena estímulo-acción que, a su vez, requerirá de una percepción visual (en nuestro caso) del entorno, capaz de entregar una representación adecuada a nuestros objetivos. La representación requerida que mejor se adapta a la tarea orientada al razonamiento geométrico necesario para la generación y análisis del agarre es el contorno de las imágenes 2D de los objetos presentes en la escena. Para ello, se ha considerado suficiente utilizar, como fuente única de información sensorial exteroceptiva, un sistema de visión consistente en una tarjeta digitalizadora y una cámara CCD, montada en la muñeca del robot³ que tomará siempre imágenes que serán proyecciones

³ En el capítulo VI se muestran imágenes reales del sistema completo (cámara más robot).

superiores planas de los objetos. El tamaño de las imágenes es de 256×256 píxeles y la resolución de 64 niveles de gris. Como ya dijimos anteriormente, se garantiza que la posición de la muñeca obliga a que el eje óptico sea perpendicular a la escena, simplificando el razonamiento geométrico posterior. Todos los aspectos relacionados con la adquisición, segmentación y tratamiento de imágenes se relegan al capítulo IV.

La sensorización equivalente en HANDEY es llevada a cabo mediante técnicas de triangulación basadas en barrido láser [Lozano-Pérez et al., 87] que permiten construir un mapa de profundidad de la escena, representado como un vector de dos dimensiones cuyos índices codifican la posición y cuyos elementos representan la altura de una superficie en dicha localización.

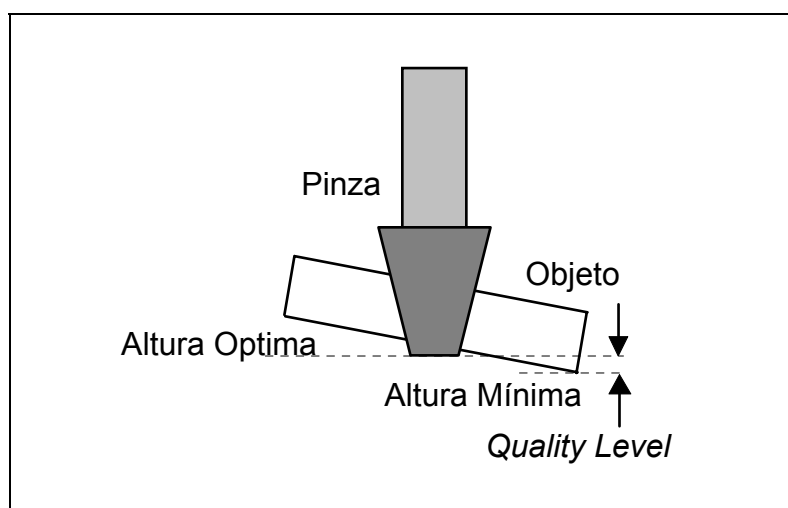


Figura 3.4. Vista lateral de un objeto sujetado por la pinza, adaptado de [Kamon et al., 96]. El nivel de confianza asociado al agarre (*Quality Level*), depende de la diferencia entre las alturas óptima y mínima.

Por otra parte, el otro trabajo referenciado, LGUVI, utiliza dos cámaras CCD estáticas. Una ofrece una vista global superior de la escena completa, y la otra captura una imagen lateral de la misma, permitiendo determinar un nivel de calidad asociado a cada agarre (véase la figura 3.4). Es decir, dado un objeto aislado en la escena, el sistema toma una imagen superior con la primera, usa la imagen para extraer dos puntos de agarre del contorno, efectúa un intento de agarre con una pinza de dedos plano-paralelos, y asigna un grado de confianza (*Quality Level*) a dicho intento mediante la cámara lateral. Como vemos, utiliza de forma exclusiva información visual, tanto para la fase de pre-contacto como para la de post-contacto. La ausencia de información táctil en esta última etapa se suple con un mecanismo de aprendizaje (prueba y error), a partir de la información

visual del agarre obtenida por la cámara lateral. El tamaño de las imágenes es de 576×768 píxeles y la resolución de 64 niveles de gris. La salida de esta etapa de procesamiento es una imagen segmentada del objeto con su centroide y la dirección del eje de mínima inercia. La entrada, un objeto aislado altamente contrastado del fondo de la escena, pues se utiliza un nivel fijo de umbral. A partir de aquí se determina el contorno del objeto, formando una curva cerrada. Mediante un proceso de adelgazamiento (*thinning*) se garantiza que cada píxel del contorno posea exactamente dos vecinos.

3.2.3. Universo de Trabajo.

¿Qué objetos son manipulables por éste tipo elemental de pinza plano-paralela, con la información sensorial anteriormente citada?

En general, aquellos objetos cuasi-rígidos⁴ que presenten en su contorno más externo y, aproximadamente, a lados opuestos del centro de gravedad y cercano a él, puntos y/o regiones que admitan coincidencia con la geometría de los dedos de la pinza proyectada sobre ellos. Además, como únicamente se dispone de información 2D, deberán cumplirse una serie de condiciones iniciales que garanticen, a partir de características extraídas mediante el sistema de visión, la estabilidad del agarre generado. Es decir: ¿Podemos garantizar la estabilidad del agarre a partir de sólo dos puntos pertenecientes al contorno de la proyección plana superior de un objeto?. En general, la respuesta es no. En la figura 3.5, se observa un ejemplo que ilustra este problema. Un rebaje oculto a la proyección plana superior del objeto obtenida por la cámara genera el consiguiente error en la obtención del centroide, situandolo en "cdm1", en lugar de "cdm2", como correspondería a su realidad física.

Además de la rigidez y de la ausencia de huecos y agujeros ocultos, otra cualidad importante que habrán de cumplir nuestros objetos será la homogeneidad, es decir, serán intratables objetos como el martillo de carpintero que, debido a que posee dos materiales de densidades muy diferentes, madera y acero, impiden la extracción automática del centroide, físicamente incorrecto si se determina exclusivamente a partir de su forma. Así pues, toda aquella información que no sea posible extraer mediante visión no podrá participar en nuestro esquema, haciendo intratable su uso.

En resumen, nuestro universo de trabajo estará formado por todos aquellos objetos que poseyendo, al menos, una posición estable sobre el plano horizontal de la escena, permitan inferir a partir de su morfología, una representación física correcta. Esto se traduce en la práctica, en la coincidencia del centroide, generado a partir de la imagen 2D obtenida, con la proyección sobre su plano superior del centro de masas (cdm) real del objeto. Podrán ser concavos o convexos, con o sin agujeros (visibles), con centroides interiores o exteriores a su contorno, etc.

⁴ Recordemos que un sólido rígido es aquél en que las distancias mutuas entre todos sus puntos permanecen constantes.

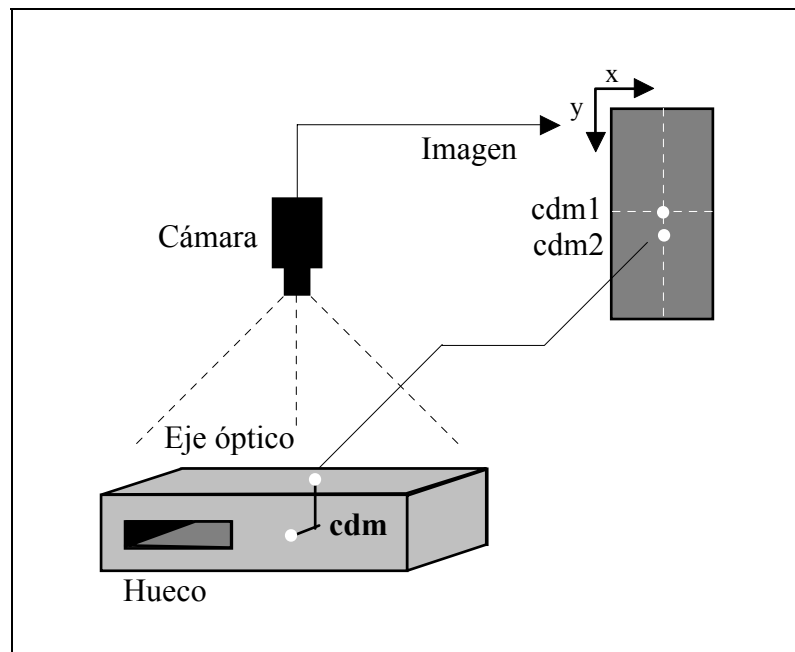


Figura.3.5. Ejemplo de objeto intratable con las restricciones de nuestro sistema. El objeto presenta un rebaje, por su cara oculta, no detectable a partir de la información visual disponible.

Los objetos son desconocidos a priori y no se intentará en este trabajo reconocimiento posterior alguno, pero se dejará una vía abierta a ello.

En contraste, el universo de trabajo en HANDEY estará formado por cualquier pieza susceptible de ser aproximada mediante modelos geométricos 3D asociados a una posición particular. HANDEY es capaz de manipular objetos suaves, no-convexos, e incluso hacer uso de agujeros presentes en dichos objetos para generar el agarre. El único requerimiento es que exista una aproximación poliédrica adecuada para cada objeto a ser manipulado. No obstante, este enfoque conlleva un alto coste computacional que lo hace inasumible para los objetivos aquí planteados.

Por otra parte, en LGUVI se utilizan dos tipos de objetos:

1. Sintéticos. Generados de forma aleatoria a partir de conos generalizados.
2. Reales. Objetos diversos (e.g. piedra, pelota, taza, etc.), en los que se supone que el centroide calculado por el sistema de visión será

aproximadamente coincidente con el centro de gravedad (cdg) real del objeto (o mejor, con su proyección plana superior).

Una restricción importante asociada a ambos tipos de objetos (1 y 2) del trabajo anterior, es que todos presentan simetría especular respecto de la dirección del eje de mínima inercia⁵ (*I_{min}*), lo cual, a diferencia de nuestro trabajo, es una simplificación decisiva para la etapa de razonamiento geométrico posterior. Otra observación a tener en cuenta es el tratamiento de los agujeros asociados a los objetos, los cuales son "rellenados" con píxeles del mismo nivel de gris que los objetos segmentados. Esto conllevaría, en principio, que algunos objetos fueran intratables debido al consiguiente cálculo erróneo del centroide. No obstante, dicho problema se reduce en la medida en que los objetos presenten una mayor simetría especular (respecto de la dirección de *I_{min}*), como hipótesis de partida.

3.3. Modelo de Agarre Elegido.

Tal y como se dijo en el apartado 3.1, existen dos aproximaciones básicas para modelar el agarre: la geométrica y la mecánica. El trabajo aquí presentado seguirá, básicamente, una aproximación geométrica, pero teniendo en cuenta características mecánicas relacionadas con el problema de la estabilidad y la condición de no-deslizamiento entre pinza y objeto, en presencia de rozamiento. Existen algunos antecedentes que han puesto de manifiesto la conexión existente entre ambas aproximaciones [Montana, 91], [Rao & Goldberg, 94].

El problema del agarre, en general, es encontrar superficies en un objeto, y una configuración de los dedos del robot, satisfaciendo las siguientes tres condiciones [Peshkin & Sanderson, 86]:

- (1^a) Los dedos deben estar en contacto con el objeto.
- (2^a) La configuración debe ser alcanzable (i.e. debe existir una trayectoria libre de colisiones para llevar al robot a la configuración del agarre).
- (3^a) El objeto debe quedar estable una vez agarrado.

La característica (2^a) es estudiada en profundidad por otros investigadores [del Pobil & Serna, 95] y no será tratada en el trabajo aquí presentado. La (1^a) se inserta dentro de una problemática denominada, genéricamente, "contacto entre dos cuerpos" [Mason & Salisbury, 85], y contempla, entre otras cosas, la posible existencia de rozamiento que nos permitirá incorporar la condición de no-deslizamiento, indispensable en nuestro modelo de agarre. Y por último, la (3^a) se

⁵ Este concepto será analizado en el capítulo IV.

relaciona con el otro factor clave que será considerado en este trabajo, la denominada "estabilidad del agarre" [Wolter et al., 85], [Montana, 91].

3.3.1. El Contacto entre dos Cuerpos.

Existen algunos trabajos dedicados en profundidad a este importante problema [Mason & Salisbury, 85]. Partiremos de la siguiente argumentación [Mason & Salisbury, 85]: cuando dos cuerpos rígidos se acercan hasta tocarse en uno o más puntos, aparece una aplicación biyectiva entre ambos. Se denomina contacto a la colección de puntos adyacentes donde ocurre lo anterior sobre un área contigua. Dos áreas de contacto disjuntas se considerarán dos contactos distintos. En el límite, un punto aislado se considerará un área de contacto.

En general, considerando las tres características posibles de una superficie en relación al contacto: punto, línea y plano, existen nueve posibilidades para los contactos entre pares de superficies (véase la tabla 3.1). Obsérvese que en relación a los grados de libertad existentes en el contacto entre dos cuerpos es irrelevante qué característica pertenece a cada cuerpo, lo que importa son las características que intervienen en el contacto con independencia del cuerpo a que pertenezcan. Asimismo puntualicemos que de los posibles contactos representados en la tabla 3.1, únicamente existirán tres pares estables entre las características de los dos cuerpos, es decir:

- Contacto puntual (punto con plano, plano con punto o línea con línea no paralela).
- Contacto lineal (línea con plano o plano con línea).
- Contacto plano (plano con plano).

Por otra parte, el movimiento relativo entre dos cuerpos limitado por uno o más contactos dependerá de:

Las restricciones impuestas por la forma de cada área de contacto.

Las localizaciones relativas de las áreas de contacto.

El efecto del rozamiento en cada área de contacto.

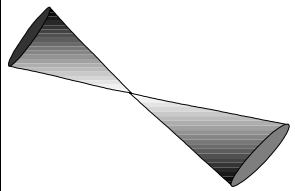
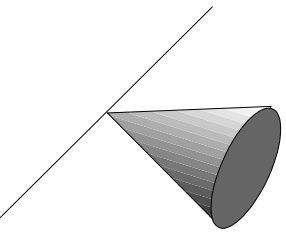
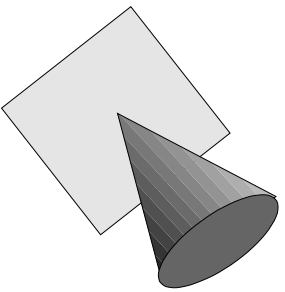
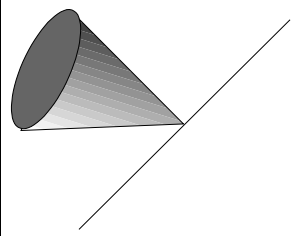
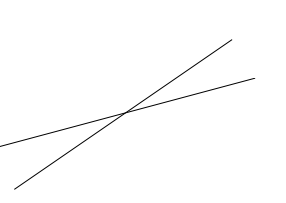
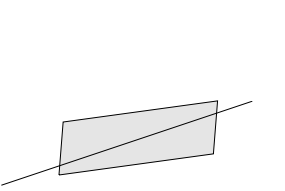
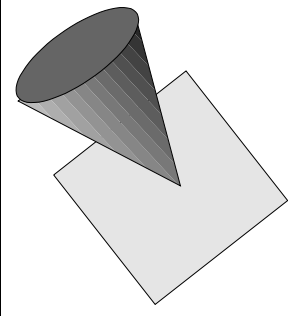
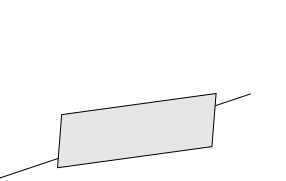
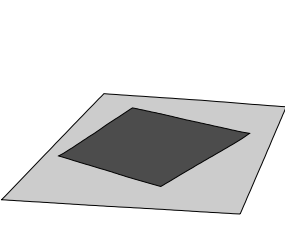
	Punto	Línea	Plano
Punto			
Línea			
Plano			

Tabla.3.1. Adaptada de [Mason & Salisbury, 85]. Los nueve posibles pares de características (punto, línea y plano) entre dos superficies.

Teniendo en cuenta lo dicho, pasamos a considerar a continuación los posibles tipos de contacto que se contemplan en el contexto de la planificación del agarre en robótica. Lo más general sería considerar el denominado contacto plano (plano con plano), pero para nuestros propósitos bastará con suponer, en principio, un punto de contacto donde se aplicará la fuerza resultante ejercida en el agarre. Básicamente, pueden considerarse en este último supuesto, los siguientes modelos de contacto entre pinza y objeto [Nguyen, 88], también denominados contactos primitivos:

- (1) **Punto de contacto sin rozamiento.** El dedo puede sólo ejercer una fuerza normal a través del punto de contacto. Se trata de una aproximación ideal (no existe en la práctica).
- (2) **Punto de contacto con rozamiento (*hard-finger*).** El dedo podrá ejercer cualquier fuerza dentro del cono de rozamiento⁶ con el punto de contacto.
- (3) **Punto de contacto con rozamiento (*Soft-finger*).** El rozamiento sobre el área de contacto permite al dedo ejercer momentos (o torques) puros, además de fuerzas puras dentro del cono de rozamiento.

La diferencia básica entre (2) y (3), reside en el hecho de que este último es una extensión del primero al caso 3D, tomando en consideración la existencia de momentos.

El agarre plano puede ser descrito [Murray et al., 94], considerando únicamente la transmisión de fuerzas entre un conjunto de puntos de contacto, como los del tipo (1) definido anteriormente, y el objeto. En una primera aproximación al problema, podemos suponer que el contacto entre los dedos y el objeto tendrá lugar en torno a dichos puntos de contacto (véase la figura 3.6).

Sin embargo, como ya se ha indicado, los puntos de contacto tipo (1), sin rozamiento, no existen en la práctica, son meros artificios matemáticos útiles sólo como primera aproximación a la realidad física del problema. No obstante, si existieran, únicamente ejercerían presión en dirección a la normal. El interés de este modelo radica en que no presupone la existencia de rozamiento, es decir, si no se conoce el coeficiente de rozamiento estático, μ , de los contactos en el agarre el modelo no se ve afectado puesto que la transmisión de fuerzas se realiza exclusivamente en el sentido de las normales a cada superficie. En la práctica éste último será difícil de garantizar debido a pequeños errores de posicionamiento, irregularidades en las superficies, etc.

Un modelo más acorde con la realidad será el definido anteriormente como tipo (2), considerando la existencia de rozamiento. El modelo más ampliamente utilizado que lo considera es el denominado modelo de rozamiento de Coulomb [Mason & Salisbury, 85] (véase la figura 3.7).

El modelo de rozamiento de Coulomb es un modelo empírico en el que las fuerzas ejercidas en dirección tangencial a la superficie del contacto son proporcionales a las fuerzas normales aplicadas, y la constante de proporcionalidad, μ , es función de los materiales puestos en contacto.

⁶ Se refiere a la Ley de Coulomb, explicada más adelante.

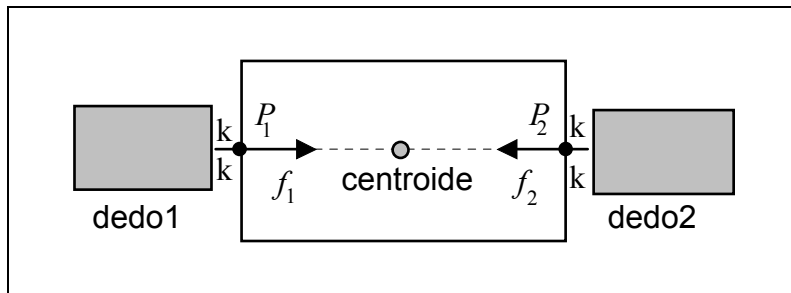


Figura.3.6. Adaptada de [Murray et al.,94]. Agarre 2D, con los puntos de contacto, P_1 y P_2 , y las fuerzas normales, f_1 y f_2 , debidas a los dedos de la pinza. Donde k representa el radio del dedo, de acuerdo con la figura 3.2.

Sean $\{f^t, f^n\} \in \mathcal{R}$, los módulos de las componentes tangencial y normal, respectivamente, de la fuerza ejercida sobre la superficie del objeto por el dedo de agarre. Dicha fuerza genérica, f , aplicada en un punto de contacto determinado, formará un cierto ángulo, θ , con la normal a la superficie (ó contorno) en dicho punto, admitiendo la descomposición en sus componentes tangencial y normal dada por $f^t = f \cos \theta$, y $f^n = f \sin \theta$, respectivamente. Por otra parte, debido a la existencia de rozamiento en el punto de contacto, la Ley de Coulomb [Coulomb, 1781], establece que dichas componentes están relacionadas mediante $f^t = \mu f^n$, donde $\mu > 0$, es el coeficiente de rozamiento estático. A partir de aquí se observa que si se cumple que $f^t > \mu f^n$, aparece deslizamiento, lo que implica que el rango de fuerzas tangenciales que pueden ser aplicadas en un contacto viene dado por,

$$f^t \leq \mu f^n \quad (3.1)$$

La ecuación 3.1, representa la denominada "condición de no-deslizamiento", y su interpretación geométrica se visualiza en la figura 3.7. Parece ser [Mason & Salisbury, 85] que el primero en dar dicha interpretación geométrica fué Moseley [Moseley, 1839]. Si se construye la normal a la superficie, la ley de Coulomb es equivalente a establecer que la fuerza de contacto total, f , formará un ángulo, $\theta = \arctan (f^t / f^n)$ con la normal. El conjunto de vectores formando el correspondiente ángulo con la normal representan un cono en el espacio denominado "cono de rozamiento", y al ángulo θ se le conoce como ángulo de rozamiento; dicho ángulo viene dado por,

$$\theta = \arctan (\mu) \quad (3.2)$$

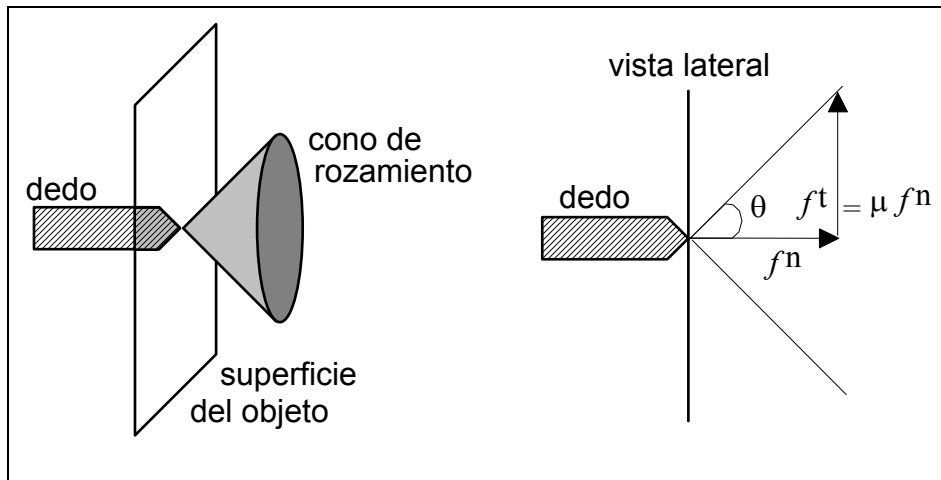


Figura 3.7. Adaptada de [Murray et al.,94]. Interpretación geométrica del modelo de rozamiento de Coulomb.

La ecuación (3.2) sintetiza la relación existente entre el coeficiente de rozamiento estático, μ , asociado a un contacto (dedo-superficie), y la geometría que se impone, definida por el correspondiente cono de rozamiento de ángulo total 2θ . Lo realmente interesante es que permite traducir la condición de no-deslizamiento anterior (véase la ecuación 3.1) a una representación puramente geométrica del contacto establecido.

Acero con acero	0.58	Madera con madera	0.25-0.5
Polietileno con acero	0.3-0.35	Madera con metales	0.2-0.6
Polietileno con Polietileno	0.5	Madera con piel	0.3-0.4
Caucho con sólidos	1-4	Piel con metal	0.6

Tabla 3.2. Adaptada de [Murray et al.,94]. Coeficientes de rozamiento estáticos, μ , para algunos materiales comunes.

Observación:

Empíricamente (véase la tabla 3.2), se sabe que la mayor parte de los contactos entre materiales que se dan en la práctica cumplen, $0.2 \leq \mu \leq 1$, lo que implica, $11^\circ \leq \theta \leq 45^\circ$. En nuestro caso, suponemos que los objetos son

desconocidos, pero lo anterior justifica el establecimiento de un umbral, $\theta = 10^\circ$ que, nosotros renombraremos⁷ como β , en el capítulo V, que permitirá vincular nuestro algoritmo con la condición de no-deslizamiento aquí planteada.

Una característica que ha servido de piedra angular en la mayor parte del trabajo de investigación realizado hasta ahora, en cuanto a modelar el agarre se refiere, es la denominada clausura de fuerza (*force-closure*) [Nguyen, 88], la cuál facilitará la búsqueda de la estabilidad en el agarre. Aunque lo relativo a estabilidad se trata en el apartado siguiente (3.3.2) al respecto, es conveniente considerar aquí el concepto de clausura de fuerza desde la perspectiva de su interpretación geométrica, pues está muy ligado al problema de modelar el contacto entre dos cuerpos, incluyendo la existencia de rozamiento en dicho contacto. Se dice que un agarre posee clausura de fuerza cuando puede resistir (permanecer en equilibrio) cualquier empuje (fuerza o momento de cualquier intensidad) aplicado al mismo. La clausura de fuerza se reduce a la denominada clausura de forma (*form-closure*) en el agarre [Lakshminarayana, 78] cuando únicamente se consideran contactos sin rozamiento; e.g. se demuestra [Markenscoff & Papadimitriou, 89] que un agarre plano de una forma poligonal necesita como mínimo cuatro puntos de contacto para alcanzar la clausura de forma.

Formalmente [Nguyen, 88], un agarre plano formado por dos puntos de contacto (P_1, P_2) con rozamiento se dice de clausura de fuerza si y sólo si la línea de agarre (P_1P_2) que conecta los puntos de contacto yace en el interior de ambos conos de rozamiento (véase la figura 3.8). Dichos puntos reciben el nombre de puntos de agarre antípodos (*antipodal point grasps*).

Inicialmente los trabajos de investigación se centraron en formas poligonales (2D) o poliédricas (3D) [Nguyen, 88]. Una primera extensión de este trabajo fué considerar formas planas cuyos contornos fueran curvas aproximadas por funciones polinómicas [Faverjon & Ponce, 91]. En esta línea otros autores [Rao & Goldberg, 95] consideraron formas planas cuyos contornos formados por segmentos de curva polinómicas (*algebraic parts*) eran aproximados por su envolvente convexa (*convex hull*) asociada; distinguiendo dos tipos de situaciones (véase la figura 3.9), cuando existe la denominada configuración de rozamiento (*frictional configuration*), en la cuál la línea de agarre, determinada por los dos puntos de contacto considerados, cae dentro de los dos conos definidos para cada uno de ellos, y cuando no existe dicha configuración. Como ya vimos, cada punto de contacto con rozamiento determina un cono de rozamiento de anchura 2α dirigido hacia el interior del objeto y simétrico respecto de la normal al objeto en dicho punto.

⁷ Se define β (capítulo V), como umbral angular en el paso MS2 del algoritmo desarrollado para la determinación automática de los puntos de agarre.

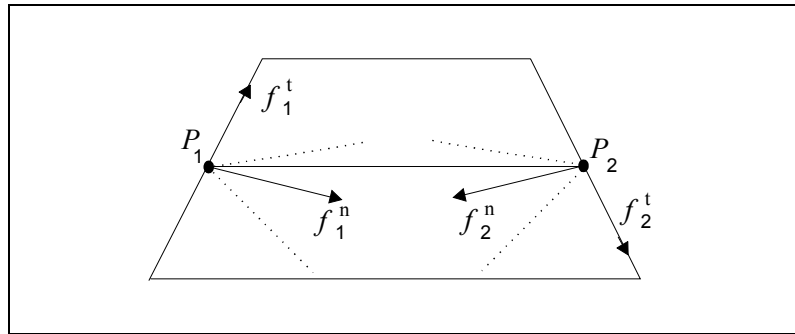


Figura 3.8. Adaptada de [Murray et al., 94]. Relaciones geométricas en el agarre plano con clausura de fuerza, considerando dos puntos de contacto (P_1, P_2) con rozamiento .

Una novedad aportada por otros autores [Chen & Burdick, 92] es el uso de curvas "B-spline" para modelar objetos de formas irregulares, así como la consideración del agarre por expansión (*expanding grasps*), además del convencional, por presión (*squeezing grasps*). Se observa que un objeto convexo puede cogerse únicamente mediante agarre por presión, mientras que los no-convexos admiten también agarre por expansión. Su objetivo se centra en determinar todos los agarres de puntos antípodos existentes mediante técnicas de optimización.

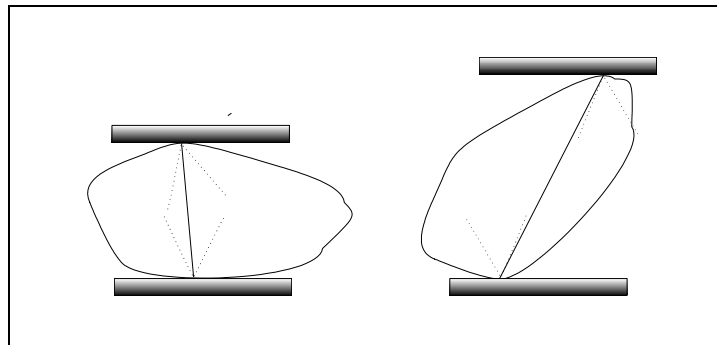


Figura 3.9. Adaptada de [Rao & Goldberg,95]. Interpretación geométrica cuando existe configuración de rozamiento (izquierda) y caso contrario (derecha).

En el trabajo presente, teniendo en cuenta lo dicho anteriormente, se supondrá un modelo de agarre con rozamiento en sus puntos de contacto (véase la figura 3.9). Pero teniendo en cuenta la curvatura del objeto en intervalos de radio k (radio de la pinza), centrados en dichos puntos. Ha quedado reflejado en la exposición anterior, la necesidad de que las fuerzas ejercidas por los dedos en el agarre, posean direcciones contenidas siempre en los respectivos conos de rozamiento definidos en cada punto de contacto, con vistas a cumplir la condición de no-deslizamiento (necesaria para la clausura de fuerza), que constituye un requisito básico inicial para asegurar la estabilidad del agarre. En el apartado 3.3.3 se explica, mediante la definición de tres umbrales, el criterio seguido para configurar un agarre estable a partir de información geométrica basada en parámetros visuales extraídos de un sistema de visión. De hecho, uno de los problemas fundamentales a resolver en el capítulo V, será incorporar la condición de no-deslizamiento de nuestro modelo geométrico a partir, exclusivamente, de información visual.

3.3.2. Acerca de la Estabilidad.

Entramos aquí de lleno en el campo de la mecánica conocido como "estática", cuyo objetivo fundamental será determinar la resultante general y el momento resultante de todas las fuerzas que actúan sobre un cuerpo, a fin de establecer las condiciones de equilibrio. Recordemos que un cuerpo en reposo estará en equilibrio cuando la resultante general y el momento resultante de todas las fuerzas aplicadas sean nulos. El estudio de las condiciones de equilibrio se simplifica si se considera:

- (1°) Dos fuerzas iguales y opuestas, actuando sobre la misma línea de acción, producen equilibrio.
- (2°) En un cuerpo en equilibrio cada fuerza es igual y opuesta a la resultante de todas las demás.

En el contexto del agarre, la línea de acción vendrá fijada por la denominada línea de agarre, definida por la línea que une los puntos de contacto de los dedos de la pinza en su posición sobre la superficie del objeto.

La estabilidad, genéricamente hablando, es la tendencia de un sistema para regresar a un estado de equilibrio cuando, por causas externas, está desplazado del mismo. Se habla entonces de equilibrio estable.

En general, siguiendo el trabajo de Wolter [Wolter et al., 85], se pueden considerar tres tipos de movimientos posibles entre pinza y objeto (véase la figura 3.10), debido a las fuerzas de gravedad e inerciales: deslizamiento de traslación (a), deslizamiento de rotación (b), y giro (c).

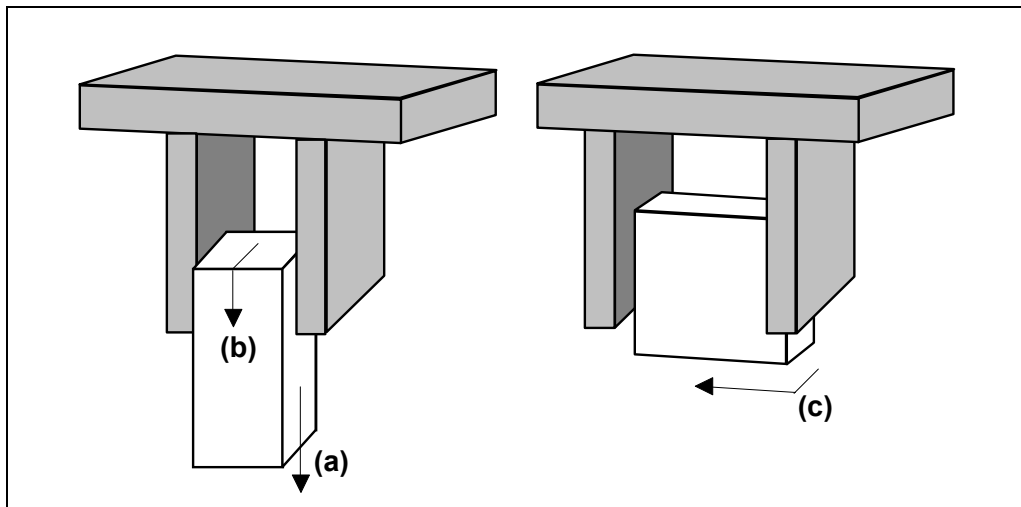


Figura.3.10. Adaptado de [Wolter et al., 85]. Tipos de movimientos posibles entre pinza y objeto, en general (3D). (a) *Sliding* (deslizamiento de traslación); (b) *Rotation* (deslizamiento de rotación); y (c) *Twisting* (giro).

La cinemática de los tres efectos anteriores sirve de base para establecer unos criterios de estabilidad generales, pero a costa de un conocimiento previo absoluto, tanto de la pinza como de los objetos.

Algunos autores [Montana, 91] distinguen entre dos tipos de estabilidad:

Estabilidad Espacial del Agarre (*spatial grasp stability*).

Se refiere a la tendencia del objeto a volver a su posición de equilibrio original en el espacio, relativa a la pinza, al ser desplazado éste por una fuerza externa.

Estabilidad de Contacto del Agarre (*contact grasp stability*).

Es la tendencia de los puntos de contacto a regresar a sus localizaciones originales en respuesta a una perturbación.

En la línea de investigación que intersecta con el primer enfoque (estabilidad espacial), se han realizado importantes contribuciones a su caracterización, como el concepto de "clausura de fuerza" [Nguyen, 88] anteriormente introducido, o la denominada "matriz de agarre" (*the grip transform*) [Mason & Salisbury, 85], la cuál constituye una representación en el agarre equivalente a la representación de la matriz Jacobiana en el contexto de la cinemática del manipulador. Otros trabajos en la línea del segundo enfoque [Cutkoski & Wright, 86], analizan la

estabilidad de contacto del agarre para dedos humanos, mostrando como la naturaleza viscoelástica presente en dichos dedos acrecienta dicha estabilidad.

Para dejar clara la necesidad del segundo enfoque (estabilidad de contacto) conviene plantear algunos aspectos que influyen en la estabilidad y sin embargo no son diferenciados al construir la "matriz de agarre" asociada. Aquellos aspectos a considerar que afectan en alguna medida a la estabilidad del agarre son [Montana, 91]:

- (1°) La curvatura (del objeto o de los dedos).
- (2°) La distancia entre los puntos de contacto.
- (3°) La viscoelasticidad de los dedos o del objeto.
- (4°) La existencia o no de realimentación de fuerza.

Para explicar el aspecto (1°) se pueden considerar dos elipsoides (véase la figura 3.11), uno dado por la ecuación $x^2 / a^2 + y^2 / b^2 + z^2 / b^2 = 1$, y el otro dado por la ecuación $x^2 / a^2 + y^2 / c^2 + z^2 / c^2 = 1$, donde $b < a < c$. Si suponemos un agarre con dedos plano-paralelos en los puntos $(a, 0, 0)$ y $(-a, 0, 0)$, desde el punto de vista de la estabilidad espacial, los dos agarres son equivalentes puesto que dan lugar a matrices de agarre formalmente idénticas (véase la parte superior de la figura 3.11). Una situación análoga ocurre al considerar la curvatura de los dedos de la pinza (véase la parte inferior de la figura 3.11). Queda ilustrado con ello la importancia de la curvatura, asociada tanto al objeto como a los dedos, en relación a la estabilidad del agarre.

En cuanto al aspecto (2°), considerando dos elipsoides (véase la figura 3.12), uno dado por la ecuación $x^2 / a^2 + y^2 / c^2 + z^2 / c^2 = 1$, y el otro dado por la ecuación $x^2 / b^2 + y^2 / c^2 + z^2 / c^2 = 1$, donde $a < c < b$. Si suponemos un agarre con dedos plano-paralelos en los puntos $(a, 0, 0)$ y $(-a, 0, 0)$, en el primer caso, y $(b, 0, 0)$ y $(-b, 0, 0)$, en el segundo caso. Se tiene que las curvaturas de ambos objetos en todos los puntos de contacto son idénticas. Siendo las formas de los dedos en ambos casos también las mismas. Sin embargo el agarre de la derecha (figura 3.12) parece más estable que el de la izquierda a pesar de que los objetos y los dedos poseen la misma geometría local, debido a que los puntos de contacto en el primer caso (izquierda) están más cerca el uno del otro que en el segundo.

Respecto a la posible consideración o no de viscoelasticidad (3°), es evidente que dos agarres idénticos en todos los aspectos salvo en las propiedades materiales de los dedos o del objeto, presenta mayor estabilidad aquél que posea dicha propiedad [Cutkoski & Wright, 86]. Su consideración se relaciona con el modelo definido anteriormente de "contacto con rozamiento (*soft-finger*)".

Por último, supondremos la no existencia de realimentación de fuerza en el agarre considerado (4°). Debido a que no se dispone de la información de un posible sensor de fuerza, no contemplado en nuestro caso, no será posible realizar movimientos de acomodación de los dedos de la pinza sobre la superficie del objeto.

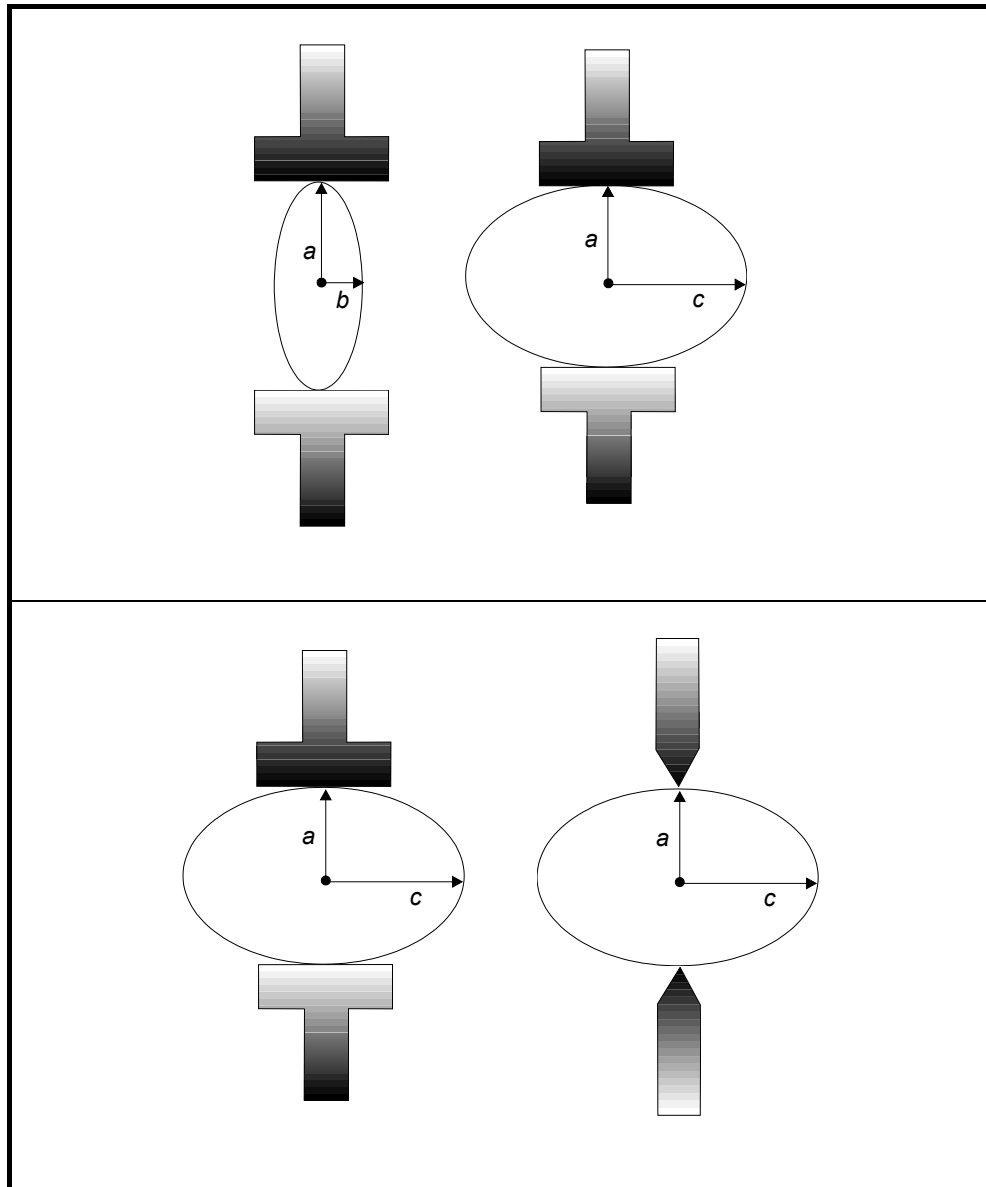


Figura.3.11. Adaptado de [Montana, 91]. Efecto de la curvatura sobre la estabilidad considerando: (Arriba) los objetos, si $b < a \Rightarrow$ inestable (izquierda), si $c > a \Rightarrow$ estable (derecha); (Abajo) los dedos, planos \Rightarrow estable (izquierda), puntiformes \Rightarrow inestable (derecha).

Teniendo en cuenta todos los aspectos anteriores (1°-4°) es posible obtener [Montana, 91] una medida de estabilidad del agarre que incorpore todos ellos. Sin embargo, al igual que otros modelos considerados [Wolter et al., 85], [Mason & Salisbury, 85] se requiere un conocimiento previo absoluto de la geometría de los objetos y de la pinza, para poder construir modelos cinemáticos representados por matrices, análogos a los Jacobianos en el contexto de la cinemática de los manipuladores.

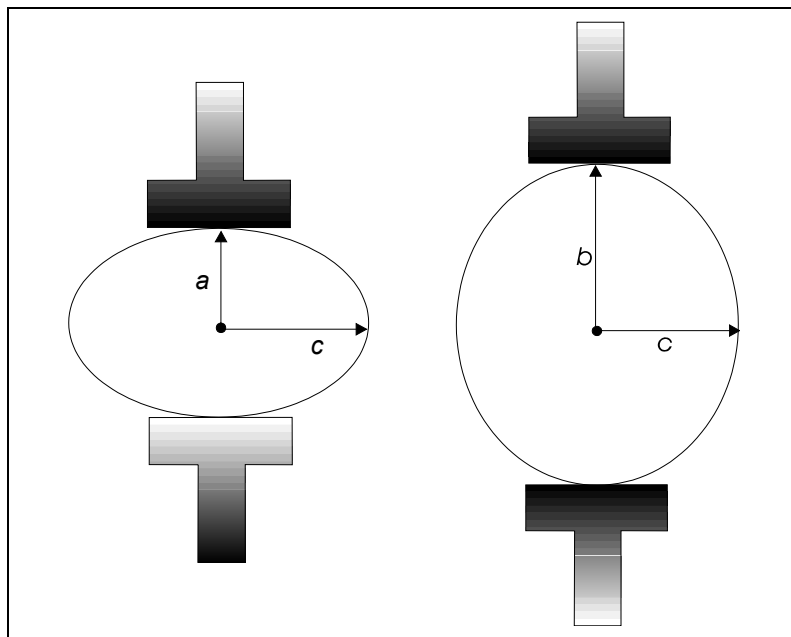


Figura.3.12. Adaptado de [Montana, 91]. Efecto de la distancia entre los puntos de contacto sobre la estabilidad: si $c > a \Rightarrow$ estabilidad (izquierda); si $c < b \Rightarrow$ inestabilidad.

Con respecto a la incorporación de criterios que garanticen estabilidad en planificadores como HANDEY, digamos que son excesivamente simples, aunque deja abierta la posibilidad de introducir tratamientos más completos de la misma (no deja claro cómo). De hecho, considera objetos "ligeros", en el sentido de que el agarre no se ve afectado siquiera por la distancia al centro de masas de los mismos. Su criterio consiste únicamente en requerir que un círculo inscrito en los dedos de la pinza intersecte con las regiones que garantizan el agarre del objeto.

Finalmente, debido a que el trabajo aquí representado es una extensión (en determinados aspectos) del otro trabajo "guía", LGUVI, relegamos el comentario de su enfoque sobre la incorporación de criterios de estabilidad al apartado siguiente 3.3.3.

3.3.3. Configuración del Agarre.

En el contexto del agarre clásico (pinzas de dos dedos plano-paralelos) que nos ocupa, tres posibles requerimientos geométricos son [Peshkin & Sanderson, 86]:

- (1°) El objeto debe tocar los dedos en la superficie interna de ambos, ó en su superficie externa, pero no una de cada. Esto se denomina "*the matter distribution requirement*" [Laugier, 81].
- (2°) Las superficies que se espera toquen los dedos deben ser planas y paralelas, aproximadamente.
- (3°) Dichos pares de superficies paralelas deben ser alcanzables desde el mismo lado.

En nuestro caso, relativo al requerimiento (1°), únicamente se considera la posibilidad del agarre por presión, por tanto solo intervendrán en el contacto las superficies internas de ambos dedos. La condición (2°) dará lugar a la definición de un umbral de curvatura y un umbral angular (véase más adelante). Por supuesto, el requerimiento (3°) siempre se garantizará en nuestro caso, debido a la vía de aproximación elegida (*grip*, en la figura 3.1).

Siguiendo otros trabajos cercanos al nuestro [Kamon et al., 96], y teniendo en cuenta las ideas expuestas anteriormente, referentes al contacto entre dos cuerpos (condición de no-deslizamiento) y a las condiciones de estabilidad (clausura de fuerza, curvatura, etc.), plantearemos nuestra configuración del agarre basada en el conocimiento de tres umbrales básicos:

Umbral de curvatura (α).

Tendrá en cuenta las condiciones mínimas de suavidad de los contornos del objeto para garantizar la adaptación de los dedos de la pinza al mismo, con el máximo contacto.

Umbral angular (β).

Asegurará la condición de no-deslizamiento, teniendo en cuenta puntos de contacto con rozamiento.

Umbral de distancia (γ).

Evaluará la distancia máxima permitida de alejamiento al centroide de la línea de agarre (P_1P_2).

Los detalles de implementación de estos umbrales se posponen al capítulo V, pero la idea fundamental, con vistas a preservar la estabilidad, es que los posibles candidatos a puntos de agarre habrán de cumplir, básicamente, tres propiedades:

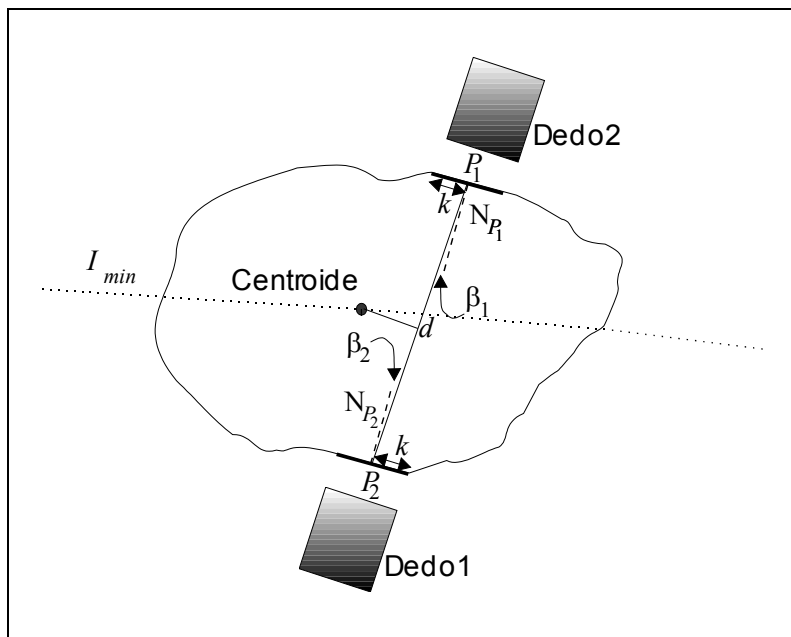


Figura.3.13. Interpretación geométrica de las condiciones de estabilidad asumidas: Curvatura, evaluación en regiones del contorno de radio k (radio de la pinza) centradas en los puntos (P_1 y P_2); determinación de ángulos β_1 y β_2 , de la línea de agarre con las normales (N_{P_1} , N_{P_2}); evaluación de la distancia, d , al centroide.

- 1^a La adaptación de los dedos de la pinza al objeto. Implica que habrán de encontrarse intervalos de radio k (radio de los dedos) centrados en los puntos de contacto (P_1 ó P_2) en condiciones de curvatura muy baja.
- 2^a La línea de agarre, el segmento de línea que une los puntos de contacto (P_1P_2), será aproximadamente (dentro del cono de rozamiento) normal al contorno en ambos puntos.
- 3^a La línea de agarre ha de permanecer próxima al centroide.

Como se puede observar, existe una relación directa entre cada una de estas tres propiedades y los umbrales definidos previamente. Un detalle importante es que el conocimiento para la evaluación de los umbrales anteriores y la etapa de razonamiento geométrico posterior, se hace exclusivamente a partir de dos parámetros visuales capturados en tiempo de ejecución: el centroide y la dirección del eje principal o de mínima inercia (I_{min}) y de una descripción del contorno del objeto.

Es conveniente observar la interpretación geométrica de estas propiedades a la vista de la figura 3.13, que muestra los posibles puntos de agarre, P_1 y P_2 , en el modelo aquí asumido. Obsérvese que d es la distancia de la línea de agarre al centroide y , β_1 y β_2 son los ángulos que forma dicha línea con las normales en cada uno de los puntos en cuestión (N_{P_1} y N_{P_2}).

La idea que subyace en las tres condiciones anteriores estaría en correspondencia con el razonamiento de sentido común [Davis, 90], empleado por la gente cuando de forma refleja procede al agarre de un objeto sobre una superficie plana, posicionando los dedos en lados opuestos del objeto y cercanos al centro de gravedad del mismo, e.g. un lápiz sobre una mesa o similar; existiendo evidencias de tipo psicofísico que lo demuestran [Opitz et al., 93].

3.4. Conclusiones Preliminares.

Todo lo desarrollado en este capítulo puede resumirse en las siguientes consideraciones:

- ◆ Un punto crucial es establecer una relación entre los criterios que configuran el agarre estable, definido en términos propios de la estática, y los parámetros obtenidos mediante un sistema de visión en tiempo de ejecución. La necesidad de incorporar visión al modelo de agarre planteado en este trabajo, desemboca en la caracterización del mismo a partir de la definición de tres umbrales. La heurística aquí planteada representa una extensión del trabajo de Kamon [Kamon et al., 96], añadiendo un umbral de curvatura a los dos inicialmente propuestos.
- ◆ A diferencia de otros autores [Chen & Burdick, 92] que, mediante técnicas de optimización buscan todos los agarres posibles cumpliendo determinadas condiciones (e.g. puntos de agarre antípodos), nuestro interés se centra en encontrar un agarre único, garantizando su estabilidad.
- ◆ Con relación a la morfología del universo de trabajo, mientras HANDEY presupone la existencia de modelos geométricos de los objetos, y LGUVI, la no existencia de huecos así como objetos con simetría especular en la dirección de I_{min} , nosotros no presuponemos nada. Esta complejidad añadida se resolverá mediante el análisis global de las formas a partir de propiedades de simetría y curvatura, fundamentalmente (véase el capítulo V).

Capítulo IV

Información Visual: Adquisición y Análisis

En este capítulo se trata el problema previo a la determinación del agarre propiamente dicho, es decir, todo el tratamiento de la imagen desde su captura hasta la extracción de características.

4.1. Introducción

Como ya se comentó en el punto 2.1, al hilo de la percepción visual, hay que resaltar la importancia de la experiencia cognoscitiva previa en la interpretación de una imagen [Ferraté et al, 86]. Una persona no ve niveles de gris caprichosamente repartidos en una superficie bidimensional, ni tampoco ve contornos, ni saltos bruscos de intensidad, ni ninguna otra característica definida numéricamente sobre una matriz. Lo que una persona realmente ve son objetos tridimensionales con los que ha sido familiarizado en el transcurso de su vida. He aquí la diferencia entre sensación y percepción. La sensación es la información captada por nuestros receptores; la percepción incluye además la interpretación dada a esa información gracias a la experiencia previa obtenida en situaciones semejantes. Además, el contexto ayuda a eliminar posibles ambigüedades. Otro factor a tener en cuenta es el principio gestáltico de que "el todo es más que la suma de las partes". Un ejemplo evidente de ello será observado en el capítulo V, demostrando la existencia de propiedades como la simetría, que sólo pueden ser captadas a nivel global y desaparecen en cuanto se

analiza localmente la imagen. Por último, destacar que los objetivos que se persigan tendrán una influencia decisiva en el diseño del mecanismo de percepción implementado. En nuestro caso, como se trata de razonar a partir de la forma de los contornos de los objetos, prescindiremos de toda aquella información que no sea relevante al respecto.

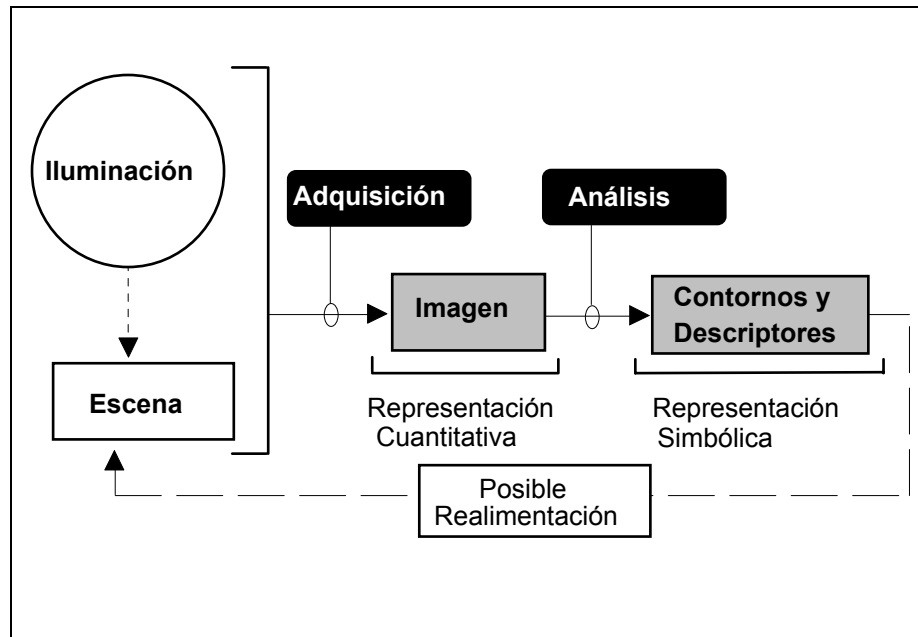


Figura 4.1. Adaptada de [Horn, 86]. Diagrama de bloques simplificado relativo al módulo de visión desarrollado.

De acuerdo con otros investigadores [Horn, 86], podemos afirmar que estamos todavía muy lejos de poder conseguir un sistema de visión "universal", que funcione bien con independencia del contexto y de los objetivos a alcanzar. Por el contrario, la mayor parte de los sistemas de visión tienen un alcance bastante limitado, resolviendo únicamente aquella aplicación concreta para la que fueron diseñados bajo unas determinadas condiciones iniciales. No obstante, se están empezando a crear las bases necesarias para construir módulos que solucionen áreas acotadas del proceso de visión con la idea de lograr una aproximación cada vez mayor a un sistema de visión "universal" (o de propósito general). Estos módulos exigirán adaptaciones mínimas a problemas específicos cualesquiera. Esta es la idea que subyace en la creación de un módulo previo al de "determinación de puntos de agarre basado en visión" (véase el capítulo siguiente). El módulo de visión requerido ha de capturar la escena y entregar a la salida una representación simbólica formada por cada uno de los contornos de los objetos que en ella aparezcan así como una

serie de descriptores asociados (véase la figura 4.1). Con la filosofía de fondo explicada anteriormente, hemos tratado de relajar las condiciones iniciales fundamentalmente en dos aspectos:

- 1° Condiciones de iluminación, admitiendo luz ambiente difusa, y
- 2° Morfología de los objetos que constituyen el universo de trabajo, permitiendo todo tipo de formas planas, con o sin huecos, etc.

Este capítulo muestra cada una de las etapas seguidas en la evolución de la información visual para el problema aquí planteado, tal y como se observa de forma simplificada en la figura 4.1. En ella se representan dos módulos básicos: *adquisición* y *análisis*. En *adquisición* englobaremos la problemática de la iluminación, la captura por el sistema de visión utilizado (cámara CCD más tarjeta digitalizadora), y el preprocesado (filtros, etc.). Mientras que en *análisis* abarcaremos las etapas de segmentación, extracción de contornos, cálculo de momentos y extracción de características basadas en los momentos.

4.2. Adquisición de la Imagen

Este problema es de importancia crucial para poder garantizar que el procesamiento de la imagen, a realizar posteriormente, será satisfactorio. Podemos considerar tres aspectos fundamentales [Davies, 90]:

- (1) El esquema de iluminación empleado.
- (2) La tecnología utilizada en la captura y digitalización de las imágenes.
- (3) La teoría básica subyacente al proceso de muestreo (e.g. el teorema de muestreo [Aström & Wittenmark, 88]).

El punto (2) está relacionado, básicamente, con los dispositivos físicos necesarios para capturar la imagen utilizando una cámara (CCD, en nuestro caso); digitalizarla a continuación mediante la tarjeta diseñada al efecto; y ubicarla, finalmente, en la memoria del ordenador, como una matriz de píxeles. Como este es un tema cerrado y de interés lateral en relación a nuestros objetivos, no será tratado aquí.

Relativo al punto (3), sólo recordaremos que debido a la naturaleza digital de las imágenes aparece el problema añadido de la fidelidad con que se aproximan éstas a los objetos de que provienen. Dicha fidelidad está relacionada con el factor de cuantización elegido en el proceso de muestreo (paso de la señal

analógica a digital). Existen fuentes excelentes al respecto [Rosenfeld & Kak, 82].

El punto (1), por su importancia en el trabajo aquí desarrollado, será tratado en el apartado siguiente.

Finalmente, debido tanto al ruido de cuantización [Coster & Chermant, 89] inherente al proceso de muestreo (3), como a la contaminación provocada por el tratamiento electrónico de la señal, estamos obligados a filtrar en alguna medida las imágenes mediante técnicas de procesamiento (véase el apartado 4.2.2).

4.2.1. El Esquema de Iluminación Empleado

Al describir el universo de trabajo en el capítulo III se omitió una propiedad de los objetos, la relativa a las características ópticas de la superficie de los mismos. Es decir, no se habló de la posibilidad de que una determinada superficie fuera especular, opaca, etc. De hecho las características ópticas de la superficie de los objetos están en conexión directa con el esquema de iluminación empleado, por ello se ha relegado su tratamiento a este apartado.

Un objetivo planteado en el presente trabajo ha sido estudiar las ventajas e inconvenientes de trabajar en un entorno de luz difusa, donde usualmente la fuente de iluminación es del tipo "tubo fluorescente". Encontrándose las mayores dificultades para las superficies cromadas (especulares) típicas, por otra parte, de la mayor parte de nuestro universo de trabajo (herramientas, etc.). La complejidad de este esquema de iluminación puede seguirse a partir de los siguientes aspectos clave [Davies, 90]:

- (1) Diferentes partes de los objetos aparecen iluminados de forma diferente, a causa de variaciones en el ángulo de incidencia, y las subsiguientes diferencias de brillo son capturadas por la cámara.
- (2) Los valores de brillo también varían a causa de diferencias en las reflectividades absolutas de las superficies del objeto.
- (3) Los valores de brillo varían con las especularidades de las superficies en lugares donde el incidente, el emergente y los ángulos de fase son compatibles con la reflexión especular.
- (4) La presencia de sombras afecta al valor del brillo en diferentes regiones de la imagen.
- (5) Aparecen efectos complejos debidos a posibles desviaciones de luz reflejada desde algunos objetos que iluminan otros. Factores que pueden conducir a variaciones muy complicadas de brillo sobre la imagen.

Ignorando los efectos (5) y (3), el esquema presentado todavía sigue siendo de complejidad considerable, debido a los factores (1), (2) y (4).

Se observa que, suponiendo una luz suficientemente difusa, el efecto (2) prevalece sobre (1) de manera que, en una primera aproximación, los valores de brillo observados representarán variaciones en la reflectividad de la superficie. De hecho, la iluminación difusa también suaviza las reflexiones especulares (efecto (3)), de manera que para la mayoría de los objetos puede ignorarse.

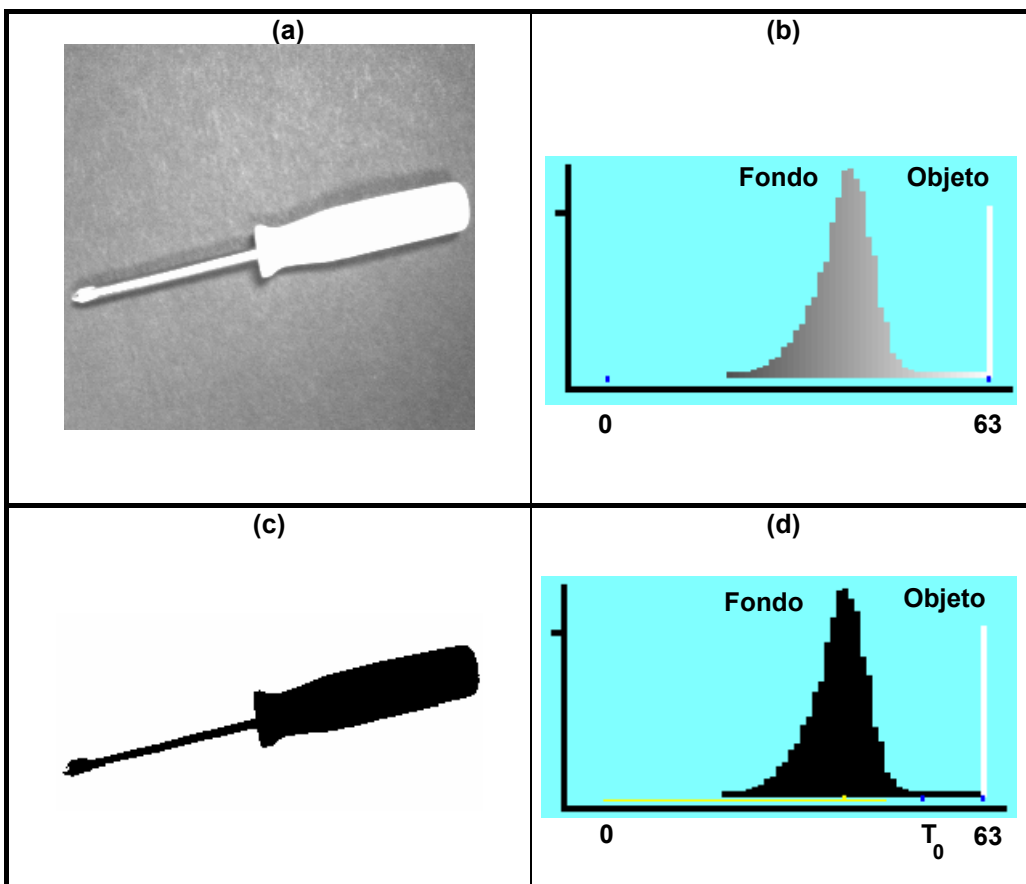


Figura 4.2. Caso simple de eliminación de sombras: objeto claro sobre fondo oscuro (destornillador). Imagen original (a). Histograma asociado (b). Resultado de umbralizar con frecuencia T_0 (c). Representación de la frecuencia de corte T_0 (d).

Es interesante puntualizar que se ha construido un algoritmo que, a partir de una imagen dada, distingue el modo asociado al objeto, del asignado al fondo.

Se basa en analizar los niveles de gris de las líneas que constituyen el borde de la imagen; se halla su valor medio y se compara con los niveles de gris asociados a los picos de máxima frecuencia que aparecen en el histograma. Esta estrategia ha demostrado un comportamiento enteramente satisfactorio en todos los casos experimentados. Como ejemplo pueden observarse las figuras 4.2 (objeto claro sobre fondo oscuro) y 4.3 (objeto oscuro sobre fondo claro).

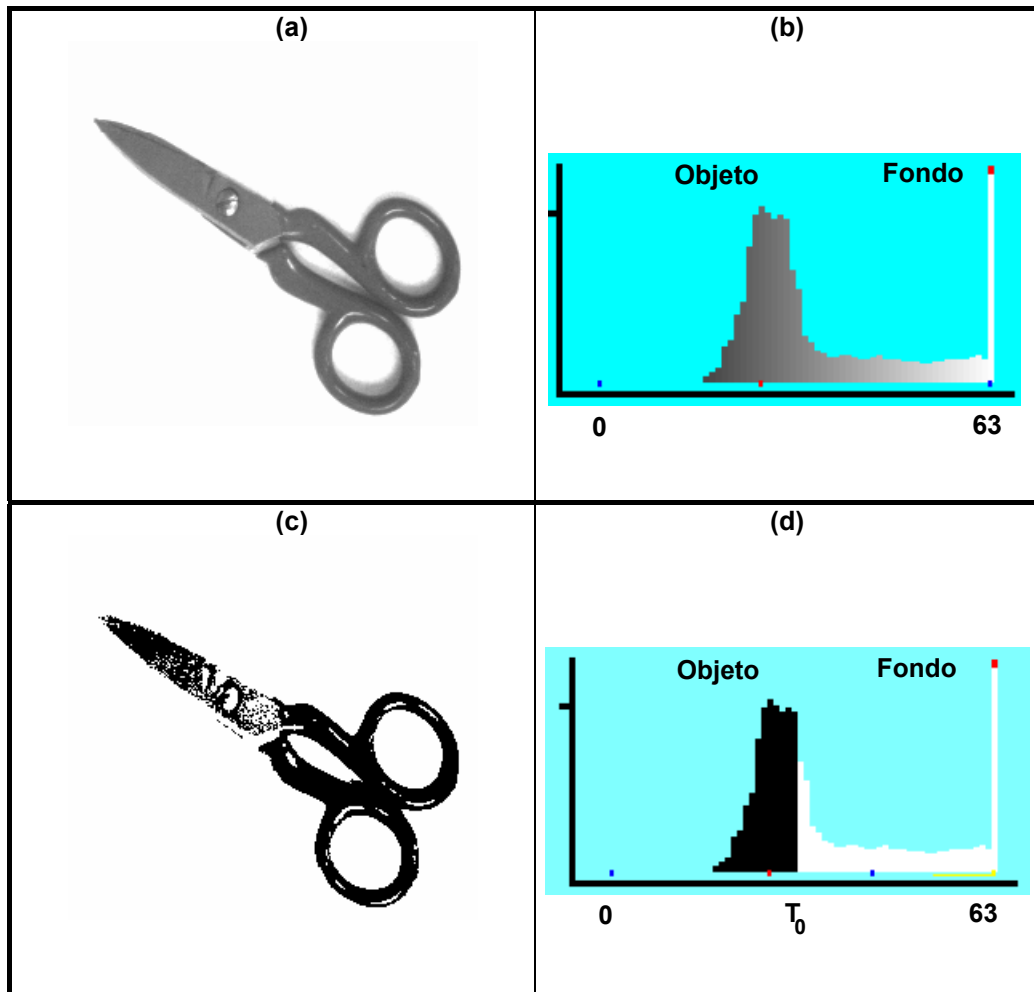


Figura 4.3. Caso complejo de eliminación de sombras: objeto oscuro sobre fondo claro (tijeras). Imagen original (a). Histograma asociado (b). Resultado de umbralizar con frecuencia T_0 (c). Representación de la frecuencia de corte T_0 (d).

Una de las problemáticas que sigue quedando por resolver es la de las sombras. Aquí se ha constatado la diferencia existente entre objetos claros sobre fondo oscuro y viceversa. En el primer caso la eliminación de sombras representa menos problema que en el segundo, tal y como afirman otros autores

[Davies, 90]. En la figura 4.2 se observa una imagen relativa al primer caso (destornillador), mostrando su histograma asociado. Es fácil observar que una técnica de segmentación directa, basada en umbralización global eliminará las sombras del contorno, como se constata que ocurre (c). La elección de un umbral óptimo, T_0 , en estos casos, consiste en encontrar una frecuencia mínima entre los dos modos existentes [Weska, 78]. El otro caso se representa en la figura 4.3. A primera vista se sigue tratando de un histograma bimodal, como el anterior, pero la diferencia crucial es que el histograma, en este caso, presenta una dispersión notable, de manera que no aparecen dos modos completamente diferenciados como en el caso anterior, sino que existen una serie apreciable de valores en las frecuencias que provocan la conexión entre ambos haciendo muy difícil la búsqueda de una frecuencia de corte. Esto se debe a que en este caso aparecen píxeles pertenecientes a regiones del objeto, con el mismo nivel de gris que los pertenecientes a la sombra (fondo). Esto implica que un algoritmo de segmentación, como el anterior, basado en umbralización global, no será enteramente satisfactorio. En la parte (c) de la figura 4.3 se observa el resultado de una umbralización, donde cuidadosamente (a mano) se ha elegido T_0 como frecuencia de corte. Se comprueba que a medida que desaparecen los píxeles de las sombras, desaparecen también píxeles de una región del objeto. Además, realizar esta elección de umbral de forma automática resulta imposible. Existen trabajos [Rosenfeld & Kak, 82], en los que se intenta incorporar el conocimiento de la distribución de los objetos en relación a las regiones espaciales de la imagen, lo cuál tampoco garantiza el correcto funcionamiento de la segmentación en casos dispares, y a cambio, es muy costoso a nivel computacional; nos referimos al denominado "umbral dinámico" [Fu et al., 88]. Como se observa, los métodos empleados en la segmentación de imágenes están totalmente mediatizados por los esquemas de iluminación utilizados. En el apartado 4.3 será tratado, a fondo, la estrategia de segmentación utilizada.

4.2.2. Preprocesado

Como se comentó anteriormente, debido al ruido proveniente del tratamiento electrónico de la señal y al inherente a la naturaleza misma del muestreo (analógico-digital), es prácticamente imposible obtener imágenes limpias. El conjunto de técnicas empleadas en tratamiento digital de imágenes para la eliminación de ruido se agrupa dentro de la etapa denominada de preprocesado de la imagen [Fu et al., 84]. Existen básicamente dos enfoques: uno en el dominio de las frecuencias, a partir de la Transformada de Fourier, y otro en el dominio espacial a partir de los niveles de gris de píxeles adyacentes. Debido a restricciones de tiempo real en el campo de la robótica, el primer enfoque, computacionalmente intensivo, está descartado.

Las funciones de preprocesamiento en el dominio espacial se pueden expresar, en general, como $g(x,y) = h[f(x,y)]$, donde $f(x,y)$ es la imagen de entrada, $g(x,y)$ es la imagen obtenida (preprocesada) y h es un operador en f , definido sobre el entorno de vecindad de (x,y) .

Una de las técnicas de dominio espacial usadas con mayor frecuencia se basa en el uso de las denominadas máscaras de convolución (plantillas, ventanas o filtros). Se trata, genéricamente hablando, de una matriz bidimensional cuyos coeficientes se eligen de forma que podamos detectar una propiedad dada para una imagen (véase la figura 4.4). Se utilizan, en general, para las llamadas "operaciones" en entorno de vecindad (reducción de ruido, apreciaciones de textura, esqueletizado, etc.). En nuestro caso, sólo estamos interesados en las propiedades de reducción de ruido.

A partir de la máscara visualizada en la figura 4.4 (izquierda), se obtiene el siguiente resultado:

$$h[f(x, y)] = \frac{1}{\sum_{i=1}^9 w_i} \left[\begin{array}{l} w_1 f(x-1, y-1) + w_2 f(x, y-1) + w_3 f(x+1, y-1) + \\ + w_4 f(x-1, y) + w_5 f(x, y) + w_6 f(x+1, y) + \\ + w_7 f(x-1, y+1) + w_8 f(x, y+1) + w_9 f(x+1, y+1) \end{array} \right]$$

donde w_1, w_2, \dots, w_9 , son los coeficientes de la máscara y se supone 8-conectividad. En nuestro caso se ha empleado una máscara como la visualizada en la figura 4.4 (derecha). Esta máscara detecta puntos aislados, de forma que aquellos que superen un cierto umbral, en su respuesta $h[f(x,y)]$, son eliminados.

w_1 (x-1,y-1)	w_2 (x,y-1)	w_3 (x+1,y-1)
w_4 (x-1,y)	w_5 (x,y)	w_6 (x+1,y)
w_7 (x-1,y+1)	w_8 (x,y+1)	w_9 (x+1,y+1)

-1	-1	-1
-1	8	-1
-1	-1	-1

Figura 4.4 Máscara 33 general (izquierda). Máscara para detección de puntos aislados (derecha).

El tamaño de la máscara es importante por cuestiones de eficiencia. Se observa que en nuestro caso se ha elegido un tamaño 3×3 , que ofrece un comportamiento adecuado a un coste razonable.

Puntualicemos que existen otros muchos filtros en la literatura, adecuados para mejorar el contraste, favoreciendo el proceso de segmentación subsiguiente, pero a costa, la mayor parte de las veces, de no ser suficientemente respetuosos con los contornos originales de las imágenes, así como de un elevado coste computacional.

En resumen, la estrategia utilizada ha consistido en pasar el filtro anterior, segmentar y extraer contornos. Eliminando en este último paso aquéllos que no superen un determinado tamaño.

4.3. Segmentación de la Imagen

El primer objetivo una vez que la imagen ha sido digitalizada y preprocesada es obtener una descripción de partes específicas presentes en ella. Para generar esta descripción es necesario segmentar la imagen en las distintas partes que van a ser analizadas; es decir, aislar los objetos de interés del fondo. Típicamente se obtiene una nueva imagen cuya función característica f_c tendrá el valor 1 para píxeles pertenecientes a los objetos y 0 en cualquier otro caso. Una buena revisión de métodos de segmentación puede encontrarse en [Zenko, 83] o más recientemente en [Pal & Pal, 93].

Esta etapa puede ser bastante compleja si no existe un dominio absoluto de las condiciones de iluminación del área de trabajo. Y bastante simple, en caso de disponer de dicho control, e.g. la típica iluminación posterior que, obvia de una sola vez multitud de problemas como reflejos, cambios de texturas, luces parásitas, etc., entregando, en tiempo real, las imágenes cuasi segmentadas. Nosotros prescindiremos de este tipo de iluminación controlada, intentando buscar una cierta relajación de las condiciones iniciales pero partiendo siempre de imágenes cuyo histograma sea aproximadamente bimodal. Supondremos que la escena cumple dos tipos básicos de restricciones:

- 1^a Para facilitar la segmentación de la imagen a partir de la selección automática del umbral y consiguiente umbralización, se supondrá que la textura de los objetos es cuasi uniforme.
- 2^a Para facilitar la extracción de bordes supondremos la no existencia de objetos contiguos o solapados.

Las restricciones anteriores se asocian a las denominadas escenas "no complejas" [Maravall, 93], en el sentido de que la primera restricción anterior evitará el costoso, computacionalmente hablando, tratamiento de texturas, y la segunda garantizará la posible extracción de contornos del objeto, sin conocimiento a priori de su morfología.

En general, un proceso de segmentación puede clasificarse bajo dos grupos fundamentales: segmentación contextual y no contextual. El primero emplea relaciones de características entre vecinos. De esta forma la información local puede reforzar la decisión de clasificación; sin embargo, suele ser información difícil de cuantificar teóricamente y farragosa de implementar. El segundo evita estos problemas; suele basarse en criterios estadísticos para la clasificación de

cada píxel individual utilizando normalmente la información de su intensidad. Por otra parte, cuando las propiedades de la imagen son insuficientes para discriminar los objetos presentes en ella, se suele hacer uso de información *a priori* para identificarlos y aislarlos. En nuestro caso nos movemos en un entorno de laboratorio (no hay situaciones imprevisibles) pero con un entorno de iluminación no controlada (luz ambiente). No obstante, si se siguen determinadas restricciones, se puede utilizar un enfoque de tipo no contextual, segmentando las imágenes por umbralización de su nivel de gris; es decir, clasificando los píxeles en dos clases: "oscuro" y "claro" en un intento de distinguir entre los objetos y el fondo (o viceversa). Esto es posible por cuanto las imágenes que tenemos son bien contrastadas y ambas clases son diferenciables. Utilizaremos una aproximación basada en la determinación adaptativa del umbral, que se ajustará automáticamente a las condiciones de iluminación de cada escena.

4.3.1. Determinación del Umbral Óptimo

Un sencillo caso de segmentación no contextual hemos dicho que es la umbralización en una distribución bimodal. Vamos a ver la formalización del caso en el que el umbral óptimo es seleccionado por ser aquél que minimiza el error cometido durante el proceso de clasificación de los píxeles en función de la probabilidad de cada punto de pertenecer a una clase u otra. Un histograma bimodal corresponde a una imagen bien contrastada, y su distribución de frecuencias puede ser aproximada por una combinación lineal de dos distribuciones gaussianas:

$$p(z) = P(0) p(z_0) + P(1) p(z_1)$$

siendo $P(0)$ y $P(1)$ las probabilidades de ocurrencia de ambas clases (porcentaje de píxeles asignables a cada modo), y $p(z_0)$ y $p(z_1)$ dos funciones de distribución de probabilidad, en este caso las gaussianas caracterizadas por sus medias μ_0 y μ_1 y desviaciones estándar σ_0 y σ_1 que aproximan ambos modos. Esta aproximación es válida si σ_0 y σ_1 son pequeñas y similares, y las medias están separadas de los límites del rango de grises (0 y 63). Cada una de estas gaussianas será unimodal y ambas estarán separadas por un valle profundo. Entonces se elige el umbral que cause el mínimo error al clasificar los píxeles entre ambos modos. Es fácil de demostrar [Fu et al., 84] en estas condiciones, que el umbral que minimiza el error de asignar los píxeles a una u otra clase es:

$$\theta_0 = \frac{\mu_0 + \mu_1}{2} + \frac{\sigma^2}{\mu_0 - \mu_1} \ln \frac{P_1}{P_0} \quad (4.1)$$

Es importante que el umbral sea determinado de manera automática, pues esto hace al sistema insensible a cambios en las condiciones de iluminación (siempre que se respete el principio de contraste entre objetos y fondo).

El caso extremo de un histograma bimodal es, de hecho, una imagen binaria en la que todos los puntos del objeto estarían con un nivel z_0 y todos los del fondo con otro, z_1 . Pero la realidad es que las irregularidades en la iluminación (iluminación arbitraria, no controlada) y la presencia de sombras, reflejos y texturas en el objeto y en el fondo, hacen que los modos tengan una dispersión. El cálculo de las medias y desviaciones de los dos modos presentes nos permitirá conocer en torno a qué valores de z se encuentran más a menudo los píxeles y con qué dispersión. Si así lo hacemos habremos caracterizado el histograma en función de distribuciones de probabilidad de que cada nivel pertenezca a un modo u otro.

Para evaluar el umbral óptimo de forma adaptativa, estableceremos una condición de tipo morfológico que nos permitirá asegurar que la morfología de los objetos segmentados es fiel a la imagen originalmente digitalizada. Si trazamos las fronteras de los objetos por aquellos píxeles ocupados al menos en un 50% por el objeto y un 50% de fondo, deberíamos asignar el umbral al nivel de gris intermedio en nuestra escala ($z = 31$), pero eso sólo es posible si los modos de nuestras imágenes estuvieran centrados en $z_0 = 0$ y $z_1 = z_{max} = 63$. Así pues, nuestro objetivo será modificar la imagen original, unificando las condiciones de las imágenes a segmentar de forma que lleguen a este proceso con similares características de iluminación aparente. Esta imagen la conseguiremos mediante la aplicación de una transformación que modificará la imagen original obteniendo una imagen pseudo-binaria mediante un proceso de estiramiento de contraste adaptativo [Iñesta, 94].

4.3.2 Obtención de una Imagen Pseudo-Binaria

Entre los procesos de tratamiento digital de imagen ocupan un lugar destacado las transformaciones de histograma $T[H(z)]$, que mediante un adecuado tratamiento de los valores de las frecuencias de ocurrencia de cada nivel de intensidad permiten mejoras concretas en la imagen transformada [Pratt, 91]. Entre éstas se encuentran las operaciones puntuales, que son operaciones en las que a cada nivel de gris $z[z_{min}, z_{max}]$ se le hace corresponder un nivel $z'[z_{min}, z_{max}]$ de acuerdo con una aplicación $z'=F(z)$. Son fácilmente implementables mediante tablas de transformación LUT¹ *hardware* o *software*. Se denominan operaciones puntuales porque el valor transformado depende únicamente del original, sin tener en cuenta relaciones entre propiedades de otros niveles del histograma. Una de estas modificaciones es la de estiramiento del contraste (*contrast stretching*) [Pratt, 91], que es una función lineal a tramos en la que una zona con pendiente > 1 es aplicada a la región del histograma que quiere mejorarse en su contraste. Si la pendiente sólo es distinta de cero en dicha zona, entonces se dice que es una transformación de recorte (*clipping*) [Jain, 89], que eliminará el ruido y la estructura interna de las zonas distintas a aquéllas en las que se aplica el estiramiento de contraste. Estas últimas

¹ LUT \equiv Look Up Table

corresponderán a zonas de transición entre objetos y fondo, que es donde vamos a tener concentrada la información morfológica y de caracterización en nuestras imágenes.

Basándose en la discusión del apartado anterior se puede establecer los límites de cada zona en el histograma de la imagen original:

1 *Píxeles claramente asignables al objeto:*

Aquellos con niveles de gris comprendidos entre el nivel mínimo ($z_{min} = 0$) y la media del modo oscuro más un número determinado de veces (n) su desviación estándar ($\mu_0 + n \sigma_0$). A estos píxeles se les atribuirá en la imagen transformada el nivel de gris $z' = 0$.

2 *Píxeles claramente asignables al fondo:*

Aquellos con niveles de gris comprendidos entre la media del modo oscuro menos un número determinado de veces (n) su desviación estándar ($\mu_1 - n \sigma_1$) y el nivel máximo ($z_{max} = 63$). A estos píxeles se les atribuirá en la imagen transformada el nivel de gris máximo, $z' = z_{max} = 63$.

3 *Píxeles en las zonas de transición:*

Que tomarán valores en el resto de niveles, expandiendo el histograma linealmente para que abarquen toda la amplitud de niveles de gris (entre 0 y 63).

Esta transformación puede expresarse analíticamente de la siguiente manera:

$$z' = T[H(z)] = \begin{cases} 0 & \text{si } z \leq \mu_0 + n\sigma_0 \\ \frac{z_{max}}{\mu_1 - n\sigma_1 - (\mu_0 + n\sigma_0)} z + \frac{z_{max}(\mu_0 + n\sigma_0)}{\mu_0 + n\sigma_0 - (\mu_1 - n\sigma_1)} & \text{si } \mu_1 - n\sigma_1 > z > \mu_0 + n\sigma_0 \\ z_{max} & \text{si } z \geq \mu_1 - n\sigma_1 \end{cases} \quad (4.2)$$

que se representa gráficamente en la figura 4.5.

Cada tramo lineal de esta transformación corresponde a cada una de las clases en las que hemos clasificado los píxeles de la imagen. La linealidad del tramo central expande los niveles de gris intermedios preservando la contribución de cada uno, lo cual no se conseguiría con un tramo cuadrático, exponencial, etc.

La primera tarea a afrontar para dicho procesamiento de la imagen original es estudiar el histograma y caracterizar la distribución de frecuencias mediante la evaluación de los parámetros μ_0 , σ_0 , μ_1 y σ_1 . El algoritmo utilizado funciona adecuadamente si presuponemos que la entrada va a ser una imagen bimodal. Para ello se analiza secuencialmente el histograma, eliminando los

niveles con poca ocurrencia respecto a las frecuencias máximas detectadas y aquéllos que después de esta criba hayan quedado aislados. De esta manera quedará un histograma con un grupo de niveles a cero entre los dos modos detectados (objetos y fondo), así caracterizados por el algoritmo descrito en 4.2.1. Para lograr automatismo en este proceso, el análisis del histograma requiere un preproceso de sus valores que en bruto, tal y como se extraen de las imágenes originales, suelen ser bastante abruptos. Por ello conviene suavizarlos para facilitar la aplicación de metodologías de análisis. Con este fin se ha empleado una técnica de filtrado del histograma mediante un filtro gaussiano con muy buenos resultados:

$$G(z, \omega) = \frac{1}{\sqrt{2\pi\omega}} e^{-\frac{z^2}{2\omega^2}}$$

donde la anchura de la gaussiana, ω , se ha fijado empíricamente a 2.5 que representa una anchura suficiente para tener un filtro ancho y muy suavizador, dejando los histogramas suficientemente suaves como para facilitar la búsqueda de las crestas y los valles que caracterizan los posibles modos encontrados.

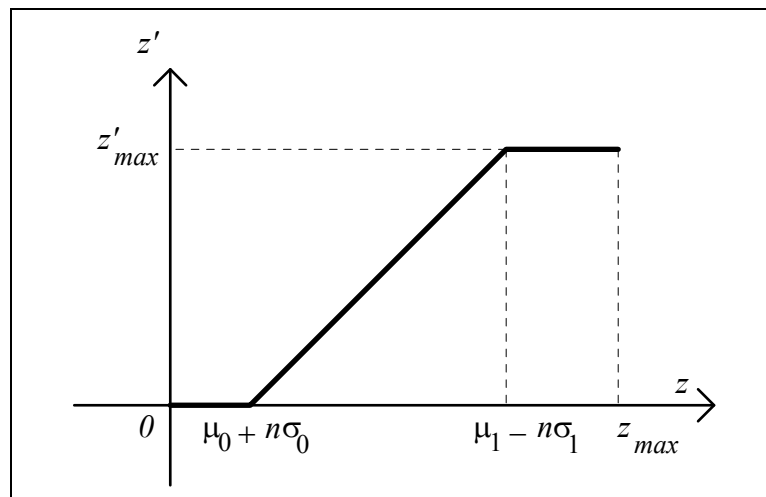


Figura 4.5. Representación gráfica de la transformación del histograma.

A cada uno de estos modos así aislados se aplican las fórmulas de cálculo de media y desviación estándar a partir de las frecuencias de una distribución:

Media	Desviación estándar
$\mu_m = \frac{\sum_i z_i f(z_i)}{\sum_i f(z_i)}$	$\sigma_m^2 = \frac{\sum_i z_i^2 f(z_i)}{\sum_i f(z_i)} - \mu_m^2$

donde las $f(z_i)$ representan las frecuencias de cada nivel z_i .

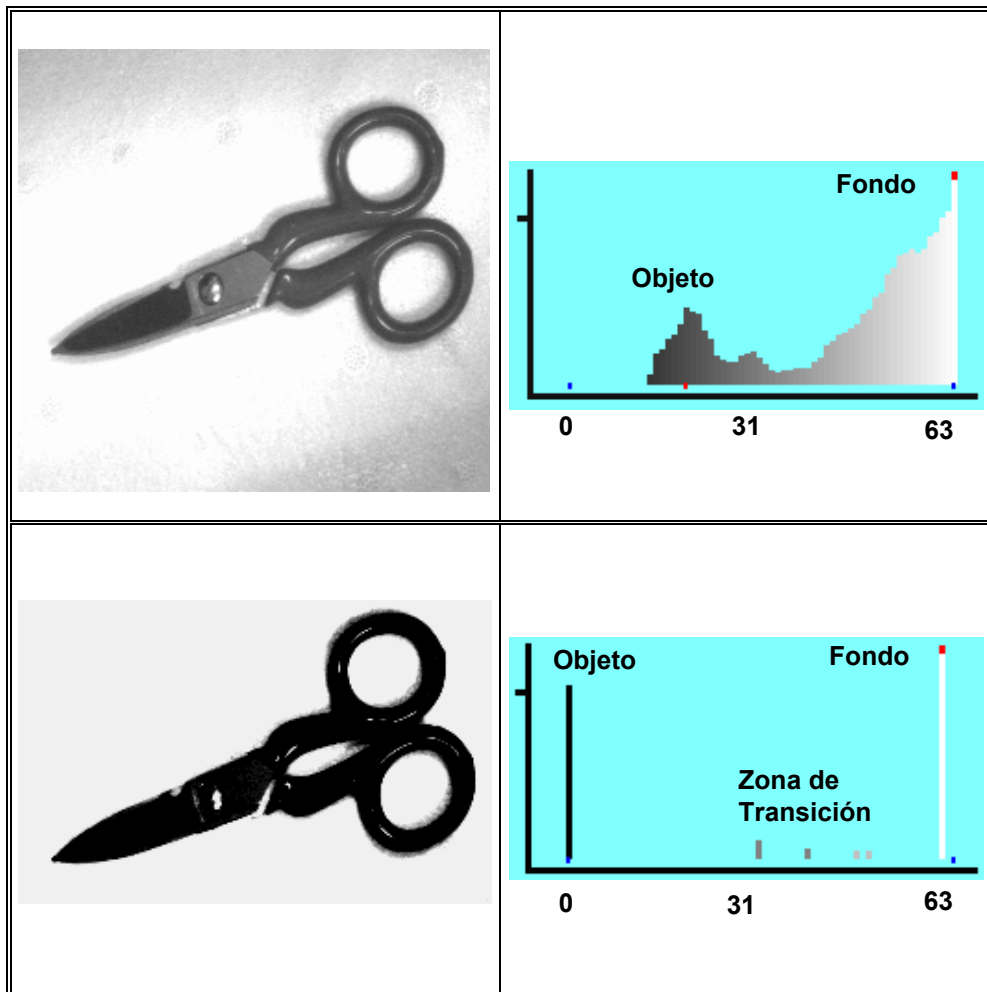


Figura 4.5. Arriba: Imagen original (izquierda), y su histograma asociado (derecha). Abajo: Imagen pseudo-binaria (izquierda), resultado de aplicar la transformación de *contrast stretching* propuesta; obsérvese el efecto de la transformación sobre el histograma (derecha), dejando constancia de las tres regiones establecidas.

Una vez caracterizados los modos, se aplica la expresión general de la función de transferencia (4.2) y se obtiene la transformación pretendida. Tan sólo falta por determinar el valor de n necesario para controlar la anchura de los tramos a uniformizar en ambos modos. Para ello, el procedimiento seguido ha sido el de ir variando $n = 1, 2, 3, \dots$, y calculando para cada valor $\mu_0 + n \sigma_0$, y $\mu_1 - n \sigma_1$, hasta que se verifique que \exists un $n = N / \mu_0 + n \sigma_0 \geq \mu_1 - n \sigma_1$. En ese momento se fija $n = N-1$.

La implementación de esta función se realiza mediante una LUT implementada en Pascal como un array que se aplica a todos los puntos de la imagen, transforma sus niveles, que hacen de índice, en niveles transformados z' del array:

$z'(x,y)$ **array** [$z(x,y)$] **of** Byte

cuyos valores son previamente calculados mediante la expresión analítica de la transformación (4.2).

En la figura 4.5 se visualiza un ejemplo del comportamiento obtenido para una imagen de trabajo típica: *tijeras*. Se observan cada una de las tres regiones impuestas por la transformación (4.2). A partir de esta imagen pseudo-binaria será bastante sencillo aplicar un algoritmo de extracción de contornos, como se propone en el apartado siguiente. Nótese que aparecerán contornos falsos, inducidos por falta de homogeneidad en las texturas de los objetos; no obstante, es relativamente simple su eliminación en base al tamaño de los mismos.

Finalmente, podemos decir que el proceso de segmentación y posterior extracción de contornos acaban la labor comenzada con el preprocesado: separar los objetos de interés, para nuestros objetivos, del resto de la escena.

4.4. Extracción de Contornos.

El algoritmo utilizado es una variante del algoritmo de codificación de contornos mediante códigos de cadena descrito en [Gonzalez & Wintz, 87], el cual reduce la imagen pseudobinaria, obtenida en el paso anterior (segmentación), a una *lista* de contornos, unívocamente determinados mediante la especificación de:

- 1 Su nivel de gris.
- 2 La posición (fila y columna) de un píxel en su contorno, llamado punto inicial (PI).
- 3 Una secuencia de direcciones que permiten trazar el borde externo del contorno.

Un código de cadena puede representar un contorno mediante una secuencia de segmentos rectos de longitud y dirección específica, como por ejemplo el código 4 u 8-direccional de la figura 4.6. Supondremos en lo que sigue una teselación rectangular del espacio, por otra parte la más utilizada con diferencia, en el procesamiento digital de imágenes. Es importante puntualizar que los códigos de direcciones representados en la figura 4.6 definen una relación de vecindad, es decir, el código 4-direccional da lugar a la 4-conectividad, y el 8-direccional a la 8-conectividad. Para evitar ambigüedades hay que fijar a priori uno de los dos. En nuestro caso, elegimos 8-conectividad por ser la que mejor se adapta a las representaciones de los contornos digitalizados y ser más compacta que la 4-conectividad. Tengamos en cuenta que la consideración de un mayor número de direcciones no mejora la precisión de la cuantización de una curva [Saghri & Freeman, 81].

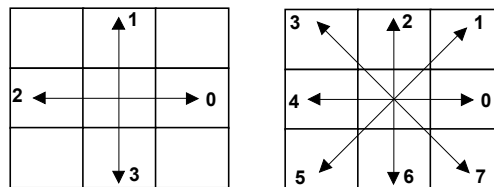


Figura.4.6. Representación gráfica del código 4-direccional (izquierda) y 8-direccional (derecha), de Freeman.

El algoritmo utilizado se basa en dos subalgoritmos:

Subalgoritmo de búsqueda de puntos iniciales (**PI**).

Definido un nivel de gris para el fondo de la imagen (Blanco ó 63, en nuestro caso), el algoritmo hace un barrido por líneas de toda la imagen buscando puntos con un nivel de gris distinto del indicado. Una vez encontrado tal punto (si existe), llama al algoritmo de trazado de contorno. Cuando este termina, se continúa con la búsqueda de puntos iniciales evitándose que cualquier punto del objeto ya contorneado sea considerado como punto inicial mediante un proceso de marcado de los puntos del contorno y un inteligente manejo de una pila de niveles de gris (véase la figura 4.7). Nótese que el píxel alcanzado será un punto inicial si su atributo es izquierda (I, en la figura 4.7) y su nivel de gris no está en la misma mitad del rango de grises que el de la última entrada en la pila.

Subalgoritmo de trazado de un contorno (**T**).

Desde el punto inicial de un contorno (que establecemos como el de mínima coordenada "y" del objeto) se utiliza la regla clásica de "atravesar

un laberinto girando siempre en el mismo sentido" (supondremos sentido horario, en nuestro caso).

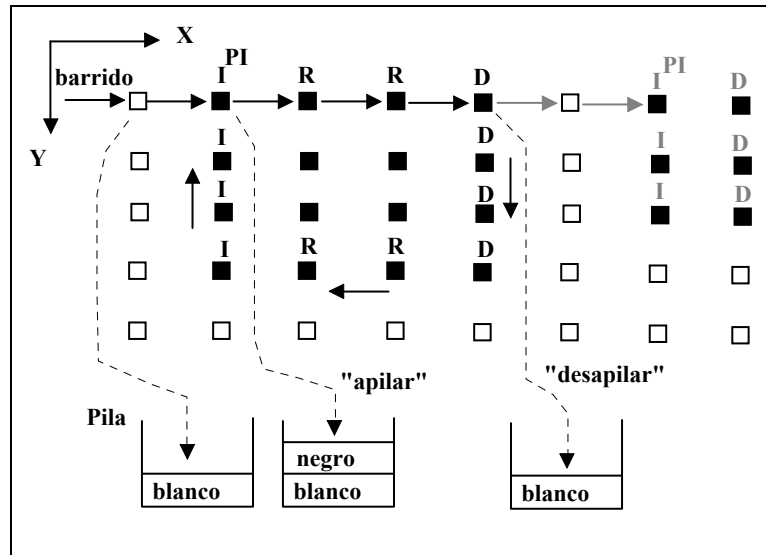


Figura.4.7. Representación gráfica del esquema de trabajo del algoritmo de "búsqueda de puntos iniciales". Se observa la detección de dos puntos iniciales, uno para cada contorno, cumpliendo la caracterización dada por el subalgoritmo correspondiente ().

Cuestión-1: ¿Cómo podemos almacenar los contornos? (véase la figura 4.8).

Una primera idea puede ser almacenar cada una de las coordenadas de los puntos (*pixels*) que forman parte de cada contorno. Nosotros seguiremos la idea propuesta en [Gonzalez & Wintz, 87], y guardaremos únicamente las coordenadas de los puntos iniciales, PI, de cada uno, con la secuencia correspondiente de direcciones asociadas a los puntos que forman parte de dicho contorno. La implementación de esta estructura puede realizarse utilizando un vector cuyos elementos serán registros con tres campos: *Pixc*, *Piyc* (de tipo entero), que representarán las coordenadas del PI de cada contorno; y un campo de tipo puntero, *listado*, que apuntará a una lista cuyos nodos tendrán un único campo de información, *dato* (de tipo *byte*) que contendrá el código del segmento de la cadena, además del puntero al siguiente elemento (*next*). Los sucesivos contornos que se vayan detectando se almacenarán siguiendo la estructura de la figura 4.8. Posteriormente, con el objetivo de facilitar el razonamiento geométrico necesario en las fases posteriores del algoritmo, esta estructura de datos inicial del contorno, la lista

dinámica, se trasvasa a un vector dinámico², donde se guardan las coordenadas de los puntos del contorno, aplicando simultáneamente un *m-muestreo* (ver apartado 4.5).

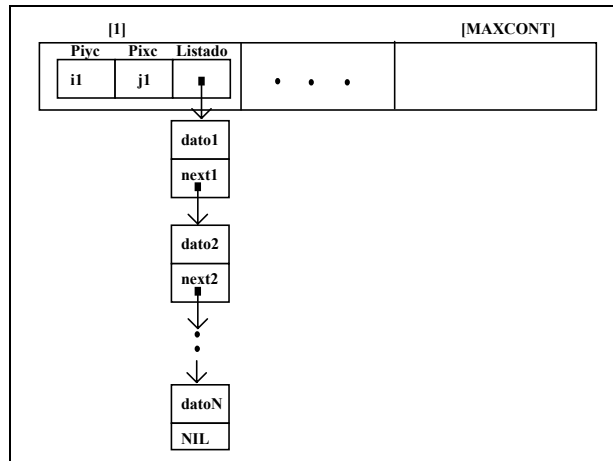


Figura 4.8. Cada contorno se almacena, guardando las coordenadas de su punto inicial, PI, y creando una Lista a partir de éste que guarde las direcciones de los sucesivos puntos del contorno que encontramos a partir de dicho punto inicial.

Cuestión-2: ¿Cómo procederemos a una búsqueda exhaustiva de todos los contornos?:

Representaremos (en el programa), la **A** por un cero, la **I** por un uno, la **D** por un dos y la **R** por un tres. Empezaremos creando una matriz de atributos de las mismas dimensiones que la imagen. Se inicializa a cero (**A**). Y la recorreremos punto a punto. Al comienzo estaremos en un área de unos, es decir fuera de todo objeto. Cuando encontramos un **cero** éste será un punto inicial (**PI**). Entonces llamamos al procedimiento **T** que busca el contorno a partir de dicho punto. Cuando acaba lo recorreremos de principio a fin aplicando la siguiente regla:

Sí el punto tiene un atributo A (no se ha pasado anteriormente por él) aplicamos la **regla 1ª** (vista posteriormente). Sí por el contrario el punto ya tiene un atributo aplicaremos la **2ª regla** (vista posteriormente).

Cada vez que pasemos de una zona a otra (de nivel de gris 0 a 1 o viceversa) lo indicaremos con una variable Booleana (en general, se utilizará una pila, véase la figura 4.7). Los puntos iniciales (**PI**) coinciden con el paso de un nivel

² Véase el apéndice D.

al siguiente. El algoritmo termina cuando se han utilizado todos los puntos de la imagen.

A continuación se expone la estrategia seguida para implementar el algoritmo de seguimiento de contornos, teniendo en cuenta 8-conectividad:

Entrada: Array de niveles de gris (imagen pseudo-binarizada). Umbral: T_0

Salida: Lista de contornos (PI, Codificación de Freeman y atributo de objeto o hueco)

Pasos:

1. Partir del punto superior izquierda de la imagen y suponer a éste perteneciente al fondo ($z > T_0$).
2. Barrer la imagen de izquierda a derecha y de arriba a abajo hasta encontrar un punto en el que se cambie de zona de gris y con atributo **0**. Marcar este punto como **PI** de este contorno.
3. Trazar el contorno de la región detectada:

Inicializar valores del contorno y empezar por **PI**

repetir

repetir

 mirar el píxel correspondiente según la regla de la figura 4-9

Si z está al otro lado de T_0 en el rango de grises **entonces**

 avanzar a la siguiente dirección de salida posible según la figura 4-9

hasta encontrar un píxel con z en el mismo lado del rango de grises

O volver a la dirección de llegada sin haberlo conseguido.

si se encuentra un píxel con z en el mismo lado del rango de grises **entonces**

 Asignar la dirección de salida que se dirige hacia él desde el punto de entrada

si no

 Se trata de un contorno de un solo punto

hasta Llegar de nuevo a **PI**
4. Marcar el contorno en la matriz de atributos recorriendo el contorno en sentido horario aplicando las reglas pertinentes para la asignación de atributos a cada píxel del contorno.
5. Repetir 2, 3 y 4 hasta llegar a la esquina inferior derecha de la pantalla.

Finalmente, lo relativo a las Reglas de Asignación de Atributos explícitamente sería:

```

regla1 :
           r1 = ((3,3,3,0,2,2,2,2), (1,1,1,1,0,3,3,3),
                (1,1,1,1,3,0,3,3), (1,1,1,1,3,3,0,3),
                (1,1,1,1,3,3,3,0), (0,3,3,3,2,2,2,2),
                (3,0,3,3,2,2,2,2), (3,3,0,3,2,2,2,2));

regla2 :
           r2 = ((1,3,1), (3,2,2), (1,2,3));
    
```

Donde, r1, representa las 64 posibilidades asociadas con las ocho direcciones de entrada y salida (véase la figura 4.9), contempladas por el subalgoritmo de trazado de contornos, y r2, las nueve asociadas con los tres atributos posibles (I,D y R), cuando dicho algoritmo pasa por segunda vez por el mismo píxel del contorno.

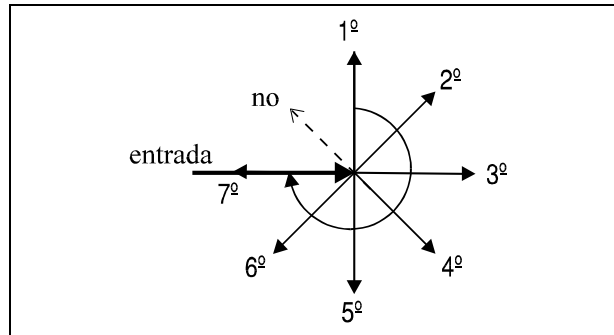


Figura 4.9. Adaptada de [Iñesta, 94]. Regla de salida en función de la entrada a cada píxel.

En resumen, una curva digital cerrada C queda representada por una secuencia de N puntos de coordenadas enteras:

$$C = \{ p_i = (x_i, y_i), \quad i = 1, \dots, N \}$$

donde p_{i+1} es un 8-vecino de p_i (módulo N), puede expresarse mediante un punto inicial $PI = (x_0, y_0)$ y un código de cadena de Freeman que se explicita como una sucesión de N vectores $\{c_i, i=1, \dots, N\}$, donde

$$c_i = \overline{p_{i-1}p_i}$$

y cada vector c_i es codificado mediante el valor de un entero $c_i \in [0,7]$ (siendo $c_i(\pi/4)$ el ángulo entre cada vector y el eje x).

4.5. Representación Eficiente de los Contornos.

La técnica del *m-muestreo* [Proffit & Rosen, 79] originalmente propuesta para paliar el efecto de la naturaleza discreta del plano imagen cuando se pretende medir la longitud de una curva digital (e.g. perímetro de un contorno), es aquí utilizada como método sencillo y rápido de reducción de datos. A continuación justificaremos esta metodología y la elección de un valor adecuado para m .

4.5.1. Justificación de la elección del 3-muestreo.

Después del proceso de segmentación, es a menudo importante encontrar una representación adecuada de los contornos del objeto con vistas a ganar eficiencia en el cálculo de los descriptores, pero preservando la morfología de los objetos para poder aplicar razonamientos geométricos sobre los mismos. Tengamos en cuenta que todos los algoritmos utilizados a partir de aquí tienen un coste $O(N)$ directamente proporcional al número de puntos del contorno.

Las técnicas para representar más eficientemente una curva mediante una aproximación poligonal se conocen habitualmente como de segmentación de curvas, y han sido propuestos cierto número de algoritmos para llevar a cabo esta tarea ([Pavlidis, 78], [Toussaint, 85], [Imai & Iri, 86], [West & Rosin, 91]), pero todos ellos son excesivamente costosos a nivel computacional para nuestros objetivos, ya que intentan una segmentación inteligente basada en el estudio de propiedades geométricas en cada punto del contorno, y aplican luego un procedimiento de selección para elegir un subconjunto adecuado de puntos con un alto contenido en información. Pero nuestro sistema no puede funcionar con este enfoque; ya que debido a restricciones temporales no puede detenerse a conseguir una aproximación morfológicamente valiosa y le bastará con alcanzar una reducción de datos que aproxime lo suficiente para que el polígono resultante no distorsione en exceso la forma original. Por todo ello se ha utilizado un método menos inteligente pero suficientemente efectivo para reducción de datos: el m -muestreo, descrito por Profit [Proffit & Rosen, 79].

Se determinará cuál es el valor más adecuado para m , con el fin de lograr un compromiso entre una reducción efectiva de puntos del contorno y un respeto a la morfología de los objetos analizados. Esto será de importancia vital en nuestro caso ya que necesitaremos eficiencia para el proceso de realimentación visual y precisión para seleccionar los puntos de agarre adecuados; de manera que si resultara dañado el contorno por la reducción de datos, los algoritmos podrían llevarnos a conclusiones erróneas. La técnica del m -muestreo se basa en la selección de un punto cada m puntos de la curva (véase la figura 4.10). Así, si $m = 1$, el efecto en la curva es dejarla intacta y la reducción de datos no ocurre. Por otra parte, un alto valor para m implica un importante ratio para la reducción de datos, pudiendo sufrir la morfología una seria alteración.

Como se comprueba en la figura 4.10, la línea digital resultado de muestrear los puntos del contorno original se aproxima bastante mejor a la línea digitalizada real para algunos valores pequeños de m que la obtenida uniendo todos los puntos originales. La longitud de la curva muestreada se evalúa como

$$L_m(m) = \sum_{j=0}^{N/m} d(m)_j$$

que depende del valor elegido para m , siendo $d(m)_j$ la distancia euclídea entre los puntos $p_{j,m}$ y $p_{(j+1),m}$. Mediremos también la longitud de una

aproximación poligonal, calculada a partir de un procedimiento de selección de puntos dominantes basado en el respeto a la morfología original del contorno [Iñesta, 95], tomada como valor de referencia y para fines comparativos. Este valor tiene validez sólo si el criterio de selección de puntos está relacionado con la acción de preservar la morfología de la forma, y el método usado está diseñado con este fin. Su longitud será la suma de las longitudes de los segmentos rectilíneos que construyen la aproximación poligonal:

$$L_{PA} = \sum_{j=0}^n l_j$$

siendo n el número de segmentos de la aproximación y l_j la distancia euclídea entre los extremos del segmento j -ésimo.

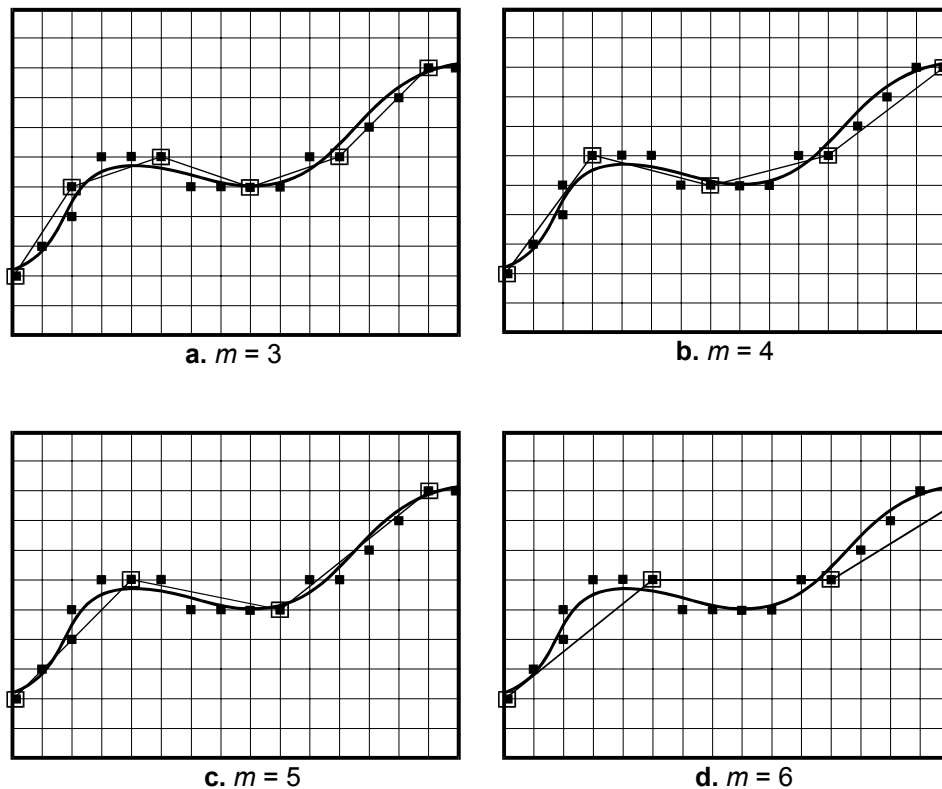


Figura 4.10. Adaptada de [Iñesta, 94]. m -muestreo de una línea digital con distintas m . Los cuadrados negros marcan los nodos de la rejilla seleccionados por proximidad a la curva original, mientras que los nodos encerrados por un recuadro son los seleccionados por el muestreo (1 de cada m). Se observan diferentes grados de adaptación a la curva original según el valor de m empleado.

Cuanto mayor es la constante de muestreo, m , menor es el perímetro resultante. Sin muestreo ($m=1$) estaríamos ante el estimador de Freeman L_F . En este caso los perímetros quedarán sobrestimados. Así pues, es razonable pensar que con alguna constante $m>1$ a determinar en cada circunstancia o aplicación, nos aproximaremos a un valor del perímetro razonablemente cercano al real. Este procedimiento se comporta especialmente bien para segmentos rectilíneos [Proffit & Rosen, 79]. Por otra parte, L_m está especialmente desaconsejado cuando los contornos a analizar contienen picos muy agudos o abruptas concavidades, pues el muestreo puede conducir a eliminar algunos de ellos y falsear por completo la forma de la curva original (sobre todo si m es grande) y, por tanto, su longitud. Así mismo, contornos muy ruidosos pueden provocar artificialmente dichos accidentes en el perfil, llevando a idénticos e indeseados efectos.

Nosotros vamos a cuantificar los efectos del m -muestreo para un simple descriptor como el perímetro. Trataremos con formas de perímetro conocido con vistas a comparar los resultados para diferentes valores de m . También estudiaremos la exactitud de las medidas tomadas.

Históricamente, el problema de la estimación del valor del perímetro original de una curva digital, ha sido tratado como un intento de inferir la información perdida en el proceso de digitalización de una entidad de naturaleza continua. Dicho de otra manera, nadie puede asegurar qué es lo que sucede con la curva original en el intervalo inaccesible que existe entre los nodos de la rejilla de cuantización. Este planteamiento asume la imposibilidad de valorar con fiabilidad arbitraria la longitud original del suceso físico (*línea digitalizada*) representado por la línea digital [Dorst & Smeulders, 87]. Por esta razón, al hablar de perímetros digitales, hemos hablado de *estimadores*.

Consideraremos dos puntos de vista para valorar la calidad del m -muestreo en cuanto a la reducción de datos se refiere:

1º) Reproducibilidad de las medidas, o cuál es el intervalo de error esperado para un valor del perímetro extraído a partir del m -muestreo (véase la figura 4.11); los ocho estimadores computados para hallar su dispersión son los siguientes:

L_m 2-8	7 m -muestreros con m desde 2 hasta 8;
L_{AP}	Longitud de la aproximación poligonal.

En la figura se muestran las dispersiones para estos ocho estimadores cuando se han aplicado a 16 digitalizaciones de dos objetos bajo distintas condiciones. Como se observa, las menores dispersiones (y por tanto la mejor reproducibilidad) se presenta para la aproximación poligonal inteligente; para los m -muestreros, la menor dispersión se da para valores intermedios de m .

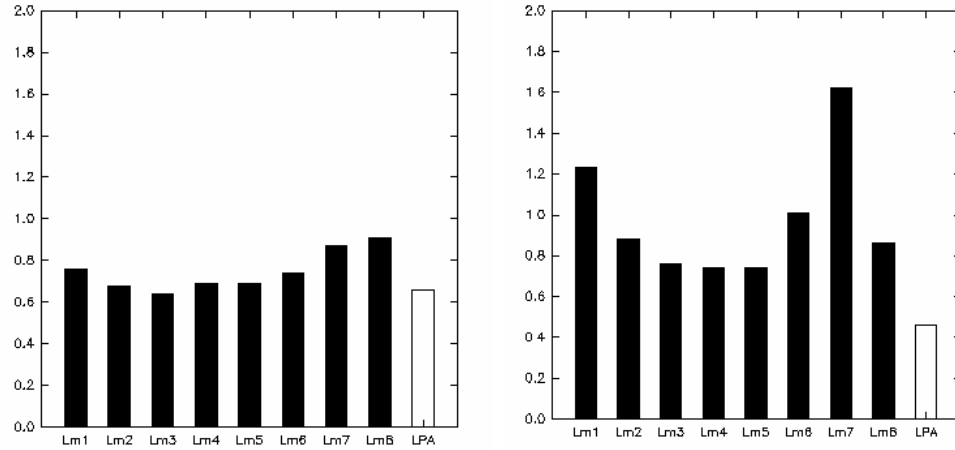


Figura 4.11. Adaptada de [Iñesta, 94]. Porcentajes encontrados para las dispersiones con respecto a los valores medios para estimaciones de perímetros con diferentes valores para m y la aproximación poligonal para m y la aproximación poligonal para dos objetos de prueba. Las dispersiones más pequeñas se encuentran para los valores centrales de m y para LPA . Estos resultados han sido reproducidos para otros contornos testeados.

2º) Proximidad del valor obtenido al valor real: otros experimentos [Iñesta, 94] muestran que con un valor para m igual a 3 o 4, la estimación del perímetro toma valores cercanos a los reales, medidos manualmente sobre objetos reales. Entre estos dos valores de m , 3 representa la mejor elección para preservar la forma original cuando se presentan ángulos agudos.

En resumen, después de todo lo visto anteriormente, hemos encontrado dos buenos estimadores para la longitud del perímetro de una curva digital. El mejor de todos parece ser LPA , ya que combina reproductibilidad y precisión, como se esperaba teniendo en cuenta la calidad de los puntos seleccionados para construir la aproximación poligonal; pero, como ya dijimos anteriormente, cuando el tiempo de procesado es una restricción, se necesita un procedimiento alternativo menos inteligente de reducción de datos, y en ese caso, $Lm3$ parece ofrecer los mejores resultados, con un buen comportamiento, rápido y sencillo de implementar. La ventaja inmediata será la reducción de puntos a un tercio del inicial, y el consiguiente ahorro en los tiempos de procesado de los algoritmos subsiguientes (véase apartado siguiente al respecto).

4.5.2. Impacto del 3-muestreo en la eficiencia

Una de las cuestiones cruciales a considerar será la del análisis del impacto del 3-muestreo sobre la eficiencia de los algoritmos utilizados [sanzp et al., 96]. Para analizar el impacto del 3-muestreo (reducción de datos) en la eficiencia de los algoritmos, se ha realizado un total de 4 digitalizaciones sobre cada una de las 10 imágenes³ que se nombran en la tabla adjunta (véase la tabla 4.1). Se muestra una comparativa entre los tiempos de CPU para los algoritmos que utilizan los contornos procesados con dicho muestreo ($m = 3$) y los que no lo hacen ($m = 1$). Se observa un mejor rendimiento de los algoritmos para el caso $m = 3$, debido a la dependencia lineal de los mismos con el número de puntos del contorno de los objetos. Los picos que aparecen en la representación gráfica de la figura 4.12 son debidos a la sobrecarga del procesamiento previo, en relación con la complejidad de las estructuras de contornos que aparecen para las imágenes tijeras1 y tijeras2, por poseer sendos agujeros cada una de ellas.

Imágenes	NºP ₁ (μ)	CPU ₁ (μ)	CPU ₁ (σ)	NºP ₃ (μ)	CPU ₃ (μ)	CPU ₃ (σ)
Tuerca	125	0.32	0.02	42	0.30	0.03
Tornillo	238	0.34	0.03	80	0.35	0.03
Allen	306	0.40	0.03	102	0.33	0.00
Tijeras1	438	0.52	0.03	146	0.47	0.03
Destornillador	467	0.45	0.02	156	0.41	0.03
Tijeras2	573	0.60	0.00	191	0.49	0.01
Pinzas	582	0.59	0.03	194	0.47	0.03
Calibre-A	695	0.60	0.00	232	0.46	0.03
Alicates	705	0.62	0.03	235	0.47	0.07
Tenazas	763	0.66	0.00	254	0.52	0.03

Tabla 4.1. Se observan tres columnas de datos para cada uno de los algoritmos utilizados. Siendo "NºP_n", el número de puntos del contorno, "CPU_n" el tiempo asociado en segundos, " μ ", la media y " σ ", la desviación típica para 1-muestreo ($n = 1$) y 3-muestreo ($n = 3$), respectivamente.

Recordemos que los tiempos de CPU han sido obtenidos a partir del reloj del sistema, con un ordenador 486 a 33 MHz con 8 Mb de RAM.

Como se deduce de los resultados obtenidos, parece ser que el rendimiento de los algoritmos mejora apreciablemente con el 3-muestreo y dicha tendencia parece incrementarse conforme aumenta el número de puntos de los contornos.

³ Véase el apéndice C, donde se muestran los contornos de cada una de las imágenes utilizadas.

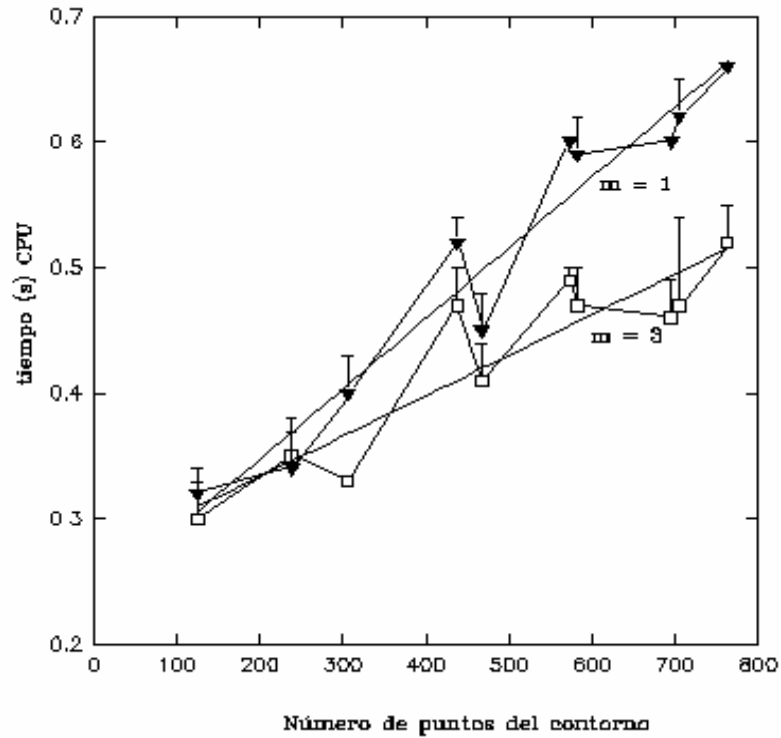


Figura 4.12. Relación entre el número de puntos del contorno y el tiempo de CPU, en segundos, para el 1-muestreo ($m = 1$) y el 3-muestreo ($m = 3$). Las líneas de regresión muestran cuáles son las tendencias cuando se incrementa dicho número de puntos en cada caso.

Finalmente, digamos que se plantea una ligera desventaja referente a la etapa de razonamiento geométrico posterior (capítulo V). Básicamente, el problema surge al calcular intersecciones del contorno en direcciones cualesquiera. Habrá que imponer un umbral de distancia para resolver ambigüedades entre puntos contiguos, etc., como se verá en dicho capítulo.

4.6. Computación Eficiente de los Momentos.

En general, dada una imagen representada por una función arbitraria representativa de los niveles de gris en cada imagen, $f(x,y)$, se define el momento de orden (p,q) , m_{pq} como

$$m_{pq} = \int_{-\infty-\infty}^{\infty \infty} x^p y^q f(x,y) dx dy \quad p, q \in \{0,1,2,\dots,\infty\}$$

Si dicha función, $f(x,y)$, es acotada (como siempre ocurre en el caso de imágenes digitales) existirá un conjunto único de momentos generales asociados

que la describen y viceversa [Duda & Hart, 73], [Maravall, 93]. Este resultado explica el porqué de la importancia de los descriptores basados en el cálculo de momentos en el área de reconocimiento de formas, con vistas a construir vectores de características, etc. En el presente trabajo, estos momentos geométricos serán utilizados para calcular el centroide, la orientación de la forma estudiada y las magnitudes de los semiejes principales de la elipse de mejor ajuste de los objetos de interés.

Teniendo en cuenta que las coordenadas del centroide de la función $f(x,y)$ se definen como:

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad ; \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

podremos definir los denominados momentos centrales:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy \quad p, q \in \{0, 1, 2, \dots, \infty\}$$

En nuestro caso, la función $f(x,y)$ representa la intensidad de cada uno de los puntos (x,y) de una imagen digital; por tanto, necesitamos la representación discreta de las definiciones anteriores [Fu et al., 88], [Maravall, 93]. Es decir,

$$m_{pq} = \sum_{x=0}^{N_x} \sum_{y=0}^{N_y} x^p y^q f(x, y) \quad p, q \in \{0, 1, 2, \dots, \infty\}$$

para los momentos geométricos, y

$$\mu_{pq} = \sum_{x=0}^{N_x} \sum_{y=0}^{N_y} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad p, q \in \{0, 1, 2, \dots, \infty\}$$

para los centrales.

En el caso de una imagen binaria, la función $f(x,y)$ es una función acotada que toma valores distintos de cero únicamente en el contorno de los objetos presentes en ella y en su interior. Para caracterizar una forma mediante un conjunto de momentos se puede optar por dos aproximaciones: (1) Considerar la función definida exclusivamente en el conjunto de puntos que definen su contorno; (2) Establecer el campo de existencia de dicha función en el conjunto de puntos del contorno del objeto más su interior.

Los momentos geométricos que se obtienen a partir del contorno, aunque lo caracterizan adecuadamente, sufren el defecto de ser muy sensibles al ruido y a pequeñas variaciones en la forma de un contorno. Por contra, los basados en el contorno más su interior presentan una mayor robustez [Maravall, 93]. Nosotros seguiremos este segundo enfoque.

Zakaria [Zakaria et al., 87] propuso un método llamado *regla delta* en el que los momentos geométricos totales se calculan sumando las contribuciones de

cada una de las líneas de las que está compuesto el objeto sometido a estudio. Para ello basta con conocer la posición de los puntos del contorno, lo que reduce la complejidad a un orden lineal con el número de puntos del mismo, ver figura 4.13.

Este método será el utilizado cuando calculemos los m_{pq} , pues es sencillo de implementar y fácilmente generalizable para tratar objetos con agujeros, concavidades arbitrarias e imágenes no binarias, aspectos estos no contemplados en el trabajo original [Zakaria et al., 87]. El método integral [Dai et al., 92] también es eficiente y sencillo, pero sus ventajas respecto a la *regla delta* residen fundamentalmente en su implementación para el cálculo a partir del código de cadena de un contorno y este aspecto no contempla la posibilidad de objetos con agujeros internos.

Se ha desarrollado un algoritmo basado en la *regla delta* que lo extiende para alcanzar estos objetivos de generalización necesarios para nuestras formas (con agujeros y concavidades arbitrarias) [Sanz et al., 94].

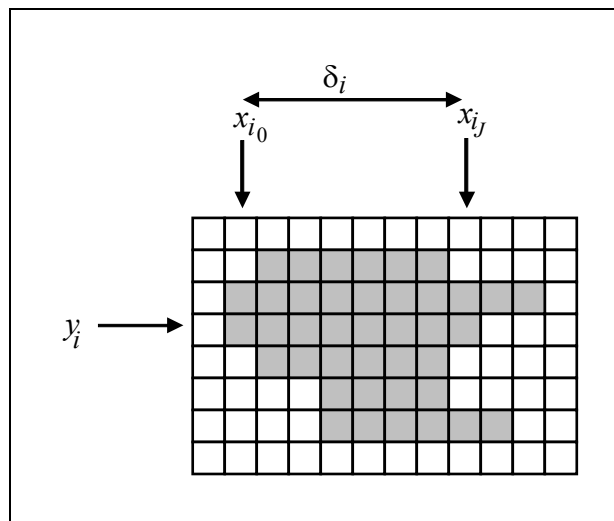


Figura 4.13. Interpretación geométrica de la regla delta original. Se observa la contribución de la fila y_i , correspondiente a $\delta_i = x_{i_j} - x_{i_0} + 1$.

A partir de la definición para la regla delta original:

$$m_{pq} = \sum_{i=0}^{N_y} m_{pq,i} \quad p, q \in \{0, 1, 2, \dots\}$$

se obtienen los siguientes momentos de hasta orden dos:

$$\begin{aligned}
 m_{00,i} &= \delta_i \\
 m_{01,i} &= \delta_i y_i = m_{00,i} y_i \\
 m_{02,i} &= \delta_i y_i^2 = m_{01,i} y_i \\
 m_{10,i} &= \delta_i x_{i_0} + (\delta_i^2 - \delta_i)/2 \\
 m_{11,i} &= m_{10,i} y_i \\
 m_{12,i} &= m_{11,i} y_i \\
 m_{20,i} &= \delta_i x_{i_0}^2 + 2x_{i_0}(\delta_i^2 - \delta_i)/2 + \delta_i^3/3 - \delta_i^2/2 + \delta_i/6 \\
 m_{21,i} &= m_{20,i} y_i
 \end{aligned}$$

y sólo resta sumar las contribuciones de las N_y filas de cada momento de orden (p,q) , para obtener los momentos totales.

Como interesa trabajar con objetos sin restricciones (cóncavos, con huecos etc.), utilizaremos la regla delta extendida, la cual considera la posibilidad de que las contribuciones por filas de los píxeles de la imagen esté particionada en J_i secciones para cada línea i ,

$$m_{pq,i} = \sum_{j=1}^{J_i} m_{pq,ij}$$

De esta manera, los m_{pq} calculados con este método se obtienen en un tiempo del orden de 30 veces menor que los calculados a partir de la definición aplicada sobre la imagen binaria $I(x,y)$ del objeto,

$$m_{pq} = \sum_{j=0}^{N_x} \sum_{i=0}^{N_y} x_j^p y_i^q I(x_j, y_i)$$

que exhibe complejidad $O(N_x N_y) = O(N^2)$; lo cual es muy conveniente para nuestros propósitos. Podemos ver su significado en el ejemplo de la figura 4.14.

A partir de estos m_{pq} se definen los correspondientes momentos centrales [Gonzalez & Wintz, 87]:

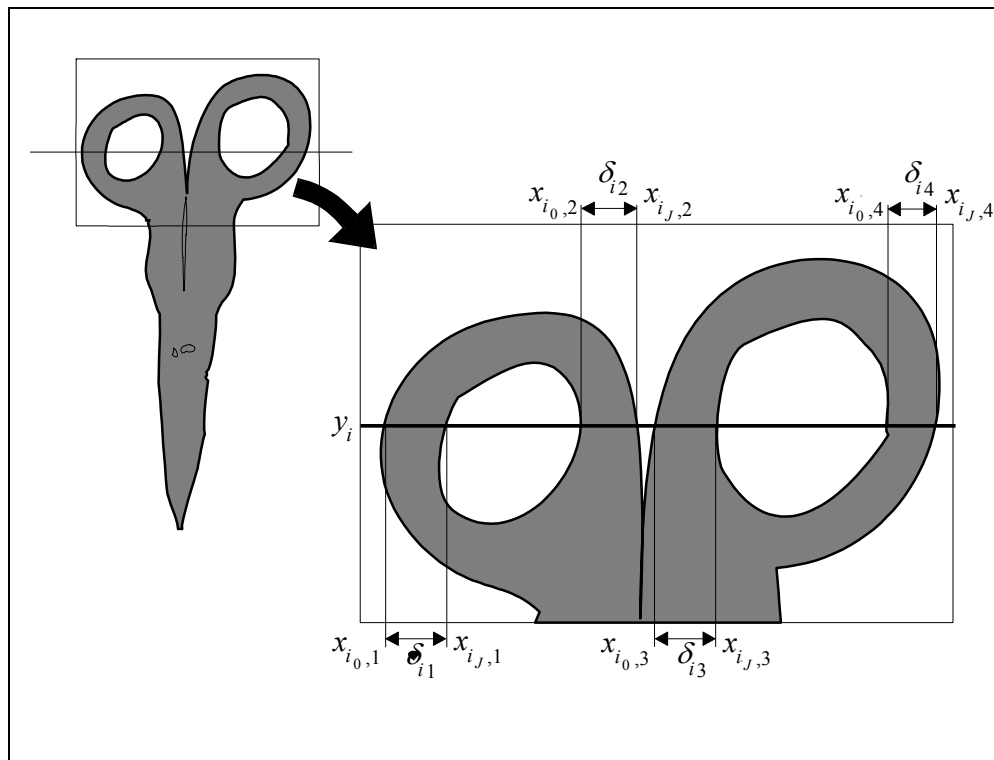


Figura 4.14. Interpretación geométrica de la regla delta extendida. Se observa la contribución de la fila y_i , correspondiente a $\delta_{i1} + \delta_{i2} + \delta_{i3} + \delta_{i4}$.

$$\mu_{00} = m_{00} ; \quad \mu_{10} = \mu_{01} = 0$$

$$\mu_{02} = m_{02} - \bar{y}m_{01}$$

$$\mu_{11} = m_{11} - \bar{y}m_{10}$$

$$\mu_{12} = m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2\bar{y}^2m_{10}$$

$$\mu_{20} = m_{20} - \bar{x}m_{10}$$

$$\mu_{21} = m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2\bar{x}^2m_{01}$$

Puntualicemos que los momentos centrales serán invariantes a traslaciones, al definirse relativos al centroide de la forma y no a coordenadas absolutas. Es interesante recordar que existen un conjunto de momentos, denominados "invariantes de Hu" [Hu, 62] que son invariantes a traslaciones, rotaciones y cambios de escala. Estas combinaciones no lineales de los momentos μ_{pq} son de importancia crucial como descriptores en el área de reconocimiento de formas. No obstante, no se requiere su uso para los objetivos aquí planteados.

4.7. Extracción de Características Basadas en los Momentos.

Los dos descriptores fundamentales sobre los que se soporta el razonamiento geométrico inherente al algoritmo que desarrollamos en el capítulo V serán:

1º El centro de masas o *Centroide* de la función de distribución de intensidades:

$$\bar{x} = \frac{m_{10}}{m_{00}} ; \bar{y} = \frac{m_{01}}{m_{00}}$$

2º La orientación del *Eje Principal*⁴, que coincide con la del eje de mínima inercia del objeto (I_{min}):

$$\theta = \frac{1}{2} \text{atan} \left[\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right]$$

dicho eje pasa por el centroide del objeto y, a partir del mismo, es trivial calcular el eje ortogonal a él pasando por el centroide: dirección del eje de máxima inercia (I_{max}). Para evitar posibles ambigüedades en el cálculo de la orientación anterior, se hará uso de la función "atan2" (véase Apéndice A) que es una función de dos argumentos que permite discriminar cualquier región angular sin posibilidad de error, teniendo en cuenta el seno y el coseno simultáneamente:

$$\theta = \frac{1}{2} \text{atan} 2(2\mu_{11}, \mu_{20} - \mu_{02})$$

Una visualización gráfica de estos dos descriptores puede verse en la figura 4.15.

Estos dos ejes representan dos direcciones privilegiadas en el plano, de gran ayuda para automatizar la localización de los objetos en una escena. Dichas direcciones pueden visualizarse mediante la denominada "elipse de mejor ajuste" [Jain, 89], cuyos semiejes (mayor, a , y menor, b), coinciden en dirección con los ejes de mínima y máxima inercia (I_{min} , I_{max}) respectivamente:

$$I_{min} = \frac{\mu_{20} + \mu_{02} - \sqrt{4\mu_{11}^2 + (\mu_{20} - \mu_{02})^2}}{2}$$

$$I_{max} = \frac{\mu_{20} + \mu_{02} + \sqrt{4\mu_{11}^2 + (\mu_{20} - \mu_{02})^2}}{2}$$

⁴ Véase el apéndice A para función atan2.

calculándose el módulo de dichos semiejes a partir de:

$$a = \left(\frac{4}{\pi}\right)^{1/4} \left[\frac{I_{\max}^3}{I_{\min}}\right]^{1/8} ; b = \left(\frac{4}{\pi}\right)^{1/4} \left[\frac{I_{\min}^3}{I_{\max}}\right]^{1/8}$$

Se observa que ambas características se definen en base a los momentos centrales de segundo orden. Puntualicemos que es además en los momentos de menor orden donde se concentran las principales características de la forma de un contorno cerrado [Maravall, 93].

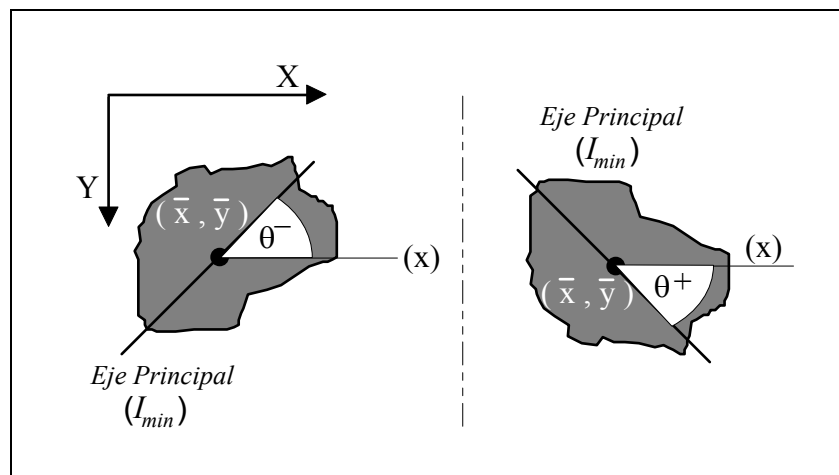


Figura 4.15. Interpretación geométrica de los descriptores utilizados: centroide y orientación (θ) del *Eje Principal*. Obsérvese el convenio seguido para dicha orientación.

En la figura 4.16, se visualiza un ejemplo de una imagen real (alicates) donde se muestran los descriptores calculados, representándose además la elipse de mejor ajuste, de forma que se evidencia la bondad de los mismos. Obsérvese que la dirección del eje principal coincide con la del momento de inercia mínimo (semieje mayor de la elipse), y la del momento de inercia máximo, con la dirección de la recta que pasando por el centroide es ortogonal al eje principal (semieje menor de la elipse). También se observa el convenio seguido en la orientación del eje principal (I_{\max}), en la esquina superior izquierda de dicha figura.

El porqué hemos extraído como descriptores únicamente aquéllos relacionados con el centroide del objeto y la orientación del eje de mínima inercia se justifica, como ya se comentó en la introducción de este capítulo, por

los objetivos que se proponen alcanzar en las etapas siguientes. No podemos perder de vista que el interés será doble: por un lado, garantizar la localización del objeto en la escena y por otro, contar con la información necesaria y suficiente para alimentar la entrada al módulo de determinación de puntos de agarre, desencadenando el razonamiento geométrico subsiguiente (véase el capítulo V). Es decir, se ha de generar la mínima información indispensable para que el robot manipulador pueda finalmente, una vez detectado un agarre estable, ejecutar la acción de coger dicho objeto (véase el capítulo VI).

Una muestra del comportamiento del proceso de localización para diferentes objetos en una escena real, atendiendo al tipo de simetría global que presentan, se visualiza en la figura 4.17.

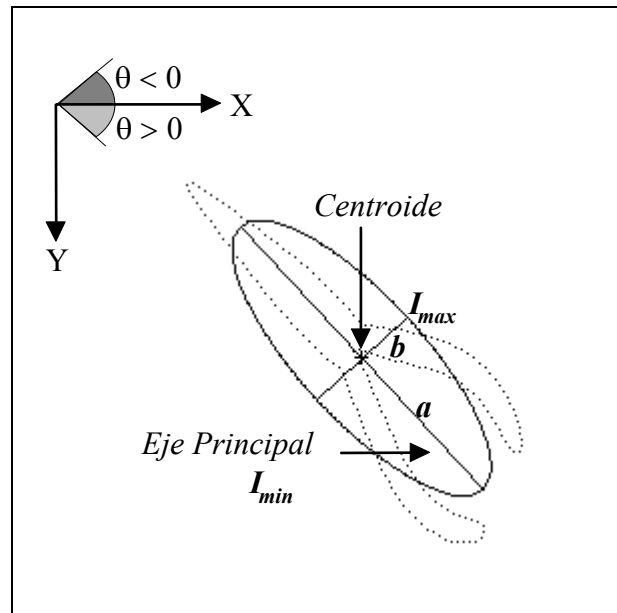


Figura 4.16. Visualización sobre una imagen real (alicates) de la elipse de mejor ajuste, con sus semiejes mayor, a (I_{min}) y menor, b (I_{max}) y de los descriptores utilizados. En este caso el valor de la orientación de I_{min} es de 47° , siguiendo el convenio visualizado en la esquina superior izquierda.

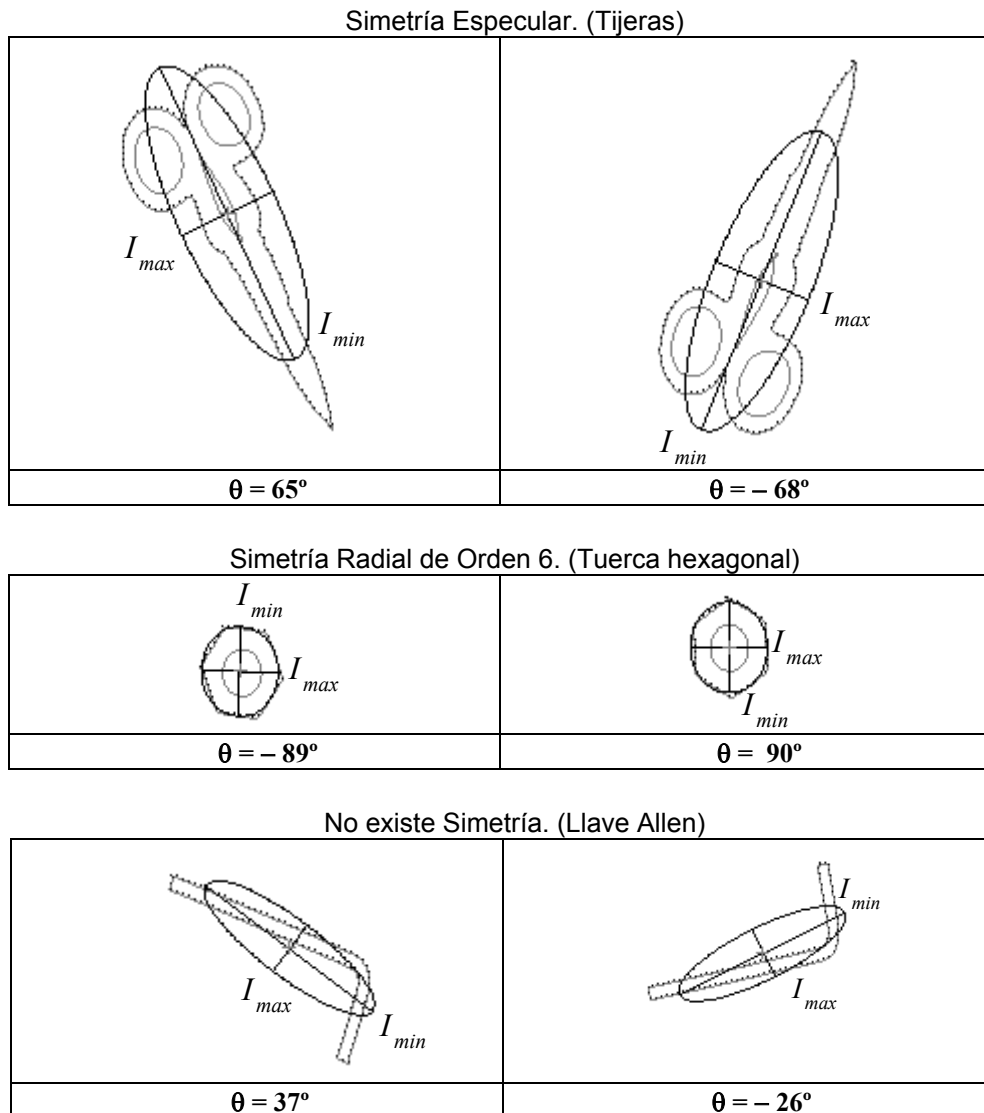


Figura 4.17. Visualización sobre diferentes imágenes del proceso de localización a partir de los descriptores basados en los momentos. Al pie de cada imagen se representa el ángulo (θ) en grados de la orientación del eje de mínima inercia (I_{min}).

Aunque lo referente al concepto de simetría y la notación empleada se relega al capítulo V, es interesante considerarlo aquí para facilitar la interpretación de resultados. Dicho concepto sirve para clasificar las diferentes formas encontradas en el universo de trabajo desde una perspectiva global.

Nótese que en los objetos que presenten simetría radial (e.g. la "tuerca" en la figura 4.17), las direcciones de los ejes de la elipse de mejor ajuste permanecen invariantes respecto a los cambios en la orientación de los mismos.

Es interesante puntualizar que cuando una forma posee un único eje de simetría especular, la dirección de dicho eje coincide con la orientación de alguno de los ejes del elipsoide de inercia, existiendo en el plano dos posibilidades: eje de mínima inercia (I_{min}) del objeto o el de máxima inercia (I_{max}). Dicho de otra manera, si la forma no posee simetría especular respecto de alguno de los dos ejes anteriores (I_{min} o I_{max}) implica que esta no existe respecto de ningún otro eje. Ahora bien, aunque no exista simetría especular, puede existir de otro tipo (e.g. rotacional).

A la vista de la figura 4.17, se observa que en aquellos objetos que presentan simetría radial (e.g. tuerca) las direcciones de los ejes de la elipse de mejor ajuste no aportan información respecto a los cambios en la orientación de los mismos.

4.8 Especificaciones del Algoritmo

A continuación se resumen los pasos fundamentales que resuelven la etapa de adquisición y procesado, visto a lo largo de este capítulo:

1º Captura de la imagen y preprocesado.

2º Segmentación de la imagen.

Entrada: Matriz imagen de 256×256 píxeles y 64 niveles de gris, correspondiente a la porción del área de trabajo de que se trate⁵.

Salida: Matriz imagen pseudo-binarizada, clasificando en tres clases los píxeles pertenecientes a los objetos, al fondo y a la zona de transición, respectivamente.

Estrategia: A partir del histograma (suponiéndolo aproximadamente bimodal), se procede a la búsqueda adaptativa de un umbral óptimo y a la pseudo-binarización [Iñesta, 94].

3º Extracción de contornos.

Entrada: Matriz pseudo-binaria obtenida en el paso anterior.

Salida: Vector de contornos. Cada elemento del vector es un registro donde se guarda la información asociada con cada contorno, el

⁵ No olvidemos que la cámara está montada en la muñeca del robot (véase el capítulo VI).

cual se representa a partir del código de cadena de Freeman 8-direccional [Freeman, 61].

Estrategia: Separación de regiones por encima y por debajo del valor central del rango de niveles de gris, de forma que se sigue la frontera de la zona correspondiente al objeto [Gonzalez & Wintz, 87].

4° Cálculo de momentos.

Entrada: Código de cadena de Freeman obtenido en el paso anterior.

Salida: Momentos geométricos y centrales hasta orden dos.

Estrategia: Cálculo de momentos geométricos y centrales hasta orden dos, a partir del método de la regla delta extendida [Sanz et al., 94].

5° Extracción de características basadas en el cálculo de momentos.

Entrada: Momentos geométricos y centrales hasta orden dos (obtenidos en el paso anterior).

Salida: Coordenadas en píxeles del centroide, y orientación en grados del *eje principal* (I_{min}) de cada contorno encontrado en la imagen.

Estrategia: A partir de las expresiones en función de los momentos del centroide y la orientación del *eje principal* [Jain, 89].

Resumiendo, este algoritmo a partir de la captura de una imagen, entrega una representación de los contornos asociados a los objetos de la misma, cada uno caracterizado con dos descriptores: centroide y orientación del eje principal. Esto es justo lo que se necesita para abordar la siguiente etapa de razonamiento geométrico, con vistas a la determinación de posibles puntos de agarre asociados al contorno exterior de un objeto determinado.

4.9 Conclusiones Preliminares

Se pueden extraer del presente capítulo las siguientes conclusiones:

- ◆ Se ha demostrado la conveniencia de la elección del 3-muestreo por su impacto positivo en la eficiencia de los algoritmos [Sanz et al., 96], respetando simultáneamente la morfología de los objetos. No todo son

ventajas, hemos de puntualizar que este efecto beneficioso en la reducción de datos no es gratuito, pues acaba añadiendo cierta complejidad a los algoritmos de razonamiento geométrico posteriores (véase el capítulo V). De todas formas se constata que son mayores las ventajas que los inconvenientes.

- ◆ Se ha diseñado una estrategia que permite la automatización completa del proceso de segmentación de imágenes, en condiciones adecuadas de luz ambiente difusa, de manera robusta y eficiente. Garantizando la extracción de contornos de objetos de formas cualesquiera.

- ◆ El cómputo de los momentos, necesarios para obtener los descriptores, se realiza a partir de la regla delta extendida [Sanz et al., 94], quedando patente su eficiencia y simplicidad respecto a otros métodos, así como su idoneidad para alcanzar los objetivos planteados. La bondad de su comportamiento se observa en relación al proceso de localización mostrado.

Capítulo V

Determinación Automática de Puntos de Agarre

En este capítulo se abordará de forma concisa, pero sin omitir detalle relevante alguno, el algoritmo que constituye la parte central del trabajo, así como los fundamentos matemáticos y heurísticos de cada uno de sus pasos.

5.1. Introducción

Hasta hace muy pocos años el campo de investigación referente a la problemática relacionada con la planificación del agarre, como se vió en el capítulo II, se ha centrado sobre todo en aspectos de modelización cinemática (e.g. [Montana, 91]), dinámica (e.g. [Mason & Salisbury, 85]), o puramente geométrica (e.g. [Markenshoff et al., 90]), siendo muy reducido el número de trabajos que hacen uso de información sensorial (e.g. visión, sensores de fuerza, de contacto, etc.) para alcanzar sus objetivos. Únicamente trabajos muy recientes, algunos de los cuales figuran todavía como *technical reports*, hacen uso extensivo de visión en las diferentes etapas de los sistemas de agarre propuestos (e.g. [Jarvis, 88], [Stansfield, 91], [Fuentes et al., 94], [Bard et al., 95], [Kamon et al., 96]).

Siguiendo a Stansfield [Stansfield, 91] es importante tener en cuenta la diferencia entre la fase de pre-contacto, anterior al agarre propiamente dicho, y la de post-contacto, posterior al mismo. Debido a que la etapa final del agarre

(post-contacto) ocurre una vez que los dedos de la garra entran en contacto con el objeto, habrá de utilizarse realimentación de información táctil para conseguir un buen ajuste del agarre final, teniendo en cuenta propiedades que no pueden extraerse visualmente (e.g., densidad y peso), con vistas a mantener dicho agarre durante la manipulación del objeto. No obstante, con las restricciones impuestas en el presente trabajo (véase el capítulo III) se puede obviar el uso de la información táctil o de fuerza.

El principal antecedente del algoritmo que desarrollamos a partir del punto 5.2 es el trabajo de Kamon y su grupo [Kamon et al., 94], [Kamon et al., 96], el cual se ha implantado en un sistema real con un robot *AdeptOne* y un sistema de visión consistente en dos cámaras, una primera situada sobre el área de trabajo y otra lateral a la misma (véase el capítulo III).

El presente capítulo se centra en la fase de pre-contacto citada anteriormente, con vistas a la generación y análisis del agarre a partir exclusivamente de información visual, dejando la problemática de la ejecución del mismo para el capítulo VI. Las condiciones iniciales han sido descritas en el capítulo III. Es importante dejar claro que a diferencia de otros trabajos [Jarvis, 88] en los que el objetivo es manipular el objeto sin tener en cuenta nada más que la adaptación de la pinza al contorno del mismo (*preshape*), nosotros estamos interesados en garantizar la unicidad y estabilidad del agarre encontrado. Las razones son evidentes: si el agarre es único evitaremos conflictos de elección y si es estable evitaremos problemas en la fase posterior al mismo (la de transporte del objeto por la mano del robot). Así pues, nuestro algoritmo estará fuertemente inducido por estas consideraciones.

5.2. Descripción del algoritmo

El algoritmo desarrollado para la determinación automática de los puntos de agarre, una vez finalizada la etapa de procesamiento de la imagen descrita en el capítulo anterior, puede descomponerse en las siguientes tres fases:

- 1^a Obtención de Candidatos a Puntos de Agarre (OCPA)
- 2^a Mecanismo Supervisor (MESU)
- 3^a Condiciones de Seguridad (COSE)

Recordemos que aunque la etapa previa de procesamiento de la imagen se describe en el capítulo anterior, ésta siempre estará mediatizada por los objetivos que se pretendan alcanzar; en nuestro caso, la determinación de los puntos de agarre. No obstante, se ha considerado conveniente separar dicha etapa de la presente, orientada al razonamiento geométrico y específica de nuestro problema. Puntualizemos que la fase OCPA anterior garantiza la

unicidad del agarre, definido única y exclusivamente por dos puntos: los candidatos a puntos de agarre. Por otra parte, la fase MESU se ocupa de evaluar/corregir el agarre definido en el paso anterior, de manera que si se satisfacen todas y cada una de las condiciones impuestas por ciertos valores umbrales, diremos que *existe* un posible agarre estable y, en otro caso (si falla alguna de dichas condiciones), no se podrá garantizar su *existencia*. Lógicamente, si se cubre con éxito el criterio de estabilidad, dicho agarre habrá de evaluarse a continuación por la última de las fases del algoritmo: la comprobación de que el agarre es factible por superar las condiciones mínimas de seguridad exigidas en la fase COSE.

A continuación daremos una descripción de alto nivel relativa a cada una de las tres fases anteriores, dejando para el apartado 5.3 la descripción a bajo nivel de las mismas.

OCPA. Obtención de Candidatos a Puntos de Agarre.

Entrada: Vector dinámico de coordenadas visuales de los puntos del contorno y descriptores asociados .

Salida: Coordenadas en píxeles de los puntos P'_1 y P'_2 , candidatos a puntos de agarre.

Estrategia: Cálculo de los dos puntos de corte de la recta perpendicular al *eje principal* por el centroide con los bordes opuestos más externos del contorno.

MESU. Mecanismo Supervisor.

Entrada: Vector dinámico de coordenadas visuales de los puntos del contorno, descriptores asociados y candidatos a puntos de agarre, obtenidos por OCPA.

Salida: Resultado de la variable lógica *Estable*, tal que si $Estable = 1$, se garantiza con "certeza" la estabilidad del agarre, devolviendo las coordenadas en píxeles de los puntos P_1 y P_2 que lo describen; si $Estable = 0$, no se garantiza estabilidad.

Estrategia: Una vez encontrados los dos candidatos a puntos de agarre (P'_1 y P'_2), necesitamos evaluarlos en relación a las tres condiciones de estabilidad expuestas anteriormente¹; es decir:

- 1^a Garantizar la constancia de la normal en un entorno de radio k (radio de la pinza), centrado en dichos puntos.
- 2^a Que los ángulos entre la línea de agarre y las normales en dichos puntos del contorno no superen un valor umbral.

¹ Véase el capítulo III, en relación a los tres umbrales introducidos.

3ª Que la distancia entre la línea de agarre y el centroide no supere un cierto umbral.

COSE. Condiciones de Seguridad.

Entrada: Coordenadas en píxeles de los puntos P_1 y P_2 , que describen el agarre final asociado a cada objeto, con un grado de confianza ($Estable = 1$) que garantiza la estabilidad del agarre, obtenidos en el paso anterior.

Salida: Coordenadas en píxeles de los puntos P_1 y P_2 , que describen el agarre final asociado a cada objeto, con un grado de confianza entre 0 y 1; 1 cuando se garantiza con "certeza" la seguridad del agarre y 0 en otro caso.

Estrategia: Una vez encontrados los dos puntos de agarre (P_1 y P_2), necesitamos evaluar la distancia entre los dos puntos de agarre, para garantizar que no exceda la apertura máxima de la pinza, así como asegurar la existencia de un camino, libre de obstáculos, en la trayectoria seguida por los dedos de la pinza al efectuar la acción de agarrar.

5.3. El Algoritmo a bajo nivel.

Una vez justificada la reducción de datos mediante L_{m3} (véase el apartado 4.5.1.) tenemos como entrada a nuestro algoritmo un vector dinámico con las coordenadas visuales de los puntos del contorno 3-muestreado. Además, para cada contorno se ha determinado su centroide y la orientación del *eje principal*. A partir de esta información inicial se ha elaborado un procedimiento heurístico que, razonando geoméricamente sobre la representación anterior, facilita una búsqueda eficiente y robusta del agarre.

5.3.1. Obtención de candidatos a puntos de agarre.

Antes de mostrar los pasos seguidos por el algoritmo que nos ocupa es necesario introducir la noción de intersección discreta utilizada, inducida por la granularidad inherente al 3-muestreo que representa un contorno genérico. En la figura 5.1 se visualiza la interpretación geométrica del intervalo cerrado de centro P_i y de radio $\delta > 0$ tal que, centrado en los puntos de un contorno, es necesario imponer en relación a su posible intersección con una recta determinada (de dirección I_{min} o I_{max} , normalmente), dicho radio se obtiene a partir del peor de los casos posibles, es decir, cuando más alejado se puede encontrar un punto del contorno de la recta correspondiente.

Veamos, a continuación, la estrategia adoptada para la obtención de candidatos a puntos de agarre, la cual viene definida por la siguiente heurística:

- OCPA_1** A partir del *eje principal* (dirección del eje de mínima inercia: I_{min}), se calcula la recta ortogonal pasando por el centroide (dirección del eje de máxima inercia: I_{max}).
- OCPA_2** A continuación se buscan y guardan todos los puntos intersección entre esta línea (dirección del eje de máxima inercia: I_{max}) y el contorno 3-muestreado del objeto (C), es decir, se construye el conjunto de "precandidatos": $\Omega_i = \{P'_i / P'_i \in I_{max} \cap C\}$.
- OCPA_3** Debido a la granularidad de los contornos, a causa del 3-muestreo empleado, la condición anterior no garantiza un único punto de corte del contorno con I_{max} ; así pues, del conjunto anterior (Ω_i) se eliminan aquellos puntos que son contiguos, obteniéndose el conjunto Ω_f de todas las intersecciones, univaluadas, del contorno con I_{max} .
- OCPA_4** A partir del conjunto de puntos obtenido (Ω_f) se seleccionan únicamente aquellos dos puntos más externos del contorno en esa dirección y, en principio, serán un buen par de candidatos a puntos de agarre.

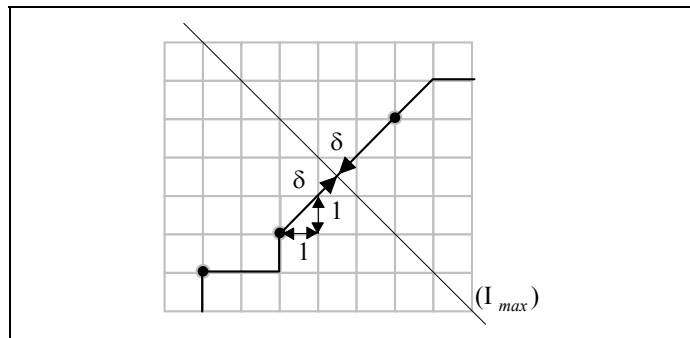


Figura 5.1. Interpretación geométrica de la definición de intersección (contorno, recta) discreta empleada. Se representa el caso más desfavorable en cuanto a distancia se refiere. A partir del mismo se obtiene el radio utilizado para la intersección discreta en el algoritmo ($\delta = 2$).

Visualizamos en la figura 5.2 un caso típico de herramienta: *alicates*, que permite una rápida comprensión de la problemática inherente al objetivo que se intenta alcanzar. El seguimiento del algoritmo se facilita observando el ejemplo de dicha figura. En primer lugar se impone un umbral de distancia, δ , $d(\mathbf{C}, I_{max})$, de los puntos del contorno 3-muestreado a la recta de dirección I_{max} , de manera que aquellos puntos del contorno cuyo entorno cerrado de radio δ intersekte con I_{max} , formarán un primer conjunto de puntos definido por:

$$\Omega_i = \{P'_i / P'_i \in I_{max} \cap \mathbf{C}\}$$

Es decir, debido a la naturaleza granular de los contornos, provocada por el 3-muestreo realizado, hemos de traducir la condición anterior a la siguiente:

$$\Omega_i = \{P'_i / d(P'_i, I_{max}) \leq \delta\}$$

En nuestro caso, como se ha demostrado (véase la figura 5.1) $\delta = 2$ píxeles. Luego esta es la condición para formar el conjunto de puntos $\Omega_i = \{P'_i / P'_i \in I_{max} \cap \mathbf{C}\}$. Estos puntos se guardan en un vector auxiliar, donde también se guardarán, posteriormente, las distancias a un punto, χ , externo al contorno en la dirección I_{max} . El estado de dicho vector, llegados a éste punto, para el ejemplo representado en la figura 5.2, será el siguiente:

$$\Omega_i = [((150,110),0); ((153,110),0); ((139,121),0); ((133,125),0); ((127,132),0)]$$

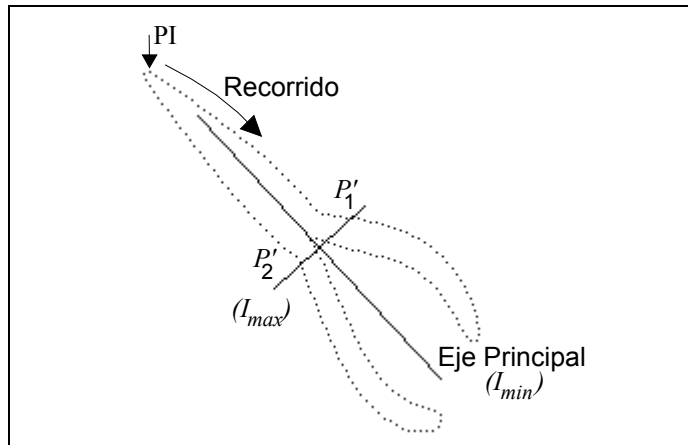


Figura 5.2. Interpretación geométrica del algoritmo descrito en el paso OCPA, sobre un objeto real: *alicates*. Obsérvese la granularidad del contorno 3-muestreado. Existen cuatro puntos teóricos de corte, $P'_i \in \{\mathbf{C} \cap I_{max}\}$.

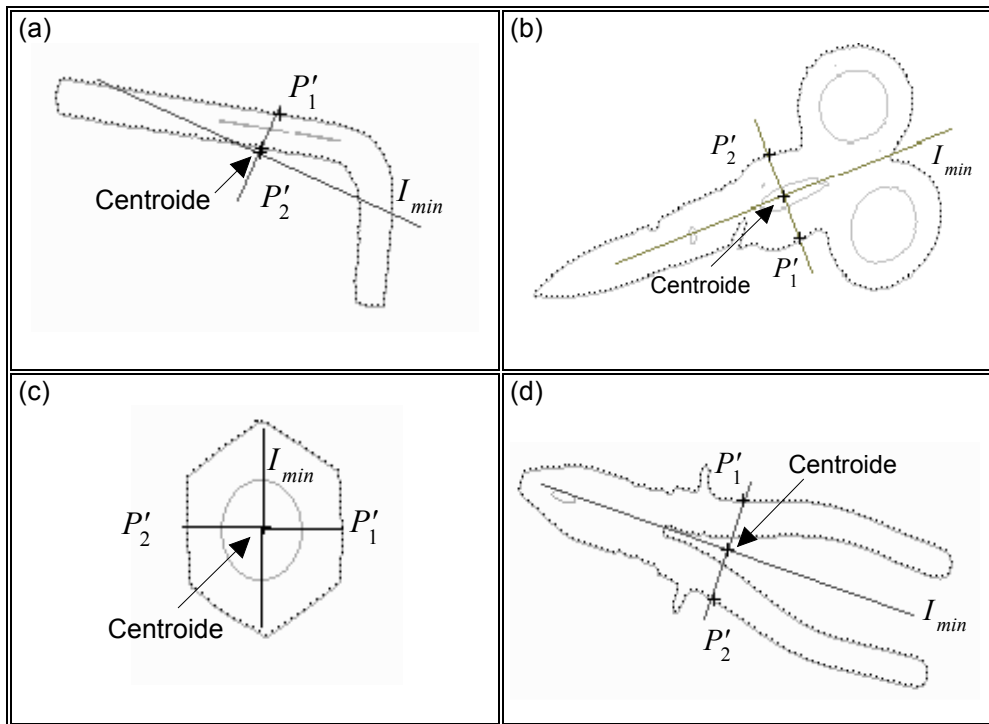


Figura 5.3. Comportamiento del algoritmo OCPA para diferentes imágenes del universo de trabajo: (a) llave allen, (b) tijeras, (c) tuerca, y (d) tenazas.

Problema: se han encontrado cinco puntos en lugar de los cuatro esperados.

Solución: La condición de distancia impuesta a priori, teniendo en cuenta que los puntos del contorno están espaciados una cierta distancia, conlleva el que, además, filtremos los puntos contiguos. Es decir, en nuestro caso, a partir del punto inicial se recorre el contorno y se llega al punto número 35 (a partir del inicial), de coordenadas (150,110), cuya distancia a I_{max} es menor que δ , pero el punto siguiente, número 36, de coordenadas (153,110), también cumple dicha condición, por tanto se almacenaría en el vector, como vemos que ocurre. Filtrando puntos contiguos desaparece el problema, y se obtiene lo que deseamos:

$$\Omega_f = [((150,110),0); ((139,121),0); ((133,125),0); ((127,132),0)]$$

Es decir, un único punto de corte en cada intersección del contorno con I_{max} .

A partir de aquí se determina fácilmente que los dos puntos más externos son:

$$P'_1 = (150,110) \text{ y } P'_2 = (127,132).$$

Tal y como se ha descrito anteriormente, se utiliza un vector auxiliar para ir guardando cada uno de los puntos intersección de la dirección de I_{max} con el contorno de la imagen. A continuación, siguiendo el paso OCPA_4 del algoritmo, se seleccionan los dos más externos al contorno, es decir, el de máxima y mínima distancia al punto χ , recordando que todos los puntos se encuentran alineados en la recta de dirección I_{max} . Estos puntos, P'_1 y P'_2 , serán en principio un par de precandidatos a puntos de agarre.

Con la estrategia aquí descrita se garantiza siempre la obtención de dos puntos candidatos (existencia y unicidad), descansando el resto del esfuerzo referente a comprobar si se alcanza un agarre estable, en la fase MESU del algoritmo. En la figura 5.2. se visualiza el comportamiento del algoritmo con diversos objetos del universo de trabajo, incluida la llave allen.

5.3.2. Mecanismo supervisor.

El paso anterior (OCPA) demuestra la existencia de un posible agarre y, tal y como se determina, será univaluado, pero no garantiza ni la preadaptación de la garra (pinza de dos dedos plano-paralelos, en nuestro caso) al contorno del objeto, ni su estabilidad. Por tanto, hemos de proceder al análisis y evaluación del agarre inicialmente encontrado, desencadenando acciones correctoras en caso necesario.

La primera cuestión a resolver será la relativa a la preadaptación de la pinza de dedos plano-paralelos (descrita en el capítulo III) al contorno del objeto. Este tópico fundamental de la planificación del agarre en general (*preshape*), se puede reducir, en nuestro caso particular, a la búsqueda de dos condiciones:

- (1.1) Constancia de la normal al contorno a lo largo de cada uno de los intervalos, de radio k (radio de la pinza), centrados en los respectivos puntos de agarre.
- (1.2) Que los ángulos formados por la línea de agarre ($P'_1 P'_2$) y las normales en cada uno de dichos puntos ($N_{P'_i}$) sean suficientemente pequeños.

La condición (1.1) se desarrollará en el punto 5.3.2.3 y básicamente estará relacionada con el análisis de la curvatura discreta en los contornos de los objetos, definiéndose un umbral de curvatura (α). Así mismo, la condición (1.2) será analizada en el punto 5.3.2.4, donde se definirá un umbral angular (β). Esta última condición será la responsable de incorporar la denominada en el capítulo III, "condición de no-deslizamiento", relacionándose este umbral (β) con el llamado "ángulo de rozamiento", allí expuesto.

La otra cuestión que analiza el mecanismo supervisor que nos ocupa, será la de la estabilidad (expuesta en el capítulo III). Tenemos de nuevo dos condiciones a imponer:

(2.1) La línea de agarre ($P'_1 P'_2$) habrá de mantenerse próxima al centroide.

(2.2) Dicha línea de agarre ($P'_1 P'_2$) será, aproximadamente, normal al contorno en ambos puntos.

Como la condición (2.2) está subsumida en la condición (1.2), impuesta para la preadaptación de la pinza, sólo habremos de preocuparnos de la condición (2.1), la cual se desarrolla en el punto 5.3.2.5, donde se define el umbral de distancia (γ).

Además de lo anterior, será necesario, para facilitar el análisis de las condiciones anteriores, contar con la ayuda de una propiedad clave, inherente a la forma: la simetría; a desarrollar en el punto 5.3.2.1.

Finalmente, solo nos restará exponer, en el punto 5.3.2.6, la estrategia utilizada para sintetizar, en un algoritmo, todo el proceso a seguir en la supervisión de las condiciones aquí expuestas.

5.3.2.1. Acerca de la simetría.

La importancia del concepto de simetría queda reflejada en la gran cantidad de disciplinas que hacen uso de ella en alguna medida, ya sea para establecer leyes de comportamiento en ciertos campos de la física [Joshua, 91] o caracterizar formas planas para su posterior reconocimiento, manipulación, etc. [Brady & Asada, 84], [Van Gool et al. 95]; esta última aplicación dentro ya del campo de nuestro interés: la robótica.

Desde un punto de vista matemático se plantea la existencia de tres tipos básicos de transformaciones [Ledermann, 85]: rotaciones, reflexiones y traslaciones que constituyen los bloques elementales para las simetrías que subyacen en la geometría Euclídea, denominadas isometrías, en el sentido de que cada isometría del espacio Euclídeo puede formarse por la combinación de rotaciones, reflexiones y traslaciones. Además, dichos tipos de simetrías son las observadas con mayor frecuencia en la naturaleza y en los objetos hechos por el hombre.

En el presente trabajo, antes de proceder a la sucesiva evaluación de los distintos umbrales planteados, interesa extraer información acerca del tipo de simetría existente [Zabrodsky, 93] en un contorno dado. Esto es así porque para poder desplegar el razonamiento geométrico propio de un mecanismo evaluador-corrector como el que se propone, necesitamos pasar con facilidad

del análisis global de los contornos al análisis local de los mismos. En concreto interesa conocer si el eje principal es eje de simetría o no, es decir, si existe simetría especular respecto de dicho eje. Ejemplo de objetos que poseen simetría especular (respecto del eje principal), serán la mayor parte de herramientas con las que trabajamos: tenazas, tijeras, destornillador, etc.; por contra, ejemplos que no poseen dicha simetría pueden ser: calibrador, llave allen, martillo, etc. Además, pueden aparecer objetos con simetría cuasi circular como tuercas, etc.

Tengamos en cuenta que en el problema de la determinación de puntos de agarre que nos ocupa, la existencia de simetría especular respecto de la orientación del eje principal (I_{min}) facilita en gran medida el razonamiento geométrico posterior y prueba de ello es que algunos investigadores consideran este caso preferentemente para evitar complicaciones y poder focalizar su atención en otros aspectos del agarre como el aprendizaje [Kamon et al., 96].

En lo que sigue vamos a utilizar la siguiente notación [Zabrodsky et al., 95]:

Simetría Especular

Una forma (2D) posee simetría especular si es invariante bajo la reflexión en una dirección, línea representada por el eje de simetría especular.

Simetría Rotacional de Orden n (C_n)

Una forma (2D) posee simetría C_n si es invariante bajo rotación de $2\pi/n$ radianes alrededor de su centroide. Un caso particular de simetría C_n es la circular, que representa la simetría C_∞ .

Simetría Radial de Orden n (D_n)

Una forma (2D) posee simetría D_n si posee tanto simetría especular como simetría C_n .

Además, si una determinada forma no se ajusta a ninguna de las definiciones anteriores se dice, en principio, que no posee simetría (véase la figura 5.4). Es interesante resaltar, a partir de la figura 5.4, lo siguiente:

- Existen ejes prioritarios para la simetría especular, los ejes principales de inercia (en el plano: I_{min} e I_{max}). De existir dicha clase de simetría estas son las primeras direcciones en acusarlo.
- Por otra parte, el que no se cumpla con la definición *sensu strictum* no quiere decir que no exista simetría en absoluto. De hecho observando la figura, vemos que el calibre (c), es más simétrico que la llave allen (d).

- Finalmente, en (a) se observa un objeto con simetría radial de orden $n = 6$ (D_6); ya que, dado un punto cualquiera del contorno, si lo rotamos $2\pi/6$ radianes se obtiene otro punto con igual distancia al centroide. Se puede observar, así mismo, la existencia de 6 ejes de simetría especular.

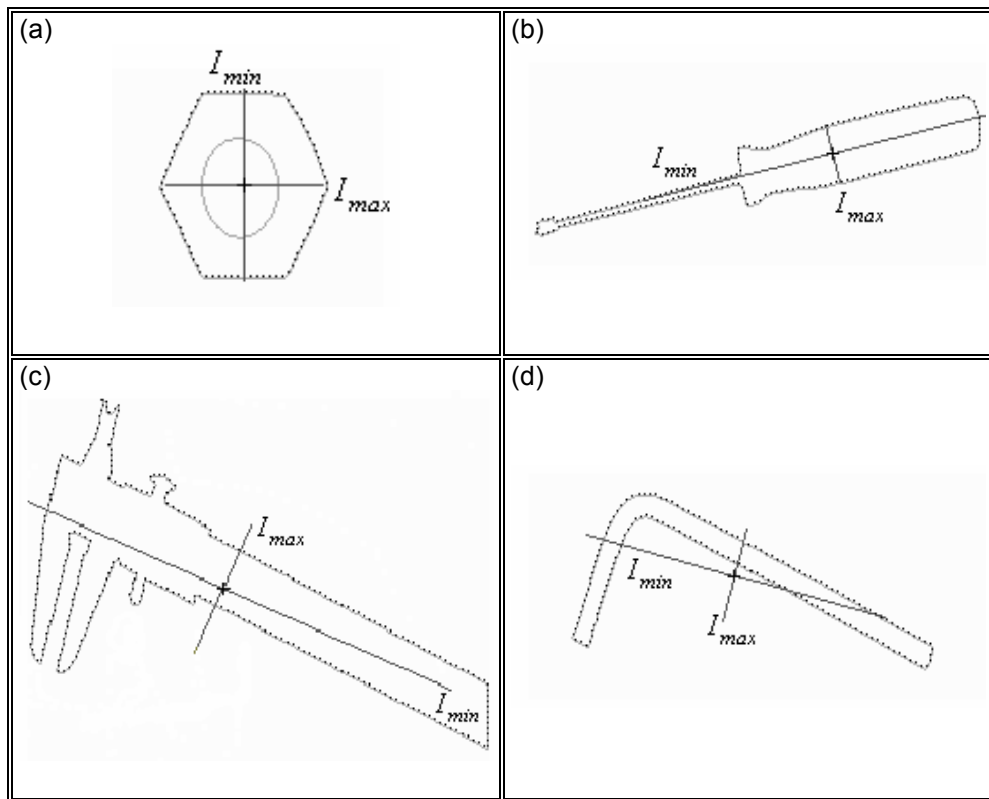


Figura 5.4. Por orden de simetría decreciente: (a) Simetría D_6 -tuerca-; (b) Especular -destornillador-; (c) No existe -calibre-; (d) No existe -llave allen-.

En conexión con el agarre, algunos autores se han valido de métodos puramente geométricos para determinarlos. Un artificio empleado [Markenscoff et al., 90], particularmente dirigido a formas poligonales, consiste en determinar todas las regiones que garantizan puntos de agarre antípodos (visto en el capítulo III), mediante el uso de circunferencias inscritas. Este enfoque no es aplicable al universo de trabajo aquí propuesto, pero la idea de propagar distancias desde el centroide a puntos opuestos del contorno exterior, es muy útil y será tenida en cuenta en ciertos casos, en particular cuando exista simetría radial, como es el caso de la figura 5.5.

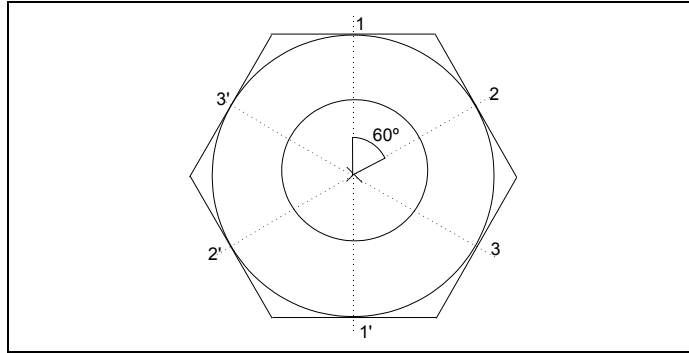


Figura 5.5. En un hexágono ideal la circunferencia inscrita tocaría el polígono en 6 puntos, perpendicularmente opuestos dos a dos, dando lugar a tres posibles agarres (1,1'), (2,2'), y (3,3'). Se observa, así mismo, la existencia de seis ejes de simetría especular.

Ahondando en esta idea de utilizar círculos como ayuda a la caracterización del agarre plano, existen algunos trabajos recientes [Blake, 95] que hacen uso de círculos bitangentes al contorno, para encontrar así todos los pares de puntos que en potencia constituyen agarres planos (ver la figura 5.6). Dichos pares de puntos forman un conjunto denominado simétrico, cumpliendo: $\tan\alpha_1 = -\tan\alpha_2$, que garantiza, como se vió en el capítulo III, clausura de fuerza. Nótese que lo anterior es aplicable únicamente a contornos de curvatura suave, generalmente modelados a partir de *B-splines*.

Volviendo al trabajo que nos ocupa, como entrada al algoritmo MESU, se cuenta con una dirección privilegiada en el espacio (I_{min}), así como los dos puntos del contorno candidatos al agarre, por tanto, interesa sacar partido a dicha información de la forma más efectiva posible, con vistas a evaluar el tipo de simetría existente. A priori, lo más eficiente será plantear las dos condiciones siguientes:

Condición 1. Caracterización de la simetría especular respecto del eje principal (I_{min}):

$$d(P'_1, cdm) = d(P'_2, cdm) \pm \tau_1$$

donde τ_1 representa una tolerancia establecida empíricamente.

Condición 2. Caracterización de la simetría especular respecto del eje I_{max} :

$$d(P_3, cdm) = d(P_4, cdm) \pm \tau_2$$

donde τ_2 representa una tolerancia establecida empíricamente, como en el caso anterior y P_3 y P_4 son los dos puntos de corte de I_{min} con el contorno .

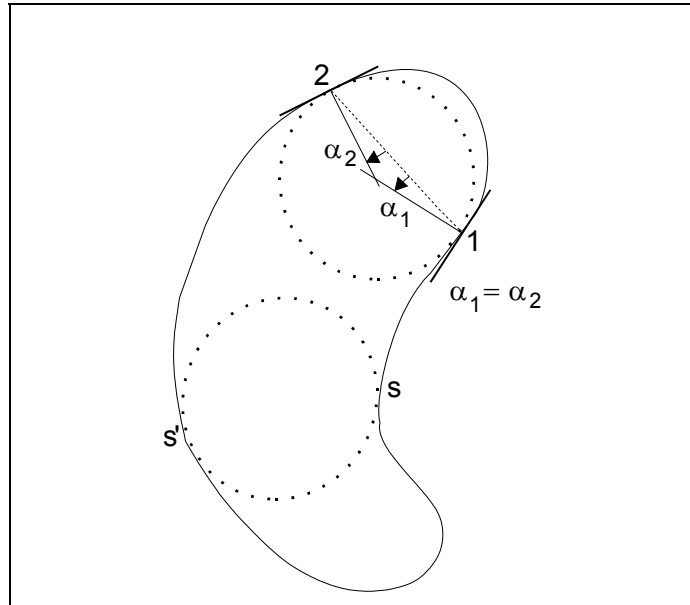


Figura 5.6. Adaptada de [Blake, 95]. El conjunto de todos los pares de puntos de la curva (s, s'), que yacen en círculos bitangentes a la misma, forman el denominado conjunto simétrico que caracteriza todos los agarres extremos existentes.

Si se verifican las dos condiciones anteriores tenemos un claro indicio de la existencia de simetría radial. Si únicamente se verifica la condición 1 o 2, existirá simetría especular respecto de I_{min} o I_{max} , respectivamente, y sino se verifica ninguna de las dos será indicativo de que no existe simetría especular en este caso. Estas consideraciones fueron ya comentadas en el capítulo IV, apartado 4.7, en relación con el elipsoide de inercia y sus direcciones principales.

5.3.2.2. Análisis global vs análisis local.

Para desencadenar el razonamiento geométrico necesario a nuestro objetivo de la determinación del agarre, se requiere encontrar ciertas regiones del contorno de un objeto cumpliendo determinadas condiciones modeladas en base a la definición de los tres umbrales introducidos en el capítulo III. Para ello es obligado:

- 1º Incorporar la geometría de la pinza, trasladando la longitud del radio de sus dedos, k , a los puntos del intervalo equivalente sobre el propio contorno.
- 2º Evaluar la curvatura localmente en cada punto del contorno, almacenando el resultado en un vector (vector de torsiones).

- 3° Incorporar el conocimiento sobre la simetría del contorno, evaluado a partir de la información anterior (ver más adelante), con vistas a simplificar el algoritmo final (MESU). Si hay que trasladar el intervalo de radio k , centrado en los candidatos a puntos de agarre, con vistas a encontrar regiones donde se cumplan las condiciones de estabilidad, impuestas por los tres umbrales vistos anteriormente, será necesario encontrar las posibles direcciones de mejora en base a una determinada propiedad (e.g. la curvatura), empleando para ello el conocimiento previo de la simetría.

A continuación se desarrollan cada uno de los puntos considerados.

1° Acerca del tamaño del intervalo del contorno a considerar

Teniendo en cuenta la presencia del ruido de cuantización en los contornos así como la posibilidad de que estos sean abruptos (con salientes, etc.), habrá que traducir la distancia definida por el radio de los dedos de la pinza en un número de puntos equivalente sobre el contorno 3-muestreado. Para ello habrá de proyectarse el contorno en la dirección del dedo de la pinza. Es decir, en la tangente al contorno en cada punto definido, en principio, por los candidatos a puntos de agarre. Nótese que esta dirección irá cambiando a medida que se trasladen dichos puntos sobre el contorno, tratándose así de un intervalo de curva dinámico que podrá cambiar tanto de dirección como de tamaño (número de puntos que lo forman en cada instante). Una interpretación geométrica puede observarse en la figura 5.7; en este caso particular se requiere un intervalo $k(p_i) = 5$; el tratamiento se ha mostrado para p_i , pudiendo ser este cualquiera de los candidatos a puntos de agarre (P'_1 o P'_2) entregados por el paso anterior (OCA).

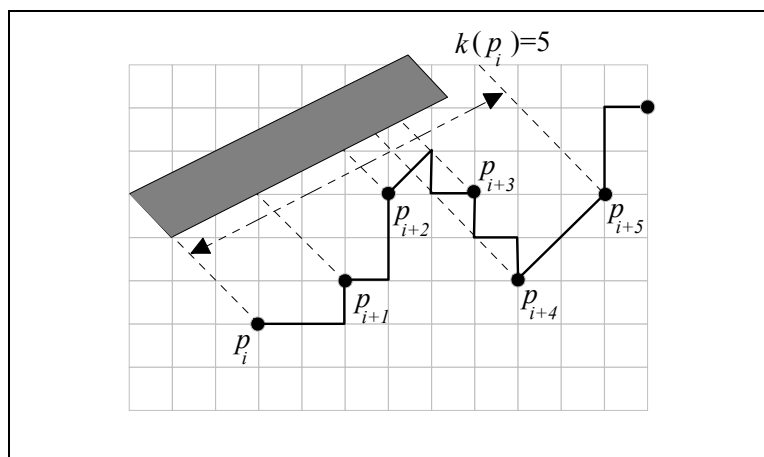


Figura 5.7. Interpretación geométrica de la construcción del intervalo del contorno equivalente a la longitud de la pinza, proyectada según la dirección tangente a p_i .

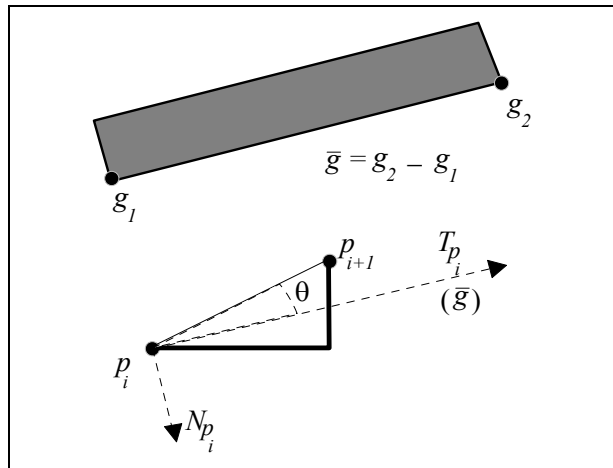


Figura 5.8. (Continuación de la figura 5.7). La orientación de la pinza, g , se obtiene a partir de la tangente a p_i , T_{p_i} , acumulándose las distancias, $p_i p_{i+n}$, hasta alcanzar el radio de la pinza (conocido a priori). En la figura 5.7, se obtiene $n = 5$.

A partir de la figura 5.8 es directo seguir el cálculo de distancias siguiente:

$$d(p_{i+1}, p_i) = (x_{p_{i+1}} - x_{p_i}, y_{p_{i+1}} - y_{p_i}) \cos \theta = \overline{p_i p_{i+1}} \cdot \cos \theta$$

siendo $\cos \theta$

$$\cos \theta = \frac{\overline{p_i p_{i+1}} \cdot \overline{T_{p_i}}}{|\overline{p_i p_{i+1}}| \cdot |\overline{T_{p_i}}|}$$

Las distancias así calculadas se van acumulando hasta que se alcanza con seguridad el tamaño del radio de la pinza, k (en píxeles, conocido a priori), en el ejemplo de la figura 5.7, este intervalo equivale a cinco puntos del contorno 3-muestreado.

Resumiendo, como resultado de esta operación se obtendrá el intervalo $[p'_{i-k}, p'_{i+k}]$ del contorno, equivalente en cada instante a la longitud de la pinza orientada en la dirección del posible agarre.

2º Análisis local a partir de la curvatura

Nuestro objetivo será garantizar la constancia de la normal en un entorno de radio k , centrado en los respectivos puntos de agarre. Dicho análisis puede inferirse a partir del conocimiento de la curvatura en dicho intervalo $[p_{i-k}, p_{i+k}]$.

En general, la curvatura en un punto da una medida de cómo se separa la curva de la tangente en dicho punto; es decir, de cómo se separa la curva del

comportamiento rectilíneo. De hecho, existe un corolario que afirma: "la condición necesaria y suficiente para que una curva sea de curvatura nula es que sea una recta". Por otra parte, la normal en un punto de la curva podremos definirla como la recta que pasando por él es perpendicular a la tangente en dicho punto.

Trasladaremos a continuación los conceptos anteriores de curvatura y normal en un punto a nuestro espacio discreto. En dicho contexto, para el cálculo de la curvatura discreta, se han desarrollado una serie de "medidas de significancia" (denominadas así en la literatura), de las cuales las más interesantes para nuestros fines son [Teh & Chin, 89]:

k -curvatura [Groan et al., 78]

k -torsión angular [Pla, 93]

k -coseno [Rosenfeld & Johnston, 73].

Tanto la k -curvatura como la k -torsión angular, ofrecen medidas más robustas al ruido de cuantización de la curva que el k -coseno, debido al promediado empleado en ambas. Sin embargo, después de innumerables pruebas sobre nuestras imágenes, no se detectó ninguna ventaja en el uso de una de las medidas con respecto a las otras dos. Así pues, la elección de una de ellas debería responder a criterios de predictibilidad del resultado a obtener o de sencillez o eficiencia de la implementación del procedimiento de medida.

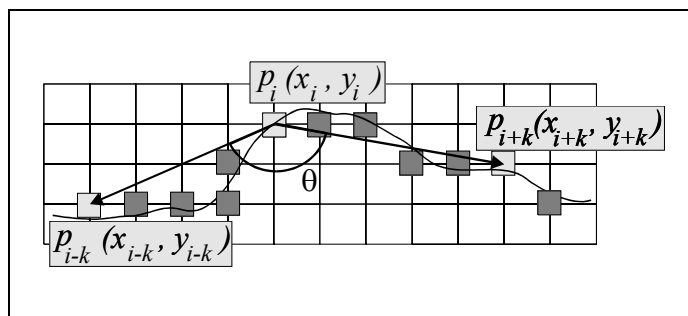


Figura 5.9. Interpretación geométrica de los k -vectores, utilizados para el cálculo de la curvatura discreta.

Con la notación introducida en la figura 5.9, se definen los k -vectores en un punto genérico del contorno, p_i , como

$$\vec{a}_{ik} = (x_i - x_{i+k}, y_i - y_{i+k})$$

$$\vec{b}_{ik} = (x_i - x_{i-k}, y_i - y_{i-k})$$

y el k -coseno en p_i como,

$$\cos_{ik} = \frac{\vec{a}_{ik} \cdot \vec{b}_{ik}}{|\vec{a}_{ik}| \cdot |\vec{b}_{ik}|}$$

Aquí \cos_{ik} es el coseno del ángulo formado por los k -vectores \vec{a}_{ik} y \vec{b}_{ik} , de manera que: $-1 \leq \cos_{ik} \leq 1$. De donde obtendremos el ángulo, θ , en grados, considerando²,

$$\theta = \text{atan2}(\text{sen}_{ik}, \cos_{ik})$$

El valor obtenido para θ , normalmente en radianes, deberá ser traducido a grados sexagesimales multiplicando por el factor $180 / \pi$.

De lo anterior, se observa el significado geométrico directo que extraemos para la curvatura, e.g. si $\theta = 180^\circ$, significa *curvatura nula*, etc. Para nuestros fines será conveniente asignar a dicho valor una tolerancia.

Sí en lugar de calcular el ángulo entre los k -vectores directamente, como se hace para el k -coseno, se definen un punto anterior, p_a , y otro posterior, p_b , al punto considerado p_i , promediando entre cada uno de los k puntos anteriores y posteriores al central, se obtienen los k -vectores siguientes:

$$\vec{a}_{ki} = (x_a - x_i, y_a - y_i)$$

$$\vec{b}_{ki} = (x_i - x_b, y_i - y_b)$$

definiéndose la k -torsión angular como el ángulo entre ambos vectores:

$$\tau_{ki} = \vec{a}_{ki} \wedge \vec{b}_{ki}$$

Es importante observar que la k -torsión angular es en esencia un k -coseno suavizado y expresado en términos de ángulos, y cuando se toman únicamente dos puntos consecutivos, 1 -torsión, el comportamiento coincide con el del 1 -coseno. Es interesante computar para todos los puntos del contorno la 1 -torsión, que se almacenará en un vector (vector de torsiones). Se observa la necesidad de suavizar dicha información para filtrar el ruido inherente a los contornos digitales, utilizando para ello un filtro gaussiano (el mismo que se usó en el capítulo IV):

$$G(z, \omega) = \frac{1}{\sqrt{2\pi\omega}} e^{-\frac{z^2}{2\omega^2}}$$

con anchura de la gaussiana, ω , fijada empíricamente a 2.5.

Por último, referente a la k -curvatura, se define esta como la diferencia entre la dirección angular media de los k segmentos anteriores y posteriores a p_i , en la región de radio k definida, a partir de las direcciones, c_i , de la representación de Freeman utilizada:

² Véase el apéndice A, para concepto y algoritmo empleado para la función *atan2*.

$$\kappa_{ki} = \frac{1}{k} \sum_{j=-k}^{-1} c_{i-j} - \frac{1}{k} \sum_{j=0}^{k-1} c_{i-j}$$

No obstante, esta medida de la curvatura no se ha considerado necesaria en la implementación final del presente trabajo.

3° Análisis global a partir de la simetría

A partir del conocimiento de la simetría introducido en el apartado anterior, se obtiene una caracterización global de la morfología de los objetos, que puede combinarse con el conocimiento local de la curvatura en cada punto del contorno para obtener una potente herramienta que facilite el razonamiento geométrico. De hecho a partir del cálculo de la curvatura en cada uno de los puntos de un contorno, representado mediante el vector de 1-torsiones, junto con la posición del punto en dicho contorno a partir del punto inicial, se obtiene una estructura adecuada para visualizar la simetría y además cuantificarla.

Para visualizar el impacto de la simetría sobre el mecanismo evaluador/corrector de la curvatura se consideran tres casos, en orden de simetría decreciente. El objetivo será utilizar esta información para trasladar los intervalos de radio k , centrados en P'_1 y P'_2 , buscando regiones del contorno que cumplan las condiciones de suavidad en la curvatura impuestas por el umbral definido al efecto (α).

Caso 1: Simetría radial (véase la figura 5.10). Una vez detectada la alta curvatura existente en P'_1 y P'_2 y teniendo en cuenta que se trata de simetría radial (evaluada previamente), se aplica un giro (horario, en dicha figura) para buscar regiones donde la curvatura disminuya. Nótese que sino se giraran P'_1 y P'_2 en el mismo sentido (horario o antihorario) se produciría un alejamiento de la línea de agarre respecto del centroide, lo cual (véase el capítulo III) conlleva inestabilidad. El mecanismo de parada, en este caso de simetría radial, viene impuesto por la condición añadida de mínima distancia al centroide.

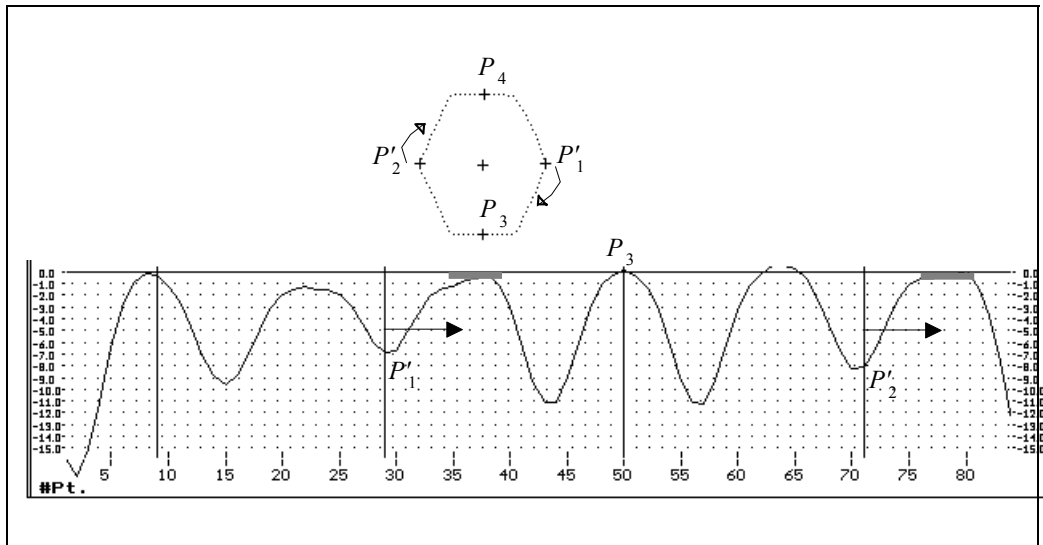


Figura 5.10. Tuerca. Representación (curvatura , punto) del vector de torsiones asociado al contorno de la imagen. Las flechas marcan las posibles direcciones de mejora en la disminución de la curvatura. Los dos rectángulos indican el agarre seleccionado.

Caso 2: Simetría especular (véase la figura 5.11). En este caso las direcciones de mejora a partir de P'_1 y P'_2 son siempre en sentidos opuestos una de otra (hacia P_3 o hacia P_4) siguiendo el recorrido horario a partir del punto inicial del contorno PI. El problema surge, cuando en la búsqueda de regiones de curvatura suave dentro del umbral (α) se sobrepasa el umbral de distancia (γ) al centroide, pues indica inestabilidad del agarre.

Caso 3: No existe simetría (véase la figura 5.12). Al no existir simetría no se puede actuar como en los casos anteriores; es decir, el movimiento de un intervalo no sabemos como afectará al intervalo opuesto. Esto requiere un mecanismo más costoso computacionalmente hablando, que evalúe el impacto sobre la línea de agarre cada vez que se traslade un intervalo, y obrar en consecuencia.

La idea de cuantificar el grado de simetría asociado a una imagen ha sido tratada con éxito, de forma general y sin restricciones temporales, en otros trabajos [Zabrodsky, 95]. No obstante, con los requerimientos de tiempo real que exige la robótica se ha seguido un enfoque simplificado con vistas a disminuir la carga computacional, centrandose exclusivamente en calcular el grado de simetría especular existente, en relación al eje principal asociado al

contorno. Para ello a partir del vector de torsiones obtenido anteriormente y conociendo los puntos de corte $[P_3, P_4] \in \{C \cap I_{min}\}$, se propaga Δ_n , tal que:

$$\Delta_n = 1_torsión(P_3 - n) - 1_torsión(P_3 + n); \quad n = 1, 2, \dots, N/2$$

siendo $P_3 \in \{C \cap I_{min}\}$, en la región del contorno orientada (sentido horario a partir del PI) comprendida entre P'_1 y P'_2 . Es decir, se calcula: Δ_n , entre cada par de puntos equidistantes de P_3 , hasta cubrir todo el contorno. Para ello se sigue un tratamiento de vector circular en la manipulación del vector de torsiones asociado a un contorno dado. Al final se obtiene la que llamaremos deficiencia simétrica global normalizada:

$$\Phi = \frac{1}{N} \sum_{n=1}^{N/2} \Delta_n$$

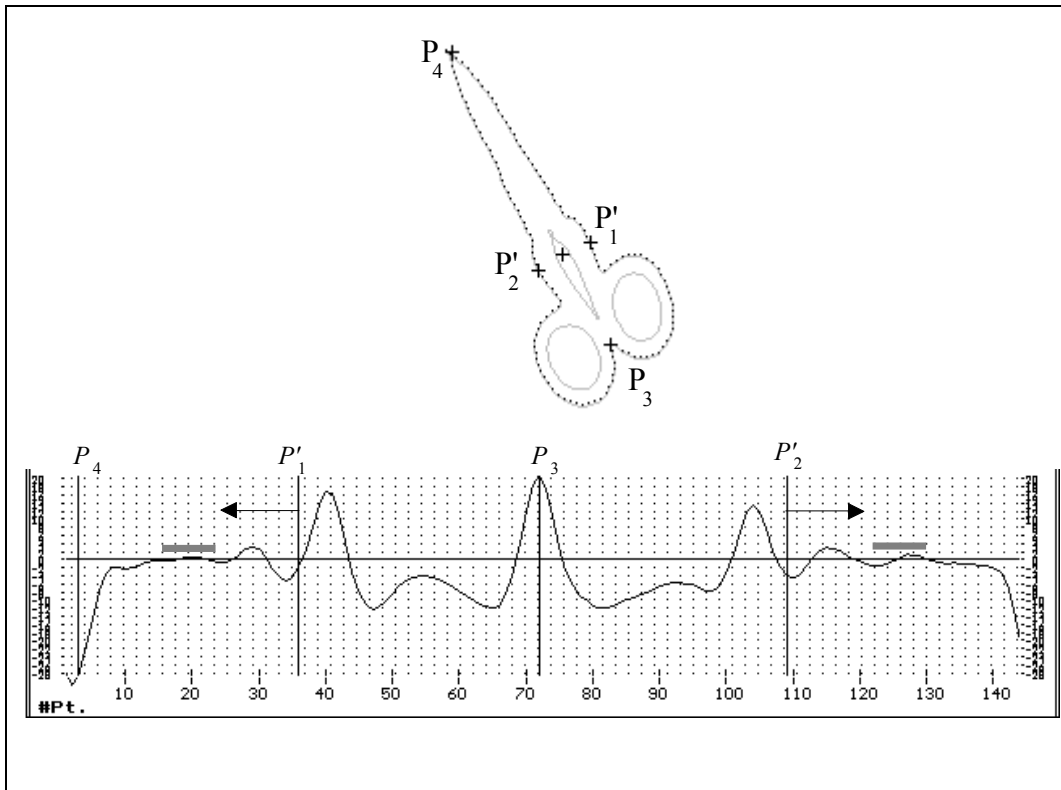


Figura 5.11. Tijeras. Representación (curvatura , punto) del vector de torsiones asociado al contorno de la imagen. Las flechas marcan las posibles direcciones de mejora en la disminución de la curvatura. Los dos rectángulos indican las regiones válidas para el agarre, en cuanto a curvatura se refiere.

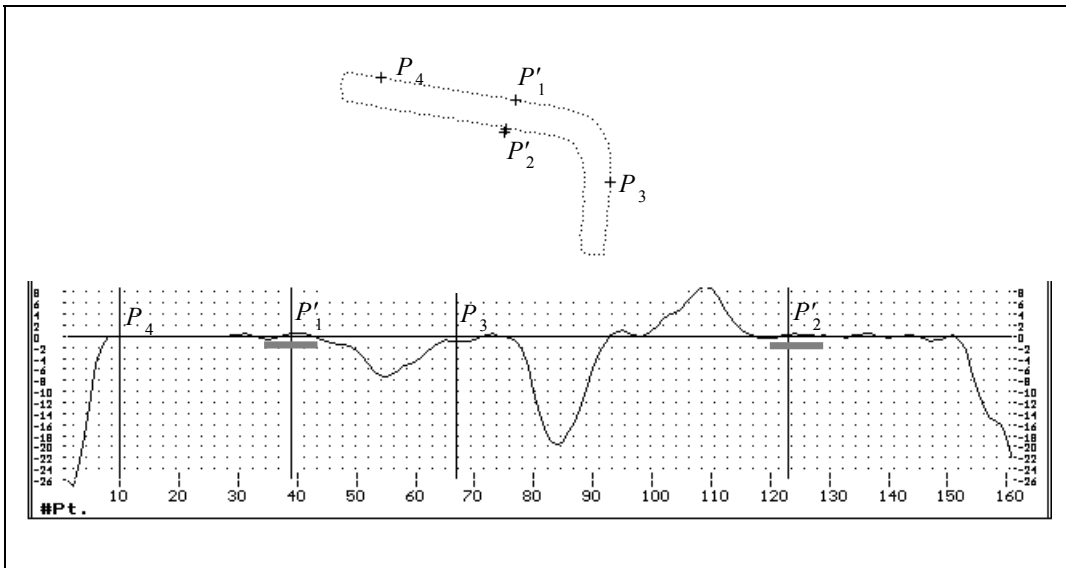


Figura 5.12. Llave allen. Representación (curvatura , punto) del vector de torsiones asociado al contorno de la imagen. En este caso no ha lugar la búsqueda de mejora en la curvatura. Los dos rectángulos indican las regiones necesarias para el agarre.

donde N = número de puntos total pertenecientes al contorno. En la tabla 5.1 se observan algunos resultados orientativos. Puntualicemos que para un objeto ideal, de simetría especular perfecta su correspondiente deficiencia simétrica global, Φ , sería nula ($\Phi = 0$). Ver el apartado 5.4.1 sobre resultados de la evaluación de la simetría al respecto. Se ha determinado, empíricamente, que a aquellos objetos que poseen $\Phi \leq 3$ (en valor medio) se les puede dar un tratamiento claro de simetría especular respecto de I_{min} , existiendo una frontera delimitada por: $3 \leq \Phi \leq 5$, donde se desvanece dicho tratamiento (e.g. la pinza), para pasar a la clasificación de no simétrico a partir de $\Phi \geq 5$ (e.g. la llave allen).

Imagen	$\Phi(\mu)$	$\Phi(\sigma)$
Alicates	0.87	0.44
Destornillador	1.58	0.82
Pinzas	4.42	0.61
Llave allen	6.38	0.58

Tabla 5.1. Deficiencia simétrica global (Φ) calculada para diferentes imágenes, en orden de simetría decreciente. Se da la media (μ) y la desviación típica (σ) asociada.

5.3.2.3. Umbral de curvatura.

Al introducir el concepto de curvatura y, sobre todo, al ver su análisis a partir de la información del vector de torsión, queda patente la necesidad de asignar una tolerancia a la medida de la misma. Esta tolerancia recibe el nombre de umbral de curvatura, α , fijado de modo empírico como $|\alpha| = 20^\circ$.

Observación:

En lo que sigue vamos a hacer uso de la siguiente notación: definiremos el vector \mathbf{P} , como aquél formado por dos componentes, P_1 y P_2 , de tal forma que: $\mathbf{P} = (P_1, P_2)$. Y, a partir de aquí, definiremos la variable booleana, nivel de confianza, $a_j(\mathbf{P})$, asociada a un determinado umbral: j , tal que si $a_j(\mathbf{P}) = 1$, indica que se satisface la condición impuesta por el umbral j , y en otro caso ($a_j(\mathbf{P}) = 0$) que no se satisface. (i.e. $j=1$, para el umbral de curvatura, etc.)

Resumiendo, con lo visto hasta ahora para la curvatura, podemos construir el siguiente algoritmo (**MESU_1**):

Mientras ($\alpha_i > \alpha$) **Hacer** /* existe alta curvatura */
 trasladar P_i en la dirección que corrija más rápido la condición anterior, teniendo en cuenta el tipo de simetría asociado.
Si se encuentra una región del contorno, centrada en P_i , tal que ($\alpha_i < \alpha$)
 Entonces devuelve P_i
 Sino devuelve "error".

Siendo la variable α_i , la curvatura asociada al punto P_i y α , el denominado umbral de curvatura.

El mecanismo corrector consistirá en ir trasladando el intervalo inicial en la dirección que disminuya la curvatura, hasta conseguir un ángulo $\alpha_i \leq \alpha$ en todos sus puntos. Este mecanismo tiene su correspondencia con el algoritmo expuesto en el punto 5.3.2.6, en el procedimiento *Calcula_* α_i . Devolviendo el resultado en la variable booleana $a_1(\mathbf{P})$. Si lo encontramos $a_1(\mathbf{P}) = 1$ y sigue adelante con los restantes pasos del algoritmo, en otro caso, $a_1(\mathbf{P}) = 0$, que establece una condición de parada. En este último caso el algoritmo se detendría, informando de la imposibilidad de encontrar, por tanto, un agarre que garantice estabilidad.

Observación:

Es muy importante, al considerar el problema del análisis de la curvatura, la denominada región de soporte [Teh & Chin, 89], que puede ser determinada de forma automática, lo cual permite, entre otras cosas, hacer que la evaluación de aquella no dependa de parámetro de entrada alguno. No obstante, en nuestro caso, la geometría de la pinza es conocida

a priori siempre, de manera que el radio de la región de soporte, k , no es calculado para cada punto sino que es fijado en función de dicha geometría (radio correspondiente a la anchura de los dedos de la pinza), centrado en los puntos en cuestión. Así pues, la región considerada corresponde a la zona sobre la cuál incidirán los dedos de la pinza.

5.3.2.4. Umbral angular.

Si se supera con éxito la etapa anterior, significa que la zona del contorno, en un entorno de radio k , con centro en cada punto de agarre, cumple con las condiciones de suavidad que garantizan la adaptación de la pinza de dedos plano-paralelos, siempre y cuando se cumpla, además, el paralelismo que ha de existir entre dichos intervalos opuestos en el contorno exterior del objeto dado. Es decir, hemos de evaluar cuáles son los ángulos entre la línea de agarre y las normales en dichos puntos del contorno. El umbral angular, β , a partir de consideraciones empíricas, queda fijado en 10° . El mecanismo corrector consistirá en ir trasladando los puntos, P_i , hasta encontrar:

$$\beta_1 = P_1P_2 \wedge N_{P_1} \leq \beta; \text{ y } \beta_2 = P_1P_2 \wedge N_{P_2} \leq \beta$$

Siendo la variable β_i , el ángulo entre la línea de agarre P_1P_2 y la normal en el punto P_i ; N_{P_i} y β el umbral angular.

Resumiendo, podemos construir el siguiente algoritmo (**MESU_2**):

Mientras ($\beta_i > \beta$) **Hacer**

trasladar P_i en la dirección que corrija más rápido la condición anterior, teniendo en cuenta el tipo de simetría asociado.

Si se encuentra una región del contorno, centrada en P_i , tal que ($\beta_i < \beta$)

Entonces devuelve P_i

Sino devuelve "error".

Este mecanismo tiene su correspondencia con el algoritmo expuesto en el punto 5.3.2.6, en el procedimiento *Calcula_β_i*. Siguiendo la notación introducida en el apartado anterior, se devuelve el resultado en la variable booleana $a_2(\mathbf{P})$. Si la respuesta es favorable implica $a_2(\mathbf{P}) = 1$ y sigue adelante con los restantes pasos del algoritmo, en otro caso, $a_2(\mathbf{P}) = 0$, que establece una condición de parada. En este último caso el algoritmo se detendría, informando de la imposibilidad de encontrar, por tanto, un agarre que garantice estabilidad.

Observación:

El mecanismo corrector anterior puede modificar la posición en el contorno de los puntos P_i , iniciales, provenientes de la condición impuesta para la curvatura, por lo que si se comprueba que ocurre dicha

modificación habremos de volver de nuevo al paso anterior, "umbral de curvatura" (MESU_1), y comprobar que se sigue cumpliendo la constancia de la normal en el nuevo intervalo (curvatura suave).

5.3.2.5. Umbral de distancia.

Siguiendo con las condiciones de estabilidad a imponer, sólo nos resta analizar la distancia crítica, es decir: cuál es la distancia máxima permitida entre la línea de agarre y el centroide, tal que se preserve la estabilidad.

Se ha obtenido, empíricamente, el siguiente umbral de distancia:

$$\gamma = 0.03 \cdot [2I_a]$$

donde I_a , representa el semieje mayor de la elipse de mejor ajuste, que representa un segmento de recta en la dirección de I_{min} . Es decir, el 3 % de la longitud asociada con el eje mayor de la elipse de mejor ajuste (véase el capítulo IV). Ver tabla 5.2.

Resumiendo, podemos construir el siguiente algoritmo (MESU_3):

Si $d > \gamma$ Entonces $a_3(\mathbf{P}) = 0$. /* Condición de Parada */
En otro caso, $a_3(\mathbf{P}) = 1$. /* Ejecutar paso siguiente */

siendo la variable d , la distancia de la línea de agarre, P_1P_2 , al centroide; $a_3(\mathbf{P})$, su nivel de confianza asociado; y γ el umbral de distancia definido anteriormente.

En la tabla 5.2 se muestra el resultado obtenido para dicho umbral, γ , dando su valor medio, μ , y su desviación típica, σ , sobre un conjunto de cuatro muestras procesadas de cada objeto, con los mismos parámetros de cámara pero variando su localización (posición y orientación) en la escena. Se observa una clara dependencia del umbral γ con la elongación del objeto y su tamaño o número de puntos asociado al contorno. En el apéndice C se muestran cada una de las imágenes de la tabla.

Recordemos que este umbral no tiene sentido plantearlo para objetos "ligeros", en relación a la razón de presión ejercida por la pinza contra el peso del objeto, cuya resultante se sitúa en el centroide. Si el objeto fuera "ligero" el agarre no se vería afectado por la mayor o menor distancia de la línea de agarre al centroide. Lo cual simplificaría el modelo de agarre propuesto haciendo innecesario incluir en la caracterización del agarre las restricciones impuestas por el umbral γ .

Imagen	$\gamma (\mu)$	$\gamma (\sigma)$
Tuerca	1.35	0.01
Tornillo	2.77	0.20
Llave allen	3.32	0.24
Pinzas	3.44	0.22
Tijeras-1	4.48	0.36
Tijeras-2	6.65	0.19
Calibre-A	6.89	0.15
Calibre-B	7.00	0.20

Tabla 5.2. Umbral de distancia (γ) para diferentes imágenes. Se observa la correspondencia lógica entre el valor alcanzado y la mayor o menor elongación del objeto.

5.3.2 6. Estrategia adoptada.

Como se ha visto a partir de la introducción al concepto de simetría y su potencial uso en la discriminación de las direcciones de mejora en la curvatura (apartado 5.3.2.2), el primer paso a dar en el mecanismo evaluador/corrector propuesto (MESU) será la determinación de la misma (véase la figura 5.13).

Se tienen en cuenta las tres posibilidades de simetría vistas anteriormente, cada una de las cuales dará paso a una estrategia ligeramente diferente en el modo de trasladar los intervalos centrados en los posibles puntos de agarre, buscando regiones de los contornos que cumplan las condiciones impuestas por los umbrales definidos anteriormente. Esto ya se ha expuesto en el apartado (5.3.2.2), en relación con la búsqueda de regiones de curvatura suave para los tres tipos de simetría planteados. Ahora se ha de observar el problema en conjunto y tener en cuenta qué efectos puede tener trasladar un intervalo respecto del opuesto, en relación a la curvatura, ángulo de la línea de agarre con las normales respectivas y distancia de la misma al centroide, y de qué manera ayuda el conocimiento de la simetría en todo ello. Veamos los tres casos planteados:

Caso_1 (Simetría = 0). No se podrá hacer movimientos a priori porque no se sabe cómo afecta el movimiento de un intervalo sobre el opuesto en relación a la morfología global del objeto. Así pues, este caso obliga a evaluar el efecto de mover un intervalo sobre su opuesto, cada vez que se requiera dicho movimiento. En principio, a partir de los candidatos a puntos de agarre entregados por OCPA, (P'_1, P'_2) existen cuatro posibilidades en cuanto al efecto de evaluar la curvatura se refiere:

Curvatura_Correcta	$(\alpha_1 < \alpha \wedge \alpha_2 < \alpha) \Rightarrow$	No mover intervalos centrados en (P'_1, P'_2) .
Curvatura_Incorrecta	$(\alpha_1 < \alpha \wedge \alpha_2 > \alpha) \Rightarrow$	Mover intervalo centrado en (P'_2) .
	$(\alpha_1 > \alpha \wedge \alpha_2 < \alpha) \Rightarrow$	Mover intervalo centrado en (P'_1) .
	$(\alpha_1 > \alpha \wedge \alpha_2 > \alpha) \Rightarrow$	Mover intervalos centrados en (P'_1, P'_2) .

La primera posibilidad (Curvatura_Correcta) es trivial. En la otra (Curvatura_Incorrecta) habrá de evaluarse, antes de mover, que dirección representa un valor mayor de mejora de curvatura. Lo acertado o no de los posibles movimientos realizados será sopesado al evaluar la siguiente condición (umbral angular) en cada uno de los puntos.

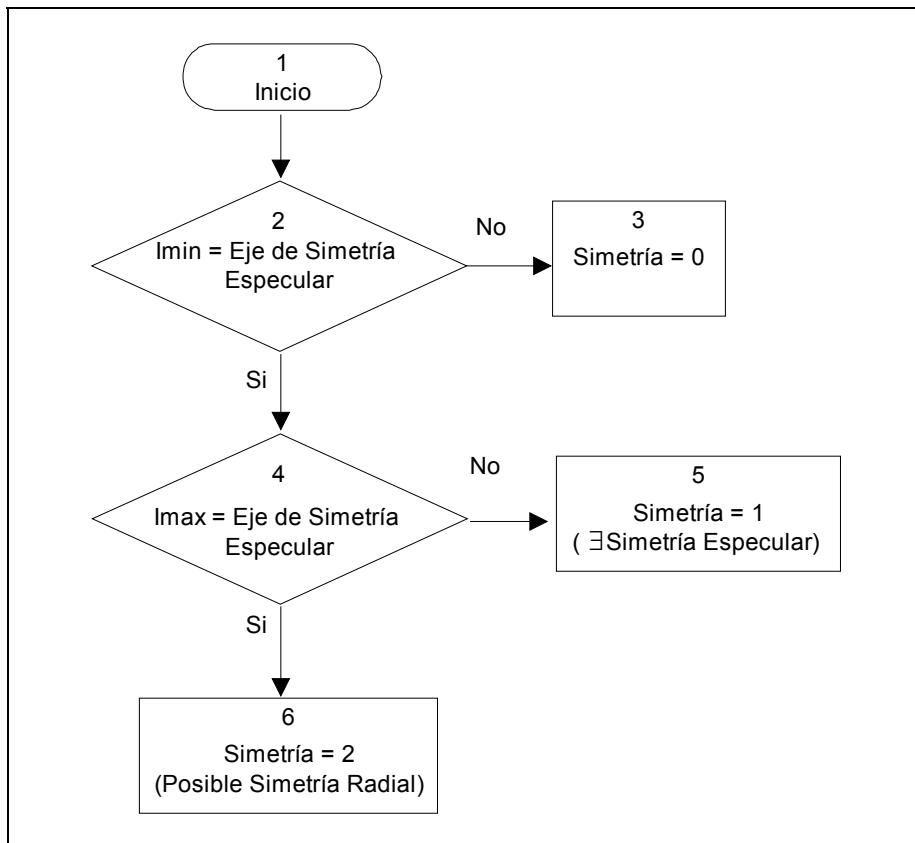


Figura 5.13. Diagrama de flujo para el primer paso de MESU. Se plantean tres posibilidades en cuanto a simetría se refiere. De tal forma que según sea simetría = 0, 1 ó 2, se dispararán a continuación estrategias distintas para determinar el agarre final.

Caso_2 (Simetría = 1). Este caso, como se observó en el apartado 5.3.2.2, tiene la ventaja de predecir el efecto del movimiento de un intervalo sobre el opuesto, en el sentido de que, elegida una dirección de mejora de curvatura para un intervalo, la dirección de mejora del opuesto será en sentido contrario (a partir del punto inicial del contorno PI). Las posibilidades que se plantean son las mismas que las del caso anterior pero teniendo en cuenta lo dicho para el movimiento simultáneo de los dos intervalos.

Caso_3 (Simetría = 2). Como en el caso anterior aquí también se puede predecir las direcciones de mejora de curvatura, cuando existe la necesidad de mover algún intervalo. Como se vió en el apartado 5.3.2.2, el sentido impuesto por el movimiento de un intervalo obliga al opuesto a seguirlo (sentido horario a partir del punto inicial del contorno PI).

Como se observa de los casos 2 y 3 planteados, el conocimiento de la simetría existente facilita las direcciones de mejora convenientes en cada caso, adelantándose así al cumplimiento de la siguiente condición: la impuesta por el umbral angular. Los tres casos contemplados se implementan en el procedimiento "Calcula_ α_i " que hace un tratamiento diferenciado para cada una de las tres posibilidades existentes a partir del conocimiento del tipo de simetría, siguiendo lo dicho en el apartado 5.3.2.1, asociada al contorno. En los tres casos contemplados existe un mecanismo de parada en las sucesivas iteraciones del algoritmo para trasladar el intervalo en busca de condiciones favorables de curvatura, el cual se fija a partir del umbral de distancia, γ , de la línea de agarre al centroide, calculado *a priori* para cada contorno.

El paso siguiente, la evaluación/corrección angular, se implementa en el procedimiento "Calcula_ β_i ". Aquí es extrapolable todo lo expuesto para la curvatura, haciendo uso como allí del conocimiento de la simetría.

Obsérvese que el procedimiento "Calcula_ β_i ", puede conllevar la traslación de alguno de los puntos P_1 , P_2 , o de ambos; es decir, puede modificar el vector $\mathbf{P} = (P_1, P_2)$, entregado por el procedimiento anterior: "Calcula_ α_i ". En ese caso ya no se garantizaría la constancia de la normal en el intervalo correspondiente, obligando, por tanto, a recalcularla mediante "Calcula_ α_i ".

En la figura 5.14, se visualizan los efectos correctores en relación al umbral β , así como parte de la notación empleada en esta estrategia.

Si se superan cualquiera de estos tres umbrales (α , β , o γ), no podrá garantizarse la estabilidad del agarre. Necesitaremos aplicar, por tanto, una estrategia correctora para intentar salvar la situación. En resumen, se trata de un mecanismo evaluador/corrector del agarre alcanzado previamente, el cuál se basa en la definición y uso de dichos umbrales.

Resumiendo, a lo largo de los puntos anteriores se han expuesto todas las características a tener en cuenta para garantizar la preadaptación de la pinza y la estabilidad en el agarre, a continuación las integramos en el siguiente algoritmo:

```

Calcula  $\alpha_i$ 
Si  $a_1(\mathbf{P}) = 0$  Entonces Estable(P) 0 /* falla MESU_1 */
Sino
  Calcula  $\beta_i$ 
  Si  $a_2(\mathbf{P}) = 0$  Entonces Estable(P) 0 /* falla MESU_2 */
  Sino
    Si P ha cambiado Entonces
      Calcula  $\alpha_i$  /* evalúa pero no corrige la posición */
      Si  $a_1(\mathbf{P}) = 0$  Entonces Estable(P) 0 /* falla MESU_1 */
      Sino
        Estable(P) 1 /* Agarre Estable! */
    FinSi
  Sino /* P no ha cambiado */
    Estable(P) 1 /* Agarre Estable! */
  FinSino
FinSino

```

Nótese que no aparece explícitamente ningún procedimiento del tipo "Calcula γ_i ", para comprobar si falla o no MESU_3. La razón es obvia, dicho procedimiento está implícito, en cada uno de los otros dos ("Calcula α_i " y "Calcula β_i "), operando a modo de mecanismo de parada, tal y como se pondrá de manifiesto en los ejemplos posteriores del apartado 5.4 de resultados.

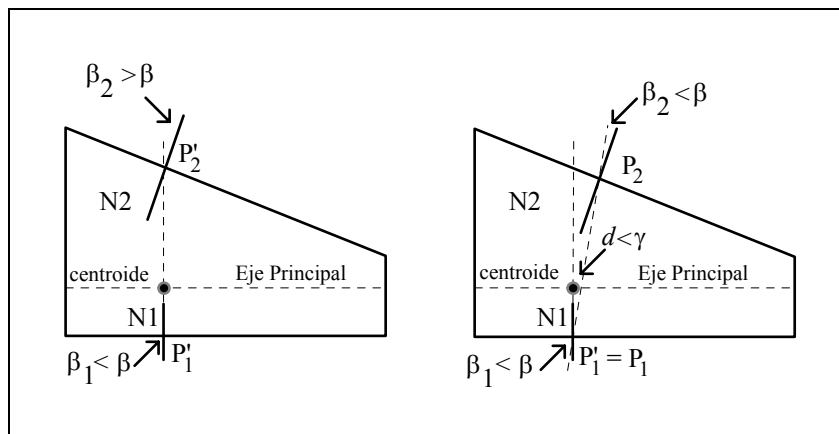


Figura 5.14. Interpretación geométrica del algoritmo descrito en los pasos MESU_2 y MESU_3. Candidatos iniciales (izquierda) y resultado de aplicar el mecanismo evaluador/corrector angular (derecha).

La actuación del mecanismo supervisor podemos resumirla, definiendo la variable booleana: $Estable(\mathbf{P})$, como el productorio siguiente:

$$Estable(\vec{P}) = \prod_{j=1}^3 a_j(\vec{P}) \quad (5.1)$$

donde,

$$\begin{aligned} a_1(\mathbf{P}) &= \text{grado de curvatura}(P_1). & /* \text{ Definido en MESU_1} \\ */ & \\ a_2(\mathbf{P}) &= \text{grado angular}(P_1P_2, NP_i). & /* \text{ Definido en MESU_2} \\ */ & \\ a_3(\mathbf{P}) &= \text{grado de distancia } (P_1P_2, \text{centroide}). & /* \text{ Definido en MESU_3} \\ */ & \end{aligned}$$

Tanto $a_1(\mathbf{P})$, como $a_2(\mathbf{P})$ y $a_3(\mathbf{P})$, hemos visto que toman únicamente dos valores posibles: $\{0,1\}$. Luego,

$$Estable(\vec{P}) = \begin{cases} 0 & \text{sii } (a_1(\vec{P}) \vee a_2(\vec{P}) \vee a_3(\vec{P})) = 0 \\ 1 & \text{sii } a_1(\vec{P}) = a_2(\vec{P}) = a_3(\vec{P}) = 1 \end{cases}$$

Esta variable tiene como misión fundamental, una vez satisfechas las condiciones de seguridad (apartado siguiente), la de comunicar rápidamente al manipulador si las coordenadas de los puntos encontrados son útiles o no para proceder al posicionamiento de la pinza, mediante el mecanismo de realimentación visual expuesto en el capítulo VI.

Un trabajo reciente de la Universidad de Rochester [Fuentes et al., 94] define un productorio, del estilo del seguido aquí para el mecanismo supervisor (5.1), pero tal que aplicando una serie de restricciones a priori, consigue expresiones analíticas para cada uno de los componentes de dicho productorio. De ésta forma traslada el problema de la planificación del agarre a un problema típico de optimización, empleando algoritmos genéticos para maximizar la función:

$$Q(\vec{s}) = \prod_{j=1}^5 g_j(\vec{s}) \quad ; \text{ siendo } \vec{s} = \langle s_0, s_1, s_2, s_3 \rangle$$

donde el vector s posee cuatro dimensiones relativas a cada uno de los cuatro contactos posibles, ya que se trabaja con una mano antropomorfa (Utah/MIT hand) de cuatro dedos, y las cinco funciones analíticas g se utilizan para caracterizar el modelo elegido de agarre con rozamiento.

5.3.3. Condiciones de seguridad.

Básicamente habrán de tenerse en cuenta dos propiedades:

- 1ª) La distancia entre los dos puntos de agarre no exceda la apertura máxima de la pinza.
- 2ª) Los dedos de la pinza no colisionen con el objeto.

La primera condición, "que la distancia entre P_1 y P_2 no exceda la máxima apertura de la pinza", se aplicará justo al salir del algoritmo MESU, visto anteriormente, cuando ocurra $Estable(\mathbf{P}) = 1$. La única precaución a tener en cuenta radica en calibrar previamente la distancia en mm de la apertura máxima de la pinza de trabajo traduciéndola a píxeles justo sobre el plano de la escena, donde tiene lugar siempre el agarre del objeto. Para ello se calcula la razón de aspecto en dichas condiciones (conocidos los parámetros de cámara asociados).

Si el cálculo anterior resulta satisfactorio se aplicará la segunda condición: "que los dedos de la pinza no colisionen con el objeto en su trayectoria de agarre". La idea es proyectar sobre la imagen el espacio utilizado por los dedos desde su apertura inicial (máxima) hasta que alcanzan la configuración final, y entonces comprobar que no existe intersección con el objeto (véase la figura 5.15).

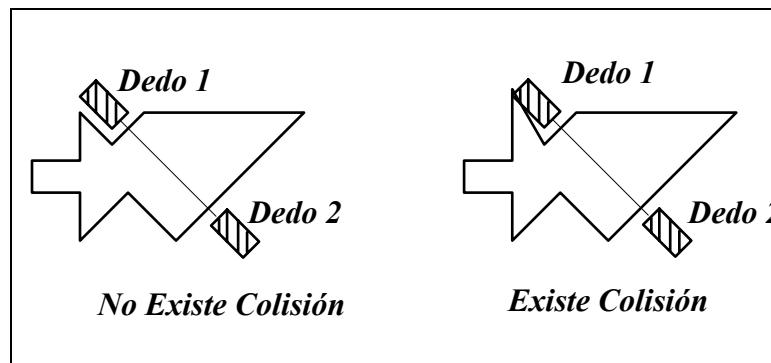


Figura 5.15. Interpretación geométrica de la condición de no colisión en el agarre.

Una idea simple pero efectiva para dicha condición de "no colisión", será utilizar un algoritmo adaptado de Sedgewick [Sedgewick, 88] ("Función booleana. $Intersect(l_1, l_2: lines)$ "). Se extiende la línea de agarre, P_1P_2 , por ambos extremos, una distancia que garantice el agarre efectivo. Si no existe intersección entre $P_1 + d$, P_1 y el resto del polígono, se hace lo mismo para dos líneas paralelas a $P_1 + d$, P_1 , por la izquierda y por la derecha, situadas en los

extremos del intervalo de radio k , centrado en P_1 , lo anterior se repite análogamente para el punto P_2 .

5.4. Resultados

5.4.1 Evaluación de la simetría

Los resultados mostrados anteriormente en relación a la deficiencia simétrica, Φ , en la tabla 5.1, representan una muestra extraída del experimento realizado sobre un total de once objetos. Se han digitalizado todos ellos con los mismos parámetros de cámara pero cambiando su localización (posición y orientación) en la escena un total de cuatro veces por objeto, para poder obtener la media (μ) y desviación típica (σ) asociada a cada uno de ellos. Los resultados obtenidos se muestran en la tabla 5.3. Como ya se comentó en el apartado 5.3.2.2 en relación con la tabla 5.1 (extraída de esta), se observa que la cuantificación de la simetría a partir de Φ puede servir de base para realizar una primera clasificación en la evaluación de la simetría. Si $\Phi \leq 3$ se evidencia la necesidad de un tratamiento de simetría especular respecto de I_{min} . Por contra, si $\Phi \geq 5$ no se podrá dar dicho tratamiento. Además, en el rango entre ambos, $3 < \Phi < 5$, se pierde gradualmente dicho tratamiento. Nosotros para simplificar utilizaremos el valor de corte dado por $\Phi = 5$, para valores menores se dará el tratamiento de simetría especular y para valores mayores no. Como información complementaria en el apéndice C se visualiza la morfología de cada uno de los objetos mostrados en la tabla 5.3.

Imagen	$\Phi(\mu)$	$\Phi(\sigma)$
Alicates	0.87	0.44
Destornillador	1.58	0.82
Tijeras-1	1.59	0.92
Tornillo	1.69	0.62
Tuerca	1.86	0.74
Tenazas	2.68	0.54
Calibre-B	3.08	0.38
Tijeras-2	3.20	0.46
Calibre-A	5.09	0.30
Pinzas	4.42	0.61
Llave allen	6.38	0.58

Tabla 5.3. Deficiencia simétrica global (Φ) calculada para diferentes imágenes, en orden de simetría decreciente. Se da la media (μ) y la desviación típica (σ) asociada a cada una de ellas.

5.4.2 Ejemplos con solución

Este es el caso en el que el mecanismo supervisor (MESU) propuesto, encuentra un agarre estable. Dividiremos el universo de trabajo en relación a la simetría asociada a cada una de las imágenes.

Caso 1: Existe simetría radial.

Ejemplo: tuerca (simetría D_6), ver la figura 5.16. De acuerdo con lo explicado en el apartado 5.3.2.2, para el caso de simetría radial, aquí se sigue un movimiento de traslaciones predeterminado, en este caso "sentido horario". La condición de parada viene impuesta simultáneamente por la satisfacción de las condiciones de curvatura suave y mínima distancia de los puntos (P_1 y P_2) al centroide.

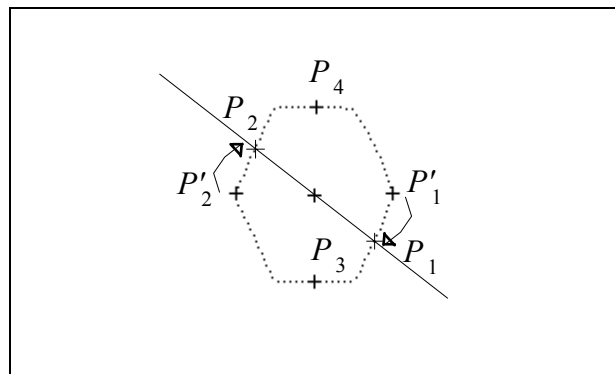


Figura 5.16. Tuerca: determinación de los puntos de agarre en el caso estudiado previamente en el apartado 5.3.2.2.

Caso 2: Existe simetría especular respecto de la dirección del eje principal (I_{min}).

Ejemplos:

2.1. Destornillador. Aquí el Mecanismo Supervisor (MESU) evalúa satisfactoriamente los puntos de agarre entregados por el paso OCPA, visto anteriormente. El efecto de MESU sobre este objeto puede apreciarse en la figura 5.17. En este caso MESU se ha limitado a evaluar y verificar el cumplimiento satisfactorio de las tres condiciones impuestas por cada uno de los umbrales, no siendo necesario aplicar traslación alguna a los puntos iniciales (P'_1 y P'_2).

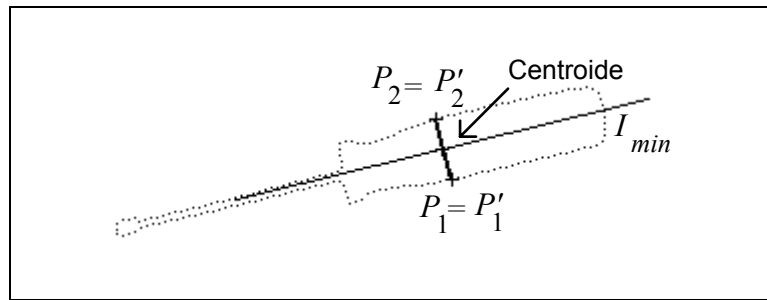


Figura 5.17. Destornillador. Determinación de los puntos de agarre en el caso más simple posible.

2.2. Llave inglesa. Aquí, como en el caso anterior, el Mecanismo Supervisor (MESU) evalúa satisfactoriamente los puntos de agarre entregados por el paso OCPA. Un seguimiento del comportamiento de MESU_1 sobre este objeto puede seguirse en la figura 5.18, donde se aprecian las regiones del contorno correspondientes a los intervalos de radio k centrados en los puntos (P'_1, P'_2), cumpliendo las condiciones impuestas para la curvatura y, además, para los otros dos umbrales definidos. El efecto corrector de MESU_2 puede observarse en la imagen superior derecha de la misma.

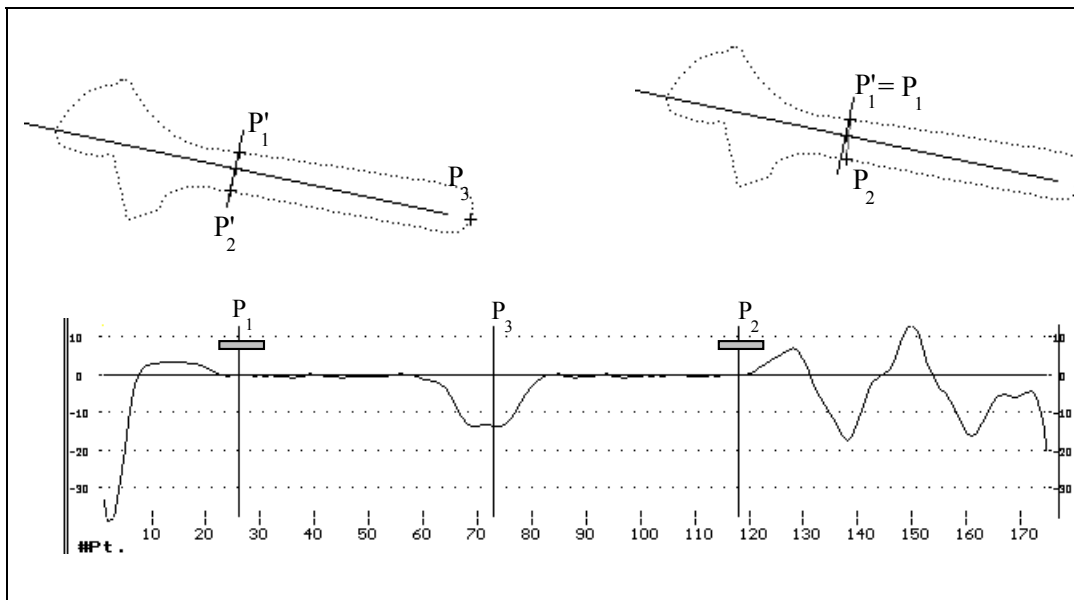


Figura 5.18. Llave inglesa. Se observa a partir del gráfico de torsiones, para la imagen superior izquierda, que los puntos candidatos cumplen las condiciones de curvatura suave (MESU_1). La imagen de la derecha corresponde a la corrección final por MESU_2.

Caso 3: No existe simetría.

Ejemplos:

3.1. Calibre. Aquí el Mecanismo Supervisor (MESU) evalúa y corrige los puntos de agarre entregados por el paso OCPA. Inicialmente, se corrige el intervalo centrado en P'_2 , para satisfacer los requerimientos de curvatura (MESU_1), lo cual obliga a corregir a su vez el intervalo centrado en P'_1 , para satisfacer la condición impuesta en MESU_2. Finalmente, se encuentra que se cumplen todas las condiciones y concluye que el agarre es estable (ver figura 5.19). En la tabla 5.4 se observa la evolución de los algoritmos a partir de los valores que se extraen sucesivamente para los puntos, ángulos (β_1, β_2) y distancia, d , al centroide, respectivamente.

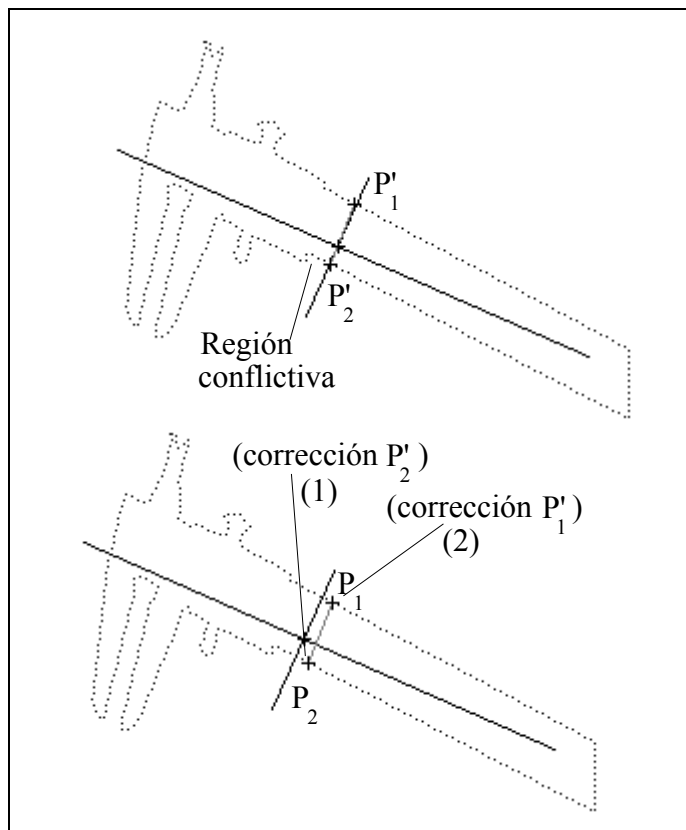


Figura 5.19. Calibre. Candidatos a puntos de agarre (P'_1, P'_2), entregados por el paso OCPA, y los corregidos por el paso MESU (P_1, P_2). La corrección en curvatura lleva P'_2 a P_2 (1), obligando a trasladar P'_1 a P_1 (2), para compensar el efecto angular introducido (MESU_2).

	[1ª] OCPA	[2ª] MESU_1	[3ª] MESU_2	[4ª] MESU_2
P_1	(118,84)	(118,84)	(112,81)	(124,87)
P_2	(106,114)	(112,117)	(112,117)	(112,117)
β_1	—	21°	32°	6° < β
β_2	—	15°	25°	3° < β
d	0	—	—	6.5 < γ

Tabla 5.4. Calibre. Datos obtenidos en la evolución del algoritmo en la búsqueda de las condiciones de estabilidad del agarre, a partir de los puntos entregados por OCPA. La columna [3ª] muestra un intento infructuoso de compensar la condición impuesta por el umbral angular, siendo en la [4ª], mediante la dirección opuesta, donde se alcanza dicha condición. Finalmente, comprobado el umbral de distancia ($\gamma = 8$) se garantiza un agarre final estable.

3.2. Llave allen. Aquí el Mecanismo Supervisor (MESU) evalúa y corrige los puntos de agarre entregados por el paso OCPA. Finalmente, encuentra que se cumplen todas las condiciones y concluye que el agarre es estable (ver figura 5.20). Como se mostró anteriormente (figura 5.12), los intervalos centrados en los candidatos a puntos de agarre cumplen las condiciones de curvatura suave impuestas por MESU_1. A continuación se evalúa/corrigue P'_1 a partir de MESU_2, dejando P'_2 invariante a dicha traslación. Una vez satisfecho MESU_2, se comprueba favorablemente MESU_3, observando un agarre final estable. Los resultados de la evolución, como en el caso anterior, se representan en la tabla 5.5.

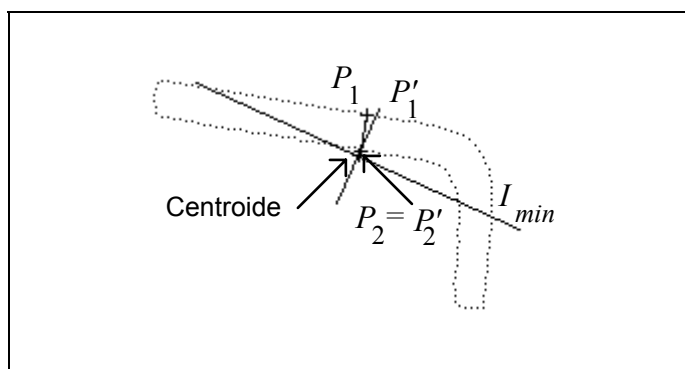


Figura 5.20. Llave allen. Candidatos a puntos de agarre (P'_1 , P'_2), entregados por el paso OCPA, y los corregidos por el paso MESU (P_1 , P_2).

	[1ª] OCPA	[2ª] MESU_1	[3ª] MESU_2
P_1	(156,75)	(156,75)	(150,74)
P_2	(147,92)	(147,92)	(147,92)
β_1	—	20°	0° < β
β_2	—	18°	0° < β
d	0	—	1 < γ

Tabla 5.5. Llave allen. Datos obtenidos en la evolución del algoritmo en la búsqueda de las condiciones de estabilidad del agarre, a partir de los puntos entregados por OCPA. La columna [3ª] muestra que la traslación se ha efectuado en la dirección correcta, corrigiendo la condición angular impuesta por el umbral (β). Finalmente, comprobado el umbral de distancia ($\gamma = 5$) se garantiza un agarre final estable.

5.4.3 Ejemplos sin solución

Este es el caso en el que el mecanismo supervisor (MESU) propuesto no encuentra un agarre estable porque no se cumple alguna de las tres condiciones impuestas por los umbrales. En realidad, visto como funcionan los algoritmos, la tercera condición queda subsumida, como mecanismo de parada, en las sucesivas iteraciones desarrolladas por las dos primeras, en la búsqueda de una región del contorno que supere las condiciones de curvatura y angular impuestas por MESU_1 y MESU_2, respectivamente. Así pues, mostraremos a continuación los dos casos de error posibles: *caso-1*, cuando falla MESU_1 (e.g. tijeras), y *caso-2*, cuando falla MESU_2 (e.g. pinzas o alicates).

Caso-1: Tijeras (ver figura 5.21). A partir de los puntos entregados por el paso OCPA, el Mecanismo Supervisor (MESU) evalúa, en primer lugar, las condiciones de curvatura. La figura 5.22 muestra porqué estos puntos (el efecto sobre P_2 es similar al de P_1) no cumplen las necesarias condiciones de suavidad requeridas para alcanzar una preadaptación satisfactoria de la pinza de dedos plano-paralelos. Se observa en esta figura la evolución de las normales en los puntos que delimitan el intervalo centrado en P_1 dejando patente su falta de constancia, como requiere MESU_1 para garantizar la suavidad de la curvatura. Esta forma de visualizar el comportamiento de la curvatura es alternativa (o complementaria) de la vista anteriormente mediante la curva del vector de torsiones.

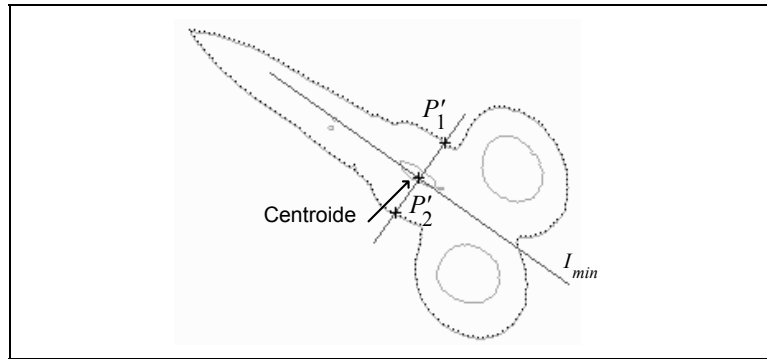


Figura 5.21. Tijeras: candidatos a puntos de agarre a partir del algoritmo OCPA.

Estado Inicial		Estado Final	
Coordenadas	Curvatura	Coordenadas	Curvatura
P'_1 (171,115)	46°	P'_1 (166,111)	2°
P'_{1-k} (160,107)	13°	P'_{1-k} (154,104)	25°
P'_{1+k} (182,113)	50°	P'_{1+k} (177,118)	86°

Tabla 5.6. Tijeras. Evolución cuantitativa de MESU_1 (curvatura en grados) para P'_1 .

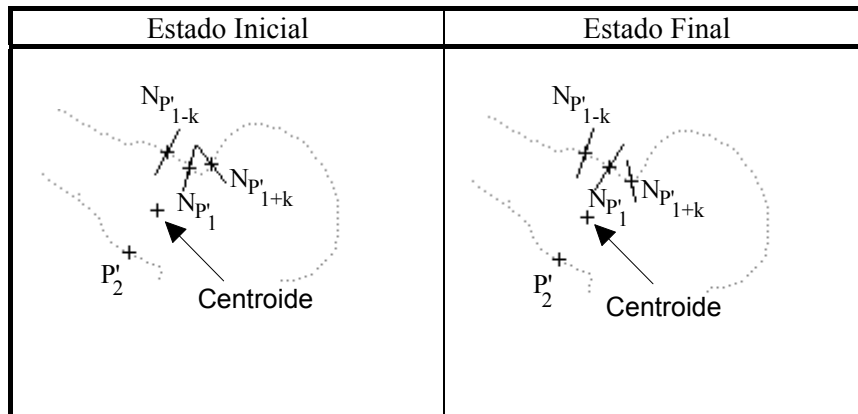


Figura 5.22. Tijeras. Evolución de MESU_1. Se visualizan las normales en los puntos extremos del intervalo de radio k , alrededor de P'_1 .

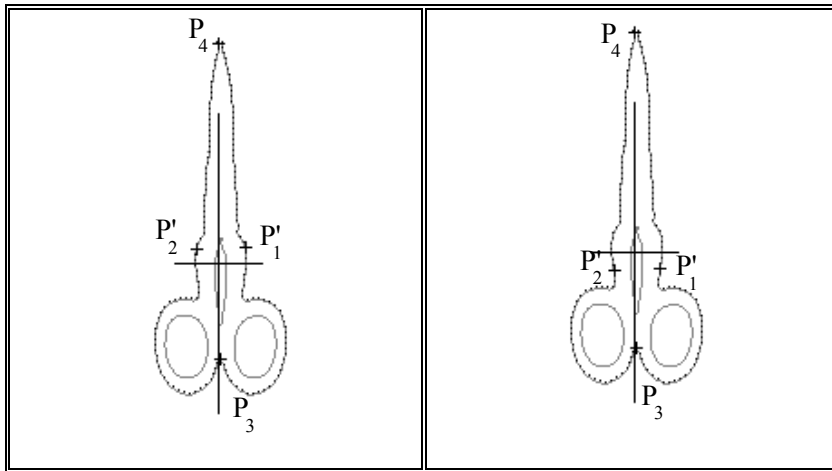


Figura 5.23. Tijeras. Evolución del mecanismo evaluador/corrector de curvatura, MESU_1. Se intentan las dos direcciones posibles: izquierda y derecha. Se observa el máximo alcanzado en las traslaciones de P'_1 y P'_2 , antes de la "parada" del procedimiento iterativo que lo implementa, al sobrepasarse el umbral de distancia ($\gamma = 5.0$, en este caso).

En la figura 5.23 se muestra el efecto de la condición de parada sobre las tijeras. Dichas tijeras son distintas a las de la figura 5.21, y se encuentran en una localización (posición más orientación) distinta de la escena. Tengamos en cuenta que el algoritmo prueba todo el tiempo el alejamiento de la línea de agarre respecto del centroide, como se explicó anteriormente. Hay que puntualizar que el algoritmo, como se indicó en el apartado 5.3.2.2 (caso2: simetría especular), hace uso de la existencia de simetría especular y, por tanto, sigue una búsqueda predeterminada de las regiones de buen comportamiento angular, intentando en primer lugar la traslación de los puntos hacia el inicial, PI, (dir-B):

$$P_1 \mapsto P_1 - n \wedge P_2 \mapsto P_2 + n; \quad n = 1, 2, \dots$$

en dos iteraciones se alcanza la condición de parada: $d (=5.46) > \gamma (=5.0)$, intentando a continuación la traslación en el sentido opuesto (dir-A):

$$P_1 \mapsto P_1 + n \wedge P_2 \mapsto P_2 - n; \quad n = 1, 2, \dots$$

aquí se producen dos iteraciones más sin éxito, alcanzándose la condición de parada: $d (=6.5) > \gamma (=5.0)$, por lo que se concluye que no se garantiza la estabilidad del agarre en este caso. La evolución completa de MESU_1 para el caso representado en la figura 5.23 se muestra en la tabla 5.7.

		(dir-B)	(dir-B)'	(dir-A)	(dir-A)'
	OCPA	MESU_1	MESU_1	MESU_1	MESU_1
P_1	(160,138)	(160,135)	(160,132)	(160,141)	(159,144)
P_2	(132,139)	(132,136)	(132,133)	(133,142)	(133,145)
α_1	40°	34°	30°	26°	24°
α_2	38°	39°	34°	20°	22°
d	0.5	$2.46 < \gamma$	$5.46 > \gamma$	$3.5 < \gamma$	$6.5 > \gamma$

Tabla 5.7. Tijeras. MESU_1 comprueba que no se satisfacen las condiciones de curvatura (OCPA) y dispara el mecanismo evaluador/corrector, primero en la dirección hacia PI (figura 5.23-izquierda), y a continuación en sentido opuesto (figura 5.23-derecha). Obsérvese las dos condiciones de parada, una en cada sentido, indicadas por la condición $d > \gamma$ ($= 5.0$, en este caso).

Caso 2:

Ejemplos:

2.1. Pinzas. A partir de los puntos entregados por el paso OCPA, el Mecanismo Supervisor (MESU) evalúa en primer lugar las condiciones de curvatura (MESU_1). La figura 5.24, muestra que estos puntos (P_1 y P_2) sí cumplen las necesarias condiciones de suavidad en cuanto a curvatura se refiere, por tanto se satisface MESU_1. El problema radica en el paso MESU_2. Una evolución completa de la actuación de los algoritmos prediciendo la inestabilidad del agarre en este caso puede seguirse en la tabla 5.8. De nuevo el algoritmo, como en el caso anterior de MESU_1 sobre las tijeras, hará uso de la existencia de simetría especular y, por tanto, sigue el mismo criterio para la búsqueda predeterminada de las regiones de buen comportamiento angular, intentando en primer lugar la traslación de los puntos hacia el inicial, PI, (dir-B):

$$P_1 \mapsto P_1 - n \wedge P_2 \mapsto P_2 + n; \quad n = 1, 2, \dots$$

alcanzando en una sola iteración la condición de parada: d ($=3.8$) $> \gamma$ ($=3.0$) e intentando a continuación la traslación en el sentido opuesto (dir-A):

$$P_1 \mapsto P_1 + n \wedge P_2 \mapsto P_2 - n; \quad n = 1, 2, \dots$$

aquí se producen dos iteraciones sin éxito, alcanzándose la condición de parada: d ($=5.2$) $> \gamma$ ($=3.0$), por lo que se concluye, como en el caso anterior, la imposibilidad de garantizar la estabilidad del agarre propuesto. Una visualización del efecto de la inestabilidad de dicho agarre, en el caso de las pinzas por el robot manipulador real utilizado, puede observarse en el apartado

de resultados del capítulo VI (6.4), donde aparecen imágenes del manipulador en acción (véase la figura 6.8).

			(dir-B)	(dir-A)	(dir-A)'
	OCPA	MESU 1	MESU 2	MESU 2	MESU 2
P_1	(140,88)	(140,88)	(137,86)	(143,89)	(146,90)
P_2	(142,157)	(142,157)	(139,159)	(145,156)	(148,154)
β_1	—	24°	23°	24°	26°
β_2	—	24°	24°	24°	23°
d	0	0	$3.8 > \gamma$	$2.2 < \gamma$	$5.2 > \gamma$

Tabla 5.8. Pinzas. Aquí se satisfacen las condiciones de curvatura (MESU_1), pero no las angulares (MESU_2). Obsérvese las dos condiciones de parada, una en cada sentido, indicadas por la condición $d > \gamma$ ($= 3$, en este caso).

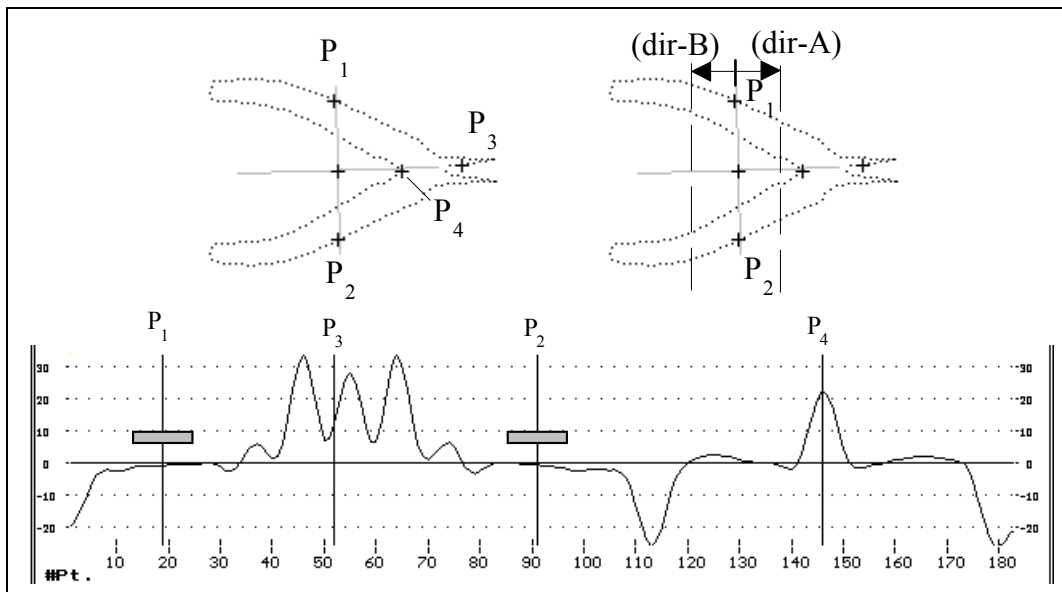


Figura 5.24. Pinzas. Aquí se observa el cumplimiento de las condiciones de curvatura (MESU_1), ver los dos "rectángulos" en las regiones de posible adaptación de los dedos de la pinza. En la parte superior derecha se visualiza el convenio seguido por (MESU_2) para evaluar/corregir las condiciones angulares (β_i).

2.2 Alicates. Aquí, como en el caso anterior, el Mecanismo Supervisor (MESU) no encontrará regiones que satisfagan el paso MESU_2,

prediciendo en consecuencia inestabilidad en el agarre. Un seguimiento del comportamiento de MESU_1 sobre este objeto puede seguirse en la figura 5.25, donde se aprecian las regiones del contorno correspondientes a los intervalos de radio k centrados en los puntos (P_1, P_2) , cumpliendo las condiciones impuestas para la curvatura. La evolución de MESU_2 puede seguirse en las tablas 5.9 y 5.10.

			(dir-B)	(dir-B)'	(dir-B)''
	OCPA	MESU_1	MESU_2	MESU_2	MESU_2
P_1	(142,131)	(142,131)	(144,128)	(146,125)	(147,122)
P_2	(104,87)	(104,87)	(107,86)	(110,86)	(113,85)
β_1	—	18°	15°	13°	10°
β_2	—	22°	23°	24°	24°
d	0	0	$2.4 < \lambda$	$5.3 < \lambda$	$8.1 > \lambda$

Tabla 5.9. Alicates. Aquí se satisfacen las condiciones de curvatura (MESU_1), pero no las angulares (MESU_2). Obsérvese la condición de parada, en el sentido (dir-B), indicada por la condición $d > \gamma$ (= 5.9, en este caso).

	(inicializa)	(dir-A)	(dir-A)'
	OCPA	MESU_2	MESU_2
P_1	(142,131)	(141,134)	(139,137)
P_2	(104,87)	(101,88)	(98,89)
β_1	—	19°	18°
β_2	—	23°	22°
d	0	$3.6 < \gamma$	$6.9 > \gamma$

Tabla 5.10. Alicates. Es continuación de la tabla 5.9, explicando la evolución en la dirección opuesta (dir-A). Obsérvese de nuevo la condición de parada para $d > \gamma$ (= 5.9).

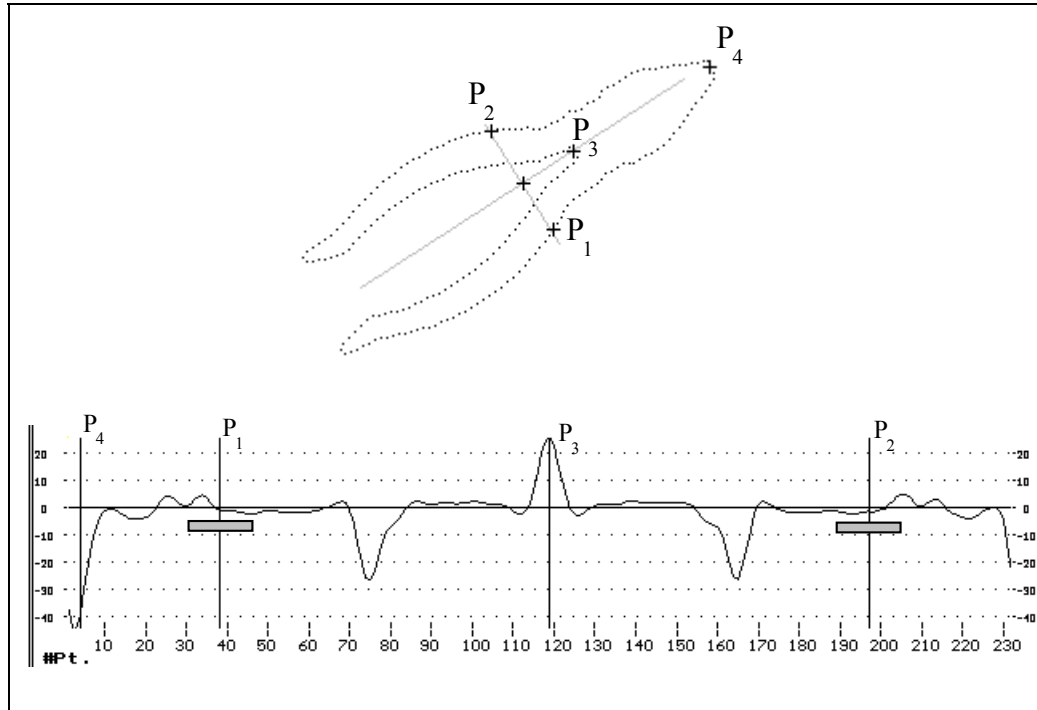


Figura 5.25. Alicates. Se observa a partir del gráfico de torsiones que los puntos candidatos cumplen las condiciones de curvatura suave impuestas por MESU_1.

5.4.4 Invariancia del agarre obtenido

Dado que la estrategia de determinación de puntos de agarre aquí expuesta se basa en información capturada por un sistema de visión artificial, es muy importante que el agarre obtenido no se vea afectado por las transformaciones afines habituales: cambios de escala, traslaciones y orientaciones de los objetos en la escena; es decir, se requiere que sea invariante frente a dichas transformaciones. Para demostrar dicha invariancia haremos uso de la transformación correspondiente entre el sistema de referencia externo y el ligado al objeto (ver la figura 5.26).

Una forma usual de construir transformaciones, homogéneas principalmente, en robótica es resolver el denominado "grafo de la transformación" [Paul, 81] a partir del cuál (véase la figura 5.27) se obtiene:

$${}^W T_P = {}^W T_C \cdot {}^C T_P \Rightarrow {}^C T_P = {}^W T_C^{-1} \cdot {}^W T_P$$

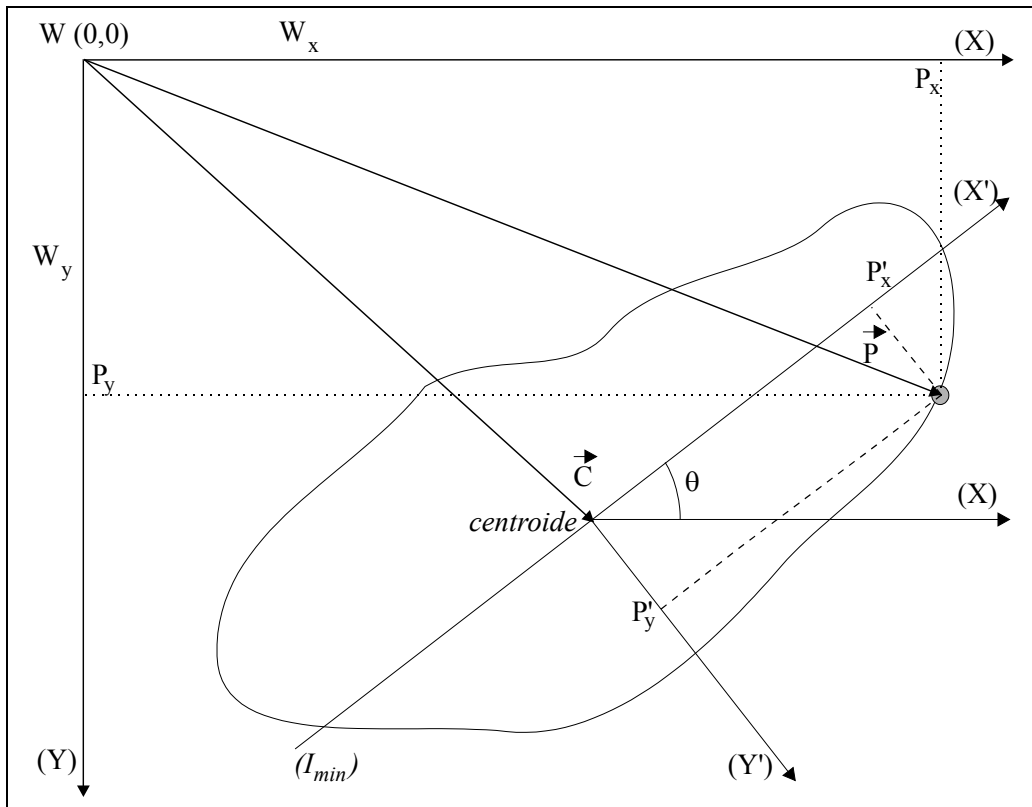


Figura 5.26. Interpretación geométrica de la transformación seguida para demostrar la invariancia de los puntos de agarre determinados. A partir de las coordenadas absolutas de un punto $P (P_x, P_y)$, se obtienen las relativas (P'_x, P'_y) al sistema de referencia ligado al objeto.

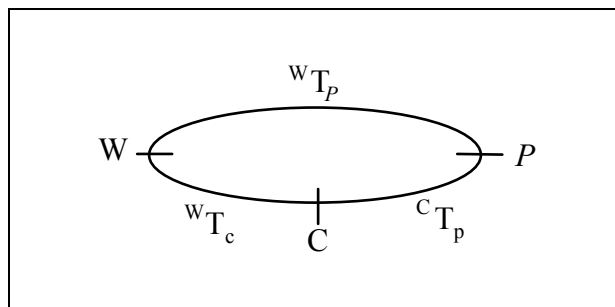


Figura 5.27. Grafo de la transformación asociado al problema de cambio de base representado en la figura 5.26.

y teniendo en cuenta que

$${}^wT_P = \begin{bmatrix} 1 & 0 & 0 & P_x \\ 0 & 1 & 0 & P_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad {}^wT_C = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & C_x \\ \sin \theta & \cos \theta & 0 & C_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

calculando la inversa de esta última se obtiene

$${}^wT_C^{-1} = \begin{bmatrix} \cos \theta & \sin \theta & 0 & -(C_x \cos \theta + C_y \sin \theta) \\ -\sin \theta & \cos \theta & 0 & C_x \sin \theta - C_y \cos \theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Por lo que sustituyendo, se llega a

$${}^cT_P = \begin{bmatrix} \cos \theta & \sin \theta & 0 & -(C_x \cos \theta + C_y \sin \theta) + (P_x \cos \theta + P_y \sin \theta) \\ -\sin \theta & \cos \theta & 0 & (C_x \sin \theta - C_y \cos \theta) + (-P_x \sin \theta + P_y \cos \theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

por tanto, se obtiene para las coordenadas nuevas (relativas) en función de las viejas (absolutas)

$$\begin{aligned} P'_x &= (P_x - C_x) \cos \theta + (P_y - C_y) \sin \theta \\ P'_y &= (C_x - P_x) \sin \theta + (P_y - C_y) \cos \theta \end{aligned} \quad (5.2)$$

Lo anterior representa el tratamiento genérico de un cambio de base del sistema de referencia absoluto (fijo en el espacio) a uno local (solidario al objeto). Para adaptar lo anterior al problema que nos ocupa hemos de hacer uso de la información disponible: centroide y orientación del eje principal (I_{min}). En definitiva, las coordenadas de los puntos de agarre P_1 y P_2 calculados respecto del sistema de referencia absoluto, deben transformarse respecto del sistema de referencia solidario con el objeto. Para ello, se traslada el origen desde la esquina superior izquierda de la pantalla al centroide asociado al contorno, y se toma el eje de abscisas en la dirección del eje principal (I_{min}) cuyo sentido se adapta al convenio visto en el capítulo IV para la dirección del mismo. Para visualizar exactamente la adaptación al presente trabajo se muestra un ejemplo detallado relativo a una imagen, "Destor_1" (véase la figura 5.28) y el efecto ante un cambio de traslación y orientación, "Destor_2" (véase la figura 5.29). Tengamos en cuenta que el criterio de asignación de orientación a los ejes del sistema de referencia local ha sido dejar el ángulo θ igual para valores positivos ($\theta \geq 0$), y tomar el suplementario de dicho ángulo para valores negativos ($\theta < 0$). Los resultados numéricos se muestran en la tabla 5.11.

	Coord-Abs	Coord-Rel
Destor_1 P_1	(177,124)	(0,-13)
Destor_1 P_2	(159,143)	(-1,13)
Destor_2 P_1	(103,146)	(-1,-13)
Destor_2 P_2	(87,127)	(1,12)

Tabla 5.11. Resultados obtenidos aplicando la transformación (5.2) sobre los puntos de agarre visualizados para "Destor_1" y "Destor_2" en las figuras 5.28 y 5.29, respectivamente.

A partir de los valores de la tabla 5.11, se observa que $P_1(\text{Des}_1) = P_1(\text{Des}_2) \pm \sigma$, donde, σ , es la tolerancia inducida por la granularidad de los contornos 3-muestreados (y análogamente para P_2). Dicha tolerancia depende además, del tipo de transformaciones sufridas por un objeto y su grado de simetría asociado. Para explicar esto último se muestra la figura 5.30. A partir de la tabla 5.12, se observa que un simple ajuste permite establecer la correspondencia entre los puntos (P_1, P_2) de un objeto y los de su transformado correspondiente. Dicha correspondencia se encuentra evaluando las diferencias en valor absoluto entre las coordenadas de los puntos, siendo los valores menores los que representan la correspondencia buscada. En el ejemplo de la figura 5.30, con los valores dados por la tabla 5.12, puede observarse que los valores correctos para los puntos originales (Tijer1) vienen marcados por [P_1] y [P_2], una vez realizado el cálculo anterior.

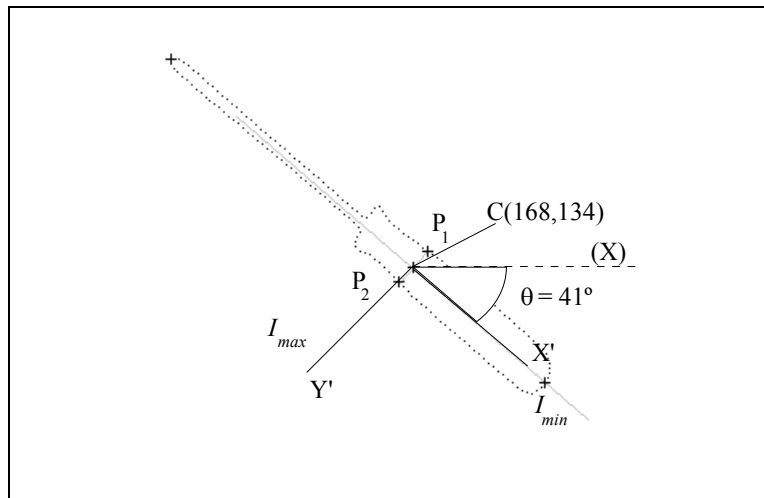


Figura 5.28. Destor_1. Centroide, puntos de agarre y orientación. Obsérvese el sistema de referencia que generado de forma automática se fija al contorno.

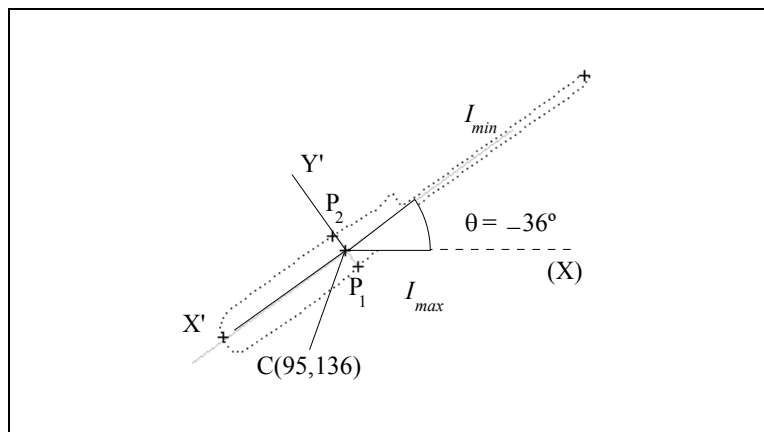


Figura 5.29. Destor_2. Centroide, puntos de agarre y orientación. Obsérvese la adaptación del sistema de referencia a cambios en la orientación (θ). En este caso al cumplirse $\theta < 0$, el eje X' se orienta a partir del suplementario del ángulo ($\theta' = 180 - |\theta|$)

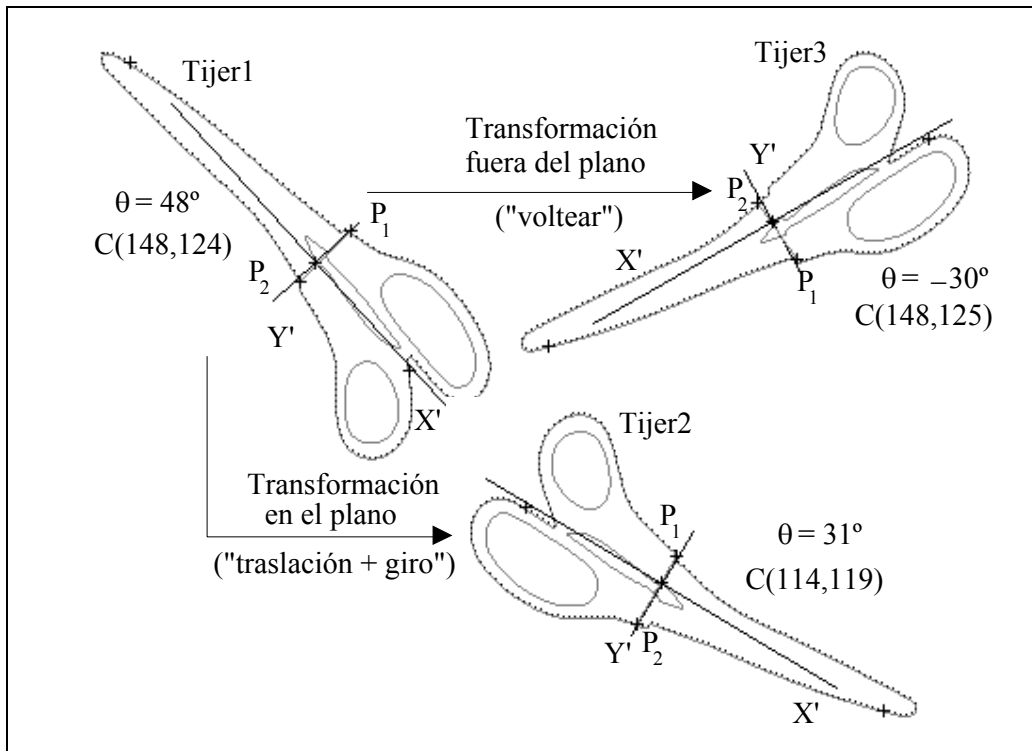


Figura 5.30. Tijeras. Centroide, puntos de agarre y orientación. Obsérvese la adaptación del sistema de referencia a cambios en la orientación (θ). En este caso al cumplirse $\theta < 0$, el eje X' se orienta a partir del suplementario del ángulo ($\theta' = 180 - |\theta|$).

	P_1 Coord-Rel	P_2 Coord-Rel
Tijer1	(0,-25)	(2,13)
Tijer2	(0,-16) [P_2]	(0,26) [P_1]
Tijer3	(-2,-24)	(2,13)

Tabla 5.12. Resultados obtenidos aplicando la transformación (5.2) sobre los puntos de agarre visualizados en las imágenes de la figura 5.30. Se observa la corrección necesaria en Tijer2 (hay que intercambiar los puntos).

5.5. Conclusiones preliminares

Se pueden extraer del presente capítulo las siguientes conclusiones:

- ◆ La estrategia utilizada garantiza la existencia y unicidad del agarre, en cuanto a la determinación de los puntos que constituyen el mismo se refiere.
- ◆ Los algoritmos son capaces de predecir la estabilidad o no del agarre en base a las condiciones impuestas por la definición de umbrales que modelan cada uno de los aspectos necesarios para su caracterización, teniendo en cuenta la presencia de rozamiento en el contacto.
- ◆ Los requerimientos de los algoritmos expuestos, en cuanto a tiempo de CPU se refiere, son mínimos por cuanto todos ellos poseen ordenes de magnitud lineales, en el peor de los casos, con el número de puntos de los contornos, pero en general el coste es $O(1)$.
- ◆ Buen comportamiento ante un universo de trabajo sin apenas restricciones en cuanto a la morfología de los objetos se refiere, a diferencia de otros trabajos [Kamon et al., 96] que no contemplan la posibilidad de formas con centroide exterior a su contorno (como la llave allen), o la existencia de agujeros (asimétricos respecto de I_{min}).
- ◆ Quizás uno de los aspectos más importantes a destacar sea el principio de economía de medios. Prácticamente no existen trabajos de investigación en el tópico de la determinación/análisis del agarre que estén implementados totalmente en ordenadores personales y con lenguajes de programación de uso general como Pascal o C.

Capítulo VI

La Ejecución del Agarre

En este capítulo se muestra, como culminación del trabajo expuesto anteriormente, una vía de integración con un módulo de control con realimentación visual inspirado en un sistema biológico.

6.1. Introducción.

La idea inicial, posteriormente implantada [Sanz et al., 96], fue la de integrar en un mismo sistema (como dos módulos independientes pero conectados entre sí) por un lado lo visto en los capítulos anteriores para la generación del agarre, y por otro, un sistema de control que, haciendo uso de la información visual comunicada por el anterior, fuera capaz de llevar la mano del robot a la localización del agarre. Este proceso se realizará de modo iterativo, haciendo uso de realimentación visual.

El primer módulo, relativo al agarre, juega el papel de un módulo comportamental, llamado por el sistema de control cada vez que lo requiera, con vistas a posicionar la mano del robot en un punto del espacio donde la imagen de la escena cumple ciertos requisitos. En este sentido, el posicionamiento del manipulador se obtiene exclusivamente a partir de las coordenadas visuales y del conocimiento previo de la transformación real cámara-articulaciones en un reducido conjunto de puntos. Lo anterior representa una implantación modificada del reflejo Vestíbulo-Ocular (VOR)¹ [Churchland & Sejnowski, 92],

¹ *Vestibulo-Ocular Reflex*

formalizado por la denominada Teoría de la Red de Tensores [Pellionisz & Llinás, 80]. El uso de realimentación visual permite aumentar la precisión hasta los límites del sistema de visión utilizado.

La implantación real del sistema descrito se llevó a cabo en un robot manipulador, tipo SCARA con seis grados de libertad (véase la figura 6.1), de los cuales se usaron sólo 4, con una microcámara fijada en la muñeca.

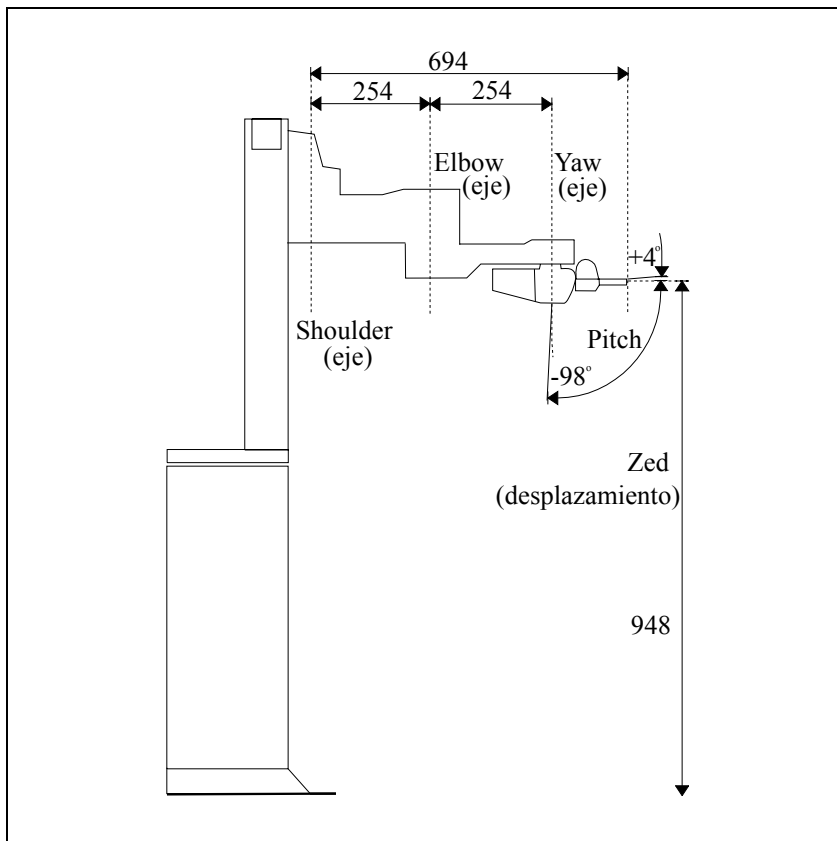


Figura 6.1. Adaptada de [UMI, 91]. Vista lateral del manipulador tipo SCARA utilizado y sus articulaciones. Se representan las cotas en *mm* asociadas a dichas articulaciones en la posición cero ("home") del mismo.

6.2. Estrategia de control empleada

Como se comentó anteriormente, el algoritmo que determina los puntos de agarre devuelve coordenadas visuales. El objetivo principal del sistema visuo-motor será posicionar el brazo de modo que cualquier punto sea visto con las coordenadas deseadas, o en este caso particular, en que la cámara está fijada al brazo, posicionar el brazo de manera que la característica relevante se vea en un punto dado de la imagen, moviendo de forma apropiada el brazo con la cámara. Este proceso teóricamente conducirá el brazo a la posición en la que la pinza podrá realizar un agarre seguro del objeto, orientándose previamente de forma adecuada. Esto ha de realizarse mediante una correspondencia entre las coordenadas de entrada (visuales) y las de salida (motoras), las cuales pueden ser almacenadas previamente o aprendidas *on line*. Nuestro enfoque construye esta correspondencia a partir del conocimiento de las transformaciones visuo-motoras y sus derivadas en un pequeño conjunto de puntos. Está inspirado en la observación del reflejo Vestíbulo-Ocular, siendo uno de los objetivos del presente trabajo, utilizar modelos biológicos de reflejos existentes (*reflexive behaviour*) como fuente de inspiración de un sistema robotizado real. Así, la formulación aquí propuesta para los algoritmos de posicionamiento visuo-motor es una adaptación libre de dicho reflejo, que controla el movimiento del globo ocular para mantener su orientación fija a pesar del movimiento de la cabeza. El trabajo de los neurofisiólogos Pellionisz y Llinás [Pellionisz & Llinás, 80], [Pellionisz, 85] muestra que existe una correspondencia directa entre las lecturas de los canales semicirculares y las señales de los músculos oculares. Los primeros son sensibles a la orientación de la cabeza, mientras que los últimos mueven de forma apropiada cada globo ocular en tres orientaciones diferentes a partir de tres pares de músculos.

La explicación sugerida por estos autores se basa en el hecho de que el cerebelo parece aplicar una transformación lineal local a los estímulos de entrada (la señal proveniente de los canales semicirculares) generando directamente las señales de salida sin procesado posterior. La transformación no es la misma para el espacio de entrada completo sino que varía de unos puntos a otros, lo cual permite que una función global no lineal sea modelada hasta un cierto límite de precisión. La transformación lineal debe ser medida o aprendida en algún conjunto conocido de puntos discretos del área de trabajo, y mediante un determinado mecanismo seleccionar la mejor transformación a usar para un punto dado. Esto puede formalizarse como una red de Jacobianos extendida a todo el espacio de entrada, los cuales juegan el papel de transformaciones diferenciales que actúan sobre las diferencias entre los puntos deseados y los más próximos a los puntos sensoriales almacenados.

Es conveniente introducir a continuación la notación que se utiliza en lo que sigue:

- v : coordenadas visuales.
- x : coordenadas cartesianas.
- m : coordenadas motoras.

Con vistas a determinar las relaciones entre los vectores diferenciales en un punto dado, llamemos f_1 a la función que transforma coordenadas visuales en coordenadas cartesianas, $f_1 : \mathbf{v} \rightarrow \mathbf{x}$ obtenido en aproximaciones clásicas a partir de procedimientos analíticos de calibrado de cámara [Tsai, 86]; f_1 puede ser desarrollado en serie alrededor de cada punto \mathbf{v}_0 como

$$\mathbf{x} = \mathbf{x}_0 + (J_{f_1})_0 \cdot (\mathbf{v} - \mathbf{v}_0) + O(\|\mathbf{v} - \mathbf{v}_0\|^2) \quad (6.1)$$

donde $(J_{f_1})_0$ es la matriz Jacobiana de la transformación en el punto $f(\mathbf{v}_0) = \mathbf{x}_0$, representada por

$$(J_{f_1})_0 = \begin{bmatrix} \frac{\partial x}{\partial v_1} & \dots & \frac{\partial x}{\partial v_k} \\ \frac{\partial y}{\partial v_1} & \dots & \frac{\partial y}{\partial v_k} \\ \frac{\partial z}{\partial v_1} & \dots & \frac{\partial z}{\partial v_k} \end{bmatrix} \quad (6.2)$$

donde k es el número de coordenadas visuales elegidas para determinar un punto, normalmente tres, pero pudiendo elegir más para mejorar la robustez en un sistema sobredeterminado (redundante). En el mismo sentido, la transformación de coordenadas cartesianas a motoras (cinemática inversa) es una función $f_2 : \mathbf{x} \rightarrow \mathbf{m}$ que puede también desarrollarse como

$$\mathbf{m} = \mathbf{m}_0 + (J_{f_2})_0 \cdot (\mathbf{x} - \mathbf{x}_0) + O(\|\mathbf{x} - \mathbf{x}_0\|^2) \quad (6.3)$$

siendo $(J_{f_2})_0$ la matriz Jacobiana de la transformación evaluada en el punto motor \mathbf{m}_0 , representada por

$$(J_{f_2})_0 = \begin{bmatrix} \frac{\partial m_1}{\partial x} & \frac{\partial m_1}{\partial y} & \frac{\partial m_1}{\partial z} \\ \vdots & \vdots & \vdots \\ \frac{\partial m_l}{\partial x} & \frac{\partial m_l}{\partial y} & \frac{\partial m_l}{\partial z} \end{bmatrix} \quad (6.4)$$

donde l es el número de coordenadas motoras, normalmente igual al número de grados de libertad del manipulador. Combinando las ecuaciones (6.3) y (6.1) se obtiene

$$\mathbf{m} = \mathbf{m}_0 + (J_{f_2})_0 \cdot (J_{f_1})_0 \cdot (\mathbf{v} - \mathbf{v}_0) = \mathbf{m}_0 + H_0 (\mathbf{v} - \mathbf{v}_0) \quad (6.5)$$

siendo $H_0 = (J_{f_2})_0 \cdot (J_{f_1})_0$, que relaciona directamente variaciones en las coordenadas visuales cerca del punto m_0 con variaciones en las coordenadas motoras (para este punto las coordenadas visuales son v_0 , ya que corresponde a la localización física x_0).

Las matrices de las derivadas, H_0 , son medidas y almacenadas, junto con las coordenadas visuales del mismo punto, y también las coordenadas motoras (*joints*) para ese punto, como se explica en el apartado siguiente.

6.3. Descripción del sistema

El dispositivo experimental está formado por un manipulador tipo SCARA con baja repetibilidad (0.5 mm), más una microcámara CCD situada anexa al elemento terminal, tal y como se observa en la figura 6.1. El robot posee 6 grados de libertad, pero sólo se usaron 4, pues son suficientes para el agarre de objetos planos: *zed*, *shoulder*, *elbow*, y *yaw* (véase la figura 6.1, donde se aprecia el manipulador empleado y sus articulaciones). La articulación *pitch* se fijará a -90° . Estos valores se enviarán al manipulador como pasos para los *encoders*. La altura sobre la escena, variable con la coordenada *zed*, sólo tomará tres valores posibles cuya utilidad se explicará a continuación.

El *hardware* para capturar y procesar imágenes consiste en una microcámara CCD ($3 \times 3 \times 5$ cm) con una resolución de 256×256 píxeles, la cual genera una señal de vídeo que es digitalizada y almacenada mediante una tarjeta Matrox MVP-AT. Se han utilizado 64 niveles de gris. Todos los programas para detectar, obtener los puntos de agarre y generar las señales de posicionamiento del manipulador se han ejecutado en un ordenador 486DX, a 33 MHz con 8 Mb de RAM. Una panorámica del sistema completo, incluyendo una escena típica de su área de trabajo, se muestra en la figura 6.2.

Veamos a continuación el uso particular de la formulación tensorial descrita anteriormente para el posicionamiento del brazo. La fase de aproximación (*approaching phase*) se dividió en dos partes: posicionamiento grueso (*gross positioning*), que trabaja sobre la imagen tomada por la cámara con el brazo en su altura máxima ("posición alta"), y posicionamiento fino (*fine positioning*), que opera con la imagen tomada a una altura apropiada para ver al objeto ocupando la mayor parte de la misma ("posición media"). El agarre se ejecuta con los dedos de la pinza casi tocando el plano de la escena ("posición baja"). Estableceremos dos correspondencias (véase la figura 6.3):

- C1 La primera obtendrá las coordenadas (*shoulder*, *elbow*, *yaw*) del brazo en su altura media, a partir de las coordenadas visuales (v_{xh} , v_{yh}) del centroide aproximado de cada objeto visto por la cámara en la posición alta.

- C2** La segunda cambiará las coordenadas motoras antedichas, añadiéndoles las correcciones que harán que el centroide, en principio, del objeto elegido coincida con alguna localización visual dada para, finalmente, posicionarse sobre el punto medio de la línea de agarre. Esta fase trabaja con las imágenes tomadas en la posición media.

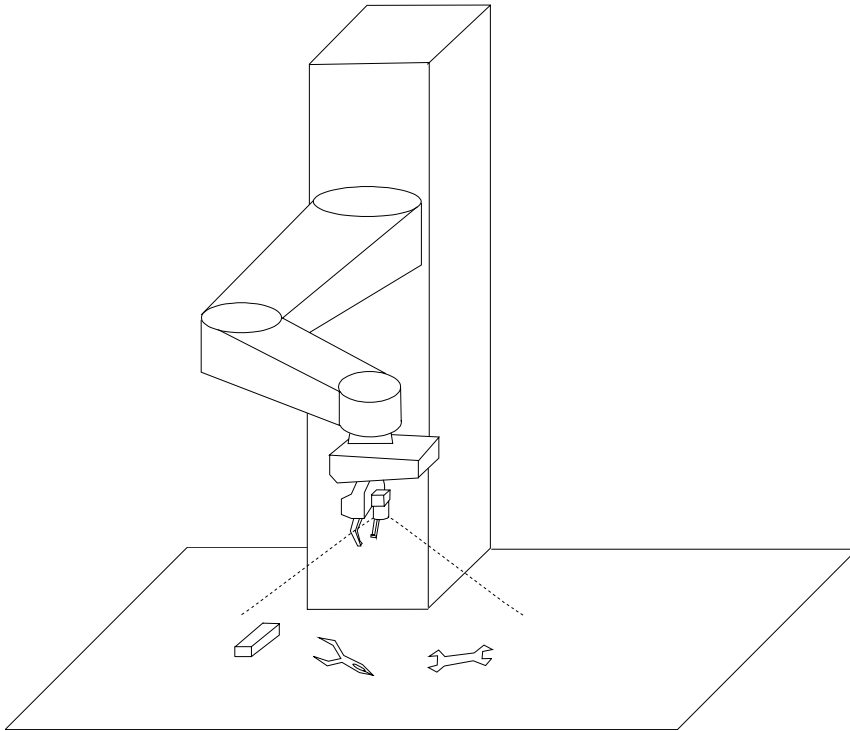


Figura 6.2. Representación gráfica del sistema completo, robot y microcámara, con su área de trabajo.

Es importante puntualizar que tal y como queda reflejado en la figura 6.3, es en la segunda fase (posicionamiento fino) donde la correspondencia establecida (**C2**) incorpora realimentación visual para efectuar las correcciones de posicionamiento necesarias hasta hacer coincidir el centroide calculado (v_{xm} , v_{ym}) con la posición visual conocida más próxima, perteneciente a una red de puntos adecuadamente equiespaciados en la escena y generado previamente (*off-line*). Una vez alcanzada dicha posición se extraen los puntos de agarre y el punto medio de la línea que los une (línea de agarre), utilizando la matriz Jacobiana más próxima para llevar el brazo desde el punto sobre el centroide (donde ahora se encuentra) al punto situado sobre el punto medio de la línea de agarre, todo realizado en la denominada posición media. La convergencia del último paso se garantiza por ser la función de transformación cámara-

articulaciones suave y localmente diferenciable. El procedimiento para encontrar la red de puntos (*grid*) y su significado se explicará en el siguiente apartado (6.3.1).

Todo el proceso anterior de reajuste fino en el posicionamiento mediante realimentación visual se realiza exclusivamente en la denominada "posición media", restando una etapa final de rotación de los dedos de la pinza, mediante el cambio angular de la articulación *roll*, para poder orientar la pinza a lo largo del eje que define la línea de agarre (la definida entre los dos puntos de agarre).

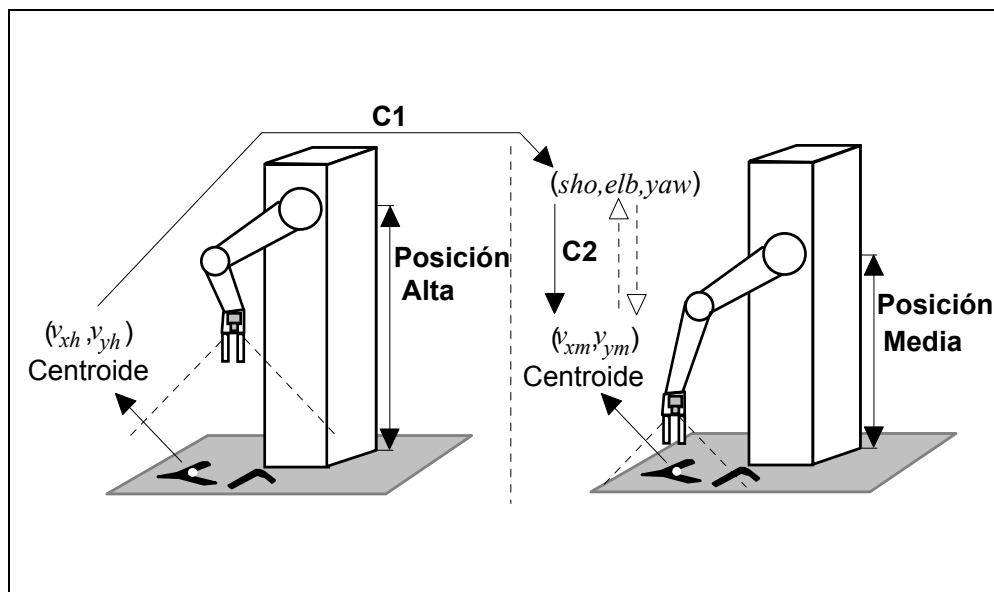


Figura 6.3. Representación gráfica de las dos correspondencias establecidas (**C1** y **C2**) para desencadenar la fase de aproximación inmediatamente anterior a la ejecución del agarre. Posicionamiento grueso (izquierda). Posicionamiento fino (derecha). Obsérvese que **C2** hace uso de realimentación visual.

6.3.1. Proceso "off-line"

La fase previa de toma de medidas para el calibrado, realizada *off-line*, utiliza una pieza simple de madera de color blanco como la mostrada en la figura 6.4.

El manipulador coge y deja la pieza en un punto específico de su área de trabajo, tomando imágenes para localizar el punto central como el centro del rectángulo que encierra el perímetro detectado, posiblemente ruidoso. Una vez que la pieza ha sido dejada en la escena, sólo se toma una imagen con el brazo (y la cámara) en una única posición alta. Llamando (v_{xh}, v_{yh}) a las coordenadas visuales del centro visto desde la posición alta, se almacenará una lista con los pares $((v_{xh}, v_{yh}), (sho, elb, yaw))$ donde (sho, elb, yaw) son las coordenadas motoras del brazo justo cuando se encuentra sobre el objeto, conocidas por ser las utilizadas al dejar el objeto en la escena, es decir, cuando fue visto desde la posición media. (Si la altura no dependiera únicamente de zed , sería necesario almacenar una correspondencia diferente).

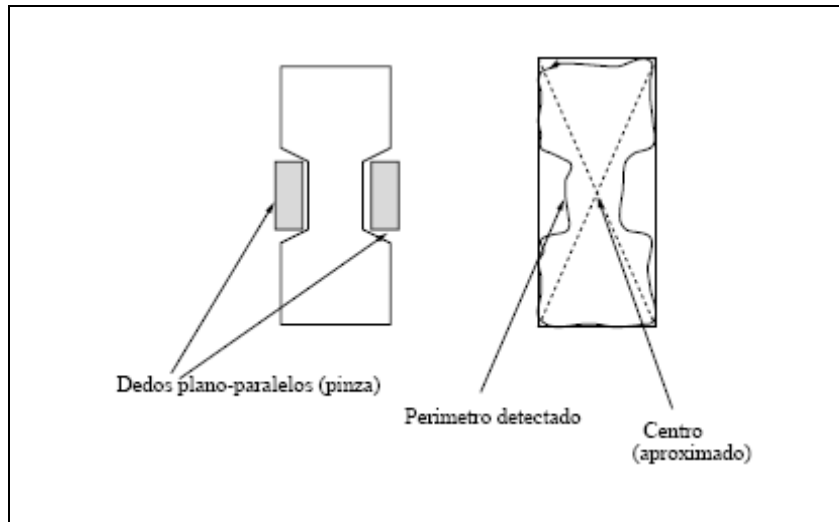


Figura 6.4. Pieza de madera utilizada en el calibrado previo (proceso *off-line*).

A continuación, se toman cinco imágenes con el brazo en la posición media: una en el centro, y las otras moviendo el brazo una pequeña distancia a lo largo de las direcciones x , $-x$, y , $-y$. A partir de aquí, las primeras derivadas de las coordenadas visuales y motoras con respecto al punto (x, y) se calcularán como:

$$\frac{\partial sho}{\partial x} \cong \frac{\Delta sho}{\Delta x} \quad ; \quad \frac{\partial v_{xm}}{\partial x} \cong \frac{\Delta v_{xm}}{\Delta x} \quad (6.6)$$

y análogamente para el resto de coordenadas y sus derivadas.

A partir de aquí se construyen las dos matrices siguientes:

$$(J_{f_2})_0 = \begin{bmatrix} \frac{\partial sho}{\partial x} & \frac{\partial sho}{\partial y} \\ \frac{\partial elb}{\partial x} & \frac{\partial elb}{\partial y} \\ \frac{\partial yaw}{\partial x} & \frac{\partial yaw}{\partial y} \end{bmatrix}; (J_{f_1})_0^{-1} = \begin{bmatrix} \frac{\partial v_{xm}}{\partial x} & \frac{\partial v_{xm}}{\partial y} \\ \frac{\partial v_{ym}}{\partial x} & \frac{\partial v_{ym}}{\partial y} \end{bmatrix} \quad (6.7)$$

Y multiplicando la primera por la inversa de la segunda, se obtiene

$$\begin{bmatrix} \frac{\partial sho}{\partial x} & \frac{\partial sho}{\partial y} \\ \frac{\partial elb}{\partial x} & \frac{\partial elb}{\partial y} \\ \frac{\partial yaw}{\partial x} & \frac{\partial yaw}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial v_{xm}}{\partial x} & \frac{\partial v_{xm}}{\partial y} \\ \frac{\partial v_{ym}}{\partial x} & \frac{\partial v_{ym}}{\partial y} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{\partial sho}{\partial v_{xm}} & \frac{\partial sho}{\partial v_{ym}} \\ \frac{\partial elb}{\partial v_{xm}} & \frac{\partial elb}{\partial v_{ym}} \\ \frac{\partial yaw}{\partial v_{xm}} & \frac{\partial yaw}{\partial v_{ym}} \end{bmatrix} = H_0 \quad (6.8)$$

que es justo la matriz necesaria para relacionar las variaciones en las coordenadas visuales con aquéllas relativas a las coordenadas motoras en un punto dado, a altura media.

La toma de medidas se realizó en puntos separados 5 cm unos de otros, lo que en nuestro caso garantiza un grado aceptable de linealidad en el área circundante de cada punto. El número total de puntos en nuestro espacio de trabajo fue de 85, formando una red (*grid*) de 5×17 puntos.

Puede argumentarse que la medición de estas cantidades usa la cinemática inversa para posicionar el brazo en cada punto, y también para desplazarlo pequeños incrementos en cada dirección, pero conviene puntualizar que, si ésta no se conoce con precisión, podrán utilizarse modelos aproximados, ya que las imprecisiones serán subsumidas y corregidas por el proceso de realimentación *on-line*. Además, los cálculos requeridos para la cinemática inversa se efectuarán en su totalidad *off-line*, consiguiendo que el movimiento del brazo, *on-line*, sea mucho más rápido.

Únicamente el producto, H_0 , de ambas matrices, J_{f_1} y J_{f_2} , (ver ecuaciones 6.2 y 6.4), será guardado, junto con las coordenadas visuales y motoras del punto correspondiente, formando una estructura de datos conectada. Toda la información cartesiana carece de importancia y puede descartarse, ya que las transformaciones cinemáticas son subsumidas por los valores numéricos de la red de Jacobianos. Se necesitará un procedimiento eficiente de almacenamiento de estas estructuras de datos para facilitar su rápida recuperación *on-line*. Se ha utilizado para ello un fichero que guarda la información en binario y que

permite ser cargado en memoria de forma rápida, donde se construyen las tablas de indexado que usará un algoritmo de recuperación de puntos, basado en la búsqueda del vecino más próximo. Dicha búsqueda, si el número de puntos es pequeño (nuestro caso), puede hacerse de forma secuencial, pero en general, lo más eficiente es seguir el mecanismo propuesto por otros autores [Bentley & Papadimitriou, 80], al respecto.

6.3.2. Proceso "on-line"

Podemos resumir el algoritmo (*on-line*) empleado en los siguientes ocho pasos:

1. Ir a la "posición alta" y tomar una imagen global del área de trabajo.
2. Procesar dicha imagen para encontrar las partes de la escena asociada a los diferentes objetos presentes en ella, obteniendo un centro visual aproximado para cada uno de ellos, que podrá ser su centroide o simplemente el centro del rectángulo que lo encierra.
3. Seleccionar un objeto para el agarre. Esto será realizado por un planificador de tareas global o por algún otro procedimiento, cuyo diseño está fuera de los objetivos del presente trabajo.
4. Recuperar el punto visual más cercano de entre aquéllos almacenados en la primera etapa (*off-line*) de medida (el punto visual alto denominado (v_{xh}, v_{yh})). Recuperar su punto visual medio asociado (v_{xm}, v_{ym}) , su punto en coordenadas motoras (sho, elb, yaw) y su matriz Jacobiana.
5. Dirigir la mano a (sho, elb, yaw) con altura media y tomar una imagen de nuevo. Utilizar el enfoque explicado en el capítulo IV, para obtener el centro visual y orientación adecuados.

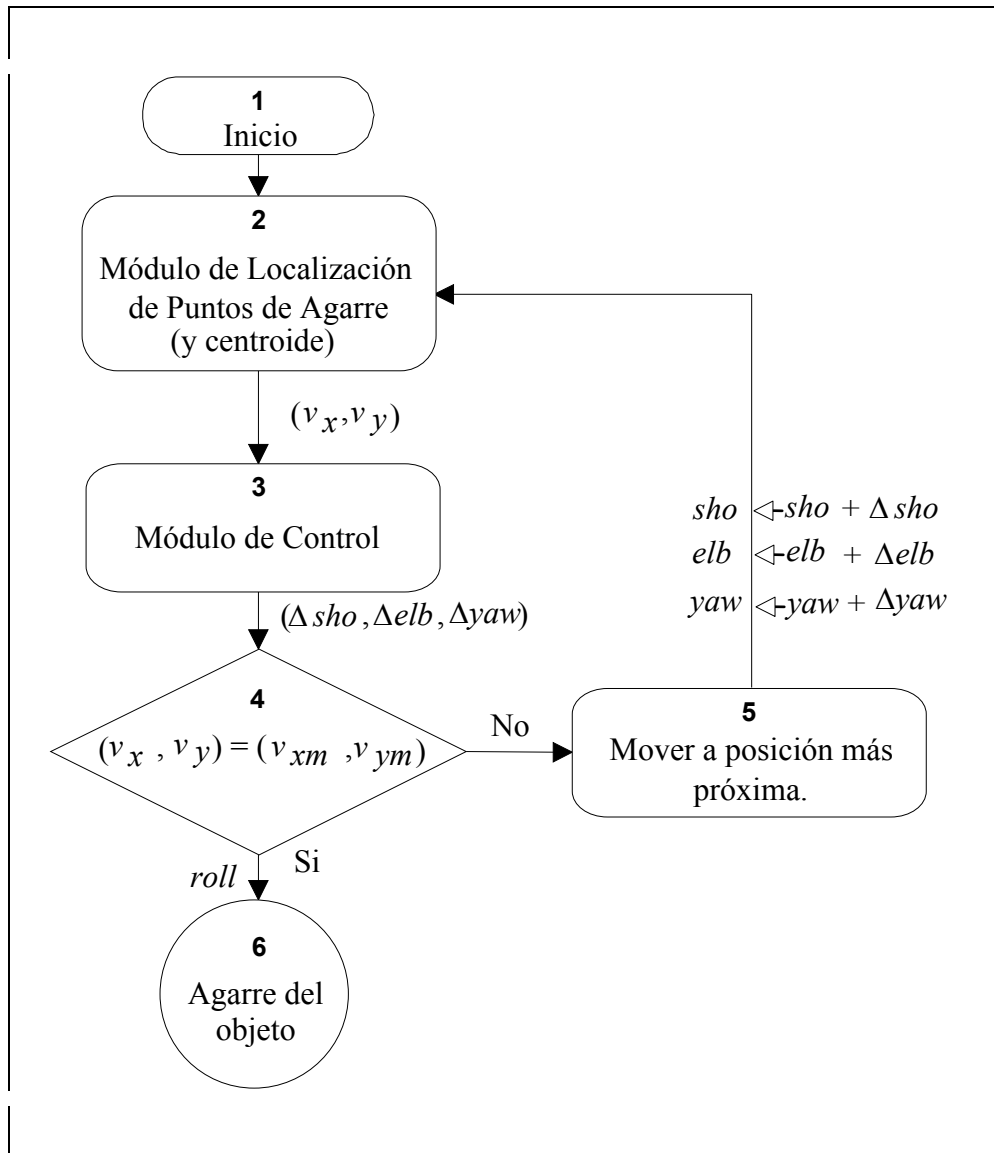


Figura 6.5. Diagrama de flujo del esquema global de trabajo (proceso *on-line*). A destacar el proceso de realimentación existente entre los bloques 2 y 3. Nótese que el módulo 2, encapsula dos operaciones: extracción del centroide utilizado en la realimentación y determinación de puntos de agarre, en el último paso.

6. Mientras el centroide del objeto no coincida con (v_{xm}, v_{ym}) , calcular la diferencia entre ambos y multiplicar por la matriz Jacobiana. Añadir a la posición motora actual el vector de articulaciones (motor) resultante y situar la mano en dicha posición.
7. Iterar este procedimiento hasta que el centroide del objeto coincida con (v_{xm}, v_{ym}) . Esto habrá conducido la mano del robot a un punto próximo

al de agarre del objeto, ya que éste es el punto en el que el objeto de prueba (proceso *off-line*) fue visto justo después de ser soltado por la pinza del robot.

8. Finalmente, llevar la mano sobre el punto medio de la línea de agarre, utilizando la matriz Jacobiana más próxima, rotando, a continuación, la articulación *roll* el mismo ángulo que dicha línea tenga con el eje visual de abscisas (x).

La última etapa supone cierto conocimiento geométrico, ya que, por la ubicación de la cámara elegida, las orientaciones real y visual casi coinciden. Esta suposición puede no ser cierta para otros sistemas, pero en tal caso, una tercera correspondencia entre el ángulo de la articulación *roll* y la variable orientación puede ser almacenada tomando varias medidas en cada punto para diferentes orientaciones.

Como ya se comentó anteriormente, el algoritmo que se ha utilizado para encontrar el punto visual más cercano a uno dado es una adaptación del propuesto por [Bentley & Papadimitriou, 80] para realizar una búsqueda n -dimensional, asumiendo la distancia euclídea.

En la figura 6.5, se visualiza gráficamente el esquema de trabajo global de los algoritmos descritos. Se trata de un esquema simplificado en el que se aprecia la integración de los dos módulos desarrollados, de manera que, como se ha explicado anteriormente, el módulo de control (3) llamará al de localización de puntos de agarre (2) cada vez que lo considere necesario, hasta desencadenar la ejecución del mismo. Obsérvese que la determinación de puntos de agarre sólo se realiza una vez, siendo la extracción del centroide la que realmente interviene en el proceso de posicionamiento con realimentación visual. Esta estrategia presupone, debido a las condiciones de estabilidad asumidas (ver capítulo III), que el centroide siempre estará próximo al punto medio de la línea de agarre.

6.4. Resultados.

Puntualicemos que el módulo de localización de puntos de agarre, visto anteriormente, se comporta de forma autónoma, tomando una imagen y devolviendo coordenadas visuales cada vez que es invocado por el módulo de control motor. Análogamente, el módulo visuo-motor toma coordenadas visuales, moviendo a continuación el manipulador de un modo completamente independiente. A pesar de que estos módulos no pueden denominarse realmente módulos comportamentales² en sentido estricto, ya que (particularmente el módulo de generación del agarre) usan modelos simbólicos, representan un

² Ver el punto 1.2 acerca de su definición.

primer intento en la construcción de una arquitectura más flexible para una clase restringida de sistemas de ensamblado.

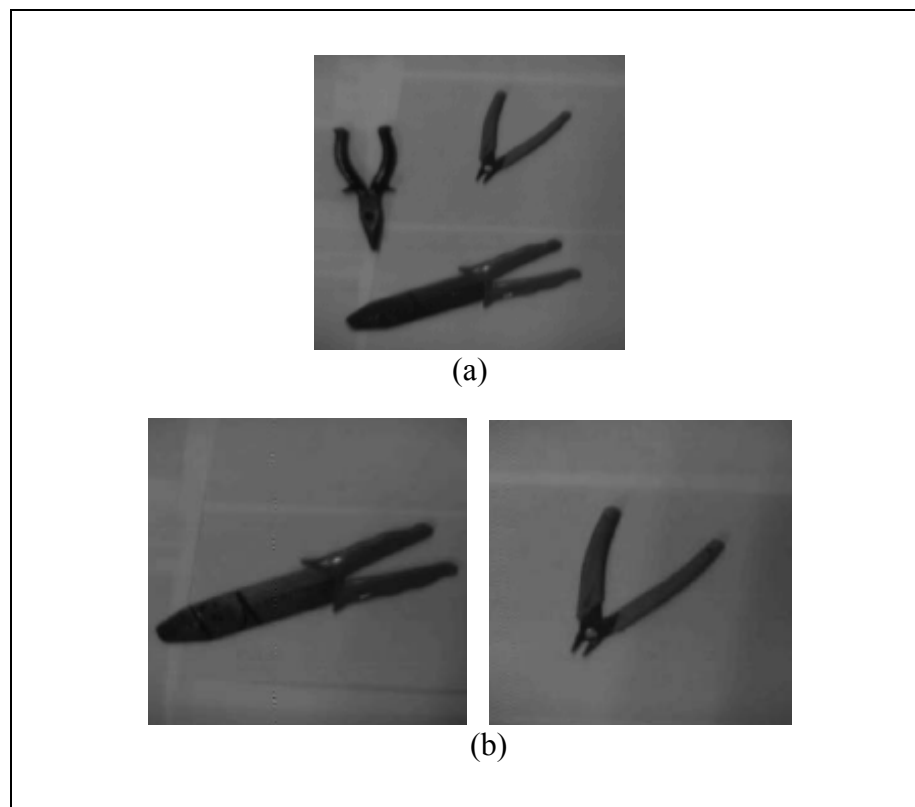


Figura 6.6. Imágenes capturadas por la microcámara en: (a) "posición alta" y (b) "posición media" correspondientes al esquema de trabajo *on-line* comentado.

El algoritmo de posicionamiento fue evaluado conduciendo al manipulador a cada uno de los 85 puntos formados por una red (*grid*) de 5 × 17 puntos (tal y como se indicó en la fase "*off-line*"), elegidos en el centro de la red original donde fueron tomadas las medidas, de forma que éstos serán los puntos donde se espera el máximo error. Esto es así, debido a que los Jacobianos están medidos en los nodos de la red (*grid*), de manera que cuanto más alejado esté el punto de medición más próximo, mayor será dicho error, ya que la aproximación es lineal a tramos, y en una red regular, los centros de cada polígono en que se divide el espacio son los puntos más lejanos a los nodos, por lo que el error medio en dichos puntos es cota superior del mismo. El error medio sin realimentación visual fue de 1.6 mm. No hubo diferencias significativas en el comportamiento del error dependiente de la posición. Este error medio se redujo a 1.2 mm con un ciclo de realimentación visual, a 0.9

con dos ciclos y a 0.7 con tres. Más ciclos no mostraron una mejora sustancial. Esto se debe a la limitación en la precisión inherente al sistema de visión, ya que el tamaño de un píxel a nuestras distancias es de 0.6 mm, aproximadamente. No obstante, no existe en teoría ninguna razón por la que dicha precisión no pueda mejorarse, bien utilizando cámaras de mayor resolución o bien utilizando precisión subpíxel en los algoritmos de visión. En tal caso, deberían medirse los Jacobianos en más puntos, ya que la memoria requerida está lejos todavía de alcanzar los límites de cualquier ordenador corriente, y el tiempo de recuperación del punto visual más cercano, con el algoritmo utilizado, es aproximadamente constante si los puntos visuales se distribuyen de manera uniforme en el espacio de entrada, como es el caso.

En la figura 6.6, se pueden apreciar las dos fases de trabajo del algoritmo *on-line*, a través de las correspondientes imágenes obtenidas: (a) "posición alta", desde donde se contempla toda la escena cubierta por el campo visual de la microcámara. Y (b) "posición media", a partir de la cuál, el algoritmo, mediante tres o cuatro iteraciones a lo sumo, es capaz de posicionar el manipulador justo sobre el objeto para, a continuación, proceder a su agarre.

Es interesante observar el comportamiento del manipulador en acción ante dos situaciones antagónicas: caso-1, cuando se determina un agarre estable (véase la figura 6.7), y caso-2, cuando se incumple algún requisito necesario para garantizar la estabilidad del agarre (véase la figura 6.8).

Puntualicemos que en el caso-2 (ver figura 6.8) relativo a las pinzas, la no garantía de un agarre estable no implica imposibilidad absoluta de ejecutar el agarre. No obstante el sistema predice que existe una probabilidad muy alta de que se produzca algún tipo de alteración en la posición del objeto relativa a la pinza, como de hecho se observa que ocurre. De esta manera se facilita el tomar las decisiones adecuadas en fases de manipulación inmediatamente posteriores al agarre propiamente dicho (e.g. en aplicaciones de ensamblado, inserción, etc.).

El tiempo total de ejecución en un ordenador 486-DX33 está entre 0.3 y 0.8 segundos para el módulo de localización de puntos de agarre (incluyendo todas las etapas previas), dependiendo de la longitud de los contornos, ya que es lineal con ellos; dicho tiempo se desdobra en el necesario para obtener el centroide a partir de la captura de la imagen (0.5 s), y a partir de aquí, determinar los puntos de agarre (0.3 s). Y 0.1 segundos para el módulo de control. Como se pueden requerir hasta cuatro ciclos visuales de realimentación, el tiempo total será de $4 \times (0.1 + 0.5) + 0.3 = 2.7$ segundos. Esto, no es de hecho exactamente así, debido a que una gran cantidad del tiempo se consume en el transcurso de la lenta reacción del manipulador para seguir las instrucciones de posicionamiento.

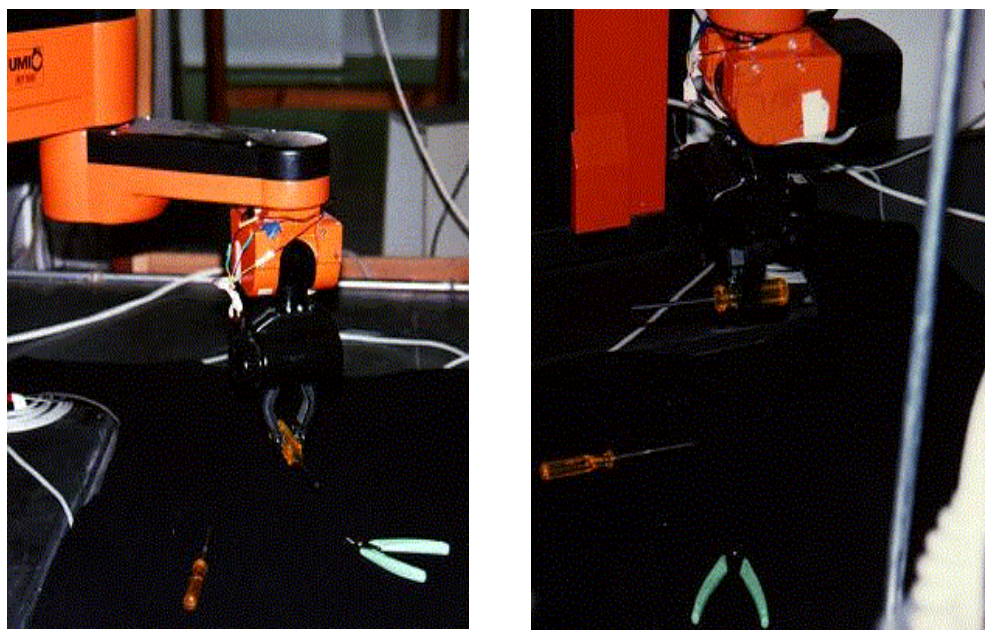


Figura 6.7. El manipulador en acción: agarre "estable" de un destornillador (caso-1). Tal y como se demostró en el capítulo V, este caso cumple todos los requisitos para que el sistema garantice la estabilidad en el agarre.

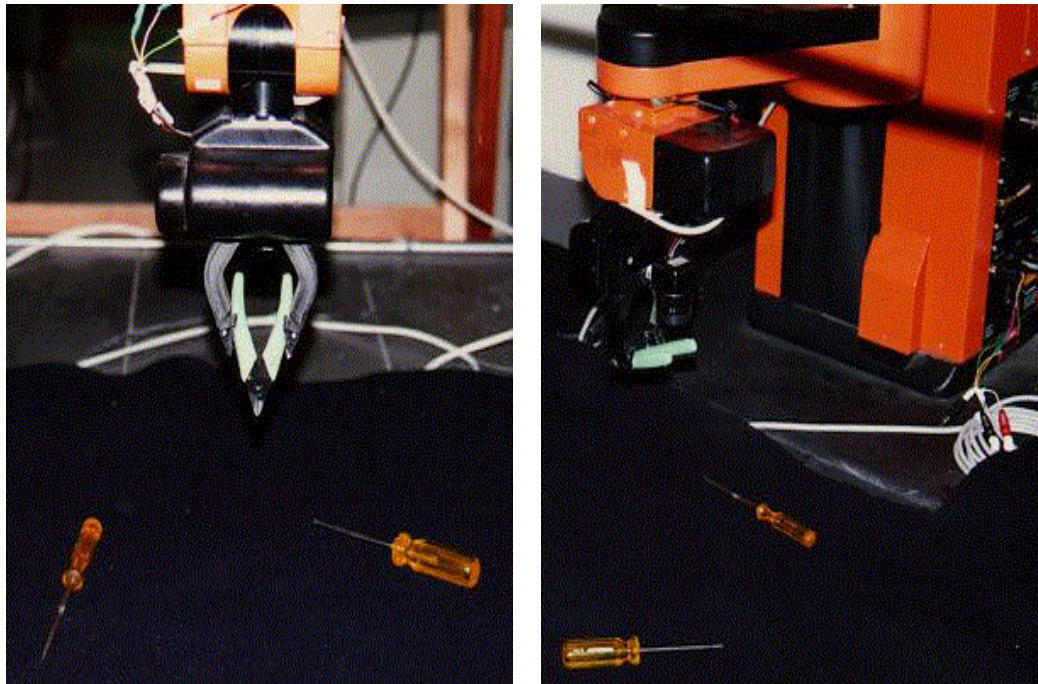


Figura 6.8. El manipulador en acción: agarre "inestable" de unas pinzas (caso-2). Como se demostró en el capítulo V, este caso no cumple la 2ª condición del Mecanismo Supervisor (MS), por tanto el sistema lo detecta no garantizando su estabilidad.

6.5. Conclusiones preliminares.

Se pueden extraer del presente capítulo las siguientes conclusiones:

- ◆ En cuanto a los algoritmos de posicionamiento visual empleados, ha quedado demostrado que modelos geométricos de la transformación de cámara o de la cinemática del manipulador no son necesarios para conseguir una buena actuación del mismo, y que la observación de los modelos biológicos es una fuente válida de inspiración para los sistemas robotizados reales.
- ◆ Se ha demostrado la viabilidad de la integración de un módulo de determinación del agarre basado en visión que sirve de entrada al módulo de control que dispara la ejecución del agarre, poniendo de manifiesto su robustez, eficiencia y economía de medios en un entorno real.
- ◆ Hay que hacer especial hincapié en que uno de los méritos de la implementación mostrada aquí, radica precisamente en la sencillez de los medios empleados. No es fácil gestionar una aplicación de esta complejidad con un simple "pc". Esto permite empezar a pensar en abrir una posible vía para su implantación en entornos industriales o de otro tipo, con necesidades de manipulación en general.

Capítulo VII

Conclusiones y Trabajo Futuro

En este capítulo se hace una recapitulación de los objetivos alcanzados, limitaciones a superar y, posibles vías abiertas para seguir avanzando.

7.1. Aportaciones del presente trabajo.

El objetivo central cubierto por esta tesis ha sido demostrar la viabilidad de un sistema completo para la determinación y ejecución del agarre en robots manipuladores mediante razonamiento geométrico basado en visión. Dicho sistema actúa en un entorno real, demostrando robustez, eficiencia y economía de medios, haciendo uso del principio "percepción-razonamiento-acción" inherente a la investigación en el campo de la robótica inteligente.

Los algoritmos desarrollados se han implementado en un robot manipulador con 4 GDL y fijando una cámara CCD en su muñeca. El sistema de control utiliza realimentación visual, consiguiendo incrementar la precisión hasta el límite del sistema de visión utilizado.

A modo de síntesis se pueden destacar las siguientes conclusiones extraídas de las diferentes conclusiones preliminares de los capítulos anteriores:

Un umbralizado automático de escenas aproximadamente bimodales que hacen al sistema robusto a pequeños cambios globales de iluminación.

En la fase de procesamiento de imágenes de trabajos cercanos al nuestro [Fuentes et al., 94], [Kamon et al., 96], deja bastante que desear algún procedimiento, como el de tener que seleccionar el usuario cierto nivel de umbral "a mano", para facilitar la segmentación; en éste sentido nuestro sistema es más robusto, obteniendo resultados aceptables.

Un procesado rápido del cálculo de momentos y, por ende, de los descriptores utilizados que permiten, utilizando vectores dinámicos para los puntos del contorno obtenido con 3-muestreo, establecer el razonamiento geométrico necesario en un tiempo óptimo.

Una estrategia extremadamente simple que permite la extracción automática de los dos candidatos a puntos de agarre, de forma eficiente, sobre formas planas sin restricciones, garantizando la existencia y unicidad del agarre.

Un mecanismo de validación ("mecanismo supervisor") de los candidatos a puntos de agarre obtenidos inicialmente que ha mostrado un comportamiento eficiente y robusto sobre una gran variedad de objetos utilizados.

Demostrar la validez y viabilidad de los algoritmos desarrollados, mediante su implementación en un sistema real, integrado con un módulo de control que, haciendo uso de la información entregada por dichos algoritmos, mediante realimentación visual, ha verificado un comportamiento aceptable en las diferentes situaciones a las que ha sido sometido.

7.2. Líneas de investigación futura.

Siguiendo otros trabajos [Kamon et al., 96] sería muy conveniente la incorporación al sistema desarrollado de mecanismos de aprendizaje que permitan al mismo mejorar el agarre a medida que este se experimenta con objetos de distinta morfología, en base a mecanismos de prueba y error. Recordemos que la información visual empleada se ha limitado a inducir todo el razonamiento necesario a partir de los contornos y, en general, esto no es suficiente en el proceso de agarre; se debe poder estimar de alguna manera la distribución de masas del objeto, que no tiene porqué ser homogéneo, rígido, etc.

El esquema anterior podría así mismo conectarse para trabajar con objetos 3D, empleando para ello un sistema de posicionamiento ya desarrollado

[Domingo, 91], [Domingo, 93] que utiliza la formulación TNT (véase el capítulo VI) con dos cámaras no alineadas enfocando el área de trabajo. Este sistema podría utilizarse en la fase de posicionamiento grueso, dejando la cámara montada en la muñeca del robot para el posicionamiento fino en la fase de ejecución del agarre, evitando así la fase de toma de imágenes a diferentes alturas aquí desarrollada, a costa de incrementar el número de cámaras necesario y la complejidad añadida.

Otra línea de investigación abierta en conexión con el sistema desarrollado consistiría en integrarlo con un planificador de agarre de nivel superior que, mediante el uso de sensores de fuerza permita mejorar la manipulación de forma global, alternando la fase de precontacto (información visual) con la fase de contacto (información táctil o de fuerza). Ya existe un proyecto de investigación en este sentido, donde se han dado ya los primeros pasos en la creación de un módulo de acomodación de la pinza al objeto utilizando sensores de fuerza [Cervera & del Pobil, 95a], [Cervera & del Pobil, 95b].

En la línea de los sistemas dinámicos (en movimiento) se podría enlazar el trabajo desarrollado con el tratamiento, mediante los denominados *snakes* [Blake, 95] que permiten la representación (mediante *B-splines*) de contornos que se adaptan de forma dinámica a los objetos en movimiento, abriendo así la posibilidad de desarrollar un sistema de agarre capaz de manipular objetos con movimiento relativo uniforme en cintas transportadoras o similar.

Precisamente en la línea de seguimiento de objetos en escenas 3D para su posible manipulación robotizada, se enmarca un ambicioso proyecto europeo: "*Intelligent object tracking and analysis using smart sensor technology (IOTA)*", dentro del programa ESPRIT, en el que además de la Univ. Jaume I, participan: "Sensomotoric Instruments" (Teltow, Alemania), "Logitech" (Suiza), "Integrated Vision Products" (Suecia), "Freie Universität Berlin" (Alemania), y "University of Strathclyde" (U.K.).

Para terminar, digamos que de las diferentes áreas de aplicación posibles del trabajo desarrollado destacamos tres que hacen uso intensivo de herramientas u objetos diseñados para la mano del hombre (universo de trabajo preferente utilizado en esta tesis):

- 1^a Industria de automoción, talleres, etc. donde se haga uso continuado de herramientas de todo tipo. Aquí el robot puede hacer de asistente, pasando herramientas al operario humano cuando éste está ocupado en reparaciones mecánicas etc., o procediendo a su ordenación final, cuando se requiera.
- 2^a Biomedicina. "Manipulación de instrumental quirúrgico" [Pozuelo, 88]. Esta tarea es imprescindible en intervenciones desarrolladas siempre en quirófanos y consiste en pasar la herramienta adecuada en el momento adecuado al cirujano en cuanto este lo solicita (pone la palma de la mano hacia arriba). En este caso se trata de un entorno

totalmente controlado en cuanto a condiciones de iluminación (lámparas de 100.000 lux diseñadas para evitar la generación de sombras), posicionamiento estructurado del instrumental (las herramientas se sitúan ordenadas en sus bandejas correspondientes), etc. Aquí la ventaja decisiva de colocar un robot es la necesidad de conseguir el mayor nivel de asepsia posible y en esto el hombre no puede competir con el robot.

- 3^a Robótica asistencial [Minor et al., 94]. Como ayuda a personas con discapacidades o personas mayores con dificultades psicomotoras (parkinson, etc.), se podrían aplicar variantes adaptadas del sistema aquí desarrollado como robots asistenciales en estación fija para puestos de trabajo o portátiles para ser montados en silla de ruedas.

Estas aplicaciones constituyen un reto en diferentes aspectos. En primer lugar requieren una interfaz amigable: el lenguaje natural a ser posible y, sobre todo en la segunda, gran eficiencia y fiabilidad de la implantación final. En un quirófano un error de funcionamiento puede costar una vida. En este último caso además, sustituir a un buen instrumentista está ahora mismo bastante alejado de la realidad tecnológica actual, pues éste tiende a adelantarse a la petición de instrumental del cirujano simplemente a la vista de lo que ocurre en cada momento en el transcurso de la operación, es decir, entra en juego la experiencia previa y toda una gama infinita de posibilidades que es capaz de desplegar la mente del hombre a partir del complejo proceso de percepción sensorial. Finalmente, la última aplicación reseñada (robótica asistencial), está siendo impulsada cada vez con mayor fuerza por los diferentes países del mundo occidental, visto que el impacto sobre la población potencial va rápidamente *in crescendo*.

Referencias

Bibliografía-Capítulo I

- [Albus & Evans, 76] Albus JS, Evans JM. *Robot Systems*. Scientific American. February, 1976.
- [Agin & Duda, 75] Agin GJ, Duda RO. *SRI Vision Research for Advanced Industrial Automation*. Second USA-JAPAN Computer Conference, Proceedings, pp. 113-117, Tokyo, Japan. 1975.
- [Ambler & Popplestone, 75] Ambler AP, Popplestone RJ. *Inferring the Positions of Bodies from Specified Spatial Relationships*. Artificial Intelligence, vol.6, pp.157-174. 1975.
- [Arkin & Grupen, 93] Arkin RC, Grupen R. "*Behavior-Based, Reactive Robotic Systems*". Tutorial for 13th IJCAI. Chambéry, France. 1993.
- [Brooks, 86] Brooks R. *A Robust Layered Control System for a Mobile Robot*. IEEE Jour. of Robotics and Automation, 2(1): 14-23. 1986.
- [Cervera & del Pobil, 95a] Cervera E, del Pobil AP. *Geometric reasoning for fine motion planning*. Proc. IEEE International Symposium on Assembly and Task Planning (ISATP'95), Pittsburg, Pennsylvania. 1995.
- [Cervera & del Pobil, 95b] Cervera E, del Pobil AP. *A hybrid qualitative-connectionist approach to robotic spatial planning*. Workshop on Spatial and Temporal Reasoning, Int. Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada. 1995.
- [del Pobil & Serna, 95] del Pobil AP, Serna MA. *Spatial Representation and Motion Planning*. Springer-Verlag, Berlin, 1995.
- [Gardner, 61] Gardner M. *More Mathematical Puzzles and Diversions*, 1961. Penguin 1966 edition.
- [Grimson & Lozano-Pérez, 87] Grimson WEL, Lozano-Pérez T. *Localizing overlapping parts by searching the interpretation tree*.

- IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-9(4): 469-482. 1987.
- [Groover & Mikell, 86] Groover MP, Mikell P. *Industrial Robotics*. MacGraw-Hill. 1986.
- [Inaba & Inoue, 90] Inaba M, Inoue H. *Vision-Based Robot Programming*. In "Robotics Research", pp. 129-136, ed. by Miura H and Arimoto S. The MIT Press. 1990.
- [Jarvis, 83] Jarvis RA. *Application-Oriented Robotic Vision-a Review*. Robotica, vol 2, pp 3-15. 1983.
- [Koutsou, 81] Koutsou A. *A Survey of Model-Based Robot Programming Languages*. Working paper No. 108, Department of Artificial Intelligence, Univ. of Edinburgh. 1981.
- [Kuniyoshi & Inoue, 93] Kuniyoshi Y, Inoue H. *Qualitative Recognition of Ongoing Human Action Sequences*. In 13th Int. Joint Conference on Artificial Intelligence, pp. 1600-1609. Chambéry, France. 1993.
- [Latombe, 79] Latombe JC. *Une Analyse Structurée D'outils de Programmation pour la Robotique Industrielle*. Proc. of the International Seminar on Programming Methods and Languages for Industrial Robots. IRIA, Rocquencourt, France. 1979
- [Lieberman & Wesley, 75] Lieberman LI, Wesley MA. *AUTOPASS: A Very High Level Programming Language for Mechanical Asembler Systems*. IBM Research Report RC-5599. August, 1975.
- [Lozano-Pérez, 82] Lozano-Pérez T. *Task Planning*. In Cap.6 of "Robot Motion: Planning and Control". The MIT Press. 1982.
- [Lozano-Pérez, 87] Lozano-Pérez T. *A simple motion planning algorithm for general robot manipulators*. IEEE Journal on Robotics and Automation, RA-3(3): 224-238. 1987.
- [Lozano-Pérez et al., 87] Lozano-Pérez T, Jones JL, Mazer E, O'Donnell PA, Tournassoud P, Lanusse P. *Handey: A task-level robot system*. Fourth International Symposium on Robotics Research. PP 29-36, Santa Cruz, California. 1987.
- [Lozano-Pérez et al., 92] Lozano-Pérez T, Jones JL, Mazer E, O'Donnell PA. *Handey: A robot task planner*. MIT Press Series in Artificial Intelligence. MIT Press. 1990.

- [Malcolm, 90] Malcolm C. *Behavioural modules: a new approach to robotic assembly*. Communication from author. Dept. of Artificial Intelligence, Univ. of Edinburgh, Scotland. 1990.
- [Mason & Salisbury, 85] Mason MT, Salisbury JK. *Robot Hands and the Mechanics of Manipulation*. The MIT Press. 1985
- [Mazer, 83] Mazer E. *LM-geo: Geometric programming of assembly robots*. A. Danthine & M. G eradin, editors. *Advanced Software in Robotics*, pp: 99-110. 1983.
- [Mitchell, 87] Mitchell T. *Becoming Increasingly Reactive*. Proc. of the AAAI, pp. 1051-1058. 1987.
- [Musliner, 96] Musliner D. *Workshop on Planning and Execution*. IEEE Int. Conf. on Robotics and Automation, Minneapolis, Minnesota. 1996.
- [Poplestone et al., 78] Poplestone RJ, Ambler AP, Bellos IM. *Rapt, a language for describing assemblies*. *Industrial Robotics*, 5(3): 131-137. 1978.
- [Poplestone et al., 80] Poplestone RJ, Ambler AP, Bellos IM. *An interpreter for a language for describing assemblies*. *Artificial Intelligence*, 14(1): 79-107. 1980.
- [Sanz, 90] Sanz PJ. *Sistema de Visi n Integrado en el Sistema Robotizado IRB 1000 de ASEA: Puesta a Punto y Aplicaci n en una C lula Flexible*. Tesina de MASTER de CAD / CAM. Univ. Polit cnica de Valencia. 1990.
- [Sanz, 94] Sanz PJ. *A little reflection about assembly planning based in vision*. Visiting Report, LIFIA-IMAG, Grenoble CEDEX, France. 1994.
- [Sanz et al., 96] Sanz PJ, Domingo J, Del Pobil AP, Pelechano J. *An Integrated Approach to Position a Robot Arm in a System for Planar Part Grasping*. *Advanced Manufacturing Forum*, Vol. 1, Special issue on Applications of Artificial Intelligence, (in press). 1996.
- [Van de Velde, 93] Van de Velde W. *Toward Learning Robots*. In "Toward Learning Robots", pp. 1-6, ed. by Van de Velde W. The MIT Press. 1993.
- [Winston, 93] Winston PH. *Inteligencia Artificial*. Addison-Wesley Iberoamericana. 1993.

- [Wolter et al., 85] Wolter JD, Volz RA, Woo AC. *Automatic Generation of Gripping Positions*. IEEE Transactions on Systems, Man, and Cybernetics, Vol.SMC-15, No. 2, pp. 204-213. 1985.

Bibliografía-Capítulo II

- [Albus, 81] Albus SJ. *Brains, Behaviour and Robotics*. Chapter 6 (A Neurological Model). BYTE Books, subs. of McGraw Hill. 1981.
- [Agin, 80] Agin GJ. *Computer Vision System for Industrial Inspection and Assembly*. IEEE Transactions on Computers, Vol 13, pp. 11-20. 1980.
- [Agin & Duda, 75] Agin GJ, Duda RO. *SRI Vision Research for Advanced Industrial Automation*. Second USA-JAPAN Computer Conference, Proceedings, pp. 113-117, Tokyo, Japan. 1975.
- [Arkin & Grupen, 93] Arkin RC, Grupen R. *Behavior-Based, Reactive Robotic Systems*. Tutorial for 1993 Int. Joint Conf. on A.I. Chambery, France. 1993.
- [Atkeson, 91] Atkeson CG. *Using Locally Weighted Regression for Robot Learning*. Proceedings of the IEEE International Conference on Robotics and Automation, Sacramento, CA, pp. 958-963, April 1991.
- [Bard et al., 95] Bard C, Laugier C, Milési-Bellier C, Troccaz J, Triggs Bill, Vercelli G. *Achieving Dextrous Grasping by Integrating Planning and Vision-Based Sensing*. The Int. Journal of Robotics Research, Vol.14, No. 5, pp. 445-464. 1995.
- [Blake, 95] Blake A. *A Symmetry Theory of Planar Grasp*. The Int. Journal of Robotics Research, Vol.14, No, pp. 425-444. 5. 1995.
- [Brady, 84] Brady M. *Artificial Intelligence and Robotics*. MIT, A.I. Memo 756. 1984.
- [Chen & Burdick, 92] Chen IM, Burdick JW. *Finding Antipodal Point Grasps on Irregularly Shaped Objects*. IEEE Proc. Robotics and Automation, pp. 2278-2283. May 1992.
- [Childress, 72] Childress DS. *Artificial hand mechanisms*. Mechanisms Conference and International Symposium on Gearing and Transmissions. 1972.

-
- [Churchland & Sejnowski, 92] Churchland PS, Sejnowski TJ. *The Computational Brain*. The MIT Press. 1992.
- [Clocksin & Moore, 89] Clocksin WF, Moore AW. *Experiments in Adaptive State-Space Robotics*. Proceedings of the 7th AISB Conference. Morgan Kaufman. 1989.
- [Conkie & Chongstitvatana, 90] Conkie A, Chongstitvatana P. *An Uncalibrated Stereo Visual Servo System*. Proceedings of the British Machine Vision Conference, Oxford, pp. 277-280, 1990.
- [Corke, 93] Corke PI. *Visual Control of Robot Manipulators - A Review*. In "Visual Servoing", pp. 1-31, Hashimoto Ed., World Scientific. 1993.
- [Cutkosky, 89] Cutkosky M. *On Grasp Choice, Grasp Models, and the Design of Hands for Manufacturing Tasks*. IEEE Transactions on Robotics and Automation, 5(3): 269-279. 1989.
- [Dario, 89] Dario P. *Tactile Sensing for Robots: Present and Future*. The Robotics Review 1. Khatib, Craig & Lozano-Pérez ed., pp.: 133-146. The MIT Press. 1989.
- [Domingo, 93] Domingo J. *Implantación y Comparación de diversos métodos para Control de Robots basados en Visión*. Tesis Doctoral. Facultad de Física, Valencia, 1993.
- [Domingo & Pelechano, 96] Domingo J, Pelechano J. *A Discrete Matrix-Network Formulation for the Visual Positioning of a Robot Arm*. Journal of Intelligent and Robotic Systems, (en prensa). 1996.
- [Dreyfus, 92] Dreyfus HL. *What Computers Still Can't Do*. Cap. 7: "The Role of Body in Intelligent Behavior". 1972, 1977, 1992.
- [Faverjon & Ponce, 91] Faverjon B, Ponce J. *On Computing Two-Finger Force-Closure Grasps of Curved 2D Objects*. IEEE Proc. Robotics and Automation, pp. 424-429. April 1991.
- [Fuentes et al., 94] Fuentes O, Marengoni HF, Nelson RC. *Vision-based Planning and Execution of Precision Grasps*. Technical Report 546, The University of Rochester, Computer Science Department, New York. 1994.
- [Gleason & Agin, 79] Gleason GJ, Agin GJ. *A Modular Vision System for Sensor Controlled Manipulation and Inspection*. Proc. of

- the Ninth International Symposium on Industrial Robots, pp. 57-70. 1979.
- [Groover & Mikell, 86] Groover MP, Mikell P. *Industrial Robotics*. MacGraw-Hill. 1986.
- [Hallam & Hayes, 92] Hallam B, Hayes G. *Comparing Robot and Animal Behaviour*. Dept. of A.I. (Univ. of Edinburgh), Research Paper No 598. 1992.
- [Hashimoto, 93] Hashimoto. *Visual Servoing*. Hashimoto Ed., World Scientific. 1993.
- [Hershkovitz et al., 95] Hershkovitz M, Tasch U, Teboulle M. *Toward a Formulation of the Human Grasping Quality Sense*. Journal of Robotics Systems 12(4), 249-256. 1995.
- [Horn, 86] Horn BKP. *Robot Vision*. The MIT Press. 1986.
- [Huber & Grupen, 94] Huber M & Grupen RA. 2-D Contact Detection and Localization Using Proprioceptive Information. IEEE Transactions on Robotics and Automation, Vol. 10, No. 1, pp. 23-33. 1994.
- [Iberall, 87] Iberall T. *Grasp Planning for Human Prehension*. International Conference on Artificial Intelligence. 1987.
- [Inaba & Inoue, 90] Inaba M, Inoue H. *Vision-Based Robot Programming*. In "Robotics Research", ed. by Miura H & Arimoto S, pp.129-136. 1990.
- [Jacobsen et al., 84] Jacobsen S, Wood J, Bigger K, Iverson E. The Utah/MIT hand: Work in progress. International Journal of Robotics Research, Vol.4, No.2, pp.21-50.
- [Jarvis, 88] Jarvis R.A. *Automatic Grip Site Detection for Robotics Manipulators*. Australian Computer Science Communications, Vol 10, No 1, pp. 346-356. 1988.
- [Kamon et al., 94] Kamon I., Flash T., Edelman S. *Learning to Grasp Using Visual Information*. Internal Report. Dept. of Applied Mathematics and Computer Science. The Weizmann Institute of Science, Israel, 1994.
- [Kamon et al., 96] Kamon I., Flash T., Edelman S. *Learning to Grasp Using Visual Information*. Proc. of the IEEE Int.Conf. on Robotics and Automation. Minneapolis, Minesota. Abril, 1996.

-
- [Klafter et al., 89] Klafter RD, Chmielewski, Negin. *Robotic Engineering: An Integrated Approach*. C.6, pp. 499-506, Prentice-Hall International Inc. 1989.
- [Krose et al., 93] Krose JA, Van der Smagt PP, Groen FCA. *Neural Networks in Robotics*. Bekey et al. eds. Chapter (A One Eyed Self Learning Robot Manipulator), pp. 19-27. Kluwer Academic Publishers. 1993.
- [Lozano-Pérez et al., 92] Lozano-Pérez T, Jones JL, Mazer E, O'Donnell PA. *Handey: A robot task planner*. MIT Press Series in Artificial Intelligence. MIT Press. 1990.
- [McCloy & Harris, 93] McCloy D, Harris DMJ. *Robótica una introducción*. Ed. Limusa. 1993.
- [Mackerrow, 93] Mackerrow J. *Introduction to Robotics*. Addison-Wesley. 1993.
- [Macmillan, 50] Macmillan. *The Cerebral CórteX of Man: A Clinical Study of Localization of Function*. 1950.
- [Markenscoff & Papadimitriou, 89] Markenscoff X, Papadimitriou CH. *Optimum Grip of a Polygon*. The International Journal of Robotics Research, Vol 8, No 2, pp. 17-29. April 1989.
- [Mason & Salisbury, 85] Mason MT, Salisbury JK. *Robot Hands and the Mechanics of Manipulation*. The MIT Press. 1985
- [Monkman, 95] Monkman GJ. *Robot Grippers for Use With Fibrous Materials*. The International Journal of Robotics Research, Vol 14, No 2, pp. 144-151. April 1995.
- [Montana, 91] Montana DJ. *The Condition for Contact Grasp Stability*. Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 412-417, Sacramento, California. April, 1991.
- [Murray et al., 94] Murray RM, Li Z. y Sastry S.S. *A Mathematical Introduction to Robotic Manipulation*. CRC Press. 1994.
- [Nguyen, 88] Nguyen VD. *Constructing Force Closure Grasps*. The International Journal of Robotics Research, Vol 7, No 3, pp. 3-16. June 1988.
- [Pertín-Troccaz, 89] Pertín-Troccaz J. *Grasping: A state of the art*. In O.Khatib, J.Craig, and T. Lozano-Pérez, editors, The Robotics Review 1, pages 71-98. MIT Press, Cambridge, Massachusetts. 1989.

- [Peshkin & Sanderson, 85] Peshkin MA, Sanderson AC. *Reachable Grasps on a Polygon: The Convex Rope Algorithm*. IEEE Journal of Robotics and Automation, Vol. RA-2, No. 1, pp. 53-58. March 1986.
- [Russell, 48] Russell B. *Human Knowledge. Its Scope and Limits*. George Allen & Unwin. 1948.
- [Salisbury & Craig, 82] Salisbury JK, Craig JJ. *Articulated Hands: Force Control and Kinematic Issues*. The International Journal of Robotics Research, Vol 1, No 1. 1982.
- [Sanz et al., 96] Sanz PJ, Domingo J, Del Pobil AP, Pelechano J. *An Integrated Approach to Position a Robot Arm in a System for Planar Part Grasping*. Advanced Manufacturing Forum, Vol. 1, Special issue on Applications of Artificial Intelligence, (in press). 1996.
- [Smith et al., 81] Smith V, Almenar L, Celma J, Ferrer M, Ferres E, Ganau A, et al. *Anatomía Aplicada y Biomecánica de los Miembros*. Gráficas Sol. 1981.
- [Stansfield, 88] Stansfield SA. *Reasoning about grasping*. Proceedings of the AAAI-88 Conference, pp. 768-773. Morgan Kaufmann Publishers. 1988.
- [Stansfield, 91] Stansfield SA. *Robotic Grasping of Unknown Objects: A Knowledge-based Approach*. The International Journal of Robotics Research, Vol 10, No 4, pp. 314-326. August 1991.
- [Tham & Prager, 92] Tham CK, Prager RW. *Reinforcement Learning for Multi-Linked Manipulator Control*. Technical Report CUED/F-INFENG/TR 104, Cambridge University Engineering Department, June, 1992.
- [Tsai, 86] Tsai RY. *An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision*. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, pp. 364-374. 1986.
- [Tsai, 87] Tsai RY. *A Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses*. IEEE Transactions on Robotics and Automation, Vol(3), pp. 323-344. 1987.

- [Trinkle et al., 88] Trinkle JC, Abel JM, Paul RP. *An Investigation of Frictionless Enveloping Grasping in the Plane*. The International Journal of Robotics Research, Vol 7, No 3, pp. 33-51. June 1988.
- [Weiss, 84] Weiss LE. *Dynamic Visual Servo Control of Robots: An Adaptive, Image-Based Approach*. Ph.D. thesis. Carnegie Mellon University. 1984.
- [Wiener, 48] Wiener N. *Cybernetics, or Control and Communication in the Animal and the Machine*. John Wiley, New York. 1948.
- [Wijesoma et al., 93] Wijesoma SW, Wolfe DFH, Richards RJ. *Eye-to-Hand Coordination for Vision-Guided Robot Control Applications*. The International Journal of Robotics Research, Vol 12, No 1, pp. 65-78. Feb 1993.
- [Wolter et al., 85] Wolter JD, Volz RA, Woo AC. *Automatic Generation of Gripping Positions*. IEEE Transactions on Systems, Man, and Cybernetics, Vol.SMC-15, No. 2, pp. 204-213. 1985.
- [Zhuang et al., 95] Zhuang H, *Simultaneous Calibration of a Robot and a Hand-Mounted Camera*. IEEE Transactions on Robotics and Automation, Vol 11, No.5. October, 1995.
- [Zucker, 81] Zucker SW. *Computer Vision and Human Perception*. Proc. of the 7th IJCAI, pp. 1102-1116. 1981.

Bibliografía-Capítulo III

- [Bard et al., 95] Bard C, Laugier C, Milési-Bellier C, Troccaz J, Triggs Bill, Vercelli G. *Achieving Dextrous Grasping by Integrating Planning and Vision-Based Sensing*. The Int. Journal of Robotics Research, Vol.14, No. 5, pp. 445-464. 1995.
- [Chen & Burdick, 92] Chen IM, Burdick JW. *Finding Antipodal Point Grasps on Irregularly Shaped Objects*. Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 2278-2283. Nice, France. May, 1992.
- [Coulomb, 1781] Coulomb CA. "Théorie des machines simples en ayant égard au frottement de leurs parties et à la roideur des cordages...". Mémoires de Mathématique et de Physique présentés à l'Académie Royale des Sciences, par divers savans. Paris, 1781.

- [Cutkoski & Wright, 86] Cutkoski MR, Wright PK. *Friction, Stability and the Design of Robotic Fingers*. The International Journal of Robotics Research. Vol.5, No 4, pp. 20-37. 1986.
- [del Pobil & Serna, 95] del Pobil AP, Serna MA. *Spatial Representation and Motion Planning*. Springer-Verlag, Berlin, 1995.
- [Davis, 90] Davis E. *Representations of Commonsense Knowledge*. Morgan Kaufmann Publishers. California. 1990.
- [Faverjon & Ponce, 91] Faverjon F, Ponce J. *On Computing Two-Finger Force Closure Grasps of Curved 2D Objects*. IEEE Int. Conf. on Robotics and Automation, pp. 424-429. Sacramento, CA. 1991.
- [Francois, 91] Francois C, Ikeuchi K, Hebert M. *A Three-Finger Gripper for Manipulation in Unstructured Environments*. Proceedings of the IEEE Int. Conf. on Robotics and Automation, pp. 2261-2266. Sacramento, California. 1991.
- [Howe & Cutkosky, 92] Howe RD & Cutkosky MR. *Touch Sensing for Robotic Manipulation and Recognition*. En "The Robotics Review 2", ed. Khatib, Craig & Lozano-Pérez, pp. 55-112. The MIT Press. 1992.
- [Kamon et al., 94] Kamon I, Flash T, Edelman S. *Learning to Grasp Using Visual Information*. Internal Report. Dept. of Applied Mathematics and Computer Science. The Weizmann Institute of Science. Israel. 1994
- [Kamon et al., 96] Kamon I., Flash T., Edelman S. *Learning to Grasp Using Visual Information*. Proc. of the IEEE Int. Conf. on Robotics and Automation. Minneapolis, Minesota. Abril, 1996.
- [Khatib et al., 92] Khatib O, Craig JJ, Lozano-Pérez T. *The Robotics Review 2*. The MIT Press. 1992.
- [Lakshminarayana, 78] Lakshminarayana K. *Mechanics of Form Closure*. ASME Paper 78-DET-32. 1978.
- [Laugier, 81] Laugier C. *A Program for Automatic Grasping of Objects with a Robot Arm*. Proc. 11th Int. Symp. Ind. Robots, pp. 287-294. 1981.
- [Lozano-Pérez et al., 87] Lozano-Pérez T, Jones JL, Mazer E, O'Donnell PA, Tournassoud P, Lanusse P. *Handey: A task-level robot system*. Fourth International Symposium on Robotics Research. PP 29-36, Santa Cruz, California. 1987.

-
- [Lozano-Pérez et al., 92] Lozano-Pérez T, Jones JL, Mazer E, O'Donnell PA. *Handey: A robot task planner*. MIT Press Series in Artificial Intelligence. MIT Press. 1990.
- [Markenscoff et al., 90] Markenscoff X, Ni L, Papadimitriou CH. *The Geometry of Grasping*. The International Journal of Robotics Research. Vol.9, No 1, pp. 61-74. February 1990.
- [Markenscoff & Papadimitriou, 89] Markenscoff X, Papadimitriou CH. *Optimum Grip of a Polygon*. The International Journal of Robotics Research. Vol.8, No 2, pp. 17-29. April, 1989.
- [Mason & Salisbury, 85] Mason MT, Salisbury JK. *Robot Hands and the Mechanics of Manipulation*. The MIT Press Series in Artificial Intelligence. Edited by P.H. Winston and M. Brady. 1985.
- [Montana, 91] Montana DJ. *The Condition for Contact Grasp Stability*. Proceedings of the IEEE Int. Conf. on Robotics and Automation, pp. 412-417. Sacramento, California. Abril 1991.
- [Murray et al., 94] Murray RM, Li Z, Sastry SS. *A Mathematical Introduction to Robotic Manipulation*. CRC Press. ISBN 0-8493-7981-4. 1994.
- [Nguyen, 88] Nguyen V-D. *Constructing Force-Closure Grasps*. The International Journal of Robotics Research. Vol.7, No 3, June 1988.
- [Opitz et al., 93] Opitz D, Bulthoff HH, Blake A. *Optimal Grasp Points: Computational Theory and Human Psychophysics*. Perception, pp. 22-123. 1993.
- [Pertin-Troccaz, 89] Pertin-Troccaz J. *Grasping: A state of the art*. En "The Robotics Review 1", ed. O.Khatib, J.Craig, and T. Lozano-Pérez, pp. 71-98. The MIT Press, Cambridge, Massachusetts. 1989.
- [Pertin-Troccaz, 92] Pertin-Troccaz J. *Task Level Robot Programming: On the HANDEY System*. En "The Robotics Review 2", ed. Khatib, Craig & Lozano-Pérez, pp. 31-36. The MIT Press. 1992.
- [Peshkin & Sanderson, 86] Peshkin MA, Sanderson AC. *Reachable Grasps on a Polygon: The Convex Rope Algorithm*. IEEE Journal of Robotics and Automation, Vol. RA-2, No. 1, pp. 53-58. March, 1986.

- [Rao & Goldberg, 95] Rao AS, Goldberg KY. *Friction and Part Curvature in Parallel-Jaw Grasping*. Journal of Robotic Systems 12(6), 365-381. 1995.
- [Salisbury & Roth, 83] Salisbury JK, Roth B. *Kinematic and force analysis of articulated mechanical hands*. J. Mech. Trans. Automat. Design ASME-105:35-41. 1983.
- [Wolter et al., 85] Wolter JD, Volz RA, Woo AC. *Automatic Generation of Gripping Positions*. IEEE Transactions on Systems, Man, and Cybernetics, Vol.SMC-15, No. 2, pp. 204-213. 1985.

Bibliografía-Capítulo IV

- [Aström & Wittenmark, 88] Aström KJ, Wittenmark B. *Sistemas Controlados por Computador*. Cap. 2: "Muestreo de Señales Continuas". Paraninfo. 1988.
- [Coster & Chermant, 89] Coster M, Chermant JL. *Precis D'Analyse D'Images*. Chap. XI: Erreurs en Analyse D'Images. Presses du CNRS. 1989.
- [Dai et al., 92] Dai M, Baylou P, Najim M. *An Efficient Algorithm for Computation of Shape Moments from Run-Length Codes or Chain Codes*. Pattern Recognition, vol. 25(10), pp. 1119-1128. 1992.
- [Davies, 90] Davies ER. *Machine vision: theory, algorithms, practicalities*. Academic Press Ltd. 1990.
- [Dorst & Smeulders, 87] Dorst L, Smeulders AWM. *Length estimators for digitized contours*. Comput. Vision, Graph., Image Process., 40: 311-33. 1987.
- [Duda & Hart, 73] Duda RO, Hart PE. *Pattern, clasiffication and scene analysis*. John Wiley & Sons. 1973.
- [Ferraté et al., 86] Ferraté G et al. *Robótica Industrial*. Marcombo, S.A. 1986.
- [Freeman, 61] Freeman H. *On the encoding of arbitrary geometric configurations*. IRE Trans. Electronic Computers, vol. EC-10(2): 260-268. 1961.

- [Fu et al., 88] Fu KS, Gonzalez RC, Lee CSG. *Robótica: Control, detección, visión e inteligencia*, pp. 388-392. Mc.Graw-Hill, Madrid. 1988.
- [Gonzalez & Wintz, 87] Gonzalez RC, Wintz P. *Digital Image Processing*. 2ª Ed., Addison-Wesley, Reading, Massachussets. 1987.
- [Groan & Verbeek, 78] Groan FCA, Verbeek PW. *Freeman-code probabilities of object boundary quantized contours*. Comput. Vision, Graphics, Image Processing, vol. 7. pp. 301-402. 1978.
- [Horn, 86] Horn BKP. *Robot Vision*. The MIT Press. 1986.
- [Hu, 62] Hu MK. *Visual Pattern Recognition by Moment Invariants*. IRE Trans. Information Theory, vol. IT-8: 179-87. 1962.
- [Imai & Iri, 86] Imai H, Iri M. *Computational Geometric Methods for Polygonal Approximations of a Curve*. Computer Vision, Graphics and Image Processing, vol. 36: 31-41. 1986.
- [Iñesta, 94] Iñesta JM. *Algoritmos de Visión Artificial y de Reconocimiento de Patrones para el Estudio Morfométrico del Eje Raquídeo Humano*. Tesis Doctoral. Facultad de Física, Univ. de Valencia, 1994.
- [Iñesta, 95] Iñesta JM, Buendía M, Sarti MA. *Some improvements to Apply High Curvature Point Detection on the Shape of Real Objects*. Technical Report DI 04-11/95. Depto. Informática. Univ. Jaume I, Noviembre 1995.
- [Jain, 89] Jain AK. *Fundamentals of Digital Image Processing*. Prentice Hall, Englewood Cliff. pp. 391-394. 1989.
- [Maravall, 93] Maravall D. *Reconocimiento de Formas y Visión Artificial*. RA-MA. 1993.
- [Pal & Pal, 93] Pal NK, Pal SK. *A review on image segmentation techniques*. Pattern Recognition, vol. 26(9): 1277-1294. 1993.
- [Pavlidis, 78] Pavlidis T. *A review of algorithms for shape analysis*. Comput. Graph. Imag. Processing, vol. 7: 243-258. 1978.
- [Pratt, 91] Pratt WK. *Digital Image Processing*. J. Wiley and Sons, New York. 1991.
- [Proffit & Rosen, 79] Proffit D, Rosen D. *Metrication errors and coding efficiency of chain-encoding schemes for the representation*

of lines and edges. Comp. Graph. Image Processing, 10: 318-32. 1979.

- [Rosenfeld & Johnston, 73] Rosenfeld A, Johnston E. *Angle detection on digital curves*. IEEE Transactions on Computers. Vol. C-22, pp.875-878. September, 1973.
- [Rosenfeld & Kak, 82] Rosenfeld A, Kak AC. *Digital Picture Processing*. Academic Press, Inc. 1982.
- [Saghri & Freeman, 81] Saghri JA, Freeman H. *Analysis of the Precision of Generalized Chain Codes for the Representation of Planar Curves*. IEEE Trans. Pattern Anal. Mach. Intell. vol. PAMI-3(5), pp. 533-539. 1981.
- [Sanz et al., 94] Sanz PJ, Iñesta JM, Buendia M, Sarti MA. *A Fast and Precise Way for Computation of Moments for Morphometry in Medical Images*. Proceedings of the V Int. Symposium on Biomedical Engineering, pp. 99-100. Santiago de Compostela, Spain, Sept. 1994.
- [Sanz et al., 96] Sanz PJ, Iñesta JM, del Pobil AP. "Towards an Automatic Determination of Grasping Points Through a Machine Vision Approach". In *Proc. of the Ninth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems (IEA/AIE)*, pp. 767-772. Fukuoka, Japan. 1996.
- [Toussaint, 85] Toussaint GT. *On the Complexity of Approximating Polygonal Curves in the Plane*. Proc. IASTED, International Symposium on Robotics and Automation. Lugano, Switzerland. 1985.
- [Weska, 78] Weska JS. *A survey of threshold selection techniques*. Comput. Graph. Image Process. Vol (7), pp. 259-265. 1978.
- [West & Rosin, 91] West GAW, Rosin PL. *Techniques for Segmenting Image Curves Into Meaningful Descriptions*. Pattern Recognition, vol. 24(7): 634-652. 1991.
- [Zakaria et al., 87] Zakaria MF, Vroomen LI, Zsombor-Murray PJA, van Kessel JMHM. *Fast Algorithm for the Computation of Moment Invariants*. Pattern Recog., 20(6): 639-643. 1987.
- [Zenzo, 83] Zenzo SD. *Advanced in image segmentation*. Image and Vision Computing, vol. 1(4): 196-210. 1983.

Bibliografía-Capítulo V

- [Bard et al., 95] Bard C, Laugier C, Milési-Bellier C, Troccaz J, Triggs Bill, Vercelli G. *Achieving Dextrous Grasping by Integrating Planning and Vision-Based Sensing*. The International Journal of Robotics Research, Vol.14, No. 5, pp. 445-464. October, 1995.
- [Blake, 95] Blake A. *A Symmetry Theory of Planar Grasp*. The International Journal of Robotics Research, Vol.14, No. 5, pp. 425-444. October, 1995.
- [Brady & Asada, 84] Brady M, Asada H. *Smoothed Local Symmetries and Their Implementation*. The International Journal of Robotics Research, Vol.3, No. 3, pp. 36-61. 1984.
- [Fuentes et al., 94] Fuentes O, Marengoni HF & Nelson RC. *Vision-based Planning and Execution of Precision Grasps*. Technical Report 546. Computer Science Department. The University of Rochester, New York. December, 1994.
- [Groan & Verbeek, 78] Groan FCA, Verbeek PW. *Freeman-Code Probabilities of Object Boundary Quantized Contours*. Computer Vision, Graphics, Image Processing, vol. 7, pp. 391-402. 1978.
- [Iñesta, 94] Iñesta JM. *Algoritmos de Visión Artificial y de Reconocimiento de Patrones para el Estudio Morfométrico del Eje Raquídeo Humano*. Tesis Doctoral. Facultad de Física, Univ. de Valencia, 1994.
- [Iñesta, 96] Iñesta JM, Buendía M, Sarti MA. *Some improvements to Apply High Curvature Point Detection on the Shape of Real Objects*. Technical Report DI 04-11/95. Depto. Informática. Univ. Jaume I, Noviembre 1995.
- [Jarvis, 88] Jarvis RA. *Automatic Grip Site Detection for Robotics Manipulators*. Australian Computer Science Communications, Vol 10, No 1, pp. 346-356. 1988.
- [Joshua, 91] Joshua SJ. *Symmetry principles and magnetic symmetry in solid state physics*. Adam Hilger. 1991.
- [Kamon et al., 96] Kamon I, Flash T, Edelman S. *Learning to Grasp Using Visual Information*. Proc. of the IEEE Int.Conf. on Robotics and Automation. Minneapolis, pp. 2470-2476. Minesota, April, 1996.

- [Ledermann, 85] Ledermann W. *Handbook of applicable mathematics*. Vol. V: Combinatorics and Geometry, Part B, chapt 11: "Symmetry". John Wiley. 1985.
- [Mackerrow, 93] Mackerrow J. *Introduction to Robotics*. Addison-Wesley. 1993.
- [Markenscoff et al., 90] Markenscoff X, Ni L, Papadimitriou CH. *The Geometry of Grasping*. The International Journal of Robotics Research. Vol.9, No 1, pp. 61-74. February 1990.
- [Mason & Salisbury, 85] Mason MT, Salisbury JK. *Robot Hands and the Mechanics of Manipulation*. The MIT Press Series in Artificial Intelligence. Edited by P.H. Winston and M. Brady. 1985.
- [Montana, 91] Montana DJ. *The Condition for Contact Grasp Stability*. Proceedings of the IEEE Int. Conf. on Robotics and Automation, pp. 412-417. Sacramento, California. Abril 1991.
- [Paul, 81] Paul RP. *Robot Manipulators: Mathematics, Programming, and Control*. MIT Press. 1981.
- [Pla, 93] Pla F. *Estudios de Técnicas de Análisis de Imagen en un Sistema de Visión para la Recolección Robotizada de Cítricos*. Tesis Doctoral. Facultad de Física, Univ. de Valencia, 1993.
- [Rosenfeld & Johnston, 73] Rosenfeld A, Johnston E. *Angle detection on digital curves*. IEEE Transactions on Computers. Vol. C-22, pp.875-878. September, 1973.
- [Sanz et al., 96] Sanz PJ, Domingo J, del Pobil AP, Pelechano J. *An Integrated Approach to Position a Robot Arm in a System for Planar Part Grasping*. Advanced Manufacturing Forum, Vol. 1, Special issue on Applications of Artificial Intelligence, (in press). 1996.
- [Sedgewick, 88] Sedgewick R. *Algorithms*. Addison Wesley, 1988.
- [Stansfield, 88] Stansfield SA. *Reasoning about grasping*. Proceedings of the AAAI-88 Conference, pp. 768-773. Morgan Kaufmann Publishers. 1988.
- [Stansfield, 91] Stansfield SA. *Robotic Grasping of Unknown Objects: A Knowledge-based Approach*. The International Journal of

- Robotics Research, Vol 10, No 4, pp. 314-326. August 1991.
- [Teh & Chin, 89] Teh C-H, Chin RT. *On the Detection of Dominant Points on Digital Curves*. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol II. No.8, pp. 859-872. August, 1989.
- [Van Gool et al. 95] Van Gool L, Moons T, Ungureanu D, Pauwels E. *Symmetry From Shape and Shape From Symmetry*. The International Journal of Robotics Research, Vol.14, No. 5, pp. 407-424. October, 1995.
- [Wolter et al., 85] Wolter JD, Volz RA, Woo AC. *Automatic Generation of Gripping Positions*. IEEE Transactions on Systems, Man, and Cybernetics, Vol.SMC-15, No. 2, pp. 204-213. 1985.
- [Zabrodsky, 93] Zabrodsky H. *Computational Aspects of Pattern Characterization-Continuous Symmetry*. PhD Thesis, Hebrew Univ., Jerusalem, Israel. 1993.
- [Zabrodsky et al., 95] Zabrodsky H. *Symmetry as a Continuous Feature*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol (17), No (12), pp. 1154-1166. 1995.

Bibliografía-Capítulo VI

- [Bentley & Papadimitriou, 80] Bentley JL, Papadimitriou CH. *Two Papers on Computational Geometry*. Technical Report CMU-CS-80-109, Dept. of Computer Science. Carnegie-Mellon University. 1980.
- [Churchland & Sejnowski, 92] Churchland PS, Sejnowski TJ. *The Computational Brain*. Capítulo 6: Sensorimotor Integration. Sejnowski TJ & Poggio TA, eds. The MIT Press. 1992.
- [Pellionisz & Llinás, 80] Pellionisz A, Llinás R. *Tensorial Approach to the Geometry of the Brain Function: Cerebellar Coordination via a Metric Tensor*. Neuroscience,5: 1125-1136. 1980.
- [Pellionisz, 85] Pellionisz A. *Adaptive mechanisms in gaze control*. Capítulo: Tensorial aspects of the multidimensional approach to the vestibulo-oculomotor reflex and gaze. Berthod & Melvill Jones, eds. 1985.

- [Sanz et al., 96] Sanz PJ, Domingo J, Del Pobil AP, Pelechano J. *An Integrated Approach to Position a Robot Arm in a System for Planar Part Grasping*. Advanced Manufacturing Forum, Vol. 1, Special issue on Applications of Artificial Intelligence, (in press). 1996.
- [Tsai, 86] Tsai RY. *An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision*. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, pp. 364-374. 1986.
- [UMI, 91] UMI. *RT 1000 User Manual*. UMI Group Ltd. Surrey, England. 1991.

Bibliografía-Capítulo VII

- [Blake, 95] Blake A. *A Symmetry Theory of Planar Grasp*. The International Journal of Robotics Research, Vol.14, No. 5, pp. 425-444. October, 1995.
- [Cervera & del Pobil, 95a] Cervera E, del Pobil AP. *Geometric reasoning for fine motion planning*. Proc. IEEE International Symposium on Assembly and Task Planning (ISATP'95), Pittsburg, Pennsylvania. 1995.
- [Cervera & del Pobil, 95b] Cervera E, del Pobil AP. *A hybrid qualitative-connectionist aproach to robotic spatial planning*. Workshop on Spatial and Temporal Reasoning, Int. Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada. 1995.
- [Domingo, 91] Domingo J. *Stereo part Mating*. Master's thesis, Department of Artificial Intelligence, Univ. of Edinburgh, Sept. 1991.
- [Domingo, 93] Domingo J. *Implantación y Comparación de diversos métodos para Control de Robots basados en Visión*. Tesis Doctoral. Facultad de Física, Valencia, 1993.
- [Kamon et al., 96] Kamon I, Flash T, Edelman S. *Learning to Grasp Using Visual Information*. Proc. of the IEEE Int.Conf. on Robotics and Automation. Minneapolis, pp. 2470-2476. Minesota, April, 1996.
- [Minor et al., 94] Minor A, Leija L, Hernandez P, Cancino A. *Robotic Assistance for Disabled*. Proceedings of the V Int.

Symposium on Biomedical Engineering, pp. 111-112.
Santiago de Compostela, Spain, Sept. 1994.

[Pozuelo, 88] Pozuelo S. *Enfermería de quirófano*. Ed. Toray, S.A. 1988.

Bibliografía-Apéndices

[Paul, 81] Paul RP. *Robot Manipulators: Mathematics, Programming, and Control*. MIT Press. 1981.

[Yoshikawa, 90] Yoshikawa T. *Foundations of Robotics: Analysis and Control*. The MIT Press. 1990.

[Mackerrow, 91] Mackerrow J. *Introduction to Robotics*. Addison Wesley. 1991.

[Angulo & Avilés, 89] Angulo JM, Avilés R. *Curso de Robótica*. Ed. Paraninfo. 1989.

[UMI, 91] UMI. *RT 1000 User Manual*. (The Gripper) pp.B-9, B-10, UMI Group Ltd. Surrey, England. 1991.

[Breen et al., 92] Breen EJ, Joss GH, Williams KL. *Dynamic arrays for fast, efficient, data manipulation during image analysis: a new software tool for exploratory data analysis*. Computer Methods and Programs in Biomedicine, N°37, pp. 85-92, Elsevier Science Publishers B.V. 1992.

[Ching Hong, 93] Ching Hong JS. Extraído de las *News* de Pascal (Newsgroups: comp.lang.pascal), de Internet. Dirección (e-mail): eng35799@leonis.nus.sg. 1993.

Apéndices

Apéndice-A: La función *atan2*

La función *atan2* fué introducida por primera vez en el contexto de la robótica por Paul [Paul, 81] en relación con el problema del modelado cinemático de los manipuladores. Dichos modelos cinemáticos, en especial el inverso, requieren la evaluación de funciones trascendentes (senos, cosenos, etc.) en sus ecuaciones correspondientes, las cuales pueden dar lugar a imprecisiones e indeterminaciones. La causa principal, tratada en detalle por algunos autores [Mackerrow, 91], reside en el hecho de que al evaluar una determinada orientación, ϕ , en el espacio, se utiliza una función con un único argumento (e.g. $\cos(-\phi) = \cos(\phi)$). Sin embargo, considerando la región: $-\pi < \phi < \pi$, a partir del conocimiento de $\sin(\phi)$ y $\cos(\phi)$, es obvio que existe una solución única para el valor del ángulo ϕ asociado. Precisamente la función *atan2* provee un medio de expresar dicha solución extendiendo la función $\text{atan}(\phi)$ a dos argumentos de entrada. En consecuencia, la función *atan2* es una función escalar de dos argumentos escalares tal que [Yoshikawa, 90],

$$\text{atan2}(a, b) = \arg(b + j a)$$

donde a y b son dos números reales, j es la unidad imaginaria, y \arg representa el argumento de un número complejo (véase la figura A.1).

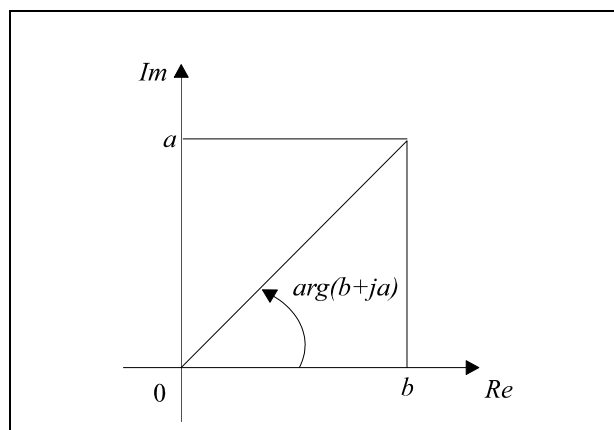


Figura A.1. Representación geométrica del argumento de un número complejo.

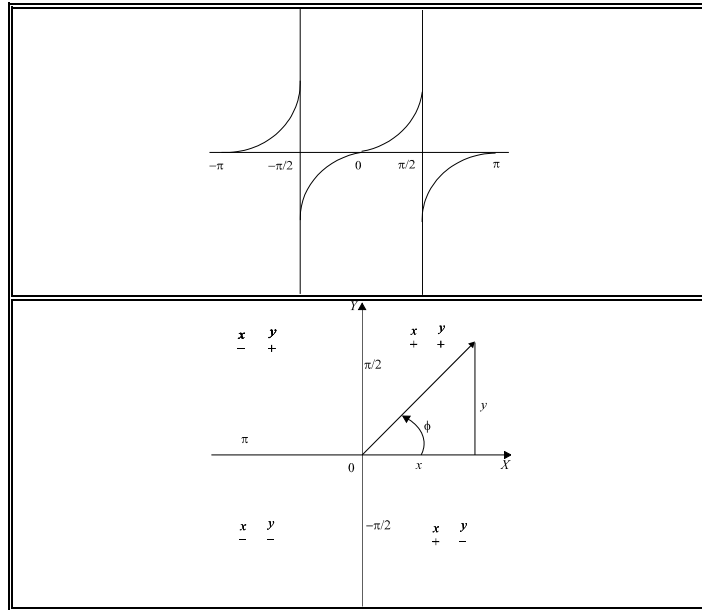


Figura A.2. (Abajo) valores posibles del argumento para $\text{atan2}(y,x)$.
(Arriba) representación de $\arctan(\phi)$.

Es obvio que, $\text{atan2}(\sin(\phi), \cos(\phi)) = \phi$, y que $\text{atan2}(k \sin(\phi), k \cos(\phi)) = \phi$ para cualquier escalar k . Pueden seguirse una serie de propiedades interesantes de esta función en el apéndice 1 de [Yoshikawa, 90] dedicado enteramente a la misma. Esta función devuelve valores angulares ϕ univaluados, tal que $-\pi < \phi < \pi$ (véase la figura A.2). A partir de las consideraciones anteriores se ha construido el siguiente algoritmo:

```

Función  $\text{atan2}(y,x: \text{Real})$ : Real;
Variables
  aux : Real;
Inicio
  Si  $(x = 0) \wedge (y = 0)$  Entonces  $\text{atan2} \ 0$ 
  Sino
    Si  $(\text{abs}(x) < \text{epsilon})$  Entonces
      Si  $(y > 0.0)$  Entonces  $\text{atan2} \ \pi / 2.0$ 
      Sino  $\text{atan2} \ -\pi / 2.0$ 
    Sino
      aux  $\arctan(y/x)$ ;
      Si  $x < 0.0$  Entonces
        Si  $y > 0.0$  Entonces  $\text{atan2} \ \text{aux} + \pi$ 
        Sino  $\text{atan2} \ \text{aux} - \pi$ ;
      FinSi;
       $\text{atan2} \ \text{aux}$ 
    FinSino;
  Fin;

```

Apéndice-B: La pinza del Robot "UMI RT 100"

En este apéndice se muestran algunos detalles técnicos en cuanto a los elementos físicos de interés relativos a los manipuladores utilizados. En la figura B.1 se observa la muñeca y la pinza de dos dedos plano-parallel perteneciente al robot UMI RT 100, utilizado en la implantación real descrita en el capítulo VI.

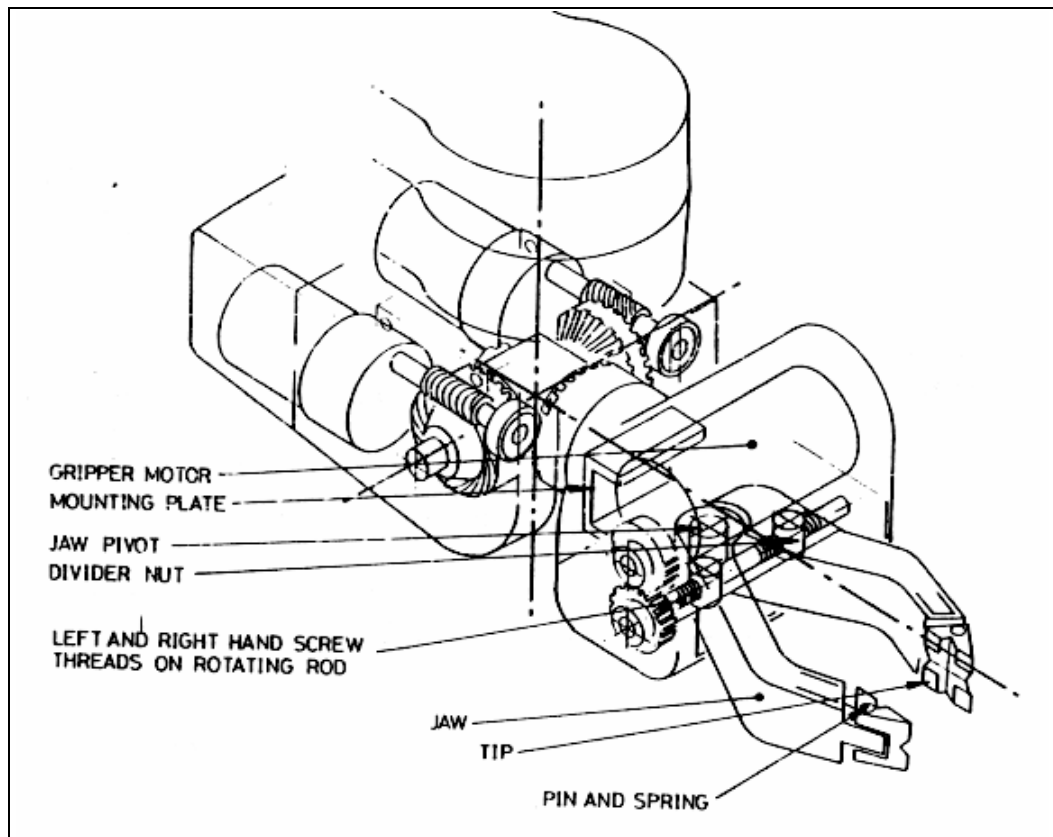


Figura B.1. Adaptada de [UMI, 91]. Se muestran en detalle las diferentes partes de la pinza y la muñeca de tres articulaciones (roll, pitch, yaw) donde se ensambla esta.

En la figura B.2, se representan gráficamente algunas disposiciones posibles de la pinza de dicho robot. En particular, es interesante el uso de los dedos curvos para el agarre de objetos cilíndricos. Con ello se intenta dotar de una cierta versatilidad las limitadas posibilidades de la pinza de dos dedos plano-parallel convencional.

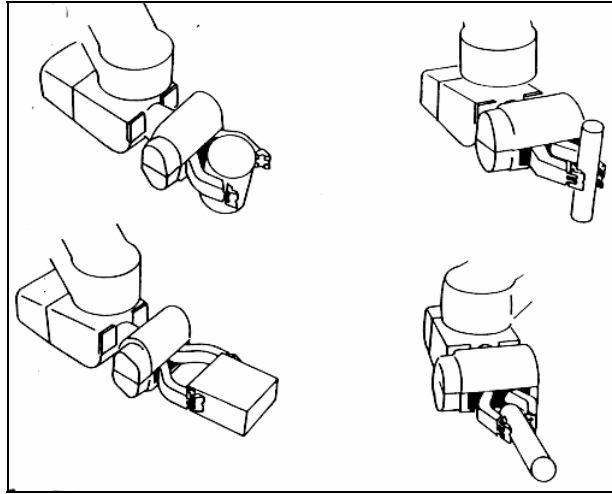


Figura B.2. Adaptada de [UMI, 91]. Se muestra la pinza en ejecución, actuando de manera distinta según el objeto y la localización del mismo respecto de la muñeca del manipulador.

Por último, en la figura B.3, se muestra la pinza del manipulador genéricamente conocido como MINIMOVER.

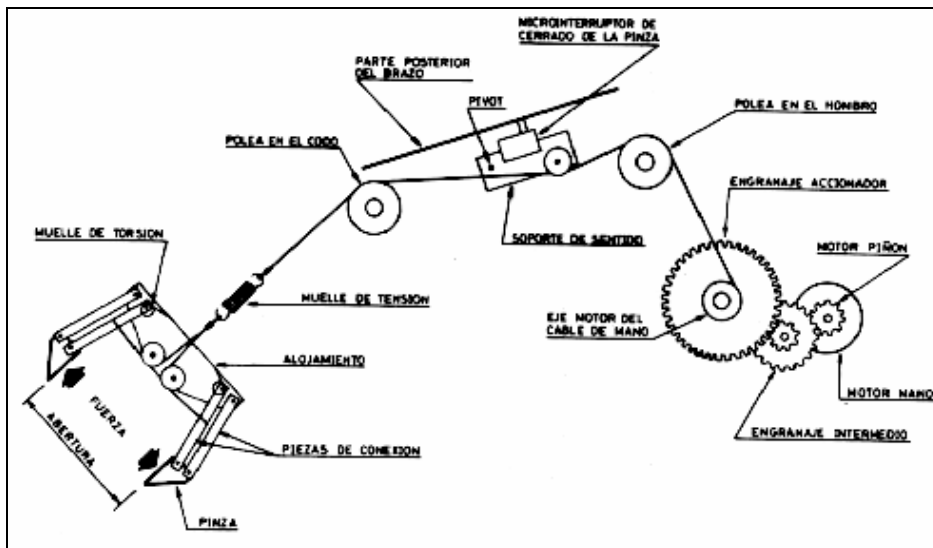

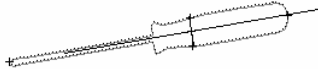
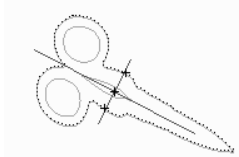


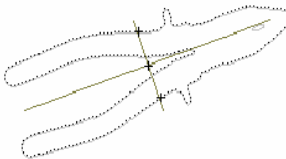
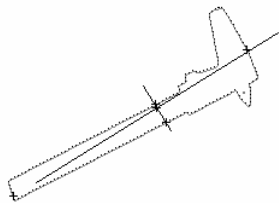
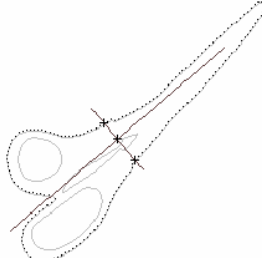
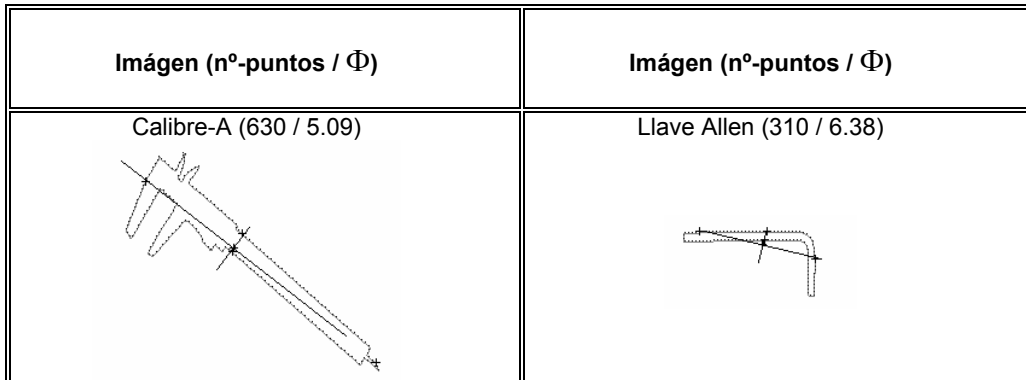
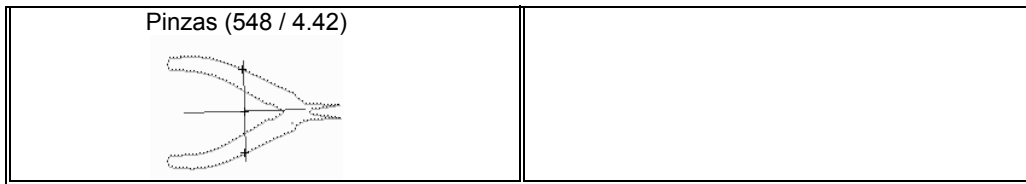


Figura B.3. Adaptada de [Angulo & Avilés, 89]. Modelo detallado de pinza, mecanismo de actuadores y transmisión para los manipuladores de tipo MINIMOVER. Inicialmente se utilizó este tipo de manipulador (CSS-113) para los primeros experimentos.

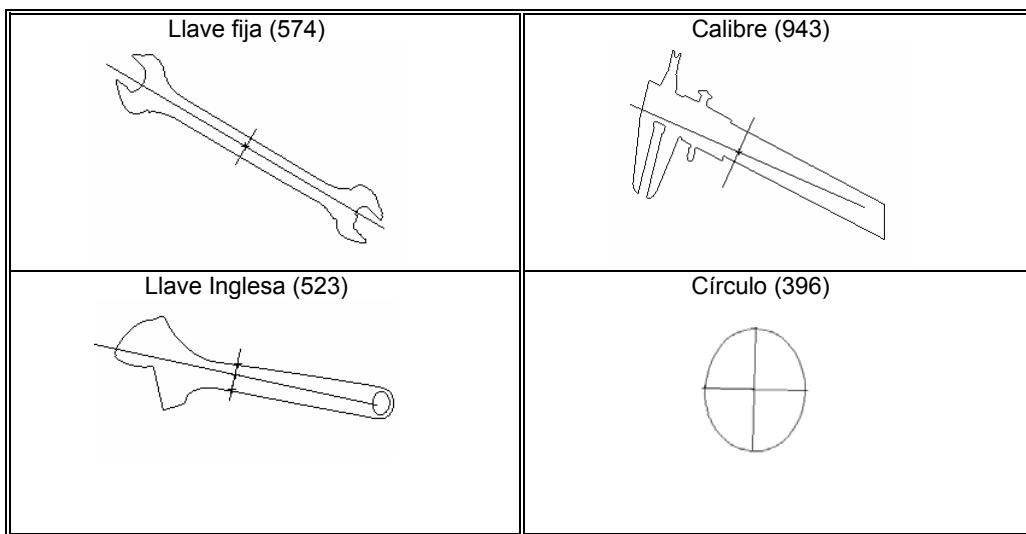
Apéndice-C: Una muestra del universo de trabajo

A lo largo del presente trabajo se han procesado multitud de objetos, dando a continuación una muestra representativa de los mismos. En primer lugar se ofrece una tabla con las once imágenes representativas del experimento mostrado en 5.4.1, en orden decreciente (de izquierda a derecha y de arriba abajo) respecto de la deficiencia simétrica global, Φ . Puntualicemos que estas imágenes han sido tomadas con los mismos parámetros de cámara para dar una idea de su tamaño relativo (se da el número de puntos en todas ellas) además de su morfología.

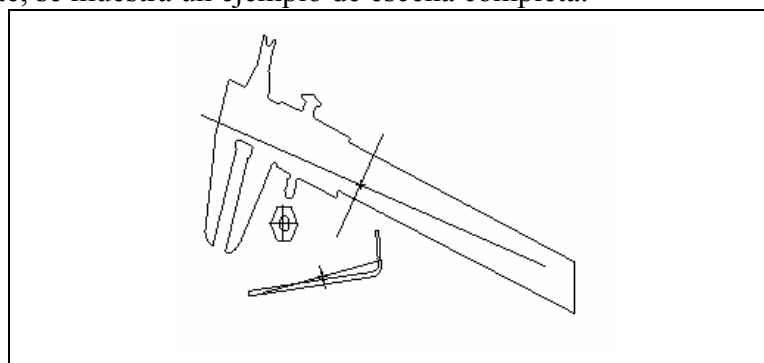
Imagen (nº-puntos / Φ)	Imagen (nº-puntos / Φ)
<p>Alicates (703 / 0.87)</p> 	<p>Destornillador (493 / 1.58)</p> 
<p>Tijeras1 (415 / 1.59)</p> 	<p>Tornillo (245 / 1.69)</p> 
<p>Tuerca (119 / 1.86)</p> 	<p>Tenazas (785 / 2.68)</p> 
<p>Calibre-B (538 / 3.08)</p> 	<p>Tijeras2 (544 / 3.2)</p> 



A continuación se muestra otra tabla con objetos manejados, de forma aislada, en otros experimentos del trabajo.



Finalmente, se muestra un ejemplo de escena completa:



Apéndice-D: Acerca de los vectores dinámicos

Tal y como se comentó en el Capítulo IV, a partir de la estructura de lista dinámica utilizada inicialmente para representar los contornos de los objetos, se crea un vector dinámico para facilitar el razonamiento geométrico, aplicando simultáneamente una reducción de datos, mediante el método, explicado en dicho Capítulo, del 3-muestreo.

Sobre el concepto de vector dinámico es interesante el trabajo de [Breen EJ et al, 92], que consideran un vector que se expande a medida que se insertan nuevos elementos. Esta expansión se controla mediante un factor de modulación, cuyo tamaño ajustan de modo empírico.

No olvidemos que la implementación de los algoritmos ha sido realizada en Turbo Pascal y, no hace mucho tiempo en la red Internet hubo un acalorado cruce de opiniones sobre la posibilidad o no de construir vectores dinámicos con dicho compilador. Al final, la respuesta fué positiva y prueba de ello es que este trabajo lo utiliza con éxito. Sin duda, el vencedor de la confrontación mencionada fué Ching Hong [Ching Hong JS, 93], de la *National University of Singapore*, el cuál ofreció el código a utilizar para la construcción de vectores dinámicos con Turbo Pascal. A partir del mismo fué relativamente sencillo, adaptarlo a nuestro problema construyendo un módulo ("Unit") con las operaciones necesarias para trabajar con ellos.

UNIT ADinam;

INTERFACE

Type

```
Punto = Record
    x,y:Integer;
End;
Rec = Record
    a:Punto;
    b:Byte;
End;
TElemento = Rec;
```

```
Procedure CreaDArray(VAR Dynamic:PArray; Talla:Integer);
{ Crea un Array Dinámico de tamaño "Talla"}
```

```
Procedure SupDArray(VAR Dynamic:PArray; Talla:Integer);
{ Suprime un Array Dinámico de tamaño "Talla"}
```

```
Procedure IniDArray(VAR Dynamic:PArray; Talla:Integer);
{ Inicializa a "0" un Array Dinámico de tamaño "Talla"}
```

IMPLEMENTATION

```
Procedure CreaDArray(VAR Dynamic:PArray; Talla:Integer);  
BEGIN  
  IF MaxAvail < SizeOf(TElemento) * talla THEN  
    OutTextXY(10,10,'Not enough memory')  
  else  
    { Allocate memory on heap }  
    GetMem(Dynamic,SizeOf(TElemento)*talla);  
  
  END;  
  
Procedure SupDArray(VAR Dynamic:PArray; Talla:Integer);  
BEGIN  
  FreeMem(Dynamic,SizeOf(TElemento)*talla);  
END;  
  
Procedure IniDArray(VAR Dynamic:PArray; Talla:Integer);  
VAR  
  i:Integer;  
BEGIN  
  For i:=1 TO Talla DO  
    Begin  
      Dynamic^[i].a.x:= 0;  
      Dynamic^[i].a.y:= 0;  
      Dynamic^[i].b := 0;  
    End;  
  
  END;  
  
End. { Fin Unit ADinam }
```
