

Parte I

APÉNDICES

Apéndice A

INFERENCIA BAYESIANA

El razonamiento bayesiano proporciona un enfoque probabilístico a la inferencia. Está basado en la suposición de que las cantidad de interés son gobernadas por distribuciones de probabilidad y que se pueden tomar decisiones óptimas razonando sobre estas probabilidades junto con los datos obtenidos. Este enfoque está siendo utilizado en multitud de campos de investigación, de los que cabe destacar la robótica móvil y la visión computacional, ambas relacionadas con el contenido de esta tesis. En este apéndice queremos definir dos de las herramientas utilizadas en el desarrollo de esta tesis: el teorema de Bayes y el principio de longitud de descripción mínima.

A.1 Teorema de Bayes

A menudo nos surgen problemas en los cuales estamos interesados en determinar la mejor hipótesis h , dados los datos que hemos observado D . Una forma más correcta de expresar esto es decir que buscamos la hipótesis h más probable, dados los datos observados D más un conocimiento inicial sobre las probabilidades a priori de h . El teorema de Bayes nos proporciona un método directo para calcular estas probabilidades.

El teorema de Bayes se define con la siguiente ecuación:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Vamos a comentar el significado de cada término. $P(h)$ es el conocimiento inicial que tenemos sobre que la hipótesis h sea la correcta. Se le suele denominar la

probabilidad a priori de h . $P(D)$ se define de forma similar, pero esta vez sobre los datos D . $P(D|h)$ denota la probabilidad de observar los datos D dado que tenemos la hipótesis h . Se le suele denominar verosimilitud. Por último, $P(h|D)$ es la probabilidad a posteriori que la hipótesis h tiene, dados los datos observados D . En la mayoría de problemas donde se plantea la inferencia bayesiana, se parte de un conjunto de hipótesis H y se trata de encontrar la hipótesis más probable $h \in H$. De esta forma, a esta hipótesis más probable se le suele denominar hipótesis maximum a posteriori o MAP. Utilizando el teorema de Bayes, diremos que h_{MAP} es una hipótesis MAP de acuerdo a:

$$h_{MAP} \equiv \arg \max_{h \in H} P(h|D) = \arg \max_{h \in H} \frac{P(D|h)P(h)}{P(D)} = \arg \max_{h \in H} P(D|h)P(h)$$

En el último paso hemos eliminado $P(D)$ porque es independiente de h .

En algunos casos todas las probabilidades en H son igualmente probables a priori ($P(h_i) = P(h_j), \forall h_i, h_j \in H$). En este caso sólo utilizaríamos el término de verosimilitud, $P(D|h)$, y podemos simplificar aún más la anterior ecuación:

$$h_{ML} \equiv \arg \max_{h \in H} P(D|h)$$

donde a la hipótesis h_{ML} se le suele nombrar como hipótesis de máxima verosimilitud (Maximum Likelihood).

Supongamos ahora que debemos elegir entre dos hipótesis, h_1 y h_2 , dados los datos D . El criterio de elección para responder de forma eficiente sería seleccionar la hipótesis más probable. Es decir, aplicaríamos lo que se conoce como regla de decisión:

$$\text{si } P(h_1|D) > P(h_2|D) \text{ elegir } h_1, \text{ sino elegir } h_2$$

Si aplicamos la regla de Bayes a cada término nos queda:

$$P(D|h_1) \frac{P(h_1)}{P(D)} > P(D|h_2) \frac{P(h_2)}{P(D)}$$

$$\underbrace{\frac{P(D|h_1)}{P(D|h_2)}}_{\text{ratio verosimilitud}} > \underbrace{\frac{P(h_2)}{P(h_1)}}_{\text{ratio a priori}}$$

Aplicando logaritmos a ambas partes nos queda:

$$\ln \frac{P(D|h_1)}{P(D|h_2)} > \ln \frac{P(h_2)}{P(h_1)}$$

En ausencia de información a priori todas las hipótesis son igualmente probables y el término de la derecha es $\ln 1 = 0$. La regla de decisión en ausencia de información a priori queda:

$$\text{si } \ln \frac{P(D|h_1)}{P(D|h_2)} > 0 \text{ elegir } h_1, \text{ sino elegir } h_2$$

A.2 Principio de longitud de descripción mínima

El principio de longitud de descripción mínima (*minimum description length (MDL)*) puede ser resumido como “elegir la explicación más corta a los datos observados”. Esta íntimamente relacionada con el criterio MAP antes comentado, incorporando conceptos básicos de teoría de la información. Retomando la definición de h_{MAP} :

$$h_{MAP} = \arg \max_{h \in H} P(D|h)P(h)$$

y, de forma equivalente, expresando esta ecuación en términos de la maximización de \log_2 :

$$h_{MAP} = \arg \max_{h \in H} \log_2 P(D|h) + \log_2 P(h)$$

o, alternativamente, minimizando el negativo de esta cantidad:

$$h_{MAP} = \arg \min_{h \in H} -\log_2 P(D|h) - \log_2 P(h)$$

Esta última ecuación puede ser interpretada como que se prefieren hipótesis cortas. Cada uno de estos términos se puede entender como la longitud de descripción de las distribuciones bajo una codificación óptima. No vamos a entrar en comentar los términos de teoría de información. El principio MDL recomienda la elección de las hipótesis que minimizan estas dos longitudes de descripción. Así, este principio se puede definir como elegir la hipótesis h_{MDL} dada:

$$h_{MDL} = \arg \max_{h \in H} L_{C_1} P(D|h) + L_{C_2} P(h)$$

siendo L_{C_i} la longitud de descripción del mensaje i con respecto a C , que es el número de bits requeridos para codificar el mensaje i utilizando el código C . En el caso de que C_2 sea la codificación óptima de las hipótesis (h) y C_1 sea la codificación óptima de $(D|h)$, entonces $h_{MDL} = h_{MAP}$.

A.3 Hipótesis de máxima verosimilitud y error cuadrático medio

También vamos a demostrar una equivalencia entre la hipótesis de máxima verosimilitud y el método que encuentra la hipótesis que minimiza el error cuadrático medio. Como ya hemos comentado:

$$h_{ML} = \arg \max_{h \in H} P(D|h)$$

Vamos a asumir que los datos $D = (d_1, d_2, \dots, d_m)$ son independientes dado h , y así escribir podemos escribir $P(D|h)$ como el producto de los distintos $P(d_i|h)$:

$$h_{ML} = \arg \max_{h \in H} \prod_{i=1}^m P(d_i|h)$$

Suponiendo que las funciones de distribución son normales, con media μ y varianza σ^2 , vamos a hacer coincidir la hipótesis h con la media μ . Sustituyendo en la ecuación anterior tenemos:

$$h_{ML} = \arg \max_{h \in H} \prod_{i=1}^m \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i-\mu)^2}$$

Siendo $\ln P$ una función monotónica de P , maximizar $\ln P$ equivale a maximizar P , y eliminando la varianza, que no depende de la hipótesis h , nos queda:

$$h_{ML} = \arg \max_{h \in H} \sum_{i=1}^m -\frac{1}{2\sigma^2} (d_i - \mu)^2 = \arg \min_{h \in H} \sum_{i=1}^m (d_i - \mu)^2$$

Esta ecuación muestra que la hipótesis de máxima verosimilitud es la que minimiza la suma de los errores cuadráticos entre la hipótesis y los datos, siempre en los supuestos antes mencionados.

Apéndice B

BÚSQUEDAS DENTRO DE UN RANGO

Un problema clásico en geometría computacional es la búsqueda dentro de un rango. El problema se puede definir de la siguiente manera:

Tenemos un conjunto de puntos en el espacio n -dimensional. Se trata de encontrar de forma eficiente el conjunto de puntos cuyas coordenadas se encuentran situadas dentro de un rango. En el caso de dos dimensiones el rango especificado podría ser $[x_1 : x_2] \times [y_1 : y_2]$, indicando que las coordenadas de los puntos que se devuelven tienen que estar comprendidas entre dichos intervalos.

Se pueden encontrar varias propuestas en la literatura de tipos de datos para realizar estas búsquedas. En [de Berg *et al.*, 1997] se presentan algunas de las estructuras de datos más eficientes para realizar estas consultas, de las cuales se ha seleccionado la que presenta una menor complejidad en el tiempo de consulta. La estructura seleccionada es el árbol de rango (*Range Tree*) que es una ampliación del clásico árbol binario en una dimensión. En la siguiente sección se comenta la estructura del árbol binario y la forma de interrogar a dicho árbol, analizándose la complejidad de dicha estructura. En la Sección B.2 se detalla cómo se puede crear la estructura de árbol de rango y cómo se realiza la consulta de rango a dicha estructura.

B.1 Búsqueda unidimensional

Una búsqueda en una única dimensión es trivial: ordenamos todos los valores en un vector y simplemente realizamos una búsqueda lineal viendo qué valores se encuentran entre el rango especificado. Sin embargo aquí vamos a introducir una estructura de datos llamada *árbol binario* y que sirve de base para ampliar a mayores dimensiones. Cada nodo de este árbol tendrá la siguiente información: Un valor \mathcal{V} , un puntero a un subárbol izquierda \mathcal{L} y otro a un subárbol derecha \mathcal{R} . Dado un conjunto de valores ordenados de menor a mayor, la creación del árbol binario se puede definir de forma recursiva tal como se muestra en la Figura B.1.

Algoritmo de creación de un árbol binario ALG_CAB

Entrada: Lista de valores ordenados L .
Salida: Nodo raíz del árbol creado T .

Si $Longitud(L)=1$ **entonces** /*Un único elemento*/
 Crear un nuevo nodo N con $\mathcal{V} = Obtener_Elemento(L)$ y los punteros de los subárboles apuntando a un árbol vacío.
sino
 Obtener el nodo cuyo valor es el central de la lista de valores (la mediana): v_{med}
 Particionar la lista L en dos sublistas L_1 y L_2 de tal forma que L_1 contiene todos los elementos menores o iguales a v_{med} y L_2 los estrictamente mayores.
 Crear un nuevo nodo N con $\mathcal{V} = v_{med}$. El puntero izquierdo \mathcal{L} apuntará al árbol devuelto en la llamada a la función $Alg_CAB(L_1)$ y el derecho \mathcal{R} ídem $Alg_CAB(L_2)$.
FinSi
Devolver N .

Figura B.1: Algoritmo de creación de un árbol binario.

Si tenemos los siguientes valores: 3, 5, 10, 15, 16, 20, 22, 23, 30, el resultado de la creación de este árbol se puede observar en la Figura B.2. Los valores almacenados en los nodos internos permiten guiar la búsqueda hacia los valores en las hojas. Todos los valores en el subárbol izquierdo de un determinado nodo contienen valores menores o iguales que el valor almacenado en dicho nodo, mientras que en el subárbol derecho estos valores son mayores.

El método de consulta parte de un determinado rango a buscar $[x_1 : x_2]$ y devuelve los puntos del árbol que se encuentran dentro de ese rango. El método se divide en dos fases: primero encontrar el nodo de partición para, a continuación, ir explorando a partir de este nodo e ir devolviendo los valores dentro del rango. Este nodo de partición (v_{part}) es aquel nodo del árbol cuyo valor es mayor que x_1 y menor que x_2 , es decir se encuentra dentro del rango a buscar, y que además tiene

menor profundidad (más próximo a la raíz). En la Figura B.2 se muestra el nodo de partición para un rango de búsqueda de $[4 : 15]$. La Figura B.3 muestra el resultado de buscar un determinado rango en el árbol binario. Primero encontramos el nodo de partición. Después buscamos por el hijo izquierdo, quedándonos con los subárboles derechos que se encuentren dentro del rango, mientras que cuando buscamos por el hijo derecho nos quedamos con los subárboles izquierdos. En la Figura B.5 se detalla el método de consulta de rango.

La función `Devolver_Subárbol()` devuelve una lista con todos los puntos que pertenecen al árbol pasado como parámetro a la función.

La creación de este árbol se puede realizar con una complejidad temporal de $\mathcal{O}(n \log n)$ y una espacial de $\mathcal{O}(n)$. En cuanto a la complejidad de la consulta, siendo k el número de puntos devueltos se tiene una complejidad temporal de $\mathcal{O}(k + \log n)$.

B.2 Árboles de rango

Este tipo de árbol permite obtener, de manera eficiente, los puntos dentro de un espacio bidimensional cuyas coordenadas se encuentran dentro de un determinado rango. Si P es el conjunto de puntos totales, una consulta de rango

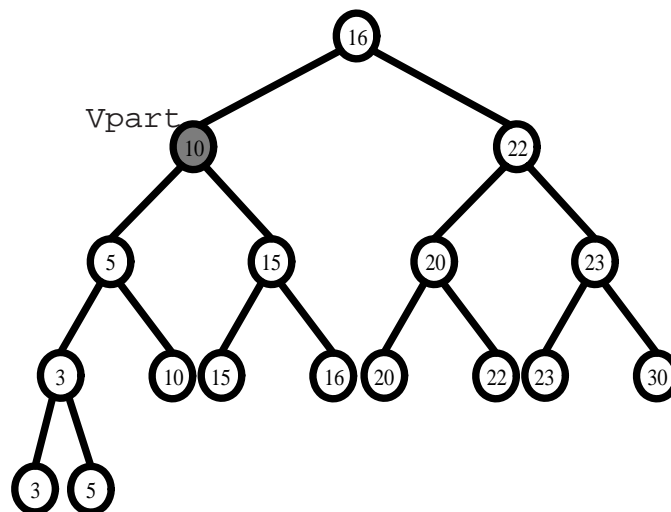


Figura B.2: Árbol binario creado indicando el nodo de partición.

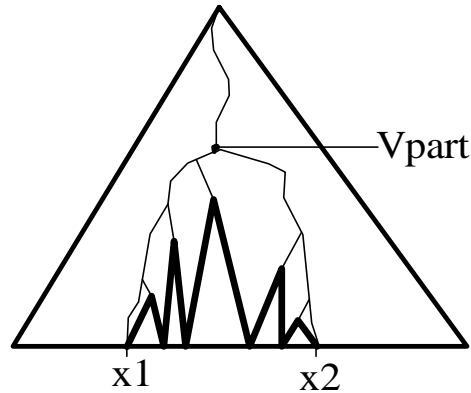


Figura B.3: Obtención de los puntos dentro de un rango.

$([x_1 : x_2] \times [y_1 : y_2])$ en dos dimensiones sobre P devolverá todos aquellos puntos que estén situados dentro del rectángulo definido por el rango de consulta. Un punto $p = (p_x, p_y)$ está situado dentro de dicho rectángulo si se cumple que:

$$p_x \in [x_1 : x_2] \quad \text{y} \quad p_y \in [y_1 : y_2]$$

Una consulta de este tipo se puede dividir en dos subconsultas, una para las coordenadas x y otra para las y . El árbol de rango es un árbol binario en el cual se han utilizado las coordenadas x de los puntos para su construcción. La consulta principal se realiza sobre estas coordenadas. Asociado a cada nodo de este árbol principal tenemos un árbol binario como el descrito en la anterior sección pero construido a partir de las coordenadas y de los puntos hijos de este nodo. La construcción de esta estructura es idéntica a la del árbol binario, excepto que el primer paso del algoritmo debe ser construir un árbol binario, asociado al nodo en cuestión, con las coordenadas y de los puntos (ver Figura B.4).

En cuanto a la consulta, el algoritmo también es muy similar al descrito para consulta en árboles binarios. Se realiza una consulta sobre las coordenadas x del árbol, y cuando se llamaba a la función `Devolver_Subárbol` ahora se debe realizar una consulta al árbol asociado con el rango de las y .

Si tenemos un conjunto P de n puntos, la creación de un árbol de rango se puede realizar con una complejidad temporal de $\mathcal{O}(n \log n)$ y una espacial de $\mathcal{O}(n \log n)$. En cuanto a la complejidad de la consulta, siendo k el número de puntos devueltos se tiene una complejidad temporal de $\mathcal{O}(k + \log^2 n)$.

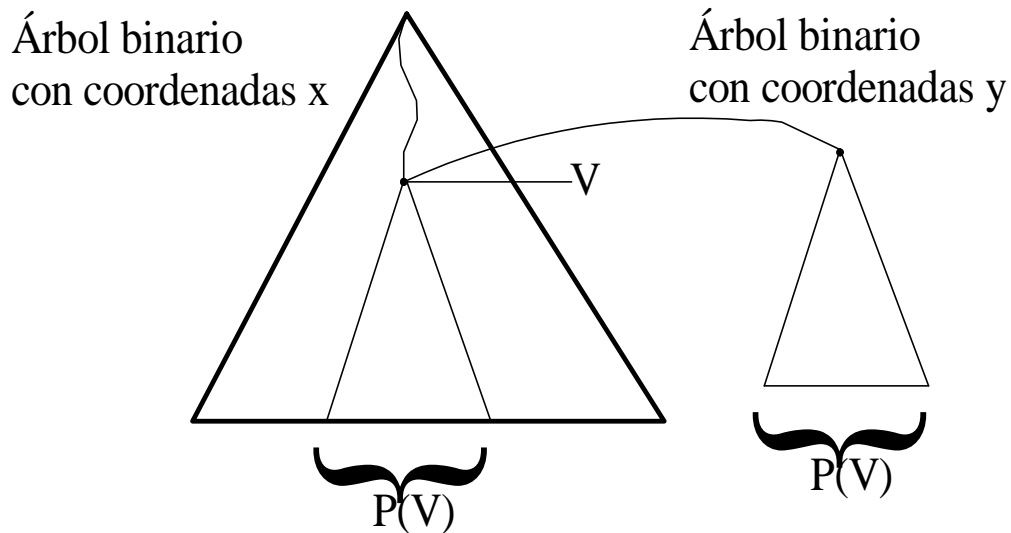


Figura B.4: Árbol de rango con el árbol binario asociado.

Algoritmo de consulta en un árbol binario ALG_COAB

Entrada: Un árbol \mathcal{T} y un rango de búsqueda $[x_1 : x_2]$

Salida: Lista de puntos del árbol que se encuentran dentro del rango.

Crear una lista vacía $Lista$

$v_{part} = \text{raíz}(\mathcal{T})$

$x_{part} = \text{valor}(v_{part})$

mientras v_{part} no es hoja **y** $(x_2 \leq x_{part} \text{ o } x_1 > x_{part})$ **hacer**

Si $x_2 \leq x_{part}$ **entonces**

$v_{part} = \text{Hijo_izquierdo}(v_{part})$

sino

$v_{part} = \text{Hijo_derecho}(v_{part})$

FinSi

$x_{part} = \text{valor}(v_{part})$

FinMientras

```

Si  $v_{part}$  es hoja entonces
  Si  $(x_{part} \geq x_1 \text{ y } x_{part} \leq x_2)$  entonces Insertar  $x_{part}$  en Lista FinSi
sino
  /* Primero se sigue el camino de la izquierda */
   $v = v_{part}$ 
   $v = \text{Hijo\_izquierdo}(v)$ 
  mientras  $v$  no es hoja hacer
     $x_v = \text{valor}(v)$ 
    Si  $x_1 \leq x_v$  entonces
      Concatenar (Lista, Devolver_Subárbol(Hijo_derecho( $v$ )))
       $v = \text{Hijo\_izquierdo}(v)$ 
    sino
       $v = \text{Hijo\_derecho}(v)$ 
    FinSi
  FinMientras
  /* El nodo hoja también debe de ser chequeado */
  Si  $(x_v \geq x_1 \text{ y } x_v \leq x_2)$  entonces Insertar  $x_v$  en Lista FinSi
  /* Ahora se continúa por la derecha */
   $v = v_{part}$ 
   $v = \text{Hijo\_derecho}(v)$ 
  mientras  $v$  no es hoja hacer
     $x_v = \text{valor}(v)$ 
    Si  $x_2 \geq x_v$  entonces
      Concatenar (Lista, Devolver_Subárbol(Hijo_izquierdo( $v$ )))
       $v = \text{Hijo\_derecho}(v)$ 
    sino
       $v = \text{Hijo\_izquierdo}(v)$ 
    FinSi
  FinMientras
  /* El nodo hoja también debe de ser chequeado */
  Si  $(x_v \geq x_1 \text{ y } x_v \leq x_2)$  entonces Insertar  $x_v$  en Lista FinSi
FinSi
Devolver Lista

```

Figura B.5: Algoritmo de consulta de un árbol binario.

Apéndice C

EXTRACCIÓN DE SEGMENTOS RECTOS

En este apéndice vamos a detallar un algoritmo de extracción de segmentos rectos, a partir de un segmento de forma libre. El algoritmo es el siguiente:

Algoritmo de extracción de segmentos rectos ALG_EXT_SEGMENTOS

Entrada: Conjunto de segmentos a particionar.

Salida: Conjunto de segmentos particionados.

Para cada segmento del conjunto de entrada **hacer**

 Encontrar un punto de ruptura. Este punto es aquel cuya distancia a la recta definida por los puntos inicial y final del segmento sea máxima.

 Creamos dos segmentos resultado de particionar el segmento original por el punto de ruptura.

 Cada segmento tendrá asociado un valor que viene determinado por el ratio longitud/desviación. Si un segmento es demasiado pequeño, su ratio es cero. Si el valor de un segmento es el mismo que el segmento que lo originó, no se realiza la partición. En caso contrario, calculamos recursivamente el valor del segmento.

 Para los segmentos finales se ajusta el mejor segmento que pase por todos los puntos, mediante un esquema de mínimos cuadrados.

FinPara

La Figura C.1 muestra un ejemplo de aplicación de este método.

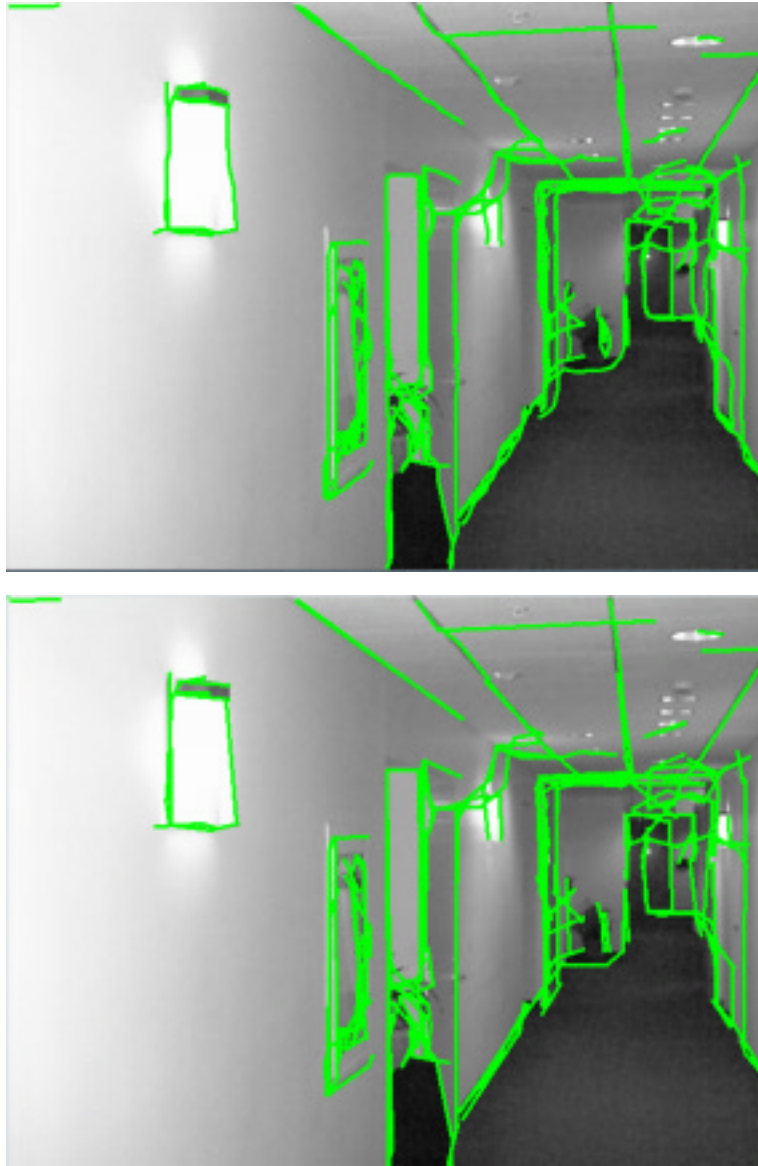


Figura C.1: Ejemplo de obtención de los segmentos rectos a partir del resultado del agrupamiento.

BIBLIOGRAFÍA

- [Aloimonos *et al.*, 1988] Y. Aloimonos, I. Weiss, y A. Bandopadhyay. Active vision. *International Journal of Computer Vision*, 2:333-356, 1988.
- [Alvarez y Morales, 1997] L. Alvarez y F. Morales. Affine morphological multiscale analysis of corners and multiple junctions. *International Journal of Computer Vision*, 25(2):95-107, 1997.
- [Asada y Brady, 1986] H. Asada y M. Brady. The curvature primal sketch. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):2-14, 1986.
- [Bajcsy, 1988] R. Bajcsy. Active perception. In *Proc of the IEEE*, 1988.
- [Ballard, 1991] D.H. Ballard. Animate vision. *Artificial Intelligence*, 48:57-86, 1991.
- [Barnard, 1983] S. Barnard. Interpreting perspective images. *Artificial Intelligence*, 21, 1983.
- [Bartlett *et al.*, 1998] M. Bartlett, H. Martin, y T.J. Sejnowski. Independent component representations for face recognition. In *Proc de la SPIE Symposium on Electric Imaging*, 1998.
- [Beaudet, 1978] P.R. Beaudet. Rotational invariant image operators. In *Proc. of the 4th. International Conference on Pattern Recognition*, 1978.
- [Belhumeur, 1996] P. Belhumeur. A bayesian approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3), 1996.
- [Bell y Pau, 1990] B. Bell y L.F. Pau. Contour tracking and corner detection in a logic programming environment. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(9):913-917, September 1990.
- [Brillault-O Mahoney, 1991] B. Brillault-O Mahoney. New method for vanishing pint detection. *Computer Vision, Graphics and Image Processing: Image Understanding*, 54, 1991.

- [Brunstrom *et al.*, 1996] K. Brunstrom, J. Eklundh, y T. Uhlin. Active fixation for scene exploration. *International Journal of Computer Vision*, 17(2), February 1996.
- [Canny, 1986] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679-698, November 1986.
- [Cazorla *et al.*, 1999a] M.A. Cazorla, F. Escolano, D. Gallardo, y O. Colomina. A competition-based deformable template for junctions detection. In *Proc de VIII SNRFAI*, 1999.
- [Cazorla *et al.*, 1999b] M.A. Cazorla, F. Escolano, D. Gallardo, y R. Rizo. Bayesian models for finding and grouping junctions. In *Proc of the EMMCVPR 99. Lectures Notes in Computer Science*, 1999.
- [Coughlan y Yuille, 1998] J. Coughlan y A.L. Yuille. Bayesian a* tree search with expected $o(n)$ convergence rates for road tracking. *Artificial Intelligence*, 1998.
- [Coughlan y Yuille, 1999] J. Coughlan y A. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Proc of the International Conference on Computer Vision*, 1999.
- [Coughlan, 1998] J. Coughlan. *Efficient Search Algorithms for Finding Deformable Templates*. PhD thesis, Harvard University, Cambridge, Massachusetts, January 1998.
- [Cover y Thomas, 1991] T. Cover y J. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications, 1991.
- [de Berg *et al.*, 1997] M. de Berg, M. van Kreveld, M. Overmars, y O. Schwarkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 1997.
- [Deriche y Faugeras, 1990] R. Deriche y O.D. Faugeras. 2d curve matching using high curvature points: Application to stereo vision. In *Proc. of the 10th. International Conference of Pattern Recognition*, pages 240-242, 1990.
- [Deriche y Giraudon, 1993] R. Deriche y G. Giraudon. A computational approach for corner and vertex detection. *International Journal of Computer Vision*, 10(2):101-124, 1993.
- [Dickmanns, 1997] E.D. Dickmanns. Vehicles capable of dynamic vision. In *Proc de la 15 IJCAI*, 1997.

- [Dreschler y Nagel, 1982] Dreschler y Nagel. On the selection of critical points and local curvature extrema of region boundaries for interframe matching. In *Proc of the International Conference of Pattern Recognition*, pages 542–544, 1982.
- [Duda y Hart, 1973] R. O. Duda y P. E. Hart. *Pattern Classification and Scene Analysis*, page 20. John Wiley and Sons, 1973.
- [Escolano, 1997] F. Escolano. *Plantillas deformables extendidas: modelización local robusta y caracterización*. PhD thesis, Universidad de Alicante, Mayo 1997.
- [Faugeras, 1993] O. Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, Cambridge, Massachusetts, 1993.
- [Figuereido *et al.*, 1997] M.A. Figueredo, J. Leitao, y A.K. Jain. Adaptive parametrically deformable contours. In *Proc of the Energy Minimization Methods in Computer Vision and Pattern Recognition Workshop*, 1997.
- [Gallardo, 1999] D. Gallardo. *Aplicación del muestreo bayesiano en robots móviles: estrategias para localización y estimación de mapas del entorno*. PhD thesis, Universidad de Alicante, Mayo 1999.
- [Geman y Geman, 1984] S. Geman y D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.
- [Geman y Jedynak, 1996] D. Geman y B. Jedynak. An active testing model for tracking roads in satellite images. *IEEE tran. on Pattern Analysis and Machine Intelligence*, 18(1), January 1996.
- [Giraudon y Deriche, 1991] G. Giraudon y R. Deriche. On corner and vertex detection. In *Proc of the Conference on Computer Vision and Pattern Recognition*, pages 650–655, 1991.
- [Grzywacz y Escolano, 2000] N. Grzywacz y F. Escolano. Visions of vision: Vision in brains and computers. MIT Press. Computational Neuroscience Series. En preparación, 2000.
- [Han *et al.*, 1989] M.H. Han, D. Jang, y J. Foster. Identification of corners points of two-dimensional images using a line search method. *Pattern Recognition*, 22(1):13–20, 1989.
- [Harris y Stephens, 1988] C. G. Harris y M. Stephens. A combined corner and edge detection. In *Proc of the 4th Alvey Vision Conference*, pages 147–151, 1988.

- [Kitchen y Rosenfeld, 1982] L. Kitchen y A. Rosenfeld. Gray level corner detection. *Pattern Recognition Letters*, 1(2):95-102, 1982.
- [Kittler, 1997] J. Kittler. Probabilistic relaxation: Potential, relationships and open problems. In *Proc of the Energy Minimization Methods in Computer Vision and Pattern Recognition Workshop*, pages 393-408, 1997.
- [Knill y Richards, 1996] D. Knill y W. Richards. *Perception as Bayesian inference*. Cambridge University Press, 1996.
- [Koenderink y Richards, 1988] J.J. Koenderink y W. Richards. Two-dimensional curvature operators. *Journal of the Optical Society of America*, 5(7):1136-1141, July 1988.
- [Konishi *et al.*, 1999] S. Konishi, J. Coughlan, A. Yuille, y C. Zhu. Fundamental bounds on edge detection: Edge cues. In *Proc of the International Conference on Computer Vision and Pattern Recognition*, 1999.
- [Krotkov, 1987] E.P. Krotkov. *Exploratory Visual Sensing for Determining Spatial Layout with an Agile Stereo Camera System*. PhD thesis, Univ. of Pennsylvania, 1987.
- [Landy *et al.*, 1996] M.S. Landy, T. Maloney, y M. Pavel, editors. *Exploratory Vision. The Active Eye*. Springer Verlag, 1996.
- [Lindeberg y Li, 1997] T. Lindeberg y M. Li. Segmentation and classification of edges using minimum description length approximation and complementary junction cues. *Computer Vision and Image Understanding*, 67(1):88-98, July 1997.
- [Lowe, 1985] D.G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer, 1985.
- [Lowe, 1987] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31, 1987.
- [Magee y Aggarwal, 1984] M.J. Magee y J.K. Aggarwal. Determining vanishing points from perspective images. *Computer Vision, Graphics and Image Processing*, 26, 1984.
- [Malik, 1987] J. Malik. Interpreting line drawings of curved objects. *International Journal of Computer Vision*, 1(1):73-104, 1987.
- [Marr, 1982] D. Marr. *Vision*. Freeman, San Francisco, 1982.

- [Matas y Kittler, 1995] J. Matas y J. Kittler. Contextual junctions finder. In J. Crowley y H. Christensen, editors, *Vision as process*. Springer, 1995.
- [McLean y Kotturi, 1995] G.F. McLean y D. Kotturi. Vanishing point detection by line clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17, 1995.
- [Medioni y Yasumoto, 1987] G. Medioni y Y. Yasumoto. Corner detection and curve representation using cubic B-splines. *Computer Vision, Graphics, and Image Processing*, 39:267–278, 1987.
- [Mehrota y Nichani, 1990] R. Mehrota y S. Nichani. Corner detection. *Pattern Recognition*, 23(11):1223–1233, 1990.
- [Micheli *et al.*, 1989] E. De Micheli, B. Caprile, P. Ottanelo, y V. Torre. Localization and noise in edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(10), 1989.
- [Mitchell, 1997] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
- [Moons *et al.*, 1996] T. Moons, L. Van Gool, M. Proesmans, y E. Pauwels. Affine reconstruction from perspective image pairs with a relative object-camera translation in between. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18, 1996.
- [Moravec, 1977] H. P. Moravec. Towards automatic visual obstacle avoidance. In *Proc. of the International Joint Conference on Artificial Intelligence*, page 584, 1977.
- [Mundy y Zisserman, 1992] J.L. Mundy y Z. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, 1992.
- [Nitzberg *et al.*, 1993] M. Nitzberg, D. Mumford, y T. Shiota. *Filtering, Segmentation and Depth*. Springer-Verlag, 1993.
- [Parida *et al.*, 1997] L. Parida, D. Geiger, y R. Hummel. Kona: A multi-junction detector using minimum description length principle. In *Proc of the Energy Minimization Methods in Computer Vision and Pattern Recognition Workshop*, pages 51–65, 1997.
- [Parida *et al.*, 1998] L. Parida, D. Geiger, y R. Hummel. Junctions: Detection, classification, and reconstruction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(7), 1998.

- [Parodi y Piccioli, 1996] P. Parodi y G. Piccioli. 3d shape reconstruction by using vanishing points. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18, 1996.
- [Pearl, 1984] J. Pearl. *Heuristics. Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, 1984.
- [Quan y Mohr, 1989] L. Quan y R. Mohr. Determining perspective structures using hierarchical hough transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9, 1989.
- [Rangarajan *et al.*, 1989] K. Rangarajan, M. Shah, y D. Van Brackle. Optimal corner detector. *Computer Vision, Graphics, and Image Processing*, 48:230-245, 1989.
- [Rattarangsi y Chin, 1992] A. Rattarangsi y R.T. Chin. Scale-based detection of corners of planar curves. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(4):430-449, 1992.
- [Rimey y Brown, 1992] R. Rimey y C. Brown. Task-oriented vision with multiple bayes nets. In A. Yuille y A. Blake, editors, *Active Vision*. MIT Press, Cambridge, MA, 1992.
- [Sarkar y Boyer, 1993] S. Sarkar y K.L. Boyer. Perceptual organization in computer vision: a review and a proposal for a classificatory structure. *IEEE Trans. on Systems, Man, and Cybernetics*, 23:382-399, 1993.
- [Shufelt, 1999a] J.A. Shufelt. Performance evaluation and analysis of monocular building extraction from aerial imagery. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21, 1999.
- [Shufelt, 1999b] J.A. Shufelt. Performance evaluation and analysis of vanishing point detection techniques. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21, 1999.
- [Smith y Brady, 1997] S. M. Smith y J. M. Brady. Susan= a new approach to low level image processing. *Int. J. Comp. Vision*, 23(1):45-78, 1997.
- [Torre y Poggio, 1986] V. Torre y T.A. Poggio. On edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(2):147-163, March 1986.
- [Trucco y Verri, 1998] S. Trucco y A. Verri. *Introductory Techiques in Computer Vision*. Prentice-Hall, 1998.
- [Ullman, 1996] S. Ullman. *High-Level Vision*. MIT Press, 1996.

- [Vasselle *et al.*, 1994] B. Vasselle, G. Giraudon, y M. Berthod. Following corners on curves and surfaces in the scale space. In *Proc. of the European Conference on Computer Vision 94*, volume 800, pages 109–114, 1994.
- [Waltz, 1975] D. Waltz. Understanding line drawing of scenes with shadows. In *The Psychology of Computer Vision*. McGraw-Hill, New York, 1975.
- [Wandell, 1995] B.A. Wandell. *Foundations of Vision*. Sinauer Associates, Inc., 1995.
- [Yuille y Blake, 1992] A.L. Yuille y A. Blake. *Active Vision*. MIT Press, 1992.
- [Yuille y Coughlan, 1997] A.L. Yuille y J. Coughlan. Twenty questions, foci of attention, and a*: A theoretical comparison of optimization strategies. In *Proc of the EMCCVPR 99*. Lectures Notes in Computer Science, 1997.
- [Yuille y Coughlan, 1999] A. Yuille y J. Coughlan. Visual search: Fundamental bounds, order parameters, and phase transitions. *IEEE tran. on Pattern Analysis and Machine Intelligence*, 1999.
- [Zerubia y Chellapa, 1993] J. Zerubia y R. Chellapa. Mean field approximation using compound gauss markov random fields for edge detection and image estimation. *IEEE Trans. on Neural Networks*, 8(4):703–709, 1993.
- [Zhu *et al.*, 1997] S.C. Zhu, Y. Wu, y D. Mumford. Minimax entropy principle and its applications to texture modeling. *Neural Computation*, 9(8), 1997.
- [Zhu y Yuille, 1996a] S. C. Zhu y A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(9):884–900, September 1996.
- [Zhu y Yuille, 1996b] S.C. Zhu y A.L. Yuille. Forms: A flexible object recognition and modelling system. *International Journal of Computer Vision*, 20(3):187–212, 1996.
- [Zuniga y Haralick, 1983] O. A. Zuniga y R. M. Haralick. Corner detection using the facet model. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 30–37, 1983.