Universidad Zaragoza
1542

Proyecto Fin de Carrera

Reconocimiento de acciones egocéntricas desde una cámara RGB-D montada en un casco

Autor

Pablo Azagra Millán

Directores

Ana Cristina Murillo Arnal
Luis Montesano del Campo

Departamento de Informática e Ingeniería de Sistemas (DIIS)
Escuela de Ingeniería y Arquitectura

Abril 2014

Reconocimiento de acciones egocéntricas desde una cámara RGB-D montada en un casco.
RESUMEN

En este proyecto se ha implementado un sistema de reconocimiento de acciones mediante el uso de una cámara que lleva el propio usuario sujeta en un casco. El objetivo es analizar la capacidad de reconocimiento de una cámara RGB-D portada por el propio usuario (es decir, según el término en inglés, una camara "wearable"). El objetivo a más largo plazo será poder reconocer las acciones de un individuo en primera persona, para su posterior análisis en diferentes aplicaciones, desde sistemas de guiado de instrucciones para realizar una tarea complicada, asistencia a discapacitados visuales o monitorización de la actividad de una persona por motivos de salud o rehabilitación. Este proyecto está incluido dentro de las líneas de investigación del grupo de Robótica, Percepción y Tiempo Real de la Universidad de Zaragoza.

Las tareas realizadas en este proyecto han sido las siguientes: como parte del análisis del problema, se ha realizado un estudio de las imágenes RGB-D y de la información que proporcionan. También se han estudiado distintas opciones de segmentación en dichas imágenes para mejorar la extracción de información y de los posibles descriptores para representar y comprimir dicha información. Por último se han estudiado dos de los clasificadores más utilizados en materia de visión por computador para tareas de reconocimiento. Además se ha realizado un etiquetado de referencia con varias secuencias de imágenes para su uso en los experimentos.

En materia de implementación, se ha diseñado e implementado un módulo que procesa y segmenta las imágenes, y obtiene los descriptores deseados. Se ha implementado un módulo de clasificación que, utilizando los descriptores calculados previamente, realiza un entrenamiento y dada una nueva secuencia realiza un reconocimiento de acciones.

Como análisis de rendimiento del reconocedor, se han diseñado y realizado un conjunto de experimentos comparativos entre las distintas posibilidades de descripción y clasificación. Se ha analizado y documentado los resultados de dichos experimentos. En estos experimentos, hemos probado una clasificación en distintos niveles mediante el uso de los descriptores estudiados y hemos visto que la clasificación para un nivel básico de manipulación o no manipulación funciona muy bien, pero que los descriptores estudiados no dan suficiente información como para realizar una clasificación más precisa. Además, también se ha analizado en detalle cuánto influyen los distintos descriptores y las posibles combinaciones de los mismos.

Agradecimientos

En primer lugar quiero agradecer a Ana Cris y a Luis la labor de dirección y supervisión realizada, sin la cual este trabajo no se hubiera podido llevar a cabo. Quiero expresar mi gratitud sobre todo por ser pacientes conmigo, ayudarme cuando algo no funcionaba y guiarme a lo largo de todo el proyecto.

En segundo lugar, quiero mostrar mi agradecimiento a mi novia Clara que ha estado soportándome y dándome consejos durante todos los meses que ha durado este proyecto. Sin sus visitas al laboratorio con una sonrisa y sus ánimos este proyecto hubiera supuesto un camino mucho más difícil.

Finalmente quiero agradecer muy especialmente a mi familia y amistades más allegadas su apoyo e interés incondicional, así como una enorme paciencia.

Gracias a todos.

Pablo Azagra Millán

Índice general

Capítulo 1
Introducción

La captura fotográfica existe desde el Siglo XIX, siendo los primeros modelos analógicos, extremadamente caros y muy grandes. Durante el Siglo XX la evolución de la tecnología permitió que estas cámaras pasaran de ser analógicas a digitales, de caros a precios asequibles para la mayoría de la gente y de tamaños grandes a ser compactas e incluso pequeñas. La reducción del tamaño de las cámaras ha traído consigo la posibilidad de utilizar cámaras que sean 'vestibles', o como se denominan en inglés *wereables*, como por ejemplo el modelo GoPro[1] utilizado ampliamente para grabación de deportistas. Usando estas cámaras es posible grabar la actividad diaria de un usuario desde su propia perspectiva.

En este tipo de grabaciones se basa este proyecto.Se podrían ver como un *"diario"* visual [1]. Las aplicaciones que se pueden encontrar a este tipo de procesado de imágenes van desde simple "diario"de información hasta aplicaciones médicas, como la monitorización de la evolución de un paciente, aplicaciones de entretenimiento (juegos de realidad virtual) o aplicaciones demográficas (analizar qué porcentaje del tiempo dedica la gente a distintas actividades al día) o sociales (análisis de interacción entre grupos y personas).

En materia de reconocimiento de acciones hay una gran cantidad de trabajos basado en reconocer acciones donde el usuario es completamente visible e incluso se puede obtener un exoesqueleto de él. Sin embargo, el cambiar de perspectiva a verlo en primera persona, donde se ve parcialmente zonas del cuerpo y no siempre centrado o con movimientos del usuario que dificultan el reconocimiento, provoca que haya que cambiar la forma de tratar la información y que los trabajos basados en la otra perspectiva no sean validos para este nuevo enfoque. Dependiendo de qué cámara y qué posición, vamos a encontrar problemas y ventajas muy distintas (unas gafas *vuzix*[2],un iPhone[3] con objetivo *fisheye*[4] colgando del cuello, cámara RGB-D (visió y profundidad) montada en un casco, una GoPro a la altura del pecho...). Para este proyecto se va a utilizar una cámara RGB-D, en particular una *Asus Xtation Pro*, en el Anexo A hay más información, montada en un casco.

[1] `http://es.gopro.com/`
[2] `http://www.vuzix.com/UKSITE/consumer/products_wrap920ar.html`
[3] `http://www.apple.com/es/iphone-4s/specs/`
[4] `http://es.wikipedia.org/wiki/Objetivo_ojo_de_pez`

El objetivo principal de este proyecto es realizar un reconocedor de acciones en imágenes obtenidas de una cámara de visión y profundidad (RGB-D) portada por el usuario. En más detalle, este proyecto ha consistido en las siguientes tareas:

1 Estudiar como procesar e interpretar imagenes RGB-D. El estudio de cómo procesar las imágenes RGB-D se centró en cómo segmentar la imagen. La segmentación estudiada ha sido basada en segmentar pixels que contienen piel, segmentar planos en la escena y segmentación en superpixels.

   Después de estudiar la segmentación, se ha trabajado en diseñar un buen descriptor para estas imágenes, en particular se ha diseñado como representar la distribución de pixels de piel en la imagen.

2 Estudiar y analizar distintos clasificadores para identificar distintas acciones que pueden ocurrir en la imagen. Con los descriptores diseñados y otros estudiados de la literatura, se han estudiado sistemas de clasificación de distintos tipos y se ha realizado una propuesta de un sistema de clasificación de acciones.

3 Realizar un etiquetado de las secuencias del dataset dado para entrenamiento y test del sistema de clasificación. El dataset utilizado está público en la web del grupo de investigación[5], como se detalla más adelante en esta memoria.

4 Diseñar y realizar un conjunto de experimentos exhaustivos para evaluar el rendimiento de distintas modificaciones del sistema propuesto. Se diseñaron y realizaron pruebas para un conjunto de combinaciones de los distintos descriptores junto con los clasificadores y sus posibles variaciones. También se experimento con distintas formas de combinar la clasificación (por niveles o todo junto,etc).

5 Analizar los resultados de esos experimentos. Se realizó un análisis y documentación de estos experimento que nos indicaban las mejores opciones de todas las combinaciones y como de fiables es la clasificación.

1.1. Distribución del tiempo empleado

En la Figura 1.1 se muestra como se distribuido el tiempo empleado mediante un diagrama de Gantt.

1.2. Trabajo relacionado

El trabajo de visión por computador utilizando cámaras 'egocéntricas' es bastante novedoso pero cuenta con algunos artículos interesantes en los últimos años que muestran distintas técnicas y posibles aplicaciones. Encontramos trabajos como [2] que se centra en

[5] `http://robots.unizar.es/omnicam/wcvs_data/`

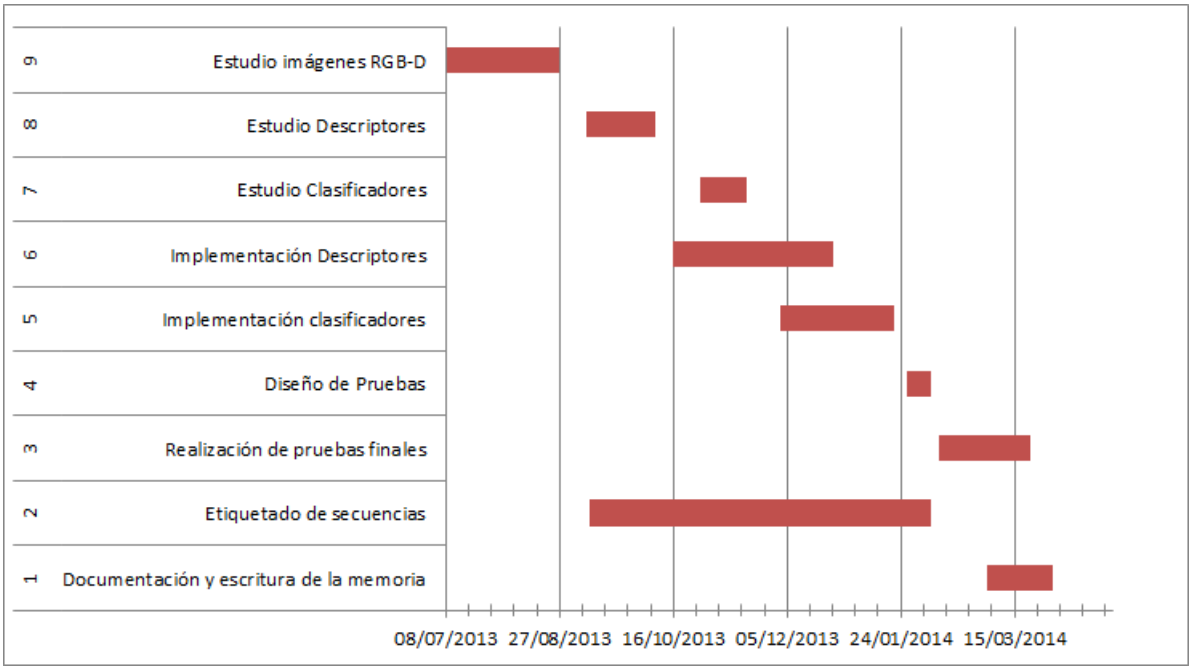


Figura 1.1: Distribución tiempo empleado.

el reconocimiento de una mano y su posible forma (cuantos dedos, a donde apunta, etc) para identificar las acciones e intenciones de interacción con el entorno o [3] que estudia la interacción de otras personas sobre el usuario y [8] que busca realizar un resumen de acciones de la interacción. En el caso de [4] tenemos otro trabajo que se centra en el intento de predecir la mirada del usuario como elemento clave para identificar la acción realizada.

Los trabajos relacionados con el reconocimiento de acciones manuales son comunes en el ámbito de la visión por computador, por ejemplo, en [7] se utiliza una ayuda externa, un guante, para poder recrear en 3D los movimientos de la mano o en [9] se centran en reconocer diversos gestos de una mano, algo parecido a lo visto en [2], realizándose ambos trabajos desde una perspectiva distinta a la egocéntrica.

Tanto en los trabajos con cámaras egocéntricas como en trabajos de visión por computador en general,se utiliza la segmentación de la imagen como pre-procesado. Mas detalles sobre técnicas de segmentación estudiadas se pueden encontrar en la seccion 2.2.2 de este proyecto. De especial importancia para nuestro trabajo son métodos anteriores que se centran en la segmentación de pixels que contienen piel, como [5] y [6]. Otros trabajos de segmentación más sencillos y rápidos, se basan solo en máscaras simples, como el trabajo de [13], más cercano al tipo de segmentación estudiada en este trabajo.

1.3. Entorno de trabajo

En cuanto a las herramientas utilizadas, este proyecto se ha desarrollado sobre la distribucion de Linux *Ubuntu*, en sus versiones 10.04 y 12.04. Inicialmente se empezó utilizando el pseudo sistema operativo *ROS*[6] muy utilizado en materia de robótica y sistemas embebidos. Este sistema se ejecuta sobre *Ubuntu* y tiene varias herramientas que facilitan el manejo de periféricos. Este sistema se utilizó solamente durante las capturas de los datos y procesado inicial de las mismas. Después se siguió desarrollando solamente sobre *Ubuntu*. El proyecto ha sido realizado con el lenguaje C++, utilizando como base de procesamiento las librerías OpenCV[21] y PCL[14].

1.4. Estructura de la memoria

La memoria se va a dividir en 4 apartados: En el capítulo 2 desarrollaremos lo relacionado con las imágenes RGB-D utilizadas y de como han sido procesadas así como los descriptores obtenidos de ellas, en el capítulo 3 los posibles clasificadores utilizados, en el capítulo 4 los experimentos realizados y los resultados obtenidos y, por último, en el capítulo 5 haremos una conclusión del trabajo realizado así como de posibles trabajos futuros.

[6] `http://www.ros.org/`

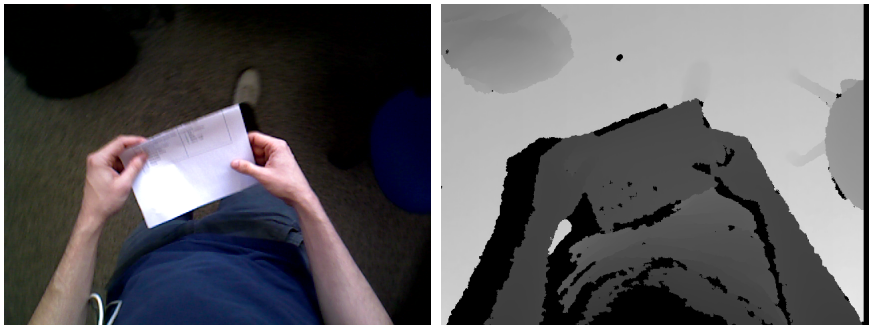
Capítulo 2

Procesado de imágenes RGB-D

El tratamiento de las imágenes adquiridas mediante una cámara de visión y profundidad (RGB-depth), en nuestro caso la cámara *Asus Xtion Pro*[1] es distinto al de las cámaras convencionales. Las cámaras RGB-D adquieren dos tipos de datos a la hora de grabar secuencias. El primero son las imágenes RGB que cualquier cámara obtiene de base. Las resoluciones de éstas suelen ser VGA o QVGA. Ademas de la imagen convencional, adquieren mediante otro sensor, un sensor de profundidad, un mapa de profundidad. En el Anexo A se describe el funcionamiento de este sensor.
En este capítulo se discutirá primero de cómo son estas imágenes y cómo las vamos a procesar, de los distintos tipos de segmentación estudiados e implementados en este proyecto y de los distintos descriptores de imagen implementados.

2.1. Imágenes RGB-D

Las imágenes RGB-D están compuestas realmente por dos imágenes como se acaba de explicar, las llamaremos: Imagen RGB y mapa de profundidad. La primera es una fotografía común obtenida con la cámara RGB como se ve en la Figura 2.1a.



(a) Imagen RGB (b) Mapa Profundidad

Figura 2.1: Imágenes RGB-D.

[1] `http://www.asus.com/Multimedia/Xtion_PRO/`

La segunda imagen obtenida representa un mapa de profundidad. Los valores de profundidad obtenidos por la cámara son entre [0,2048]. Sin embargo, para guardarlo se comprimen en imágenes de 8 bits con 1 canal. Por lo tanto tenemos imágenes como la Figura 2.1b, en escala de grises.

2.2. Segmentación de una imagen RGB-D

Después de ver cómo son las imágenes, haremos un repaso a los modelos de segmentación utilizados en este proyecto además de una comparativa entre ellos. Estos modelos no son excluyentes entre sí, se pueden aplicar conjuntamente. Por lo tanto veremos tres tipos de segmentación de la imagen y sus posibles variaciones.

2.2.1. Segmentación en superpixels

La segmentación en superpixels es un método de *clustering* de pixels. Este modelo crea una imagen formada por conjuntos de superpixels. Un superpixels es un conjunto de pixels adyacentes cuyo valor entra dentro de un rango de aceptación. El resultado final es una imagen donde cada pixel ha sido asignado a un superconjunto o 'superpixel'. Si visualizamos la imagen asignando a todos los pixels que han caido dentro del mismo superpixel el valor de la media de su superpixel veremos una imagen parecida a la Figura 2.2. Este paso nos va a ser muy útil a la hora de trabajar con los colores de pixels ya que evita variaciones entre pixels colindantes y añade robustez, por lo tanto será nuestro primer paso de pre-procesado de imagen.

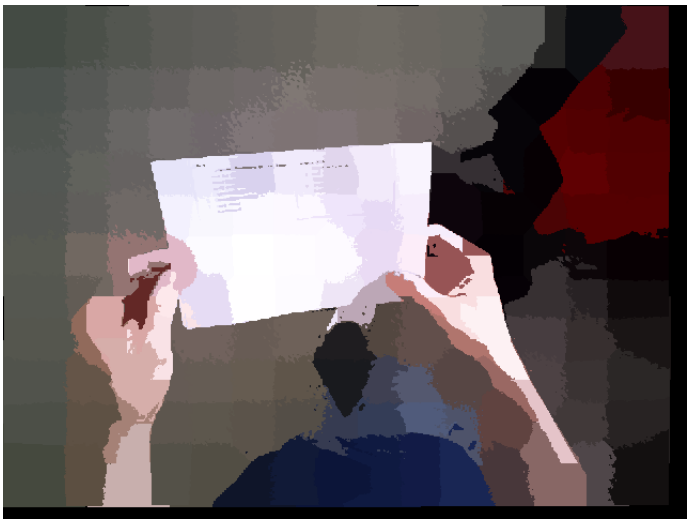


Figura 2.2: Visualización de la segmentación de una imagen en superpixels.

Se estudiaron dos tipos de técnicas recientes de entre las más utilizadas para segmentación de superpixels: *Seeds* y *SLIC*. Se decidió utilizar estos métodos debido a que es de los métodos que, por lo visto en lecturas relacionadas, mejor resultado están dando y por

ser el estado del arte de los métodos de cálculo de superpixels.

SLIC

El algoritmo de segmentación *SLIC* utilizado en este proyecto fue presentado en [10].La implementación que utilizamos está disponible en la página del trabajo[2]. Las ventajas de este algoritmo son la facilidad de uso y rapidez. El funcionamiento de este algoritmo se basa en el crecimiento de regiones.Para ello inicializa los centros de los cluster muestreando pixels. Luego, para cada cluster, calcula la distancia a los pixels más cercanos. Si esa distancia es menor que la que tenía anteriormente, ese pixel cambia la etiqueta a la del cluster actual. Esto se repite hasta alcanzar un error residual menor a un umbral. En este algoritmo el parámetro modificable es K, el número de superpixels finales deseados.

(a) 20 Superpixels (b) 50 Superpixels (c) 100 Superpixels (d) 200 Superpixels

Figura 2.3: Visualización de un ejemplo de segmentación *SLIC* con distinto número de superpixels.

En la Figura 2.3 podemos observar que con un número bajo de superpixels la imagen queda subsegmentada y se acoplan diferentes colores de forma que no deseamos. Vemos que a partir de 100 la imagen se segmenta de una manera más acertada y elegiremos 200 por ser la que mejor segmenta.

SEEDS

Este método de segmentación en superpixels fue propuesto por [11]. La implementación utilizada se encuentra en la página del trabajo[3].

El funcionamiento de este método se basa en el movimiento de bordes utilizando el algoritmo de *'Hill-climbing'* [12]. Inicialmente, divide la imagen en una cuadrícula con el número de superpixel deseado. A partir de ahí se utiliza el algoritmo de *'hill-climbing'* junto con un histograma basado en el color y una función de cálculo de bordes para calcular la posible mejoría del cambio. Por lo tanto, la modificación de un borde de un superpixel viene dada cuando la función de energía de ese superpixel aumente con el cambio. En la Figura 2.4 podemos ver la diferencia de funcionamiento con respecto a otros metodos.

[2] `http://ivrg.epfl.ch/research/superpixels`
[3] `http://www.vision.ee.ethz.ch/~boxavier/seeds/`

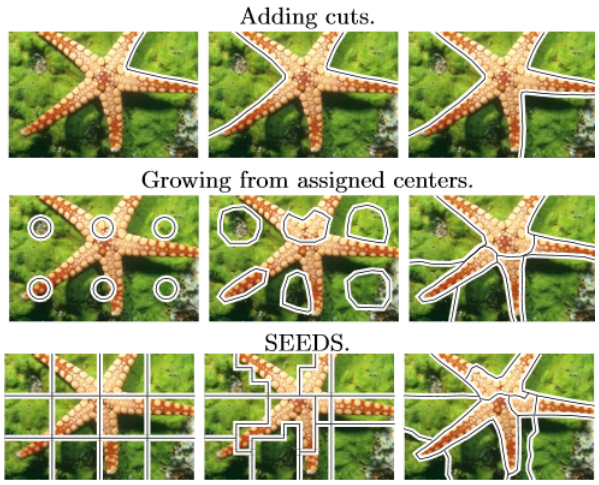
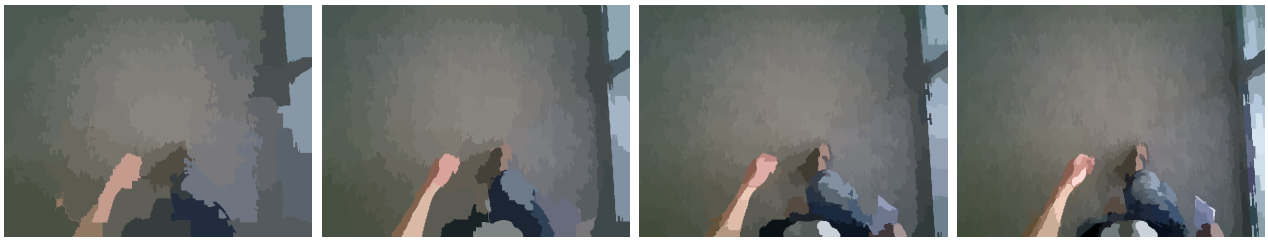


Figura 2.4: Comparación Seeds con otros métodos. El primer método busca bordes con los que separar superpixels. El segundo realiza un crecimiento de regiones desde unos centros asignados.Figura obtenida del artículo [11].



(a) 20 Superpixels (b) 50 Superpixels (c) 100 Superpixels (d) 200 Superpixels

Figura 2.5: Visualización de un ejemplo de segmentación *Seed* con distinto número de superpixels.

Algunos ejemplos (véase Figura 2.5) con las secuencias con las que trabajamos y un distinto numero de superpixels. Veremos que al igual que en el caso de los superpixel, si el número de superpixels es muy pequeño la acoplación de colores es muy alta. Sin embargo, a partir de 100 la imagen queda segmentada de manera que los bordes están mejor definidos.

2.2.2. Segmentación de pixels pertenecientes a piel

La segmentación en pixels de piel que queremos realizar se basa tanto en el color de los pixels como en la profundidad a la que se encuentran estos. Se propone realizar un barrido sobre toda la imagen desechando pixels que no cumplan una cierta restricción.

Para el color se utilizó un filtro detallado en la Fórmula 2.1.

$$\begin{aligned}(R > 95)\&(G > 40)\&(B > 20)\& \\ ((MAX(R,G,B) - MIN(R,G,B)) > 15) \\ \&(|R - G| > 15)\&(R > G)\&(R > B).\end{aligned} \tag{2.1}$$

Este filtro para el color de piel se basa en los resultados del trabajo mostrado en [13]. A esto se le añadió la limitación de profundidad. Después de una serie de pruebas de calibración, se vio que el valor 110 de profundidad se correspondía bien con la distancia con la que suelen estar los brazos. Aproximando este valor a la realidad y basándose en algunos resultados se podría aproximar a una distancia de 100 cm. En la Figura 2.7, donde los pixeles en blanco son pixeles aceptados por el filtro de piel pero no por el filtro de profundidad, vemos que el número de pixels aceptados incorrectamente aumenta sin el filtro de profundidad. Sin embargo, aún con este paso, la segmentación no es perfecta ya que algunas mesas o libros que están a una distancia parecida a la de los brazos y tienen un color parecido son reconocidos como piel. En la Figura 2.6 comprobamos como resulta la imagen tras el filtro tanto de color como de profundidad.

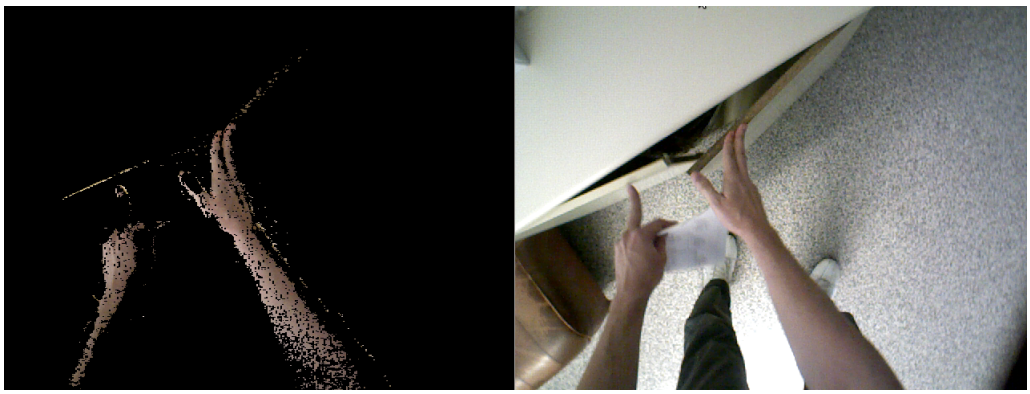
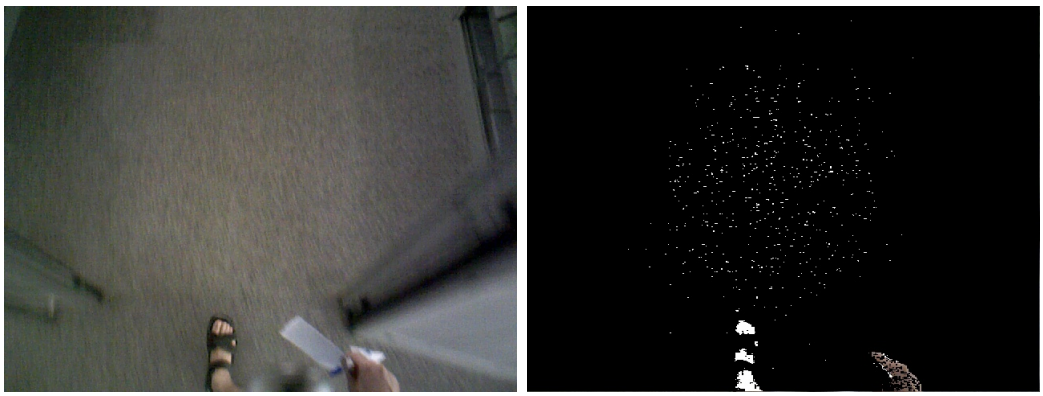


Figura 2.6: Visualización de una imagen tras el filtrado de color y profundidad.

2.2.3. Segmentación según restricciones geométricas: Planos

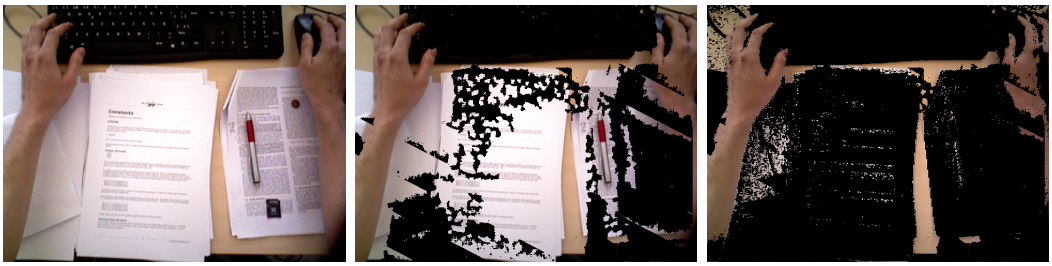
Con este tipo de segmentación buscábamos evitar o desechar falsos positivos de puertas, mesas, suelos... asi como poder utilizar la información para describir la imagen. Para ello haremos uso de la biblioteca PCL [14] que trabaja con nube de puntos y tiene funciones que facilitan la búsqueda y segmentación de planos. Usando la clase *SACSegmentation*[4] con un modelo de plano y el método iterativo *Ransac*, buscamos planos en las imágenes. Este método busca planos en la imagen calculándolos en base a ciertos puntos de la nube y luego comprueba qué cantidad de puntos de la nube entran dentro del plano calculado. En la Figura 2.8 se puede observar un ejemplo de segmentación del plano dominante.

[4] `http://docs.pointclouds.org/trunk/a01380.html`



(a) Imagen Original (b) Visualización del filtro de profundidad.

Figura 2.7: Visualización del la reducción de ruido mediante el uso del filtro de profundidad. Los pixels blancos son aquellos que el filtro de color admite pero el filtro de profundidad rechaza.



(a) Imagen Original (b) Segmentación Plano (c) Segmentación Plano y Piel

Figura 2.8: Segmentación del plano dominante de fondo.

El algoritmo permite modificar la precisión mediante la distancia de *threshold* mínima, el número de iteraciones, el método de búsqueda... Sin embargo, cuanta mayor precisión se busca mayor será el coste de ejecución del algoritmo hasta llegar a un punto donde el aumento de precisión no sea visible.

2.2.4. Comparativa segmentación

En esta sección vamos a realizar una comparativa entre cada una de las posibilidades de las distintas segmentaciones posibles, así como una entre distintas segmentaciones.

Comparativa Superpixels

Como vemos en la Figura 2.9 al comparar los algoritmos de segmentación de superpixels nos encontramos que, como bien se menciona en su publicación[11], el algoritmo *Seed* respeta mejor los bordes originales de la imagen. Sin embargo, en cuestión de coste

(a) Imagen Seeds (b) Imagen SLIC

Figura 2.9: Visualización de la comparación entre la segmentacion Seeds(a) y SLIC(b).

SLIC es mucho más rápido que *Seed*. Por lo tanto se nos plantea el problema de calidad contra tiempo. Como en un principio no estamos restringidos por la limitación de la ejecución en tiempo mínimo, tomaremos la calidad. Por tanto, utilizaremos en las pruebas la segmentación *Seed* para calcular los superpixels.

Conclusión Segmentación de Piel

(a) Imagen Original (b) Imagen Piel y Seeds (c) Imagen Piel y SLIC

Figura 2.10: Comparativa piel con SuperPixel.

En la Figura 2.10 vemos que los pixels de piel son segmentados en su mayoría, pero sin los superpixels tiene mayor forma de ruido, lo que podría empeorar siguientes etapas de extracción de descriptores. Cuando utilizamos superpixels, la segmentación es más limpia, aunque puede darse el caso de que segmente pixels que no pertenezcan como son elementos de la escena cuyo color y profundidad sean parecidos.

Conclusión Segmentación de planos

En el caso de la segmentación de los planos nos encontramos con la situación de que, siendo su coste de ejecución elevado, puede no merecer la pena combinarlo con la

(a) Imagen Original (b) Imagen segmentación Plano,Piel y Seeds (c) Imagen segmentación Plano,Piel y SLIC

Figura 2.11: Visualización de la segmentaciíon del plano dominante y las diferencias en entre *SLIC* y *Seeds*.

segmentación de piel.Se han realizado varios experimentos para combinar las segmentación de planos junto con las otras dos segmentaciones, como en la Figura 2.8c.En ellos se puede observar qué pixels pertenecientes a piel sean tomados como pixels de un plano y desechados. Por lo tanto para las siguiente fases de reconocimiento se prescindió de la segmentación de planos ya que no solo es ineficiente sino que también empeora el resultado.

Conclusión Segmentación

Viendo los resultados obtenidos se ha decidido hacer uso de la segmentación en superpixels *Seeds* junto con la segmentación de piel dado que son los que aparentemente mejor segmentan los pixels de piel. La segmentación de planos no está incluida dado que aumenta el coste y empeora la precisión.

2.3. Descripción de una imagen RGB-D

En cualquier tarea de reconocimiento, es difícil trabajar con los datos en crudo de la imagen, usando los valores individuales de cada pixel. En general hay que buscar descriptores de la imagen que compriman la información que pueda resultar más discriminante para la tarea que se quiere realizar y que sean más descriptivos que la información de cada pixel por separado. El rango y tipo de descriptores de imágenes que encontramos en la literatura es muy amplio, puede variar desde estadísticas sencillas como el número de pixels de un color hasta otras mucho más complejas basadas en los gradientes de la imagen.

Se puede hacer una división del tipo de descriptores según sobre qué parte de la imagen se calculan: globales y locales. Los descriptores de tipo global son aquellos que se calculan sobre el conjunto total de la imagen. Los descriptores de tipo local son aquellos que se

calculan sobre zonas de interés o partes de la imagen.

A continuación se describen los descriptores estudiados para nuestro problema, según su tipo.

2.3.1. Descriptores globales

En este apartado se discutirá qué descriptores globales se han implementado.

Histograma de piel

Ya que muchas de las actividades que realiza una persona están relacionadas con los objetos que manipula, se quería buscar un descriptor basado en los pixels de piel que aparecen en la imagen. Para analizar como están distribuidos los pixels que corresponden con piel en la imagen, calculamos dos histogramas de pixels de piel para cada eje como se ve en la imagen 2.12c y 2.12b. A partir de este análisis de la distribución, se diseño un descriptor global obtenido de manera similar, pero construyendo un único histograma, con un valor en el histograma para cada celda en vez de separar filas y columnas. Se dividió la imagen en una cuadrícula de un tamaño modificable y se calculó un histograma de cada celda.

(a) Imagen segmentada con cuadricula (b) Histograma en el eje Vertical. (c) Histograma en el eje Horizontal.

Figura 2.12: Histogramas de los ejes Horizontal(b) y Vertical(c) donde cada valor representa el número de pixels de piel en esa fila o columna. Visualización de la cuadrícula(a) con la que se calcula el histograma de piel.

De esta forma ya tenemos un descriptor global de la imagen basado en los pixels de piel segmentados, que además es rápido de calcular y fácil de manejar. Lo siguiente que se buscaba era decidir qué tamaño de cuadrícula sería el aconsejable. Al ser una cuadrícula, y por no complicar en exceso el descriptor, se decidió que ambos ejes tuvieran el mismo tamaño. Dado que no nos interesaba un descriptor cuyo tamaño fuera excesivamente grande experimentalmente se decidió probar con tamaños de cuadricula con 5,10 y 15. El tamaño del descriptor es el cuadrado de los tamaños con lo cual tendríamos posibles descriptores de tamaños 25,100 y 225. Se decidió finalmente utilizar el tamaño de 10 ya que nos daba la suficiente información sin excederse en tamaño.

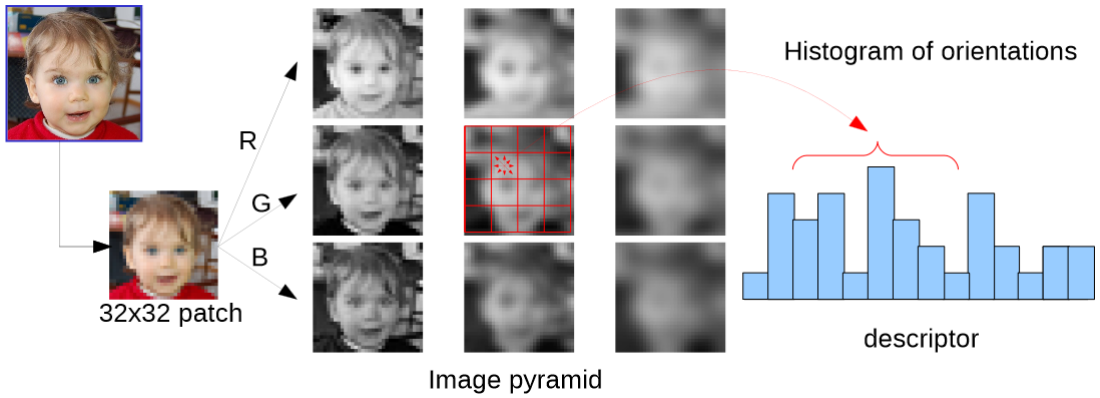


Figura 2.13: Explicación del funcionamiento del descriptor *GIST* obtenida del trabajo [15].

GIST

El descriptor de imagen *GIST* fue propuesto en [16]. Este descriptor es utilizado en trabajos como [15] donde muestra buenos resultados en materia de reconocimiento de elementos en un conjunto de imágenes. En este trabajo se deja disponible una implementación del descriptor en C++ que es la que utilizaremos en este proyecto.

El funcionamiento de este descriptor es el siguiente. Inicialmente se reduce la imagen debido a que es un descriptor bastante robusto frente al redimensionamiento. Luego se separan los posibles canales de color y se trabaja sobre cada uno de ellos. Se divide la imagen en una cuadrícula de dimensiones definidas, dentro de la cual se calculan las orientaciones y el histograma de estas orientaciones por celda. Estos histogramas se concatenan formando el descriptor total *GIST*. El tamaño del *GIST* viene fijado por el tamaño que fijemos en la cuadrícula, no por el tamaño de la imagen. La Figura 2.13 muestra un resumen de como funciona.

El descriptor *GIST* es calculado sobre la imagen total, sin ningún tipo de segmentación. En una serie de pruebas que se realizaron calculándolo imagen segmentada en superpixels, la fiabilidad del descriptor se redujo a la mitad. En nuestro trabajo, utilizaremos el GIST calculado solo sobre la imagen en nivel de gris, ya que ocupa un tercio que el descriptor de la imagen a color y no aportaba grandes diferencias.

Otros descriptores estudiados

También se estudiaron otros descriptores de imagen típicos en tareas de reconocimiento que estaban disponibles en la biblioteca *OpenCV* [21]: *Brief* [17] y *Orb* [18]. Sin embargo, tras realizar una serie de cálculos con varias imágenes vimos que el tamaño de estos descriptores (5000 y 16000 aproximadamente) era demasiado alto y rompía con nuestra idea de descriptores compactos, ademas del problema de memoria que pueda ocasionar

un conjunto alto de imágenes con descriptores de ese tamaño.

2.3.2. Descriptores Locales

Un descriptor local es aquel cuyos cálculos se basan en una o varias regiones de la imagen. Por lo tanto, al revés que los descriptores globales, no tiene en cuenta la imagen en conjunto sino una serie de regiones de interés. La búsqueda de estas regiones depende del tipo de descriptor elegido. En nuestro caso, nos centramos en calcular descriptores basados en el *Bounding Box* de los brazos/manos explicado a continuación.

Bounding Box

El termino *Bounding Box* proviene del ingles *Minimal Bounding Rectangle*[19], que significa rectángulo delimitador mínimo. En nuestro caso sera el rectángulo mínimo que contenga la mano o el brazo. Para ello haremos uso de la imagen segmentada tanto en superpixels como en piel.

Para calcular las *Bounding Box* partimos de dos histogramas previamente calculados de la cantidad de pixels en cada uno de los ejes (vease Figura 2.12). A partir de estos histogramas buscamos 'zonas' que parezcan de interés, porque contienen muchos pixels de piel. Estas zonas podrán ser luego los límites en un eje de un *Bounding Box*. Para aceptar unos límites tienen que contener más de un mínimo de pixels de piel y además no haber huecos. Esto último se mira con el hecho de que en una fila/columna no haya ningún pixel, es decir el valor del histograma sea 0.

Una vez calculadas las posibles zonas, el número de posibles *Bounding Box* puede ser mayor del necesario. Para ello revisamos en cada candidato la cantidad de pixels aceptados que hay en la celda de la cuadrícula correspondiente a su centro. Si esta cantidad es superior a un umbral, se comprueba que los límites son correctos y se trata como un *Bounding Box*. Esta comprobación es necesaria debido a que no siempre se calculan los límites correctamente (puede haber confusión entre dos candidatos solapados en un eje). En la figura 2.14 podemos observar un ejemplo de *Bounding Box* donde se ha dibujado un cuadrado de color azul.

Una vez calculado el *Bounding Box* se pueden analizar una serie de propiedades de dicho rectangulo. Entre las opciones estudiadas están: los momentos[5] basados en la imagen binaria de Piel/No-piel, atributos estadísticos de media, moda sobre cada canal de color, etc. Nos decantamos por dos cálculos sencillos y que podrían añadir información interesante. Primero el cálculo de los vectores propios y luego el cálculo del ratio de completitud del *Bounding Box*.

Para calcular los vectores propios realizamos una *Principant Component Analysis (PCA)*[20] sobre todos los pixels de piel pertenecientes al *Bounding Box*. De ahí obtenemos los vectores propios y con ello el vector orientación, aunque este último solo sirva

[5] `http://en.wikipedia.org/wiki/Moment_(mathematics)`

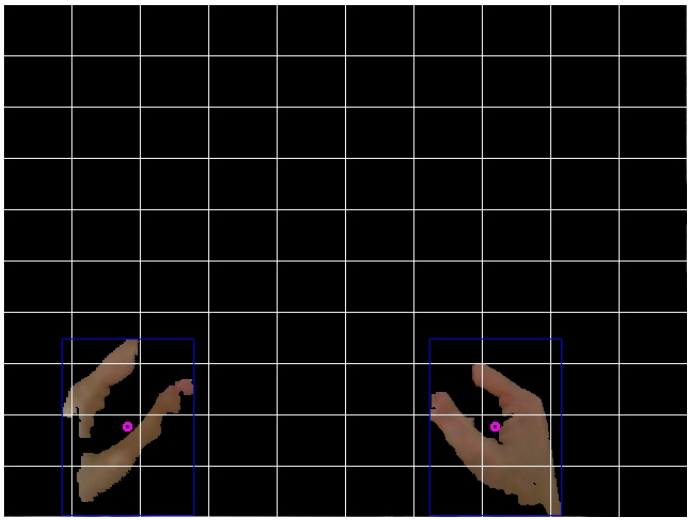


Figura 2.14: *Bounding Box* en las manos

como visualización. Los vectores propios de un conjunto de datos indican como de dispersos están los datos en el eje correspondiente. Cuanto mayor es, más dispersos están los datos. En nuestro caso, como utilizamos las posiciones de los pixels nos va a indicar la forma y orientación del contenido del *Bounding Box*.

Para el cálculo del ratio de completitud, simplemente calcularemos cuantos pixels son de piel con respecto al total de pixels del *Bounding Box*. Esto nos dará una idea de como de lleno se encuentra.

Tambien hemos analizado los posibles inconvenientes de este descriptor. El primero es que no aparecerán siempre en la imagen, al menos ambos. Con lo cual, cuando el número es inferior a 2 se usan valores estándar de vacío para los *Bounding Box* que se necesite. El segundo es que la segmentación de piel no es perfecta y puede generar zonas separadas o zonas que confunde con piel. Por lo tanto, cuando el número de *Bounding Box* es superior a 2, se ordenan por tamaño y se utilizan los dos mayores.

Capítulo 3
Reconocimiento de Acciones

Este capítulo describe el desarrollo del clasificador de acciones de este proyecto. Una vez obtenidos los descriptores tendremos que analizarlos y clasificarlos. En general, un sistema de clasificación necesita dos partes: Entrenamiento y Reconocimiento. En el entrenamiento utilizaremos una serie de secuencias compuestas por un conjunto de imágenes junto con su etiqueta correspondiente, denominado de la siguiente manera:

$$\text{Conjunto de entrenamiento} = (x_1, l_1), ..., (x_n, l_n), \tag{3.1}$$

donde x_i es el dato i y l_i es la etiqueta correspondiente a ese dato. Y para el reconocimiento utilizaremos imagenes x_* a las que buscaremos asignarles una etiqueta l_*. De todos los clasificadores existentes nos centramos en dos de los más típicos y utilizados en la literatura: *Nearest neighbor*(Vecino más próximo) y *SVM* (*Support vector machine*).

3.1. Clasificadores utilizados

3.1.1. *Nearest Neighbor*(Vecino más próximo) con Clusterización

El primero de los clasificadores que utilizamos fue el *Nearest Neighbor* debido a que es el que más simple funcionamiento y tiene ademas de un buen potencial. El algoritmo *Nearest Neighbor* es un algoritmo que dado una conjunto de datos y un dato de entrada devuelve la etiqueta del dato del conjunto al que más próximo se encuentra mediante la siguiente formula:

$$\begin{aligned} x_{NN} &= \arg\min_{x_i} D(x_*, x_i) \\ l_* &= l_{NN}, \end{aligned} \tag{3.2}$$

donde D es la función de calculo de distancia. En nuestro caso, el conjunto normal de entrenamiento son varias secuencias con un número grande de imágenes, lo que hace que el coste lineal de este algoritmo sea prohibitivo. Para poder obtener un coste computacional más aceptable se planteó utilizar la variante del método que añade la *clusterización* del conjunto de datos. De esta forma el coste computacional es lineal en el número de clusters.

Un algoritmo de agrupamiento (en inglés, *clustering*) es un procedimiento de agrupación de una serie de datos de acuerdo con un criterio. Esos criterios son por lo general distancia o similitud. La cercanía se define en términos de una determinada función de distancia. En este proyecto utilizaremos el método de *clustering kmeans*[22]. Este método tiene como parámetros la función de distancia, en nuestro caso la Euclídea, y el número de *clusters* deseado. El número de clusters a escoger es relativamente importante de elegir debido a que un número muy pequeño de *clusters* puede mezclar las clases y un número muy grande nos presenta el mismo problema de coste computacional que sin clusterización. El funcionamiento del algoritmo de clusterización *kmeans* es el siguiente: Dado un conjunto de datos 3.1 el algoritmo intenta realizar un agrupamiento en k *clusters* ($k << n$) $S = S_1, S_2, ..., S_k$ donde minimice la suma de cuadrados dentro de cada *cluster*:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x_j \in S_i} ||x_j - \mu_i||^2, \tag{3.3}$$

donde μ_i es la media de los puntos en S_i. Luego para cada cluster se calcula que etiqueta se le asigna mediante la búsqueda de la que mayor incidencia tenga.

Para la parte de implementación del módulo de entrenamiento, utilizaremos la función *kmeans*[1] que implementa este método en la biblioteca *OpenCV*[21]. Para la parte de implementación del módulo de reconocimiento se implementó una función propia de *Nearest Neighbor*, basada en el método antes explicado, que dado un dato y la matriz de centros, calcula el *cluster* más cercano a ese dato. Más información sobre el *Nearest Neighbor* en el Anexo B.

3.1.2. Clasificador *SVM (Support Vector Machine)*

El otro clasificador que estudiamos fue el *SVM (Support Vector Machine)*[23]. El *SVM* funciona de la siguiente manera: Dado un conjunto de entrenamiento $D = \{(x_i, l_i)|x_i \in \Re^p, l_i \in -1, 1\}_{i=1}^{n}$, queremos encontrar el hiperplano de mayor margen que divida los puntos de ambas clases. Todo hiperplano puede ser escrito como un conjunto de puntos x que satisfacen:

$$\mathbf{w} \times x - b = 0, \tag{3.4}$$

donde $\times$ denota el producto escalar y $\mathbf{w}$ el vector normal, no siempre normalizado, al hiperplano. Por lo tanto, para calcular el mejor hiperplano necesitaremos minimizar la siguiente formula:

$$\begin{aligned} &\arg\min_{(\mathbf{w},b)} \frac{1}{2}||\mathbf{W}||^2 \\ &\text{sujeto a(Para cualquier } i = 1, ..., n) \\ &l_i(\mathbf{w} \times x_i - b) \geq 1. \end{aligned} \tag{3.5}$$

[1] `http://docs.opencv.org/modules/core/doc/clustering.html#kmeans`

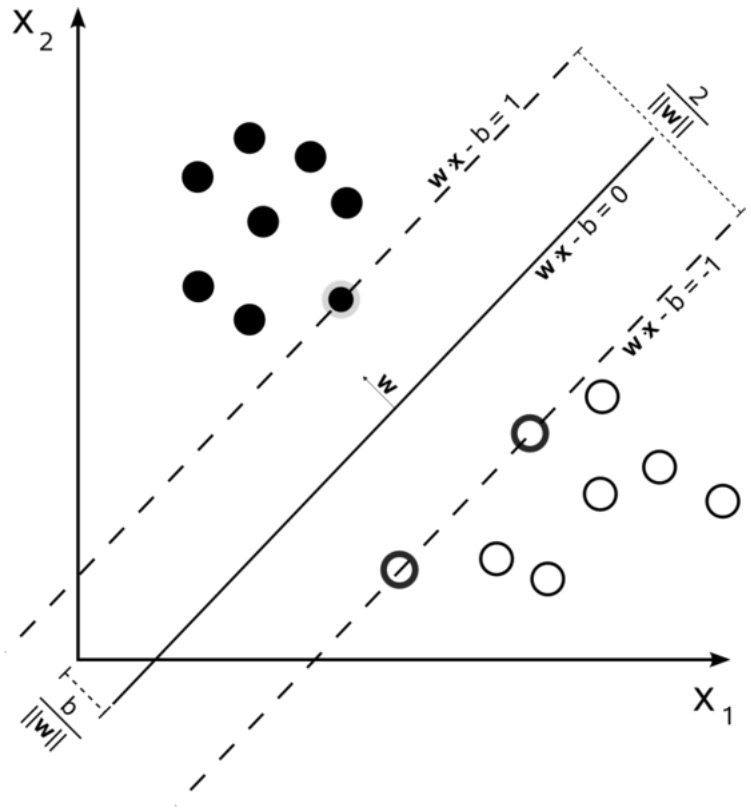


Figura 3.1: Ejemplo de clasificación SVM. El hiperplano calculado divide el espacio en dos clases separando lo máximo posible los puntos frontera.

En la Figura 3.1, tenemos un ejemplo de clasificación con *SVM*. Esta explicación sirve para clasificación lineal de clases, sin embargo, cuando necesitamos realizar una clasificación no lineal cambiamos el producto escalar por una función *kernel*. Esta función proyecta la información a un espacio de características de mayor dimensión donde poder realizar mejor la clasificación. Más información sobre los *kernel* en el Anexo B. Dada la naturaleza de la clasificación de *SVM* podríamos decir que no busca las mejores zonas sino las mejores fronteras. Por lo tanto no necesita tener presente todos los puntos, sino que son los puntos frontera los que modifican la clasificación. Para la parte de reconocimiento calcula la zona en la que cae el vector con respecto a los hiperplanos calculados.

Utilizaremos la versión implementada en *OpenCV* que esta basada en la versón *libSVM* [24]. Como se ha visto es un clasificador de dos clases, sin embargo utilizaremos una implementación de *one-versus-all*. Como parámetros del clasificador tenemos varios: primero esta el tipo de *SVM*, el *kernel* utilizado para calcular los hiperplanos, parámetros correspondientes tanto al tipo como al *kernel*, etc. El tipo de *SVM* que utilizaremos siempre sera el *C_SVC* que da mejores resultados en la clasificación que el *NU_SVC*. Los diferentes tipos de *kernel* que utilizaremos nos ayudaran, además de para elegir el mejor, para entender cómo están distribuidos nuestros datos. Más información sobre *SVM* en el

Anexo B.

3.2. Normalización

Los descriptores tienen distintos rangos de valores y eso dificulta un correcto entrenamiento de los sistemas de clasificación. Por lo tanto se han normalizado los descriptores que no lo estaban inicialmente: en el caso del Histograma de piel se utilizó una normalización por ratio de celda. Por tanto, el valor del histograma se divide por el número máximo de pixels por celda. En el caso del descriptor *GIST* ya está normalizado por defecto. Mientras que los posibles valores de las *Bounding Box*, el vector viene normalizado en su cálculo y la ratio es un valor normalizado en sí.

3.3. *Cross-Validation*

Como hemos visto en el apartado 3.1.2, los clasificadores tienen ciertos parámetros que deberemos calcular. Para ello realizaremos una validación cruzada, o *'cross-validation'*, con el conjunto de datos que tenemos disponibles de forma que el conjunto de test sea siempre distinto al de entrenamiento. Dado que el conjunto de datos está compuesto por secuencias de imágenes, existe una correlacción alta entre imágenes de una secuencia tanto a nivel temporal (imágenes cercanas temporalmente tienen mayor grado de semejanza) como espacial (las mismas acciones dentro de una secuencia tienen una alta similitud). Para evitar esta correlación entre imágenes de una misma secuencia utilizaremos las secuencias como bloques mínimos. Utilizaremos el método de validación cruzada dejando uno fuera, o *Leave-one-out cross-validation*, en el que utilizamos una secuencia de test y el resto de entreno como vemos en la Figura 3.2. Realizamos esto para todas las secuencias y luego sacamos la media de los resultados. Por lo tanto esta será la forma de evaluar los experimentos en el siguiente apartado.

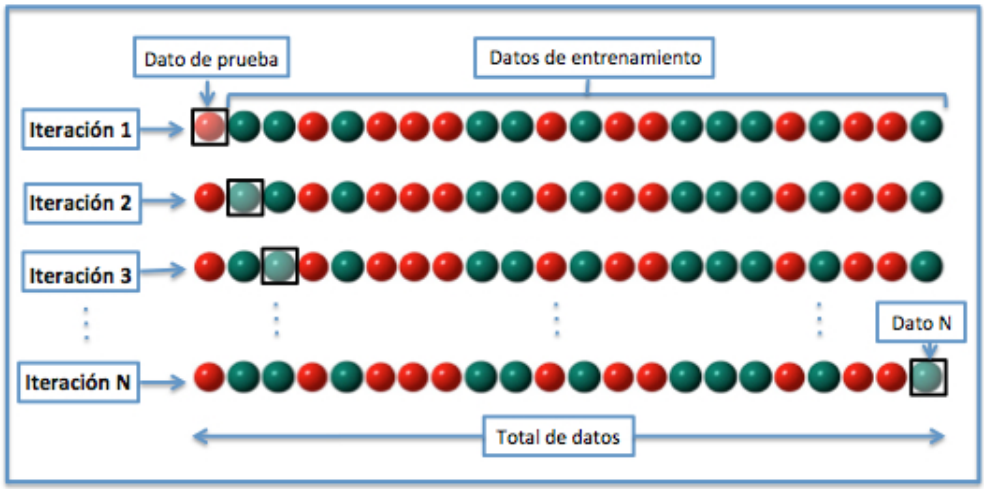


Figura 3.2: Validación cruzada dejando uno fuera ($LOOCV$)

Capítulo 4

Experimentos sobre reconocimiento de acciones

Una vez hemos presentado las variantes tanto de descriptores como de clasificadores vamos a realizar unos experimentos sobre ellos. A lo largo de todo el proyecto se han ido realizando pequeños experimentos en cada una de las fases para comprobar resultados, muchos de los cuales se pueden encontrar en el Anexo C. Sin embargo, aquí analizaremos los experimentos más importantes siendo éstos los últimos realizados. Primero hablaremos de la configuración de los experimentos y luego los resultados obtenidos.

4.1. Configuración de los experimentos

En este capítulo hablaremos de qué datos utilizaremos para los experimentos y de cómo los clasificaremos.

4.1.1. Datos Utilizados

El data set utilizado ha sido adquirido por miembros del grupo de Robótica, Percepción y Tiempo Real (ROPERT)[1], de la universidad de zaragoza[2]. Este data set fue presentado en el *ICCV 2013*[3]. Las distintas secuencias utilizadas fueron adquiridas por distintos usuarios y/o en distintos edificios del campus Río Ebro. Las secuencias fueron grabadas con la cámara *Asus Xtion Pro* colocada en un casco y enfocada a la zona común de trabajo manual, es decir, enfocada hacia abajo justo delante del cuerpo. Lógicamente el movimiento de la cabeza influye en la zona de enfoque pero como se vio en [3] la mirada suele estar enfocada en la zona de acción. Para poder realizar los experimentos necesitábamos que las secuencias de datos tuvieran un *ground truth* o datos de verificación. Para ello en primer lugar se diseño el conjunto de etiquetas o actividades a reconocer, y se realizó un procesado/clasificación manual de las secuencias para asignar los valores de verificación/referencia.

[1] `http://robots.unizar.es/html/home.php`
[2] `http://www.unizar.es`
[3] `http://www.iccv2013.org/`

Se han utilizado 5 secuencias en total y, como se ha explicado en el apartado 3.3, en cada experimento utilizamos una secuencia de testeo y las otras cuatro de entrenamiento. Para realizar los experimentos se han limitado el número de imágenes de entrenamiento, debido tanto a las posibles limitaciónes técnicas de los ordenadores como a la búsqueda de un entrenamiento más equilibrado. Por lo tanto, para el primer nivel se acepta un total de 6000 ejemplos (un ejemplo quiere decir un 'frame' de la secuencia) por etiqueta. En siguientes niveles, acepta un máximo de 1200 ejemplos por etiqueta, habiendo algunas etiquetas que lo saturan y otras de las cuales hay pocos ejemplos y no llegan a este límite, sino que se quedan en unos pocos cientos. Para elegir las imágenes de entreno se coge las cuatro secuencias y se van añadiendo un ejemplo sacado de cada secuencia hasta completar los límites o hasta acabar con las secuencias.

4.1.2. Acciones a reconocer

La precisión o definición y cantidad de las etiquetas que definamos influirán en el nivel de precisión del reconocedor. Inicialmente partimos de un total de 27 etiquetas definidas en el nivel semántico más alto, más detallado (nivel 3 en la Tabla 4.3 junto con el nivel 2 de no-manipulacion en la Tabla 4.2).

Como el reconocimiento muy detallado de acciones puede resultar muy complicado con poco ejemplos de cada una (los descriptores solo capturan la información 2D, no 3D) se decidió por una división de acciones por niveles. Definimos los niveles descritos en las Tablas 4.1, 4.2 y 4.3.

Nivel 1	
Nombre	Descripción
Manipulación	Acciones que conllevan manipulación manual
No Manipulación	Acciones que no conllevan manipulación manual

Cuadro 4.1: Nivel 1 de división de etiquetas.

La división se realizó en base a las distintas posibilidades de acción. La primera división está basada en si en la imagen hay o no acción de manipulación. En el caso de que haya manipulación, qué tipo de manipulación hay(con una mano,dos manos o con un objeto) y en el caso de que no haya manipulación qúe tipo de acción podría definirse. Esta división nos dará un alto porcentaje de aciertos en el primer nivel que se reducirá en el segundo nivel.El tercer nivel es meramente orientativo debido que no buscamos tanta precisión de etiquetas. En el apartado de resultados veremos cómo de bien funciona cada unos de los niveles.

4.2. Resultados de clasificación

En este capítulo hablaremos sobre los experimentos más importantes realizados para validar y evaluar el rendimiento del sistema propuesto. Para estos experimentos hemos

Nivel 2 (*Manipulación*)		**Nivel 2 (*No Manipulación*)**	
Nombre	Descripción	Nombre	Descripción
DosManos	Acciones que conllevan el uso de dos manos	*Andar*	Movimiento del usuario
UnaMano	Acciones que conllevan el uso de una mano	*Escaleras*	Subir/bajar las escaleras
Interacción	Acciones que conllevan la interacción(coger/dejar) con objectos	*Parado*	El usuario está estático
Desconocido	Acciones no incluidas en el resto de etiquetas	*Sentado*	El usuario está sentado
		Pantalla	Mirar/Leer en una pantalla
		Poster	Mirar/Leer un poster/cartel
		Hablar	Hablar con alguien

Cuadro 4.2: Nivel 2 de división de etiquetas.

Nivel 3(*DosManos*)	**Nivel 3(*UnaMano*)**	**Nivel 3(*Interacción*)**
Escribir en el Teclado	Abrir/Cerrar Puerta	Beber
Utilizar el ratón	Abrir/Cerrar Ventana	Comer
Leer un papel	Abrir/Cerrar Nevera	Coger Objeto
Leer un libro	Abrir/Cerrar Microondas	Dejar Objecto
Escribir a mano en un papel	Abrir/Cerrar Armario	Utilizar maquina de cafe
	Escribir en la pizarra	Utilizar Maquina expendedora
	Dar la mano/Saludar a alguien	Hablar por teléfono

Cuadro 4.3: Nivel 3 de división de etiquetas.

usado las posibles configuraciones de los distintos clasificadores y los distintos descriptores. Las posibilidades en los descriptores eran las vistas en el capitulo 2.3 : *GIST*, histograma de piel y *Bounding Box* incluyendo los descriptores separados que se calcularon. Para las posibilidades de los clasificadores teníamos el *Nearest Neighbor* y el *SVM*, utilizando los posibles *kernel* de lineal,*RBF*, sigmoidal y polinómico. El uso de distintos *kernel* nos ayudará además a conocer como se comportan nuestros datos.

En una serie de pruebas intermedias vimos que el comportamiento del *kernel* sigmoidal y polinómico era peor o igual que el del *RBF*, con lo que reducimos el número de kernels a lineal y RBF. Los resultados de estas pruebas se encuentran en el Anexo C.

Por lo tanto, para los experimentos que explicaremos a continuación utilizaremos como combinaciones de descriptores: Histograma de piel, *GIST*, *GIST*+Histograma de piel, *GIST*+Histograma de piel+Total de descriptores de las *Bounding Box*, *GIST*+Total de

descriptores de las *Bounding Box*, *GIST*+Vectores propios de las *Bounding Box* e Histograma de piel+Ratio completitud de las *Bounding Box*. Como clasificadores utilizaremos nuestra versión de *Nearest Neighbor* y el *SVM* con dos posibles *kernels*: *RBF* y lineal.

4.2.1. Resultados experimentos con la misma configuración

El experimento de este apartado estudia los resultados cuando se usa la misma configuración en los tres niveles es decir, si en el primer nivel se utiliza la configuración *GIST* con *SVM* de kernel lineal, en los siguientes niveles se utiliza esa misma configuración.

En las Figuras 4.1, 4.2 y 4.3 se muestran los resultados de probar todos los clasificadores con distintas combinaciones de descriptores. Estos resultados son la media de los resultados de 5 experimentos: cada uno usando como test cada una de las 5 secuencias, y las otras 4 de entrenamiento. La configuración mostrada es la misma utilizada en todos los niveles de reconocimiento. Los resultados de cada secuencia se encuentran en el Anexo C.

Primer Nivel

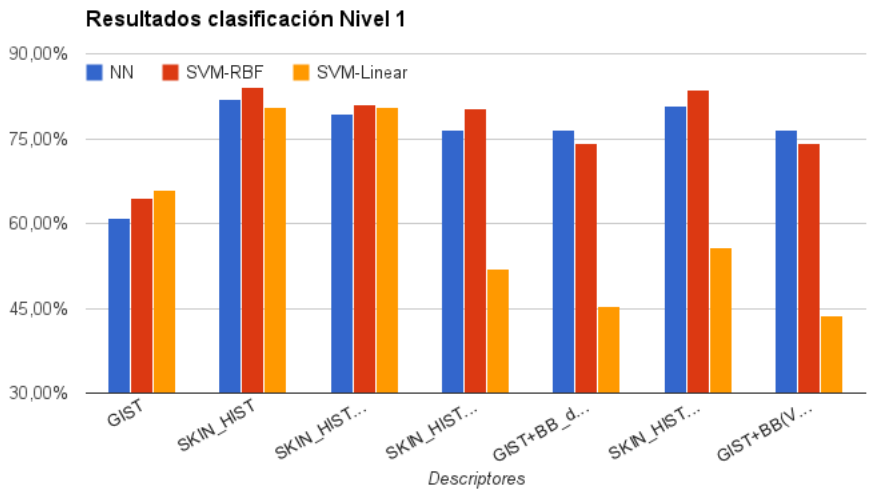


Figura 4.1: Gráfica que muestra el % de aciertos medio en el primer nivel de cada una de las posibilidades. En el Eje X tenemos los distintos descriptores, cada barra representa cada uno de los clasificadores y en el Eje Y tenemos los porcentajes de aciertos para el descriptor y el clasificador elegido.

En la Figura 4.1, que nos muestra el resultado del experimento para el primer nivel, vemos que en materia de descriptores tanto el Histograma de piel solo o combinandolo con *GIST* o el descriptor de ratio de las *Bounding Box* funcionan de manera parecida, rondando los 84 % de aciertos. Para los clasificadores vemos que el *SVM* con *kernel RBF* tiene un ligero aumento de aciertos frente al *Nearest Neighbor*. Además podemos observar que el *Bounding Box* junto con el *kernel* lineal del *SVM* reduce los aciertos entre un 20 y un 30 %.

Matriz de confusión

Para analizar en más detalle el funcionamiento de un clasificador se calcula una matriz, llamada matriz de confusión, en la que se indica para cada etiqueta según el *ground truth* cuántas veces se ha reconocido otra etiqueta. Por ejemplo, siendo $e_{i,j}$ un elemento de la matriz de confusión nos indica el porcentaje de veces que se ha reconocido la etiqueta i siendo la etiqueta del *ground truth* j. Esto es muy útil a la hora de comprobar como funciona nuestro clasificador. En la Tabla 4.4 tenemos la matriz de confusión para el primer nivel en el mejor caso.

	Manipulación(2532)	*No-Manipulación*(2538)
Manipulación	0.72	0.09
No-Manipulación	0.28	0.91

Cuadro 4.4: Ejemplo de matriz de confusión para el nivel inicial. La fila representa la etiqueta reconocida y la columna el *ground truth*. Los números entre paréntesis indican el número de imágenes de test que teníamos de cada etiqueta.

Nivel Manipulación

En la Figura 4.2 se encuentran la media de los resultados del nivel 2 de manipulación. Vemos que los resultados se reducen bastante. Tanto estos porcentajes como los de no manipulación son relativos a los aciertos del primer nivel, es decir, no es el porcentaje total de apariciones de la etiqueta frente al número de aciertos. Esta reducción es comprensible debido a que buscamos una definición algo más concreta y las posibles confusiones aumentan. Por lo tanto tenemos que en esta división *GIST* funciona el que mejor con los tres posibles clasificadores.

Nivel No Manipulación

En la Figura 4.3 se encuentran la media de los resultados del nivel 2 de no-manipulación. Vemos que el nivel de acierto es parecido al del primer nivel. La mayoría de las opciones funcionan a un nivel parecido de aciertos, pero como veremos en el análisis maś detallado a continuación, este resultado era engañoso y no resulta tan bien como en el primer nivel.

4.2.2. Experimentos fijando primer nivel de clasificación

Los resultados obtenidos en las pruebas anteriores para los niveles de manipulación y no-manipulación están influenciados por el nivel de división inicial. Por lo tanto realizaremos una prueba en la que fijaremos el primer nivel de etiquetas con el histograma de piel como descriptor y el *SVM* con el *kernel RBF* como clasificador. De esta manera podremos realizar un análisis más exhaustivo sobre las diferencias. Como podemos ver en los apartados siguientes, los porcentajes de acierto han variado lo que demuestra que

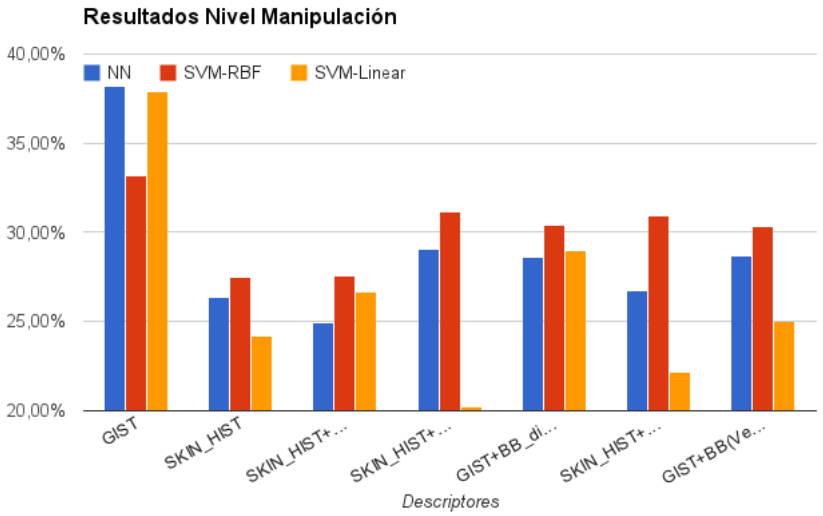


Figura 4.2: Gráfica que muestra el % de aciertos medio en el nivel de Manipulación de cada una de las combinaciones de descriptores y clasificadores utilizados.

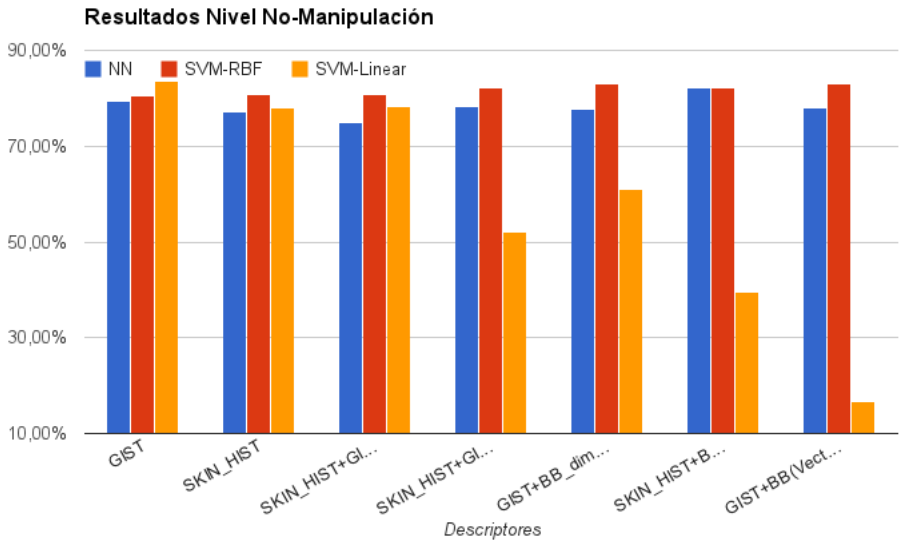


Figura 4.3: Gráfica que muestra el % de aciertos medio en el nivel de No Manipulación de cada una de las combinaciones de descriptores y clasificadores utilizados.

los resultados del primer nivel influyen en el tipo de imágenes que pasan a siguientes clasificaciones.

Nivel Manipulación

Vemos en la Figura 4.4 que para la división de manipulación la mayoría de las opciones dan resultados bastante reducidos, estando estos en un rango parecido a resultados ante-

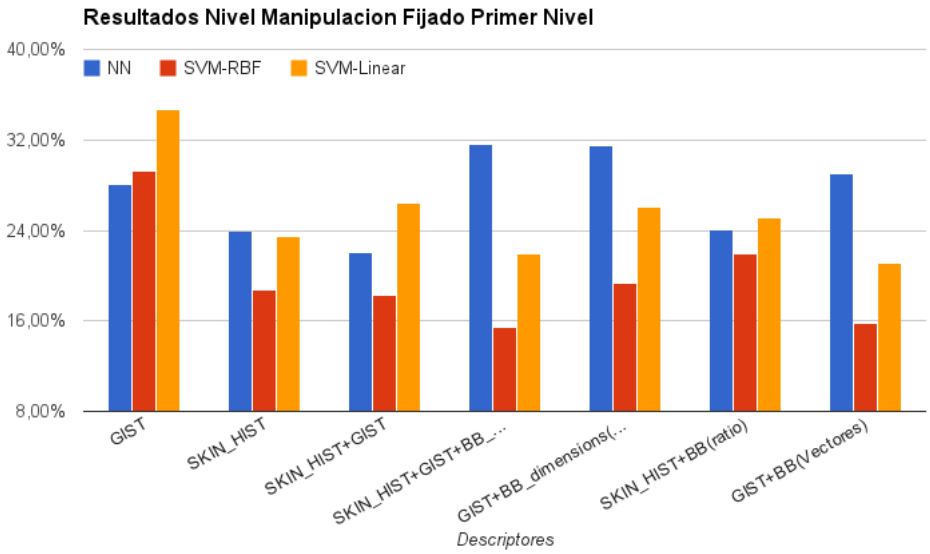


Figura 4.4: Gráfica que muestra el % de aciertos medio en el nivel de Manipulación, habiendo fijado el primer nivel, de cada una de las combinaciones de descriptores y clasificadores utilizados.

riores. Como ha sido explicado antes, las acciones incluidas en este nivel son altamente confundibles . Por lo tanto, el *GIST* se alza como mejor candidato de los descriptores seguido por el conjunto de todos los descriptores. Junto al *GIST* se utilizara un clasificador *SVM* lineal, y junto al conjunto esta el clasificador *Nearest Neighbor*.

En la Tabla 4.5, donde mostramos la matriz de confusión del mejor ejemplo para comprobar cómo se comporta nuestro clasificador, vemos que la dispersión es mucho mayor a la del nivel inicial, destacando las etiquetas de *Interacción* y *Desconocido*. Ambas tiene sentido que sean más problemáticas. La primera incluye acciones fácilmente confundibles con las acciones de una o dos manos. Y *Desconocido* incluye todo tipo de imágenes difícilmente clasificables. Por lo tanto podemos afirmar que para *Dos Manos* y *Una Mano* realiza una clasificación satisfactoria, pero que para una clasificación más óptima del resto de etiquetas haría falta mejor información discriminante.

Nivel No Manipulación

En el caso del nivel de No Manipulación nos encontramos con un problema. Si miramos los resultados obtenidos en porcentajes de aciertos, como se pueden ver en el Anexo C, tenemos que las mejores opciones obtienen porcentajes del 80 %. Sin embargo, tras estudiar más a fondo estos resultados, decidimos calcular la matriz de confusión.

En la Tabla 4.6, donde está dicha matriz, vemos que los resultados son muy negativos. Estamos ante el problema de que el clasificador no está bien entrenado o no consigue discriminar bien las clases, ya que tiene un sesgo muy alto hacia la etiqueta de Andar.

	Dos Manos (1062)	*Una Mano* (333)	*Interacción* (597)	*Desconocido* (313)
Dos Manos	0.59	0.1	0.38	0.26
Una Mano	0.1	0.54	0.13	0.33
Interacción	0.11	0.27	0.25	0.15
Desconocido	0.2	0.09	0.24	0.26

Cuadro 4.5: Ejemplo de matriz de confusión para el nivel de manipulacion. La fila representa la etiqueta reconocida y la columna el *ground truth*. Los números entre paréntesis indican el número de test de cada etiqueta.

	Andar (1634)	*Escaleras* (33)	*Parado* (105)	*Sentado* (3)	*Pantalla* (5)	*Poster* (7)	*Hablar* (38)
Andar	0.98	0.94	1	1	0.2	1	0.95
Escaleras	0	0	0	0	0	0	0
Parado	0.01	0.06	0	0	0	0	0.05
Sentado	0	0	0	0	0	0	0
Pantalla	0	0	0	0	0	0	0
Poster	0	0	0	0	0.8	0	0
Hablar	0	0	0	0	0	0	0

Cuadro 4.6: Ejemplo de matriz de confusión para el nivel de no-manipulacion. La fila representa la etiqueta reconocida y la columna el *ground truth*. Los números entre paréntesis indican el número de test de cada etiqueta.

Por lo tanto, el porcentaje del 80 % es engañoso debido al alto porcentaje de Etiquetas Andar en el *ground truth*. Para poder obtener una clasificación más óptima habría que buscar información más discriminativa.

Conclusión

Por lo tanto, y gracias a este experimento, vemos que la mejor combinación final por niveles sería: En el primer nivel el descriptor del Histograma de piel junto con el clasificador *SVM* de *kernel RBF*, en el nivel de manipulación utilizaremos el descriptor *GIST* junto con el clasificador *SVM* de *kernel* lineal y en el tercer nivel utilizaremos el descriptor del histograma de piel junto con el clasificador *Nearest Neighbor*. Sin embargo, como hemos explicado antes, el comportamiento del clasificador en niveles superiores al primero no es deseado, aunque no pueda ser solucionado con los métodos estudiados(falta información discriminativa para la clasificación).

4.2.3. Experimento sin niveles

Por último, realizaremos un experimento comprobando la diferencia entre la división de etiquetas en niveles o utilizar todas a un mismo nivel.

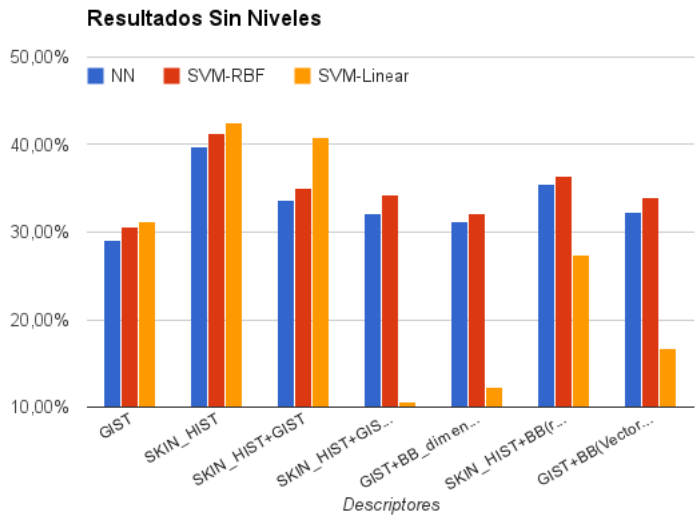


Figura 4.5: Gráfica que muestra el % de aciertos medio en un único nivel con cada una de las combinaciones de descriptores y clasificadores utilizados.

En la Figura 4.5 tenemos la gráfica con los resultados del experimento explicado antes. En ella podemos ver que la configuración con mejor porcentaje de acierto es utilizar el descriptor del histograma de piel junto con el clasificador *SVM* y su *kernel* lineal. El resultado obtenido no es concluyente a la hora de afirmar que la división es mejor o peor. Para ello, calcularemos la matriz de confusión donde nos mostrara el comportamiento de esta clasificación.

En la Tabla 4.7 tenemos la matriz de confusión para una de las secuencias. Como se puede observar, el comportamiento del clasificador es muy parecido a lo que explicábamos en la sección 4.2.2, no discrimina suficientemente bien y utiliza en exceso una etiqueta(Andar). Pero además ahora, con respecto a los experimentos con división, ese exceso incluye las etiquetas de Manipulación (llegando a tener un 30 % de alguna de ellas). Por lo tanto, y dado que nuestro clasificador con la división por niveles era capaz de discriminar satisfactoriamente el nivel de Manipulación/No-Manipulación, nos será más útil utilizar la división por niveles para futuros análisis.

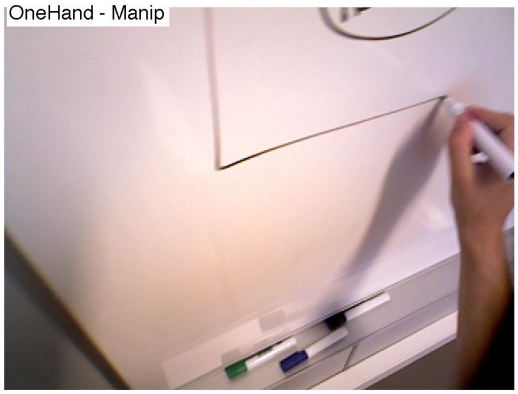
4.2.4. Experimento adicionales

Como última prueba hemos clasificado una secuencia que no habíamos utilizado en ninguno de los experimentos anteriores. Esto nos indicara cómo de condicionado está

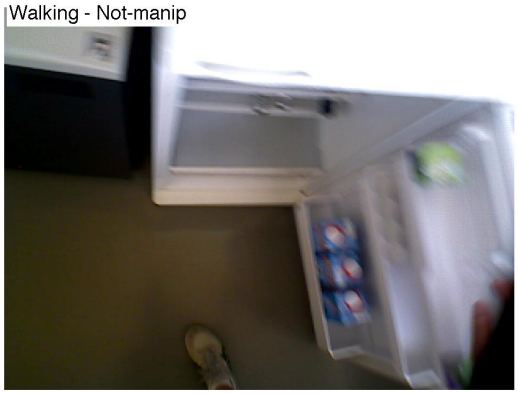
	Andar (1980)	Escaleras (87)	Parado (260)	Sentado (9)	Pantalla (0)	Poster (78)	Hablar (73)	Dos Manos (484)	Una Mano (351)	Interaccion (331)	Desconocido (556)
Andar	0.97	1	0.81	0.22	0	0	0.68	0.01	0.35	0.27	0.12
Escaleras	0	0	0	0	0	0	0.01	0.11	0	0	0
Parado	0.01	0	0.05	0.22	0	0	0.05	0.04	0.13	0.13	0.18
Sentado	0	0	0	0	0	0	0	0	0	0	0
Pantalla	0	0	0	0	0	0	0.03	0.01	0	0	0.02
Poster	0	0	0	0	0	0	0.04	0	0	0.02	0
Hablar	0	0	0	0	0	0	0	0	0.02	0.01	0.03
DosManos	0	0	0.02	0	0	0	0.04	0.15	0.21	0.08	0.13
UnaMano	0.01	0	0.05	0.11	0	0	0.12	0.12	0.17	0.26	0.2
Interaccion	0.01	0	0.06	0.44	0	0	0	0.25	0.11	0.18	0.22
Desconocido	0	0	0	0	0	0	0.01	0.3	0.01	0.05	0.09

Cuadro 4.7: Ejemplo de matriz de confusión. La fila representa la etiqueta reconocida y la columna el *ground truth*. Los números entre paréntesis indican el número de test de cada etiqueta.

nuestro clasificador con respecto a las secuencias utilizadas. La Figura 4.6 muestra un conjunto de 'frames' de la secuencia con la etiqueta insertada en la imagen. Mediante una inspección visual observamos que el comportamiento es parecido al explicado en los experimentos anteriores: el primer nivel discrimina correctamente entre manipulación/no-manipulación, pero, en el nivel dos, esta discriminación es mucho más difusa.

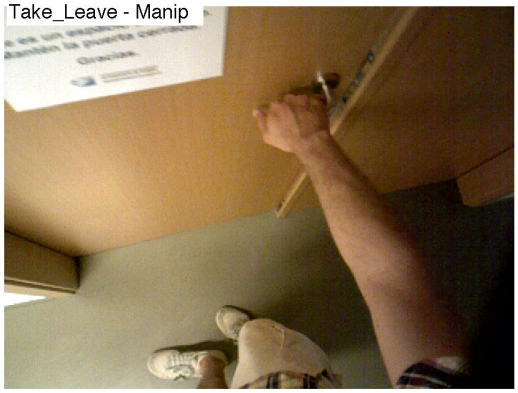


(a) El primer y el segundo nivel correcto.



(b) El primer nivel incorrecto.

(c) El primer nivel correcto y el segundo nivel incorrecto.



(d) El primer nivel correcto y el segundo nivel incorrecto.

Figura 4.6: Visualización de un conjunto de ejemplos de imágenes de una secuencia con el texto de la etiqueta reconocida insertada. La secuencia mostrada no ha sido utilizada en ningún otro experimento.

Capítulo 5
Conclusiones

La meta de este proyecto era la de estudiar y realizar un reconocedor de acciones humanas para imágenes RGB-D simples obtenidas desde una cámara RGB-D en un casco. Se obtuvieron varias secuencias para su uso durante el proyecto. Primero se realizó un etiquetado manual para conseguir un *ground truth* con el que comparar los resultados futuros. Luego se estudió e implementó la extracción de información de estas imágenes. Se estudiaron y plantearon dos tipos de clasificación: todas las etiquetas a un mismo nivel o una división por niveles. Por último se realizó y documentó un conjunto de experimentos para la eficiencia de las distintas combinaciones de clasificación.

A partir de los resultados de estos experimentos podemos extraer varias conclusiones. La clasificación en la división inicial de manipulación/no-manipulación es aceptable, siendo la mejor combinación el descriptor Histograma de Piel junto con el clasificador *SVM* y el *kernel RBF*, dando un resultado del 84 %. En el nivel de manipulación, las etiquetas *Interacción* y *Desconocido* no se clasifican correctamente (están al nivel de aleatorio), pero las etiquetas *Dos Manos* y *Una Mano* son clasificadas de manera aceptable. En el nivel de no-manipulación nos encontramos con un sesgo hacia la etiqueta *Andar* que hace que nuestro clasificador esté desbalanceado. En la clasificación sin niveles nos encontramos con el mismo problema que en el nivel de no-manipulación, pero esta vez el sesgo influye en todas las etiquetas, reduciendo la eficiencia en las etiquetas del nivel de manipulación.

Otra de las conclusiones importantes que podemos sacar es sobre los usos de los clasificadores. Por norma general, en muchos trabajos vemos que cuando utilizan un clasificador de tipo *SVM* utilizan todos los descriptores juntos sin comprobar si funcionan bien o mal. En nuestro caso al mirar distintas combinaciones hemos visto que no siempre es lo mejor utilizar todos, sino elegir los mejores.

El tiempo utilizado para este proyecto ha sido divido como se muestra en la Figura 5.1.

Por ultimo, como conclusión personal, este proyecto estaba altamente relacionado con la mecánica de trabajo de estudios de investigación y me ha hecho tener que modificar mi forma de trabajo a este tipo de mecánicas. Ha sido una experiencia enriquecedora y muy apasionante. No ha sido siempre perfecto, pero en conjunto estoy satisfecho con el

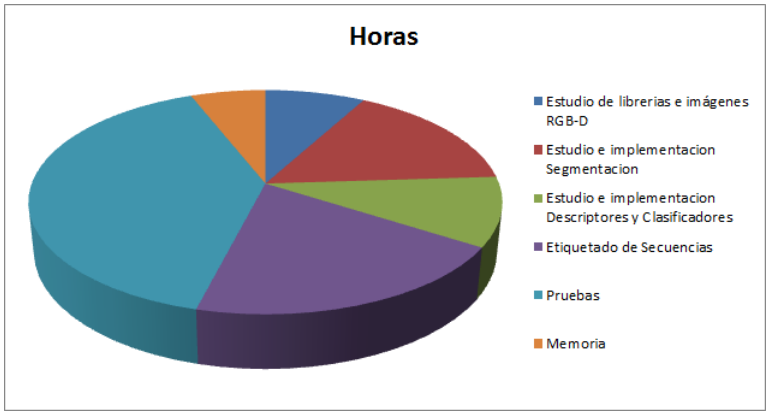


Figura 5.1: Distribución Horas

trabajo realizado.

5.1. Trabajo Futuro

Como posible trabajo futuro se debería añadir al reconocedor la información temporal de la acción. Habría varias maneras de hacerlo añadiendo mayor complejidad. Por ejemplo, se podría utilizar descriptores que hagan uso del momento temporal de la imagen dentro de un conjunto. Esto llevaría a que el reconocedor no trabajara con una única imagen sino con un conjunto de ellas que formen una acción.

Otros posibles trabajos futuros serian el análisis de otro tipo de cámaras colocadas en otras partes del cuerpo. Esto se podría utilizar para comprobar si la pérdida de la información sobre la profundidad influiría en el resultado.

También podría ser interesante el optimizar la ejecución del reconocedor para que funcionara a tiempo real y poder realizar un análisis automático durante la captura. Una de las utilidades para las que este proyecto se pensó fue el uso terapéutico y de control que pueda tener. En caso de hacerlo en tiempo real se podría enviar información de forma automática cada cierto tiempo para su análisis.

Por ultimo, el uso de otros clasificadores y/o descriptores distintos a los utilizados en este proyecto, ademas de otras posibles combinaciones que no hayan sido pensadas.

Bibliografía

[1] O'Hara Kieron, Mischa M. Tuffield, Nigel Shadbolt, *Lifelogging: Privacy and empowerment with memories for life.* Identity in the Information Society (Springer) 1 2009

[2] W.W. Mayol, A.J. Davison, B.J. Tordoff, N.D. Molton, and D.W. Murray, *Interaction between hand and wearable camera in 2D and 3D environments.* Proc. British Machine Vision Conference 2004. London, UK, September. 2004.

[3] Yin Li, Alireza Fathi, James M. Rehg, *Learning to Predict Gaze in Egocentric Video.* International Conference on Computer Vision (ICCV) 2013.

[4] D. Damen, A. Gee, W. Mayol-Cuevas, and A. Calway, *Egocentric Real-time Workspace Monitoring using an RGB-D Camera.* IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vila Moura, Portugal, October 2012.

[5] S.L.Phung,A. Bouzerdoum,D. Chai, Sr., *Skin segmentation using color pixel classification: analysis and comparison.* Pattern Analysis and Machine Intelligence, IEEE Transactions on Jan. 2005.

[6] S.L.Phung,A. Bouzerdoum,D. Chai, *Skin segmentation using color and edge information.* Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on July 2003.

[7] Robert Y. Wang, Jovan Popović, *Real-Time Hand-Tracking with a Color Glove.* ACM SIGGRAPH 2009 Papers.

[8] Yong Jae Lee, Joydeep Ghosh, Kristen Grauman, *Discovering Important People and Objects for Egocentric Video Summarization.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, June 2012.

[9] M.Kolsch, M.Turk, *Robust hand detection.* Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on May 2004

[10] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk, *SLIC Superpixels.* EPFL Technical Report no. 149300, June 2010.

[11] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani and Luc Van Gool, *SEEDS: Superpixels Extracted via Energy-Driven Sampling.* 12th European Conference on Computer Vision (ECCV), October 2012.

[12] Stuart J. Russell, Peter Norvig, *Artificial Intelligence: A Modern Approach.* Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2.

[13] Vladimir Vezhnevets, Vassili Sazonov, Alla Andreeva, *A Survey on Pixel-Based Skin Color Detection Techniques.* In Proceedings of the GraphiCon 2003.

[14] R.B. Rusu ,S. Cousins, *3D is here: Point Cloud Library (PCL).* Robotics and Automation (ICRA), 2011 IEEE International Conference on May 2011

[15] Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, Cordelia Schmid, *Evaluation of GIST descriptors for web-scale image search.* International Conference on Image and Video Retrieval - july 2009.

[16] Aude Oliva, Antonio Torralba, *Modeling the shape of the scene: a holistic representation of the spatial envelope.* International Journal of Computer Vision, Vol. 42(3): 145-175, 2001.

[17] Michael Calonder, Vincent Lepetit, Christoph Strecha, Pascal Fua, *BRIEF: Binary Robust Independent Elementary Features.* 11th European Conference on Computer Vision (ECCV), Heraklion, Crete. LNCS Springer, September 2010

[18] Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary R. Bradski, *ORB: An efficient alternative to SIFT or SURF.* ICCV 2011

[19] `http://en.wikipedia.org/wiki/Minimum_bounding_rectangle`

[20] K. Pearson, *On Lines and Planes of Closest Fit to Systems of Points in Space.* Philosophical Magazine 2 1901.

[21] G. Bradski, *The OpenCV Library.* Dr. Dobb's Journal of Software Tools 2000.

[22] J. B. MacQueen, *Some Methods for classification and Analysis of Multivariate Observations.* Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability (1967).

[23] Corinna Cortes , Vladimir Vapnik, *Support-vector networks.* In Machine Learning 20, 1995.

[24] Chih-Chung Chang, Chih-Jen Lin, *LIBSVM: a library for support vector machines.* ACM Transactions on Intelligent Systems and Technology, 2011

Índice de figuras

Anexo A
Cámaras RGB-D

Las cámaras RGB-D llevan en el mercado desde principio de siglo cuando la tecnología fotográfica digital estaba en auge, con modelos como la cámara *ZCAM*[1], sin embargo en los últimos años con la presentación, por parte de *Microsoft*, de *Kinect*[2] y el desarrollo de cámaras más asequibles están aumentando el número de proyectos relacionados con ellas.

Figura A.1: Modelo *Asus Xtation Pro*

En nuestro proyecto utilizaremos un modelo parecido al de Kinect pero desarrollado por *Asus*: la *Asus Xtation Pro*, como vemos en la imagen A.1.

[1] `http://en.wikipedia.org/wiki/ZCam`
[2] `http://en.wikipedia.org/wiki/Kinect`

A.1. Funcionamiento

El sensor de profundidad está formado por dos componentes: un proyector de luz infrarroja (IR) y un sensor CMOS monocromo estándar. La idea principal consiste en un proceso en dos fases, una primera de calibración, y otra de funcionamiento.
En la fase de calibración, se emplea el proyector de luz infrarroja para proyectar un patrón de puntos sobre un plano de la escena, variando su distancia entre posiciones conocidas. A su vez, la cámara captura una imagen del patrón proyectado sobre el plano para cada una de estas distancias. Las imágenes obtenidas se denominan imágenes de referencia y se almacenan en el sensor.
En la fase de funcionamiento se emplean las imágenes de referencia para sustituir 'virtualmente' al emisor del patrón IR, de tal manera que para cada nueva imagen capturada por el sensor, el cálculo de profundidad se resume a un problema de visión estéreo con configuración ideal: cámaras idénticas, ejes alineados y separados una distancia base.

A.2. Especificaciones Técnicas *Asus Xtation Pro*

Power Consumption	below 2.5W
Distance of Use	between 0.8m and 3.5m
Field of View	58° H, 45° V, 70° D (Horizontal, Vertical, Diagonal)
Sensor	RGB+depth
Depth Image Size	VGA (640x480) : 30fps QVGA (320x240): 60fps
Platform	Intel X86 & AMD OS Support Win 32/64:XP/Vista/7/8 Linux Ubuntu 10.10:X86, 32/64bit Android(by request)
Interface	USB2.0
Software	software development kit(OpenNI SDK bundled)
Programming Language	C++/C# (Windows) C++(Linux) JAVA
Operation Environment	Indoor
Dimensions	18 x 3.5 x 5 cm

Anexo B
Clasificadores

B.1. *Nearest Neighbor*(Vecino más proximo)

El método Nearest neighbor, también conocido como búsqueda de proximidad o búsqueda del punto más cercano, es un problema de optimización para encontrar el punto(o valor) más cercano.La cercanía es típicamente expresada en términos de una función de desemejanza: Cuanto menos parecidos son los objetos mayor es el valor de la función. Formalmente, el problema de búsqueda de Nearest Neighbor es definido de la siguiente manera: Dado un conjunto S de puntos en un espacio M y un punto de consulta $q \in M$, encontrar el punto más cercano en S a q. Donald Knuth en el Volumen 3 de 'The Art of Computer Programming' (1973) lo llamó el problema de correos, refiriéndose a la aplicación de asignar a una residencia la oficina de correos más cercana. Una generalización directa de este problema es la búsqueda k-NN, donde necesitaremos encontrar los k puntos más cercanos.

Se han propuesto varias soluciones al problema de la búsqueda NN. La calidad y utilidad de estos algoritmos viene determinada por la complejidad temporal de las consultas así como de la complejidad espacial de cualquier estructura de búsqueda que necesite mantenimiento. La observación informal, normalmente referida como la maldición de la dimensionalidad, mantiene que no hay una solución exacta de propósito general para el problema NN en el espacio de alta dimensián euclidiana utilizando preprocesamiento polinomial y un tiempo de bśqueda polilogarítmico. Algunos métodos utilizados son:

Búsqueda Lineal

La solución más simple al problema NN es calcular las distancias desde el punto de consulta a todos los demás puntos de la base de datos, manteniendo el mejor hasta el momento. Este algoritmo, algunas veces referido como aproximación ingenua, tiene un tiempo de ejecución de $O(N_d)$ donde N es la cardinalidad de S y d es la dimensionalidad de M. No hay ningún tipo de estructura de datos que mantener, asi que la búsqueda lineal no tiene complejidad espacial más allá del almacenamiento de la base de datos.

Particionamiento del espacio

Otro enfoque a este problema es el particionamiento del espacio. En el caso de un espacio Euclidiano este enfoque es conocido como índice espacial o método de acceso espacial. Varios métodos de particionamiento del espacio se han desarrollado para resolver el problema de NN. El más sencillo y utilizado es el 'k-d tree', el cual iterativamente disecciona el espacio de búsqueda en dos regiones que contienen la mitad de los puntos de la región padre. Las búsquedas son realizadas a través de un árbol desde la raíz hasta una hoja evaluando el punto de consulta en cada separación. Dependiendo de la distancia especificada en la búsqueda, ramas vecinas que pueden contener aciertos también necesitarán ser evaluadas.

Fichero de vector de aproximación

En espacio de alta dimensión, las estructuras de indexación en árbol se vuelve inútiles por el incremento del porcentaje de nodos que necesitan ser examinados de cualquier manera. Para acelerar la búsqueda lineal, una versión comprimida del vector de características guardado en la RAM es usada para pre-filtrar la base de datos en una pasada. Los candidatos finales son determinados en una segunda etapa usando los datos descomprimidos de disco para el calculo de distancias.

Búsqueda basada en la 'clusterización'

La anterior aproximación es un caso especial de la búsqueda basada en la compresión, donde cada característica es comprimida uniformemente e independientemente. La técnica de compresión óptima en espacios multidimensionales es la Cuantificación Vectorial, implementada a traves de la clusterización. La base de datos es clusterizada y los clusters más prometedores son recuperados.

B.2. *SVM (Support Vector Machine)*

Las máquinas de soporte vectorial o máquinas de vectores de soporte (*Support Vector Machines, SVMs*) son un conjunto de algoritmos de aprendizaje supervisado desarrollados por *Vladimir Vapnik* y su equipo en los laboratorios *AT&T*.

Estos métodos están propiamente relacionados con problemas de clasificación y regresión. Dado un conjunto de ejemplos de entrenamiento (de muestras) podemos etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra. Intuitivamente, una SVM es un modelo que representa a los puntos de muestra en el espacio, separando las clases por un espacio lo más amplio posible. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de su proximidad pueden ser clasificadas a una u otra clase.

Más formalmente, una SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación o regresión. Una buena separación entre las clases permitirá un

clasificación correcta.

Funcionamiento

Dado un conjunto de puntos, subconjunto de un conjunto mayor (espacio), en el que cada uno de ellos pertenece a una de dos posibles categorías, un algoritmo basado en SVM construye un modelo capaz de predecir si un punto nuevo (cuya categoría desconocemos) pertenece a una categoría o a la otra.

Como en la mayoría de los métodos de clasificación supervisada, los datos de entrada (los puntos) son vistos como un vector p-dimensional (una lista de p números).

La SVM busca un hiperplano que separe de forma óptima a los puntos de una clase de la de otra, que han podido ser previamente proyectados a un espacio de dimensionalidad superior.

En ese concepto de 'separación óptima' es donde reside la característica fundamental de las SVM: este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia (margen) con los puntos que estén más cerca de él mismo. Por eso también a veces se les conoce a las SVM como clasificadores de margen máximo. De esta forma, los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado.

Los algoritmos SVM pertenecen a la familia de los clasificadores lineales. También pueden ser considerados un caso especial de la regularización de Tikhonov.

En la literatura de los SVMs, se llama atributo a la variable predictora y característica a un atributo transformado que es usado para definir el hiperplano. La elección de la representación más adecuada del universo estudiado, se realiza mediante un proceso denominado selección de características.

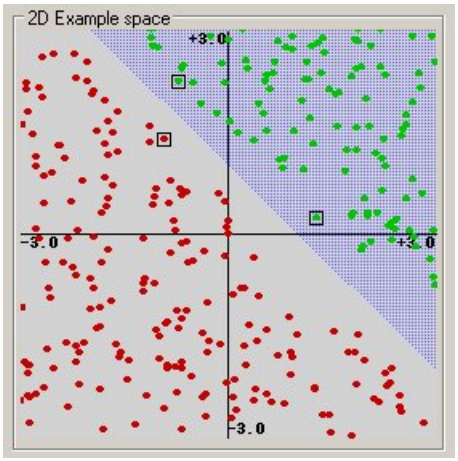
Al vector formado por los puntos más cercanos al hiperplano se le llama vector de soporte.

Los modelos basados en SVMs están estrechamente relacionados con las redes neuronales. Usando una función kernel, resultan un método de entrenamiento alternativo para clasificadores polinomiales, funciones de base radial y perceptrón multicapa.

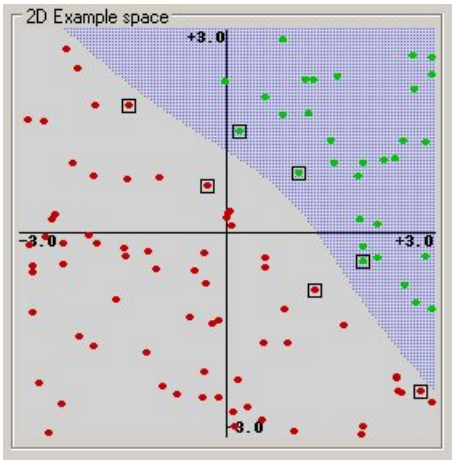
B.2.1. Kernel

Debido a las limitaciones computacionales de las máquinas de aprendizaje lineal, éstas no pueden ser utilizadas en la mayoría de las aplicaciones del mundo real. La representación por medio de funciones Kernel ofrece una solución a este problema, proyectando la información a un espacio de características de mayor dimensión el cual aumenta la capacidad computacional de la máquinas de aprendizaje lineal. Es decir, mapearemos el espacio de entradas X a un nuevo espacio de características de mayor dimensionalidad. Los kernels más utilizados son los siguientes:

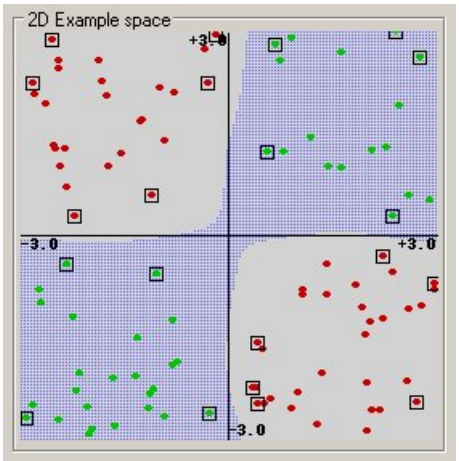
- Lineal(Figura B.1a): $K(x_i, x_j) = x_i^t x_j$
- Polinomial-homogénea(Figura B.1b):$K(x_i, x_j) = (\gamma x_i^T x_j + coef0)^{degree}, \gamma > 0$
- Función de base radial (RBF)(Figura B.1c): $K(x_i, x_j) = e^{-\gamma ||x_i - x_j||^2}, \gamma > 0$
- Sigmoidal(Figura B.1d):$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + coef0)$



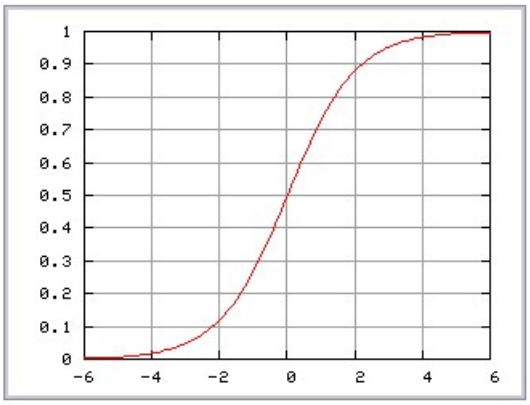
(a) Ejemplo kernel lineal



(b) Ejemplo kernel Polinómico



(c) Ejemplo kernel RBF



(d) Ejemplo función sigmoidal

Figura B.1: Visualización de ejemplos de los kernels lineal, polinómico y RBF y de la forma de la función sigmoidal.

Anexo C
Resultados

En este Anexo mostraremos todos los resultados utilizados o referenciados en la memoria principal. Primero mostraremos los valores medios de los datos para un resumen inicial y luego mostraremos los propios de cada secuencia.

C.1. Resumen Resultados

En esta sección mostraremos un resumen de los experimentos de las secuencias.

C.1.1. Experimentos Iniciales

En esta sección, y en la de cada secuencia, se mostrará el resultado de un experimento inicial con todos los posibles *kernel* de *SVM* para comprobar si era necesario continuar con todas las opciones de clasificador. En la Tabla C.1 vemos el resumen del resultado de este experimento.

C.1.2. Experimentos por niveles

En esta sección, y en la de cada secuencia, se mostrará el resultado de la ejecución del experimento utilizando la misma configuración en todos los niveles y la división por niveles. En las Tablas C.2, C.3 y C.4 vemos el resumen del resultado de este experimento.

	NN	SVM-RBF	SVM-Lineal	SVM-Sigmoidal	SVM-Polinómico
GIST	61.91	60.78	49.64	57.63	58.102
SKIN_HIST	78.437	73.52	73.44	25.543	49.38
SKIN_HIST+GIST	64.02	74.25	58.086	57.63	47.914
SKIN_HIST+GIST+ BB_dimensions	76.53	79.811	0	0	0
GIST+BB_dimensions	0	79.141	0	0	0

Cuadro C.1: Resumen de resultados de la ejecución de un experimento inicial.

	NN	SVM-RBF	SVM-Lineal
GIST	60.91	64.43	65.83
SKIN_HIST	81.90	**84**	80.5
SKIN_HIST+GIST	79.48	81.05	80.56
SKIN_HIST+GIST+BB_dimensions(Total)	76.54	80.22	51.85
GIST+BB_dimensions(Total)	76.63	74.28	45.45
SKIN_HIST+BB(ratio)	80.69	**83.67**	55.8
GIST+BB(Vectores)	76.60	74.26	43.64

Cuadro C.2: % Aciertos medios primer nivel

	NN	SVM-RBF	SVM-Lineal
GIST	**38.20**	33.21	37.9
SKIN_HIST	26.38	27.5	24.19
SKIN_HIST+GIST	24.95	27.55	26.68
SKIN_HIST+GIST+BB_dimensions(Total)	29.08	31.16	20.24
GIST+BB_dimensions(Total)	28.62	30.4	29.02
SKIN_HIST+BB(ratio)	26.77	30.95	22.19
GIST+BB(Vectores)	28.7	30.36	25

Cuadro C.3: % Aciertos medios nivel manipulación.

	NN	SVM-RBF	SVM-Lineal
GIST	79.42	80.63	**83.59**
SKIN_HIST	77.16	80.87	78.01
SKIN_HIST+GIST	75.05	80.93	78.32
SKIN_HIST+GIST+BB_dimensions(Total)	78.42	82.11	52.07
GIST+BB_dimensions(Total)	77.79	**83.14**	60.99
SKIN_HIST+BB(ratio)	82.06	82.32	39.45
GIST+BB(Vectores)	77.93	82.94	16.58

Cuadro C.4: % Aciertos medios nivel no-manipulación

	NN	SVM-RBF	SVM-Lineal
GIST	28.03	29.20	**34.70**
SKIN_HIST	23.98	18.70	23.43
SKIN_HIST+GIST	22.04	18.32	26.46
SKIN_HIST+GIST+BB_dimensions(Total)	31.59	15.43	21.97
GIST+BB_dimensions(Total)	31.46	19.31	26.02
SKIN_HIST+BB(ratio)	24.06	21.95	25.07
GIST+BB(Vectores)	29.04	15.76	21.16

Cuadro C.5: % Aciertos medios del nivel manipulación habiendo fijado el primer nivel.

	NN	SVM-RBF	SVM-Lineal
GIST	57.98	64.52	68.62
SKIN_HIST	**80.16**	79.10	80.12
SKIN_HIST+GIST	60.65	69.65	74.66
SKIN_HIST+GIST+BB_dimensions(Total)	79.57	68.59	29.03
GIST+BB_dimensions(Total)	79.51	67.01	18.56
SKIN_HIST+BB(ratio)	74.5	77.94	64.24
GIST+BB(Vectores)	78.95	71.15	29.61

Cuadro C.6: % Aciertos medios del nivel no-manipulación habiendo fijado el primer nivel.

C.1.3. Experimentos fijando nivel 1

En esta sección, y en la de cada secuencia, se mostrará el resultado de la ejecución del experimento habiendo fijado la configuración del primer nivel dado que este influía en el segundo nivel. No habrá resultados del primer nivel ya que son los mismos que en el experimento anterior. En las Tablas C.5 y C.6 vemos el resumen del resultado de este experimento.

C.1.4. Experimentos 11 etiquetas

En esta sección, y en la de cada secuencia, se mostrará el resultado de la ejecución del experimento utilizando las 11 etiquetas del nivel 2 sin división en el primer nivel. En la Tabla C.7 vemos el resumen del resultado de este experimento.

C.2. Secuencia User_Ada_Byron-1(Alejandro)

En esta sección veremos los resultados de los experimentos para la secuencia User_Ada_Byron-1. En la Figura C.1 vemos un resumen de esta secuencia en forma de mosaico.

	NN	SVM-RBF	SVM-Lineal
GIST	29.13	30.54	31.19
SKIN_HIST	39.82	41.36	**42.51**
SKIN_HIST+GIST	33.61	35.05	40.79
SKIN_HIST+GIST+BB_dimensions(Total)	32.15	34.22	10.56
GIST+BB_dimensions(Total)	31.18	32.09	12.22
SKIN_HIST+BB(ratio)	35.45	36.36	27.46
GIST+BB(Vectores)	32.36	33.98	16.65

Cuadro C.7: % Aciertos medios sin niveles

Figura C.1: Resumen Secuencia User_Ada_Byron-1

C.2.1. Experimentos Iniciales

En la Tabla C.8 se muestran los resultados del experimento inicial.

C.2.2. Experimentos por niveles

En las Tablas C.9, C.11 y C.10 se muestran los resultados del experimento utilizando la misma configuración en los dos niveles.

	NN	SVM-RBF	SVM-Lineal	SVM-Sigmoidal	SVM-Polinómico
GIST	66.16	64.8	54.14	56.56	60.696
SKIN_HIST	84.27	89.24	89.577	21.8	44.67
SKIN_HIST+GIST	67.2	74.78	44.62	56.56	46.02
SKIN_HIST+GIST+ BB_dimensions	84.52	90.01	0	0	0
GIST+BB_dimensions	0	90.11	0	0	0

Cuadro C.8: Resultados del experimento inicial para la secuencia User_Ada_Byron-1.

	NN	SVM-RBF	SVM-Lineal
GIST	62.82	65.063	69.42
SKIN_HIST	85.1	90.35	83.99
SKIN_HIST+GIST	82.75	87.98	83.73
SKIN_HIST+GIST+BB_dimensions(Total)	82.49	88.31	51.875
GIST+BB_dimensions(Total)	83.23	83.27	58.28
SKIN_HIST+BB(ratio)	89	90.66	39.227
GIST+BB(Vectores)	82.87	83.27	38.625

Cuadro C.9: Resultados del experimento del primer nivel con la misma configuración en todos los niveles para la secuencia User_Ada_Byron-1.

	NN	SVM-RBF	SVM-Lineal
GIST	84.44	88.74	87.21
SKIN_HIST	80.52	84.05	81.73
SKIN_HIST+GIST	79.55	84.31	81.546
SKIN_HIST+GIST+BB_dimensions(Total)	81.97	85.32	58.08
GIST+BB_dimensions(Total)	82.249	87.94	61.415
SKIN_HIST+BB(ratio)	83.64	84.32	12.5
GIST+BB(Vectores)	82.17	87.27	1.7398

Cuadro C.10: Resultados del experimento del nivel de no manipulación con la misma configuración en todos los niveles para la secuencia User_Ada_Byron-1.

	NN	SVM-RBF	SVM-Lineal
GIST	31.25	30.383	31.95
SKIN_HIST	26.117	26.593	23.53
SKIN_HIST+GIST	21.17	30.61	29.047
SKIN_HIST+GIST+BB_dimensions(Total)	32.843	31.2	23.9
GIST+BB_dimensions(Total)	31.89	29.62	28.688
SKIN_HIST+BB(ratio)	28.22	27.51	18.691
GIST+BB(Vectores)	32.203	29.273	25.492

Cuadro C.11: Resultados del experimento del nivel de manipulación con la misma configuración en todos los niveles para la secuencia User_Ada_Byron-1.

	NN	SVM-RBF	SVM-Lineal
GIST	62.476	69.749	70.827
SKIN_HIST	82.62	82.92	82.75
SKIN_HIST+GIST	62.39	74.36	77.17
SKIN_HIST+GIST+BB_dimensions(Total)	82.88	75.05	1.1199
GIST+BB_dimensions(Total)	82.75	73.765	9.0391
SKIN_HIST+BB(ratio)	82.88	82.92	83.31
GIST+BB(Vectores)	82.17	87.27	1.7398

Cuadro C.12: Resultados del experimento del nivel de no manipulación habiendo fijado el primer nivel para la secuencia User_Ada_Byron-1.

C.2.3. Experimentos fijando nivel 1

En las Tablas C.13 y C.12 se muestran los resultados del experimento habiendo fijado el primer nivel con la configuración más óptima.

C.2.4. Experimentos 11 etiquetas

En la Tabla C.14 se muestran los resultados del experimento en el que no se utilizan niveles, solo un único nivel de 11 etiquetas.

C.3. Secuencia User_Ada_Byron-2(Alejo)

En esta sección veremos los resultados de los experimentos para la secuencia User_Ada_Byron-2. En la Figura C.2 vemos un resumen de esta secuencia en forma de mosaico.

C.3.1. Experimentos Iniciales

En la Tabla C.15 se muestran los resultados del experimento inicial.

	NN	SVM-RBF	SVM-Lineal
GIST	22.34	29.78	26.778
SKIN_HIST	27.188	16.867	26.367
SKIN_HIST+GIST	19.809	16.668	28.28
SKIN_HIST+GIST+BB_dimensions(Total)	27.188	17.21	17.96
GIST+BB_dimensions(Total)	30.668	20.49	18.992
SKIN_HIST+BB(ratio)	27.66	17.082	26.778
GIST+BB(Vectores)	26.02	20.29	19.328

Cuadro C.13: Resultados del experimento del nivel de manipulación habiendo fijado el primer nivel para la secuencia User_Ada_Byron-1.

	NN	SVM-RBF	SVM-Lineal
GIST	37.633	37.633	38.92
SKIN_HIST	51.555	51.555	50.8
SKIN_HIST+GIST	43.74	43.74	49.2
SKIN_HIST+GIST+BB_dimensions(Total)	43.91	43.91	8.2011
GIST+BB_dimensions(Total)	42.27	42.27	9.31
SKIN_HIST+BB(ratio)	51.687	51.687	50.56
GIST+BB(Vectores)	42.196	42.196	5.7

Cuadro C.14: Resultados del experimento utilizando un único nivel de etiquetas para la secuencia User_Ada_Byron-1.

	NN	SVM-RBF	SVM-Lineal	SVM-Sigmoidal	SVM-Polinómico
GIST	59.74	62.133	38.804	56.75	66.06
SKIN_HIST	78.8	41.515	41.414	21.75	35.49
SKIN_HIST+GIST	62.032	69.31	37.92	56.74	32.81
SKIN_HIST+GIST+ BB_dimensions	82.19	78.796	0	0	0
GIST+BB_dimensions	0	83.91	0	0	0

Cuadro C.15: Resultados del experimento inicial para la secuencia User_Ada_Byron-2.

Figura C.2: Resumen Secuencia User_Ada_Byron-2

C.3.2. Experimentos por niveles

En las Tablas C.16, C.18 y C.17 se muestran los resultados del experimento utilizando la misma configuración en los dos niveles.

C.3.3. Experimentos fijando nivel 1

En las Tablas C.20 y C.19 se muestran los resultados del experimento habiendo fijado el primer nivel con la configuración más óptima.

C.3.4. Experimentos 11 etiquetas

En la Tabla C.21 se muestran los resultados del experimento en el que no se utilizan niveles, solo un único nivel de 11 etiquetas.

C.4. Secuencia User_Ada_Byron-3

En esta sección veremos los resultados de los experimentos para la secuencia User_Ada_Byron-3. En la Figura C.3 vemos un resumen de esta secuencia en forma de mosaico.

	NN	SVM-RBF	SVM-Lineal
GIST	59.664	63.59	66.36
SKIN_HIST	85.66	86.78	80.7
SKIN_HIST+GIST	81.05	84.001	80.42
SKIN_HIST+GIST+BB_dimensions(Total)	85.63	81.58	69.14
GIST+BB_dimensions(Total)	85.47	74.72	34.414
SKIN_HIST+BB(ratio)	81.92	86.32	32.39
GIST+BB(Vectores)	85.13	74.81	32.227

Cuadro C.16: Resultados del experimento del primer nivel con la misma configuración en todos los niveles para la secuencia User_Ada_Byron-2.

	NN	SVM-RBF	SVM-Lineal
GIST	72.92	73.31	73.313
SKIN_HIST	68.92	73.265	69.891
SKIN_HIST+GIST	62.641	72.31	68.109
SKIN_HIST+GIST+BB_dimensions(Total)	66.735	74.53	22.43
GIST+BB_dimensions(Total)	66.77	76.92	70.77
SKIN_HIST+BB(ratio)	75.28	73.656	26.32
GIST+BB(Vectores)	66.86	76.75	7.41

Cuadro C.17: Resultados del experimento del nivel de no manipulación con la misma configuración en todos los niveles para la secuencia User_Ada_Byron-2.

	NN	SVM-RBF	SVM-Lineal
GIST	34.47	36.273	45.554
SKIN_HIST	41.48	39.39	30.348
SKIN_HIST+GIST	30.84	38.93	29.707
SKIN_HIST+GIST+BB_dimensions(Total)	32.453	37.29	21.62
GIST+BB_dimensions(Total)	31.258	35.69	32.39
SKIN_HIST+BB(ratio)	26.953	38.4	24.312
GIST+BB(Vectores)	30.59	35.633	28.9

Cuadro C.18: Resultados del experimento del nivel de manipulación con la misma configuración en todos los niveles para la secuencia User_Ada_Byron-2.

	NN	SVM-RBF	SVM-Lineal
GIST	43.81	51.59	50.68
SKIN_HIST	71.673	72.02	72.25
SKIN_HIST+GIST	53.585	58.17	58.68
SKIN_HIST+GIST+BB_dimensions(Total)	71.91	57.993	72.36
GIST+BB_dimensions(Total)	71.91	54.65	2.27
SKIN_HIST+BB(ratio)	54.65	71.968	2.39
GIST+BB(Vectores)	69.52	71.859	71.501

Cuadro C.19: Resultados del experimento del nivel de no manipulación habiendo fijado el primer nivel para la secuencia User_Ada_Byron-2.

	NN	SVM-RBF	SVM-Lineal
GIST	23.367	26.293	30.19
SKIN_HIST	30.668	13.531	22.78
SKIN_HIST+GIST	27.26	15.191	23.47
SKIN_HIST+GIST+BB_dimensions(Total)	29.113	13.34	22.3
GIST+BB_dimensions(Total)	29.89	31.16	28.53
SKIN_HIST+BB(ratio)	27.85	31.06	28.34
GIST+BB(Vectores)	21.62	13.34	23.082

Cuadro C.20: Resultados del experimento del nivel de manipulación habiendo fijado el primer nivel para la secuencia User_Ada_Byron-2.

	NN	SVM-RBF	SVM-Lineal
GIST	30.55	30.55	31.016
SKIN_HIST	43.86	43.86	45.57
SKIN_HIST+GIST	36.555	36.555	40.554
SKIN_HIST+GIST+BB_dimensions(Total)	33.57	33.57	9.32
GIST+BB_dimensions(Total)	37.69	37.69	30.89
SKIN_HIST+BB(ratio)	43.585	43.585	45.29
GIST+BB(Vectores)	37.657	37.657	7.9208

Cuadro C.21: Resultados del experimento utilizando un único nivel de etiquetas para la secuencia User_Ada_Byron-2.

Figura C.3: Resumen Secuencia User_Ada_Byron-3

C.4.1. Experimentos Iniciales

En la Tabla C.22 se muestran los resultados del experimento inicial.

C.4.2. Experimentos por niveles

En las Tablas C.23, C.25 y C.24 se muestran los resultados del experimento utilizando la misma configuración en los dos niveles.

C.4.3. Experimentos fijando nivel 1

En las Tablas C.27 y C.26 se muestran los resultados del experimento habiendo fijado el primer nivel con la configuración más óptima.

C.4.4. Experimentos 11 etiquetas

En la Tabla C.28 se muestran los resultados del experimento en el que no se utilizan niveles, solo un único nivel de 11 etiquetas.

	NN	SVM-RBF	SVM-Lineal	SVM-Sigmoidal	SVM-Polinómico
GIST	65.59	70.16	51.875	66.937	58.485
SKIN_HIST	78.89	80.656	80.06	24.75	52.78
SKIN_HIST+GIST	68.25	76.84	73.16	66.937	60.984
SKIN_HIST+GIST+ BB_dimensions	72.94	74.33	0	0	0
GIST+BB_dimensions	0	68.66	0	0	0

Cuadro C.22: Resultados del experimento inicial para la secuencia User_Ada_Byron-3.

	NN	SVM-RBF	SVM-Lineal
GIST	63.55	66.11	68.05
SKIN_HIST	79.08	81.16	77.81
SKIN_HIST+GIST	77.08	80.45	78.58
SKIN_HIST+GIST+BB_dimensions(Total)	71.52	80.14	53.695
GIST+BB_dimensions(Total)	71.313	73.704	44.75
SKIN_HIST+BB(ratio)	76.98	81.109	50.49
GIST+BB(Vectores)	71.52	73.69	43.055

Cuadro C.23: Resultados del experimento del primer nivel con la misma configuración en todos los niveles para la secuencia User_Ada_Byron-3.

	NN	SVM-RBF	SVM-Lineal
GIST	87.14	88.92	92.54
SKIN_HIST	84.359	89.5	85.61
SKIN_HIST+GIST	82.71	91.02	87.33
SKIN_HIST+GIST+BB_dimensions(Total)	86.86	90.89	81.718
GIST+BB_dimensions(Total)	86.15	91.07	50
SKIN_HIST+BB(ratio)	89.032	89.83	0
GIST+BB(Vectores)	86.63	91.07	17.09

Cuadro C.24: Resultados del experimento del nivel de no manipulación con la misma configuración en todos los niveles para la secuencia User_Ada_Byron-3.

	NN	SVM-RBF	SVM-Lineal
GIST	40.65	38.17	41.946
SKIN_HIST	27.01	26.652	31.81
SKIN_HIST+GIST	33.554	25.867	33.92
SKIN_HIST+GIST+BB_dimensions(Total)	29.79	45.28	15.18
GIST+BB_dimensions(Total)	32.18	42.696	32.024
SKIN_HIST+BB(ratio)	36.1	44.3	30.65
GIST+BB(Vectores)	31.41	42.72	24.348

Cuadro C.25: Resultados del experimento del nivel de manipulación con la misma configuración en todos los niveles para la secuencia User_Ada_Byron-3.

	NN	SVM-RBF	SVM-Lineal
GIST	69.10	60.67	75.187
SKIN_HIST	87.29	85.5	87.34
SKIN_HIST+GIST	73.22	73.923	82.94
SKIN_HIST+GIST+BB_dimensions(Total)	86.86	72.19	19.988
GIST+BB_dimensions(Total)	86.75	69.8	52.04
SKIN_HIST+BB(ratio)	85.11	85.06	85.33
GIST+BB(Vectores)	86.75	59.8	66.11

Cuadro C.26: Resultados del experimento del nivel de no manipulación habiendo fijado el primer nivel para la secuencia User_Ada_Byron-3.

	NN	SVM-RBF	SVM-Lineal
GIST	26.742	34.28	41.07
SKIN_HIST	27.66	16.988	30.574
SKIN_HIST+GIST	26.44	15.68	34.1
SKIN_HIST+GIST+BB_dimensions(Total)	34.156	13.59	25.96
GIST+BB_dimensions(Total)	33.485	17.941	36.984
SKIN_HIST+BB(ratio)	29.09	15.592	30.918
GIST+BB(Vectores)	32.266	18.12	34.67

Cuadro C.27: Resultados del experimento del nivel de manipulación habiendo fijado el primer nivel para la secuencia User_Ada_Byron-3.

	NN	SVM-RBF	SVM-Lineal
GIST	30.324	30.324	30.574
SKIN_HIST	38.22	38.22	43.984
SKIN_HIST+GIST	32.125	32.125	42.72
SKIN_HIST+GIST+BB_dimensions(Total)	30.633	30.633	10.488
GIST+BB_dimensions(Total)	29.633	29.633	6.0694
SKIN_HIST+BB(ratio)	37.97	37.97	43.1
GIST+BB(Vectores)	29.68	29.68	12.45

Cuadro C.28: Resultados del experimento utilizando un único nivel de etiquetas para la secuencia User_Ada_Byron-3.

C.5. Secuencia User_i3a-2

En esta sección veremos los resultados de los experimentos para la secuencia User_i3a-2. En la Figura C.4 vemos un resumen de esta secuencia en forma de mosaico.

Figura C.4: Resumen Secuencia User_i3a-2

	NN	SVM-RBF	SVM-Lineal	SVM-Sigmoidal	SVM-Polinómico
GIST	59.704	60.923	47.13	46.83	60.984
SKIN_HIST	77.28	78.454	78.48	26.043	57.79
SKIN_HIST+GIST	63.468	74.296	63.56	46.82	54.49
SKIN_HIST+GIST+ BB_dimensions	71.53	77.27	0	0	0
GIST+BB_dimensions	0	71.891	0	0	0

Cuadro C.29: Resultados del experimento inicial para la secuencia User_i3a-2.

	NN	SVM-RBF	SVM-Lineal
GIST	61.664	62.23	61.95
SKIN_HIST	80.735	83.3	81.11
SKIN_HIST+GIST	79.64	74.69	81.454
SKIN_HIST+GIST+BB_dimensions(Total)	72.17	74.62	37.93
GIST+BB_dimensions(Total)	71.36	68.45	45.07
SKIN_HIST+BB(ratio)	80.34	82.173	80.92
GIST+BB(Vectores)	71.61	68.344	54.055

Cuadro C.30: Resultados del experimento del primer nivel con la misma configuración en todos los niveles para la secuencia User_i3a-2.

C.5.1. Experimentos Iniciales

En la Tabla C.29 se muestran los resultados del experimento inicial.

C.5.2. Experimentos por niveles

En las Tablas C.30, C.32 y C.31 se muestran los resultados del experimento utilizando la misma configuración en los dos niveles.

C.5.3. Experimentos fijando nivel 1

En las Tablas C.34 y C.33 se muestran los resultados del experimento habiendo fijado el primer nivel con la configuración más óptima.

C.5.4. Experimentos 11 etiquetas

En la Tabla C.35 se muestran los resultados del experimento en el que no se utilizan niveles, solo un único nivel de 11 etiquetas.

	NN	SVM-RBF	SVM-Lineal
GIST	75.61	71.391	81.782
SKIN_HIST	77.66	82.62	77.86
SKIN_HIST+GIST	74.28	77.66	78.53
SKIN_HIST+GIST+BB_dimensions(Total)	79.97	79.44	40.94
GIST+BB_dimensions(Total)	79.827	78.437	77.13
SKIN_HIST+BB(ratio)	82.92	83.97	79.36
GIST+BB(Vectores)	79.718	78.33	34.43

Cuadro C.31: Resultados del experimento del nivel de no manipulación con la misma configuración en todos los niveles para la secuencia User_i3a-2.

	NN	SVM-RBF	SVM-Lineal
GIST	44.234	26.742	42.976
SKIN_HIST	10.059	15.97	15.42
SKIN_HIST+GIST	10.459	13.34	18.5
SKIN_HIST+GIST+BB_dimensions(Total)	25.83	14.01	18.54
GIST+BB_dimensions(Total)	20.992	18.399	35.56
SKIN_HIST+BB(ratio)	14.541	15.55	16.86
GIST+BB(Vectores)	23.832	18.61	30.817

Cuadro C.32: Resultados del experimento del nivel de manipulación con la misma configuración en todos los niveles para la secuencia User_i3a-2.

	NN	SVM-RBF	SVM-Lineal
GIST	58.77	67.12	73.359
SKIN_HIST	81.689	78.327	81.485
SKIN_HIST+GIST	63.836	67.83	75.032
SKIN_HIST+GIST+BB_dimensions(Total)	79.735	66.673	38.01
GIST+BB_dimensions(Total)	79.563	66.63	15.842
SKIN_HIST+BB(ratio)	77.77	77.28	77.83
GIST+BB(Vectores)	79.673	66.63	5.9493

Cuadro C.33: Resultados del experimento del nivel de no manipulación habiendo fijado el primer nivel para la secuencia User_i3a-2.

	NN	SVM-RBF	SVM-Lineal
GIST	40.156	24.67	41.36
SKIN_HIST	15.121	6.94	17.133
SKIN_HIST+GIST	14.32	4.21	19.172
SKIN_HIST+GIST+BB_dimensions(Total)	39.47	4.01	16.328
GIST+BB_dimensions(Total)	37.46	13.918	30.53
SKIN_HIST+BB(ratio)	13.8	6.2607	17.93
GIST+BB(Vectores)	33.977	14.002	13.322

Cuadro C.34: Resultados del experimento del nivel de manipulación habiendo fijado el primer nivel para la secuencia User_i3a-2.

	NN	SVM-RBF	SVM-Lineal
GIST	20.832	20.832	28.18
SKIN_HIST	30.832	30.832	37.75
SKIN_HIST+GIST	20.668	20.668	35.08
SKIN_HIST+GIST+BB_dimensions(Total)	20.59	20.59	14.7
GIST+BB_dimensions(Total)	23.02	23.02	7.5
SKIN_HIST+BB(ratio)	30.15	30.15	36.4
GIST+BB(Vectores)	22.96	22.96	5.5107

Cuadro C.35: Resultados del experimento utilizando un único nivel de etiquetas para la secuencia User_i3a-2.

C.6. Secuencia User_Ada_Byron-4

En esta sección veremos los resultados de los experimentos para la secuencia User_Ada_Byron-4. En la Figura C.5 vemos un resumen de esta secuencia en forma de mosaico.

Figura C.5: Resumen Secuencia User_Ada_Byron-4

C.6.1. Experimentos Iniciales

En la Tabla C.36 se muestran los resultados del experimento inicial.

C.6.2. Experimentos por niveles

En las Tablas C.37, C.39 y C.38 se muestran los resultados del experimento utilizando la misma configuración en los dos niveles.

C.6.3. Experimentos fijando nivel 1

En las Tablas C.41 y C.40 se muestran los resultados del experimento habiendo fijado el primer nivel con la configuración más óptima.

	NN	SVM-RBF	SVM-Lineal	SVM-Sigmoidal	SVM-Polinómico
GIST	58.37	45.875	56.274	61.1	44.29
SKIN_HIST	72.92	77.735	77.687	33.35	56.2
SKIN_HIST+GIST	59.13	76.015	71.204	61.1	45.23
SKIN_HIST+GIST+ BB_dimensions	71.485	78.609	0	0	0
GIST+BB_dimensions	0	81.156	0	0	0

Cuadro C.36: Resultados del experimento inicial para la secuencia User_Ada_Byron-4.

	NN	SVM-RBF	SVM-Lineal
GIST	56.85	65.16	63.375
SKIN_HIST	78.92	78.38	78.88
SKIN_HIST+GIST	76.89	78.109	78.61
SKIN_HIST+GIST+BB_dimensions(Total)	70.89	76.45	46.64
GIST+BB_dimensions(Total)	71.77	71.249	44.75
SKIN_HIST+BB(ratio)	75.22	78.08	76
GIST+BB(Vectores)	71.89	71.17	50.2

Cuadro C.37: Resultados del experimento del primer nivel con la misma configuración en todos los niveles para la secuencia User_Ada_Byron-4.

	NN	SVM-RBF	SVM-Lineal
GIST	77	80.77	83.09
SKIN_HIST	74.37	80	74.92
SKIN_HIST+GIST	76.032	79.327	76.12
SKIN_HIST+GIST+BB_dimensions(Total)	76.577	80.39	57.16
GIST+BB_dimensions(Total)	73.94	81.33	45.65
SKIN_HIST+BB(ratio)	79.437	79.84	79.05
GIST+BB(Vectores)	74.296	81.282	22.222

Cuadro C.38: Resultados del experimento del nivel de no manipulación con la misma configuración en todos los niveles para la secuencia User_Ada_Byron-4.

	NN	SVM-RBF	SVM-Lineal
GIST	40.406	34.5	27.08
SKIN_HIST	27.22	28.92	19.809
SKIN_HIST+GIST	28.71	28.996	22.23
SKIN_HIST+GIST+BB_dimensions(Total)	24.47	28.04	21.957
GIST+BB_dimensions(Total)	26.8	25.61	16.422
SKIN_HIST+BB(ratio)	28.054	28.98	20.46
GIST+BB(Vectores)	25.492	25.593	15.43

Cuadro C.39: Resultados del experimento del nivel de manipulación con la misma configuración en todos los niveles para la secuencia User_Ada_Byron-4.

	NN	SVM-RBF	SVM-Lineal
GIST	55.75	73.47	73.06
SKIN_HIST	77.52	74.94	76.751
SKIN_HIST+GIST	50.23	73.94	79.454
SKIN_HIST+GIST+BB_dimensions(Total)	76.47	71.001	13.668
GIST+BB_dimensions(Total)	76.58	70.19	13.62
SKIN_HIST+BB(ratio)	72.077	72.48	72.36
GIST+BB(Vectores)	76.64	70.19	2.76

Cuadro C.40: Resultados del experimento del nivel de no manipulación habiendo fijado el primer nivel para la secuencia User_Ada_Byron-4.

	NN	SVM-RBF	SVM-Lineal
GIST	27.54	30.97	34.133
SKIN_HIST	19.27	39.15	20.309
SKIN_HIST+GIST	22.39	39.836	27.278
SKIN_HIST+GIST+BB_dimensions(Total)	28.02	29.023	27.278
GIST+BB_dimensions(Total)	25.77	13.04	15.07
SKIN_HIST+BB(ratio)	21.87	39.76	21.39
GIST+BB(Vectores)	31.31	13.08	15.42

Cuadro C.41: Resultados del experimento del nivel de manipulación habiendo fijado el primer nivel para la secuencia User_Ada_Byron-4.

	NN	SVM-RBF	SVM-Lineal
GIST	33.39	33.39	27.26
SKIN_HIST	42.33	42.33	34.44
SKIN_HIST+GIST	42.125	42.125	36.375
SKIN_HIST+GIST+BB_dimensions(Total)	42.375	42.375	10.09
GIST+BB_dimensions(Total)	27.863	27.863	7.31
SKIN_HIST+BB(ratio)	42.55	42.55	32.83
GIST+BB(Vectores)	27.92	27.92	6.8007

Cuadro C.42: Resultados del experimento utilizando un único nivel de etiquetas para la secuencia User_Ada_Byron-4.

C.6.4. Experimentos 11 etiquetas

En la Tabla C.42 se muestran los resultados del experimento en el que no se utilizan niveles, solo un único nivel de 11 etiquetas.