



# Reconocimiento automático de áreas de interés en secuencias de interiores

## ANEXOS

Jorge Rituerto Sin

Directora: Ana Cristina Murillo Arnal

Ingeniería Industrial  
Automatización Industrial y Robótica

Departamento de Informática e Ingeniería de Sistemas  
Centro Politécnico Superior  
Universidad de Zaragoza

Febrero 2011



## Anexo A

# Comparación del tamaño de los *superpixels*

El número de segmentos en que se divide cada imagen de la secuencia es un factor determinante en el procesado de la misma. Si cada imagen es segmentada en un número muy elevado de *superpixels*, el tiempo de procesado de la secuencia va a ser muy alto. Si cada imagen se segmenta en un número demasiado pequeño de *superpixels*, el tiempo va a ser mucho menor, pero los *superpixels* no van a ser capaces de separar de forma adecuada los objetos y regiones de interés, generando un gran número de errores.

En la librería de segmentación utilizada [13] en la configuración final de nuestro proceso, el tamaño de los *superpixels* viene determinado por tres parámetros distintos:

- *sigma*: control el suavizado de los *superpixels*
- *k*: control de la escala de los *superpixels*
- *min*: control del tamaño mínimo de *superpixel*

Se ha realizado una comparación del tamaño de los *superpixels* para distintas configuraciones de los parámetros de control de la segmentación. En la Figura A.1 se observan las variaciones en la segmentación en función del parámetro *sigma*, siendo *k* y *min* constantes en todas ellas. La Figura A.2 muestra los resultados variando el parámetro *k*. La Figura A.3 muestra la imagen segmentada utilizando distintos valores de *min*.

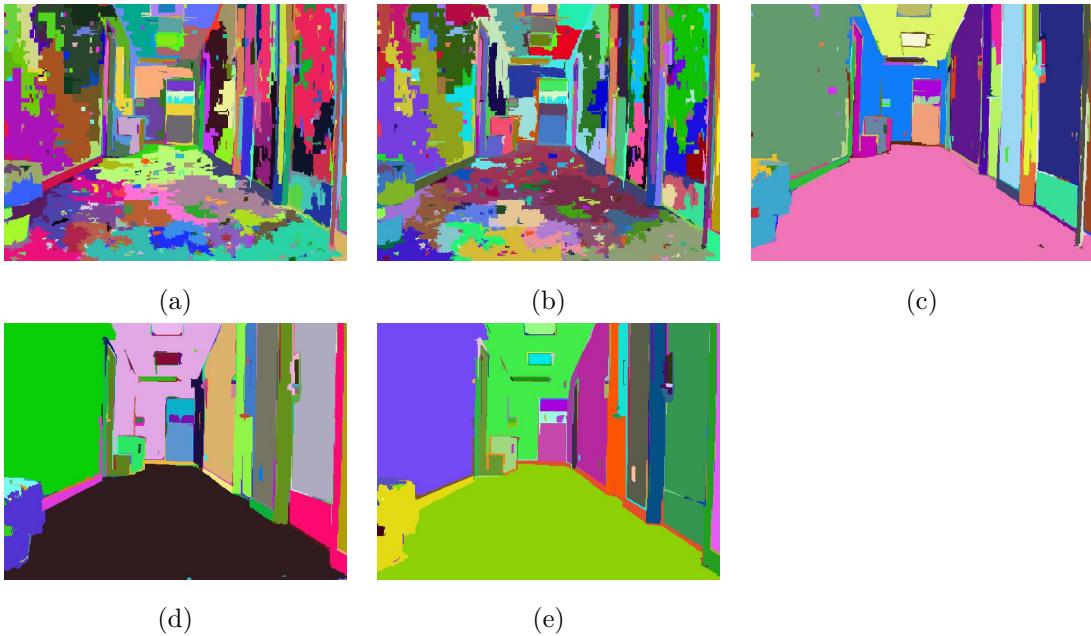


Figura A.1: Imágenes segmentadas variando el valor del parámetro  $\sigma$ , los parámetros utilizados son  $\sigma = \text{variable}$ ,  $k=500$ ,  $\min=20$ . (a)  $\sigma = 0$ ;(b)  $\sigma = 0.15$ ;(c)  $\sigma = 0.5$ ;(d)  $\sigma = 0.75$ ;(e)  $\sigma = 1$

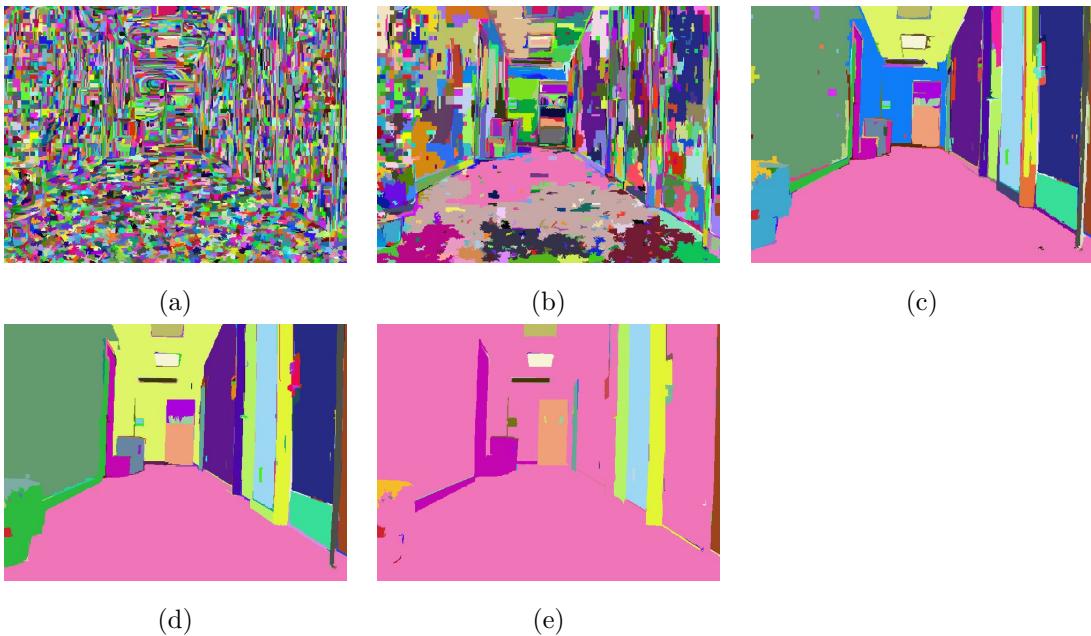


Figura A.2: Imágenes segmentadas variando el valor del parámetro  $\sigma$ , los parámetros utilizados son  $\sigma = 0.5$ ,  $k=\text{variable}$ ,  $\min=20$ . (a)  $k = 0$ ;(b)  $k = 50$ ;(c)  $k = 500$ ;(d)  $k = 1000$ ;(e)  $k = 5000$

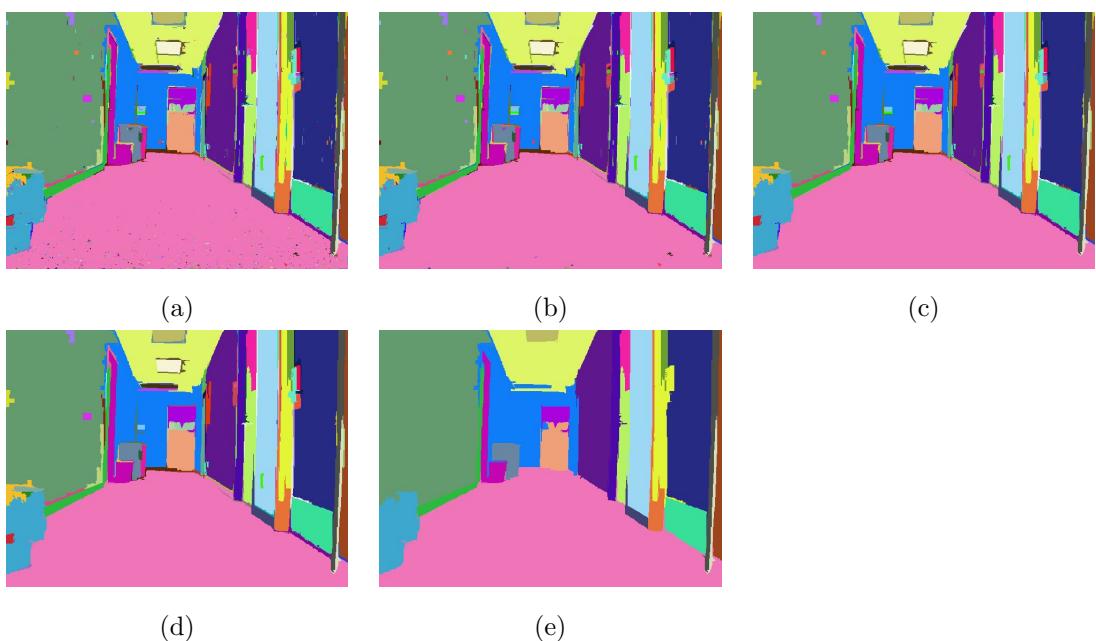


Figura A.3: Imágenes segmentadas variando el valor del parámetro *sigma*, los parámetros utilizados son *sigma* = 0.5, *k*=500, *min*=variable. (a) *min* = 0;(b) *min* = 10;(c) *min* = 40;(d) *min* = 100;(e) *min* = 1000



# Anexo B

## Descriptores

### B.1 Descriptores de color

El color es una de las características visuales más distinguibles y dominantes [20]. De acuerdo con la teoría estándar del color, existen tres tipos de conos en la retina (rojos, verdes y azules), cada uno de ellos absorbe la radiación de un determinado espectro de la luz, las señales de intensidad recibidas son traducidas antes de pasarlas al cerebro.

Existe una gran cantidad de modelos de color disponibles. Para el cálculo de los descriptores de color se han utilizado tres espacios de color representativos y analizados en [20].

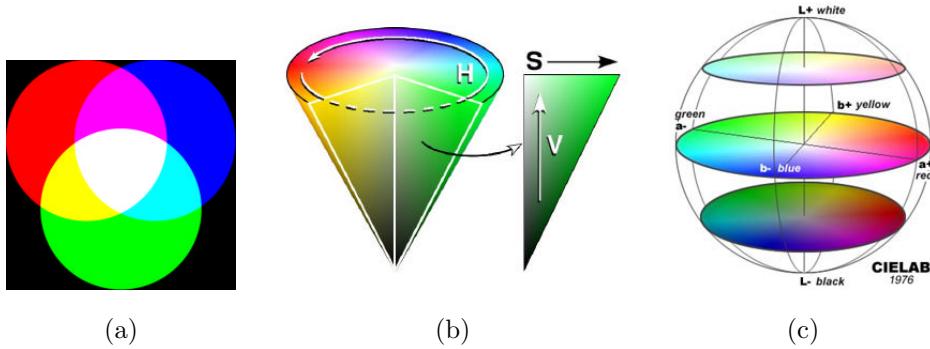


Figura B.1: (a) Espacio de color RGB; (b) Espacio de color HSV (c) Espacio de color Lab

**Descriptores basados en medidas estadísticas de primer orden** Un histograma (Figura B.2) es la representación gráfica de una variable en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados. En el caso de tratamiento de imágenes y videos, las variables representadas son las intensidades de los canales del espacio de color que corresponda.

$$h(i) = \text{número de píxeles con intensidad } i \quad (\text{B.1})$$

$$p(i) = \frac{\text{número de píxeles con intensidad } i}{\text{número total de píxeles}} \quad (\text{B.2})$$

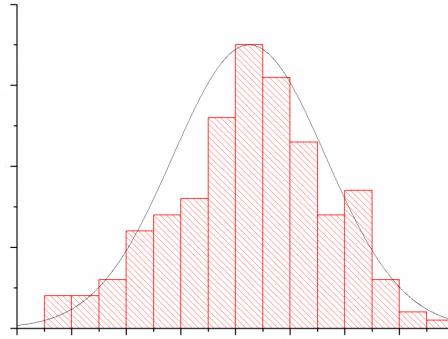


Figura B.2: Ejemplo de histograma

Cuando se trabaja en espacios de color de N canales, cada uno de los N canales es descrito por un histograma distinto, por lo que en los espacios de color *RGB*, *HSV* y *LAB* quedan definidos por tres histogramas diferentes.

**Momento de orden  $k$  respecto al origen** Se define el momento de orden  $k$  respecto al origen de la variable  $i$  como

$$m_k = E [i^k] = \sum_{n=1}^{255} i^k p(i) \quad (\text{B.3})$$

- El momento de orden 0 respecto al origen es igual a 1

$$m_0 = E [i^0] = 1$$

- El **nivel medio de intensidad** de cada canal de color de la imagen se define como el momento de orden  $k$  respecto al origen con  $k = 1$  y se le denomina  $\mu$ .

$$m_1 = \mu = E [i] = \sum_{n=1}^{255} i^1 p(i) = \text{nivel medio} \quad (\text{B.4})$$

**Momento central de orden  $k$**  En estadística, el momento central de orden  $k$  de una variable es la esperanza matemática de la diferencia entre el valor  $i$  y la media  $\mu$ . Si una variable aleatoria no tiene media el momento central es indefinido.

$$\mu_k = E [(i - \mu)] = \sum_{n=1}^{255} (i - \mu)^k p(i) \quad (\text{B.5})$$

- El momento central de orden  $\theta$  es uno.

$$\mu_0 = E[(i - \mu)^0] = 1$$

- El momento central de orden  $k = 1$  es igual a  $\theta$

$$\mu_1 = E[(i - \mu)] = E[i] - \mu = 0$$

- El momento central de orden  $k = 2$  es la varianza  $\sigma^2$ . Para poder realizar comparaciones de esta variable en situaciones en las que la unidad de medida o el orden de magnitud sea diferente, se puede utilizar el **coeficiente de variación**, que es adimensional

$$\mu_2 = Var(i) = \sigma^2 = \text{varianza}$$

$$\text{coeficiente de variación} = \frac{\pm \sqrt{\sigma^2}}{\mu} \quad (\text{B.6})$$

El coeficiente de variación representa la anchura del histograma (Figura B.3).

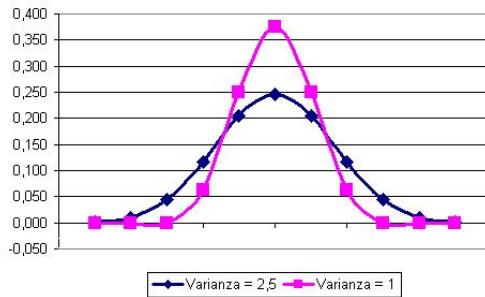


Figura B.3: Variación de la varianza según la distribución del histograma

- El momento central de orden  $k = 3$  es la asimetría de la distribución. A partir de la asimetría se puede calcular el **coeficiente de asimetría**  $\gamma_1$ , cuyas unidades son adimensionales.

$$\mu_3 = \text{asimetría}$$

$$\text{coeficiente de asimetría} = \gamma_1 = \frac{\mu_3}{\mu_2^3} = \frac{\mu_3}{\sigma_3} \quad (\text{B.7})$$

Si el coeficiente de asimetría es igual a  $0$ , la distribución es simétrica. Si el coeficiente es menor que  $0$ , la distribución está desplazada a la izquierda, mientras que si es mayor que  $0$ , la distribución está desplazada a la derecha (Figura B.4).

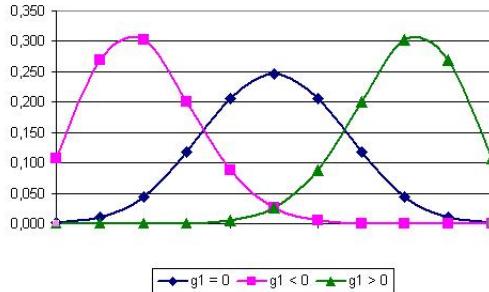


Figura B.4: Variación de la asimetría según la distribución del histograma

- El momento central de orden  $k = 4$  se denomina curtosis. El coeficiente adimensional de este parámetro es el **coeficiente de curtosis**  $\gamma_2$ .

$$\mu_4 = \text{curtosis}$$

$$\text{coeficiente de curtosis} = \gamma_2 = \frac{\mu_4^{[4]}}{\mu_2^2} = \frac{\mu_4}{\sigma^4} \quad (\text{B.8})$$

El coeficiente de curtosis representa el grado de apuntamiento o agudeza de la distribución. Las distribuciones mesocúrticas (de apuntamiento moderado) tienen un coeficiente de curtosis de igual a  $\beta$ . Las distribuciones platicúrticas (más aplanadas que la normal) tienen un coeficiente de curtosis menor que  $\beta$ , mientras que las distribuciones leptocúrticas (más altas o esbeltas que la normal) tienen un coeficiente mayor que  $\beta$ .

**Entropía** La entropía es una medida de la uniformidad del histograma. Cuanto mayor es el valor de la entropía de una distribución, mayor es la aleatoriedad de los valores de la distribución. Se puede considerar un descriptor de color, pero también puede ser considerado como descriptor de textura ya que una superficie más rugosa tendrá mayor entropía que otra más lisa. Puede ser calculada de la siguiente manera

$$H = -E [\log_2 p(i)] = -\sum_{n=1}^{255} p(i) \log_2 p(i) \quad (\text{B.9})$$

## B.2 Descriptores de textura

La caracterización de texturas se lleva a cabo a partir de medidas estadísticas de segundo orden, las cuales se definen sobre las matrices de co-ocurrencia o GLCM (*gray-level co-occurrence matrix*) [21]. Los valores de los elementos de la matriz de coocurrencia representan la frecuencia relativa con la que dos píxeles vecinos separados por una distancia  $d$  aparecen en la imágenes, donde uno de los píxeles tiene un nivel de gris  $i$  y el otro  $j$  [22]. Es una matriz simétrica y es

función del ángulo  $\phi$  entre los dos píxeles.

Sea  $I$  una imagen con una textura, la matriz de co-ocurrencia es definida como

$$A_d^\phi(i_1, i_2) = p(I(u, v) = i_1, I(u \pm df(\phi), v \pm dg(\phi)) = i_2) \quad (\text{B.10})$$

quedando definida en base a los parámetros

$$\begin{aligned} d &= \text{distancia} \\ \phi &= \text{orientación} \end{aligned}$$

Los descriptores de textura basados en medidas estadísticas de orden dos son obtenidos a partir de la matriz de co-ocurrencia o GLCM. Haralick propone trabajar sobre las cuatro orientaciones mostradas y promediar los resultados.

**Segundo Momento Angular (Angular Second Moment - ASM) o uniformidad de energía** Cuanto más suave es la textura, el valor del ASM es mayor, siendo igual a uno para superficies de intensidad constante.

$$ASM = \sum_{i=1}^{255} \sum_{j=1}^{255} p(i, j)^2 \quad (\text{B.11})$$

**Entropía de segundo orden** La entropía  $H$  toma valores más bajos cuanto más suave (menos aleatoria) es la textura.

$$H_{xy} = -\sum_{i=1}^{255} \sum_{j=1}^{255} p(i, j) \log_2 p(i, j) \quad (\text{B.12})$$

**Contraste** El contraste se define como la diferencia relativa en intensidad entre un punto de una imagen y sus alrededores.

$$CON = \sum_{i=1}^{255} \sum_{j=1}^{255} (i - j)^2 p(i, j) \quad (\text{B.13})$$

**Momento diferencial inverso (Inverse Difference Moment)** Es una medida de la homogeneidad de la imagen. Cuanto mayor es IDM, menor es el contraste.

$$IDM = \sum_{i=1}^{255} \sum_{j=1}^{255} \frac{p(i, j)}{1 + (i - j)^2} \quad (\text{B.14})$$

**Correlación** Mide la similitud entre píxeles vecinos.

$$C = \sum_{i=1}^{255} \sum_{j=1}^{255} \frac{(i - \mu_x)(j - \mu_y)}{\sigma_x \sigma_y} p(i, j) \quad (\text{B.15})$$

donde

$$\mu_x = \sum_{i=1}^{255} \sum_{j=1}^{255} i p(i, j)$$

$$\mu_y = \sum_{i=1}^{255} \sum_{j=1}^{255} j p(i,j)$$

$$\sigma_x = \sqrt{\sum_{i=1}^{255} \sum_{j=1}^{255} (i - \mu_x^2 p(i,j))}$$

$$\sigma_y = \sqrt{\sum_{i=1}^{255} \sum_{j=1}^{255} (j - \mu_y^2 p(i,j))}$$

El principal inconveniente de las medidas estadísticas de segundo orden es que el tiempo de cálculo de las mismas es muy elevado debido a la bidimensionalidad de las mismas. Para reducir el tiempo requerido por las mismas se ha modificado la escala de grises con la que se trabaja, reduciéndola de los 256 niveles originales a 16. De este modo se acelera el cálculo de los descriptores de textura al tiempo que se reduce la cantidad de memoria requerida para almacenar las matrices. Por contra, la exactitud de los resultados obtenidos será inferior.

### B.3 Descriptores de forma

**Área** Queda determinada por el número de píxeles en la forma, en este caso, el número de píxeles que componen la superficie de cada superpixel.

**Perímetro** Número de píxeles que componen el contorno del superpixel. Antes de proceder al cálculo de el perímetro, es necesario definir, de una manera o de otra, el contorno del superpixel. Para ello existe la función de Matlab *edge*. Al tratarse de una imagen binaria, los límites entre el superpixel y el fondo están perfectamente definidos, por lo que la fiabilidad al utilizar esta función predefinida es elevada, y en el resultado no aparece nada de ruido.

**Compacidad y redondez** Se define compacidad como

$$\text{compacidad} = \frac{\text{perímetro}^2}{\text{área}} \quad (\text{B.16})$$

Se define redondez como

$$\text{redondez} = \frac{\text{perímetro}^2}{\text{área}4\pi} \quad (\text{B.17})$$

Por lo tanto

$$\text{compacidad} = \text{redondez} \times 4 \quad (\text{B.18})$$

La información que se puede obtener por medio de la compacidad es proporcional a la de la redondez, por lo que solo será utilizada una de ellas. La forma con menor compacidad y redondez es el círculo.

$$\text{redondez(círculo)} = \frac{(2\pi R)^2}{4\pi R^2} = 1$$

$$\text{compacidad(círculo)} = \frac{(2\pi R)^2}{\pi R^2} = 4\pi$$

Como la mínima redondez posible es igual a 1, mientras que la compacidad de la misma figura es  $4\pi$ , se va a utilizar los valores de redondez como descriptor de la forma, dado que el resultado va a ser, en todos los casos, proporcional al de la compacidad.

**Excentricidad** Se puede definir como el ratio de la longitud de la cuerda más larga de la figura con respecto a la cuerda más larga perpendicular a ella.

$$\epsilon = \frac{(\mu_{20} + \mu_{02})^2 + 4 + \mu_{11}}{\text{área}} \quad (\text{B.19})$$

La excentricidad de una recta es  $\epsilon = 0$  y la excentricidad de un círculo es  $\epsilon = 1$ .

**Orientación** Es la dirección predominante de la figura.

$$\theta = \frac{1}{2} \tan^{-1} \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \quad (\text{B.20})$$

**Códigos de cadena** Las técnicas de códigos de cadena son ampliamente utilizadas por que permiten reducir considerablemente la cantidad de datos necesarios para la descripción de formas. Se ha utilizado un código basado en matrices de conectividad de ocho grados basado en [effChainCode]. Más información sobre los códigos de cadena en el [Anexo Códigos de Cadena].

## B.4 Descriptores de posición

Cuando un mismo objeto queda dividido por varios *superpixels* distintos, estos *superpixels* compartirán, además de la información referente a los aspectos ya tratados, como el color o la textura, una posición cercana dentro de la imagen. Por ello se van a definir distintos descriptores de posición que ayuden a situar los *superpixels*.

**Centroide** El centroide de una región queda definido por la media de las componentes  $x$  y las componentes  $y$  de los píxeles que la componen.

$$x_c = \frac{\sum_{i=1}^N x_i}{N} \quad (\text{B.21})$$

$$y_c = \frac{\sum_{i=1}^N y_i}{N} \quad (\text{B.22})$$

**Mallado** Se divide la superficie de la imagen en una malla imaginaria de 5x5 y se determina el cuadrante en el que está situado el centroide de cada superpixel.

**Límites de la imagen** Número de píxeles que pertenecen al superpixel y que están en contacto con alguno de los cuatro límites de la imagen (límite superior, inferior, izquierdo y derecho).

**Horizonte** Análogamente al mallado, esta vez se divide la imagen en dos por una línea horizontal que se considera la línea de horizonte.

## Anexo C

# Códigos de cadena

Las técnicas de códigos de cadena son ampliamente utilizadas porque permiten reducir considerablemente la cantidad de datos necesarios para la descripción de formas. El primer trabajo en la representación de curvas fue llevado a cabo por Freeman en 1961 [23]. El método aquí presentado se basa en el movimiento a lo largo de una curva digital o secuencia de pixels de contorno utilizando una conectividad de ocho grados.

La dirección de cada movimiento se representa utilizando un número  $\{i | i = 0, 1, 2, \dots, 7\}$  denotando un ángulo de  $45^\circ \times i$  en sentido antihorario a partir de el eje positivo de las  $x$  tal y como se muestra en la Figura C.1 (a). También puede utilizarse una conectividad de cuatro grados (Figura C.1 (b)).

En [24] se expone lo siguiente: el próximo elemento del código de Freeman es, muy a menudo, el mismo que su inmediato predecesor o el predecesor anterior. En otras palabras, cuando una cadena se mueve en una dirección definida por un elemento de código, la probabilidad de que una esquina gire bruscamente en el espacio de un píxel es muy baja. El análisis fue realizado para más de 1000 curvas, patrones de contornos y formas variadas encontradas de manera aleatoria en Internet. Los resultados estadísticos obtenidos en [24] se muestran en la tabla C.1.

Los códigos de cadena pueden ser aplicados en la caracterización de imágenes

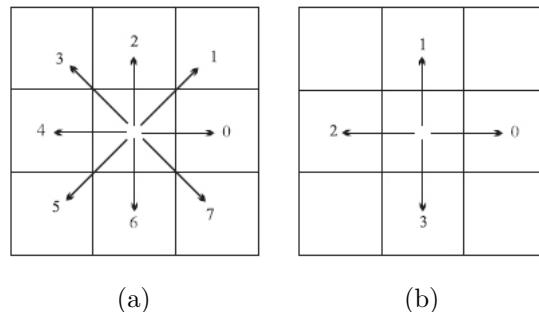


Figura C.1: (a) Conectividad de 8 grados; (b) Conectividad de 4 grados

Angle difference	$0^\circ$	$\pm 45^\circ$	$\pm 90^\circ$	$\pm 135^\circ$	$180^\circ$
Probability	0.453	0.488	0.044	0.012	0.003

Tabla C.1: Resultados estadísticos de la probabilidad de diferencia de ángulo entre segmentos contiguos. Tabla en [24]

como descriptores de la forma de los *superpixels*, como se comenta en el anterior anexo B, ya que con ellos se pueden definir varios aspectos relacionados con la forma.

- Determinación de la longitud de la cadena de cada superpixel.
- Análisis de las formas predominantes en cada superpixel por medio del histograma del código de cadena.
- Obtención de la compactación y redondez de cada superpixel por medio del perímetro y el área.

El código utilizado en este trabajo se emplea una conectividad de ocho grados priorizando la selección del siguiente píxel del contorno teniendo en cuenta los resultados estadísticos mostrados anteriormente en la tabla C.1.

Se recorre el contorno del *superpixel* en sentido horario partiendo del píxel con menor índice  $j$  (más a la izquierda) en la fila con menor índice  $i$  (fila superior) de entre los píxeles que pertenecen al *superpixel* que corresponda.

Debido a la necesidad de priorizar la dirección de búsqueda de la siguiente cadena en función de la dirección anterior, es necesario fijar una dirección origen. La dirección de giro que se busca es horaria, por lo que la dirección de cadena de origen se fija como el eje positivo de las  $x$  o dirección  $\theta$ .

En la Figura C.2 se muestra el histograma de código de cadena de un *superpixel*. Las direcciones dominantes de los *superpixels* son a su vez, dominantes en el histograma. En el caso de la Figura C.2 (a), se observa la forma claramente alagada del *superpixel*, siendo claramente predominantes las direcciones 2 y 6 de la conectividad de 8 grados (ver Figura C.1 (a)). En la Figura C.2 (b) el predominio es de las direcciones verticales 2 y 6, junto con las diagonales 3 y 7 (ver C.1 (a)).

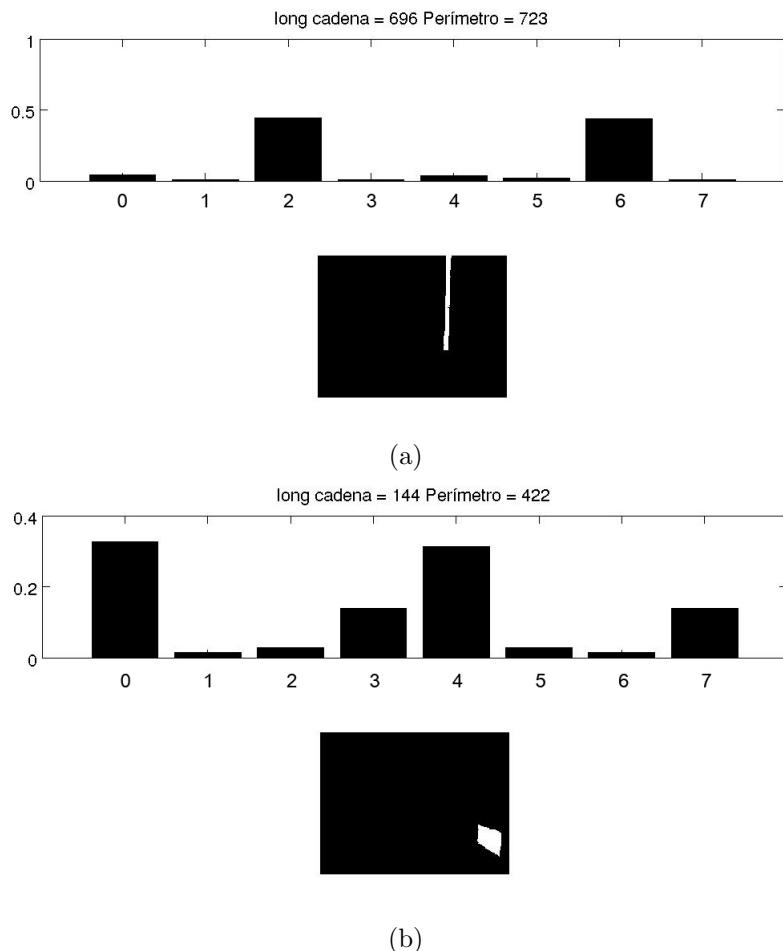


Figura C.2: (a) Ejemplo de histograma de código de cadena del *superpixel* con forma alargada mostrado en la imagen (a) inferior; (b) Ejemplo de histograma de código de cadena del *superpixel* con forma no alargada mostrado en la imagen (b) inferior;



## Anexo D

# Distancia EMD entre histogramas

El histograma  $histf(i) = n_i$  de una imagen  $f$  es muy utilizado en la práctica como representación compacta de su distribución en cualquiera de sus componentes, por ejemplo el nivel de gris, i.e., el valor  $n_i$  para cada nivel  $i$  es el número de píxeles de la imagen que tienen el valor de gris  $i$ , discretizando el numero de posibles valores de  $i$  al numero de barras del histograma [25]. Para la posterior utilización de los histogramas, necesitamos escalarlos y normalizarlos. Para ello utilizamos una escala de gris de 256 niveles y normalizaremos por el número total de píxeles de forma que obtengamos una probabilidad entre 0 y 1.

Es una práctica común el uso de distancias como  $L_2$  o  $\chi^2$  para comparar histogramas. Esta práctica asume que los dominios de los histogramas están alineados. Sin embargo, esto no es así en casos de deformación de la imagen, cambios de luz, etc.

La distancia EMD (Earth Mover's Distance) fue presentada en [26] como una manera de medir la coincidencia entre histogramas. Es definida como el mínimo coste que debe ser pagado para transformar un histograma en otro, donde existe una distancia de base entre las características básicas que están contenidas en el histograma.

Sea  $P = \{(\mathbf{p}_1, \omega_{p_1}), \dots, (\mathbf{p}_m, \omega_{p_m})\}$  la representación de un histograma con  $m$  niveles, donde  $\mathbf{p}_i$  es el nivel representativo y  $\omega_{p_i}$  el tamaño o peso que tiene ese nivel de gris en la imagen. Sea  $Q = \{(\mathbf{q}_1, \omega_{q_1}), \dots, (\mathbf{q}_n, \omega_{q_n})\}$  el segundo histograma con  $n$  niveles de gris. Dados dos histogramas  $P, Q$ , sea su matriz de distancia de base  $\mathbf{D} = [d_{ij}]$  donde  $d_{ij}$  es la distancia de base entre los niveles  $\mathbf{p}_i$  y  $\mathbf{q}_j$ . La distancia  $d_{ij}$  puede ser cualquiera, en este caso se ha utilizado la distancia Euclídea. Buscamos encontrar un flujo  $\mathbf{F} = [f_{ij}]$  donde  $f_{ij}$  es el flujo entre  $\mathbf{p}_i$  y  $\mathbf{q}_j$ , que minimiza el coste total.

La distancia EMD es definida como:

$$EMD(P, Q) = \frac{\sum_{i=1}^m \sum_{j=1}^n d_{ij} f_{ij}}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}} \quad (\text{D.1})$$

sujeto a:

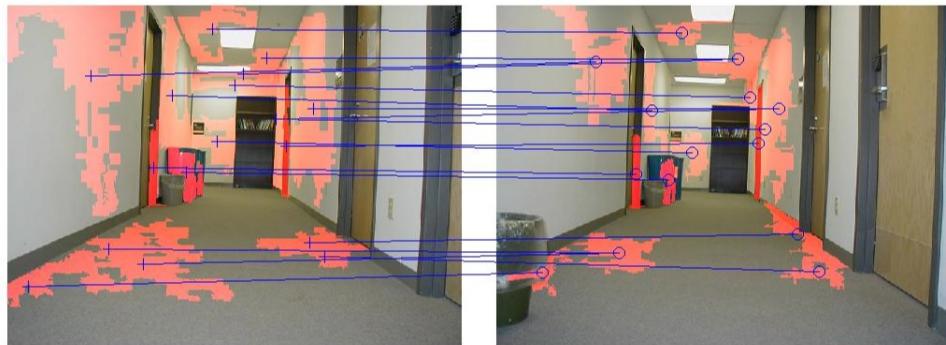
$$\begin{aligned} f_{ij} &\geq 0 \quad 1 \leq i \leq m \\ \sum_{j=1}^n f_{ij} &\leq \omega_{p_i} \quad 1 \leq i \leq m \\ \sum_{i=1}^m f_{ij} &\leq \omega_{q_j} \quad 1 \leq j \leq n \\ \sum_{i=1}^m \sum_{j=1}^n f_{ij} &= \min(\sum_{i=1}^m \omega_{p_i}, \sum_{j=1}^n \omega_{q_j}) \end{aligned}$$

La condición (1) permite flujo desde **P** hacia **Q** y no al revés. La condición (2) impide que el flujo saliente de **P** sea mayor que su tamaño mientras que la condición (3) impide que el flujo entrante en **Q** sea mayor que su tamaño. La condición (4) fuerza a mover el mayor flujo posible.

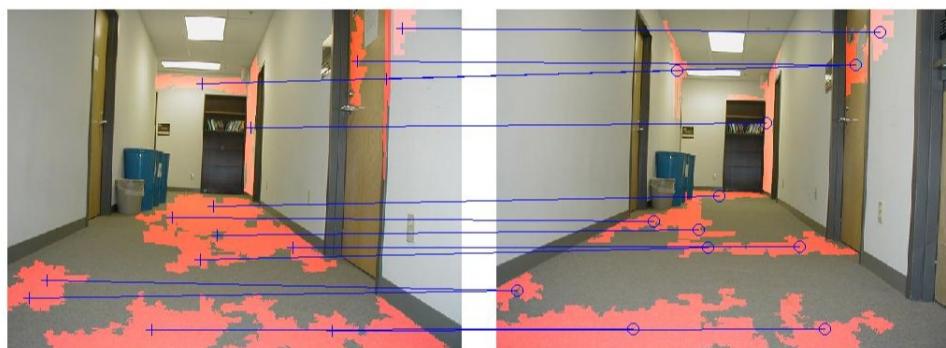
La distancia EMD ha sido utilizada para comparar histogramas de imágenes completas o directamente los píxeles de las imágenes. Según [27] el primer método no es lo suficientemente discriminativo ya que la información geométrica en la imagen se pierde, y el segundo es demasiado complejo. Propone, por lo tanto una aplicación intermedia calculando los histogramas en sub-regiones de la imagen, en concreto *superpixels*.

El código utilizado para el cálculo de las distancias EMD es el de [28], esta librería devuelve el valor de la distancia EMD y la matriz de flujos **F**. En general, la distancia de base  $d_{ij}$  puede ser cualquier distancia. [28] propone utilizar filtrar las distancias por encima de un umbral determinado, i.e., las distancias saturan a un valor constante. La matriz de saturación que se ha generado fija la distancia de base es 0 para niveles de gris equivalentes, 1 para adyacentes y 2 para los más alejados. Se muestra cómo distancias de base no saturadas afectan negativamente al desarrollo del cálculo.

La distancia EMD ha sido utilizada para establecer correspondencias entre *superpixels* de una imagen  $i$  y la siguiente imagen  $i+1$ . Como podemos observar en la Figuras D.1 y D.2, los resultados sugieren una alta fiabilidad de la propagación basada en este método al discriminar aquellas relaciones cuya distancia es mayor que un umbral.



Correspondencias entre superpixels de las imágenes 1 y 2

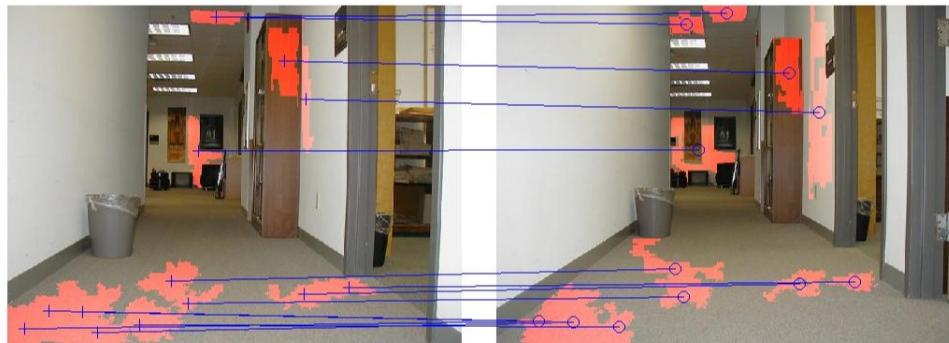


Correspondencias entre superpixels de las imágenes 2 y 3

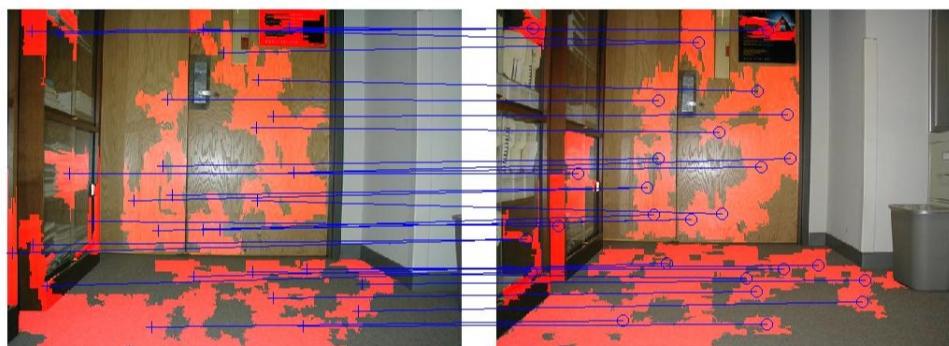


Correspondencias entre superpixels de las imágenes 13 y 14

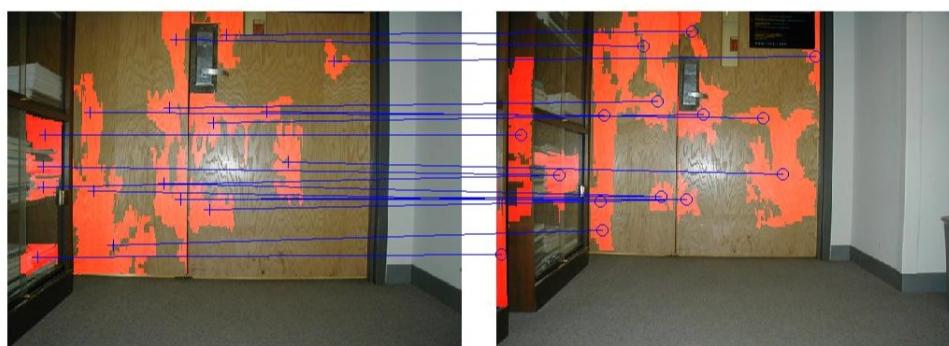
Figura D.1: Correspondencias establecidas entre *superpixels* de pares de imágenes consecutivas utilizando la distancia EMD



Correspondencias entre superpixels de las imágenes 14 y 15



Correspondencias entre superpixels de las imágenes 39 y 40



Correspondencias entre superpixels de las imágenes 40 y 41

Figura D.2: Correspondencias establecidas entre *superpixels* de pares de imágenes consecutivas utilizando la distancia EMD

## **Anexo E**

## **CD-ROM**

El archivo readme.txt indica la situación de los distintos archivos contenidos en el CD adjunto.



# Bibliografía

- [1] C. Stachnis, O. Martinez-Mozos, A. Rottman, and W. Burgard, “Semantic labeling of places,” in *ISRR*, 2005.
- [2] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen, “A discriminative approach to robust visual place recognition,” in *IROS*, 2006.
- [3] D. Anguelov, D. Koller, E. Parker, and S. Thrun, “Detecting and modelling doors with mobile robots,” in *ICRA*, 2004.
- [4] B. Limketkai, L. Liao, and D. Fox, “Relational object maps for mobile robots,” in *IJCAI*, 2005.
- [5] I. Posner, M. Cummings, and P. Newman, “Fast probabilistic labeling of city maps,” in *RSS*, 2008.
- [6] B. Douillard, D. Fox, and F. Ramos, “A spatio-temporal probabilistic model for multi-sensor multi-class object recognition,” in *RSS*, 2007.
- [7] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski, “Self-supervised monocular road detection in desert terrain,” in *RSS*, 2006.
- [8] A. Opelt, A. Pinz, and A. Zisserman, “Incremental learning of object detectors using a visual shape alphabet,” in *CVPR*, 2006.
- [9] J. Shotton, J. Winn, C. Rother, and A. Criminisi, “Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation,” in *ECCV*, 2006.
- [10] F. Schroff, A. Criminisi, and A. Zisserman, “Single-histogram class models for image segmentation,” in *ICVGIP*, 2006.
- [11] A. P. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones, “Superpixel lattices,” in *CVPR*, 2008.
- [12] F. Drucker and J. MacCormick, “Fast superpixels for video analysis,” in *Proceedings of the international conference on Motion and video computing*, 2009.
- [13] P. F. Felzenszwalb and D. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.
- [14] A. Vedaldi and B. Fulkerson, “Vlfeat—an open and portable library of computer vision algorithms,” 2010.

- [15] X. Ren and J. Malik, “Learning a classification model for segmentation,” *ICCV*, 2003.
- [16] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEECS*, 1999.
- [17] D. Nistér and H. Stewénius, “Scalable recognition with a vocabulary tree,” in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2006, pp. 2161–2168.
- [18] A. C. Murillo, J. Košecká, B. Micusik, C. Sagüés, and J. J. Guerrero, “Weakly supervised labeling of dominant image regions in indoor sequences.” in *Workshop Vision in Action: Efficient strategies for cognitive agents in complex environments, held together with ECCV*, 2008.
- [19] T. Werner, “A linear programming approach to max-sum problem: A review,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 7, pp. 1165–1179, July 2007.
- [20] M. W. Schwarz, W. B. Cowan, and J. C. Beaty, “An experimental comparison of rgb, yiq, lab, hsv, and opponent color models,” *ACM Transactions on Graphics*, 1987.
- [21] R. M. Haralick, “Statistical and structural approaches to texture,” in *Proc. of the IEEE*, 1979.
- [22] R. Sapina, “Computing textural features based on co-occurrence maatrix for infrared images,” *IEEE Conference Proceedings*, 2001.
- [23] H. Freeman, “On the encoding of arbitrary geometric configurations,” in *Institute of Radio Engineers, Transactions on Electronic Computers*, 1961.
- [24] Y. K. Liu and B. Zalil, “An efficient chain code with huffman coding,” *Pattern Recognition*, 2004.
- [25] J. Angulo and J. Serra, “Image color segmentation using bi-variate histograms in luminance/saturation/hue polar color spaces,” *Computación y Sistemas*, 2005.
- [26] Y. Rubner, C. Tomasi, and L. J. Guibas, “The earth mover’s distance as a metric for image retrieval,” *International Journal of Computer Vision*, 2000.
- [27] S. Boltz, F. Nielsen, and S. Soatto, “Earth mover distance on superpixels,” *Proceedings of the International Conference on Image Processing*, September 2010.
- [28] O. Pele and M. Werman, “Fast and robust earth mover’s distances,” *ICCV*, 2009.