UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

SEGMENTACIÓN ESPACIO-TEMPORAL DEL CONTEXTO DE UN VÍDEO

Sergio Serra Sánchez.
Tutor: Marcos Escudero Viñolo.
Ponente: Jesús Bescós Cano.

Julio 2017

SEGMENTACIÓN ESPACIO-TEMPORAL DEL CONTEXTO DE UN VÍDEO.

Sergio Serra Sánchez
Tutor: Marcos Escudero Viñolo
Ponente: Jesús Bescós Cano

Video Processing and Understanding Lab
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio 2017

Trabajo parcialmente financiado por el Ministerio de Economía y Competitividad del Gobierno de España bajo el proyecto TEC2014-53176-R (HAVideo) (2015-2017)

Resumen

El objetivo de este trabajo fin de grado es obtener la segmentación de los objetos contextuales que componen un vídeo.

La implementación general consistirá en transferir máscaras de objetos contextuales desde una base de datos a una imagen no observada. Para ello se tomará una gran importancia en la construcción de los data-sets que se emplearán en cada una de las diferentes escenas a analizar.

Se tratará de estudiar toda la información que nos aporta una imagen de entrada mediante dos modelos distinguidos con el objetivo de extraer la probabilidad de cada píxel de pertenecer a una de las dos etiquetas que se establecen en este trabajo: contexto o no contexto.

Con un modelo de energías se obtendrá la segmentación resultado, que determinará que pixeles de la imagen se corresponden con un objeto contextual. Finalmente la segmentación contextual para un video se extraerá promediando las máscaras obtenidas para los distintos frames que lo componen.

Palabras clave

segmentación, información contextual, transferencia de máscaras, etiquetas, energías unitarias

Abstract

The aim of this end-of-degree project is to obtain the segmentation of the contextual objects that compose a video.

The general implementation will consist of transferring masks of contextual objects from a database to a not observed image. For this purpose, importance will be attached to the construction of data-sets that will be used in each of the different scenes to be analyzed.

The dissertation will consist of studying all the information given by an image of entry through two distinguished models with the aim to extract every pixel's probability of belonging to one of the two labels that are established in this work: "context" or "not context".

Through a model of energies the final segmentation will be obtained, determining which of the pixels on the image correspond with a contextual object. Finally, the contextual segmentation for a video will be acquired by averaging the masks obtained for the different frames that compose it.

Keywords

segmentation, contextual information, transferring windows masks, labels, unary potentials

Agradecimientos

Agradecer totalmente la realización de este proyecto con mi tutor Marcos. Siempre presente cuando se le ha necesitado y, sobre todo, demostrando una gran confíanza sobre mi no solo en el propio trabajo. Por supuesto tengo que integrar aquí a todos los miembros del VPUlab, el buen rollo presente y siempre la mano tendida que han ofrecido para cualquier cosa ha sido realmente una gran ayuda.

Primero, mi familia, mi mamá, mi papá y mi hermana, que haría sin vosotros, el mayor apoyo que uno puede tener y que se sabe que siempre estará ahí, os quiero. Y, por supuesto mi abuelilla y mi tía, dos grandes acompañantes desde que soy un pequeñajo.

Que decir de mis compañeros de viaje en estos años. Emilio, en los momentos grandes y menos grandes el mayor apoyo durante estos años en la universidad que se ha convertido en un amigo para toda la vida, así como el Miguel un amigo siempre con el que soltar una sonrisa. Los dos ahora continuamos el viaje hacia lo más alto. Y Pablito ese tío siempre positivo que sabe como animarte y motivarte.

Esas dos chquillas Claudia y Ana, de las cuales sin duda he aprendido mucho, de la primera sobre todo.

Mencionar a Sergi, Jose, User, Juli, Paula, Julia siempre por ahi animando también el cotarro.

Tengo que meter aquí a una chiquilla que se ha colado hace poco cerca mío, Paula, creo que tenemos algo especial que nos une mucho.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En este primer capítulo de introducción se abordan las motivaciones que han influido al interés de investigación en esta técnica y se expone el objetivo principal y los sub-objetivos que se buscan junto al esquema general de organización que se lleva a cabo a lo largo de la memoria.

1.1. Motivación

La segmentación es un concepto clave en el tratamiento digital de la imagen. En la actualidad, las aplicaciones que podemos atribuir a la segmentación de imágenes o videos se reinventan y evolucionan de forma cada vez más rápida.

En este trabajo se ha implementado una herramienta que segmentará la información contextual contenida en una imagen o vídeo. Para ello definiremos como objetos contextuales aquellos que conforman la parte que no constituye el foco de análisis principal.

Buscamos un sistema que funcione en su totalidad de forma automática. Con ello, se consigue mejorar sistemas ya existentes de segmentación que requieren una primera interacción entre usuario y máquina para establecer, desde un primer momento evidencias de lo que se considera objeto contextual dentro de la imagen. Este intercambio de información facilita en gran medida los proceso de segmentación y se trata de un gran reto conseguir suprimirlo.

Existen importantes aplicaciones en las que podemos emplear la segmentación de información contextual. Por ejemplo, empleando la infromación contextual se puede restringir el análisis principal de una secuencia de vídeo a zonas específicas de la escena.

1.2. Objetivos

El objetivo principal buscado en este trabajo es la segmentación espacio-temporal de los objetos contextuales de un vídeo mediante transferencia de máscaras.
Para abordar el objetivo principal se buscará consumar los siguientes sub-objetivos:

1. Realizar un profundo análisis del estado del arte actual que envuelve a esta tecnología para englobar la mayor información posible y poder aplicarla a nuestro proyecto.
2. La construcción de consistentes data-sets específicos para cada escenario que disponemos a analizar. Valoraremos en cada marco qué consideraremos como información contextual para elaborar un conjunto de máscaras que nos definan los tipos de objeto buscados en cada caso.
3. Una correcta implementación del algoritmo de transferencias de máscaras siguiendo las etapas que se definen en [1].
4. Obtener un sistema que funcione de manera totalmente automática sin ser necesaria la interacción con el usuario.

Completando estos objetivos parciales de forma secuencial se podrá ejecutar una evaluación coherente que verifique la eficacia del sistema implementado determinado la usabilidad que le podemos otorgar.

1.3. Organización de la memoria

La memoria consta de los siguientes capítulos:

- Capítulo 1: Introducción, motivación y objetivos del proyecto (ver 1).
- Capítulo 2: Estudio del estado del arte (ver 2).
- Capítulo 3: Diseño y desarrollo del sistema de transferencia de máscaras. (ver 3).
- Capítulo 4: Experimentos realizados y resultados (ver 4)
- Capítulo 5: Conclusiones y trabajo de futuro (ver 5)
- Bibliografía.

Capítulo 2

Estado del arte

2.1. Introducción

Para la implementación de este trabajo se han estudiado y empleado diferentes herramientas de trasferencia de máscaras ya existentes así como diferentes instrumentos que se apoyan directamente en esta técnica. A su vez, se han aplicado ideas extraídas directamente de distintos trabajos de segmentación para el desarrollo del proyecto presentado. En este capítulo se lleva a cabo una descripción de las distintas herramientas bases empleadas así como de los algoritmos en los que se basa nuestro sistema para llegar a nuestro objetivo. En primer lugar, se describen las características que poseen los objetos contextuales en cada escenario de análisis que se dispone. Posteriormente se detallan las herramientas básicas empleadas. Incluye la generación de propuestas que se maneja (ver sección 2.3.1) junto con la comparación por descriptores GIST (ver sección 2.3.2). A continuación, se especifica la técnica de transferencia de máscaras (ver sección 2.4) para, posteriormente, describir el modelo de energías empleado con el que se obtiene la segmentación final (ver sección 3.5). Por último, se realiza una presentación de los data-sets empleados, así como las características que ofrecen (ver sección 2.5).

2.2. Información contextual

La información contextual de una imagen o un vídeo la componen los distintos objetos sobre los cuales no recae el foco de análisis principal. De esta forma, en este proyecto se tratará extraer dicha información de distintos escenarios que se nos presenten. Algunos ejemplos de objetos contextuales se describen en la figura 2.1 para dos escenarios distintos.

Figura 2.1: Ejemplo de objetos contextuales que componen una imagen
Columna izq: en un escenario de oficina, las pantallas de ordenador, los ratones o los teclados componen la infromación contextual. Columna dcha: en un marco de análisis exterior, la carretera, los edificios o el cielo conforman los objetos contextuales.

2.3. Herramientas básicas

En este trabajo se utilizan dos herramientas de caracter básico de gran interés: el generador de propuestas EdgesBoxes (ver sección 2.3.1) y el comparador por descriptores GIST (ver sección 2.3.2).

2.3.1. Generación de propuestas: EdgesBoxes

Un elemento esencial en este trabajo es analizar y operar una imagen a nivel parcial, no todos los píxeles en su totalidad. Son lo que denominaremos como *ventanas* o *bounding boxes* (*bbs*) representados como rectángulos (ver figura 2.2). Por ello, la selección de las propuestas que defininen las ventanas a analizar es un reto realmente importante en nuestra implementación. Para llevar acaba esta detección de los *bbs* más útiles se emplea 'EdgesBoxes Boundary & Feature Learning (Piotr Dollár)' [2].

Este generador de propuestas se basa en el uso de los bordes como característica de extracción de *bbs* sobre objetos. La observación formulada en este método es que el número de contornos contenidos en un *bb* son claramente indicativos de la probabilidad que posee cada uno de ellos de contener un objeto. Lo que se propone, por tanto, es que mediante la extracción de los contornos y los bordes de una imagen seamos capaces de dar puntuaciones a los *bbs* obtenidos asignándoles una mayor o menor significatividad. Por tanto, para hallar las propuestas con una puntuación más alta, este algoritmo se basa en comparaciones directas entre todos los posibles *bbs* encontrados. Midiendo los contornos presentes en un *bb* y comparándolos con los bordes solapados con otros *bbs* adyacentes se puede definir propuestas que se

consideran concluyentes asigándose con una mayor puntuación.

Usando una eficiente estructura de datos, todos los *bbs* candidatos pueden ser evaluados en muy poco tiempo devolviendo un ranking de las propuestas según su puntuación.

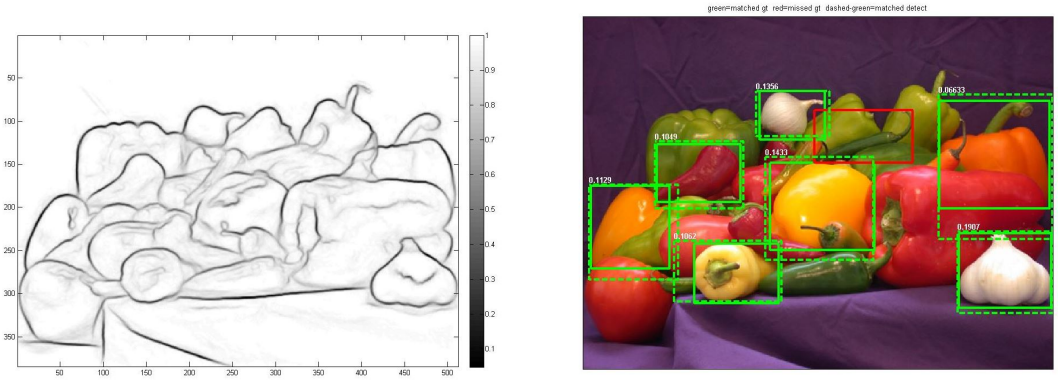

Figura 2.2: Funcionamiento EdgesBoxes Boundary & Feature Learning (Piotr Dollár). Primera columna: la herramienta calcula los contornos presentes en la imagen. Segunda columna: tras un análisis de la cantidad de bordes presentes en cada uno de los *bbs* encontrados, el sistema calcula los que poseen una mayor puntuación, descartando los que no llegan a un umbral mínimo.

Esta herramienta nos permite personalizar varios parámetros determinantes para nuestro sistema. En primer lugar, el número máximo de *bbs* que se desea que la herramienta detecte por imagen. En segundo lugar, la mínima puntuación que se define como límite entre todos los *bbs* detectados, es decir, todos los *bbs* que no cumplan una puntuación mínima son descartados.

Durante el desarrollo del trabajo se han evaluado dos sistemas alternativos para la generación de propuestas:

1. **MCG** (Multiscale Combinatorial Grouping): utiliza la técnica de análisis de segmentación por multiescalas extrayendo las características que siguen presentes en la imagen tras realizar distintas interpolaciones o diezmados [3].

2. **Objectness**: como sus herramientas análogas proporciona ventanas de análisis de una imagen con una alta probabilidad de contener cualquier clase de objeto de interés en su interior. Analiza la diferencia de contrastes presentes entre los distintos tipos de textura que se pueden encontrar a lo largo de toda la extensión de la imagen extrayendo los de mayor puntuación [4].

Finalmente, estas dos herramientas adicionales se han descartado debido a la falta

de pruebas sobre nuestro sistema empleándose solamente el generador de propuestas EdgesBoxes.

2.3.2. Comparación por descriptores: GIST

Mediante los descriptores GIST [5], dada una imagen de entrada:

1. Procesa la imagen con 32 filtros de Gabor (ver apéndice A) a 4 escalas y 8 orientaciones produciendo 32 mapas de características del mismo tamaño de la imagen de entrada.

2. Divide cada mapa de características en 16 regiones (ver figura 2.3) calculando la media de características dentro de cada una de estas regiones.

3. Concatena los 16 valores provinientes de las medias de los 32 mapas de características, devolviendo 16x32 = 512 descriptores GIST.

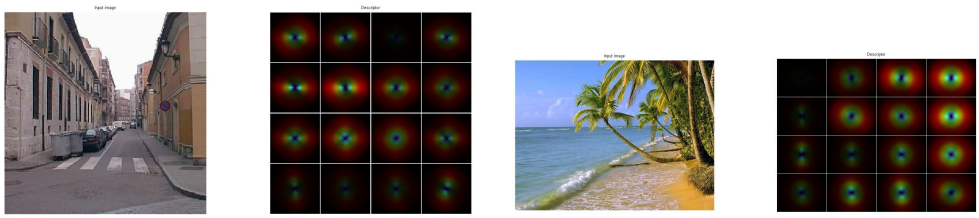

Figura 2.3: Ejemplos Descriptores GIST. La columna izquierda para cada figura muestra una imagen de entrada mientras que la columna derecha para cada figura representa los descriptores GIST que definen la imagen.

Por tanto, los decriptores GIST, resumen la información de gradiente (escalas y orientaciones) para las diferentes partes de toda una imagen. Comparando esta información entre distintas ventanas extraídas de una imagen de entrada se podrá determinar que *bbs* de entre todos los extraídos en una fase de entrenamiento anterior son más parecidos visualmente.

2.4. Transferencia de máscaras

En las técnicas de segmentación objeto-fondo se busca separar con una máscara binaria los objetos de interés de todo el fondo que lo rodea. Mediante la idea de transferencia de máscaras [1] ésto se consigue ajustando máscaras de un conjunto de entrenamiento a una imagen de test.

Sin embargo, en la idea propuesta por [1] la comparación imagen de entrada vs. entrenamiento no se realiza a nivel de la imagen en toda su extensión. En vez de

ello, se extraen automáticamente ventanas centradas sobre los objetos presentes en la imagen de entrada (ver sección 2.3.1). El procedimiento seguido en esta técnica se ve reflejado en la figura 2.4 dividiéndose en distintas etapas. Transifiriendo las máscaras de las ventanas de entrenamiento visualmente más parecidas a cada ventana de test, se aplica sobre ellas dos modelos de energías: modelo de localización y modelo de apariencia. Uniendo ambos se puede extraer una máscara de segmentación de la imagen en su totalidad mediante el cálculo de las energías finales. La idea clave de la transferencia de máscaras es, por tanto, que las ventanas de una imagen de entrada visualmente parecidas a ventanas del conjunto de entrenamiento pueden tener máscaras de segmentación similares.

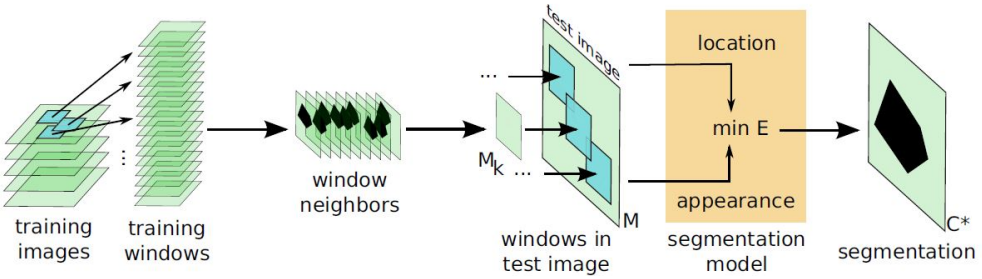

Figura 2.4: Esquema del proceso de Transferencia de Máscaras. Extraido de [1]

2.5. Data-sets analizados

La buena estructura y el buen desarrollo de unos data-sets consistentes dependiendo del tipo de datos contextuales a analizar cobran una vital importancia a lo largo de este trabajo. Dichos data-sets se han conformado mediante dos conjuntos de imágenes etiquetadas y anotadas en su totalidad proporcionadas por dos distribuidores diferentes:

2.5.1. LabelMe

LabelMe es una herramienta de anotación online empleada para la construcción de data-bases de imágenes etiquetadas y anotadas expuesta en [6]. Así mismo, ofrece una gran cantidad de *data-sets* de entrenamiento ya conformados con una gran cantidad de imágenes etiquetadas por tipos de objetos (ver figura 2.5) pudiendo obtener la máscara binaria de cada uno de ellos. Extrayendo de sus anotaciones los objetos que nos interesan pertenecientes a información de contexto podremos construir distintas bases de datos para cada escenario de análisis (e.g: oficina ó exterior):

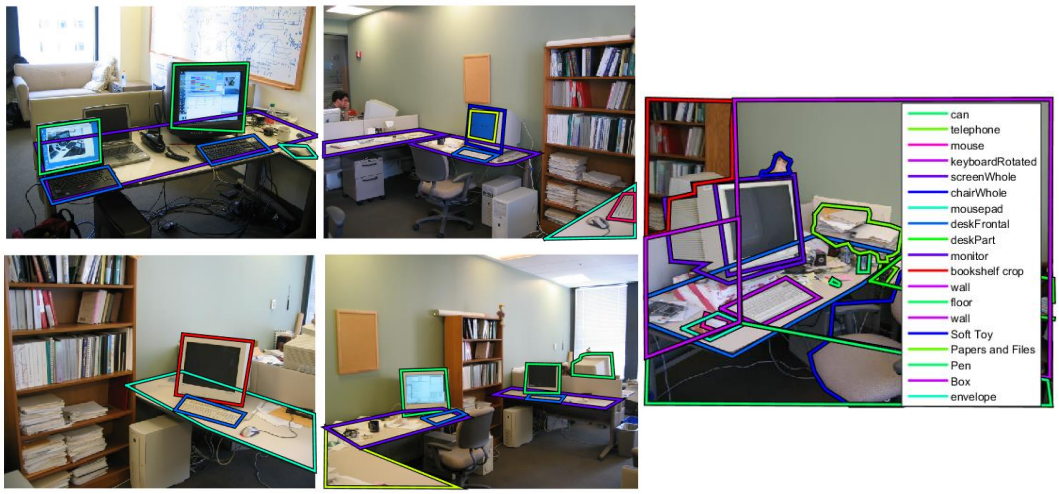

Figura 2.5: Etiquetas proprcionadas por LabelMe en un entorno de oficina
Izq: Ejemplo de imágenes etiquetadas proporcionadas por LabelMe de algunos objetos predefinidos: pantalla, teclado, ratón. Dcha: Ejemplo de todas las etiquetas que se encuentran en una imagen disponible en LabelMe.

2.5.2. ADE20K

ADE20K, expuesto en [7], proporciona un data-set de imágenes con un nivel alto de etiquetado (ver figura 2.6) presentado en estrucutura de arbol, en el que existen objetos anotados que a su vez contienen objetos más concretos anotados en su interior. Proporciona máscaras altamente definidas con un alto grado de especificidad para formar data-sets concretos para cada tipo de escenario buscado.

Con la unión de ambas base de datos se construirá un data-set personalizado para cada escenario de analisis conteniente de objetos contextuales concretos de interés.

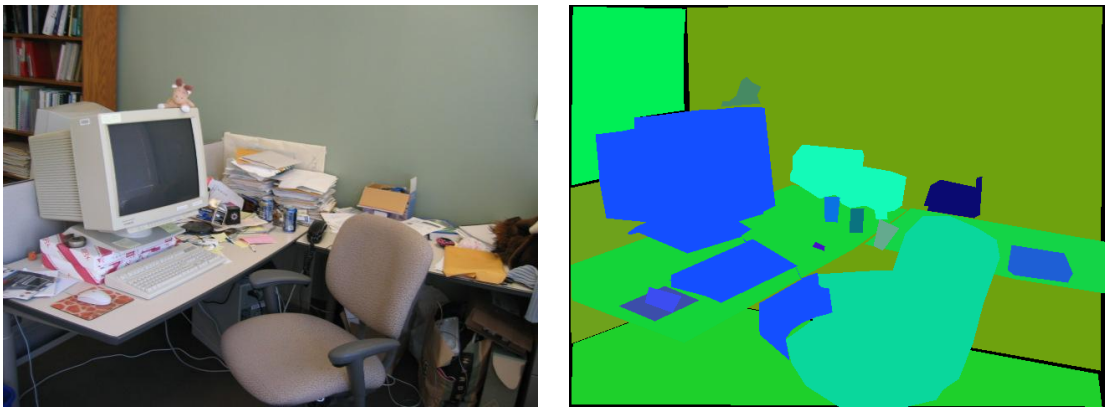

Figura 2.6: Etiquetado proporcionado por ADE20K
Columna izquierda: imagenes originales en el RGB. Columna derecha: las correspondientes imágenes etiquetadas proporcionadas por el data-set ADE20K

Capítulo 3

Diseño y desarrollo

3.1. Introducción

En este capítulo se describe como a partir de herramientas del estado del arte se ha desarrollado e implementado un sistema de segmentación de la información contextual presente en una imagen por transferencia de máscaras. Se parte de la idea principal propuesta por [1]. La idea base de este proyecto es que comparando imágenes mediante ventanas extraídas de las mismas se conseguirán resultados más correctos que realizando una comparación a nivel de las imágenes en su totalidad.

3.2. Descripción del sistema propuesto

Se realiza una descripción del procedimiento propuesto apoyado en el diagrama de bloques mostrado en la figura 3.1. Cada bloque o módulo propuesto es desarrollado, explicado y detallado atendiendo en todo momento a las referencias del estado del arte expuesto. Para cada módulo se describe tanto los problemas surgidos como las soluciones propuestas.

Cada bloque conforma un elemento principal del proceso de segmentación como se describe a continuación:

1. **Imagen test**: partimos de una imagen test de análisis de la cual se desea segmentar la información contextual que la compone.

2. **Data-set**: cada escenario de análisis de objetos contextuales dispone de una base de datos formado por *bbs* en el espacio de color rgb y su correspondiente máscara binaria de todas las imágenes de entrenamiento que contengan los objetos de contexto buscados. Estas ventanas serán extraídas de los dos data-sets LabelMe (ver sección 2.5.1) y ADE20K (ver sección 2.5.2) que disponemos.

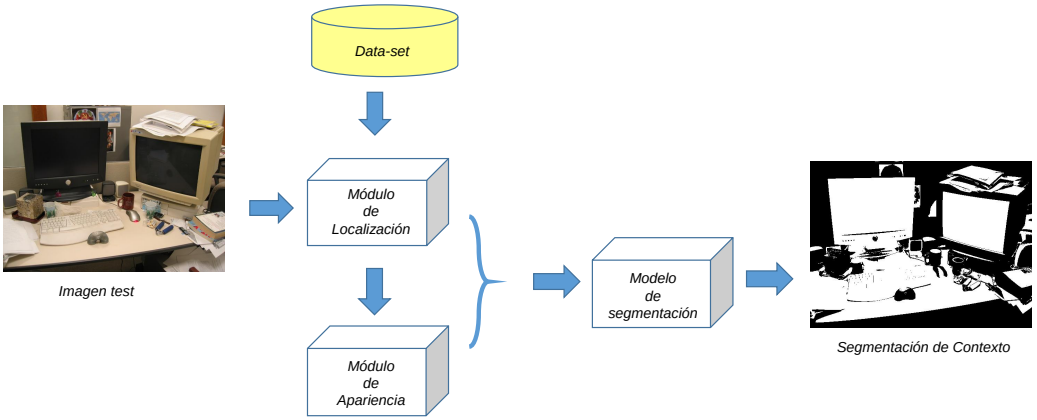

Figura 3.1: Diagrama de bloques del sistema implementado.

El contenido del data-set es un factor altamente determinante en los resultados finales que se obtienen. Por ello, para un mismo escenario se elaboran distintos data-sets con el fin de evaluar los resultados conseguidos.

3. **Módulo de localización**: con un data-set adaptado se puede analizar la imagen test deseada. En primer lugar, el módulo de localización proporciona un mapa de probabilidades que atribuye a cada pixel su probabilidad individual de pertenecer a un objeto contextual en función de su localización dentro de la imagen.

4. **Módulo de apariencia**: mediante las probabilidades del módulo de localización, el modulo de apariencia computa de nuevo la probabilidad de cada pixel de pertenecer a un objeto contextual, empleando un modelo mixto de gaussianas (GMM). Estas GMM pueden corregir malas correspondencias que se hayan producido en modelo de localización atendiendo a los niveles de color RGB que nuestro modelo ha determinado que se atribuye a un objeto contextual.

5. **Modelo final de segmentación**: anexando los dos modelos definidos anteriormente se obteniene las energías individuales que junto al modelo de energías conjuntas conformarán las energías finales para cada píxel.

3.3. Módulo de localización

En el módulo de localización integramos todo el proceso previo a la aplicación del propio modelo de localización (ver sección 3.3.3) para la creación de la máscara M de probabilidades que define la probabilidad de cada píxel de pertenecer a un objeto

contextual a partir de su localización dentro de la imagen. Sobre ella se aplica posteriormente el modelo de apariencia (ver sección 3.4). En este modelo se introducen, por tanto, todas las etapas definidas desde que se extraen los *bbs* de una imagen de entrada hasta que se obtiene el primer mapa de hipótesis de objetos contextuales. Disponiendo de un data-set adecuado a la escena de contexto a analizar para abarcar el diferente tipo de información contextual que queremos englobar, se puede aplicar el sistema desarrollado para obtener una segmentación óptima.

El módulo de localización queda esquematizado en la figura 3.2 compuesto por las estapas propuestas a continuación.

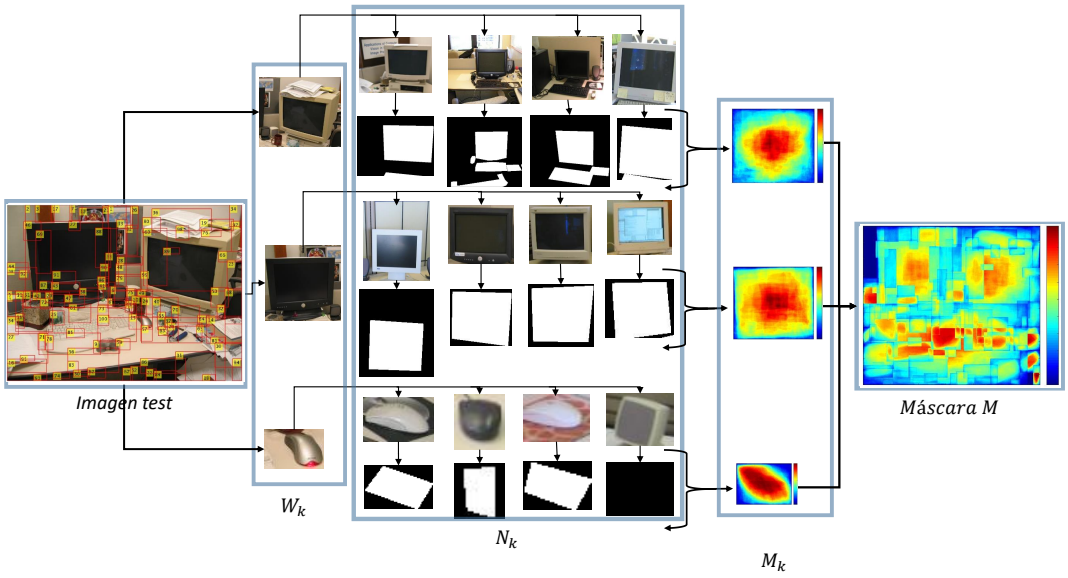

Figura 3.2: Esquema que define el módulo de localización conteniendo los distintos elementos que se generan en cada una de las etapas. El módulo de localización integra todo el proceso desde que se extraen los *bbs* de la imagen de entrada hasta que se obtiene el mapa de probabilidades del modelo de localización.

3.3.1. Generación de propuestas de objetos

Como inicio de este proceso es necesario emplear un generador de propuestas: utilizaremos EdgesBoxes (ver sección 2.3.1). Esta herramienta proporciona los *bbs* de objetos de interés que existen en nuestra imagen de entrada. Denominaremos al conjunto de *bbs* de análisis $\{W\}$. Cada uno de estos *bbs* W_k, proporcionados mediante sus coordenadas, están asociados a una puntuación según los criterios presenten en los bordes que contiene en su interior especificado en [2]. Un parámetro realmente útil que se nos permite personalizar es el número máximo de *bbs* que se desea que el

detector devuelva. Dicha limitación se puede implementar simplemente estableciendo un número máximo de ventanas que se pretenden analizar o limitando directamente una puntuación mínima, la cual es el límte que el detector no sobrepasará para devolver *bbs* con puntuaciones inferiores. De esta forma se puede descartar *bbs* que puedan tener poca relevancia en la imagen y que en la posterior aplicación del modelo de energías (ver sección 3.5.2) aporten poca información. En este trabajo se fija este máximo de ventanas en 100, número suficiente para analizar una imagen en su totalidad como se sostiene en [1]. Se puede observar en la figura 3.3 la obtención del conjunto $\{W\}$ ventanas sobre una imagen de entrada. Se aprecia como éstos pueden estar superpuestos. Esta característica es de gran utilidad ya que se podrá analizar mismas zonas de la imagen con distintas aportaciones de máscaras de entrenamiento.

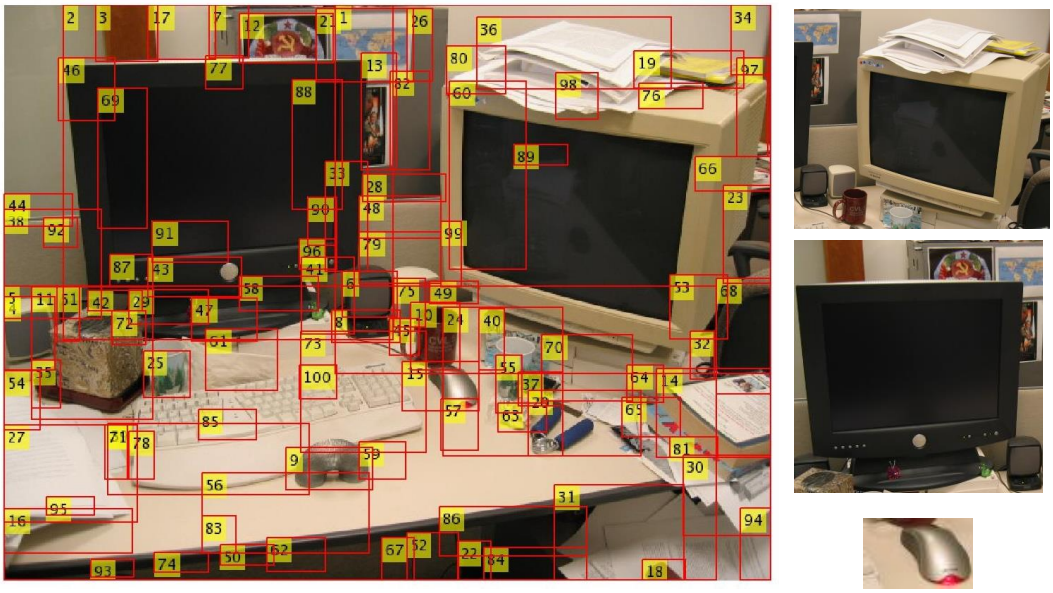

Figura 3.3: Conjunto de $\{W\}$ ventanas extraídas sobre una imagen de entrada mediante la herramienta EdgesBoxes.

3.3.2. Búsqueda de ventanas similares en los data-sets

Una vez se dispone de las ventanas $\{W\}$ se lleva a cabo su compración con todo el data-set de training que se dispone. Dicho data-set adaptado al tipo de escena que se desea analizar aporta los bbs de entrenamiento $\{N\}_{RGB}$ con un mayor parecido a cada una de las W_k.

Esta comparación de similitud se implementa empleando la herramienta descrita GIST (ver sección 2.3.2) expuesta en [5]. GIST extrae un conjunto de 16 descriptores

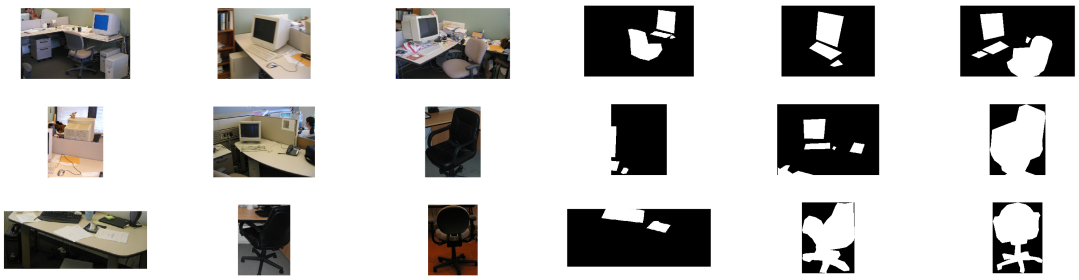

Figura 3.4: $\{N_k\}_{RGB}$ conjunto de ventanas y sus correspondientes $\{N_k\}_{MASK}$ con una mayor puntuación para la W de una propuesta concreta correspondiente al monitor de color blanco extraído de la imagen de entrada de la Figura 3.3.

por imagen que la definen en su conjunto. Obteniendo los descriptores del conjunto de ventanas $\{W\}$ así como de todas las *bbs* de entrenamiento se dispondrá de todo el grupo de ventanas parametrizadas. Realizando la comparación entre dichos parámetros de cada una de las W_k con todas las *bbs* de entrenamiento se obtiene un ranking de similitud entre ellas. Para cada W_k se guardan las 100 mejores propuestas encontradas de las *bbs* de entrenamiento que denominamos como el conjunto $\{N_k\}_{RGB}$ con sus correspondientes máscaras $\{N_k\}_{MASK}$ como se muestra en la figura 3.4. Este número concreto de ventantas de training viene determinado por [1] y es considerado óptimo para caracterizar la posterior máscara promedio por ventana W construida en el modelo de localización.

3.3.3. Modelo de localización

El modelo de localización indica la probabilidad de que un pixel i de una imagen de entrada pertenezca a un objeto contextual (clase $c_i = 1$) o al fondo (clase $c_i = 0$). Disponiendo de las *bbs* de entrenamiento $\{N_k\}_{RGB}$ y sus respectivas $\{N_k\}_{MASK}$ que mejor se corresponden con cada una de las ventanas de análisis W_k, se puede aplicar el modelo de localización expuesto en [1], para construir la máscara de probabilidades que define la totalidad de la imagen de entrada. Este proceso se realiza en varias etapas:

Máscara Mk de cada *bb* de test: En primer lugar, para cada W con sus respectivas $\{N_k\}_{MASK}$ con mayor similitud realizamos un promedio a nivel de pixel. Para ello, primero se redimensionan todas las máscaras $\{N_k\}_{MASK}$ al tamaño del *bbs* de análisis W para, posteriormente, realizar el promedio propuesto. Obtenemos con ello las máscaras de probabilidades M_k que definen cada una de las W_k . En la figura 3.5 se observa como para cada W_k obtenemos su mapa de probabilidades expresando

dentro de su área las distintas probabilidades de que en su interior se encuentre un objeto de interés de contexto que estamos analizando.

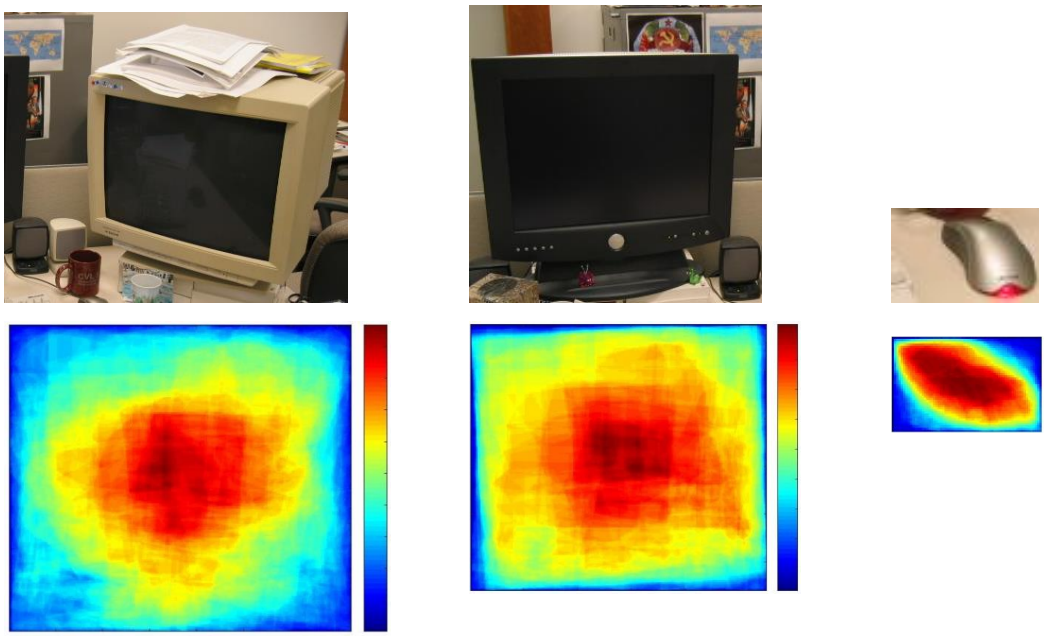

Figura 3.5: Representación de mapas de probabilidades M_k para distintas W_k

Máscara M total de una imagen de entrada: Empleando las máscaras de ventanas parciales $\{M_k\}$ obtenidas en la etapa anterior (ver sección 3.3.3) se construye una máscara total M para toda una imagen de entrada. Realizando un promedio en los píxeles donde se superponen probabilidades aportadas por diferentes *bbs* de M_k se obtiene un mapa de probabilidades global M para toda la imagen. Este mapa proporciona el modelo de probabilidades L_i de que cada pixel i pertenezca a la clase de un objeto de contexto de interés ($c_i = 1$) o a la clase complementaria de no contexto ($c_i = 0$).

$$\begin{aligned} L_i(c_i = 1) &= M(i) \\ L_i(c_i = 0) &= 1 - M(i) \end{aligned} \tag{3.1}$$

En la figura 3.6 se observa la máscara de probabilidad del modelo de localización conformada M, para una imagen de entrada.

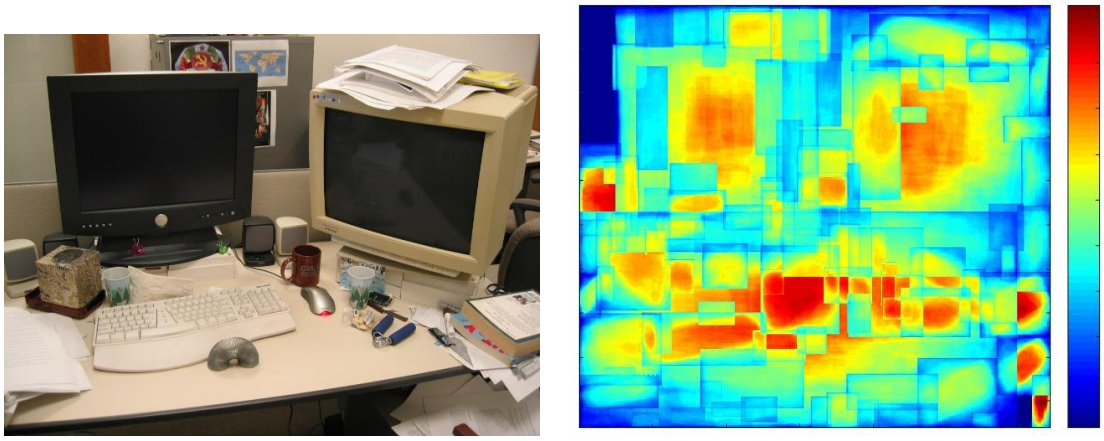

Figura 3.6: Ejemplo de una máscara M construida a partir de todas las M_k ventanas extraidas para cada W_k. Las zonas de un mayor tono rojizo (siguiendo la escala de colores) se corresponden con una zona de mayor probabilidad de contener un objeto de contexto en la escena de análisis.

3.4. Módulo de apariencia

Una vez disponemos de la máscara umbralizada M (ver sección 3.4.1) y de la imagen de entrada en el espacio de color, se puede aplicar el modelo de apariencia que definirá junto al modelo de localización nuestras potencias individuales por píxel (ver sección 3.5.1).

3.4.1. Umbralización de la máscara M

Sobre la máscara M hallada en la etapa previa (ver sección 3.3.3) aplicamos una umbralización para contruir una máscara binaria de la imagen de entrada. A su vez, se realiza la misma umbralización para la máscara inversa que contiene las probabilidades de cada píxel de contener información no contextual. Dicho umbral aplicado se obtiene por entrenamiento del sistema como se muestra posteriormente (ver sección 4.3.1).

3.4.2. Modelo de apariencia

El modelo de localización (ver sección 3.3.3) aporta una primera aproximación grosera de la localización de los objetos de contexto de interés actuando a nivel individual por píxel. El modelo de apariencia es estimado sobre una región mayor, no sólo a nivel de pixel. De esta forma la información que aporta este modelo se transfiere a los alrededores de cada píxel o incluso a distintas partes más alejadas de la imagen. Por tanto, el modelo de apariencia modela la información que define cuál es el color

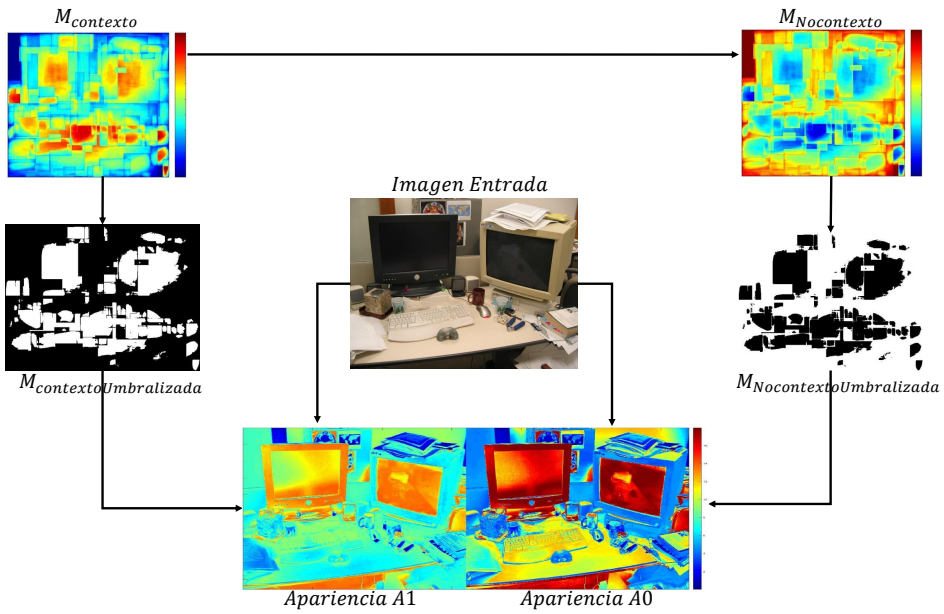

Figura 3.7: Esquema del modelo de apariencia conteniendo cada uno de los elementos intermedios que se obtienen a lo largo de su desarrollo.

predominante en los objetos de contexto de la imagen de entrada pudiendo alterar posteriormente las potencias individuales (ver sección 3.5.1) de cada píxel aunque en un primer momento nuestro modelo de localización en la transferencia de máscaras haya sugerido que una zona con dicho color pertenece a la clase de no contexto.

El modelo de apariencia se basa en el uso de dos modelos de mezclas de gaussianas (GMM) (Apendice **??**), uno para los píxeles i considerados objetos contextuales (clase $c_i = 1$) A_1 y otro para el fondo (clase $c_i = 0$) A_0 definidas en el modelo de localización (ver sección 3.3.3) expuesto en [8]. Definimos el modelo de apariencia como:

$$\begin{aligned} A(i \mid c_i = 1) &= A_1(i) \\ A(i \mid c_i = 0) &= A_0(i) \end{aligned} \tag{3.2}$$

Cada GMM tiene 5 componentes. Cada una de las componentes de los dos modelos de mezclas de gaussianas se definen mediante una media μ_g, una matriz de covarianzas σ y unos pesos asociados a la contribución sobre los datos de cada una de las componentes W_γ. Los modelo de las gaussianas generan un modelo de distribución sobre los datos de una imagen en el espacio de color RGB tanto de los datos contextuales y como de sus inversos no contextuales.

El modelo de apariencia construye, por tanto, dos mapas de probabilidades A_1 y

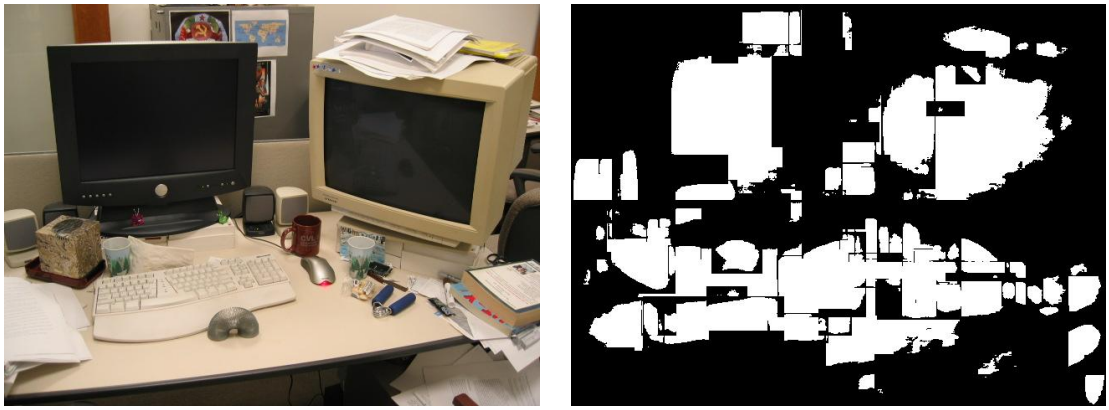

Figura 3.8: Ejemplo del umbralizado de la máscara M representada en la 3.6. Las zonas de blanco pertenecen a regiones las cuales el modelo de localización ha determinado que existe información contextual.

A_0 que atribuyen a cada píxel las posibilidades de pertenecer a la clase contexto y no contexto respectivamente como se muestra en la figura 3.9.

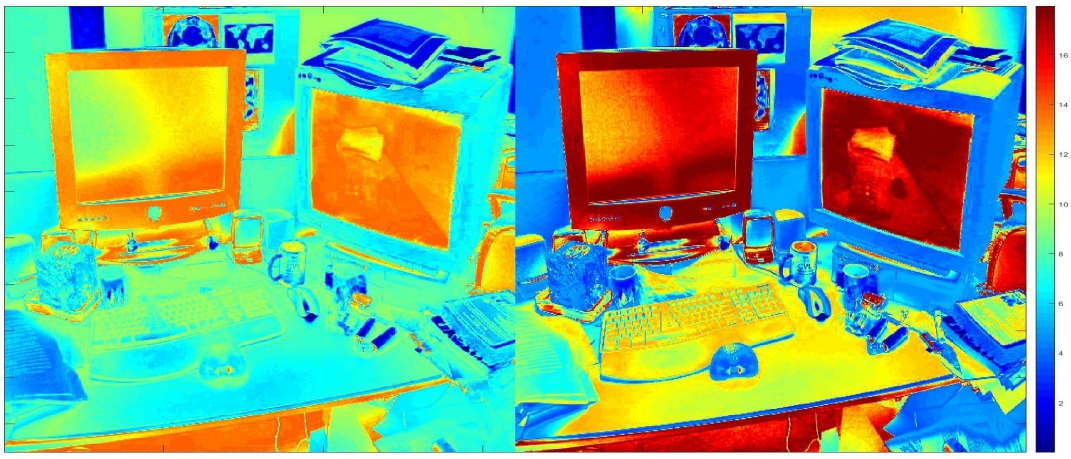

Figura 3.9: Columna izquierda: representación de mapa de probabilidades de pertenecer a no contexto A_0. Columna derecha: representación de mapa de probabilidades de pertenecer a contexto A_1. Valores más rojizos representan valores más probables de pertenecer a su respectiva clase.

3.5. Modelo final de segmentación

Disponiendo de la información de los dos modelos utilizados en este sistema de segmentación, el modelo de localización (ver sección 3.4) y el modelo de apariencia (ver sección3.3.3) se puede implementar un sistema de potencias que integre ambas

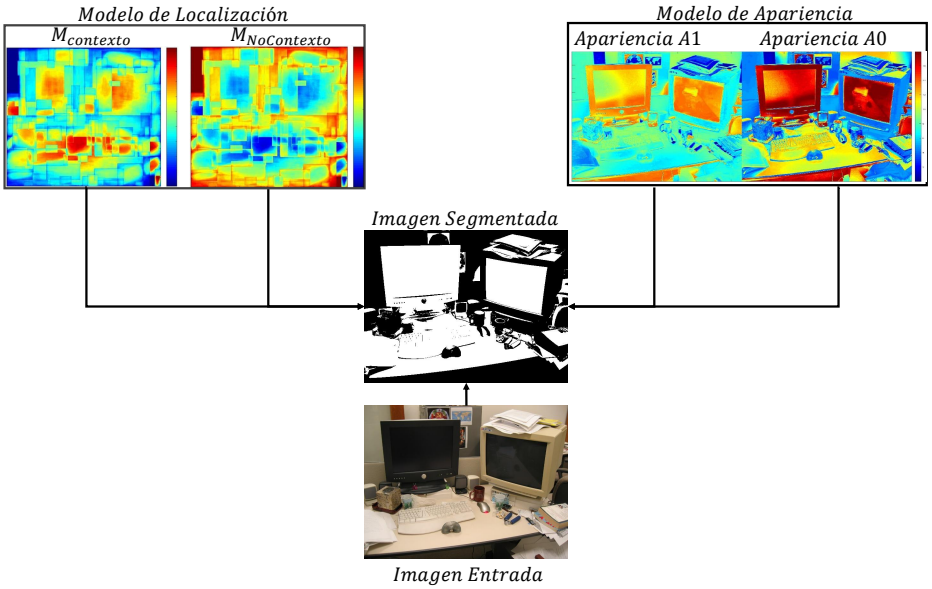

Figura 3.10: Esquema del modelo de energías para una imagen de entrada. Mediante la unión de las energías de los dos modelos implementados y el modelo de energías de segmentación final Meanfield se obtiene el resultado final.

probabilidades (ver sección3.5.1). Aplicando dicho sistema al modelo de energías final de segmentación (ver sección 3.5.2) se obtiene la segmentación resultado de nuestra imagen de entrada. El esquema que resume esta implementación se representa en la figura 3.10 para la imagen de análisis estudiada hasta ahora.

3.5.1. Energías individuales

Mediante la información de probabilidades de los dos modelos energéticos de localización y apariencia construimos el modelo de potencias individuales proporcionando la energía unitaria de cada píxel expuesto en [1]:

$$u_i(c_i) = -logA(i \mid c_i) - logL_i(c_i) \tag{3.3}$$

Esta energía u_i evalua cómo de probable es que a un pixel i se le asigne la etiqueta c_i de acuerdo a un modelo de apariencia A (ver sección 3.4.2) y a un modelo de localización L (ver sección 3.3.3) construidos. La probailidad definida por esta union de energías se puede observar visualmente en la figura 3.11 para la imagen de entrada estudiada hasta ahora.

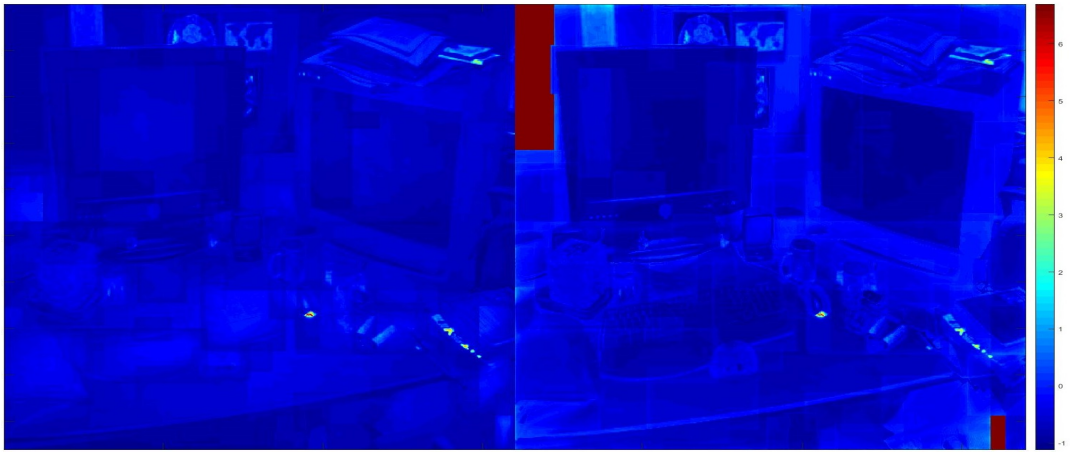

Figura 3.11: Imagen izquierda: representación de probabilidades de que cada uno de los píxeles de la imagen de entrada pertenezca a la clase de no contexto. Imagen derecha: representación de probabilidades de que cada uno de los píxeles de la imagen de entrada pertenezca a la clase de contexto. En este caso, al ser una unión de probabilidades negativa, tonos más azules representan mayor probabilidad de pertenecer a su respectiva representación.

3.5.2. Segmentación por miniminización de energías

Se propone un modelo similar al empleado en [9][10][11][12][13]. Se definen las siguientes energías.

$E(C)$ representa la unión de las energías individuales $U(C)$ junto con las energías conjuntas $V(C)$, descritas a continuación, sobre el conjunto de etiquetas C.

$$E(C) = U(C) + V(C) \tag{3.4}$$

Las energías individuales $U(C)$ se expresan mediante la suma de energías individuales en cada píxel $u_i(c_i)$ sobre una etiqueta determinada.

$$U(C) = \sum_i u_i(c_i) \tag{3.5}$$

Las energías conjuntas $V(C)$ se representan mediante la suma de las energías conjuntas individuales para cada píxel sobre cada etiqueta que dispone nuestro sistema.

$$V(C) = \sum_{ij\epsilon\varepsilon} v_{ij}(c_i, c_j) \tag{3.6}$$

, donde i expresa cada pixel de la imagen de entrada, u_i y v_i las energías indivuduales y conjuntas respectivamente y ε el conjunto de bordes que conectan los

contornos de un vecindario de una cuadrícula *3x3*. La segmentación de una imagen de entrada la definimos como el etiquetado óptimo $C*$:

$$C* = arg\,min_c E(C) \tag{3.7}$$

El valor ideal para esta función de energía es encontrada eficientemente empleando la técnica *MeanField*. Este método implementa la conexión presente en un campo aleatorio condicional (CRF) sobre los píxeles de una imagen como se define en [14]. Para que este funcionamiento sea eficiente, se realiza empleando un modelo en el que las energías conjuntas $v(i,j)$ son halladas por una combinación lineal de kernel gaussianos. Estas energías conjuntas $v(i,j)$ poseen la siguiente forma:

$$v(i,j) = \mu(i,j) \sum_{m=1} w^{(m)} k^{(m)}(f_i, f_j) \tag{3.8}$$

, donde cada $k^{(m)}$ es un kernel gaussiano $k^{(m)}(f_i,\ f_j) = exp(-\frac{1}{2}(f_i - f_j)^T \Lambda^{(m)}(f_i - f_j))$, los vectores f_i y f_j son vectores de características para los píxeles i y j en un espacio de características arbitrario, $w^{(m)}$ una combinación lineal de pesos y donde $\mu(i,j)$ representa una función de compatibilidad dada por el *modelo de Potts* en el que $\mu(i,j) = [i \neq j]$. Cada kernel $k^{(m)}$ es caracterizado por una matriz de precisión simétrica definida positivamente $\Lambda^{(m)}$, la cual establece su propia forma.

Finalmente, se obtiene la segmentación buscada representada, ver ejemplo en la Figura 3.12.

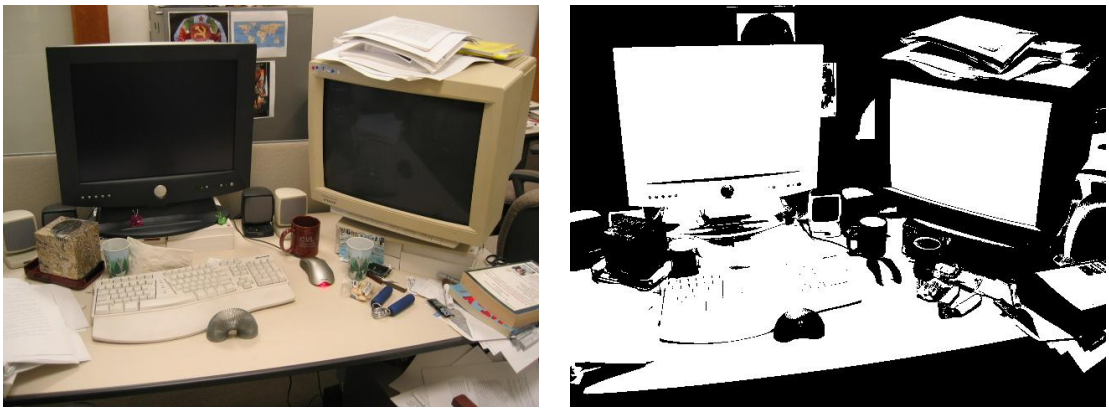

Figura 3.12: Representación de la segmentación final de los objetos contextuales buscados (representados en blanco en la máscara).

Capítulo 4

Evaluación

4.1. Introducción

En este capítulo se decribe la una evaluación del sistema implementado con el fin de comprobar su funcionalidad. Para llevarlo a cabo, se exponen diversos experimentos para distintos escenarios de información contextual.

Como ya se ha comentado, en este proyecto los data-sets cobran una vital importancia, por lo que se analizará profundamente de que forma participan los data-sets conformados sobre los resultados finales de segmentación.

4.2. Marco de evaluación

Para la realización de las pruebas posteriores se ha dividido la construcción de los data-sets para dos entornos diferentes: oficina y exterior. Para cada caso dispondremos de un data-set principal que contiene las imágenes segmentadas con todos los objetos contextuales presentes. El otro tipo de data-sets viene dado por *bbs* de interés para cada entorno. En total se crearán tres data-sets este último tipo: dos para un entorno de oficina y uno para un entorno de exterior. Por tanto, los experimentos que se realizan a continuación están organizadas de forma que para cada data-set de *bbs* se obtiene unos resultados específicos pudiendo así observar la variablidad que se producen al variar el data-set de entrenamiento empleado. En total se han realizado cuatro experimentos: tres para el escenario de oficina empleando dos data-sets alternativos y uno para un escenario exterior con un único data-set. En todos los casos se ha recurrido a las máscaras anotadas proporcionadas por LabelMe (ver sección2.5) y ADE20K (ver sección 2.6). :

4.2.1. *Data-set*

Para analizar un escenario de oficina se ha estudiado la presencia de los objetos: pantalla, teclado, ratón y silla. Para el caso de un escenario de exterior se ha analizado la presencia de la información contextual presente de: carretera, edificio, muros y suelo.

Data-sets de máscara de imágenes*:* Este data-set está constituido con las máscaras conjuntas de las imágenes que contengan todos objetos contextuales de interés como se muestra en la figura 4.1.

Data-sets de *bbs:* Estos data-sets han sido creados principalmente a partir de *bbs* de entrenamiento proporcionados directamente por LabelMe. Por tanto, estos *bbs* están centrados en los objetos contextuales de interés. Sin embargo, esta característica no siempre es útil ya que ventanas devueltas por el generados de propuestas (ver sección 3.3.1) no siempre se corresponden con la información contextual buscada o al menos no en toda su extensión. Para compensar esta singularidad se emplean en el conjunto de entrnamiento *bbs* extraídos por la herramienta generadora de propuestas *EdgesBoxes* sobre las máscaras conjuntas definidas en el parrafo anterior. Este conjunto de entrenamiento se ve reflejado con algunos ejemplos en la figura 4.2.

Figura 4.1: Columna derecha: máscaras conjuntas de imágenes en un entorno de oficina (*ground truth*). Columna izquierda: máscaras conjuntas de imágenes en un marco exterior.

Figura 4.2: Ejemplo *bbs* en RGB y sus análogas máscaras para el data-set de oficina.

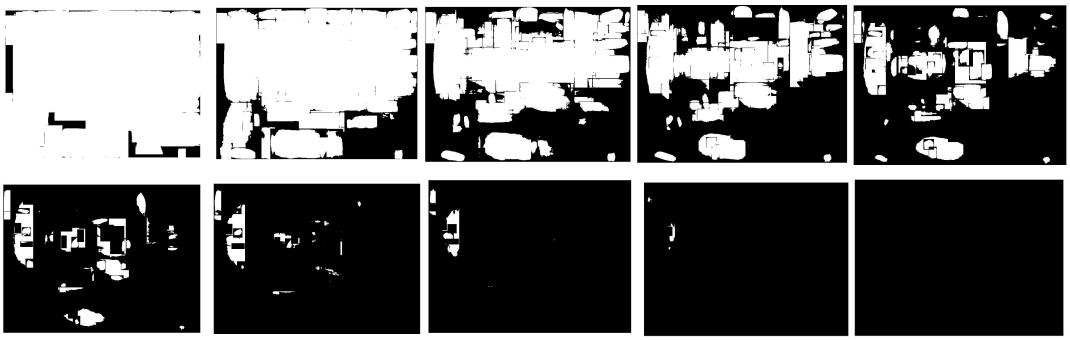

Figura 4.3: Ejemplo de evolución de la máscara binaria de una imagen de entrada obtenidas a partir de su máscara M de probabilidades extraída del modelo de localización recorriendo varios umbrales de entrenamiento. Primera fila: umbrales desde 0 hasta 0.4. Segunda fila: umbrales desde 0.5 hasta 0.9. Se observa la evolución de dichas máscaras: a emdida que aumentamos el umbral se reduce la zona de contexto seleccionado.

4.3. Entrenamiento del sistema

4.3.1. Umbral sobre las máscaras M: t_α

Aplicando nuestro sistema sobre un conjunto de imágenes de entrada de entrenamiento se puede comparar los resultados obtenidos para distintos umbrales. Rea-

lizando dicha comparación de los resultados con el mismo conjunto *ground truth* de un total de1500 máscaras para dichas imágenes, se consigue el umbral óptimo para el cual las máscaras se adaptan mejor a los resultados ideales.

Se puede observar la evolución sobre una imagen de entrada variando los umbrales sobre la máscara M de probabilidades en la figura 4.3. Se ha realizado el cálculo de dicho umbral para el *data-set* de entorno de oficina utilizándose la técnica *Valor-F* o *F-Score* [Apéndice B] obteniéndose las distribuciones presentadas en la figura 4.4. Se puede observar como el mayor F-Score se obtiene empleando un umbral de 0.5, por lo que, al analizar una imagen de entrada en el modelo de localización en este *data-set* será el umbral utilizado.

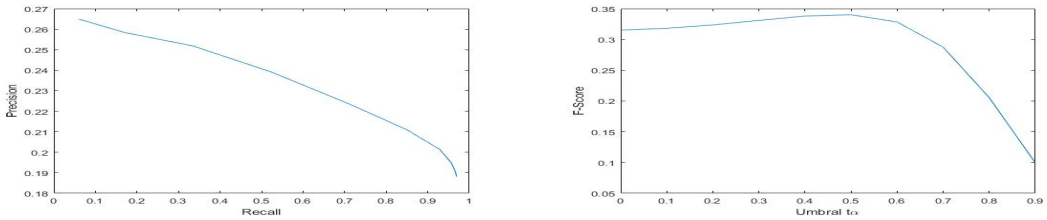

Figura 4.4: Columna izquierda: representación de los resultados obtenidos de Precisión (Precision) vs. Exhaustividad (Recall). Columna derecha: el valor *F-Score* en función de los distintos umbrales analizados sobre la base de datos de entrenamiento empleando nuestro sistema. Se observa como el mayor F-Score se obtiene para un umbral de 0.5.

4.4. Pruebas y resultados

A partir de los data-sets de entrenamiento se han realizado disitintas pruebas para distintas imágenes de entrada para evaluar el sistema implementado sistema. En todas ellas se emplea el sistema de evaluación *F-Score* para medir la eficacia de la herramienta propuesta.

4.4.1. Experimento 1

Mediante el conjunto de entrenamiento de *bbs* de elementos de oficina se ha realizado sobre diferentes imágenes de un conjunto de *test* etiquetado las prueba de nuestro sistema de obtención de la información contextual contenida. Visualmente para cada una de las imágenes test se ha obtenido las máscaras binarias finales de segmentación presentadas en la figura 4.5.

Evaluando el sistema mediante el método de puntuaciones *F-Score* se han obtenido los valores representados en la tabla 4.1.

Figura 4.5: Experimento 1: máscaras binarias obtenidas por nuestro sistema contenientes de objetos contextuales expresados con valor uno. Se muestra tanto las imagenes originales en la primera fila como sus correspondientes máscaras obtenidas en la segunda fila.

	Imagen Test 1	Imagen Test 2	Imagen Test 3	Imagen Test 4
Precisión	0.4086	0.4438	0.5331	0.4908
Exhaustividad	0.8286	0.8623	0.8554	0.8596
F-Score	0.5473	0.5860	0.6569	0.6249

Tabla 4.1: Experimento 1: resultados obtenidos para las distintas imágenes de *test* analizadas. Se observa como los valores F-Scores obtenidos ofrecen buenos resultados para este *data-set* siendo estos superiores al 0.5 de puntuación en todos los casos analizados.

4.4.2. Experimento 2

Mediante un conjunto de entrenamiento alternativo construido con un conjunto aleatorio diferente de *bbs* a la evaluación anterior seleccionados (ver sección 4.4.1) se han realizado las mismas pruebas obteniéndose los resultados expuestos en la figura 4.6.

Evaluando, de nuevo el sistema mediante el método de puntuaciones *F-Score* se han obtenido los valores representados en la tabla 4.2.

Figura 4.6: Experimento 2: máscaras binarias obtenidas por nuestro sistema contenientes de objetos contextuales expresados con unos. Se muestra tanto las imagenes originales en la primera fila como sus correspondientes máscaras obtenidas en la segunda fila.

	Imagen Test 1	Imagen Test 2	Imagen Test 3	Imagen Test 4
Precisión	0.3681	0.3857	0.1074	0.1962
Exhaustividad	0.4507	0.9088	0.4446	0.7714
F-Score	0.4053	0.5416	0.1730	0.3128

Tabla 4.2: Experimento 2: resultados obtenidos para las distintas imágenes de *test* analizadas. Se observa como para un data-set alternativo, las puntuaciones varían en gran medida, prácticamente no superando el 50 % de los casos el porcentaje de acierto aportado por F-Score.

4.4.3. Experimento 3

Realizando una prueba con sobre imágenes de *test* que contienen objetos contextuales en un entorno de exteriores y empleando una base de datos personalizada para este tipo de información contextual se han obtenido los resultados respresentados en la figura 4.7.

4.4.4. Experimento 4

Como última evaluación se ha realizado la prueba de el sistema implementado sobre un vídeo grabado estáticamente. En dicho vídeo se aprecia un entorno contextual de oficina sobre el que se cruzan dos personas andando. Analizando cada uno de los *frames* con nuestro sistema y guardando la segmentación de información contextual resultante se puede obtener que zonas que componen la parte contextual de la imagen de todos los *frames* están más de la mitad del tiempo presentes. Los resultados

Figura 4.7: Experimento 3: representación de la segmentación final de objetos contextuales para un entorno de exterior. Se observa como se consigue seleccionar información contextual de interés aunque los resultados no son del todo óptimos. El cielo que para este *data-set* no se ha considerado contexto si es correctamente descartado. Sin embargo otros objetos no considerados contextuales en este caso como son los coches si se etiquetan como información contextual.

obtenidos para algunos de los *frames* ha sido el representado en la figura 4.8.

Como resultado final se ha obtenido la mediana de todos los *frames* que componen el vídeo obteniendo la imagen representada en la figura 4.9

Figura 4.9: Experimento 4: mediana de todos los frames de segmentación que componen el vídeo. Se puede apreciar como los objetos contextuales correspondientes a las pantallas se mantienen presentes en la mediana de la segmentación.

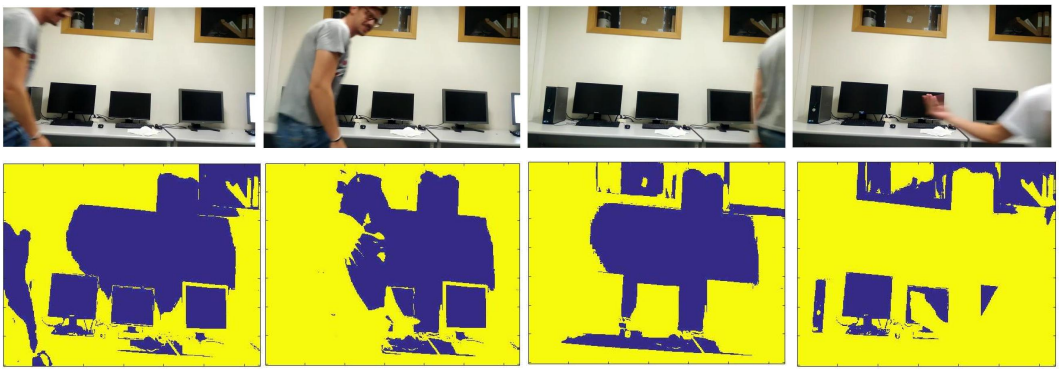

Figura 4.8: Experimento: segmentación obtenida para algunos de los frames originales del vídeo grabado en un entorno con objetos contextuales de oficina (en el VPULab).

4.5. Discusión

Se pueden extraer importantes conclusiones a cerca de los resultados obtenidos.

En general se obtiene unas soluciones aceptables para los distintos experimentos realizados. Con el primer experimento (ver sección 4.4.1) hemos comprobado como el sistema implementado reconoce correctamente los objetos contextuales que están claramente definidos y contrastados en la imagen. En su contra, objetos mimetizados entre demás objetos no contextuales son reconocidos de forma inexacta por la compensación realizada por el módulo de apariencia.

También podemos determinar, como se ha observado en el experimento número uno y dos conjuntamente (ver sección 4.4.2), el data-set utilizado para un mismo escenario tiene una gran influencia en la segmentación final obtenida.

Para un escenario de exterior (ver sección 4.4.3) los resultados obtenidos son menos positivos. Al existir tanta diversidad de objetos y utilizar para transferir las máscaras *bbs* de grandes dimensiones, la información de objetos contextuales va homogeneizar la información sobre todo el *bbs* obviando información que no es contextual.

Por último, para la prueba realizada en un vídeo (ver sección 4.4.4) se obtienen unos resultados muy favorables. Como se obtiene en la figura promedio de fondo se mantiene constante la presencia de objetos contextuales.

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

Se ha conseguido el objetivo principal de este trabajo: la segmentación espacio-temporal de los objetos contextuales de un vídeo. Para ello, como se propuso se ha empleado el método de transefrencia de máscaras de un conjuto de entrenamiento sobre una imagen de entrada.

Para abordar el objetivo general se han desarrollado distintas etapa intermedias. Se ha realizado un profundo análisis del estado del arte actual que envuelve a esta técnica abarcando una gran cantidad de información que nos ha servido de base para implementar el proyecto. También se han elaborado consistentes data-sets específicos para cada marco de análisis valorando en cada tipo de escena que consideramos objetos contextuales.

Un proceso primordial para obtener un resultado positivo ha sido la correcta organización del algoritmo que desarrolla la segmentación por tranferencia de máscaras. Finalmente, y no por ello menos importante, se ha buscado en todo momento la implementación de un sistema totalmente automático, reto que ha conseguido mejorar muchos sitemas de segmentación existentes.

5.2. Trabajo futuro

A la vista de los resultados que se han obtenido en este trabajo se propone trabajar en varias mejoras:

Desarrollo de mejoras a nivel de entrenamiento del sistema como es estimar el número de gaussianas empleadas que produciría una mejor estimación más ajustados a los datos de contexto buscados.

A su vez, la búsqueda y obtención de data-sets lo mejor adaptados a cada escenario mejoraría enormemente el sistema. Esto se podría realizar con un entrenamiento que aleatoriamente introdujese distintos *bbs* el data-sets utilizado analizando con que combinación se obtiene la mejor segmentación.

Por último, se podría probar distintos modelos de minización de energías analizando cual de ellos aporta un mejor resultado.

Bibliografía

[1] V. Ferrari, "Figure-ground segmentation by transferring window masks," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, (Washington, DC, USA), pp. 558–565, IEEE Computer Society, 2012. xiii, 2, 6, 7, 11, 14, 15, 20

[2] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV*, 2014. 4, 13

[3] J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," in *arXiv:1503.00848*, March 2015. 5

[4] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, pp. 2189–2202, Nov. 2012. 5

[5] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vision*, vol. 42, pp. 145–175, May 2001. 6, 14

[6] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *Int. J. Comput. Vision*, vol. 77, pp. 157–173, May 2008. 7

[7] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," *CoRR*, vol. abs/1608.05442, 2016. 8

[8] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut": Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, pp. 309–314, Aug. 2004. 18

[9] R. Ranftl and T. Pock, *A Deep Variational Model for Image Segmentation*, pp. 107–118. Cham: Springer International Publishing, 2014. 21

[10] J. Kim and K. Grauman, *Shape Sharing for Object Segmentation*, pp. 444–458. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. 21

[11] J. Wang, J. M. Siskind, T. Kubota, and S. Wang, "Salient closed boundary extraction with ratio contour," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 27, pp. 546–561, 2005. 21

[12] X. Zhang, Y.-H. Yang, Z. Han, H. Wang, and C. Gao, "Object class detection: A survey," *ACM Comput. Surv.*, vol. 46, pp. 10:1–10:53, July 2013. 21

[13] A. K. Sinop and L. Grady, "A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm," *2007 11th IEEE International Conference on Computer Vision*, vol. 00, pp. 1–8, 2007. 21

[14] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *CoRR*, vol. abs/1210.5644, 2012. 22

Apéndice A

Filtros de Gabor

El filtro de Gabor es un filtro lineal cuya respuesta de impulso es una función sinusoidal multiplicada por una función gaussiana. Son funciones casi paso banda.

La principal ventaja que se obtiene al introducir la envolvente gaussiana es que las funciones de Gabor están localizadas tanto en el dominio espacial como en el de la frecuencia, a diferencia de lo que ocurre con las funciones sinusoidales, que están perfectamente localizadas en el dominio frecuencial y completamente deslocalizadas en el espacial (las funciones sinusoidales cubren todo el espacio). Por tanto, son funciones más adecuadas para representar una señal conjuntamente en ambos dominios.

La transformada de Fourier de un filtro de Gabor son gaussianas centradas en la frecuencia de la función sinusoidal (siendo estas gaussianas la transformada de Fourier de la gaussiana temporal o espacial). Se puede llegar a este resultado empleando la propiedad de convolución de la Transformada de Fourier, que transforma los productos en convoluciones. Así, la transformada de la respuesta de impulso de Gabor es la convolución de la transformada de la función sinusoidal y de la transformada de la función gaussiana.

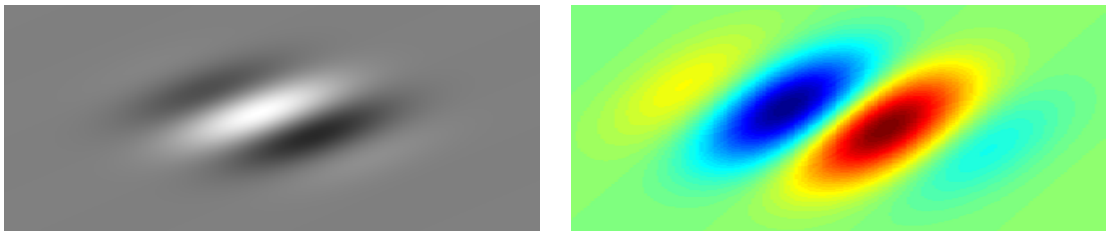

Figura A.1: Ejemplos visuales de filtros de Gabor. Columna izq: parte real de la respuesta de impulso de un filtro de Gabor. Columna dcha: filtro de Gabor de dos dimensiones diferenciadas

Apéndice B

Valor-F (F-Score)

El Valor-F (denominada también F-score o medida-F) en estadística es la medida de precisión que tiene un test. En nuestro caso dicho test es el resultado de la segmentación de la imagen de entrada que analizamos. Se emplea en la determinación de un valor único ponderado de la precisión y la exhaustividad. Es utilizado en la fase de pruebas de algoritmos de búsqueda y recuperación de información y clasificación como es en nuestro caso.

Se define como:

$$F_1 = 2 \cdot \frac{Precisión \cdot Exhaustividad}{Precisión + Exhaustividad}$$

siendo $Precisión = \frac{TruePositives}{(TruePositives+FalsePositive)}$ y $Recall = \frac{TruePositives}{(TruePositives+FalseNegative)}$.

Realizando la analogía con nuestro sistema entre la imagen de entrada de análisis y el conjunto de entrenamiento se tiene:

TruePositive: se corresponde con los aciertos de nuestro sistema sobre un píxel que debe ser contexto y lo es sobre la imagen de entrenamiento.

FalsePositive: se corresponde con los fallos de nuestro sistema sobre un píxel que debe no debe ser contexto y lo es sobre la imagen de entrenamiento.

TrueNegative: se corresponde con los aciertos de nuestro sistema sobre un píxel que debe debe no debe ser contexto y no lo es sobre la imagen de entrenamiento.

FalseNegative: se corresponde con los fallos de nuestro sistema sobre un píxel que no debe debe ser contexto y lo es sobre la imagen de entrenamiento.