

Un enfoque algorítmico para enseñar “Visión por Computador” en las titulaciones de Informática

A. B. Moreno, A. Sánchez, J. Vélez

Dept. de Ciencias Experimentales e Ingeniería

Universidad Rey Juan Carlos

28933 Móstoles (Madrid), Spain

e-mail: a.b.moreno@escet.urjc.es, an.sanchez@escet.urjc.es, j.velez@escet.urjc.es

Resumen

Este artículo describe el enfoque y desarrollo de un curso sobre “Visión por Computador”, impartido en el tercer curso de la titulación Ingeniería Técnica en Informática de Sistemas de la Universidad Rey Juan Carlos de Madrid. El artículo refleja nuestro punto de vista y experiencia sobre cómo enseñar esta materia interdisciplinar a los alumnos de titulaciones informáticas, con vistas a una mejor comprensión de las operaciones habituales sobre imágenes así como de los algoritmos más apropiados para implementarlas. Las aplicaciones de “Tratamiento de Imágenes” y de “Visión por Computador” han adquirido una gran importancia por sus múltiples usos en entornos industriales. Al mismo tiempo, ofrecen a los estudiantes la posibilidad de utilizar sus conocimientos sobre algoritmos para abordarlas. Mostrar la relación entre lo aprendido en cursos previos sobre “Algoritmos y Estructuras de Datos” con los diversos problemas de “Visión por Computador” estudiados en esta asignatura, contribuye a que el curso resulte más interesante para los alumnos.

1. Introducción

En los cursos introductorios sobre “Tratamiento Digital de Imágenes” y “Visión por Computador” se estudian los procesos sistemáticos que, aplicados sobre una imagen digital producen otra imagen modificada más apropiada para cierto propósito (por ejemplo,

para ser visualizada, analizada, comprimida, etc.). Se trata de un área con múltiples aplicaciones en diversas disciplinas tanto científicas (biología, astronomía, meteorología, medicina...) como industriales (robótica, procesamiento de documentos, arte, biometría...)[5]. Como en otras áreas científicas, los problemas de “Visión por Computador” se pueden resolver de una manera metódica y estructurada. El análisis y el diseño eficiente de los algoritmos usados en las aplicaciones que trabajan sobre imágenes son componentes críticos en toda metodología de procesamiento de imágenes. Dado el elevado número de aplicaciones que manipulan imágenes, resulta apropiado para un estudiante de Ingeniería Informática, conseguir conocimientos prácticos sobre las operaciones más importantes de tratamiento de imágenes y sobre la manera eficiente de implementar dichas operaciones. Estos conocimientos se pueden utilizar y ampliar en materias afines, por ejemplo, en cursos más avanzados de “Visión Artificial” o de “Robótica”.

El enfoque propuesto en este trabajo se ha aplicado a la asignatura “Visión Computacional” de Ingeniería Técnica en Informática de Sistemas de la Universidad Rey Juan Carlos. Esta asignatura es optativa, y se imparte en el tercer curso de la titulación durante un cuatrimestre (14 semanas) con tres horas lectivas por semana. El curso está estructurado en clases teóricas y prácticas. La teoría debe ser aprobada mediante un examen escrito. La parte práctica está organizada en dos partes principales: (1) una práctica de tratamiento de

imágenes que cada alumno realiza de forma individual, y (2) un proyecto consistente en una aplicación de “Visión por Computador” para ser realizado por grupos de tres alumnos. Para las prácticas se requiere el conocimiento de un lenguaje de programación estructurada de alto nivel (preferentemente Pascal o C). Dicho conocimiento se consigue cursando otras asignaturas troncales como “Metodología de la Programación” y “Estructuras de Datos y de la Información”, impartidas respectivamente en los cursos primero y segundo de la carrera. Para la realización de los proyectos propuestos, es posible usar las librerías de software disponibles para el curso: MATLAB [9] y la biblioteca de funciones MIL de la tarjeta digitalizadora Meteor 2/4 Matrox [10] para prototipado y desarrollo rápido de programas.

A continuación se describe la estructura del resto del artículo. La sección 2 resume nuestras ideas generales sobre la enseñanza de la “Visión Computacional” en titulaciones informáticas. La sección 3 describe los elementos de teoría del curso. En la sección 4 se presentan varios ejemplos de prácticas propuestas en el curso. En la sección 5 se describen algunas propuestas de proyectos en grupo sobre aplicaciones reales de visión por computador, donde se pone énfasis en los requerimientos de corrección y eficiencia de las soluciones programadas. Finalmente, la sección 6 resume las conclusiones extraídas de nuestra experiencia.

2. Reflexiones sobre la enseñanza de “Visión por Computador” en titulaciones informáticas.

Como hemos indicado, la “Visión por Computador” es una *materia interdisciplinar*. Está relacionada con la informática, las matemáticas y la ingeniería, y también con otras áreas donde se puede aplicar (por ejemplo, la medicina o la geología). Este carácter interdisciplinar convierte a la asignatura en atractiva y de gran utilidad profesional pero, por otro lado, hace difícil el diseño de cursos basados en tratamiento de imágenes y visión computacional. En consecuencia, el profesorado debe optar inevitablemente por diferentes

enfoques, por lo que los simples objetivos de la asignatura pueden variar muchísimo. En nuestro caso, al encontrarnos en una titulación de informática y tratarse además de un primer curso en el área de la “Visión por Computador”, nuestro planteamiento consiste principalmente en *introducir los problemas y técnicas del tratamiento digital de imágenes y visión artificial desde una perspectiva algorítmica*. Otras consideraciones a tener en cuenta al planificar los contenidos y la orientación del curso son las siguientes:

□ *La visión computacional es una materia “en auge”.*

Actualmente muchas empresas tienen una opinión bastante positiva sobre el futuro de la tecnología del tratamiento de imágenes. Su importancia está no solamente en el reconocimiento de objetos y en la interpretación de imágenes, sino en otros campos como: la realidad virtual, Internet y las bases de datos de imágenes.

□ *Los estudiantes necesitan algo más que teoría.*

Aunque existe una parte teórica de la visión computacional que está bien definida, se necesita imperativamente un trabajo de laboratorio y de proyectos de visión con una orientación sobre los sistemas. Hacen falta prácticas cortas para familiarizar a los alumnos con cámaras, tarjetas digitalizadoras, etc. Se necesita dedicar tiempo a explicar a los alumnos proyectos reales que se puedan resolver (así como su solución) usando técnicas de “Visión por Computador”. En las soluciones es importante hacer énfasis en los aspectos de eficiencia y tiempo de respuesta del sistema implementado.

□ *¿Qué enseñar?*

Este curso suele ofertarse como materia optativa en los últimos años de titulaciones informáticas (ingenierías técnicas, ingeniería superiores o licenciaturas). La manera tradicional de ofrecer la materia es mediante clases magistrales de teoría y la realización de prácticas de laboratorio. En nuestra propuesta, además de considerarse el enfoque tradicional, se aporta como novedad el uso de técnicas de diseño de algoritmos y de otros elementos

algorítmicos para analizar y resolver problemas relacionados con los contenidos de la asignatura.

□ ¿Cómo enseñar?

Al impartir cursos relacionados con esta materia hay que usar presentaciones interactivas que permitan la visualización del problema y ofrecer prácticas que tuviesen relación con problemas reales de visión artificial.

3. Elementos de teoría del curso

Como se ha señalado, este curso está orientado a estudiantes de Ingeniería Técnica en Informática, durante un cuatrimestre y tiene asignadas tres horas lectivas semanales (de las cuales se destina aproximadamente el 50% del tiempo a teoría y a prácticas). Aunque nuestro enfoque es bastante práctico, también consideramos necesario explicar los fundamentos teóricos básicos del tratamiento de imágenes digitales y la visión por computador. Durante el curso se utilizan varios textos, aunque no encontramos ninguno “ideal” para el curso completo. El libro más utilizado ha sido González y Woods [4]. Otros libros de texto complementarios son Jain et al. [6], Baxes [1], y Maravall [8]. A continuación, se muestra un esbozo de los temas cubiertos en las clases de teoría junto con el número de semanas dedicadas a cada tema (entre paréntesis):

1. Introducción: problemas y aplicaciones (1)
2. Adquisición de imágenes digitales (1)
3. Muestreo y cuantización de imágenes (1)
4. Realce y restauración de imágenes (3)
 - operaciones de tratamiento a nivel de píxeles y de regiones,
 - transformaciones geométricas sobre imágenes,
 - tratamiento en el dominio de la frecuencia.
5. Análisis de imágenes (3):
 - segmentación,
 - extracción de características,
 - reconocimiento de formas.

6. Compresión de imágenes (2):

- métodos de compresión con y sin pérdida.

7. Aspectos avanzados (3):

- morfología matemática,
- introducción a la visión tridimensional,
- visualización de datos.

Al explicar el temario del curso, nos podemos beneficiar del conocimiento recibido por los alumnos en cursos previos como los de “Metodología de la Programación” y “Estructuras de Datos y de la Información”. Por ejemplo, las técnicas de diseño de algoritmos se podrían relacionar con ejemplos de algoritmos representativos explicados en nuestro curso de “Visión Computacional”. Pensamos que esta integración, cuando se pueda llevar a la práctica, resulta ventajosa para el aprendizaje de las diferentes asignaturas interrelacionadas. A modo aclaratorio, la Tabla 1 muestra algunos ejemplos sobre cómo integrar las técnicas de diseño de algoritmos en nuestro curso.

Cuando las diferentes operaciones sobre imágenes se explican en las clases teóricas, se motiva a los alumnos con ejemplos prácticos y se muestra el carácter interdisciplinar de la asignatura mediante diversos ejemplos de aplicaciones industriales reales. Algunos de estos ejemplos simplificados se han utilizado como proyectos, para ser realizados por grupos de alumnos al final del curso.

Nosotros consideramos importante recalcar los aspectos de eficiencia de los algoritmos de imágenes. Tal vez, un algoritmo inicial diseñado usando una técnica particular no es aplicable directamente en la práctica. Por ejemplo, esto ocurre cuando se usa “backtracking” con objeto de implementar un método de etiquetado de regiones conexas en una imagen digital. En este caso, se identificará y explicará un algoritmo más eficiente, basado en el uso de la programación dinámica. Es útil para los estudiantes aplicar esta estrategia de “refinamiento” de la solución para descubrir el mejor algoritmo con respecto a los requerimientos de tiempo de ejecución y memoria.

Técnica de diseño algorítmica	Ejemplo de su uso en la asignatura
Divide-y-vencerás	Filtro de mediana (realzado) <i>Split-and-merge</i> (segmentación)
Técnica voraz	Códigos de Huffman (compresión) Árbol de recubrimiento (clasificación)
Backtracking	Componentes conexas (reconocimiento) Crecimiento de regiones (segmentación)
Programación dinámica	Contorno óptimo (extracción de características) Detección de líneas (extracción de características)
Transformación del dominio	FFT (filtrado en el dominio frecuencial) Transformada de Hough (segmentación)

Tabla 1. Técnicas de diseño de algoritmos asociadas a distintos ejemplos de procesamiento de imágenes.

4. Prácticas del curso

El curso requiere ser complementado con prácticas de laboratorio y con proyectos prácticos. En nuestra universidad, este curso se impartió por primera vez entre Octubre de 1999 y Febrero de 2000. Debido al reducido número de estudiantes (aproximadamente veinte siguieron el curso), los recursos de hardware y software disponibles eran reducidos. Estos han consistido en cuatro PC (Pentium III) cada uno con una tarjeta digitalizadora Matrox, dos cámaras de vídeo convencionales, MATLAB y el software de digitalización y tratamiento de imágenes Matrox Imaging Libraries (MIL).

Las prácticas de laboratorio consisten en la realización de una o varias operaciones básicas, descritas previamente en las clases de teoría, sobre imágenes digitales. Tras la descripción en profundidad de una clase de operaciones sobre imágenes y sus algoritmos asociados, el énfasis se pone en ayudar a los alumnos a descubrir cómo implementar estas operaciones. Es muy importante recalcar la eficiencia de los algoritmos de imágenes [7]. Tal vez un algoritmo diseñado inicialmente, de forma menos eficiente, usando una técnica particular no sea aplicable en la práctica. En este punto, se explicará un nuevo algoritmo más apropiado

para la resolución del mismo problema. Nosotros consideramos útil que los estudiantes aprendan “por refinamiento” a identificar un mejor algoritmo en relación con los requerimientos de tiempo y memoria.

Estas prácticas propuestas son individuales y diferentes para cada alumno. Cada enunciado de práctica contiene una descripción corta de su propósito, los datos de entrada, los resultados que debe proporcionar, y en algunos casos la(s) técnica(s) de diseño de algoritmo(s) que se han de utilizar con objeto de tener una implementación más eficiente. Muchas de las prácticas están orientadas a una técnica de diseño específica (por ejemplo, “divide y vencerás”, búsqueda con retroceso, etc.), y los estudiantes debían explicar también por qué la técnica elegida resulta aplicable en relación con la eficiencia de la solución. De las prácticas propuestas en el curso, resumimos tres a continuación.

□ Componentes conexas de una imagen:

Una de las operaciones más comunes en tratamiento de imágenes digitales consiste en encontrar las componentes conexas en una imagen [5]. Por ejemplo, en tareas de segmentación, los puntos (píxeles) en una componente conexa forman una región

candidata para representar un objeto (o parte de él) en una imagen. Un algoritmo de etiquetado de componentes conexas encuentra todas las componentes de una imagen y asigna una única etiqueta a todos los puntos que están en la misma componente. En esta práctica, se pide a los alumnos que diseñen e implementen un algoritmo recursivo para el problema propuesto siguiendo el esquema de “backtracking” o búsqueda con retroceso [2], dado un píxel de comienzo. También se pide a los estudiantes que identifiquen otro algoritmo equivalente y más eficiente (no recursivo) usando una estructura de datos auxiliar (árbol cuaternario o “quadtree”) para almacenar los vecinos de cada píxel. La Figura 1 muestra una imagen binaria y su correspondiente representación usando un “quadtree”.

□ *Filtro de mediana:*

El filtro de mediana es un filtro de rango, que sustituye cada píxel de una imagen por la mediana de los valores de gris en su vecindad local [1]. Si la vecindad de un píxel se considera de tamaño 3×3 , esta región se puede convertir en un vector, que se ordena primero por valores de intensidad no decreciente, y luego se selecciona el valor de intensidad situado en la posición mediana de dicho vector. Se pide a los estudiantes que diseñen un algoritmo de ordenación eficiente que siga la técnica “divide y vencerás” [2] para obtener la mediana, y luego se aplica dicho filtro a toda la imagen original para obtener una nueva imagen filtrada.

□ *Códigos de Huffman:*

El objetivo de esta práctica es construir la

representación comprimida de una imagen mediante la técnica de los códigos de Huffman [4]. La compresión de Huffman [3] de una imagen es una técnica de codificación sin pérdida. Los valores de intensidad cada píxel son sustituidos por códigos de longitud variable basados en la frecuencia de ocurrencia para cada valor de intensidad en la imagen dada. Una tarea inicial para los estudiantes consiste en obtener el histograma de la imagen con las frecuencias de ocurrencia de cada valor de intensidad en la imagen. Se sugiere, a continuación, un algoritmo voraz [2][3] que permita asignar los códigos más cortos a los valores de intensidad con ocurrencias más frecuentes. Finalmente, hay que realizar un análisis de la eficiencia del algoritmo desarrollado.

5. Proyectos del curso

En esta sección se describen de forma simplificada algunos proyectos propuestos que deben ser realizados por los alumnos, en grupos de tres personas, asignándose a cada grupo de trabajo un proyecto diferente. El principal objetivo de estos proyectos, es abordar la realización de aplicaciones reales simplificadas. Estos proyectos en grupo son una excelente oportunidad para que los alumnos intercambien conocimientos de la asignatura con sus compañeros y practiquen las técnicas y algoritmos estudiados durante el curso. También hemos recalcado que en las aplicaciones industriales el tiempo de cálculo es un factor crítico [7]. Para motivar a los estudiantes sobre la importancia de implementar programas

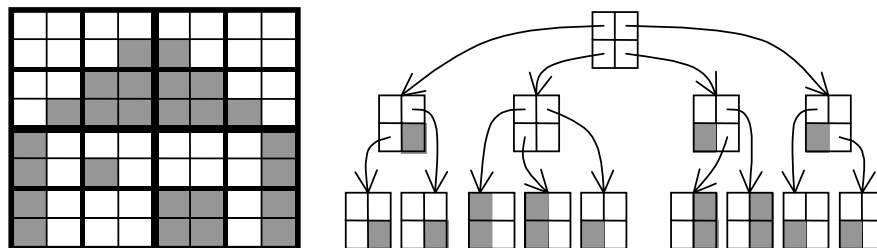


Figura 1. Ejemplo de una imagen binaria y su correspondiente representación “quadtree” para segmentarla.

eficientes, se les pide algún tipo de medida de los tiempos de ejecución. En particular, cada grupo de alumnos debe estimar la complejidad computacional de los algoritmos de imágenes involucrados en su proyecto particular. Además, el tiempo de ejecución de los programas desarrollados se debe calcular (usando una función de tiempo, como el comando UNIX `time` o el más sofisticado `gethrtime`). De los proyectos propuestos resumimos brevemente dos de ellos a continuación.

□ *Reconstrucción morfológica de imágenes de códigos de barras:*

Leer un código de barras de una imagen que ha sido escaneada previamente es un problema bien conocido en la práctica. Dicho problema lleva asociados otros debidos a la técnica no especificada usada por los escáneres para capturar las imágenes. Generalmente, éstos capturan las imágenes en niveles de gris, mientras que los lectores de códigos de barras trabajan sobre imágenes binarias (monocromas, o en blanco y negro). La conversión de una imagen en niveles de gris a una imagen binaria se denomina umbralización. Los escáneres avanzados usan procedimientos de umbralización regional para tal operación. Dichos métodos realzan la calidad visual de las imágenes eliminando zonas oscuras homogéneas. Sin embargo, tienen efectos perniciosos sobre los códigos de barras, precisamente porque los códigos de barras son zonas oscuras homogéneas (como se observa en la Figura 2).

Para el realce de la región de la imagen que contiene el código de barras, los alumnos han de proponer e implementar un procedimiento de reconstrucción. Primero, el sistema debe encontrar la región de la imagen que contiene el código de barras (por ejemplo usando un algoritmo de “*split-and-merge*”). Luego, se aplica un procedimiento de reconstrucción (por ejemplo basado en la operación de dilatación morfológica). El tiempo total de respuesta debe ser menor que un segundo por imagen en un PC estándar.

□ *Extracción automática de ayudas visuales en imágenes de pistas de aterrizaje:*

Un sistema que realiza extracción automática de objetos relevantes en imágenes de pistas de aterrizaje tiene numerosas aplicaciones. Puede ser útil en aplicaciones de importancia práctica, como el mantenimiento automático de la iluminación de las pistas de vuelo o para asistir al piloto en aviones no equipados con ILS (Sistemas de Aterrizaje Instrumental, en inglés “Instrumental Landing Systems”). Para simplificar el problema, sólo se consideran imágenes nocturnas de pistas de vuelo, de manera que aparecen iluminadas artificialmente para destacar los elementos de interés (ver Figura 3).

Se pide a los alumnos que construyan un sistema que haga una extracción automática de ciertos objetos luminosos relevantes en imágenes de pistas de aterrizaje, agrupando los objetos en líneas (ver Figuras 3 y 4). La organización de este proyecto tiene dos etapas



Figura 2. En la imagen de la izquierda, se muestra un código de barras degradado debido al uso de umbralización regional. En la imagen de la derecha, se muestra el resultado de la construcción del mismo código de barras usando una dilatación morfológica.

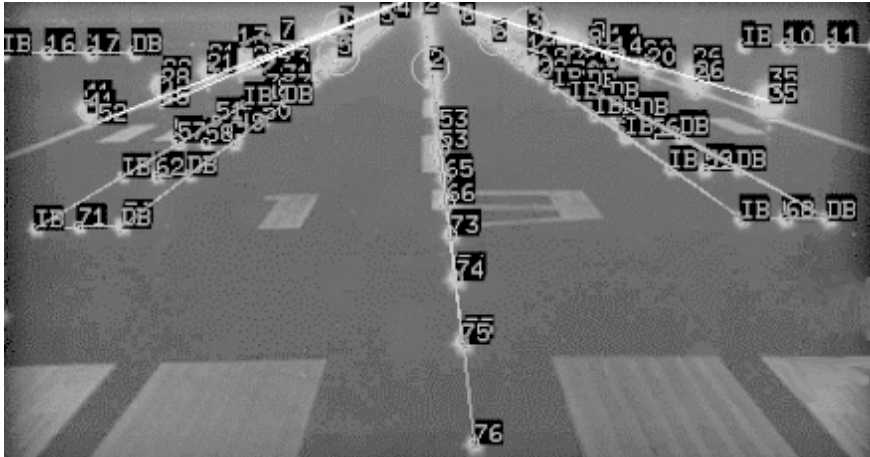


Figura 3. Imagen final segmentada con las ayudas visuales luminosas detectadas y agrupadas en líneas.

principales: 1) segmentación de las ayudas visuales luminosas (por ejemplo usando un procedimiento de umbralización), y 2) asociación en líneas de elementos de luz encontrados (por ejemplo utilizando la transformada de Hough) según un modelo propuesto de pista de aterrizaje.

6. Conclusiones

Este trabajo presenta un curso introductorio a la "Visión por Computador" con una marcada orientación algorítmica y práctica. Los requisitos de eficiencia (en tiempo de respuesta y memoria) de las aplicaciones son muy importantes y restrictivos en los algoritmos de

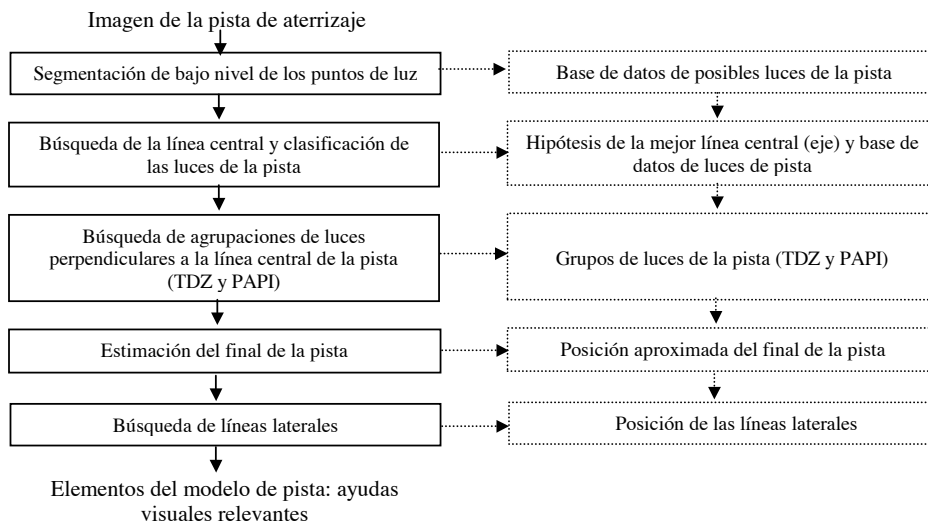


Figura 4. Diagrama de bloques del método propuesto de extracción automática de ayudas visuales relevantes.

visión. Por ello, los alumnos tienen que ser conscientes de lo importante que es el uso de algoritmos eficientes en sistemas de tratamiento de imágenes. Las técnicas de diseño de algoritmos pueden ayudar a idear dichos algoritmos y, en algunos casos, también ofrecen las soluciones óptimas para un problema de tratamiento de imágenes. Las propuestas de proyectos para ser realizados por grupos de alumnos, estaban pensadas como simplificaciones de aplicaciones industriales reales, lo que motivó enormemente a los alumnos de la titulación informática. Esto también les ha permitido analizar e implementar de forma sistemática y estructurada soluciones realistas. En algunos casos, parte de los proyectos de este curso han sido también diseñados y implementados eficientemente usando técnicas de diseño de algoritmos.

El énfasis puesto en la relación entre imágenes y algoritmos ha hecho que el curso sea más atractivo para los alumnos informáticos, quienes además se han acercado a aplicaciones de la industria. El enfoque algorítmico adoptado ha dado como resultado un porcentaje alto de alumnos aprobados en la asignatura.

Esperamos mejorar este curso cuando se vuelva a impartir en los próximos años.

Referencias

- [1] G.A. Baxes, *Digital Image Processing. Principles and Applications*, J. Wiley, 1994.
- [2] G. Brassard y P. Bratley, *Algorithmics. Theory and Practice*, Prentice-Hall, 1988.
- [3] T. H. Cormen, C. E. Leiserson y R. L. Rivest, *Introduction to Algorithms*, The MIT Press, 1990.
- [4] R.C. González y R.E. Woods, *Digital Image Processing*, Addison-Wesley, 1992.
- [5] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989.
- [6] R. Jain, R. Kasturi y B.G. Schunk, *Machine Vision*, McGraw Hill, 1995.
- [7] I. Kabir, *High Performance Computer Imaging*, Manning, 1997.
- [8] D. Maravall, *Reconocimiento de Formas y Visión Artificial*, Ed. Ra-ma, 1993.
- [9] The MathWorks Inc., *Matlab v. 5.1*, 1997.
- [10] Matrox Electronic Systems Ltd, *Matrox Imaging Library Reference Manual*, Ver 6.1, 1999, <http://www.matrox.com/imaging/prod/mil/>.