Reconocimiento automático de áreas de interés en secuencias de interiores

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

Resumen

El procesado automático de imágenes y secuencias de imágenes es una necesidad que aparece por la gran cantidad de información que se puede recopilar actualmente en forma de imágenes y vídeos. El tratamiento manual de tal cantidad de información resulta imposible. Este trabajo se centra en la detección y clasificación de objetos y/o regiones de interés en secuencias de imágenes tomadas en ambientes de interior. La idea es que el procesado de las secuencias se realice de manera semiautomática, el usuario solo actúa sobre la primera imagen de la secuencia, procesándose las demás de un modo autónomo.

En particular, este trabajo se centra en el procesado automático de secuencias de interiores adquiridas por un robot móvil. En este entorno, el tipo de regiones y objetos que vamos a detectar son: regiones características principales para la navegación en secuencias de interior, como suelo, pared y techo; objetos importantes para la navegación, como puertas; el resto de objetos que pertenezcan a otras clases han sido asociados a un grupo genérico.

Para el procesamiento automático de una secuencia, se han desarrollado los siguientes módulos partiendo de un proceso base inicial, respecto del cual todos los pasos han sido rediseñados para mejorar los resultados:

- Segmentación de las imágenes. Las imágenes que forman la secuencia son segmentadas en conjuntos de píxeles con características similares, los cuales son llamados *superpixels*. Se han estudiado distintos métodos de segmentación de imágenes. Se ha desarrollado e implementado un método de evaluación y comparación entre las imágenes segmentadas mediante cada uno de los métodos, y se ha elegido el más adecuado.

- Descripción de cada segmento. Se ha realizado un estudio de posibles descriptores de las características de cada uno de los segmentos que forman la imagen. Los descriptores se pueden dividir en cuatro grandes grupos: de color, textura, forma y posición.

- Modelado de las regiones a detectar. Se ha diseñado un modelo del entorno, el cual se inicializa utilizando los descriptores de unas cuantas regiones identificadas a mano en el primer fotograma de la secuencia, y se va actualizando automáticamente con las medidas que se obtienen de los siguientes fotogramas.

- Cada uno de los segmentos que forman la imagen es comparado con los grupos que componen el modelo y se estima la probabilidad que tiene cada segmento de pertenecer a cada una de las regiones u objetos a detectar. Finalmente, el procesado de cada fotograma, incluye un filtrado que tiene en cuenta tanto la probabilidad de cada segmento de pertenecer a un objeto/región como la relación de el segmento con los segmentos vecinos.

Índice

Resumen **i**

Índice **ii**

1 Introducción **1**
1.1 Objetivos 1
1.2 Trabajo previo 2
1.3 Proceso de reconocimiento diseñado 3

2 Representación de las imágenes **5**
2.1 Segmentación de la imagen. *Superpixels* 5
2.1.1 Métodos de segmentación basados en *superpixels* 5
2.1.2 Comparación de los distintos métodos de segmentación . 6
2.2 Descriptores 8

3 Modelado y reconocimiento **11**
3.1 Inicialización del modelo del entorno 11
3.1.1 Método 1 13
3.1.2 Método 2 13
3.1.3 Método 3 13
3.2 Relación imagen-modelo 14
3.2.1 Análisis individual de cada *superpixel:* distancias *superpixel*-modelo. 14
3.2.2 Análisis de cada imagen en conjunto: optimización de las asignaciones a cada superpixel. 16
3.3 Actualización del modelo 17
3.3.1 Criterios para establecer correspondencias entre *superpixels* 18
3.4 Método de actualización 19
3.4.1 Creación de nuevos *clusters* 20

4 Experimentos **21**
4.1 Métodos de evaluación 21
4.2 Evaluación del proceso de actualización 21
4.3 Evaluación las distancias *superpixel*-modelo 23
4.4 Comparación de resultados según el tamaño de los *superpixels* . . 26
4.5 Configuración final propuesta 28

5 Conclusiones **37**
5.1 Conclusiones personales 37
5.2 Conclusiones trabajo 37
5.3 Trabajo Futuro 38

Anexos **40**

A Comparación del tamaño de los *superpixels* **41**

B Descriptores **45**
B.1 Descriptores de color 45
B.2 Descriptores de textura 48
B.3 Descriptores de forma 50
B.4 Descriptores de posición 51

C Códigos de cadena **53**

D Distancia EMD entre histogramas **57**

E CD-ROM **61**

Bibliografía **63**

Capítulo 1

Introducción

Actualmente, es muy común conseguir acceso o adquirir grandes galerías de imágenes y vídeos, los cuales es prácticamente imposible procesar de manera manual. Este hecho lleva consigo la necesidad del procesamiento e interpretación de la información capturada en imágenes y vídeos de un modo lo más automático posible. La detección y clasificación de objetos y/o regiones de interés es uno de los objetivos importantes en este área. En este proyecto se ha trabajado en la detección/clasificación de objetos y regiones presentes en secuencias de imágenes tomadas en ambientes interiores, de manera semiautomática ya que la única supervisión del usuario se lleva a cabo en la primera imagen de la secuencia.

El objetivo general en nuestro trabajo es distinguir e identificar las áreas y objetos que ocupan la mayor parte de las imágenes (en nuestro caso suelo, pared y puertas) y agrupar en "otros" el resto de zonas "minoritarias" que suelen representar otro tipo de objetos mas pequeños.

El interés de separar estas zonas dominantes del resto es múltiple. Por un lado, obtener elementos básicos para la navegación autónoma de un robot y por otro lado restringir y reducir el procesado de reconocedores de muchos otros objetos mas pequeños. Por ejemplo, a la hora de buscar personas en las secuencias, solo habría que procesar las zonas de "otros" y que ademas cumplan ciertas restricciones (como que las personas estén aproximadamente en contacto con el suelo).

1.1 Objetivos

Los objetivos y tareas concretos planteados en este proyecto son:

- Estudio de diferentes tipos de segmentación de imágenes. Desarrollo de un método de evaluación y comparación para la selección del más adecuado para las siguientes tareas a realizar con ellos.

- Estudio de descriptores de los distintos segmentos en los que se divide la imagen. En primer lugar, implementar tanto descriptores conocidos de

la literatura como nuevas propuestas o adaptaciones adecuadas al tipo de imágenes y entornos en los que vamos a trabajar.

- Diseñar distintas medidas de similitud que permitan comparar descriptores de distintos segmentos.

- Diseño y realización de experimentos exhaustivos con secuencias de imágenes realistas para evaluar los descriptores y distancias. El objetivo es encontrar los que mejor discriminan entre segmentos correspondientes a los distintos conceptos, clases y objetos que se quieren reconocer o identificar en la secuencia.

- Estudiar distintos tipos de técnicas de clasificación para reconocer y etiquetar a qué objeto, zona o concepto pertenecen los distintos segmentos de las imágenes, utilizando los descriptores seleccionados. Implementar el método de clasificación más adecuado para realizar experimentos realistas con secuencias reales de entornos de interior y evaluar la corrección de los resultados de reconocimiento.

- Documentar los estudios, el código implementado y experimentos realizados, los cuales, si son satisfactorios se redactarán en forma de artículo de investigación.

1.2 Trabajo previo

En el campo de la robótica, se han obtenido resultados muy interesantes en los últimos años respecto a la clasificación y reconocimiento automático de lugares, utilizando distintos tipos de sensores [1], [2]. Más recientemente, ha surgido el interés de aumentar la representación que los robots autónomos tienen de su entorno (mapas) con información semántica para facilitar la autonomía de los sistemas y la interacción humano-robot. El problema ha sido tratado desde diferentes puntos de vista. En [3] los autores abordan el problema de la obtención de un modelo del entorno definiéndolo con representaciones de objetos de las clases predefinidas (puertas, paredes) dado un rango de datos e imágenes a color de una cámara omnidireccional. En [4] los autores recurren a redes de Markov para clasificar imágenes segmentadas. Las regiones del entorno son clasificadas como pared, puerta y otros, basándose en datos adquiridos por láser.
Más recientemente, en el contexto de las técnicas de etiquetado denso de las imágenes, se han obtenido muy buenos resultados integrando varios sensores tanto visuales como láser, para el reconocimiento de objetos y regiones [5], [6]. También ha aumentado el interés en procesar secuencias de imágenes, ya que en los ámbitos de robótica es la manera natural de obtener la información. Esto implica un procesado de la información conforme va llegando, lo cual sugiere construir modelos que se adapten con el tiempo y aprovechen una serie de restricciones temporales. Por ejemplo en [7] se propone un modelo que se va actualizando conforme el vehículo avanza para clasificaciones de carretera/no carretera utilizando información visual. En [8] se presenta un modelo para reconocer objetos que se puede ir actualizando on-line, también a partir de información visual.
Por otro lado, encontramos otro grupo de trabajos sobre segmentación de imágenes

en visión por computador muy relacionados con nuestros objetivos. Se han propuesto distintas técnicas de segmentación de imágenes que permiten reducir el numero de elementos a procesar (en vez de cada píxel, se trabaja con cada segmento) y facilitan el poder tener un resultado final en el que se clasifiquen de manera densa todos los píxeles sin un coste computacional demasiado alto [9], [10].

1.3 Proceso de reconocimiento diseñado

En esta sección se resumen todos los pasos del proceso diseñado. En la Figura 1.1 se puede ver un diagrama del proceso propuesto de reconocimiento de áreas de interés en secuencias.

El primer paso es la inicialización de un modelo del entorno que queremos reconocer/interpretar, a partir de la primera imagen de la secuencia. La imagen se carga, tras lo cual se segmenta tal y como se explica en la sección 2.1. Las propiedades de cada segmento de la imagen son caracterizadas por medio de los descriptores, los cuales están detallados en la sección 2.2. Tras esto, los segmentos son agrupados según sus características, el usuario realiza un etiquetado manual de inicialización y se crea el modelo del entorno tal y como se expone en la sección 3.1.

Una vez se ha generado el modelo a partir de la primera imagen y el etiquetado manual de ejemplo en ella, para conseguir un etiquetado denso de todas las zonas de la primera imagen, se realiza una comparación de los segmentos de la imagen con los grupos del modelo y se estima la probabilidad de cada segmento de pertenecer a cada uno tal y como se explica en la sección 3.2. Finalmente, una vez que hemos clasificado los elementos de una imagen según el modelo que tenemos, aplicamos un modelo gráfico que tiene en cuenta las distintas conexiones entre segmentos contiguos en la imagen para optimizar la clasificación o etiquetado de todas las partes de la imagen en bloque. El modelo gráfico que utilizamos son los "Markov Random Fields" (MRF), que se detallan en la sección 3.2.2, tras el cual se obtiene la primera imagen de la secuencia etiquetada completamente (todos los píxeles).

Una vez que hemos procesado por completo el primer fotograma de la secuencia, se pasa a procesar los siguientes fotogramas. Los pasos para procesar cada uno son parecidos a los del primero, pero sin necesidad ya de ningún etiquetado manual: se carga la imagen, se segmenta, se calculan los descriptores de cada segmento y se asignan las probabilidades de cada segmento de pertenecer a cada una de las clases que estamos analizando. Esta asignación, al igual que antes, se hace en un primer paso que depende del modelo y un segundo paso que intenta optimizar globalmente todas las asignaciones en la imagen. El paso nuevo en todos los frames a partir de ahora es la actualización del modelo con las características de los segmentos una vez clasificados, lo cual está detallado en la sección 3.3.

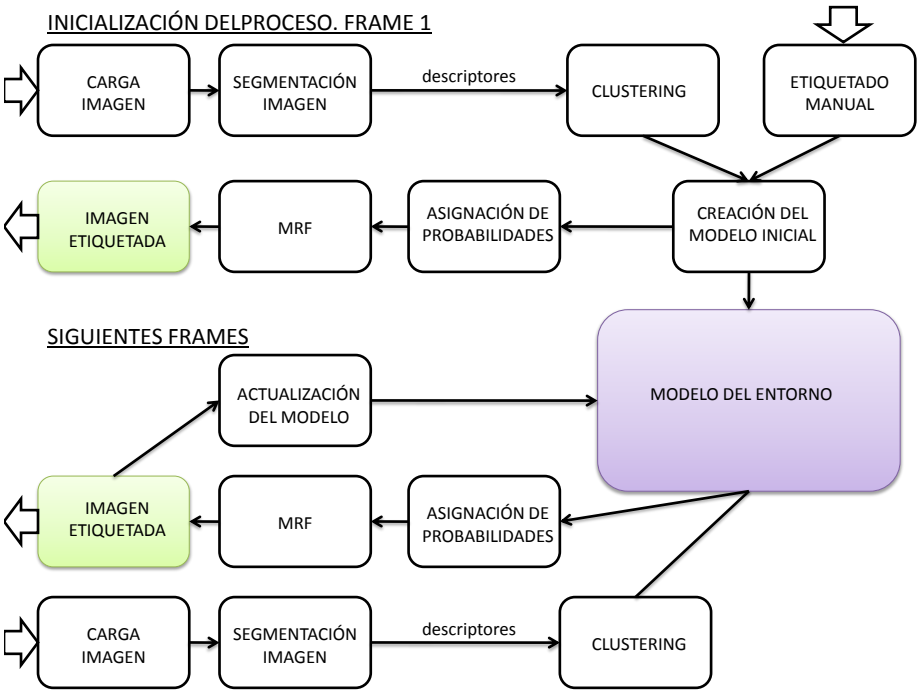


Figura 1.1: Esquema del proceso de reconocimiento (etiquetado) de áreas de interés en secuencias de interior

Capítulo 2

Representación de las imágenes

2.1 Segmentación de la imagen. *Superpixels*

La representación de una imagen por medio de píxeles es, en muchas ocasiones, redundante ya que los objetos de interés están compuestos por multitud de píxeles similares, esto lleva a malgastar recursos computacionales [11]. Los *superpixels* son regiones contiguas de una imagen perceptualmente similares. Idealmente, todos los píxeles dentro de un mismo *superpixel*, pertenecen al mismo objeto del mundo real. La segmentación de imágenes utilizando *superpixels* tiene el potencial de reducir de manera notable el coste del análisis automático de imágenes, ya que disminuye el número de elementos analizados por imagen, de miles de píxeles a cientos de *superpixels* [12]. Además, los *superpixels* pueden incorporar, implícitamente, información de la forma del objeto, y nos delimitan qué píxeles debemos procesar cojuntamente.
Es importante que el método utilizado para la segmentación tenga las siguientes características [13]:

- Captura de los grupos o regiones perceptualmente importantes, que reflejan los aspectos globales de la imagen.
- Ser eficiente, trabajando en un tiempo aproximadamente lineal con respecto al número de píxeles de la imagen. Esto es especialmente importante si el proceso se quiere utilizar en análisis de vídeo.

2.1.1 Métodos de segmentación basados en *superpixels*

De entre las muchas posibilidades existentes, se han estudiado los siguientes métodos de segmentación por ser de los más habituales en los trabajos recientes de visión por computador, además de por que ofrecen las librerías para su extracción.

Efficient Graph-Based Image Segmentation [13] Este método utiliza una aproximación basada en grafos, mide las evidencias de los contornos entre

dos regiones de la imagen comparando dos cantidades: una basada en diferencias de intensidad en los contornos de las regiones de la imagen, y la segunda basada en las diferencias de intensidad entre los píxeles vecinos dentro de cada región (Figura 2.1).

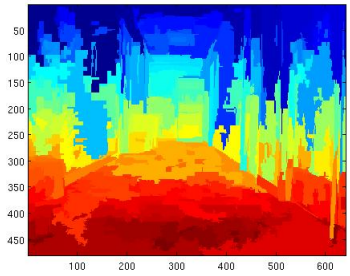


Figura 2.1: Imagen segmentada con el método *"Efficient Graph-Based Image Segmentation"*, código utilizado [13]

Quick Shift [14] Es un algoritmo de búsqueda de modelos basado en métodos no paramétricos. Utiliza un árbol de enlaces de cada píxel con los píxeles cercanos para calcular densidades, diferenciarlos y agruparlos en *superpixels* (Figura 2.2).

Figura 2.2: Imagen segmentada con el método *Quick Shift* código [14]

***Normalized Cuts Jerárquico* [15]** Este algoritmo de segmentación de *superpixels* utiliza el método *Normalized Cuts* [16] teniendo en cuenta aspectos de textura y contorno. Se obtienen tres posibles soluciones en distintos pasos (Figura 2.3) de este método jerárquico.

2.1.2 Comparación de los distintos métodos de segmentación

Para cuantificar la calidad de los distintos métodos de segmentación del la imagen, se ha diseñado una medida basada en contornos, comparando los contornos

(a) (b) (c)

Figura 2.3: Segmentación de una imagen con el método Normalized Cuts Jerárquico al final del primer(a), segundo (b) y ultimo (c) paso del algoritmo.

método de segmentación	tiempo [s]	Sobel	LoG	Canny
Geometric Context	0.63	23.71	19.62	16.31
VLfeat	9.29	39.03	17.12	17.44
Normalized Cuts (a)	246.07	21.97	14.52	10.12
Normalized Cuts (b)	246.07	21.77	16.65	10.92
Normalized Cuts (c)	331.40	21,.47	16.26	10.91

Tabla 2.1: Comparación de tiempo de extracción y de coincidencia de contornos de los métodos de segmentación

que se pueden extraer de la imagen de *superpixels* **S** (por ejemplo las de las figuras 2.1 o 2.2) con los contornos que se extraen de la imagen original **O**. En concreto se calcula el porcentaje de píxeles de contorno de la imagen original **O** que son "cubiertos" por píxeles de contorno en la imagen de *superpixels* **S**.

$$coincidencia\ de\ contornos\ = \frac{\sum p\acute{i}xel \in \mathbf{S} \cap \mathbf{O}}{\sum p\acute{i}xel \in \mathbf{O}} \tag{2.1}$$

Los contornos de la imagen pueden ser obtenidos mediante algoritmos específicos o ser marcados por una persona. En nuestro caso se han utilizado tres de las variantes (*sobel*, *LoG* y *Canny*) de la función de Matlab para la obtención de los contornos *edge*.

Los resultados de la comparación son los mostrados en la tabla 2.1, en la que la columna "tiempo" indica el tiempo empleado por cada método en la segmentación de la imagen en segundos, y las columnas "Sobel", "LoG" y "Canny" representan la coincidencia de contornos utilizando el correspondiente método para extraer los contornos según la ecuación 2.1. Las tres filas correspondientes al método "Normalized Cuts Jerárquico" indican si nos quedamos con el resultado al ejecutar solo un paso, al ejecutar dos, o los tres pasos del método completo.

Los resultados de coincidencia de contornos son ligeramente superiores utilizando la librería *VLfeat*, que emplea el método *Quick Shift*. Analizando las imágenes resultantes, en la figura 2.3(b) se obtienen unos *superpixels* con formas más regulares y definidas, además se observa direccionalidad de los *superpixels* en algunas zonas, la cual podría ayudar al reconocimiento de las regiones locales,

pero el tiempo de cálculo es considerablemente mayor que el resto de métodos. Por lo tanto, debido que el tiempo de cálculo es un parámetro importante en nuestro caso, el método que va a ser utilizado es el primero, ya que es el mejor compromiso velocidad-coincidencia de contornos.

2.2 Caracterización de los *superpixels*. Descriptores.

De acuerdo con la percepción humana, los colores dominantes, las texturas y las formas tienen una función determinante para distinguir diferentes objetos o regiones. Tratando de reproducir estos factores a la hora de interpretar las imágenes automáticamente, se han implementado una serie de descriptores, que consisten en una serie de estadísticas calculadas sobre los distintos valores de color e intensidad que tengan los píxeles de un *superpixel*. Utilizando la información que nos aportan de cada *superpixel*, se pueden comparar y agrupar los *superpixels* que componen las imágenes en grupos con características similares.

Los descriptores utilizados pueden ser divididos en cuatro clases, **color**, **textura**, **forma** y **posición**. En el Anexo B se encuentra la definición detallada de todos los descriptores utilizados.
Los espacios de color utilizados son RGB, HSV y Lab, siendo, en el caso del espacio *RGB C1=R, C2=G, C3=B*; en el caso de *HSV C1=H, C2=S, C3=V*; y en el caso de *Lab C1=L, C2=a, C3=b*. En total se obtienen un máximo de 20 descriptores x 3 espacios de color = 60 descriptores de color.

Tabla 2.2: **Descriptores de color** estudiados, donde C1, C2 y C3 significan, respectivamente, los canales 1, 2 y 3 del espacio de color utilizado. Todos los descriptores se calculan tanto en RGB como en HSV y Lab.

Dimensión	Descriptor	Expresión
9	nivel medio de intensidad (canal C1,C2,C3)	ecuación B.4
3	nivel medio de intensidad ($\frac{C1+C3+C4}{2}$)	ecuación B.4
9	coeficiente de variación (canal C1,C2,C3)	ecuación B.6
3	coeficiente de variación ($\frac{C1+C2+C3}{2}$)	ecuación B.6
9	coeficiente de asimetría (canal C1,C2,C3)	ecuación B.7
3	coeficiente de asimetría ($\frac{C1+C2+C3}{2}$)	ecuación B.7
9	coeficiente de curtosis (canal C1,C2,C3)	ecuación B.8
3	coeficiente de curtosis ($\frac{C1+C2+C3}{2}$)	ecuación B.8
9	entropía (canal C1,C2,C3)	ecuación B.9
3	entropía ($\frac{C1+C2+C3}{2}$)	ecuación B.9
	Número total de descriptores	60

Los descriptores de textura, forma y posición están resumidos en las tablas

Tabla 2.3: **Descriptores de textura** estudiados

Dimensión	Descriptor	Expresión
1	Segundo Momento Angular (ASM)	ecuación B.11
1	Entropía de segundo orden	ecuación B.2
1	contraste	ecuación B.13
1	Momento Diferencial Inverso (IDM)	ecuación B.14
1	correlación	ecuación B.15
	Número total de descriptores	12

Tabla 2.4: **Descriptores de forma** estudiados

Dimensión	Descriptor	Expresión
1	área	Anexo B.3
1	perímetro	Anexo B.3
1	compacidad	ecuación B.3
1	excentricidad	ecuación B.19
1	orientación	ecuación B.20
8	histograma de códigos de cadena	Anexo B.3
	Número total de descriptores	13

Tabla 2.5: **Descriptores de posición** estudiados

Dimensión	Descriptor	Expresión
2	centroide (x,y)	ecuación B.21
5	mallado (cuadrante 1-5)	Anexo B.4
4	límites	Anexo B.4
1	horizonte	Anexo B.4
	Número total de descriptores	12

2.3, 2.4 y 2.5 respectivamente. El número total de descriptores por cada tipo es de 5 de textura, 13 de forma y 12 de posición. Se han realizado pruebas con todos estos descriptores pero en el diseño del proceso definitivo solo se utilizan los descriptores de color, textura y posición debido a un coste demasiado alto de los de forma, que no permitían procesar la secuencia en un tiempo razonable.

Capítulo 3

Modelado y reconocimiento de los objetos y áreas de interés

Con el fin de reconocer objetos y regiones de interés en los distintos fotogramas de una secuencia, es necesario definir de alguna manera las características que las identifiquen y diferencien unas de otras. Este capítulo describe el proceso diseñado e implementado para interpretar automáticamente el contenido de una secuencia.

Partiendo de los *superpixels* y sus descriptores, el primer objetivo es crear un modelo que almacene los descriptores típicos de las distintas regiones que aparecen en la secuencia [17]. La manera de inicializar este modelo es mediante el etiquetado manual hecho por el usuario, el cual debe señalar, en el primer fotograma de la secuencia, los objetos y regiones características que son de interés a lo largo del vídeo. A partir de este etiquetado manual de la primera imagen, se crea un modelo estadístico que sirve de relación entre los descriptores de los segmentos de la imagen y los conceptos que se quiere identificar. El modelo representa las clases típicas de *superpixels*. Cada una de estas "clases" será un grupo o *cluster* del modelo. Cada *cluster* tendrá una probabilidad de pertenecer a cada uno de los objetos o áreas a reconocer.

3.1 Inicialización del modelo del entorno

El primer paso es agrupar los *superpixels* que componen la primera imagen en grupos, que llamaremos *clusters*, del modelo. Para realizar este agrupamiento se utiliza el algoritmo de *clustering* k-means con los descriptores de los *superpixels* de esta imagen. Cada *cluster* tiene asociadas las características de los *superpixels* que lo componen, que definirán las características del propio *cluster*. En este caso se definen los descriptores de un *cluster* como la media de los descriptores de los *superpixels* que lo forman. Además de esto, cada *cluster* almacenará el número de *superpixels* que lo forman, así como el área total de los mismos. El conjunto de *clusters* obtenido van a componer el modelo inicial.

Una vez creados los *clusters* del modelo, hay que estimar las probabilidades

1	descriptores del *cluster* 1	número total en 1	áreas de 1
2	descriptores del *cluster* 2	número total en 2	áreas de 2
...	...	...	...
N	descriptores del *cluster* N	número en N	áreas de N

Tabla 3.1: Estructura del modelo del entorno

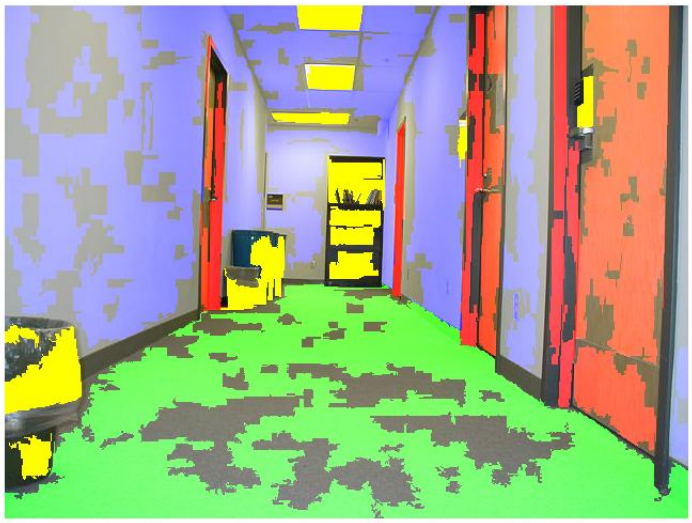


Figura 3.1: Primer fotograma de una secuencia con etiquetas de referencia asignadas por el usuario. Azul = pared/techo; Verde = suelo; Rojo = puertas; Amarillo = otros

de cada *cluster* de pertenecer a cada una de las etiquetas que se quieren identificar a lo largo de la secuencia. Para ello es necesario haber realizado un etiquetado manual de algunos ejemplos en el primer fotograma de la secuencia, donde el usuario indica qué regiones quiere identificar (Figura 3.1).

Cada *cluster* del modelo tendrá asociado un vector de probabilidades $\mathbf{P}$.

$$\mathbf{P}_j = \begin{vmatrix} P(C_j|E_1) \\ P(C_j|E_2) \\ ... \\ P(C_j|E_L) \end{vmatrix} \tag{3.1}$$

La probabilidad de que un *superpixel* que pertenece al *cluster* C_j corresponda a una determinada etiqueta E_l viene dada por el número de *superpixels* con clase asignada/conocida que haya en ese *cluster*. En base a esto se han implementado

tres métodos distintos para estimar dichas probabilidades. En todos ellos, como se verá mas adelante, se intenta evitar que se asignen probabilidades de 0% o 100% a ninguna clase, de manera "preventiva", ya que nunca se sabe qué más queda en la secuencia que pueda pertenecer a cierto *cluster*.

3.1.1 Método 1: asignación de probabilidades a cada *cluster* según el número de elementos etiquetados.

Como se ha comentado anteriormente, cada *cluster* almacena, además de la información de los descriptores, el número de *superpixels* que contiene. La probabilidad de que los *superpixels* de un *cluster* j pertenezcan a la región etiquetada como k viene dada por

$$P(C_j|E_k) = \frac{\sum N_{jk}}{\sum N_j} \tag{3.2}$$

Siendo N_{jk} los *superpixels* pertenecientes al cluster j etiquetados con la etiqueta k y N_i los *superpixels* almacenados en el *cluster* j (incluidos 4 *superpixels* "virtuales").
Se aumenta en un *superpixel* "virtual" cada una de las etiquetas para evitar probabilidades iguales a 0% o a 100% ya que así N_{jk} será al menos 1 para todo k y nunca será $N_{jk} = N_j$.

Uno de los principales problemas de este método es que, si un *cluster* tiene almacenados muchos *superpixels* pequeños etiquetados como A y uno solo grande etiquetado como B, la probabilidad global del *cluster* será mayor para la etiqueta A, generando un error en los *superpixels* parecidos al de etiqueta B.

3.1.2 Método 2: asignación de probabilidades a cada *cluster* ponderada por áreas.

Cada *cluster* almacena el área total de los *superpixels* que los forman. La probabilidad de que los *superpixels* de un cluster j pertenezcan a la región etiquetada como k viene dada por

$$P(C_j|E_k) = \frac{\sum A_{jk}}{\sum A_j} \tag{3.3}$$

Donde A_{jk} es el área de los *superpixels* pertenecientes al *cluster* j etiquetados como k y A_j es el área de los *superpixels* almacenados en el *cluster* j. Se añade un el área media de los *superpixels* de la imagen por cada una de las etiquetas para evitar probabilidades iguales a 0% o a 100%.

El problema de este método viene dado, al contrario que el anterior, por no tener en cuenta la información sobre el número de *superpixels* etiquetados.

3.1.3 Método 3: asignación de probabilidades a cada *cluster* ponderada por áreas y cantidad de *superpixels* en cada uno.

Como consecuencia a los problemas encontrados en los métodos anteriores, este último método tiene en cuenta la información correspondiente al número de

superpixels etiquetados, así como las áreas de los mismos. En este caso, la probabilidad de que los *superpixels* de un *cluster* j pertenezcan a la región etiquetada como k viene dada por

$$P(C_j|E_k) = \frac{\sum N_{jk} * \sum A_{jk}}{\sum N_j * \sum A_j} \tag{3.4}$$

Se añade el área media de los *superpixels* de la imagen por cada una de las etiquetas para evitar probabilidades iguales a 0% o a 100%.

3.2 Relación entre una imagen y el modelo del entorno

Una vez inicializado el modelo, se pasa al procesado de los siguientes fotogramas de la secuencia. El objetivo es el cálculo de un vector de probabilidades $\mathbf{P}(S_i|E_k)$ que represente la probabilidad de que el *superpixel* i pertenezca a cada una de las k etiquetas a identificar. Se busca relacionar cada *superpixel* de la nueva imagen con los "tipos" de *superpixel* del modelo. Esto puede ser definido como un problema de clasificación. Hay varios tipos de *superpixel* y se quiere ver la probabilidad de los nuevos *superpixels* de pertenecer a cada uno y clasificarlos como la clase o etiqueta del más probable.

Para ello se establece una medida basada en la distancia de cada *superpixel* con los *clusters* del modelo. Se han implementado y comparado tres métodos distintos, que se detallan en el apartado 3.2.1. Esta medida permite estimar la probabilidad de cada *superpixel* de manera individual de pertenecer a cada clase. Sin embargo, en la asignación final, teniendo en cuenta que en una escena real los *superpixels* tenderán a pertenecer al mismo objeto que sus vecinos, se aplica una optimización de las asignaciones en conjunto a todos los *superpixels* de una imagen, mediante un modelo grafico que optimiza los costes de hacer unas u otras asignaciones, según las probabilidades individuales de cada *superpixel* y según la consistencia respecto a los "vecinos" (apartado 3.2.2).

3.2.1 Análisis individual de cada *superpixel:* distancias *superpixel*-modelo.

Se han implementado tres métodos para medir la semejanza entre cada *superpixel*, de manera individual, y los *clusters* del modelo del entorno. El primer paso de cada uno de ellos consiste en calcular la distancia Euclídea (todos los componentes del vector de descriptores están normalizados entre 0-1) entre el vector de descriptores del *superpixel* i y los vectores de descriptores de todos los *clusters* del modelo. A continuación, hay que decidir a qué *clusters* del modelo se parece más el *superpixel* evaluado, y para asignar finalmente al *superpixel* un vector de probabilidades $\mathbf{P}_i$ de pertenecer a los distintos objetos o clases a reconocer.

Como acabamos de mencionar, los tres métodos están basados en el cálculo de la distancia Euclídea d_{ij}, definida en la ecuación 3.5.

$$d_{ij} = \sqrt{\sum_{k=1}^{M} (\mathbf{m_i}(k) - \mathbf{m_j}(k))^2} \tag{3.5}$$

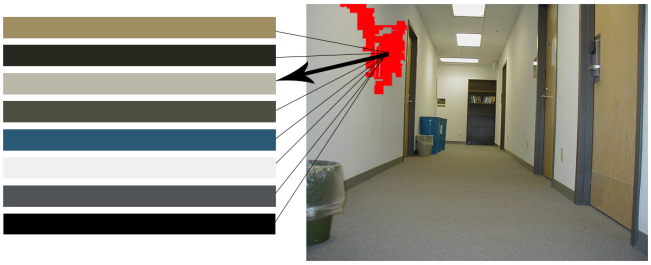


Figura 3.2: Representación gráfica del funcionamiento de los métodos 1 y 2, utilizando el método 1 la distancia Euclídea y el método 2 la exponencial de la distancia Euclídea

Donde d_{ij} es la distancia Euclídea entre el vector de descriptores del *superpixel* i y el vector de descriptores del *cluster* j, M es el número de descriptores utilizados por cada *superpixel*, $m_i(k)$ es el descriptor k del *superpixel* i y $m_j(k)$ es el descriptor k del *cluster* j.
A partir de la ecuación 3.5, se define el vector de distancias Euclídeas entre un *superpixel* y los *clusters* del modelo en la ecuación 3.6

$$\mathbf{d_i} = [d_{i1}; d_{i2}; ...; d_{iN}] \tag{3.6}$$

Donde $\mathbf{d_i}$ es el vector de distancias Euclídeas entre el *superpixel* i y los N *clusters* del modelo.

1. Asignación directa o simple Se considera que el *superpixel* i pertenece al *cluster* j cuya distancia Euclídea con i es la menor de todas. Por lo tanto, el vector de probabilidades del *cluster* es asignado directamente al *superpixel*, así que para cada posible clase o etiqueta k, la probabilidad del *superpixel* i será:

$$\mathbf{P}(S_i|E_k) = \mathbf{P}(C_j|E_k), \text{ donde } i \text{ y } j \text{ cumplen que } d_{ij} = min(\mathbf{d_i}) \tag{3.7}$$

2. Asignación ponderada En este caso la probabilidad de que un *superpixel* pertenezca a un *cluster* se calcula como la exponencial de la distancia de ese *superpixel* a ese *cluster* del modelo, ponderado con la desviación típica σ de ese *cluster* y con el parámetro δ que permite mantener las probabilidades en un rango no muy pequeño. Por lo tanto, una vez que estimamos que *cluster* es más probable de contener al *superpixel*, le asignamos las probabilidades de ser uno u otro objeto/etiqueta del *cluster* elegido:

$$\mathbf{P}(S_i|E_k) = P(C_j|E_k) \text{para cada una de las etiquetas k. Donde i y j} \tag{3.8}$$

cumplen que P_{ij}es la máxima de las posibles, donde $P_{ij} = \delta * \mathrm{e}^{\frac{-d_{ij}}{\sigma}}$

3. Asignación con V próximos En este ultimo método propuesto, en vez de considerar la menor distancia a la hora de relacionar el *superpixel* con el modelo, se van a tener en cuenta los V *clusters* con distancia Euclídea menor. Cuando uno o más de estos V *clusters* seleccionados tiene una distancia Euclídea

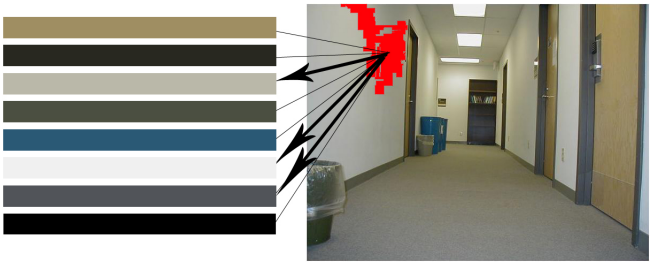


Figura 3.3: Representación del funcionamiento del método de relación con V próximos siendo en este caso V=3

con el *superpixel* evaluado mucho menor que la de los demás, los demás son descartados. A la hora del cálculo del vector de probabilidades se pondera teniendo en cuenta la distancia a cada uno de los V *clusters*, de manera que cuanto más cercano es el *cluster* j al *superpixel* i, más peso tendrá su vector de probabilidades $\mathbf{P}(C_j|E_k)$ en el vector resultante $\mathbf{P}(S_i|E_k)$.

$$\mathbf{P}(S_i|E_k) = \sum_{j=1}^{V} (\frac{\sum d - d_{ij}}{\sum d} * \mathbf{P}(C_j|E_k)) \tag{3.9}$$

donde d es la suma de las distancias de los V clusters con el superpixel evaluado.

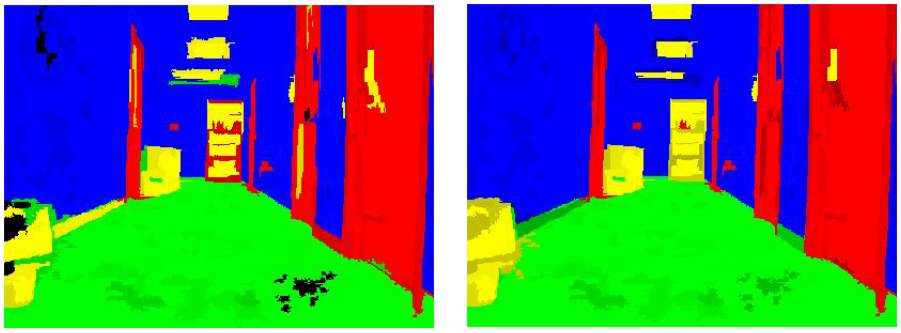
3.2.2 Análisis de cada imagen en conjunto: optimización de las asignaciones a cada superpixel.

Con el fin de tener en cuenta las relaciones espaciales entre *superpixels*, el problema de clasificación de los *superpixels* va a ser tratado en conjunto. Se busca optimizar la asignación de etiquetas teniendo en cuenta la información de toda la imagen. La formulación y resolución de esta optimización se realiza, de la misma manera que en [18] modelando la imagen, sus *superpixels* y sus relaciones de adyacencia con un Markov Random Field (MRF), un modelo gráfico no dirigido, donde cada *superpixel* es un nodo del diagrama, y cada relación de adyacencia se representa con una conexión en el diagrama/grafo.

Dados los vectores de probabilidades de cada uno de los *superpixels* que forman la imagen, este modelo gráfico valora el coste de asignar una u otra etiqueta a cada *superpixel*, teniendo en cuenta las probabilidades asignadas al *superpixel* de manera individual y las probabilidades de los *superpixels* vecinos.

El modelo gráfico considera que cada *superpixel* es un nodo y establece las conexiones entre nodos, que corresponden a *superpixels* adyacentes. Se asignan unos costes individuales a cada nodo basados en el vector de probabilidades $\mathbf{P}(S_i|E_k)$, así como unos costes binarios basados en la comparación con los *superpixels* y los vectores de probabilidades de estos últimos. Una vez establecidos todos los costes, se utiliza una librería estándar para resolver el MRF [19].

En la figura 3.4 se muestra la clasificación inicial de la imagen utilizando el modelo de probabilidades asignadas de manera individual a cada *superpixel* (a la izquierda) y la clasificación resultante después de la optimización del etiquetado



(a) (b)

Figura 3.4: (a) Etiquetado de la imagen sin utilizar MRF; (b) Etiquetado de la imagen utilizando MRF

mediante el MRF (derecha), que resulta en un etiquetado "suavizado", con cambios menos bruscos. Se han establecido dos restricciones para reducir errores en el etiquetado:

- Se ha establecido una línea de horizonte por encima de la cual ningún *superpixel* puede ser etiquetado como "suelo". La linea del horizonte se puede estimar mediante métodos estándar para la primera imagen, y mantenerlo en la misma posición para el resto. En las secuencias que usamos corresponde aproximadamente a 2/3 de la altura de la imagen.
- Los *superpixels* que están en contacto con el límite superior de la imagen, tampoco podrán ser etiquetados como "suelo".

3.3 Actualización del modelo

El avance de la secuencia conlleva determinados cambios en la posición y enfoque de los objetos y en la iluminación del entorno (Figura 3.5) entre otros. Estos cambios hacen que sea necesaria una actualización del modelo para que el modelo se adapte al entorno cambiante. Se han implementado cuatro métodos de actualización de los componentes del modelo según los criterios que se sigan, detallados en 3.3.1.

Figura 3.5: Ejemplo de variación de iluminación entre dos imágenes consecutivas

3.3.1 Criterios para establecer correspondencias entre *superpixels*

Conforme vamos identificando el contenido de los *superpixels* de los fotogramas tenemos que decidir cuáles son fiables para actualizar los *clusters* que ya existen en el modelo y sus descriptores en el modelo. Para tomar esta decisión hemos implementado los distintos métodos explicados a continuación que relacionan *superpixels* nuevos con los que ya están incluidos en el modelo.

Superposición de píxeles. Al estar trabajando con secuencias de imágenes parece razonable pensar que la probabilidad de que un píxel de una imagen y el píxel que se encuentra en la misma posición en la imagen inmediatamente posterior pertenezcan a la misma región característica es alta. Este razonamiento puede ser ampliado a los *superpixels*, de forma que dados un *superpixel* de una imagen i y un *superpixel* de una imagen *i+1*, si su intersección contiene un número de píxeles mayor del $PX\%$ de los píxeles totales de cada *superpixel*, se pueden considerar coincidentes y, por tanto, la probabilidad de que el *superpixel* de la imagen *i+1* pertenezca al mismo objeto o región que el de la imagen i será alta.

Siguiendo este criterio, el *cluster* al que pertenece el superpixel i será actualizado por los descriptores del *superpixel i+1*.

Se han realizado pruebas para valores de PX entre 60% y 90% y se ha observado que, aunque el planteamiento pueda parecer lógico, su funcionamiento no es robusto en muchas situaciones. Por ejemplo, ante cambios bruscos en la imagen el resultado de este tipo de actualización añade ruido al modelo. Ante la entrada o salida de objetos nuevos en la secuencia ocurre lo mismo. La variabilidad de forma de los *superpixels* también afecta negativamente al funcionamiento de este tipo de actualización. .

Método de distancia Euclídea. Este método calcula en primer lugar la distancia Euclídea entre el vector de descriptores de cada *superpixel* y los descriptores de cada *cluster* del modelo. Aquellos *superpixels* cuya distancia Euclídea mínima con uno de los *clusters* del modelo sea menor que el valor umbral de "aceptación" (ED), se utilizarán para actualizar el modelo.

El valor de ED es variable en función del máximo que pueda alcanzar la distancia Euclídea, dependiendo, por lo tanto, del número de descriptores. El umbral ED debe ser suficientemente estricto para no añadir ruido y bajar la representatividad de los clusters del modelo. Ha sido ajustado experimentalmente en un valor de $ED = \frac{número\ de\ descriptores}{100}$

Método de semejanza de color. En este método se propone que el modelo sea actualizado utilizando aquellos *superpixels* cuyos intensidades de color RGB son muy próximas a las de un *cluster* determinado. Para que el resultado de esta actualización no lleve a errores se debe ser muy estricto con la desviación aceptada, de esta forma solo los objetos con el mismo color en imágenes consecutivas se considerarán iguales. Se ha tomado como un valor suficientemente estricto 5/255, lo que equivale a variaciones de color menores del 1.96%.

Distancia EMD. La distancia EMD (Earth Mover's Distance) es una manera de medir la similitud entre histogramas. Esta distancia ha sido utilizada para establecer relaciones entre *superpixels* del fotograma *i* con el fotograma anterior *i-1* (Figura 3.6). En el anexo D se pueden ver algunos resultados de la aplicación de este tipo de distancia. Se ha establecido un umbral tal que, las distancias que están por encima del mismo, no se consideran a la hora de la actualización. La distancia EMD solo ha sido calculada para *superpixels* cuya área as mayor que un umbral, evitando así emparejamientos de *superpixels* demasiado pequeños.

El resultado de establecer correspondencias mediante este método es muy robusto y fiable. El principal problema radica en que al estar relacionando *superpixels* de una imagen con los de la imagen anterior, si el superpixel que se va a propagar tiene una etiqueta errónea o está ubicado en un cluster que no es el apropiado, la el método propagará el error. Debido a esto se debe estar muy seguro de que lo que se propaga es correcto antes de hacerlo, evitando umbrales demasiado bajos (>50%).

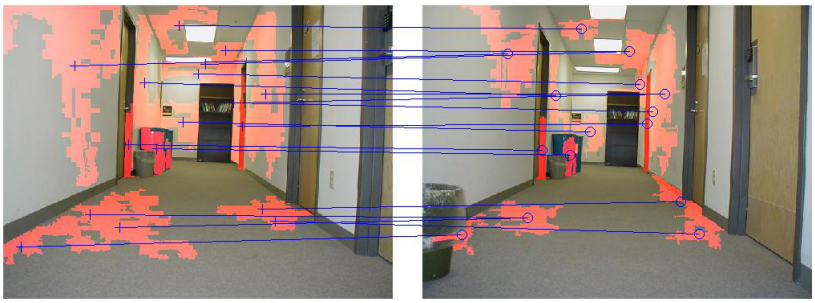


Figura 3.6: Ejemplo de la propagación de etiquetas por medio de la distancia EMD

3.4 Método de actualización

Una vez definidos los criterios para seleccionar con qué nuevos *superpixels* podemos actualizar *clusters* ya existentes, se procede a fijar la manera en la que el modelo va a ser actualizado con ellos.

Caso 1. El *superpixel* evaluado cumple uno o más criterios de actualización. En este caso, los descriptores del *cluster* al que dicho *superpixel* se ha asignado, son actualizados con los descriptores del *superpixel*, haciendo una media ponderada por su área.

Caso 2. El *superpixel* evaluado cumple el criterio de "Distancia EMD" y la probabilidad asociada a la etiqueta que le ha sido asignada es mayor del 70%. En este caso se considera muy fiable la asignación de la etiqueta, por lo que el *superpixel* pasa a formar parte del modelo al mismo nivel que los *superpixels* etiquetados a mano, modificando el vector de probabilidades del *cluster* según lo expuesto en la sección 3.1.3.

Caso 3. El *superpixel* no cumple ningún criterio de actualización. El *superpixel* puede formar un nuevo *cluster* según se expone en el siguiente apartado (3.4.1).

3.4.1 Creación de nuevos *clusters*

En los casos en los que se produzcan, por ejemplo, cambios bruscos de posición (giros de cámara en esquinas de pasillo, etc.), cambios bruscos de iluminación o aparición de nuevos objetos en la secuencia (personas, etc.) el modelo no va cubrir completamente las características de los *superpixels* de la imagen. En estos casos, el sistema debe ser capaz de reconocer estos cambios y adaptarse, en la medida de lo posible, a ellos. El primer paso es detectar aquellos *superpixels* que no se adaptan al modelo existente. Tras esto se agrupan entre sí según sus características. Finalmente se decide si los nuevos *clusters* son suficientemente representativos para formar nuevos clusters del modelo:

1- Detección de nuevos objetos Se considera que un *superpixel* no está representado por el modelo cuando no ha sido asignado a ningún *cluster* con los métodos anteriormente descritos (apartado 3.3.1).

2- Agrupación de los *superpixels* Se agrupan los *superpixels* no representados por el modelo utilizando el algoritmo de *clustering* k-means. Esta función agrupa los *superpixels* en *clusters*. El número de grupos o *clusters* que se forman dependerá del número de *superpixels* que haya.

3- Decisión de añadir al modelo A partir de los *clusters* resultantes de agrupar los *superpixels* no representados por el modelo, hay que valorar su representatividad para decidir si se incorporan al modelo o se desechan. Se ha definido un número máximo de *clusters* en el modelo. Si el número de *clusters* almacenados es menor que el máximo, los nuevos grupos se incorporan al modelo directamente. Si el número de *clusters* en el modelo es el máximo permitido, se realiza una comparación de los *clusters* candidatos con los *clusters* del modelo. En el caso de que un *cluster* candidato almacene mayor número de *superpixels* que alguno de los del modelo, pasará a sustituirlo. En caso de igualdad de número de *superpixels*, se valora el área de los *superpixels*. Los *clusters* del modelo que contienen alguno de los *superpixels* del primer fotograma que han sido etiquetados por el usuario, no pueden ser sustituidos sea cual sea su tamaño, de esta forma se protege la información aportada por el usuario, manteniendo así el vinculo que une el entorno real con el modelo.
El vector de probabilidades $\mathbf{P}$ de un nuevo *cluster* es inicializado de manera que el *cluster* tiene la misma probabilidad de pertenecer a todas los objetos/regiones de interés.

Capítulo 4

Experimentos

Este capítulo resume la validación experimental realizada de los métodos propuestos para interpretar automáticamente el contenido de secuencias de interiores.

Se ha realizado una evaluación de cómo influye en los resultados el proceso de actualización del modelo (seccion 4.2), variaciones en el método utilizado para relacionar imágenes nuevas con el modelo (sección 3.2.1), y del tamaño de los *superpixels* extraídos (seccion 2.1).

Por último en la sección 4.5 se incluyen resultados cualitativos de la configuración final del proceso en las dos secuencias anteriores y 3 adicionales.

4.1 Métodos de evaluación

Para evaluar la calidad y precisión del reconocimiento y etiquetado en las secuencias de las zonas de interés, se ha definido la medida *"precisión"* (ecuación 4.1) que evalúa la distancia de los resultados automáticos con los resultados obtenidos con un etiquetado manual de la secuencia. Se han etiquetado a mano aproximadamente 1 de cada 5 fotogramas de las secuencias.

$$precisión = \frac{\text{número de píxeles etiquetados correctamente como A}}{\text{número de píxeles que pertenecen a la región de interés A}} \quad (4.1)$$

4.2 Evaluación del proceso de actualización

Esta sección analiza los resultados obtenidos actualizando el modelo del entorno como se ha explicado en la sección 3.3 frente a no actualizar el modelo para nada y trabajar todo el rato con el modelo inicializado con los datos del primer fotograma. Esta comparación se ha realizado con las dos secuencias para las cuales tenemos información de referencia.

Secuencia 1

La *precisión* observada en la figura 4.1, en la clasificación de las dos primeras etiquetas es similar en ambos casos, mejorando la detección de puertas en el caso del modelo actualizado y siendo mayores los porcentajes de acierto para la

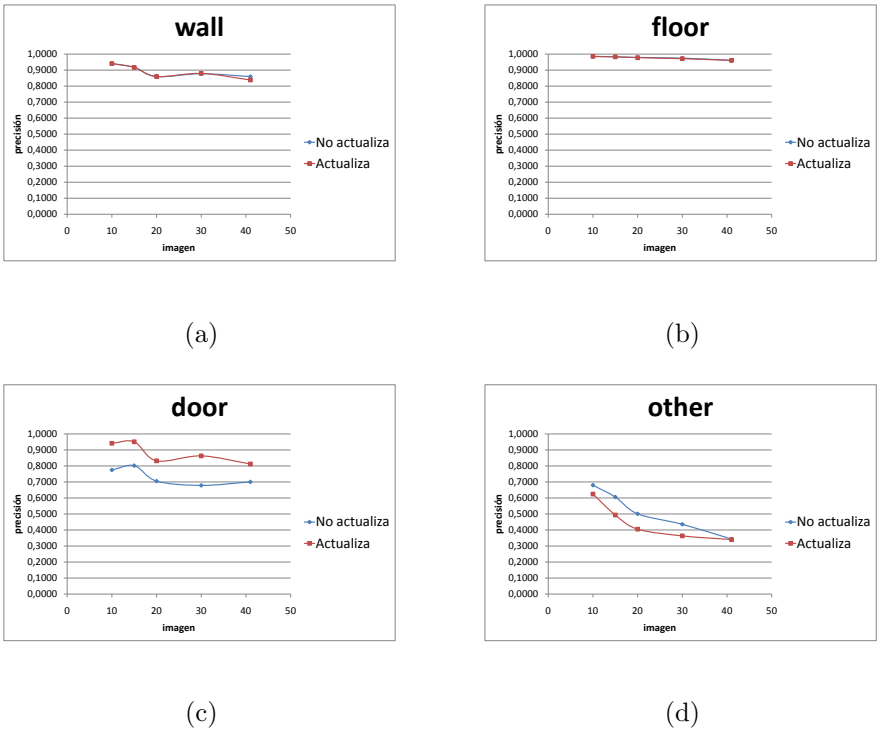


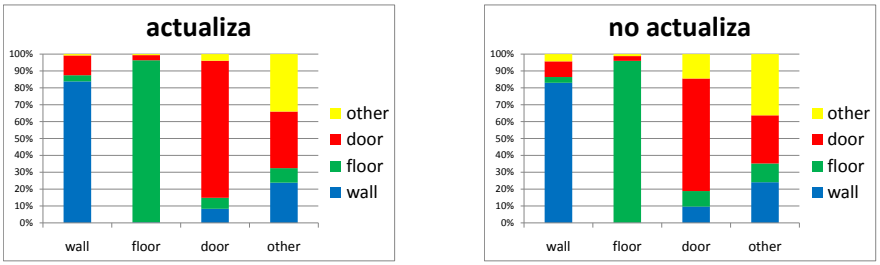
Figura 4.1: **Secuencia 1:** Comparación de la *precisión* obtenido actualizando (curva roja) y sin actualizar (curva azul) el modelo para cada zona de interés: (a) pared, (b) suelo, (c) puertas, (d) otros.

etiqueta 'other' en el caso del modelo sin actualizar. La Figura 4.2 nos sirve para analizar más en detalle que tipo de errores y confusiones esta teniendo el proceso. Como ya se ha visto en figura 4.1 la *precisión* es mayor cuando el modelo es actualizado a lo largo del proceso. Cada columna debería ser completamente del color de la etiqueta correspondiente, si la clasificación fuera perfecta. Las zonas de otros colores, implican que hemos confundido una zona marcada con una etiqueta con alguna de las otras. En estas gráficas podemos hacernos idea de con qué suele "confundir" cada una de las etiquetas, lo cual nos puede dar ideas de por donde mejorar el método.

Secuencia 2

La Figura 4.3 muestra la comparación en la *precisión* actualizando o no el modelo del entorno en la secuencia 2. Podemos observar en 4.3 (a) una precisión ligeramente mayor en la detección de pared. En el caso de no actualización del modelo. Sin embargo, el modelo actualizado da mejores resultados en los otros tres tipos de etiquetas, con grandes diferencias en suelo y otros.

De nuevo intentamos visualizar que esta ocurriendo en los casos de fallo mediante la Figura 4.4. Lo mas destacable es el hecho de que cuando no se actualiza, parece que la etiqueta "pared" se vuelve comodín, y la mayoría de las



(a) (b)

Figura 4.2: Análisis de los fallos en la clasificación de la **secuencia 1** actualizando (a) y sin actualizar (b) el modelo. Cada columna representa el 100% de los píxeles de una de las zonas, y en la barra se ve a que porcentaje de los mismos se le asigna cada una de las posibles etiquetas: (azul) pared, (verde) suelo, (rojo) puertas, (amarillo) otros

confusiones de las otras zonas son con ella. Es decir, que el modelo de la pared es el que mas abarca, por lo tanto es como el valor que pone en caso de duda. Esto explica porque solo la etiqueta pared funciona mejor en este caso de no actualizar, confirmando mas adecuado el proceso que incluye actualización del modelo.

4.3 Evaluación las distancias *superpixel*-modelo

Haciendo unas breves pruebas iniciales de los métodos descritos en la sección 3.2.1 para relacionar nuevas imágenes con el modelo, el que parecía comportarse de manera mas robusta era el método 3 (asignación con V próximos), por tanto, analizaremos este método en mas detalle. Como se explica en la sección 3.2.1, el método estima las probabilidades de cada *superpixel* i de pertenecer a cada una de las k posibles etiquetas E_k, teniendo en cuanta los V *clusters* con menor distancia Euclídea con el *superpixel* evaluado. Se ha realizado una comparación de resultados para determinar el valor de V para el que el método se comporta mejor.

En las gráficas de la figura 4.5 se observa que los resultados de la *precisión* son más bajos para los valores de V elevados. Cuando el valor de V es igual a 7, los resultados de las etiquetas 'suelo', 'puertas' y 'otros' es notablemente peor que para el resto. Esto nos hace ver que, la comparación de cada *superpixel* con un número demasiado elevado de *clusters* del modelo de referencia introduce más ruido a la hora de asignar las probabilidades para las distintas opciones de etiquetas. También se observa que para valores muy bajos de V las curvas de precisión bajan. En el caso de V=1, el error en las etiquetas 'suelo' y 'otros' es mayor. Con V=2, los resultados en la etiqueta 'pared' empeoran. No se aprecian grandes diferencias en los resultados para valores intermedios de V. Por lo tanto, teniendo en cuenta que cuanto menor es la V, menor es el tiempo de procesado, estableceremos V=*3* como el mejor compromiso eficiciencia-precisión en nuestro proceso.

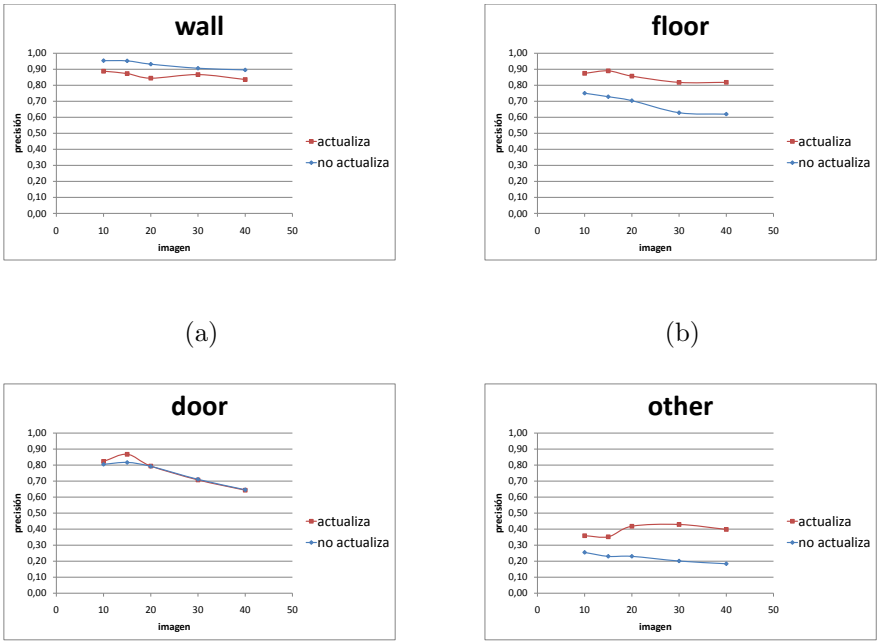


Figura 4.3: **Secuencia 2:** Comparación de la *precisión* obtenido actualizando (curva roja) y sin actualizar (curva azul) el modelo para cada zona de interés: (a) pared, (b) suelo, (c) puertas, (d) otros.

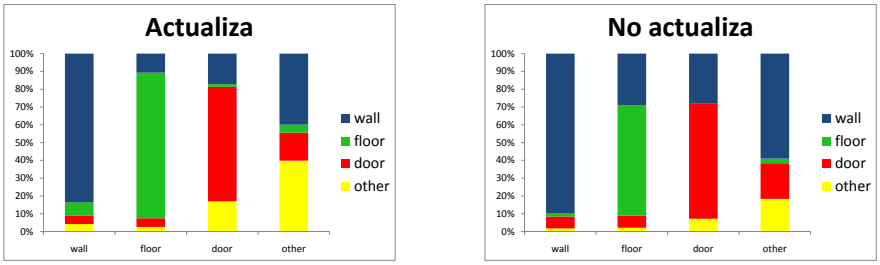


Figura 4.4: Análisis de los fallos en la clasificación de la **secuencia 2** actualizando (a) y sin actualizar (b) el modelo. Cada columna representa el 100% de los píxeles de una de las zonas, y en la barra se ve a que porcentaje de los mismos se le asigna cada una de las posibles etiquetas: (azul) pared, (verde) suelo, (rojo) puertas, (amarillo) otros

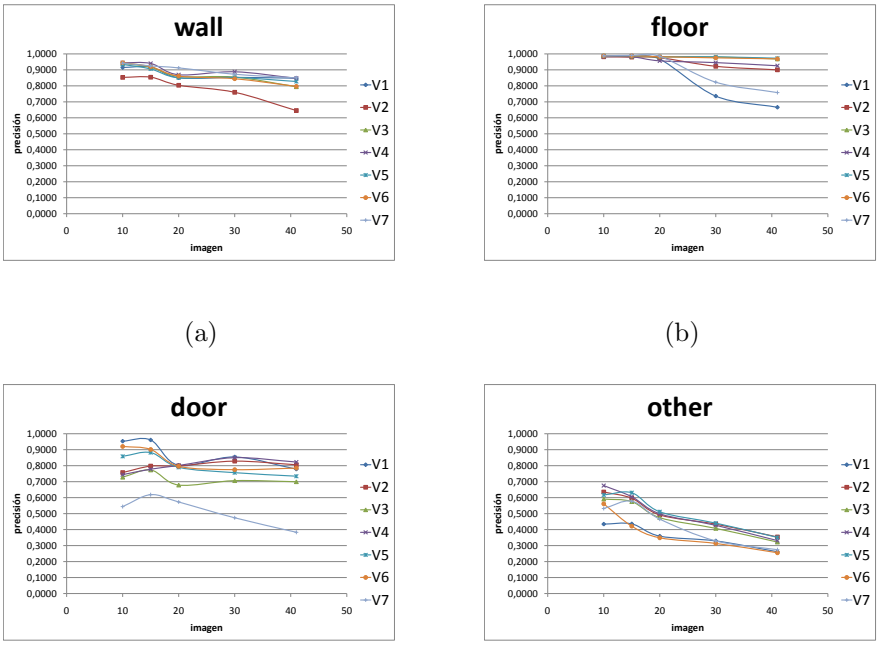
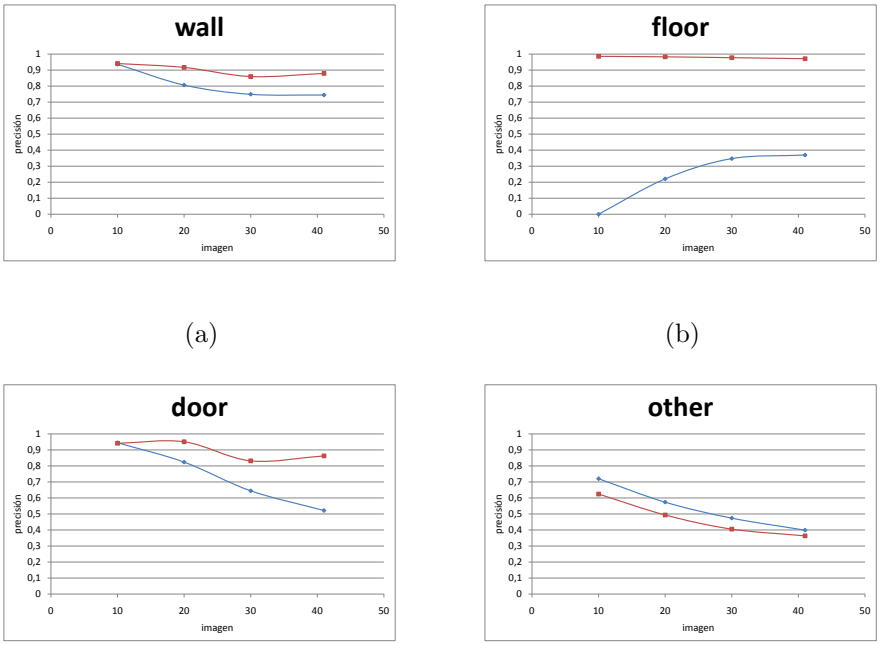


Figura 4.5: **Secuencia 1.** Comparación de la *precisión* según los "top-V" vecinos (V1 .. V7) utilizados para estimar las probabilidades para un *superpixel* de pertenecer a las distintas clases/etiquetas (pared, suelo, puerta, otros)



(a) (b)

(c) (d)

Figura 4.6: **Secuencia 1.** Comparación de la *precisión* según el tamaño de los *superpixels*. Las curvas azules representan los resultados utilizando unos *superpixels* de gran tamaño (configuración 1). Las curvas rojas representan los resultados utilizando *superpixels* de tamaño medio (configuración 2). (a) pared, (b) suelo, (c) puerta, (d) otros.

4.4 Comparación de resultados según el tamaño de los *superpixels*

El tamaño de los *superpixels* puede variar en función de unos parámetros específicos de las librerías utilizadas. El número y tamaño de los *superpixels* segmentados en la imagen son una decisión importante en el proceso. Cuanto menor sea el número de segmentos a procesar, menor es el tiempo consumido, pero si el número de *superpixels* es demasiado bajo, puede que se mezclen zonas de distintos objetos de interés en un mismo *superpixel*, provocando errores obligatoriamente en la clasificación.

En el anexo A se muestra las distintas segmentaciones que se obtienen en función de los parámetros del método de segmentación utilizado [13]: *sigma*, *k* y *min*. Se ha procesado la secuencia de imágenes con dos segmentaciones distintas: la configuración 1 utiliza *sigma=0.4*, *k=300* y *min=100*; la configuración 2 utiliza *sigma=0.3*, *k=200* y *min=100*. La figura 4.6 muestra la precisión en el etiquetado utilizando estas dos configuraciones.

En la Figura 4.7 se muestran los resultados obtenidos con *superpixels* de las dos configuraciones, y se puede apreciar claramente que si los *superpixels* son

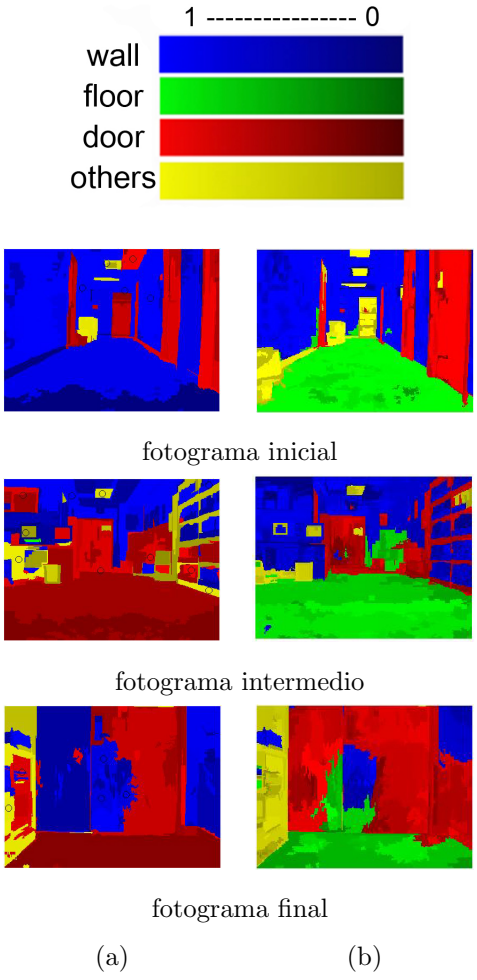
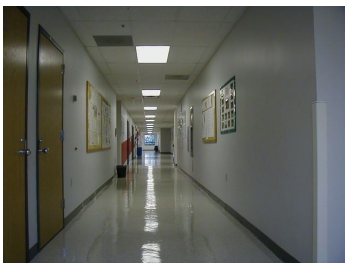
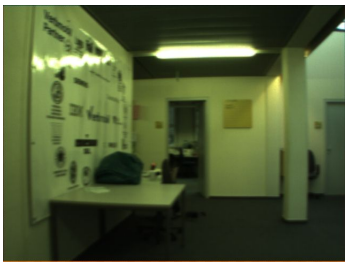


Figura 4.7: (a) clasificación de los *superpixels* de las imágenes inicial, intermedia y final segmentadas con la configuración 1. (b) clasificación de los *superpixels* de las imágenes inicial, intermedia y final segmentadas con la configuración 2.

secuencia 1



secuencia 2



secuencia 3

secuencia 4

secuencia 5

Figura 4.8: Imágenes iniciales de cada una de las secuencias procesadas.

demasiado grandes los resultados son peores, sobretodo a la hora de detectar los *superpixels* que pertenecen al suelo.

4.5 Resultados de reconocimiento obtenidos con la configuración final propuesta

Teniendo en cuenta los resultados anteriores y las configuraciones de los parámetros con las que han sido obtenidas, se han procesado varias secuencias de imágenes capturadas en entornos de interior de edificios. Las zonas a detectar han sido, en todos los casos las que ya hemos ido utilizando: pared, suelo, puerta, otros.

En la Figura 4.5 se muestran los primeros fotogramas de cada una de las cinco secuencias que se han utilizado en los experimentos.

A continuación se muestran los resultados de interpretación de cada una de las secuencias. Para las secuencias 1 y 2 se muestran no solo resultados cualitativos del reconocimiento de las distintas áreas, sino también evaluaciones

cuantitativas de la precisión ya que tenemos datos de referencia obtenidos a mano solo para esas dos secuencias.

Las secuencias 1 y 2 se muestra, además de las imágenes etiquetadas, los resultados obtenidos tras la comparación con el 'ground truth'.

Secuencia 1 En la Figura 4.9 se muestran los resultados finales obtenidos para la secuencia 1. En el caso de la pared (Figura 4.9 (a)), se ha detectado muy bien en todos los casos salvo en la última imagen (41), donde se confunde casi el 100% de la pared con el tipo puerta. En el caso del suelo (Figura 4.9 (b)) se observa un alto índice de precisión en la detección de suelos, ya que casi no se ven zonas que no sean verdes (suelo). En el caso de las puertas (Figura 4.9 (c)), también predomina el color correcto (rojo), aunque no tanto como en el caso anterior. Si atendemos el último grupo, Figura 4.9 (d), confirmamos el punto más débil del proceso, las zonas "otros", donde vemos muchas zonas donde el algoritmo se confunde. No resulta sorprendente que esta zona sea la mas complicada, ya que representa un conjunto de objetos inciertos, porque intenta englobar todo lo que no sea conocido.

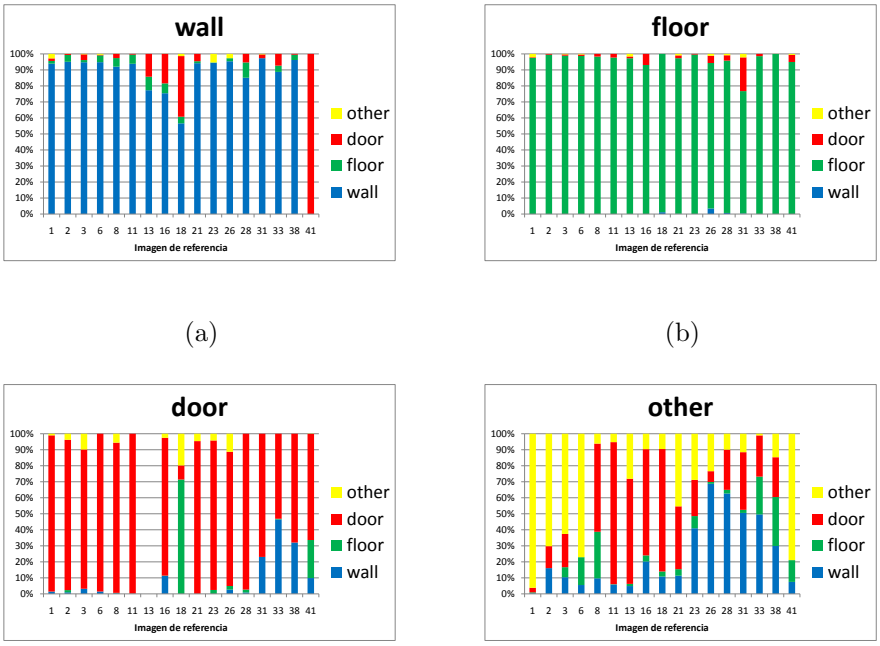
En el vídeo del CD adjunto (*./videos/secuencia 1.m4v*) se incluyen los resultados de procesar todos los fotogramas de esta secuencia de manera similar a la figura 4.10.

Secuencia 2 En la Figura 4.11 se muestra un resumen similar. El procesado de la secuencia completa puede verse en el vídeo: *./videos/secuencia 4.m4v*, del cual se muestran varios ejemplos de fotogramas en la figura 4.12.

Secuencia 3 La Figura 4.13 muestra varios ejemplos de fotogramas de la secuencia 3 una vez han sido procesados y etiquetados, el procesado de la secuencia completa puede verse en el vídeo: *./videos/secuencia 3.m4v*.

Secuencia 4 La Figura 4.14 muestra varios fotogramas de la secuencia 4 una vez han sido procesados y etiquetados, el procesado de la secuencia completa puede verse en el vídeo: *./videos/secuencia 4.m4v*.

Secuencia 5 La Figura 4.15 muestra varios fotogramas de la secuencia 5 una vez han sido procesados y etiquetados, el procesado de la secuencia completa puede verse en el vídeo: *./videos/secuencia 5.m4v*.



(a) (b)

(c) (d)

Figura 4.9: **Secuencia 1.** Resultados de la clasificación en cada frame para cada una de las regiones de interes: pared (azul), verde (suelo), rojo (puertas) y amarillo (otros).

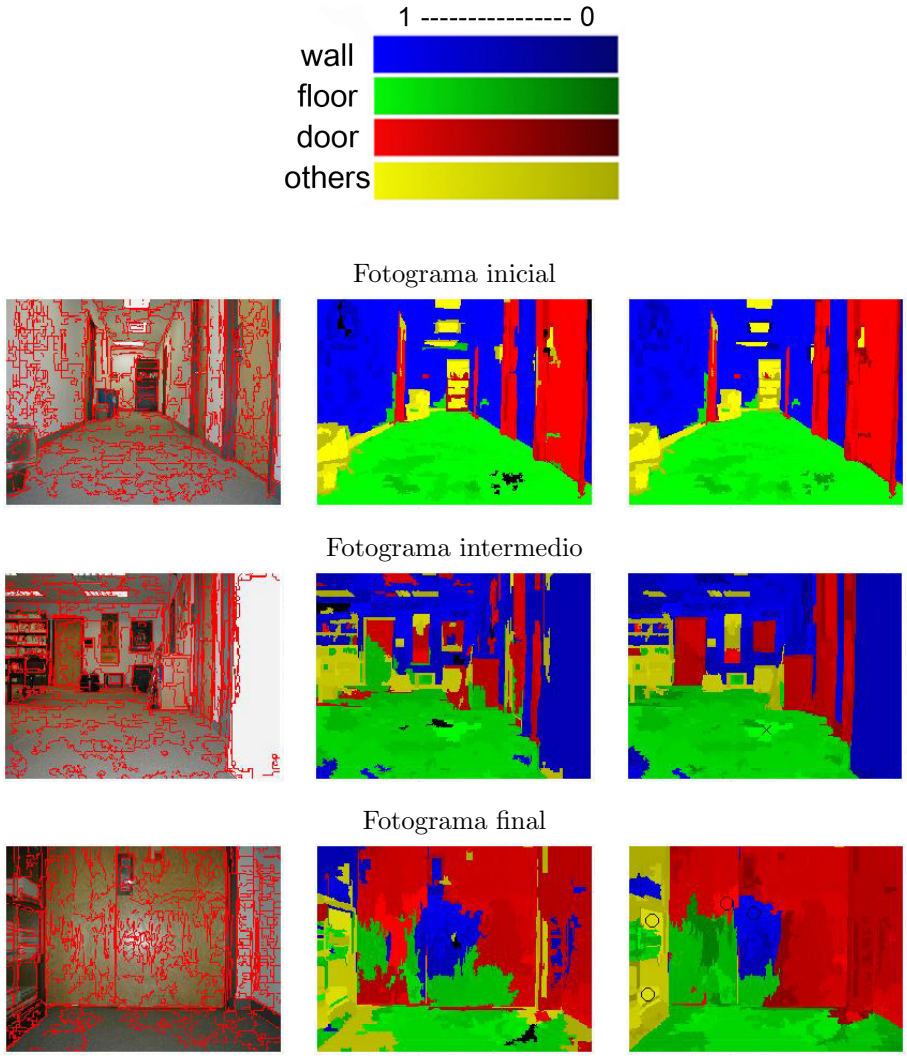
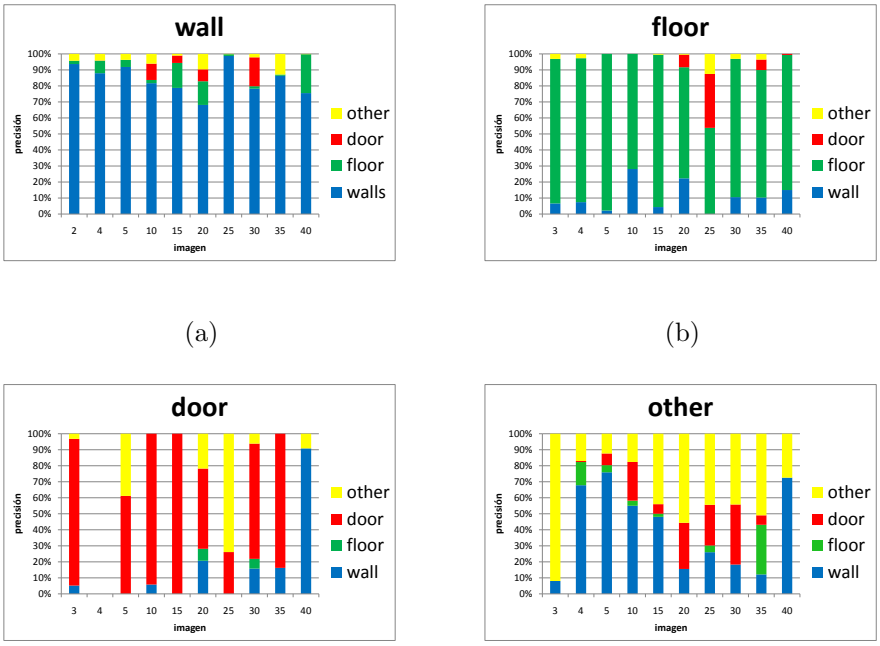


Figura 4.10: Ejemplos de fotogramas etiquetados a lo largo de la **secuencia 1**. (a) es la imagen original segmentada (b) es el etiquetado de la imagen sin usar el MRF (c) es el etiquetado final.



(a) (b)

(c) (d)

Figura 4.11: **Secuencia 2.** Resultados de la clasificación en cada frame para cada una de las regiones de interes: pared (azul), verde (suelo), rojo (puertas) y amarillo (otros).

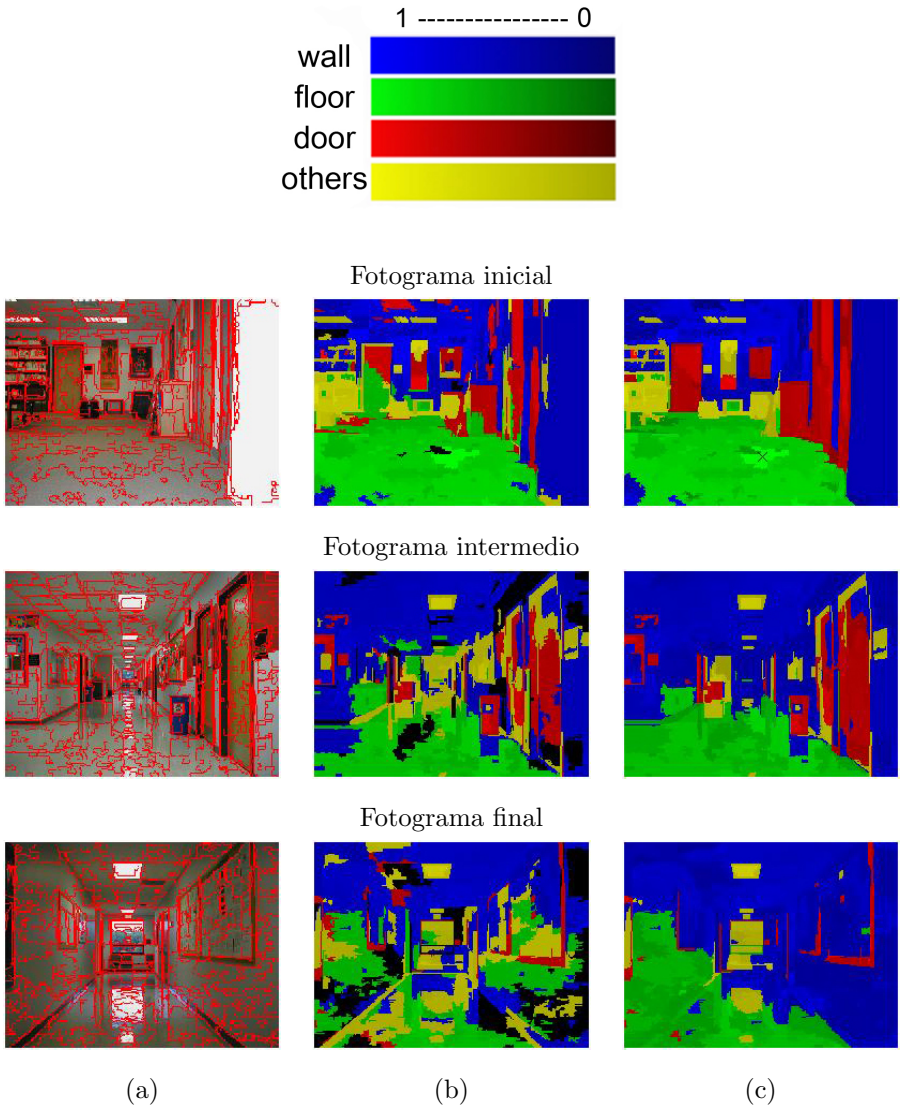


Figura 4.12: Ejemplos de fotogramas etiquetados a lo largo de la **secuencia 2**. (a) es la imagen original segmentada (b) es el etiquetado de la imagen sin usar el MRF (c) es el etiquetado final.

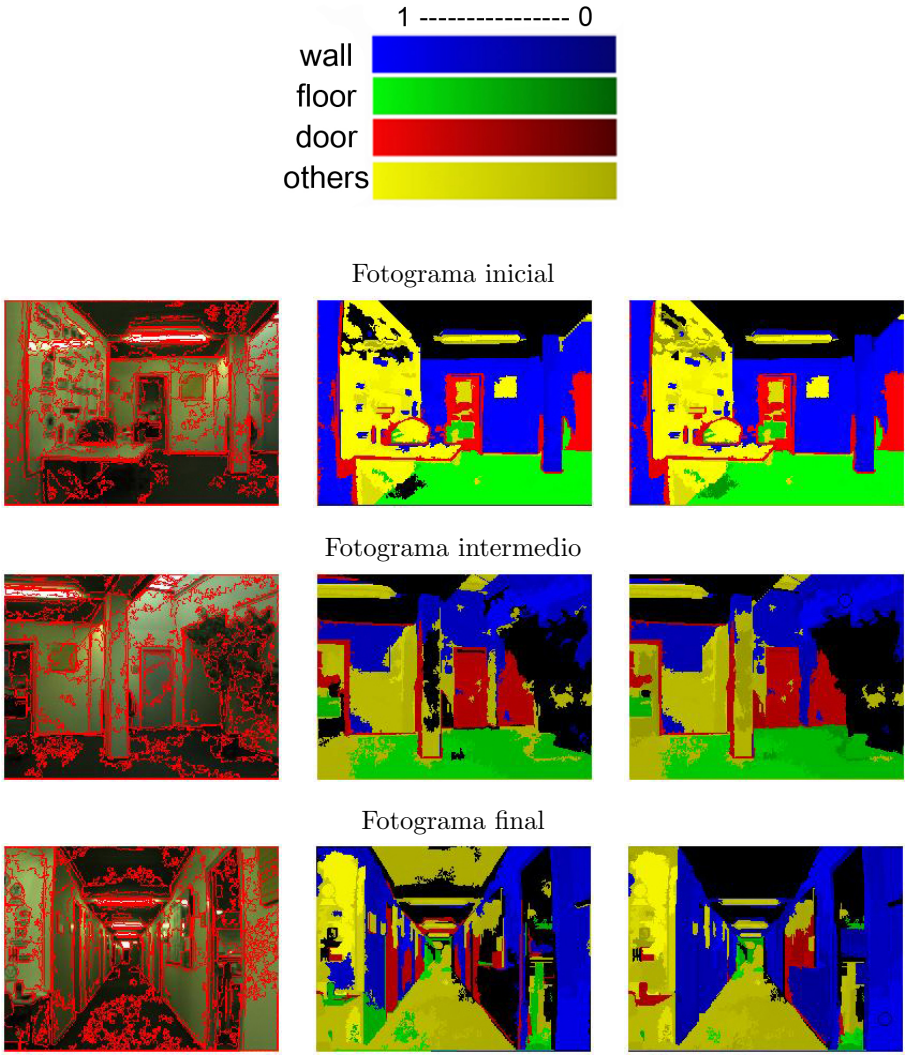


Figura 4.13: Ejemplos de fotogramas etiquetados a lo largo de la **secuencia 3**. (a) es la imagen original segmentada (b) es el etiquetado de la imagen sin usar el MRF (c) es el etiquetado final.

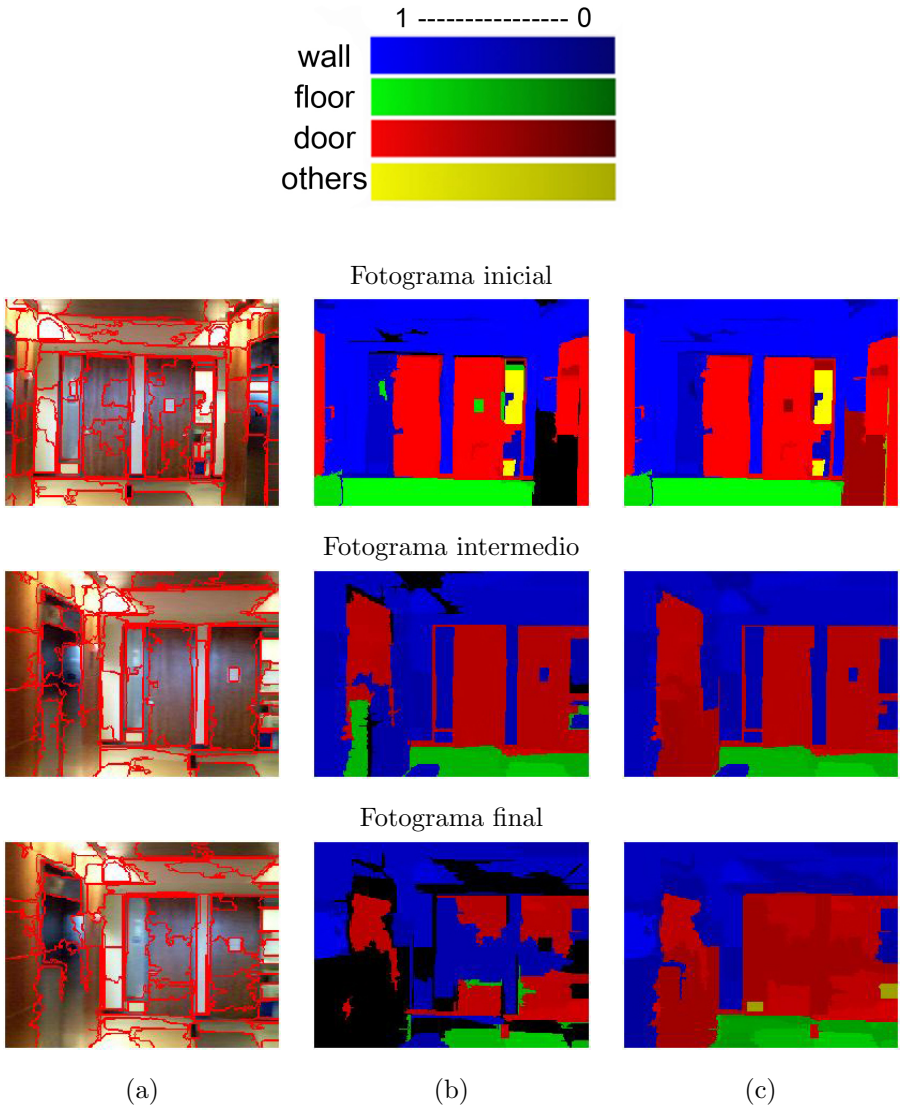


Figura 4.14: Ejemplos de fotogramas etiquetados a lo largo de la **secuencia 4**. (a) es la imagen original segmentada (b) es el etiquetado de la imagen sin usar el MRF (c) es el etiquetado final.

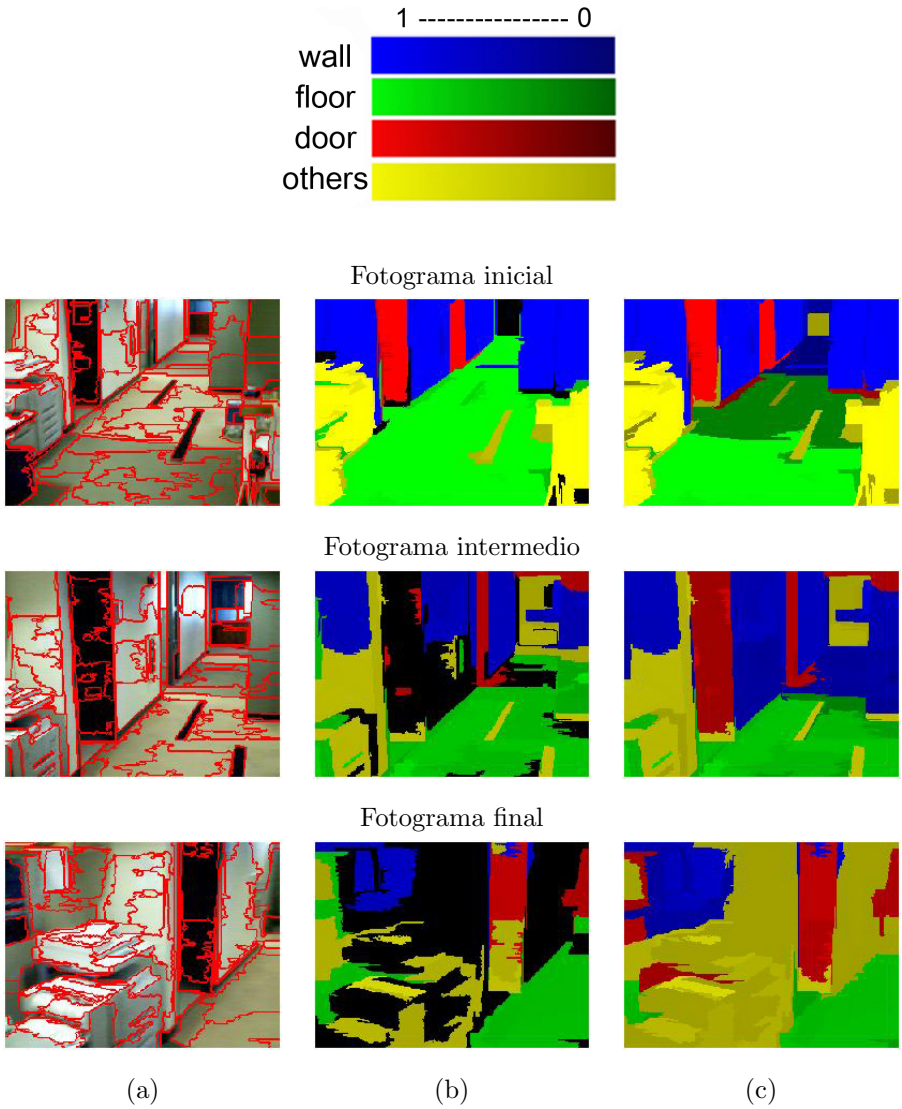


Figura 4.15: Ejemplos de fotogramas etiquetados a lo largo de la **secuencia 5**. (a) es la imagen original segmentada (b) es el etiquetado de la imagen sin usar el MRF (c) es el etiquetado final.

Capítulo 5

Conclusiones

5.1 Conclusiones personales

La realización de este proyecto se ha llevado a cabo durante siete meses. El hecho de que se trate de un proyecto de investigación ha hecho que me resulte muy interesante, pero esto le añade un punto de dificultad en algunos momentos. Considero el proyecto muy positivo para mi formación tanto en el campo de la investigación en general como el de la visión por computador en particular.

Durante la realización del proyecto ha habido partes que hemos desarrollado e implementado, por ejemplo los descriptores y el modelo del entorno, las cuales han requerido una gran documentación previa y tener claro el objetivo que se buscaba, además de muchos experimentos para comprobar su funcionalidad. El hecho de crear el código que utilizamos da un conocimiento más profundo del método y facilita su uso y modificación.

Otra parte ha consistido en el uso de herramientas existentes, como los *superpixels*, lo cual implica la búsqueda de las herramientas, el estudio de las mismas y, finalmente, su aplicación, resulta muy interesante ver el trabajo en otras universidades sobre temas parecidos, incluso gente que está tratando el mismo tema, lo cual hace que se tenga una visión más global del problema.

5.2 Conclusiones trabajo

Hemos conseguido desarrollar satisfactoriamente todos los pasos necesarios del proceso para interpretar y reconocer zonas de interés en una secuencia. Se partía de un trabajo previo [18], respecto del cual diseñando nuevos métodos para cada uno de los pasos se ha conseguido mejorar bastante la precisión de las áreas reconocidas.

En el paso de representación de la imagen hemos estudiado nuevos descriptores para los segmentos de la imagen. En la parte de modelado hemos diseñado un modelo diferente para capturar las características típicas de las zonas a reconocer y también hemos diseñado nuevos métodos para ir actualizando el método a lo largo de la secuencia.

Las nuevas propuestas han sido validadas con experimentos rigurosos y exhaustivos en múltiples ejemplos.

5.3 Trabajo Futuro

El objetivo una vez procesada una imagen para identificar dónde están las zonas dominantes de interés sería añadir pasos de reconocimiento de objetos mas particulares, que buscaríamos en las zonas identificadas como "otros".

También habría que realizar una implementación más optimizada de todos los algoritmos para poder evaluar este método en aplicaciones online, que realicen el procesado conforme se van adquiriendo las imágenes.

Otra posible mejora sería la combinación de herramientas de detección de formas geométricas en imágenes con el actual método de reconocimiento de áreas de interés.

Por último, este trabajo se va a resumir y enviar en formato de articulo de investigación a una conferencia internacional.

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 objet 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 ob ject 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 ob ject 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.