

UNIVERSIDAD AUTÓNOMA DE MADRID  
ESCUELA POLITÉCNICA SUPERIOR



# Verificación de vehículos mediante técnicas de visión artificial

**-PROYECTO FIN DE CARRERA-**

Gonzalo Ballesteros Villarreal  
Julio 2014



# Verificación de vehículos mediante técnicas de visión artificial

**Autor: Gonzalo Ballesteros Villarreal**

**Tutor: Luis Salgado Álvarez de Sotomayor**

**Ponente: Jesús Bescós Cano**

email: gonzalo.ballester@estudiante.uam.es, luis.salgadoa@uam.es, j.bescos@uam.es



**Video Processing and Understanding Lab**  
**Departamento de Tecnología Electrónica y de las Comunicaciones**  
**Escuela Politécnica Superior**  
**Universidad Autónoma de Madrid**  
**Julio 2014**



## Resumen

En este trabajo, se proponen sistemas de verificación de vehículos mediante métodos basados en aprendizaje.

En primer lugar se realiza un estudio del estado del arte para conocer los problemas actuales en la materia. Después, se muestra la arquitectura de los sistemas que se divide en dos etapas: extracción de características y clasificación. En la primera etapa se realiza una breve exposición de los tipos de características que se van a implementar (simetría, bordes, análisis de componentes principales (PCA) e histogramas de gradientes orientados (HOG)). La etapa de clasificación consiste en una explicación teórica de los clasificadores utilizados en nuestro sistema.

Posteriormente, se realiza el desarrollo de estos sistemas, efectuando mejoras para cada uno de ellos. Para el sistema basado en simetría se plantean dos métodos diferentes, introduciéndose una mejora en el segundo método, que consiste en una diferenciación entre ejes compuestos por uno y dos píxeles, junto con una penalización en los valores de simetría para conseguir una mayor diferenciación entre las clases. Respecto al sistema basado en bordes, se utilizan únicamente bordes verticales, donde se analiza el uso de vectores reducidos. Por otra parte, se presenta el uso de la matriz de correlaciones para desarrollar el sistema basado en PCA. En el sistema basado en HOG se estudia qué parámetros son los adecuados para el descriptor en el caso particular de vehículos, proponiéndose descriptores eficientes basados en esta configuración, que pueden ser implementados en sistemas en tiempo real.

Finalmente, con los resultados obtenidos en el paso previo se procede a un análisis para los distintos métodos presentando sus principales características y limitaciones.

## Palabras clave

Verificación de vehículos, extracción de características, clasificación, simetría, bordes, análisis de componentes principales, histogramas de gradientes orientados, máquinas de vectores soporte.

## **Abstract**

In this work, a vehicle verification systems using learning methods are proposed.

First, a study of related work has been done. Afterwards, the architecture of these systems is explained. The architecture is divided in two stages: feature extraction and clasification. In the first stage, a brief summary of the different features that will be implemented (simmetry, edges, principal components analysis (PCA) and histograms of oriented gradients (HOG)) is given. The second stage is a theoretical explanation of the classifiers used in this system.

Subsequently, the systems are developed with new improvements. Two different methods are proposed for the system based on symmetry. An improvement is introduced for the second method that is a differentiation between compounds axes by one and two pixels, also a penalty is introduced into the values of symmetry for greater differentiation between classes. Regarding the system based on edges, vertical edges are used, where the performance reducing the size of the vectors is analyzed. Moreover, the correlation matrix is used to develop the system based on PCA. In the system based on HOG, in the particular case of vehicles, appropriate parameters for the descriptor are studied, proposing efficient descriptors based on this configuration that can be implemented in real-time systems.

Finally, the results obtained in the previous step are analyzed for each of the methods, and their main characteristics and limitations are described.

## **Keywords**

Vehicle verification, feature extraction, clasification, symmetry, edges, principal components analysis, histograms of oriented gradients, support vector machine.

# Agradecimientos

En primer lugar, quiero agradecer a mi tutor, Luis Salgado, la atención y paciencia que ha tenido conmigo durante la realización de este PFC. Gracias por la comprensión de mis problemas y el apoyo dado todo este tiempo.

Agradecer también la ayuda prestada de los compañeros del VPU, ha sido un placer conocerlos.

Quiero dar las gracias a mis padres y mi hermana por su apoyo durante toda la carrera y por haber creído en mí, ayudándome tanto en los malos como en los buenos momentos. Un recuerdo también para mis abuelos (los que están y los que ya no) que han sido muy importantes para poder llegar a éste momento. También quiero acordarme de mis tíos, especialmente de mi tía, Ana María, con la que he pasado muy buenos momentos.

Quiero hacer especial mención a mi pareja Natalia, sin ella esto no hubiese sido posible.

También agradecer a mis compañeros y amigos de clase Cecilia, Guille, JP, Dieguete y Xarlos por compartir este viaje conmigo haciéndolo menos duro. Espero seguir pasando más buenos momentos juntos.

Por último y no menos importante, agradecer a todos mis amigos todos los ratos juntos que hemos pasado y pasaremos, por estar allí cuando lo necesitas y por todas esas historias que nos hacen reír.

Gracias a todos.

Gonzalo Ballesteros Villarreal

Julio 2014





# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	2
1.3. Estructura del documento . . . . .	2
<b>2. Estado del arte</b>	<b>3</b>
2.1. Introducción . . . . .	3
2.2. Detección de vehículos . . . . .	5
2.3. Métodos basados en visión estéreo . . . . .	8
2.4. Métodos basados en movimiento . . . . .	10
2.5. Métodos basados en apariencia . . . . .	12
2.5.1. Métodos basados en modelos . . . . .	12
2.5.2. Métodos basados en el aprendizaje . . . . .	13
2.5.3. Extracción de características . . . . .	14
2.5.4. Clasificación . . . . .	24
2.6. Conclusiones . . . . .	26
<b>3. Arquitectura del sistema</b>	<b>27</b>
3.1. Introducción . . . . .	27
3.2. Extracción de característica explícitas . . . . .	28
3.2.1. Simetría . . . . .	28
3.2.2. Bordes . . . . .	29
3.3. Extracción de característica implícitas . . . . .	31
3.3.1. Análisis de componentes principales . . . . .	31
3.3.2. Histogramas de gradientes orientados . . . . .	32
3.4. Clasificación . . . . .	34
3.4.1. Clasificador lineal . . . . .	34
3.4.2. Clasificador bayesiano . . . . .	35
3.4.3. SVM . . . . .	37

<b>4. Desarrollo</b>	<b>47</b>
4.1. Introducción . . . . .	47
4.2. Base de Datos y método de clasificación . . . . .	47
4.3. Desarrollo del sistema basado en simetría . . . . .	49
4.3.1. Simetría . . . . .	49
4.3.2. Método 1 . . . . .	49
4.3.3. Método 2 . . . . .	52
4.3.4. Clasificación . . . . .	55
4.4. Desarrollo del sistema basado en bordes . . . . .	59
4.4.1. Extracción de gradientes de bordes . . . . .	59
4.4.2. Clasificación . . . . .	61
4.5. Desarrollo del sistema basado en componentes principales . . . . .	69
4.5.1. Análisis de Componentes Principales . . . . .	69
4.6. Desarrollo del sistema basado en Histogramas de Gradientes Orientados . . . . .	72
4.6.1. SHOG . . . . .	72
4.6.2. OHOG . . . . .	74
4.6.3. PCA sobre OHOG . . . . .	80
4.6.4. Descriptores eficientes basados en OHOG . . . . .	83
<b>5. Conclusiones y trabajo futuro</b>	<b>89</b>
5.1. Conclusiones . . . . .	89
5.2. Trabajo Futuro . . . . .	90
<b>Bibliografía</b>	<b>93</b>
<b>Appendix.</b>	<b>102</b>
<b>A. Resultados</b>	<b>103</b>
A.1. Simetría . . . . .	103
A.1.1. Método 1 . . . . .	103
A.1.2. Método 2 sin mejora . . . . .	104
A.1.3. Método 2 con mejora . . . . .	105
A.2. Bordes . . . . .	106
A.2.1. Bordes completos . . . . .	106
A.2.2. Bordes 32 bins . . . . .	107
A.2.3. Bordes 16 bins . . . . .	108
A.2.4. Bordes 8 bins . . . . .	109
A.2.5. Bordes 4 bins . . . . .	110
A.2.6. Bordes máximos (2 valores) . . . . .	111

A.3. PCA . . . . .	112
A.3.1. Imagen en nivel de gris . . . . .	112
A.3.2. Imagen en el espacio de color RGB . . . . .	113
A.4. HOG . . . . .	114
A.4.1. SHOG . . . . .	114
A.4.2. OHOG . . . . .	116
A.4.3. PCA-OHOG . . . . .	124
A.4.4. VOHOG . . . . .	124
A.4.5. T-OHOG . . . . .	126
<b>B. Publicaciones</b>	<b>127</b>
<b>C. Presupuesto</b>	<b>141</b>
<b>D. Pliego de condiciones</b>	<b>143</b>



# Índice de figuras

2.1. Ejemplos de la detección de vehículos. [1] . . . . .	5
2.2. (a)(b) Variaciones en la apariencia de los vehículos (color, forma,...), efectos de perspectiva producidos por la colocación de la cámara y diferentes tipos de vehículos (coches, furgonetas, camiones,...). También se puede observar los efectos producidos por cambios de: (c) iluminación y (d) condiciones metereológicas [1]. . . . .	6
2.3. Ejemplos de las fases de la detección de vehículos. (a) Generación de hipótesis, podemos observar como en este caso se produce una falsa detección, la cual se intentará clasificar correctamente en la fase de verificación. (b) Verificación de hipótesis, en esta etapa se clasifican los dos candidatos obtenidos en sus respectivas clases. . . . .	7
2.4. Geometría de mapeado de perspectiva [2]. . . . .	10
2.5. Ejemplo de las luces de los vehículos. En la imagen se muestra en primer lugar las luces de cruce de un vehículo en condiciones de baja iluminación. La segunda imagen muestra las luces de freno en escenario diurno. La tercera imagen consta de un vehículo con las luces de marcha atrás encendidas. Por último se presenta una imagen de un vehículo con la luz de intermitencia encendida. . . . .	14
2.6. Ejemplo de las sombras provocadas por los vehículos sobre la carretera. . . . .	15
2.7. Ejemplo de los diferentes colores que nos podemos encontrar en los vehículos. . . . .	16
2.8. Ejemplo de la estructura rectangular de los vehículos. . . . .	17
2.9. Ejemplo de la característica de simetría en vehículos. . . . .	17
2.10. Ejemplo de extracción de bordes laterales. . . . .	19
2.11. Transformación de la imagen a visión aérea [3]. (a) Imagen original. (b) Imagen aérea virtual. . . . .	20
3.1. Esquema general del sistema implementado. . . . .	28
3.2. Ejemplo de los pasos seguidos en la extracción de bordes verticales. (a) Imagen original de la que partimos. (b) Imagen en nivel de gris. (c) Imagen de bordes verticales aplicando binarización. . . . .	30
3.3. Sistema de obtención del descriptor HOG . . . . .	32

3.4.	Ejemplo de mallado de la imagen. (a) Configuración R-HOG. Donde se muestran los parámetros número de pixel por celda, $s_1 \times s_2$ , número de celdas en la imagen, $\eta_1 \times \eta_2$ y tamaño de bloque, $b_1 \times b_2$ . (b) Configuración C-HOG . . . . .	33
3.5.	Posibles configuraciones cambiando el número de bins de orientación. El rango utilizado es $[0, \pi]$ el cuál se divide uniformemente en $\beta$ intervalos. (a) 8 (b) 16 (c) 32 bins de orientación. . . . .	33
3.6.	Ejemplo de funcionamiento del clasificador lineal basado en el algoritmo de Perceptrón [4]. Las imágenes muestran los datos de dos clases (roja y azul) en el espacio de características de dos dimensiones $(\phi_1, \phi_2)$ . (a) Muestra los parámetros iniciales del vector de pesos $\omega$ , ilustrados por una flecha negra junto con la correspondiente frontera de decisión (línea negra), donde la flecha apunta hacia la región de decisión que clasifica como perteneciente a la clase roja. El dato indicado mediante un círculo verde está mal clasificado y por lo tanto su vector de características se añade al vector de pesos actual, dando como nueva frontera de decisión la mostrada en (b). En (c) se muestra la siguiente muestra mal clasificada indicada por el círculo verde, y este nuevo vector de características es de nuevo añadido al vector de pesos dando la frontera de decisión mostrada en (d), para la cuál se clasifican todas las muestras correctamente. . . . .	36
3.7.	Ejemplo de cambio del espacio original de los datos de entrada a un nuevo espacio de característica, de una SVM no linealmente separable realizado mediante una función kernel [5]. . . . .	38
3.8.	El margen se define como la distancia perpendicular entre la frontera de decisión y el punto de los datos más cercano. (a) muestra como maximizar el margen y conduce a una elección particular del límite de decisión: la ubicación de este límite está determinada por un subconjunto de puntos conocidos como vectores de soporte, que en (b) están indicados mediante círculos. . . . .	39
3.9.	Ejemplo de síntesis de datos de dos clases para dos dimensiones que muestran los contornos de la constante $y(x)$ obtenidos a partir de una SVM con una función kernel gaussiana[4]. También se muestran los límites de decisión, los límites de los márgenes, y los vectores soporte. . . . .	41
3.10.	Ejemplo del efecto que provoca la constante C en la SVM con margen blando. Un valor más bajo de C, provoca un margen más amplio y más permisivo, mientras que valores más altos de C reducen el margen haciendo que este sea más restrictivo [6]. . . . .	42
3.11.	SVM con margen blando. Ejemplo de la clasificación de los datos en clases solapadas [4]. . . . .	43

3.12. Ejemplos de la generación de la frontera de decisión usando en primer lugar un kernel polinomial cambiando el grado del polinomio (parámetro $P$ ) y en segundo lugar un núcleo gaussiano cambiando su parámetro de amplitud (parámetro $\gamma$ ) [6].	45
4.1. Ejemplo Base de datos GTI. (a) Vista frontal; (b) Vista lateral izquierda; (c) Vista lateral derecha; (d) Vista lejana. . . . .	48
4.2. Elección del eje dependiendo del ancho de ventana. . . . .	53
4.3. Representación de las distribuciones de los valores de simetría de las clases vehículo y no-vehículo. (a) Eje constituido por un solo píxel en todos los casos (b) Eje constituido por uno o dos píxeles. . . . .	53
4.4. Sigmoide utilizada para la penalización del valor de simetría. . . . .	54
4.5. Representación de las distribuciones de simetría para cada una de las clases haciendo uso de la penalización mediante sigmoide. . . . .	55
4.6. Rendimiento del sistema de simetría haciendo uso del método 1 para cada una de las vistas, utilizando diferentes clasificadores. . . . .	55
4.7. Rendimiento del sistema de simetría haciendo uso del método 2 para cada una de las vistas, utilizando diferentes clasificadores. (a) Método 2 sin aplicar la penalización. (b) Método 2 aplicando la penalización. . . . .	57
4.8. Comparación entre los métodos de extracción de simetría. . . . .	58
4.9. Distribuciones de los bordes verticales en las clases vehículo y no vehículo para los métodos de extracción de bordes sin binarización y con binarización. . . . .	60
4.10. Ejemplo de la agrupación de los valores del vector completo de bordes de la imagen en bins. En este caso se pasa a un vector compuesto por 16 valores o bins. . . . .	60
4.11. Rendimiento de los diferentes clasificadores sobre el vector de 64 componentes. (a) Tasa de aciertos (b) Recall. . . . .	61
4.12. Rendimiento de los diferentes clasificadores sobre el vector de 64 componentes extraídas de la imagen binarizada. (a) Tasa de aciertos (b) Recall. . . . .	62
4.13. Rendimiento de los diferentes clasificadores sobre el vector de 32 bins. (a) Tasa de aciertos (b) Recall. . . . .	63
4.14. Rendimiento de los diferentes clasificadores sobre el vector de 16 bins. (a) Tasa de aciertos (b) Recall. . . . .	63
4.15. Rendimiento de los diferentes clasificadores sobre el vector de 8 bins. (a) Tasa de aciertos (b) Recall. . . . .	64
4.16. Rendimiento de los diferentes clasificadores sobre el vector de 4 bins. (a) Tasa de aciertos (b) Recall. . . . .	65
4.17. Evolución de los clasificadores SVM no lineal y Bayesiano al ir reduciendo la dimensionalidad del vector de características. . . . .	65

4.18. Comparación entre el rendimiento de las diferentes configuraciones del vector de bordes. (a) Tasa de aciertos (b) Recall. . . . .	66
4.19. Rendimiento de los diferentes clasificadores sobre el vector de máximos sin binarización. (a) Tasa de aciertos (b) Recall. . . . .	67
4.20. Rendimiento de los diferentes clasificadores sobre el vector de máximos con binarización. (a) Tasa de aciertos (b) Recall. . . . .	67
4.21. Comparación entre el rendimiento de las diferentes configuraciones del vector de bordes. (a) Tasa de aciertos (b) Recall. . . . .	68
4.22. Ejemplo de la interpolación de los gradientes en el descriptor HOG. . . . .	73
4.23. Configuraciones posibles del descriptor HOG cambiando el rango de los ángulos. Esta imagen es un ejemplo de la configuración para $\beta = 8$ . El rango de los ángulos esta dividido en $\beta$ secciones distribuidas uniformemente. (a) Configuración del rango $[0, \pi]$ . (b) Configuración del rango $[-\pi, \pi]$ . . . . .	76
4.24. Ejemplo de la eliminación de la interpolación de los gradientes en el descriptor HOG. . . . .	76
4.25. Comparación entre los descriptores SHOG y OHOG. . . . .	80
4.26. Evolución del rendimiento para cada vista. (a) en función del rendimiento y el porcentaje de varianza retenido; (b) en función del número de componentes y el porcentaje de varianza retenido. . . . .	82
4.27. Configuración vertical de las celdas. . . . .	83
4.28. Evolución del rendimiento en clasificación eliminando las celdas menos significativas del descriptor para cada una de las vistas. En gris se presentan las celdas eliminadas del vector. Los parámetros escogidos para llevar a cabo este estudio son $\eta = 4$ y $\beta = 16$ . . . . .	86
4.29. Máscaras propuestas para cada una de las vistas de la base de datos. (a) Frontal (b) Izquierda (c) Derecha (d) Lejana. Donde solo se tienen en cuenta las celdas de color blanco. . . . .	87
4.30. Hipótesis con (a) vista parcial, (b) bajo contraste, y (c) distorsión. . . . .	88



# Índice de cuadros

4.1. Comparación clasificador bayesiano y SVM no lineal. . . . .	56
4.2. Comparación entre los diferentes clasificadores para el método 2 con y sin penalización. . . . .	57
4.3. Comparación de nuestro sistema con la propuesta [7]. . . . .	59
4.4. Comparación entre los métodos de extracción de bordes mediante la imagen sin binarizar y la imagen binarizada. . . . .	62
4.5. Rendimiento de PCA para la imagen en nivel de gris y en el espacio de color RGB. . . . .	70
4.6. Rendimiento para diferente dimensionalidad del subespacio principal. . . . .	71
4.7. Comparación de nuestro sistema basado en PCA con otros encontrados en la literatura. . . . .	72
4.8. Tasa de acierto del descriptor SHOG. . . . .	73
4.9. Porcentaje de aciertos en la verificación usando diferentes operadores. . . . .	74
4.10. Porcentaje de aciertos en la verificación usando diferentes rangos para los ángulos. . . . .	75
4.11. Porcentaje de aciertos en la verificación usando diferentes métodos para el cálculo de histogramas. . . . .	77
4.12. Porcentaje de aciertos en la verificación usando diferentes tamaños de bloque. . . . .	78
4.13. Porcentaje de aciertos en la verificación usando diferentes métodos de normalización. . . . .	78
4.14. Porcentaje de aciertos en la verificación usando diferentes clasificadores. . . . .	79
4.15. Tasa de aciertos y número de componentes para PCA-OHOG con un 90% de varianza retenida. . . . .	81
4.16. Rendimiento del sistema PCA-OHOG cambiando el porcentaje de varianza retenida. . . . .	81
4.17. Rendimiento del sistema PCA-OHOG cambiando la dimensionalidad del espacio de características principal. . . . .	82
4.18. Rendimiento del descriptor V-OHOG. . . . .	84

4.19. Comparación entre los diferentes descriptores basados en HOG. (a) Comparativa de los mejores resultados; (b) Resultados de los métodos OHOG y V-OHOG correspondientes a valores similares del número de componentes de los mejores resultados de V-HOG. . . . .	85
4.20. Comparación entre los descriptores eficientes propuestos. . . . .	87

# Acrónimos

<b>HOG</b>	<i>Histograms of Oriented Gradients</i>
<b>PCA</b>	<i>Principal Component Analysis</i>
<b>SVM</b>	<i>Support Vector Machine: A set of related supervised learning methods used for classification and regression</i>
<b>IPM</b>	<i>Inverse Perspective Mapping</i>
<b>LDA</b>	<i>Linear Discriminant Analysis</i>
<b>NN</b>	<i>Neural Networks</i>
<b>MoG</b>	<i>Mixture of Gaussians</i>
<b>EM</b>	<i>Expectation-Maximization</i>
<b>ICA</b>	<i>Independent Component Analysis</i>
<b>SRM</b>	<i>Structure Risk Minimization</i>

# Capítulo 1

## Introducción

### 1.1. Motivación

Los problemas de tráfico, como la movilidad, seguridad y el consumo eficiente, entre otros, han sido siempre una preocupación entre la población. Esto ha generado que en los últimos años aumente el interés por disminuir o eliminar estos problemas, surgiendo así los Sistemas Inteligentes de Transporte (ITS). Lo que se pretende desde la comunidad de investigación ITS es conseguir que los sistemas incorporados en los vehículos (sensores de diferente naturaleza) sean capaces de reconocer el entorno y detectar entre otras cosas, las posibles amenazas que se puedan encontrar, como por ejemplo la posibilidad de un choque. Adicionalmente se pretende que estos sistemas se puedan comunicar entre ellos mediante una infraestructura vial inteligente y que a su vez sean capaces de recopilar información de las carreteras y monitorizar el tráfico. Centrándonos en sistemas que intentan detectar la posibilidad de accidentes de tráfico, la detección de vehículos se presenta como el elemento fundamental en la cadena de procesamiento. Para ello la utilización de sensores pasivos, en particular de cámaras de vídeo, se presentan como una buena elección, ya que no son elementos intrusivos y nos ofrecen un mejor coste y flexibilidad que los sensores activos. Además, a través del análisis de información visual, se puede obtener una comprensión mucho más amplia de la escena y así utilizar estos sistemas a parte de para la detección de vehículos, para otras funciones como podrían ser detección de carril, detección de señales de tráfico o reconocimiento de objetos entre otros. Los primeros pasos hacia el desarrollo de sistemas de ayuda a la conducción basados en visión aparecen en Europa en los años 80, con el proyecto PROMETHEUS. Proyectos análogos se pusieron en marcha en EEUU y Japón. En el año 2002 se consiguieron algunas soluciones y prototipos preliminares para la detección de carril y de vehículo. Bertozzi et al. en [8] resumieron el estado de arte de la visión artificial en escenarios de tráfico. Más tarde, en 2006, Sun et al. en [2] hicieron un excelente estudio sobre el estado del arte de la detección de vehículos. Sin embargo los estudios existentes, siguen sin poder solucionar toda la problemática debido a la fuerte variabilidad del entorno, por lo que actualmente sigue

siendo un tema de investigación activo.

Por lo tanto, la motivación principal de este trabajo es la contribución a este campo realizando un estudio de los diferentes métodos que existen en la literatura para la etapa de verificación de hipótesis y proponiendo mejoras que consigan un sistema robusto, efectivo y eficiente.

## 1.2. Objetivos

El objetivo de este PFC es el estudio de los principales métodos utilizados para la verificación de hipótesis y la mejora de estos sistemas existentes en el actual estado del arte. El objetivo inicial se divide en los siguientes sub-objetivos:

1. Estudio del estado del arte actual: En esta etapa se van a estudiar las diversas propuestas que se encuentran para la verificación de vehículos, realizando una selección de los métodos que se van a implementar.
2. Desarrollo de los métodos seleccionados: Esta etapa engloba la implementación de los algoritmos y métodos existentes. Llevando a cabo mejoras sobre estos métodos y buscando la mejor elección para conseguir una verificación de hipótesis exitosa.

## 1.3. Estructura del documento

La estructura de este documento es la siguiente:

- Capítulo 1. Introducción, motivación del proyecto y objetivos.
- Capítulo 2. Estudio del estado del arte para la detección de vehículos en carretera, centrándonos en la verificación de hipótesis.
- Capítulo 3. Arquitectura del sistema.
- Capítulo 4. Desarrollo de los descriptores seleccionados para la verificación de vehículos.
- Capítulo 5. Conclusiones y trabajos futuros.

## Capítulo 2

# Estado del arte

Este capítulo ofrece una visión de trabajos previos que se encuentran en el campo del estudio presentado en este documento. En las siguientes secciones, vamos a describir el área de la detección y verificación de vehículos, introduciendo el problema (sección 2.1), planteando una visión general del sistema del que partimos (sección 2.2) y realizando una clasificación de los métodos que encontramos en la literatura: métodos basados en visión estereoscópica (sección 2.3), métodos basados en movimiento (sección 2.4) y métodos basados en la apariencia (sección 2.5). Por último se presentan unas conclusiones (sección 2.6) que concretan el trabajo llevado a cabo en este PFC.

### 2.1. Introducción

Muchos son los problemas que encontramos en el área de la automoción, como pueden ser el consumo eficiente, la seguridad o la movilidad entre otros, por lo tanto el estudio de como poder mejorar este campo ha incrementado en los últimos años. En este contexto, la comunidad de Sistemas de Transporte Inteligente (ITS) tiene como objetivo solventar o conseguir solucionar estos problemas. Los trabajos llevados a cabo por esta comunidad se basan en sistemas integrados en los vehículos que puedan llevar a cabo un conocimiento del medio y sean capaces de proporcionar información relevante de los factores que los rodean, comunicándose entre ellos y así creando una infraestructura inteligente en carretera, que permita la recolección de datos del tráfico sobre áreas mucho mayores [1].

La detección de vehículos es una parte importante de estos sistemas, la cual se centra en aspectos relacionados con la seguridad en carretera. Se pretende que a través de sensores los vehículos sean capaces de detectar y evitar accidentes de tráfico. La elección de dichos sensores, ha propiciado un debate en cuanto al uso de sensores pasivos o activos en sistemas de detección en carretera. Los sensores activos agrupan sensores como láseres o radares de onda milimétrica, los cuales son capaces de detectar objetos, estimando la distancia hasta dichos objetos, mediante

el cálculo de tiempos entre la señal emitida y la señal reflejada que se produce al impactar contra el objeto. Las limitaciones que nos encontramos en este tipo de sensores es la baja resolución espacial y la baja velocidad de escaneo. Sin embargo, tienen un comportamiento robusto en situaciones de baja visibilidad, teniendo mayor precisión en situaciones de lluvia o niebla los radares de onda milimétrica que los sensores basados en láseres, pero con el inconveniente de ser más caros. Muchas son las aproximaciones existentes que se han llevado a cabo mediante sensores pasivos tales como vídeo cámaras. Estos sensores tienen la ventaja frente a los sensores activos de poder adquirir los datos de forma no intrusiva. Por lo tanto, también se debe tener en cuenta que los sensores activos presentan el problema de que su uso como una infraestructura de sistemas de asistencia en carretera, puede generar problemas de interferencias entre ellos. Por otro lado, los sensores pasivos nos ofrecen una perspectiva más amplia de la escena, pudiendo usar estos sistemas para otras funciones, tales como, detección de las líneas laterales del carril, detección de señales de tráfico y reconocimiento de objetos. Por lo tanto, este tipo de sensores nos ofrecen una mayor flexibilidad y un bajo coste. Sin embargo, presentan como desventaja principal su bajo rendimiento en condiciones críticas del medio se hace patente, como son escenarios de niebla espesa o nocturnos donde la carretera no esté bien iluminada, en los cuales los sensores activos tienen un comportamiento mejor. Una solución a estos problemas es la fusión con otro tipo de sensores.

En los últimos años, la detección de vehículos ha sido ampliamente estudiada, encontrando soluciones que hacen uso de visión estéreo [9], y soluciones centradas en visión monocular [1]. La ventaja de la visión estéreo es que permite la reconstrucción métrica de la escena. Si bien, implica un difícil reto con respecto a la calibración de la cámara o la sincronización entre imágenes. Esto se convierte en un problema teniendo en cuenta la inestabilidad y la vibración que produce la alta velocidad de los vehículos. A pesar de estos problemas los sistemas estéreos están teniendo un ligero crecimiento debido a los avances que se están produciendo en los últimos años en tecnología, por lo que se está haciendo posible que la implantación de estos sistemas sea viable. Las aproximaciones llevadas a cabo mediante sistemas monoculares ofrecen por otro lado, un mejor equilibrio entre la fiabilidad, efectividad y flexibilidad, por lo que en los últimos años ha sido el más utilizado.

Este PFC investiga la detección de vehículos basados en visión monocular. El rendimiento obtenido por estos sistemas si bien alcanza niveles altos de detección, todavía está lejos de lo que pretende obtener la comunidad ITS, por lo que sigue siendo un campo abierto a la investigación. Este PFC busca profundizar en los métodos existentes para la detección de vehículos basados en su apariencia y así llevar a cabo una correcta clasificación entre las dos posibles clases que encontramos en este problema, vehículos y no-vehículos.



Figura 2.1: Ejemplos de la detección de vehículos. [1]

## 2.2. Detección de vehículos

Los sistemas de detección de vehículos en carretera ponen su atención en secuencias de vídeo de escenas de tráfico. La finalidad de estos sistemas es la detección de vehículos mediante una cámara de vídeo instalada en el vehículo en el cuál nos encontramos. Las imágenes obtenidas por esta cámara son procesadas y se determina si se encuentran vehículos en la imagen y cuál es su posición. La Figura 2.1 muestra un ejemplo de ello.

Los problemas más importantes a los que se enfrentan estos sistemas son [1]:

- Pérdida de información debido al cambio de secuencias en 3D a imágenes en 2D: el proceso de adquisición de las imágenes implica una transformación del espacio 3D al espacio 2D de la imagen. Este efecto distribuye la información de forma no homogénea en la imagen, por lo que vehículos del mismo tamaño aparecen con diferentes dimensiones en las imágenes según su posición relativa a la cámara.
- Movimiento de la cámara: el movimiento de la cámara causa que elementos que son estáticos como puede ser el fondo o la carretera, aparezcan con movimiento en la secuencia. El reto es conseguir eliminar ese movimiento creado por el movimiento de la cámara.
- Variación intrínseca del vehículo: aunque los vehículos suelen tener una estructura similar, muchos factores como pueden ser: color, forma, tamaño, entre otros, presentan una alta variabilidad.
- Cambio en los patrones de apariencia: a parte de los factores comentados anteriormente, la apariencia tanto del fondo como la del vehículo pueden verse seriamente afectados por la variabilidad del medio, sobretodo cuando se producen cambios en las condiciones climáticas o en la iluminación.
- Procesamiento en tiempo real: estos sistemas requieren que la detección de vehículos se



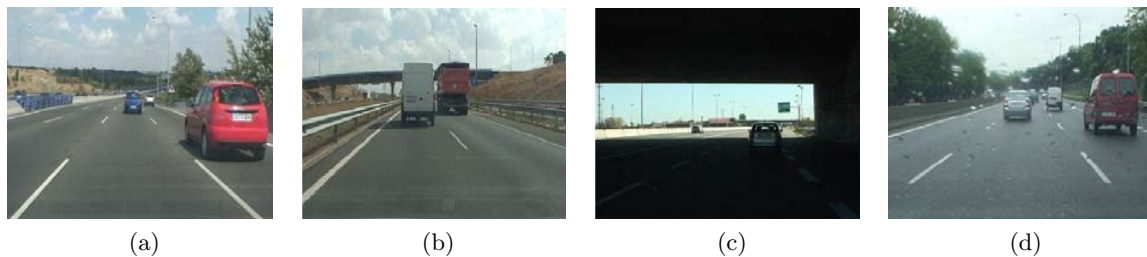


Figura 2.2: (a)(b) Variaciones en la apariencia de los vehículos (color, forma,...), efectos de perspectiva producidos por la colocación de la cámara y diferentes tipos de vehículos (coches, furgonetas, camiones,...). También se puede observar los efectos producidos por cambios de: (c) iluminación y (d) condiciones meteorológicas [1].

realice en tiempo real. Para lograr este objetivo se tiene que considerar el tiempo de cómputo de los algoritmos utilizados.

Algunos de estos problemas los podemos observar en la Figura 2.2.

Como el proceso de detección de vehículos es muy costoso, la metodología planteada en algunos trabajos de la literatura para los sistemas de detección de vehículos, [10, 3, 2], consiste en dividir la detección en dos etapas: generación de hipótesis (GH) y verificación de hipótesis (VH). La primera etapa consiste de una rápida búsqueda en la imagen de regiones que potencialmente puedan contener un vehículo, hipótesis, como se muestra en la Figura 2.3a. En la siguiente etapa de VH, las hipótesis obtenidas en la etapa anterior son verificadas. Típicamente esta verificación se realiza mediante el uso de características relacionadas con la apariencia de los vehículos. La Figura 2.3b muestra un ejemplo de los candidatos a verificar.

**Generación de hipótesis** La etapa de generación de hipótesis realiza un procesamiento rápido de la secuencia de entrada obteniendo regiones con alta probabilidad de que contengan vehículos. Para la generación de estas hipótesis encontramos que los diferentes métodos se pueden agrupar en tres categorías [2]:

- Métodos basados en el conocimiento: Se basan en la aplicación del conocimiento previo de la hipótesis, por lo tanto se busca el seguimiento de características específicas presentes en los vehículos y su entorno (sombras, simetría [11], bordes, texturas, color, luces de los vehículos [12], etc.).
- Métodos basados en visión artificial estereoscópica [13]: Consisten en procedimientos para la obtención de la forma de los objetos en la escena. La forma se determina mediante la distancia desde el objeto respecto a un sistema de referencia, típicamente estos métodos se realizan con el uso de sistemas compuestos por dos cámaras separadas, con el objetivo



Figura 2.3: Ejemplos de las fases de la detección de vehículos. (a) Generación de hipótesis, podemos observar como en este caso se produce una falsa detección, la cual se intentará clasificar correctamente en la fase de verificación. (b) Verificación de hipótesis, en esta etapa se clasifican los dos candidatos obtenidos en sus respectivas clases.

de obtener una imagen en tres dimensiones. Las aproximaciones utilizadas en la literatura son, mapas de disparidad y mapeado de perspectiva inversa.

- Métodos basados en el análisis del movimiento [14]: Los métodos discutidos anteriormente abordan el problema de la separación entre objeto y fondo, sin embargo para la generación de hipótesis también existen métodos basados en el movimiento de los objetos. Estos métodos se basan en la distinción entre vehículos y no-vehículos mediante el estudio del movimiento de los elementos en la imágenes extraídas de las secuencias de vídeo.

**Verificación de hipótesis** La etapa de verificación de hipótesis tiene como objetivo la verificación de las regiones obtenidas en la etapa anterior. Para la verificación de hipótesis encontramos dos categorías principales [2]:

- Métodos basados en modelos [15]: Utilizan modelos predefinidos de vehículos. Esta aproximación se basa en el hecho de que los vehículos son prácticamente poliedros, por lo tanto estos métodos recurren a características básicas (bordes, simetría,...) para la descripción de modelos.
- Métodos basados en aprendizaje [16]: Estos métodos se basan en la clasificación de las hipótesis de los vehículos mediante el aprendizaje, a partir de un conjunto de entrenamiento.

No obstante, una parte importante de los trabajos encontrados en la literatura no distingue entre estas dos etapas de los sistemas de detección de vehículos, sino los aborda como una única

etapa donde a partir de la secuencia de vídeo se extraen y clasifican las imágenes en vehículos y no vehículos. Por lo tanto, en las siguientes secciones se van a agrupar los métodos existentes en la literatura en: métodos basados en visión estéreo, métodos basados en movimiento y métodos basados en la apariencia de los vehículos. A partir de la información extraída mediante estos métodos se procede a una fase de clasificación, donde se determinan finalmente los vehículos que aparecen en la escena.

## 2.3. Métodos basados en visión estéreo

En el proceso de obtención de la escena para la detección de vehículos se produce la pérdida de la tridimensionalidad cuando usamos métodos que se basan en visión monocular. Estos métodos, por lo tanto, buscan la obtención de esa tercera dimensión para, en la medida de lo posible, obtener la forma de los objetos. Los sistemas estereoscópicos utilizan al menos dos imágenes distintas de la escena para obtener la información de profundidad. A partir del contenido de estas imágenes se puede determinar la distancia a la que se encuentra el objeto del observador. La captura de estas imágenes se puede realizar con una o más cámaras. No obstante, típicamente se realiza mediante sistemas compuestos por dos cámaras separadas horizontalmente, simulando el sistema humano de percepción. Otro aspecto a destacar en los sistemas de visión estereoscópica es su geometría, donde encontramos geometrías con los ejes ópticos paralelos o convergentes. El sistema visual humano se basa principalmente en ejes convergentes, donde los ojos enfocan al objeto que queremos visualizar. Por otro lado, los sistemas de visión estereoscópica artificial suelen utilizar ejes paralelos debido a la simplicidad que introducen en el cómputo frente a geometrías convergentes. Las aproximaciones utilizadas para sistemas de visión estereoscópica en detección de vehículos son las presentadas a continuación.

**Mapas de disparidad** El conjunto de diferencias que encontramos entre los píxeles de las imágenes de una misma escena se conoce como mapas de disparidad. Si los parámetros de la plataforma del sistema estéreo son conocidos, un mapa en 3D de la escena puede ser creado a partir de los mapas de disparidad. La generación de hipótesis de obstáculos se hace generalmente mediante la búsqueda de grupos de columnas de píxeles con disparidad similar (por ejemplo, [17]) o mediante la explotación de la información de profundidad de alguna otra manera. Los mapas de disparidad se han utilizado en muchos trabajos sobre detección de vehículos. Por ejemplo en [18], los mapas de disparidad, junto con el flujo óptico, son utilizados como una señal visual, y cada píxel de la imagen se clasifica como objeto en movimiento o fondo. El rendimiento óptimo de clasificación se obtiene con un perceptrón multicapa de 3 capas. Como es de esperar, el cálculo de la disparidad no obtiene resultados positivos para todos los píxeles de la imagen. En la literatura se pueden encontrar varios métodos para superar este problema. En [18], estos

problemas se solucionan a través de la fusión con la información de otros sensores. En [19], se calcula la homografía del plano de la carretera entre las vistas izquierda y derecha de la plataforma del sistema estéreo, y sólo las características correspondientes que hayan reducido los errores de re-proyección en ambas imágenes se consideran válidas. En [17], en lugar de resolver el problema correspondiente localmente, se propone un método de optimización global basado en programación dinámica que proporciona una interpretación más consistente globalmente del par de imágenes en estéreo, eludiendo las discontinuidades de profundidad. El problema de estos métodos es que computacionalmente son muy costosos. Por eso en [20], se discute que para resolver el problema, las aproximaciones basadas en áreas, conllevan un coste computacional muy alto y los mapas de disparidad obtenidos mediante métodos basados en características no son lo suficientemente densos. Por lo tanto, se propone un extractor de características local para resolver el problema de manera más rápida. De acuerdo con esta aproximación, cada píxel es clasificado en diferentes categorías (por ejemplo, píxel de borde vertical, píxel de borde horizontal, píxel de esquina, etc.) basándose en la diferencia de intensidad entre el píxel y sus cuatro vecinos directos. Para que la búsqueda de los píxeles correspondientes sea más simple la plataforma de visión estéreo esta alineada en paralelo. De acuerdo a lo explicado anteriormente, la búsqueda de los píxeles correspondientes se reduce a una simple comparación para determinar si los píxeles pertenecientes a las imágenes de una misma escena pertenecen a la misma categoría. Obviamente, hay casos en los que este enfoque no produce correspondencias únicas. Para abordar este problema, se clasifican además los píxeles por sus disparidades asociadas en varios intervalos mediante la construcción de un histograma disparidad. El número de picos significativos en el histograma indica cuántos objetos posibles estaban presentes en las imágenes.

**Mapeado de perspectiva inversa (*Inverse Perspective Mapping, IPM*)** En cualquier proceso de adquisición de imágenes existe un efecto de perspectiva inherente debido a la ubicación de la cámara. Esto no corresponde con una inversión real de mapeo de perspectiva [21], el cuál es matemáticamente imposible. Más bien, denota una inversión en virtud de la restricción adicional de que los puntos mapeados inversamente se encuentran en el plano horizontal. Podemos encontrar una explicación de la geometría del mapeado de perspectiva en [2]. Si se considera un punto  $p$  en el espacio 3D, el mapeado de perspectiva implica una línea que pasa por este punto y el centro de la proyección  $N$ , Figura 2.4. Para encontrar la imagen del punto, intersecamos la línea con el plano de la imagen. IPM se define mediante el siguiente procedimiento. Para un punto  $p'_I$  en la imagen, se traza el rayo asociado a través de  $N$  hacia el plano horizontal. La intersección del rayo con el plano horizontal es el resultado de la asignación de perspectiva inversa aplicada al punto de la imagen  $p'_I$ . Si componemos tanto la perspectiva como la perspectiva inversa, el plano horizontal se asigna a sí mismo, mientras que las partes elevadas de la escena aparecen distorsionadas.

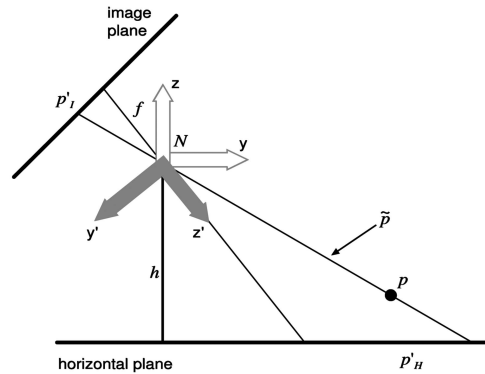


Figura 2.4: Geometría de mapeado de perspectiva [2].

La transformación matemática que consigue esta inversión es conocida como mapeado de perspectiva inversa [21, 22] o deformación de la imagen plano a plano [23]. IPM ofrece una vista panorámica de la escena, en este caso, de la carretera. En [24], el IPM se utiliza para convertir la imagen izquierda a una imagen sin perspectiva, y para transformar ésta a su vez a la perspectiva de la vista derecha (con la correspondiente transformación). Comparando esta vista con la vista real de la derecha de la plataforma estéreo, se pueden localizar los objetos encima del plano de tierra. Alternativamente, en [22], tanto la imagen izquierda, como la derecha se proyectan en el plano de la carretera mediante el uso de sus correspondientes transformaciones dadas por IPM. Aquellas imágenes en las que la diferencia de sus dos imágenes remapeadas contengan grandes agrupaciones de píxeles cuyos valores no sean cero, tienen una forma específica y por tanto pueden ser vehículos. Por otro lado, en [25], los bordes que cruzan la línea del horizonte se detectan a medida que convergen a la proyección de la cámara virtual en el dominio IPM del par estéreo.

Como ya se ha comentado con anterioridad, el rendimiento de los sistemas basados en visión estéreo están sujetos a una buena estimación de los parámetros de calibración, los cuales suponen un gran reto en sistemas instalados a bordo del vehículo, debido a la vibración que produce el movimiento. Además, el coste y la flexibilidad de las cámaras estéreo que se encuentran en el mercado siguen estando muy lejos de las cámaras utilizadas en sistemas monoculares. No obstante, este tipo de sistemas constituyen un campo de estudio muy amplio y con un futuro prometedor en sistemas de ayuda en carretera.

## 2.4. Métodos basados en movimiento

A diferencia de otros métodos que buscan la separación entre los objetos que queremos determinar, en nuestro caso los vehículos y el fondo, los métodos basados en movimiento determinan los vehículos potenciales buscando el movimiento de los objetos en la escena. La aproximación

principal que encontramos para este tipo de métodos es la obtención de movimiento mediante el cálculo del flujo óptico, el cuál crea un mapa con los vectores de movimiento de todos los píxeles (flujo óptico denso) o de un conjunto de elementos (flujo óptico disperso) [1]. En particular, los vehículos que se mueven más rápido que el observador (es decir, los adelantamientos y salidas de vehículos) producen un flujo óptico convergente en el punto de fuga, mientras que los vehículos más lentos producen una divergencia respecto a el flujo óptico.

En [26], se hace uso de flujo óptico denso. El flujo óptico se estima a partir de derivadas espacio-temporales de las imágenes de valores de gris con un enfoque local. El flujo óptico estimado se agrupa para eliminar valores atípicos. Suponiendo una cámara calibrada y conocido el ego-movimiento se detectan ambos tipos de objetos, en movimiento y estacionarios. Sin embargo, el flujo óptico denso supone un cómputo lento, por lo tanto, el uso del flujo óptico disperso generalmente se prefiere para aplicaciones en tiempo real. Este es el caso por ejemplo de [27], donde el flujo óptico se calcula sólo en los ejes relevantes de la imagen. Para la estimación de flujo óptico se utiliza el método de gradiente, llamado VIGM (*Virtual Intensity Gradient Method*), en el que se asume un gradiente de intensidad lineal alrededor de los bordes. En [28], el flujo óptico sólo se calcula en posiciones concretas, obteniéndose de la siguiente manera: una fracción de las posiciones se determina mediante el uso del algoritmo de muestreo conocido como remuestreo secuencial Monte Carlo, mientras que la fracción restante de las posiciones se determina mediante el uso de una variable aleatoria.

Sin embargo, el movimiento de la cámara que se produce al estar instalada en el vehículo supone un reto en la interpretación del flujo óptico. De hecho, los vectores de flujo óptico reflejan el movimiento de los elementos de imagen en relación con el observador. Por lo tanto, ya que la cámara se está moviendo, los píxeles pertenecientes al fondo proporcionan vectores de flujo óptico que no son cero, incluso si son estáticos, que además difieren de un píxel a otro debido al efecto de la perspectiva. Algunos métodos intentan inferir el ego-movimiento, de modo que ambas fuentes de movimiento pueden ser dissociadas. Por ejemplo, autores como [29], estiman el ego-movimiento mediante el cálculo de la matriz esencial entre dos vistas consecutivas de una escena utilizando correspondencias de características. El método explota el conocimiento del ego-movimiento, junto con la extracción de los parámetros del plano de la carretera, para calcular de manera explícita la homografía entre los planos de tráfico en los dos puntos de vista, de manera que las imágenes pueden ser alineados y los objetos en movimiento detectados. Sin embargo, la estimación del ego-movimiento no es una tarea fácil, sobre todo si las imágenes se componen de objetos en movimiento, como es el caso. En efecto, para derivar la matriz esencial sólo las características pertenecientes a elementos estáticos se deben tener en cuenta. Por lo tanto, para extraerlo, las regiones de la imagen que contiene los vehículos tienen que ser conocidas de antemano o inferidas, de modo que las características dentro de ellos se quedan fuera para el cálculo de la matriz esencial.

## 2.5. Métodos basados en apariencia

La apariencia de los vehículos ha sido siempre una línea importante de estudio cuando se trata de la detección de vehículos. Estos métodos buscan a través del conocimiento de la apariencia de los vehículos, extraer de la escena las regiones donde se encuentran. Los métodos que nos podemos encontrar se pueden agrupar a su vez en dos categorías: métodos basados en modelos y métodos basados en el aprendizaje. Ambas clases se basan en la extracción de las características que podemos encontrarnos en la apariencia de los vehículos. Los métodos basados en modelos usan esta extracción de característica para establecer una correlación entre la imagen y unas plantillas preestablecidas relativas a la apariencia de los vehículos. Por otro lado, los métodos basados en el aprendizaje tratan la detección de vehículos como un problema de clasificación entre dos clases: la clase vehículo y la clase no vehículo. En estos métodos, a priori se realiza el entrenamiento de un clasificador mediante las características extraídas de un conjunto de imágenes representativo de la variabilidad de las clases. Luego, para cada imagen que queramos determinar si se corresponde con un vehículo o no, se realiza la extracción de esas características y se clasifica mediante el clasificador entrenado. Los métodos basados en aprendizaje de la necesidad de definir un método que capturase la variabilidad de los vehículos de manera eficiente, algo muy difícil de obtener mediante los métodos basados en modelos, ya que sería necesario una gran base de datos con todas las posibles plantillas a utilizar y esto resulta inviable para aplicaciones en tiempo real.

### 2.5.1. Métodos basados en modelos

Estos métodos utilizan modelos o plantillas de vehículos las cuales se van a utilizar para hallar una medida de similitud con la hipótesis a clasificar. Estos métodos explotan el hecho de que todos los vehículos tienen una forma similar, aproximadamente poliédrica. Aunque los atributos varían en complejidad, éstos siempre se basan en atributos que se pueden observar a simple vista como son la simetría, los bordes, etc. Algunos trabajos como [13] usan la región de máxima simetría para determinar donde se encuentra hipotéticamente el vehículo y a partir de esto busca las esquinas que existen típicamente en los vehículos para detectar si se trata de un vehículo o no. Típicamente se utilizan modelos fijados o rígidos para realizar la detección, como es el caso de [15, 30]. Un ejemplo de estas aproximaciones, lo podemos ver en [15], el cuál utiliza un modelo donde los vehículos consisten en dos conjuntos de líneas paralelas, a lo largo y a lo ancho del vehículo. Alternativamente en otros trabajos se propone el uso de una cascada de clasificadores basados en modelos como es el caso de [31], donde primero se busca el número de esquinas que existen en la imagen y a partir de estas se rechazan todas las imágenes que no cumplan con el número mínimo de esquinas fijado que se determina que tienen que existir en los vehículos. Luego, se calcula una medida de similitud entre los candidatos y un conjunto de

plantillas de vehículos. La medida se basa en dos características: por una parte, el contorno del vehículo está representado por un conjunto de coeficientes inferidos por la media de la distancia de la transformada y por otra parte, se utiliza una transformada wavelet de escala 3 (*three-scale wavelet transform*) para extraer los coeficientes que caracterizan la región. Por último, la representación de la región candidata en términos de los coeficientes combinados se compara con la de las plantillas, y una medida de similitud se extrae utilizando una función gaussiana. Si ellos son similares, el candidato es aceptado, de lo contrario se rechaza.

Sin embargo, este tipo de aproximaciones presentan una limitación importante derivada de la gran variabilidad presente en los vehículos: requieren un amplio conjunto de modelos para capturar las particularidades presentes en los diferentes vehículos. Esto implica una carga computacional excesiva y se hace imposible su aplicación en sistemas en tiempo real [1]. Debido a esto, otros autores proponen modelos deformables. Estos modelos fueron introducidos en [32] para el seguimiento de vehículos. En [33], para estimar la deformación de estos modelos, se propone una función de energía que realiza una combinación de varias características, eje de simetría vertical y horizontal, gradientes e información proveniente de las sombras. Más recientemente, en [34], se propone el uso de modelos deformables 3D, los cuales están gobernados por doce parámetros relativos a la forma. Además, estos modelos tienen en cuenta tres parámetros adicionales relativos al posicionamiento del vehículo.

### 2.5.2. Métodos basados en el aprendizaje

Mediante este tipo de métodos, en el caso de los vehículos, nos encontramos con un problema de clasificación de dos clases (vehículos, no-vehículos). La primera etapa que se realiza en estos métodos, es una etapa de aprendizaje: a partir de un conjunto de imágenes, a las que llamaremos conjunto de entrenamiento, se realiza el entrenamiento de un clasificador mediante el uso de las características extraídas previamente de la imagen. El conjunto de características extraídas de la imagen constituye un vector representativo de dicha imagen. Es importante que el conjunto de entrenamiento tenga una alta variabilidad para conseguir un clasificador robusto. Para determinar que imágenes son vehículos, se extrae un vector de características y se procede a su clasificación. Trabajos como [10, 35], utilizan este tipo de métodos para la detección de vehículos. En [10], hace uso de diferentes tipos de características relativas a la apariencia como son las sombras producidas por los vehículos, la simetría inherente de estos o las esquinas que forman la estructura de los vehículos para entrenar el clasificador y realizar la detección de vehículos en la escena. Otros trabajos como [35], utilizan otro tipo de características como son los histogramas de gradientes orientados para realizar la detección.





Figura 2.5: Ejemplo de las luces de los vehículos. En la imagen se muestra en primer lugar las luces de cruce de un vehículo en condiciones de baja iluminación. La segunda imagen muestra las luces de freno en escenario diurno. La tercera imagen consta de un vehículo con las luces de marcha atrás encendidas. Por último se presenta una imagen de un vehículo con la luz de intermitencia encendida.

### 2.5.3. Extracción de características

La elección de un conjunto de características adecuado resulta realmente importante para la obtención de un buen rendimiento en la detección. Las características que podemos extraer a partir de los vehículos se pueden clasificar como, características explícitas y características implícitas. Las primeras implican un conocimiento previo de cómo es el objeto, en nuestro caso los vehículos, por lo que hacen uso de sus atributos para definir el descriptor. Por otro lado las características implícitas no hacen ninguna suposición acerca de la forma, estructura o apariencia de los objetos que se van a clasificar.

#### 2.5.3.1. Características explícitas

En el campo relativo a escenas de tráfico es muy común el uso de características explícitas, debido a que sabemos de antemano la apariencia de los vehículos. Su uso implica un cómputo sencillo y usualmente vectores de características cortos, por lo que resulta adecuado para implementarse en sistemas en tiempo real [1].

Las características explícitas incluyen medidas simples relacionadas con el conocimiento de las características del vehículo como la simetría, bordes, sombras, etc. Los métodos más utilizados para la detección de vehículos son la búsqueda de simetría y de bordes del vehículo. Atendiendo al tipo de características empleadas se presenta a continuación una agrupación de los trabajos más relevantes encontrados en la literatura.

**Luces de los vehículos** En condiciones medioambientales de baja iluminación, la extracción de este tipo de características constituye descriptores que ofrecen un alto rendimiento en detección. Además este tipo de características cobran gran importancia en situaciones donde las luces de freno, marcha atrás e intermitencia están encendidas. Esto se muestra en la Figura 2.5.

Por tanto trabajos como [36] realizan la detección del vehículo mediante la búsqueda del color rojo para las luces traseras o de freno, el color naranja para las luces de intermitencia y el color blanco para la marcha atrás. Para poder realizar la detección de estos colores utiliza una



Figura 2.6: Ejemplo de las sombras provocadas por los vehículos sobre la carretera.

transformación al espacio  $L^*a^*b^*$ , que se trata de un espacio de color perceptualmente uniforme, basado en el sistema de color XYZ. Este sistema tiene la ventaja de preservar la diferencia entre colores utilizando la distancia euclídea en el espacio de color  $L^*a^*b^*$ . Alternativamente en [12], las hipótesis de los vehículos en escenarios nocturnos se generan utilizando filtros de perspectiva de la región en la que se encuentra el objeto y posteriormente buscando el par de luces traseras mediante una aproximación de agrupamiento basada en normas. Otra aproximación muy popular es el uso de filtros *top-hat* [37]. En [38] se utilizan estos filtros para la detección de las luces delanteras del vehículo mediante la combinación de cuatro operaciones de apertura morfológicas con elementos estructurantes rotados.

La principal desventaja de estos métodos es el bajo rendimiento en escenarios diurnos, por ello trabajos más recientes como [39], presentan algoritmos que se basan en el uso de las luces de freno e intermitentes para la detección de vehículos. Este método con la ayuda de filtros Kalman [40], que ayudan a reducir la carga computacional, tratan de conseguir un método robusto además de en escenarios nocturnos o con baja visibilidad, en escenarios diurnos. Otros trabajos como [41], buscan reducir esta limitación y obtener un descriptor más robusto mediante la fusión con otro tipo de características y, en su caso, fusionan las luces de los vehículos junto a los bordes y la simetría. Específicamente, el método crea primero los candidatos de vehículos mediante la extracción de los picos prominentes de la distribución de color rojo en la imagen. A continuación, se analiza la simetría de la distribución de color rojo para derivar el eje de simetría. Por último, los pares de bordes se buscan en posición simétrica a la izquierda y a la derecha de este eje, para determinar el vehículo.

**Sombras** Las sombras han sido muy utilizadas en la literatura para la localización de los vehículos. Esta característica está presente en la mayoría de las imágenes que contienen vehículos debido a que estos reducen el brillo de la carretera, al producir una sombra cuando una fuente de iluminación incide sobre ellos, ya sea natural o artificial [42], como se puede ver en la Figura 2.6.

Este tipo de características se empezaron a utilizar en [42]. Las sombras en [43] se detectan



Figura 2.7: Ejemplo de los diferentes colores que nos podemos encontrar en los vehículos.

analizando las características de intensidad y gradiente. En este método la media de la intensidad de pavimento se caracteriza primero mediante el análisis de las regiones cercanas del pavimento. Luego los segmentos de línea que representan transiciones a gris más oscuro se determinan mediante el análisis de gradientes. Por último, las zonas de sombra se determinan mediante la evaluación de la región delimitada por las combinaciones de segmentos de línea en términos de intensidad media y la varianza, y mediante la imposición de restricciones de coherencia geométricas. Una estrategia similar se utiliza en [44] donde las regiones de la carretera son identificadas a priori a la aplicación del análisis por gradientes. La detección de sombras también es muy empleada en sistemas donde se analizan varios tipos de características como en [10], donde combinan sombras, simetría y esquinas. En este trabajo las sombras se determinan mediante el análisis de los perfiles verticales de la imagen, buscando la transición de los valores de gris en la carretera.

Estos métodos tienen como principal desventaja la dependencia de las condiciones meteorológicas y de la iluminación. Como se puede apreciar en la Figura 2.6b, en ciertas condiciones de iluminación las sombras se expanden, dejando de adaptarse a la forma de los vehículos. Otro factor a tener en cuenta es que en situaciones de baja iluminación las sombras pueden llegar a desaparecer como se muestra en la Figura 2.6c.

**Color** La mayoría de los vehículos son cromáticos, mientras que la carretera es acromática [45]. Un ejemplo de los múltiples colores que nos podemos encontrar en los vehículos lo podemos ver en la Figura 2.7.

Los métodos que nos encontramos en la literatura utilizan diferentes espacios de color, siendo los más empleados RGB y HSI. El uso de la característica de color para la detección de vehículos es relativamente nueva, en [46] se puede observar un método de segmentación basado en la intensidad y la saturación del espacio de color HSI, no obstante se ha de destacar que la característica de matiz no aporta información relevante para la clasificación y por lo tanto se desprecia. En [47], se define un nuevo espacio de color basado en la transformación Karhunen-Loève. Los autores exponen como sobre este espacio se puede obtener una mejor separación entre vehículos y carretera de acuerdo al criterio de Fisher. Otros trabajos más recientes [48], utilizan el espacio de características RGB. Este sistema cuenta con tres etapas fundamentales: pre-procesado, diferenciación de imágenes y detección y eliminación de sombras.



Figura 2.8: Ejemplo de la estructura rectangular de los vehículos.



Figura 2.9: Ejemplo de la característica de simetría en vehículos.

La limitación más importante de estos métodos es que existe una gran variabilidad en el color de los vehículos, además de cambiar su tonalidad bajo ciertas condiciones de luminosidad, lo cuál dificulta el análisis.

**Esquinas** Debido a la estructura rectangular de los vehículos la extracción de esquinas se propone como un método para la detección de los vehículos, suponiendo los vehículos como modelos de cuatro esquinas, como vemos en la Figura 2.8.

Por ejemplo, en [49] los autores proponen la búsqueda de estructuras rectangulares mediante la aplicación de un operador de gradiente. Este tipo de características no se presentan como una buena opción para la detección de vehículos, dado que en las escenas de tráfico se pueden observar un gran número de objetos con forma rectangular u obtener esquinas del fondo que dificulten la clasificación. Por este motivo este tipo de métodos suele usarse como combinación a otro tipo de características para obtener sistemas más robustos. Esto se puede observar en [49] o [50], donde se utiliza la detección de esquinas junto a otro tipo de características como los bordes.

**Simetría** Una de las características más utilizadas para la verificación de hipótesis en el campo de vehículos es la simetría. Los vehículos debido a su estructura son típicamente simétricos cuando se observa la parte delantera o trasera del vehículo, como en la Figura 2.9.

Esta propiedad ya se empezó a utilizar en 1991 [11], y todavía se sigue empleando en sistemas

de detección recientes. Son muchos los trabajos que encontramos al respecto. En [51] se propone un método de votación para determinar la situación del eje de simetría. La idea es buscar para cada par de píxeles de una misma fila su eje potencial mediante votación. Luego, a partir de todos los candidatos obtenidos en cada fila se busca el eje vertical que haya obtenido más votos, fijando este como eje de simetría de la imagen completa. Otros trabajos como [52], utilizan una combinación de sombras y simetría para determinar los vehículos. Adicionalmente se utilizan restricciones de texturas para que se cumpla que sólo las regiones con la entropía suficiente sean consideradas. El eje de simetría, a diferencia del trabajo anterior no se calcula, si no que se calcula la medida de simetría para todos los ejes potenciales. Primero, se parte de una función, la cuál representa los valores de intensidad de cada uno de los píxeles de la imagen. A partir de esta función se calculan sus funciones par e impar, fijando el eje de simetría y el ancho del intervalo de simetría. La medida de simetría es calculada a partir de estas dos funciones como se muestra en la Ecuación 2.1:

$$S = \frac{\int E^2 - \int O^2}{\int E^2 + \int O^2} \quad (2.1)$$

donde  $E$  será la función par y  $O$  la función impar. Por lo tanto el valor  $S$  indicará la medida de simetría para cada eje potencial con respecto a un ancho de intervalo concreto. Por último, para tener en cuenta el ancho de la ventana de simetría, normaliza este valor. Otros trabajos como [53, 9], utilizan una combinación de ejes para determinar el eje definitivo de la imagen. En [53], para el cálculo de los ejes se hace uso del ancho de la sombra que provoca el vehículo en la carretera. Este método utiliza una combinación de tres ejes obtenidos mediante diferentes métodos para determinar el eje de simetría de la imagen. Los métodos que considera para el cálculo de los ejes son: simetría del contorno binario, simetría en escala de grises y simetría en componente S del espacio de color HSV. Por otro lado, la simetría no siempre esta ligada al dominio de la intensidad, por ejemplo en [9], para determinar el eje de simetría elegido se realiza una combinación de diferentes métodos de obtención del eje: simetría a escala de grises, simetría por bordes, simetría por bordes verticales y simetría por bordes horizontales.

Para obtener valores altos de simetría se hace uso de imágenes en plano frontal del vehículo, como se muestra en la imagen de la Figura 2.9. Sin embargo, en situaciones típicas de tráfico, no nos encontramos exclusivamente situaciones donde el vehículo se encuentra circulando en frente de nosotros, si no que también pueden aparecer vehículos en las líneas laterales de visión y por lo tanto el rendimiento que proporciona la simetría se degrada. Además, en situaciones reales de tráfico existen múltiples regiones, pertenecientes al fondo (carreta, cielo) u objetos que son altamente homogéneos, que también dan lugar a valores altos de simetría. Por ello son pocos los sistemas que recurren a esta característica en solitario, siendo implementada en la actualidad en combinación con otro tipo de características, gracias a su rendimiento y al bajo coste computacional cuando se utiliza junto con otro tipo de características.



Figura 2.10: Ejemplo de extracción de bordes laterales.

**Bordes** El uso de bordes en sistemas de detección de vehículos está muy extendido en la literatura. Gracias a la forma rectangular de su estructura, el contorno de los vehículos da lugar a valores muy altos del gradiente. La simple extracción de bordes verticales y horizontales mediante métodos sencillos, como pueden ser Sobel, Canny o Prewitt [54], proporcionan picos en los histogramas generados por los gradientes de la imagen, los cuales delimitan la estructura del vehículo y pueden ser suficientes y fundamentales en la clasificación realizada, como se muestra en la Figura 2.10.

Trabajos como [55], utilizan estos métodos, empleando para la detección de vehículos, la extracción de bordes horizontales y verticales por separado utilizando un operador de Sobel. Alternativamente, otros autores, proponen diferentes métodos para la extracción de bordes. En [56], la extracción de bordes se basa en estrategias de predicción del gradiente vecino (NPG), lo cual se propone para la detección de vehículos al ser menos sensible a los cambios de iluminación. Por otro lado, en [3] nos encontramos una transformación de las imágenes mediante IPM, la cuál nos proporciona una visión aérea virtual que facilita la extracción de los bordes laterales del vehículo, ya que éstos se refuerzan, como se puede ver en la Figura 2.11. Cabe destacar que en la detección de vehículos mediante bordes, los bordes verticales adquieren una mayor importancia que los bordes horizontales debido a que se encuentran en dirección radial al punto focal de la cámara.

Como ocurre con otras características, los bordes también se usan habitualmente en sistemas de fusión, como en [57] donde se combinan con detección de sombras. Estos métodos, a diferencia de los anteriores presentados, son los más empleados debido a su alto rendimiento. No obstante, presentan, al igual que otros tipos de características, ciertas limitaciones. La principal desventaja radica en los bordes horizontales. Es fácil encontrar situaciones donde las imágenes que no contienen vehículos presenten bordes horizontales muy definidos (línea de horizonte,...). De igual forma ocurre con los bordes verticales, sin embargo, debido a la estructura de los vehículos en los

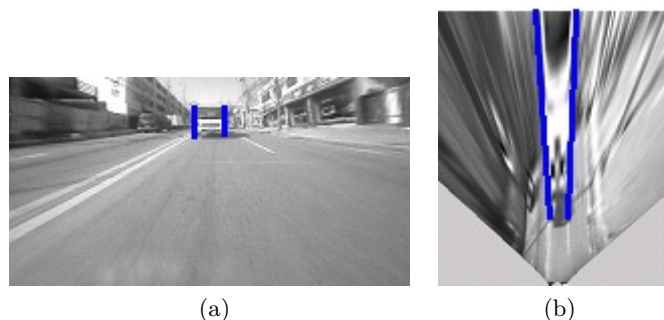


Figura 2.11: Transformación de la imagen a visión aérea [3]. (a) Imagen original. (b) Imagen aérea virtual.

cuales nos encontramos dos picos muy pronunciados en el histograma generado por los bordes verticales, resulta más difícil encontrar este tipo de distribución de los gradientes donde no se encuentran vehículos.

### 2.5.3.2. Características implícitas

El campo de la detección de vehículos mediante el uso de características implícitas está cobrando una importancia significativa en los últimos años. Las técnicas de extracción de características implícitas incluyen el análisis de componentes principales (*Principal Component Analysis, PCA*), transformada Wavelet, histogramas de gradientes orientados (*Histograms of Oriented Gradients, HOG*), filtros de Gabor y características *Haar-like*.

**Transformada Wavelet** La transformada *Wavelet* organiza la imagen en sub-bandas que se localizan en orientación y frecuencia. Existen otras descomposiciones posibles que representan la imagen en espacio, frecuencia y orientación, tales como la transformada de Fourier a corto plazo o algoritmos en pirámide. Las *wavelets*, a diferencia de estas otras transformadas, pueden reconstruir perfectamente la imagen a partir de su transformada, donde el número de coeficientes de la transformación es igual al número original de píxeles [1]. La transformada *Wavelet* se utiliza en algunos de los primeros trabajos en detección de vehículos. Por ejemplo, en [58] se explota el uso de estas características para representar un conjunto de clases de objetos y no-objetos incluyendo coches y rostros. En este trabajo, las estadísticas de la apariencia de ambas clases objetos y no-objeto se representan utilizando el producto de histogramas. Cada histograma representa las estadísticas conjuntas de un subconjunto de coeficientes *wavelet* y su posición en el objeto. En particular, las *Haar wavelet* puede ser vistas como una representación de la orientación local de las diferentes intensidades entre regiones adyacentes. En [59], se propone una variante truncada y cuantificada de la transformada *Haar wavelet*, con el fin de conseguir una disminución del coste computacional. En particular, los primeros coeficientes que tienen

magnitud pequeña se truncan y, a continuación, se cuantifican en tres niveles, -1, 1 y 0, que representan, respectivamente, grandes coeficientes negativos, grandes coeficientes positivos y los valores intermedios. Esta cuantificación abrupta se justifica por el hecho de que los valores de los coeficientes exactos podrían ser menos importantes o fiables que la mera presencia o ausencia de los mismos.

**Características Haar-like** Son una variación de la transformada tradicional *Haar wavelet* introducida en [60]. Aunque las características *Haar-like* son una reminiscencia de las funciones de base *Haar*, se utiliza un conjunto de características rectángulo sobrecompletas. Las características *Haar-like* consideran regiones rectangulares adyacentes en una ubicación específica de la ventana de detección, suma las intensidades de los píxeles en cada región y calcula la diferencia entre estas sumas. Esta diferencia se utiliza para clasificar las subsecciones de una imagen. Esta idea fue desarrollada posteriormente por Viola y Jones [61] para la detección de rostros. En este trabajo, se proponen tres tipos de características *Haar-like*: de dos rectángulos, las características de tres rectángulos y de cuatro rectángulos. El valor de las características de dos-rectángulos es la diferencia entre la suma de los píxeles dentro de dos regiones rectangulares adyacentes. Las características de tres-rectángulos consisten en tres regiones rectangulares y su valor se calcula mediante la adición de los píxeles del rectángulo interior y restando los de los rectángulos externos. Por último, las características de cuatro-rectángulos calculan la diferencia entre cada par diagonal de rectángulos. Además, proponen una nueva representación intermedia de la imagen, llamada imagen integral, que permite un rápido cálculo de estas características. Las características *Haar-like* son ampliamente utilizadas en combinación con clasificadores potenciados. Estos clasificadores se construyen mediante la combinación de funciones de clasificación débiles para formar un clasificador más fuerte. Viola y Jones propusieron el uso de AdaBoost para seleccionar las mejores características del clasificador débil. En trabajos como [62], las características *Haar-like* son utilizadas junto con histogramas de gradiente orientados, comparando el uso de un detector simple con arquitecturas en cascadas. En [63], para reducir el número de características utilizadas en el clasificador débil, se propone la programación cuadrática. En particular, la programación cuadrática asigna un orden de prioridad a los clasificadores débiles y por tanto sólo aquellos con una alta prioridad son seleccionados. Por último, se utiliza la combinación lineal de los clasificadores seleccionados.

**Filtros de Gabor** Los filtros de Gabor son filtros lineales cuya respuesta al impulso es una función sinusoidal multiplicada por una función gaussiana. La principal ventaja que se obtiene al introducir la envolvente gaussiana es que las funciones de Gabor están localizadas tanto en el dominio espacial como en el de la frecuencia, a diferencia de lo que ocurre con las funciones sinusoidales, que solamente están localizadas en el dominio frecuencial. Los filtros de Gabor están directamente relacionados con las *wavelets*, dado que son funciones aproximadamente paso banda



que pueden diseñarse como un banco de filtros con diferentes escalas y orientaciones. Uno de los inconvenientes del uso de *wavelets*, es que los filtros tienen que ser ortogonales. Una opción alternativa consiste en trabajar con representaciones sobrecompletas y considerar que son una buena aproximación al caso ortogonal. Por ejemplo, se pueden descomponer imágenes en bancos de Gabor sobrecompletos y volver a reconstruir la imagen original simplemente sumando los diferentes canales. Esto sólo se podría hacer en modo estricto si los canales de Gabor fuesen ortogonales [64]. Así podemos ver en los filtros de Gabor una alternativa al uso de *wavelets* para la detección de vehículos. Este tipo de filtros suele utilizarse en aplicaciones de análisis de texturas, como de segmentación [65], en clasificación de texturas [66], y en recuperación de imágenes [67]. Los vehículos tienen un contenido de características de bordes significativos a diferentes escalas y orientaciones (principalmente horizontales y verticales), tanto en su contorno, debido a su contraste con los elementos de fondo, como en su estructura interna. Además, estas líneas y bordes tienden a tener una distribución similar para la mayoría de las partes posteriores de vehículos. Por lo tanto, los filtros de Gabor están bien adaptados para la caracterización de vehículos. Tradicionalmente, los filtros de Gabor en diferentes escalas y orientaciones se utilizan para la extracción de características.

Por ejemplo en [68], se utilizan un banco de ocho filtros de Gabor con dos longitudes de onda y cuatro orientaciones. Normalmente, la respuesta de cada filtro en el banco se caracteriza mediante datos estadísticos como la media o la desviación estándar [69]. Como ocurre en casos anteriores, las características extraídas mediante filtros de Gabor también son utilizadas en combinación con otro tipo de características, como ocurre en [59], donde se realiza un sistema de fusión de filtros de Gabor y características *wavelet*.

**Análisis de componentes principales** Es un procedimiento estadístico que utiliza la transformación ortogonal para convertir un conjunto de observaciones cuyas variables están posiblemente correladas, en un conjunto de valores donde sus variables están distribuidas linealmente y son incorreladas, llamadas componentes principales. El número de componentes principales es menor o igual al número de variables originales. Esta transformación está definida de tal manera que la primera componente principal tiene la mayor varianza posible, y cada componente subsiguiente a su vez tiene la mayor varianza posible en virtud de la restricción que es ortogonal a las componentes anteriores. La extracción de características mediante PCA es uno de los métodos más extendidos en la detección de vehículos. Esta técnica además de utilizarse para la detección de vehículos, se utiliza en reconocimiento facial [70]. Esta técnica tiene como objetivo reducir la dimensionalidad del espacio de características mediante la búsqueda de las direcciones más discriminativas. PCA se puede aplicar directamente sobre la imagen o sobre un conjunto de características extraídas de la misma. Por lo tanto, la principal aplicación de PCA es la de reducir la dimensionalidad del conjunto de características antes de la clasificación.

Por ejemplo, en [71] primero se realiza la extracción de características mediante Haar wavelets y luego se aplica PCA para reducir el espacio de características del conjunto de entrenamiento. Luego para la clasificación de una muestra se calcula una función de error de pertenecer a cada una de las clases, y se asigna esta muestra a la clase donde el valor de la función de error obtenido es menor. En [72], nos encontramos con un sistema basado en PCA, que realiza una clasificación entre potenciales vehículos, utilitarios, furgonetas y camiones. Este sistema se reduce a un problema de clasificación de una clase, en el cuál, las nuevas muestras se clasifican en si pertenecen a una de estas tres clases o a ninguna. Trabajos como [73] exploran el uso de PCA directamente sobre la imagen, realizando pruebas de la influencia de la dimensionalidad del subespacio de características generado por PCA sobre la clasificación. Otros trabajos como [59] exploran la verificación de vehículos mediante la extracción de características usando PCA sobre dos campos diferentes de clasificación, SVM y redes neuronales.

**Histogramas de Gradientes Orientados** El descriptor HOG se propuso inicialmente en el ámbito de la detección de personas [74], y debido a los excelentes resultados obtenidos rápidamente se expandió su aplicación a otros campos de investigación como el reconocimiento de caras [75]. En estudios recientes, estos métodos también se han aplicado a sistemas de detección de vehículos basados en imágenes. Normalmente la configuración del descriptor HOG aplicado sobre vehículos sigue las recomendaciones que propone Dalal en [74] para el reconocimiento de personas, este descriptor se aplica típicamente en imágenes estáticas. La implementación de descriptores que siguen la configuración del descriptor HOG propuesto por Dalal está presente en multitud de trabajos, como por ejemplo en [76] donde se usa este descriptor para detección de vehículos en vista aérea, o en [77] donde también se utiliza este descriptor. Otros como [35] usan el descriptor HOG para la detección de vehículos en bases de datos donde los vehículos se observan desde diferentes puntos de vista. En la literatura también se han propuesto modificaciones sobre este descriptor, con el objetivo de conseguir un descriptor más eficiente. En [78], Dalal describe como se debe modificar el descriptor para conseguir descriptores adecuados para cada tipo de objeto. Así, para el caso de vehículos propone el cambio en el rango de los ángulos, al igual que modificaciones en el tipo de normalización utilizado. Otros trabajos indagan en la modificación de la forma, por ejemplo en la modificación de las celdas o bloques del descriptor HOG. El descriptor VHOG propuesto en [35] usa solamente celdas verticales, con esto se consigue la reducción del coste computacional y por lo tanto facilita su utilización en sistemas de tiempo real. En [79] se buscan distribuciones de las celdas relativas a las características de los vehículos como luces, matrícula, etc. para realizar la clasificación entre diferentes tipos de vehículos. Por otra parte, trabajos como [80], exploran el uso de diferentes tamaños de bloques. Finalmente, como ocurre en otros métodos para conseguir un aumento de la eficiencia se realiza fusión con otro tipo de característica, como en [76] donde se fusiona el descriptor HOG con

características Haar.

#### 2.5.4. Clasificación

La fase de clasificación supone una etapa fundamental en el sistema de detección de vehículos, ya que la elección de un clasificador que se adapte lo mejor posible a las distribuciones de las clases que se van a clasificar, puede suponer un aumento importante en el rendimiento del sistema. Esto se debe a que los sistemas de detección de vehículos son básicamente un problema de reconocimiento de objetos, en el cuál es necesario llevar a cabo la clasificación entre vehículos y no vehículos, para la cual existen diferentes enfoques de modelado. Estos enfoques los podemos dividir en dos categorías principales: modelos generativos y modelos discriminativos [81]. Para el problema de clasificación se parte de un escenario en el que cada imagen se describe mediante un vector  $X$  (que podría comprender las intensidades de los píxeles, o algún conjunto de características extraídas de la imagen), el cuál se tiene que asignar a una de las clases (vehículos, no vehículos),  $k = 1, 2$ . Desde la teoría de la decisión elemental [4], sabemos que la caracterización más completa de la solución se expresa en términos del conjunto de probabilidades a posteriori  $p(k|X)$ . Una vez que sabemos estas probabilidades es sencillo asignar la imagen  $X$  a una clase particular para minimizar la pérdida esperada (por ejemplo, si deseamos minimizar el número de errores de clasificación asignamos  $X$  a la clase que tiene la probabilidad posterior más grande). Los modelos discriminativos introducen un modelo paramétrico para las probabilidades a posteriori,  $p(k|X)$ , y se deducen los valores de los parámetros de un conjunto de datos de entrenamiento etiquetados. Esto se puede realizar mediante el uso de estimaciones puntuales de los parámetros utilizando la máxima verosimilitud, o mediante el cálculo de las distribuciones sobre los parámetros en un entorno bayesiano (por ejemplo mediante el uso de la inferencia variacional). Por otra parte, los modelos generativos modelan la distribución conjunta  $p(k, X)$  de las imágenes y etiquetas. Esto se puede hacer, por ejemplo, mediante el aprendizaje de las probabilidades a priori de la clase,  $p(k)$  y las densidades de la clase condicional  $p(X|k)$  por separado. Las probabilidades a posteriori requeridas se obtienen a continuación, utilizando el teorema de Bayes 2.2:

$$p(k|X) = \frac{p(X|k)p(k)}{\sum_j p(X|j)p(j)} \quad (2.2)$$

donde la suma del denominador viene dada sobre todas las clases.

Resumiendo, las aproximaciones discriminativas buscan la clasificación de nuevas muestras en vehículos o no-vehículos, teniendo en cuenta la distribución de las entradas, mientras que los modelos generativos tratan de encontrar las distribuciones de probabilidad asociadas a las características de cada una de las clases. Cada modelo presenta sus ventajas y desventajas. Hay muchos estudios recientes que tratan la comparación de estos modelos. En [82], las conclusiones

extraídas indican que aunque los modelos discriminativos obtienen un mayor rendimiento, los modelos generativos tienen un comportamiento mucho más rápido. Resultados muy similares se obtuvieron también en [83], pero se mostraron sobre un conjunto de datos simulados, por lo que sólo es cierto cuando los modelos son adecuados para los datos. En ambos trabajos se observó que a medida que aumenta el número de datos, los modelos discriminativos obtienen mejor rendimiento que los modelos generativos.

La clasificación mediante modelos generativos incluye entre otros:

- Análisis discriminante lineal (*Linear Discriminant Analysis, LDA*): Asumen que la densidad de una muestra  $X$  dada para cada clase  $k$  sigue una distribución gaussiana.
- Gaussianas mixtas: El uso de una única gaussiana para modelar las densidades de las clases puede no ser suficiente. Por lo tanto, se modela la densidad dentro de la clase de  $X$  mediante una mezcla de gaussianas.
- Naive Bayes: es un clasificador probabilístico basado en el teorema de Bayes que asume, que la presencia o ausencia de una característica particular, no está relacionada con la presencia o ausencia de cualquier otra característica, es decir, que todas las variables son independientes.

Por otro lado, la clasificación de modelos discriminativos incluye entre otros:

- Perceptrón: Es un clasificador lineal que realiza sus predicciones sobre una función lineal dada por la combinación de un conjunto de pesos y el vector de características que describe la entrada, en la cuál los elementos del conjunto de entrenamiento se procesan de uno en uno, intentando corregir todos los errores encontrados en éste.
- Máquina de vector soporte (*Support Vector Machine, SVM*): Es un algoritmo que maximiza el margen entre el hiperplano que define la frontera de decisión y las muestras del conjunto de entrenamiento.
- Redes neuronales (*Neural Networks, NN*): Son modelos computacionales inspirados en los sistemas nerviosos centrales de los animales (en especial el cerebro). Por lo general, se presentan como sistemas de "neuronas" interconectadas que pueden calcular valores usando las entradas para proporcionar información a través de la red.

Usualmente los modelos discriminativos, son los métodos más utilizados para la clasificación entre vehículos y no-vehículos, se debe a que éstos usualmente obtienen un mayor rendimiento en tareas discriminativas que los modelos generativos [4]. Entre todos los métodos discriminativos que nos encontramos las máquinas de vector soporte, son el clasificador más utilizado en todos los trabajos de la literatura que se refieren al campo de la detección de vehículos. De hecho, las

SVM implican un número mucho más reducido de parámetros ajustables que las redes neuronales. Además, se generalizan mejor, ya que realizan la minimización del riesgo con el fin de maximizar la generalización de nuevos datos [84]. Muchos trabajos de los comentados en los apartados anteriores, [80, 73, 74, 79] entre otros, llevan a cabo la clasificación mediante SVM. No obstante, también encontramos trabajos como [59], en los cuales se discute sobre el uso de SVM o NN para la clasificación. Por otra parte, podemos encontrar también trabajos relativos al campo de la detección de vehículos, que utilizan otro tipo de clasificadores como el algoritmo de los K vecinos más cercanos [85] o *random forest* [68].

Aunque los métodos discriminativos consiguen mejor rendimiento, también podemos encontrar en la literatura algunos trabajos que utilizan métodos generativos. Por ejemplo, en [86] donde asumiendo modelos gaussianos se obtiene una función de densidad probabilidad del vector de características mediante la aplicación del algoritmo EM (*Expectation Maximization*). En [85], el proceso de evaluación de la verosimilitud se basa en la probabilidad conjunta estimada de los coeficientes PCA y del análisis de componentes independientes (*Independent Component Analysis, ICA*). Se asumen funciones gaussianas multivariadas para modelar las correspondientes probabilidades de verosimilitud. Para aliviar la carga computacional, propone un modelo alternativo más eficiente de verosimilitud que se basa en el uso de un modelos de pesos de gaussianas mixtas, en el cuál los parámetros se estiman mediante el algoritmo EM.

## 2.6. Conclusiones

En la literatura como se ha observado, existen multitud de métodos relacionados con los sistemas de detección de vehículos. Si bien, aportan una gran cantidad de información y en muchos casos obtienen rendimientos bastante altos, no son los esperados por la comunidad ITS, por lo que sigue siendo un campo abierto a la investigación. Dentro de la detección de vehículos, los métodos basados en aprendizaje suponen uno de los grupos más importantes de investigación en la actualidad, encontrando una gran cantidad de trabajos relacionados con la extracción de características, ya sean explícitas o implícitas. Aunque las características explícitas suponen una comprensión más simple y generalmente implican un coste computacional menor, éstas se están viendo relegadas a sistemas de combinación de características. Sin embargo, las características implícitas al obtener mayores rendimientos son la base central de los sistemas basados en aprendizaje. Por otro lado, las máquinas de vector soporte se están consolidando como la mejor opción en sistemas de detección de vehículos. Por último, uno de los grandes problemas que se encuentran es, que cada trabajo utiliza bases de datos diferentes, y generalmente suelen estar constituidas por un bajo número de muestras, donde apenas existe variabilidad entre ellas. Así, el uso de bases de datos unificadas, de gran tamaño y alta variabilidad se hacen necesarias para este tipo de sistemas.

## Capítulo 3

# Arquitectura del sistema

En este capítulo se han abordado los diferentes aspectos del diseño del sistema de verificación de vehículos, así como la explicación de los métodos adoptados en este PFC.

Para ello analizamos las diferentes etapas del sistema, donde primero se muestran los métodos para la extracción de características y por último los métodos para la clasificación.

### 3.1. Introducción

El principal objetivo de este trabajo es la búsqueda de un sistema robusto para la verificación de hipótesis. Para ello se ha llevado a cabo un estudio de diferentes tipos de descriptores para la extracción de características, y el uso de diferentes clasificadores para conseguir la mejor verificación de vehículos. De todas las posibles características que aparecen en la literatura para la caracterización de vehículos, en este PFC se han explorado las características explícitas de simetría y bordes y las características implícitas PCA y HOG. En cuanto al tipo de clasificadores utilizados, se han elegido un clasificador lineal basado en el método de perceptrón, un clasificador bayesiano y un clasificador SVM donde se exploran diferentes núcleos lineales y no lineales. Este PFC utiliza para la implementación de estos métodos programación en MATLAB, haciendo uso de la librería PRTOOLS.

La Figura 3.1 muestra un esquema general del sistema que se ha llevado a cabo en este PFC. La primera etapa consiste en la extracción de características y se divide en características explícitas (sección 3.2) y características implícitas (sección 3.3). La segunda etapa, es la de etapa de clasificación (sección 3.4), donde se exponen los clasificadores utilizados.

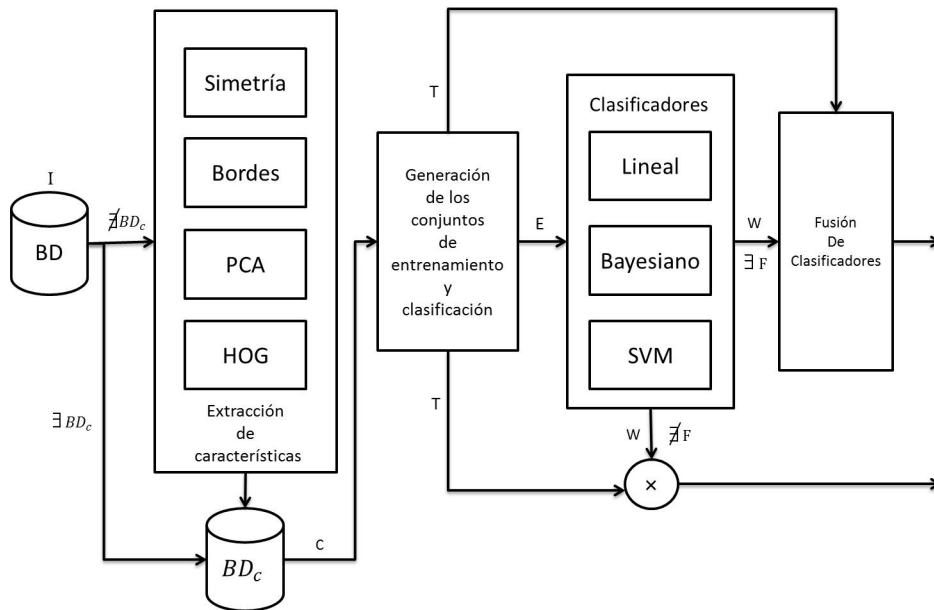


Figura 3.1: Esquema general del sistema implementado.

## 3.2. Extracción de característica explícitas

El uso de este tipo de características está muy extendido en verificación, debido principalmente a que estas características se pueden apreciar a simple vista, lo que hace que el conocimiento de la escena y de los problemas que pueden ocurrir sea mucho más intuitivo.

### 3.2.1. Simetría

Una de las características para la caracterización de vehículos es la simetría, en particular, la simetría con respecto al eje vertical. Esta característica se encuentra en todos los tipos de vehículos, pudiendo ser detectada no sólo en vistas frontales de la parte delantera o trasera del vehículo, sino en imágenes tomadas desde diferentes puntos de vista, aunque para la óptima obtención de la simetría son necesarias imágenes de plano frontal. Sin embargo, aunque las imágenes donde aparecen vehículos presenten un alto grado de simetría, imágenes donde no aparecen vehículos también pueden presentarlo y por lo tanto, llevar a clasificaciones erróneas. Algunos ejemplos de esta situación son, imágenes con un alto grado de homogeneidad, como las que muestran gran parte del pavimento de la carretera o imágenes que debido a la baja iluminación aparecen mayoritariamente en negro. Esto ha hecho que en los métodos llevados a cabo para la extracción de simetría hayan tratado de solucionarlos teniendo en cuenta el ancho de ventana relativo al vehículo.

Para el sistema de extracción de simetría llevado a cabo en este PFC se han explorado dos

métodos. El primer método tiene dos etapas muy diferenciadas. La primera etapa es la búsqueda del eje de simetría, para ello partimos de [51] donde se determina el eje de simetría mediante los bordes verticales de la imagen binarizada. Obtenido el eje y fijando un valor de ventana relativo al vehículo, se calcula el valor de simetría que presenta la imagen. Para ello se parte de la propuesta realizada en [52], donde el valor de simetría se calcula a partir de las funciones par e impar que corresponden a los valores de intensidad de la imagen.

El segundo método de cálculo de simetría se basa en [87]. Al igual que en el método anterior buscamos la simetría de la imagen respecto a un eje vertical mediante la intensidad en escala de grises escaneando la imagen de forma horizontal. Sin embargo, a diferencia del método anterior, en este caso no disponemos de un valor fijo precalculado del eje, ni se fija el ancho de ventana. Como en el caso anterior buscamos las funciones par e impar de nuestra imagen, para obtener el contenido de energía. Estas funciones en este caso estarán en función del eje y el ancho de ventana. Así, el vector final de simetría está constituido por todos los posibles valores de simetría que se pueden dar en la imagen y el valor de simetría de la imagen elegido, será el máximo valor del vector.

El vector de características que se obtiene para cada una de las imágenes en ambos métodos tiene una dimensión y está compuesto por el valor de simetría calculado,  $S$ , como se muestra en ecuación 3.1.

$$\mathbf{c} = (S) \tag{3.1}$$

### 3.2.2. Bordes

La extracción de bordes es una herramienta muy útil para verificación de ciertos objetos, entre ellos, los vehículos, debido principalmente a su estructura rectangular. Estos bordes estructurados, proporcionan información característica y discriminativa de los vehículos frente a otros objetos o, en nuestro caso, a regiones de escenas de tráfico en las cuales no se encuentran vehículos. No obstante, aunque los bordes verticales y horizontales proporcionan información de la estructura de los vehículos, el uso de los bordes horizontales en imágenes que contienen regiones donde no se encuentran vehículos, constan al igual que las imágenes donde aparecen vehículos, bordes horizontales muy pronunciados como consecuencia por ejemplo, de la presencia de la línea del horizonte, sombras, etc., lo que puede dar lugar a información engañosa que dificulte la discriminación entre las dos clases. En este trabajo se ha llevado a cabo un estudio de la extracción únicamente de bordes verticales, los cuáles son muy característicos en los vehículos.

Para la composición de los vectores de características basados en bordes verticales, se han seguido diferentes estrategias. La Figura 3.2 es un ejemplo de los pasos seguidos en la obtención de bordes verticales. Como podemos observar los bordes verticales caracterizan adecuadamente la estructura de los vehículos.



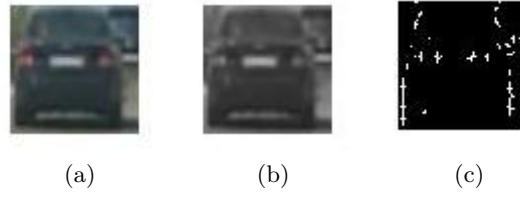


Figura 3.2: Ejemplo de los pasos seguidos en la extracción de bordes verticales. (a) Imagen original de la que partimos. (b) Imagen en nivel de gris. (c) Imagen de bordes verticales aplicando binarización.

Las estrategias llevadas a cabo se realizan como se muestra en la Figura 3.2 sobre la imagen en gris y son las siguientes:

1. Clasificación mediante el histograma horizontal completo de los gradientes verticales con y sin binarización.
2. Clasificación mediante subconjuntos compuestos por agrupaciones del histograma horizontal con y sin binarización.
3. Clasificación mediante la agrupación del pico máximo existente en la parte derecha del histograma y del pico máximo existente en la parte izquierda del histograma con y sin binarización.

Los gradientes utilizados para todas estas estrategias se obtienen mediante la aplicación de una máscara de Sobel [54]. Así, el vector de características resultante para cada una de las imágenes con el uso del histograma completo está compuesto como muestra la ecuación 3.2.

$$\mathbf{c} = (g_1, g_2, \dots, g_N) \quad g_n = \sum_{m=1}^M |\nabla I(x_n, y_m)| \quad (3.2)$$

donde  $M$  es el alto y  $N$  el ancho de la imagen. En el caso de las otras estrategias el vector resultante  $\mathbf{c}$  esta compuesto por un subconjunto del vector descrito en la ecuación 3.2. Por lo tanto, para la estrategia 2 el vector resultante es el indicado por la ecuación 3.3:

$$\mathbf{c} = (sg_1, sg_2, \dots, sg_R) \quad sg_r = \frac{1}{S} \sum_{n=S \cdot (r-1)+1}^{S \cdot r} g_n \quad (3.3)$$

donde  $S = \frac{N}{R}$  indica el número de valores que constituyen el subconjunto. Por último, el vector resultante de la estrategia 3 está determinado por la ecuación 3.4:

$$\mathbf{c} = (g_{max,i}, g_{max,d}) \quad (3.4)$$

### 3.3. Extracción de característica implícitas

Las características implícitas son en la actualidad las características más utilizadas para la verificación de vehículos. Esto se debe a que obtienen rendimientos más altos que las características explícitas. Sin embargo, el entendimiento de estos sistemas es más complejo.

#### 3.3.1. Análisis de componentes principales

El análisis de componentes principales, o PCA, es una técnica utilizada en aplicaciones de reducción de dimensionalidad, compresión de datos con pérdidas, extracción de características y visualización de datos. PCA tiene comúnmente dos definiciones que dan lugar al mismo algoritmo. Se puede definir como la proyección ortogonal de los datos sobre un espacio lineal de menor dimensionalidad, conocido como subespacio principal, en el cuál se maximiza la varianza de la proyección de los datos. También se puede definir como la proyección lineal que minimiza el coste medio de la proyección, definido este como la distancia media de los cuadrados de las diferencias entre los datos y su proyección [4].

Por lo tanto, el principal objetivo del Análisis de Componentes Principales es reducir la dimensión de un conjunto de variables manteniendo la mayor cantidad de información posible. El buen funcionamiento de PCA se debe a la eliminación de información redundante existente, por lo que con un subconjunto del espacio es capaz de obtener mejores resultados que con el espacio completo. Normalmente, las variables originales están correladas, la transformación realizada por PCA convierte las variables en incorreladas para una mejor interpretación de los datos. La reducción de la dimensionalidad se logra mediante una proyección lineal de los datos, formando un nuevo conjunto de variables que se denominan Componentes Principales (*Principal Components, PC*). Estas componentes se obtienen mediante rotación de ejes en el espacio de características y se corresponden con las direcciones en las que los datos tienen máxima varianza. Como ya se ha mencionado con anterioridad, estas componentes serán una combinación lineal de los datos originales, estarán incorreladas y ordenadas de forma que las primeras componentes sean las que tienen mayor información proveniente de los datos originales.

Más en detalle, si partimos de un conjunto de datos  $\mathbf{x} = [x_1, \dots, x_N]^T$ , el subespacio principal está formado por los primeros  $M$  autovectores de la matriz de covarianza dada por:

$$S = \frac{1}{N} \sum_{n=1}^N (x_n - \bar{x})(x_n - \bar{x})^T \quad (3.5)$$

donde  $\bar{x}$  es la media del conjunto de datos, definida por:

$$\bar{x} = \frac{1}{N} \sum_{n=1}^N x_n \quad (3.6)$$

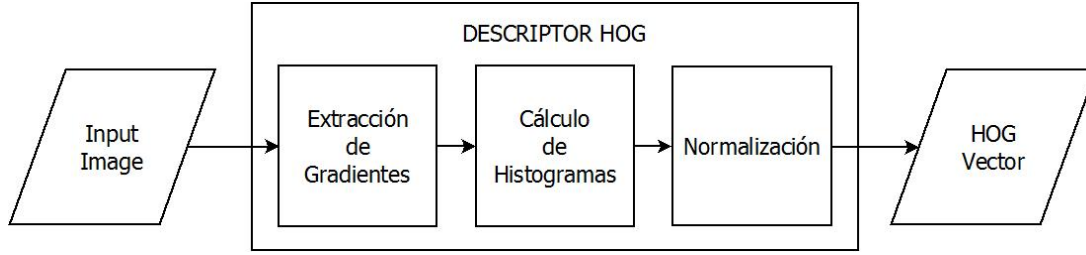


Figura 3.3: Sistema de obtención del descriptor HOG

Las direcciones de los autovectores son las conocidas como componentes principales. La representación de la imagen,  $I_n(x, y)$ , mediante la aplicación de PCA, viene dada por un vector de longitud  $R \times C$ :

$$\mathbf{z}_n = \{z_{n,k}\}_{k=1}^{R \times C}, \quad z_{n,y+C(x-1)} = I_n(x, y), \quad 1 \leq x \leq R, 1 \leq y \leq C \quad (3.7)$$

Primero se calcula la media del conjunto de imágenes:

$$\bar{z}_n = \frac{1}{N} \sum_{n=1}^N z_n \quad (3.8)$$

Luego, se derivan de la matriz de covarianza, los autovectores  $\{u_i\}_{i=1}^D$  y los autovalores  $\{\lambda_i\}_{i=1}^D$ , y solamente se retienen las componentes principales. Finalmente, se sustrae la media de cada imagen y el resultado se proyecta sobre el subespacio principal para proporcionar la representación de la imagen en el nuevo espacio de características:

$$f_n^{(i)} = (\mathbf{z}_n - \bar{z}_n)^T u_i \quad (3.9)$$

La representación final basada en PCA de los datos esta compuesta por las proyecciones sobre todas las componentes principales,  $\mathbf{f}_n = (f_n^{(1)}, \dots, f_n^{(M)})$ .

### 3.3.2. Histogramas de gradientes orientados

Los Histogramas de Gradientes Orientados (HOG, [74]) se basan en la idea de que objetos, como vehículos, pueden ser caracterizados mediante su apariencia. Estos descriptores obtienen la orientación del gradiente de cada píxel obteniendo de esta manera la distribución espacial del objeto. En la Figura 3.3 se distinguen las diferentes etapas de este descriptor.

El descriptor HOG resultante se compone del cómputo de histogramas locales de orientación de los gradientes de la imagen, la cuál ha sido previamente dividida en celdas. Por lo tanto la idea principal es implementar un sistema que divida la imagen en pequeñas regiones, a las que llamaremos celdas, y obtener para cada una de ellas un histograma a partir de la orientación de los gradientes de los píxeles que la forman. Por tanto la combinación de los histogramas

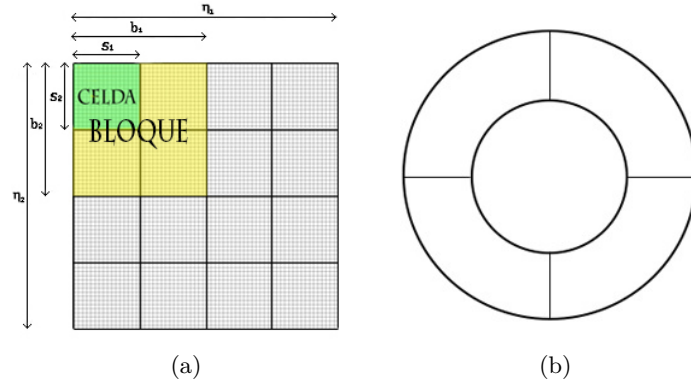


Figura 3.4: Ejemplo de mallado de la imagen. (a) Configuración R-HOG. Donde se muestran los parámetros número de pixel por celda,  $s_1 \times s_2$ , número de celdas en la imagen,  $\eta_1 \times \eta_2$  y tamaño de bloque,  $b_1 \times b_2$ . (b) Configuración C-HOG

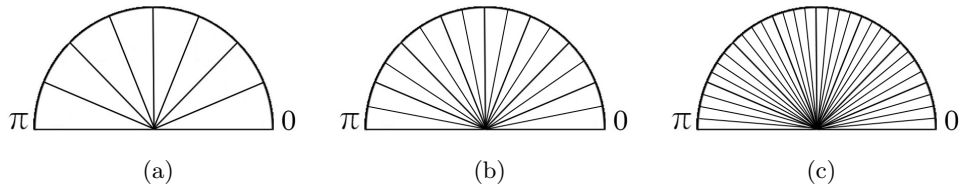


Figura 3.5: Posibles configuraciones cambiando el número de bins de orientación. El rango utilizado es  $[0, \pi]$  el cuál se divide uniformemente en  $\beta$  intervalos. (a) 8 (b) 16 (c) 32 bins de orientación.

generados para cada una de las celdas proporcionará la representación de la imagen en el espacio de características.

Partimos, como en todos los casos, de una imagen de entrada, la cuál puede ser de la clase vehículo o de la clase no-vehículo. En primer lugar se procede a una fase donde se van a extraer los gradientes de cada uno de los pixeles de la imagen y se obtiene a partir de estos, su módulo y ángulo. Para la extracción de gradientes podemos utilizar diferentes métodos mediante el uso de operadores tales como:  $[-1, 0, 1]$ , Sobel, Prewitt, etc.. El rango de los ángulos que se utiliza es  $[0, \pi]$ , debido a que el signo del gradiente es típicamente ignorado. Luego, se divide la imagen en celdas, donde se deben definir parámetros como el número de celdas,  $\eta_1 \times \eta_2$ , mostrado en la Figura 3.4a o el número de bins de orientación que se va a utilizar,  $\beta$ . En la Figura 3.5 encontramos un ejemplo de las diferentes configuraciones que podemos obtener cambiando el número de bins de orientación. Haciendo uso de estos parámetros junto al módulo y ángulo se procede al cálculo de los histogramas. La división de la imagen en celdas se puede realizar usando diferentes estructuras. Las principales topologías que se suelen seguir cuando implementamos descriptores HOG son rectangulares (R-HOG) o circulares (C-HOG), como se muestra en la

Figura 3.4. La configuración R-HOG es la más utilizada para vehículos debido a su naturaleza rectangular, la cuál se puede encontrar en el contorno, las ventanas, la matricula, las luces, etc., por lo tanto será la configuración elegida para nuestro estudio. En nuestro caso se parte de imágenes normalizadas a un tamaño fijo, debido a este hecho y siguiendo las recomendaciones que se encuentran en la literatura se va a hacer uso de celdas cuadradas, por lo tanto  $\eta = \eta_1 = \eta_2$ . Por último se añade una etapa de normalización que nos permite eliminar, en la medida de lo posible, los cambios de iluminación y las sombras que aparezcan en las imágenes. Para realizar esta normalización se procede a la agrupación de las celdas en estructuras de mayor tamaño que llamaremos bloques. Por lo tanto es necesario definir el tamaño de bloque,  $b_1 \times b_2$ , un ejemplo de esto lo encontramos en la Figura 3.4a. En muchos trabajos se propone el solapamiento de los bloques para obtener un sistema más robusto. Mediante el uso de cualquier normalización estándar como pueden ser, la normal L1, L2, etc., se procede a la normalización de cada bloque por separado. Finalmente la concatenación de estos bloques normalizados proporcionan el vector resultante HOG.

### 3.4. Clasificación

La correcta elección de un clasificador que se adapte a la distribución de los datos puede suponer mejoras importantes en el rendimiento del sistema. Esto hace que la fase de clasificación tome un papel fundamental en los sistemas de verificación de hipótesis.

#### 3.4.1. Clasificador lineal

El clasificador lineal que se ha utilizado es un clasificador discriminativo basado en el método de Perceptrón, el cual describimos a continuación:

Se parte de un vector de entrada  $x$  que primero se transforma usando una transformación no lineal cuyo resultado es un vector  $\phi(x)$ . Este se utiliza para construir un modelo lineal generalizado de la forma:

$$y(x) = f(\omega^T \phi(x)) \quad (3.10)$$

donde  $f(\omega^T \phi(x)) = 1$  para  $\omega^T \phi(x) \geq 0$  y  $f(\omega^T \phi(x)) = -1$  para  $\omega^T \phi(x) < 0$ .

Ahora es necesario obtener el vector de pesos (parámetro  $\omega$ ) mediante la minimización de la función de error. La función de error que utiliza este algoritmo es la conocida como criterio de Perceptrón. Por lo tanto, buscamos un vector de pesos que cumpla:

- Para la clase C1  $\rightarrow \omega^T \phi(x) > 0$
- Para la clase C2  $\rightarrow \omega^T \phi(x) < 0$

Para  $t \in \{-1,1\}$  se quiere que todos los patrones satisfagan:  $\omega^T \phi(x)t_n > 0$ .

Este criterio asocia el error cero a los patrones que se clasifiquen correctamente, mientras que para los patrones que se clasifiquen de manera errónea trata de minimizar la cantidad  $-\omega^T \phi(x)t_n$ . Ahora, se puede formular el criterio de Perceptrón como:

$$E_p(\omega) = - \sum_{n \in M} \omega^T \phi_n t_n \quad (3.11)$$

donde  $M$  es el conjunto de datos mal clasificados.

La función de error es lineal por tramos. Si se aplica el algoritmo del gradiente de descenso estocástico (*stochastic gradient descent algorithm*) a esta función de error se obtiene un nuevo vector de pesos.

$$\omega^{(\Gamma+1)} = \omega^\Gamma - \eta \nabla E_p(\omega) = \omega^\Gamma + \eta \phi_n t_n \quad (3.12)$$

donde  $\eta$  es el parámetro de tasa de aprendizaje y  $\Gamma$  es un número entero que indexa los pasos del algoritmo.

A raíz de esta función se pueden dar dos situaciones; clasificación correcta, el vector de pesos no sufre cambios y clasificación incorrecta, en la cuál se añade  $\phi(x)$  al vector de pesos de la clase C1 y se resta  $\phi(x)$  al vector de pesos de la clase C2. Los problemas que se encuentran en la utilización de este clasificador son los siguientes: muy lento, existen datos que no son linealmente separables y no generaliza bien para más de dos clases. Este último problema no afectaría a nuestro sistema, debido a que clasifica entre dos clases: vehículos y no vehículos. El funcionamiento de este clasificador se puede visualizar en la Figura 3.6.

### 3.4.2. Clasificador bayesiano

Las distribuciones gaussianas tienen propiedades analíticas importantes, pero sufren cuando se quiere modelar un sistema de datos reales [4]. Como consecuencia el clasificador bayesiano elegido es un clasificador MoG (*Mixture of Gaussians*), que utiliza la unión de dos gaussianas, lo que ofrece un mejor modelado de la distribución de los datos pertenecientes a las clases. Para cada clase del conjunto de datos de entrenamiento se obtiene una estimación de densidad gaussiana usando  $K$  componentes de la mezcla, en nuestro caso dos. La estimación de la función de densidad de probabilidad para el conjunto de datos de entrenamiento se realiza en cada una de las clases y luego se ponderan y combinan mediante sus probabilidades anteriores, como se observa en la ecuación 3.13, obteniendo así un único clasificador.

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathbf{N}(\mathbf{x} | \mu_k, \Sigma_k) \quad (3.13)$$

donde cada densidad gaussiana  $\mathbf{N}(\mathbf{x} | \mu_k, \Sigma_k)$  se denomina componente y tiene su propia media

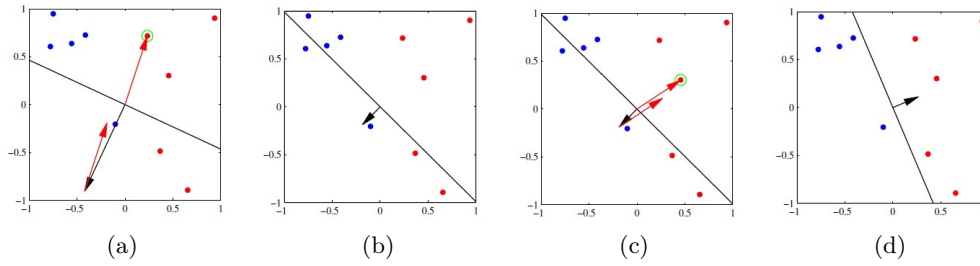


Figura 3.6: Ejemplo de funcionamiento del clasificador lineal basado en el algoritmo de Perceptrón [4]. Las imágenes muestran los datos de dos clases (roja y azul) en el espacio de características de dos dimensiones  $(\phi_1, \phi_2)$ . (a) Muestra los parámetros iniciales del vector de pesos  $\omega$ , ilustrados por una flecha negra junto con la correspondiente frontera de decisión (línea negra), donde la flecha apunta hacia la región de decisión que clasifica como perteneciente a la clase roja. El dato indicado mediante un círculo verde está mal clasificado y por lo tanto su vector de características se añade al vector de pesos actual, dando como nueva frontera de decisión la mostrada en (b). En (c) se muestra la siguiente muestra mal clasificada indicada por el círculo verde, y este nuevo vector de características es de nuevo añadido al vector de pesos dando la frontera de decisión mostrada en (d), para la cuál se clasifican todas las muestras correctamente.

$\mu_k$  y covarianza  $\Sigma_k$ . los parámetros  $\pi_k$  se denominan coeficientes de la mezcla (*mixing coefficients*), los cuáles satisfacen la propiedad  $0 \leq \pi_k \leq 1$ , junto con  $\sum_{k=1}^K \pi_k = 1$ , para poder ser probabilidades válidas.

Los objetos no etiquetados se obvian, a menos que todo el conjunto de datos de entrenamiento esté sin etiquetar. Si el número de gaussianas utilizado es '1', la estimación de la densidad se realiza mediante la media y la covarianza. Si el número de gaussianas es mayor que '1', como ocurre en nuestro caso, para realizar la estimación de la densidad de probabilidad se utiliza el algoritmo EM (*Expectation-Maximization*).

**Algoritmo EM para mezcla de gaussianas** Dado un modelo de mezcla de gaussianas, el objetivo es maximizar la función de verosimilitud con respecto a los parámetros (donde están comprendidos la media y la covarianza de las componentes y los coeficientes de la mezcla).

El paso 1 consiste en inicializar las medias  $\mu_k$ , las covarianzas  $\Sigma_k$  y los coeficientes de la mezcla  $\pi_k$ , y evaluar el valor inicial del logaritmo de la verosimilitud.

A continuación se realiza el paso 2 conocido como esperanza (*Expectation*), donde se evalúan las responsabilidades (*responsibilities*,  $\gamma$ ) usando los valores actuales de los parámetros:

$$\gamma(z_{nk}) = \frac{\pi_k \mathbf{N}(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathbf{N}(x_n | \mu_j, \Sigma_j)} \quad (3.14)$$

El paso 3 es el de maximización (*Maximization*), en el que se realiza la re-estimación de los

parámetros usando la responsabilidad actual:

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) x_n \quad (3.15)$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (x_n - \mu_k^{new})(x_n - \mu_k^{new})^T \quad (3.16)$$

$$\pi_k^{new} = \frac{N_k}{N} \quad (3.17)$$

donde

$$N_k = \sum_{n=1}^N \gamma(z_{nk}) \quad (3.18)$$

Por último, el paso 4 evalúa de nuevo el logaritmo de la verosimilitud,

$$\ln p(\mathbf{X}|\mu, \Sigma, \pi) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathbf{N}(x_n|\mu_k, \Sigma_k) \right\} \quad (3.19)$$

comprobando si ha convergido alguno de los parámetros o el logaritmo de la verosimilitud. Si no se satisface el criterio de convergencia se vuelve al paso 2.

### 3.4.3. SVM

Las máquinas de vectores soporte fueron desarrolladas por Vapnik basándose en la idea de minimización del riesgo estructural (*Structural Risk Minimization, SRM*) [5]. La función de las SVM es proyectar los puntos de entrada a un espacio de características de mayor dimensión, para luego encontrar un hiperplano que los separe y maximice el margen entre las clases. Han sido desarrolladas como una técnica robusta para la clasificación y regresión aplicada a grandes conjuntos de datos con ruido. Una propiedad importante de las SVM es que cualquier solución local es también un óptimo global, esto se debe a la forma de determinar los parámetros, ya que corresponde a un problema de optimización convexo. Estos clasificadores son máquinas de decisión y no proporcionan probabilidades a posteriori. También ofrecen la ventaja de que pueden ser utilizadas para resolver problemas tanto lineales, como no lineales, lo que hace de ellas una opción muy interesante y flexible para la clasificación.

#### 3.4.3.1. Clasificador SVM lineal

En el caso de que las clases que se van a clasificar sean linealmente separables, las SVM mediante hiperplanos separan los datos de entrada en estas dos clases, etiquetándolas como



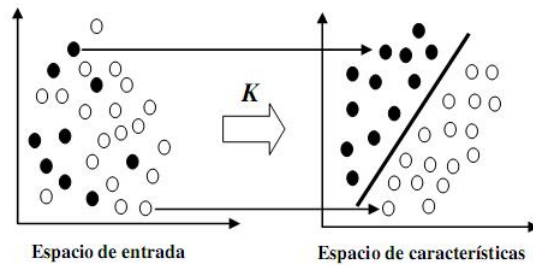


Figura 3.7: Ejemplo de cambio del espacio original de los datos de entrada a un nuevo espacio de característica, de una SVM no linealmente separable realizado mediante una función kernel [5].

$\{-1, +1\}$ . De todos los hiperplanos que pueden separar las clases sólo uno es óptimo, entendido como aquel que maximiza la distancia al valor de entrada más cercano (maximización del margen).

### 3.4.3.2. Clasificador SVM no lineal

En la mayoría de los casos que se dan en sistemas basados en situaciones reales, los datos no se pueden separar linealmente. Las SVM también son capaces de abordar estas situaciones. Las máquinas de vectores soporte no lineales se pueden dividir en dos categorías, SVM con margen máximo (*Maximum Margin*) y SVM con margen blando (*Soft Margin*).

**SVM con margen máximo** Cuando los datos de entrada no pueden separarse linealmente mediante un hiperplano óptimo en el espacio de entrada original, se hace necesario recurrir a una transformación no lineal del espacio. En muchas ocasiones este nuevo espacio puede ser separado linealmente aplicando los razonamientos de maximización del margen dados para la SVM lineal. La transformación de los datos de un espacio inicial a otro de mayor dimensión se logra mediante el uso de la función *kernel*.

Una función núcleo o *kernel* es un producto interno en el espacio de características, que tiene su equivalente en el espacio de entrada.

$$k(x, x') = \langle \phi(x), \phi(x') \rangle \quad (3.20)$$

donde  $k$  es una función simétrica definida positiva que cumple las condiciones de Mercer [88]. De manera gráfica, se puede observar en la Figura 3.7, como la función kernel permite realizar la separación y el traslado de los datos al espacio de características.

El problema de optimización de las SVM se define por un modelo de programación cuadrática con restricciones. Este problema de optimización se resuelve introduciendo los multiplicadores de Lagrange, así los datos de entrenamiento sólo aparecerán en forma de una combinación de

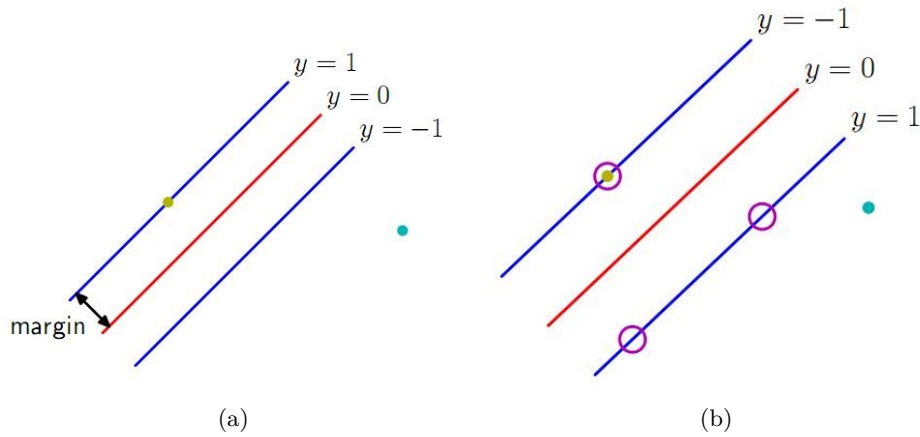


Figura 3.8: El margen se define como la distancia perpendicular entre la frontera de decisión y el punto de los datos más cercano. (a) muestra como maximizar el margen y conduce a una elección particular del límite de decisión: la ubicación de este límite está determinada por un subconjunto de puntos conocidos como vectores de soporte, que en (b) están indicados mediante círculos.

vectores, y la resolución del problema se pueden hallar resolviendo el problema dual dado por las ecuaciones y haciendo uso de las condiciones de Karush Kuhn Tucker (KKT) [4].

Más detalladamente, si tenemos dos clases utilizando un modelo lineal:

$$y(x) = w^T \phi(x) + b \quad (3.21)$$

donde  $\phi(x)$  denota la transformación del espacio de entrada de características,  $b$  el parámetro de parcialidad y  $w$  el vector de pesos.

El conjunto de entrenamiento se compone de  $N$  vectores de entrada  $x_1, \dots, x_N$  con sus correspondientes valores objetivo:  $t_1, \dots, t_N$ . Donde  $t_N \in \{-1, 1\}$  y el vector  $x$  se clasifica acorde al signo de  $y(x)$ . Este conjunto de entrenamiento es separable linealmente en el espacio de características. Por lo tanto, por definición existe al menos una opción de los parámetros  $w$  y  $b$  que den una función de la forma 3.21, que satisfaga  $y(x_n) > 0$  para  $t_n = 1$  y  $y(x_n) < 0$  para  $t_n = -1$ , obteniendo  $t_n y(x_n) > 0$  para todos los puntos de entrenamiento.

La frontera de decisión se obtiene mediante la maximización del margen. El concepto de margen se define como la distancia más pequeña entre la frontera de decisión y cualquiera de las muestras. En la Figura 3.8, se puede observar una explicación visual del concepto de margen. En este punto, se escoge la solución por la cuál los puntos de datos están correctamente clasificados, es decir,  $t_n y(x_n) > 0$  para todo  $n$ . Así la distancia del punto  $x_n$  a la superficie de decisión se determina por:

$$\frac{t_n y(x_n)}{\|w\|} = \frac{t_n (w^T \phi(x_n) + b)}{\|w\|} \quad (3.22)$$

El margen viene dado por la distancia perpendicular al punto más cercano  $x_n$  del conjunto de datos y se quiere determinar los parámetros  $w$  y  $b$  para maximizar esta distancia. Así la solución del margen máximo la encontramos resolviendo:

$$\arg \max_{w,b} \left\{ \frac{1}{\|w\|} \min_n [t_n (w^T \phi(x_n) + b)] \right\} \quad (3.23)$$

La solución directa del problema es muy compleja, por lo que se convierte el problema en uno equivalente más sencillo de resolver (explicado detalladamente en [4]). Este problema requiere la maximización de  $\|w\|^{-1}$ , lo que es equivalente a minimizar  $\|w\|^2$ . Así, este problema queda reducido a resolver:

$$\arg \min_{w,b} \left\{ \frac{1}{2} \|w\|^2 \right\} \quad (3.24)$$

Ahora ha desaparecido el parámetro  $b$ . Los cambios de  $\|w\|$  tienen que ser compensados por los cambios de  $b$ , por lo que se determina a través de restricciones. Para resolver el problema de la optimización de las restricciones, se introduce el multiplicador de Lagrange  $a_n \geq 0$ , dando la función de Lagrange:

$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{n=1}^N a_n \{t_n (w^T \phi(x_n) + b) - 1\} \quad (3.25)$$

donde  $a_n = (a_1, \dots, a_N)^T$ . Derivando  $L(w, b, a)$  con respecto a  $w$  y  $b$  e igualando a cero obtenemos las condiciones:

$$w = \sum_{n=1}^N a_n t_n (\phi(x_n)) \quad (3.26)$$

$$0 = \sum_{n=1}^N a_n t_n \quad (3.27)$$

Eliminando  $w$  y  $b$  de  $L(w, b, a)$  y usando las condiciones anteriores conseguimos una representación dual del problema del margen máximo, donde tenemos que maximizar:

$$\tilde{L}(a) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(x_n, x_m) \quad (3.28)$$

donde  $a_n \geq 0$ ,  $n = 1, \dots, N$  y  $\sum_{n=1}^N a_n t_n = 0$ . La función del núcleo se define por:  $k(x, x') = \phi(x)^T \phi(x')$ . Por lo tanto, se permite que el modelo sea formulado usando núcleos y que el clasi-

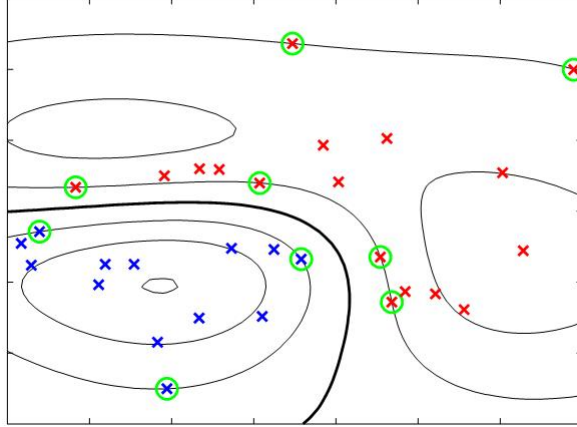


Figura 3.9: Ejemplo de síntesis de datos de dos clases para dos dimensiones que muestran los contornos de la constante  $y(x)$  obtenidos a partir de una SVM con una función kernel gaussiana[4]. También se muestran los límites de decisión, los límites de los márgenes, y los vectores soporte.

ficador del margen máximo pueda aplicarse de manera eficiente a las características espaciales en las que la dimensión exceda la dimensión inicial del espacio de características, incluyendo infinitas características espaciales. Para la clasificación de nuevas muestras usando el modelo de entrenamiento, se evalúa el signo de  $y(x)$  definido en la ecuación 3.21. Se puede expresar en términos de  $\{a_n\}$  y de la función del núcleo, mediante la sustitución de  $w$ , obteniendo:

$$y(x) = \sum_{n=1}^N a_n t_n k(x, x_n) + b \quad (3.29)$$

donde esta optimización tiene asociadas las siguientes condiciones de Karush-Kuhn-Tucker (KKT):  $a_n \geq 0$ ;  $t_n y(x_n) - 1 \geq 0$ ;  $a_n \{t_n y(x_n) - 1\} = 0$ . Esto se cumple para cada punto de datos, incluso si  $a_n = 0$  o  $t_n y(x_n) = 1$ . Habiendo resuelto el problema de programación cuadrática y encontrado el valor de  $a$ , se puede determinar el valor de  $b$ :

$$t_n \left( \sum_{m \in S} a_m t_m k(x_n, x_m) + b \right) = 1 \quad (3.30)$$

donde  $S$  denota el conjunto de índices de los vectores soporte. Podemos resolver esta ecuación para  $b$ , utilizando un vector de soporte  $x_n$  arbitrario, y así una solución más estable es:

$$b = \frac{1}{N_S} \sum_{n \in S} \left( t_n - \sum_{m \in S} a_m t_m k(x_n, x_m) \right) \quad (3.31)$$

donde  $N_S$  es el número total de vectores soporte. Podemos observar un ejemplo del uso de este clasificador en la Figura 3.9.

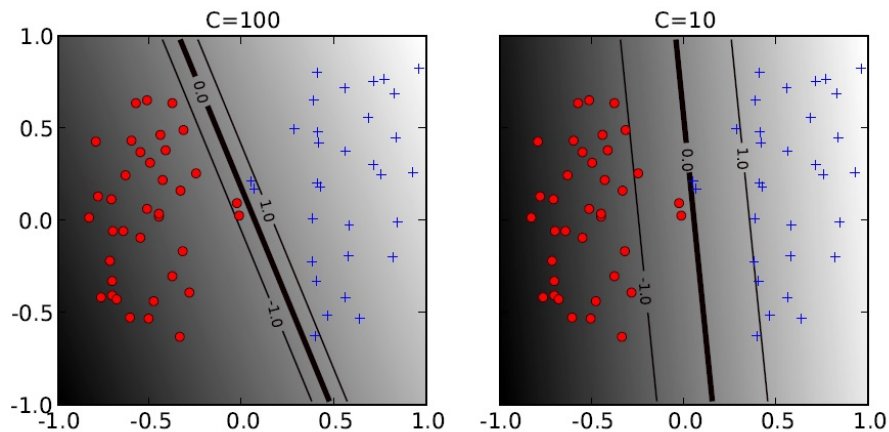


Figura 3.10: Ejemplo del efecto que provoca la constante  $C$  en la SVM con margen blando. Un valor más bajo de  $C$ , provoca un margen más amplio y más permisivo, mientras que valores más altos de  $C$  reducen el margen haciendo que este sea más restrictivo [6].

**SVM con margen blando** Este tipo particular de las SVM se utiliza cuando no es posible encontrar una transformación de los datos que permita separarlos linealmente, ya sea en el espacio de entrada o en el espacio de características. Por lo tanto, este clasificador se ocupa de aquellos casos donde existen datos de entrada erróneos, con ruido o existe un alto solapamiento de las clases en los datos de entrenamiento, donde se puede ver afectado el hiperplano clasificador. Por esta razón se cambia un poco la perspectiva y se busca el mejor hiperplano clasificador que pueda tolerar el ruido en los datos de entrenamiento.

Esto se logra relajando las restricciones, introduciendo variables de holgura no negativas  $\xi_n \geq 0$ . La función de optimización que se debe solucionar para el modelo de SVM con margen blando incluye un término de regularización que depende de las variables de holgura, el cuál, a su vez depende de la magnitud de las mismas y del margen. Además, este término incluye una constante,  $C$ , que determina la holgura del margen blando. La elección de valor del parámetro  $C$  y del tipo de núcleo influyen en el funcionamiento de las SVM. El procedimiento que se utiliza para la resolución de este problema es el mismo que en el caso anterior: viene dado por la búsqueda de las funciones de Lagrange, para lo que se construye un Lagrangiano y se resuelve con el problema dual. La función a maximizar es la misma que para el caso de margen máximo, a diferencia de la restricción  $0 \leq a_n \leq C$ . Esto implica que los datos o patrones que cumplen la condición de tener valores  $a_n \geq C$  tienen el mismo comportamiento en la SVM con margen máximo, es decir, las SVM con margen máximo se pueden obtener con  $C = \infty$ . Esto significa que mientras más grande es el valor de  $C$ , más estricta es la SVM permitiendo errores, penalizándolos con mayor rigurosidad. Como se observa en la Figura 3.10.

De manera más detallada, el proceso que sigue este tipo de SVM se explica a continuación. Para cada punto de entrenamiento se introduce una variable de holgura  $\xi_n$ . Esto permite cambiar

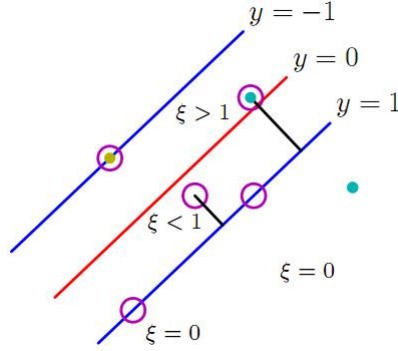


Figura 3.11: SVM con margen blando. Ejemplo de la clasificación de los datos en clases solapadas [4].

la SVM para que los puntos de los datos puedan estar en el “lado equivocado” del límite del margen, pero con una penalización que aumenta con la distancia a ese límite. Esto se define por  $\xi_n = 0$  para puntos que estén en o dentro del límite correcto del margen y  $\xi_n = |t_n - y(x_n)|$  para los otros. Así un punto que esté en el límite de decisión  $y(x_n) = 0$  tendrá  $\xi_n = 1$  y puntos con  $\xi_n > 1$  estarán mal clasificados. Por lo tanto, la limitación  $t_n(w^T \phi(x_n) + b) \geq 1$ ,  $n = 1, \dots, N$  queda reemplazada por:

$$t_n y(x_n) \geq 1 - \xi_n \quad (3.32)$$

donde  $n = 1, \dots, N$  y  $\xi_n \geq 0$ . Así, puntos para los cuales  $\xi_n = 0$ , están correctamente clasificados y están en el margen o en el lado correcto del margen. Puntos para los cuales  $0 < \xi_n \leq 1$ , se encuentran dentro del margen pero en el lado correcto de la frontera de decisión. Y por último, aquellos puntos con  $\xi_n > 1$  se encuentran en el lado equivocado de la frontera de decisión y están mal clasificados. En la Figura 3.11 se puede observar un ejemplo de esto.

Aunque las variable de holgura faciliten la clasificación para distribuciones de clases solapadas, este marco sigue siendo sensible a los valores lejanos de la frontera porque la penalización de clasificación errónea aumenta linealmente con  $\xi$ . El objetivo es maximizar el margen. Para ello primero se minimiza:

$$C \sum_{n=1}^N \xi_n + \frac{1}{2} \|w\|^2 \quad (3.33)$$

donde  $C > 0$  controla la compensación entre el margen y la penalización de la variable de holgura. Ahora se quiere minimizar 3.33 sujeto a la restricción 3.32 junto a  $\xi_n \geq 0$ :

$$L(w, b, a) = \frac{1}{2} \|w\|^2 + C \sum_{n=1}^N \xi_n - \sum_{n=1}^N a_n \{t_n y(x_n) - 1 + \xi_n\} - \sum_{n=1}^N \mu_n \xi_n \quad (3.34)$$

donde  $\mu_n, a_n \geq 0$  son multiplicadores de Lagrange con el correspondiente conjunto de condiciones KKT dado por:  $a_n \geq 0$ ;  $t_n y(x_n) - 1 + \xi_n \geq 0$ ;  $a_n(t_n y(x_n) - 1 + \xi_n) = 0$ ;  $\mu_n \geq 0$ ;  $\xi_n \geq 0$ ;  $\mu_n \xi_n = 0$ .

Ahora se optimiza  $w$ ,  $b$  y  $\xi_n$  obteniendo:

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{n=1}^N a_n t_n \phi(x_n) \quad (3.35)$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{n=1}^N a_n t_n = 0 \quad (3.36)$$

$$\frac{\partial L}{\partial \xi_n} = 0 \Rightarrow a_n = C - \mu_n \quad (3.37)$$

Se utilizan estos resultados para eliminar  $w, b$  y  $\xi_n$  del Lagrangiano, quedando:

$$\tilde{L}(a) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(x_n, x_m) \quad (3.38)$$

Hay que minimizar 3.38 con respecto a las variables duales  $\{a_n\}$ , sujeto a:

$$0 \leq a_n \leq C \quad (3.39)$$

$$\sum_{n=1}^N a_n t_n = 0 \quad (3.40)$$

Una solución obtenida mediante promediado es:

$$b = \frac{1}{N_M} \sum_{n \in M} \left( t_n - \sum_{m \in M} a_m t_m k(x_n, x_m) \right) \quad (3.41)$$

Se ve que este caso es idéntico al caso de clases separadas, excepto que las restricciones son algo diferentes.

**Kernels** La elección adecuada del kernel, como ya se ha comentado anteriormente, tiene un efecto significativo en el rendimiento obtenido en clasificación. La frontera de decisión generada por el kernel puede ser más discriminativa entre las dos clases dependiendo del kernel utilizado. Esto hace que la elección adecuada del kernel y de sus parámetros sea fundamental a la hora de la clasificación. Algunos ejemplos de los kernels que nos podemos encontrar pueden ser:

- Kernel lineal:  $k(x, x') = (x x')$
- Kernel polinomial:  $k(x, x') = (x x' + 1)^P$

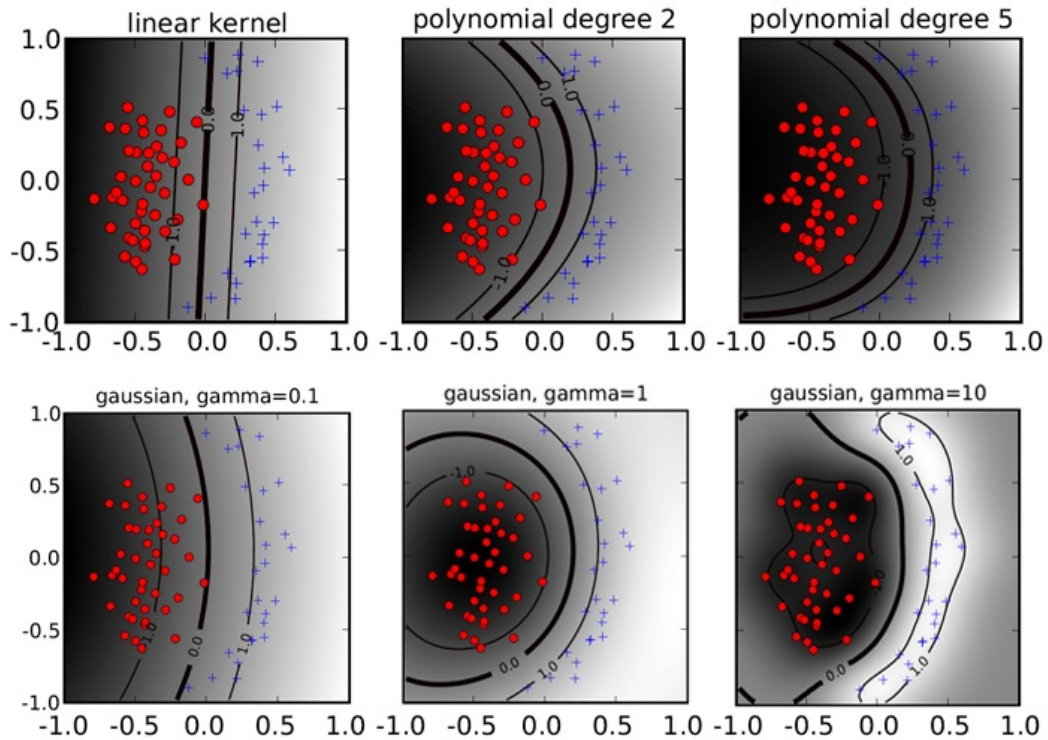


Figura 3.12: Ejemplos de la generación de la frontera de decisión usando en primer lugar un kernel polinomial cambiando el grado del polinomio (parámetro  $P$ ) y en segundo lugar un núcleo gaussiano cambiando su parámetro de amplitud (parámetro  $\gamma$ ) [6].

- Kernel exponencial:  $k(x, x') = \exp\left(\frac{-\|x-x'\|}{P}\right)$
- Kernel gaussiano:  $k(x, x') = \exp(-\gamma\|x - x'\|^2)$ , donde  $\gamma$  es el parámetro que controla el ancho de la gaussiana.

Para entender el uso de los kernels para la generación de la frontera de decisión en el espacio de características se exponen a continuación en la Figura 3.12 algunos ejemplos. Vemos como en este caso tanto el núcleo polinomial como el núcleo gaussiano se adaptan mejor a los datos cuanto mayores son sus parámetros. Este hecho no siempre ocurre, ya que la estimación de la frontera de decisión depende de la distribución de los datos. Por lo tanto, se puede concluir que una clasificación satisfactoria no depende únicamente de la elección apropiada del núcleo, sino que también es necesaria una adaptación de sus parámetros al problema.





# Capítulo 4

## Desarrollo

### 4.1. Introducción

En este capítulo se describen los métodos y experimentos llevados a cabo para el sistema propuesto de verificación de vehículos (descrito en el capítulo 3). En las siguientes secciones, primero se ha descrito la base de datos, así como los métodos utilizados para la clasificación (sección 4.2), posteriormente se ha descrito cada uno de los métodos que se ha utilizado para la verificación de vehículos, exponiendo los resultados conseguidos; los primeros métodos descritos son los basados en características explícitas, simetría (sección 4.3) y bordes (sección 4.4), por último se presentan los métodos basados en características implícitas, PCA (sección 4.5) y HOG (sección 4.6).

### 4.2. Base de Datos y método de clasificación

En los procesos de máquinas de aprendizaje supervisado, la selección de un conjunto de entrenamiento adecuado supone un papel importante a la hora de obtener buenos resultados en la clasificación. Las bases de datos existentes y utilizadas en la mayoría de los trabajos suelen ser pequeñas e insuficientes, no obteniendo una alta variabilidad de perspectivas de las imágenes. Por ejemplo, la base de datos MIT CBCL sólo comprende imágenes de vista frontal de los vehículos, mientras que otras bases de datos como la UIUC utiliza imágenes de vistas totalmente laterales. La base de datos utilizada en este trabajo, GTI [89], fue introducida recientemente por Arróspide et al., la cuál está orientada específicamente a verificación de vehículos. Esta base de datos es significativamente grande y con una alta variabilidad (en color, forma, condiciones meteorológicas, etc.), consiste en 4000 imágenes de vehículos y 4000 de no-vehículos, que son el resultado de un proceso de generación de hipótesis, es decir, que contienen lo que se espera que sea el vehículo detectado. Estas hipótesis se han escalado a imágenes de tamaño fijo de  $64 \times 64$  píxeles, además se han categorizado en diferentes vistas (frontal, lateral izquierda, lateral derecha

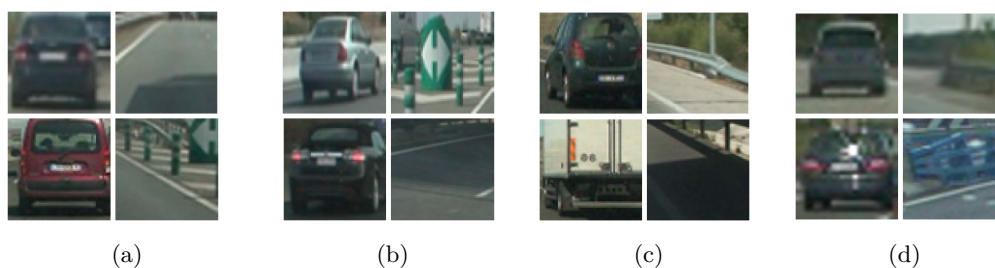


Figura 4.1: Ejemplo Base de datos GTI. (a) Vista frontal; (b) Vista lateral izquierda; (c) Vista lateral derecha; (d) Vista lejana.

y lejana). Las imágenes han sido seleccionadas de diferentes secuencias de carretera, tomadas mediante una vídeo-cámara instalada en la parte delantera del vehículo, y tomadas en diferentes ciudades (Madrid, Turin y Bruselas), en diferentes fechas del año, a diferentes horas y bajo diferentes condiciones meteorológicas, consiguiendo así una alta variabilidad de los vehículos obtenidos mediante diferentes condiciones de adquisición. Esta base de datos es suficientemente grande y tiene una variabilidad mayor que otras bases de datos. Además, a diferencia de bases de datos donde las imágenes de no-vehículos son tomadas arbitrariamente, en ésta se han obtenido de las mismas secuencias de carretera que las imágenes de vehículos, permitiendo así una mejor caracterización de la clase no-vehículo. Adicionalmente, esta base de datos también incluye imágenes recogidas de otras bases de datos (Database Caltech y Database TU Graz-02). Se puede ver un ejemplo de la base de datos en la Figura 4.1.

El proceso para la clasificación de los vectores resultantes generados a partir de la base de datos mediante los métodos expuestos en las secciones sucesivas, se realiza para cada una de las vistas. La base de datos, como ya se ha explicado, está dividida en 4000 imágenes de vehículos y 4000 de no-vehículos, grupos que a su vez están divididos en 4 subgrupos de 1000 imágenes correspondientes a las 4 vistas consideradas. Nuestro sistema considera cada una de las vistas por separado para conseguir un mejor rendimiento en clasificación, así tendríamos 2000 imágenes de ambas clases para realizar la clasificación. Los experimentos que se han llevado a cabo se basan en 5 iteraciones de la metodología de validación cruzada utilizando el 50 % de la base de datos. Así, el rendimiento del sistema es la media del obtenido en cada una de esas 5 iteraciones, donde el grupo de aprendizaje utilizado para el entrenamiento del clasificador que está compuesto de 1000 imágenes y el grupo de test compuesto de las 1000 imágenes restantes, es diferente para cada una de estas iteraciones.

El rendimiento de nuestros sistemas se mide mediante la tasa de acierto, que es el porcentaje de vehículos y no-vehículos correctamente clasificados. No obstante, existen otras medidas que son útiles para conocer el comportamiento de nuestro sistema como son: la precisión y el recall.

$$Prec = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \quad (4.1)$$

donde  $TP$  indica los vehículos correctamente clasificados,  $FP$  las imágenes que no son vehículos clasificadas como vehículos y  $FN$  los vehículos clasificados erróneamente como no vehículos. Por lo tanto, la precisión nos indica el porcentaje de vehículos correctamente detectados que se encuentran entre todos los clasificados como vehículos y el recall indica el porcentaje de los vehículos que se han clasificado correctamente. En los sistemas llevados a cabo en este PFC se busca la mayor tasa de aciertos, sin embargo en algunas ocasiones, para determinar qué configuración o sistema funciona mejor, como por ejemplo cuando comparamos dos sistemas que ofrecen tasas de acierto similares, es necesario hacer uso del valor de recall.

### 4.3. Desarrollo del sistema basado en simetría

#### 4.3.1. Simetría

La simetría es una característica que esta presente en multitud de objetos, incluidos los vehículos. Esta característica, al ser explícita, es de gran ayuda, ya que solamente mirando la imagen podemos determinar si la imagen es simétrica. Por lo tanto, teniendo en cuenta esto, podemos pensar que al ser los vehículos simétricos esta característica nos va a ser de gran ayuda a la hora de clasificar una imagen y determinar si es vehículo o no. La simetría que nos encontramos en los vehículos es respecto al eje vertical. Para el desarrollo del sistema de extracción de simetría se van a implementar diferentes métodos. Primero vamos a partir de un método por el cuál se busca el eje de simetría vertical de la imagen de entrada en escala de grises y, con el eje fijado, se calcula el valor de simetría de la imagen completa. Por otro lado, el segundo método consistirá en ir calculando el valor de simetría de la imagen, variando tanto el eje de simetría como el ancho, y a partir de dichos valores buscar el máximo valor de simetría obtenido, que será elegido como valor de simetría de la imagen completa.

#### 4.3.2. Método 1

Para la realización de este método primero vamos a efectuar la detección del eje vertical de simetría de la imagen mediante la búsqueda de bordes verticales basado en [51]. Partimos de los principios de la transformada de Hough para implementar el sistema de detección del eje vertical, dicho sistema se basa en un esquema de votaciones. La idea general es que por cada par de píxeles que no sean cero de una misma línea, es decir, cada par de píxeles donde se encuentre un borde, se vota su eje potencial. De todos los posibles candidatos para ser el eje vertical, se elige aquel que ha obtenido el mayor valor de votos. El algoritmo utilizado para detectar el eje vertical sigue:

1. Aplicación del operador de Sobel sobre la imagen para obtener una imagen binaria IMG.
2. Selección de dos umbrales ( $thr_{low}$ ,  $thr_{up}$ ) para determinar el posible ancho del vehículo.
3. Construcción de un acumulador  $A[col]$ , inicializando todos sus elementos a cero.
4. Escaneado de la imagen IMG:

FOR cada fila

FOR cada par de pixeles que no sean cero  $(x_1, y_1)$  y  $(x_2, y_2)$  en la misma fila DO

{

$distancia = |x_1 - x_2|;$

$eje = \frac{(x_1+x_2)}{2};$

IF ( $distancia > thr_{low}$  y  $distancia < thr_{up}$ )

$A[eje] = A[eje] + 1;$

}

5. Búsqueda del máximo elemento de  $A[col]$ , donde  $col$  indica el eje potencial de simetría del objeto.

Obtenido el eje es necesario hallar el valor de simetría. Para ello nos basamos en [52], donde los valores de intensidad de una fila pueden considerarse como una función de una dimensión  $I(x)$ . Toda función  $I(x)$  puede separarse en la suma de su parte par  $I_p(x)$  y su parte impar  $I_i(x)$ :

$$I(x) = I_p(x) + I_i(x) \quad (4.2)$$

donde

$$I_p(x) = \frac{I(x) + I(-x)}{2} \quad (4.3)$$

$$I_i(x) = \frac{I(x) - I(-x)}{2} \quad (4.4)$$

El eje de simetría,  $x_s$ , es el calculado por el método descrito anteriormente y este valor va a ser un valor fijo para todos los cálculos posteriores que se van a realizar. Vamos a considerar, al igual que para el eje, un ancho del objeto fijo,  $w$ , siendo este el valor más próximo desde el eje al extremo más cercano de la imagen. Antes de realizar el cálculo de la parte par e impar se va a proceder a realizar un filtrado de promediado sobre los valores de intensidad con la máscara siguiente:

$$\frac{1}{5}[1 \ 1 \ 1 \ 1 \ 1] \quad (4.5)$$

Definimos la función par para un intervalo dado de ancho  $w$ , sobre el eje de simetría  $x_s$  como:

$$E(u) = \begin{cases} \frac{1}{2}(I(x_s + u) + I(x_s - u)) & -\frac{w}{2} \leq u \leq \frac{w}{2} \\ 0 & \text{resto} \end{cases} \quad (4.6)$$

De la misma manera podemos definir la función impar:

$$O(u) = \begin{cases} \frac{1}{2}(I(x_s + u) - I(x_s - u)) & -\frac{w}{2} \leq u \leq \frac{w}{2} \\ 0 & \text{resto} \end{cases} \quad (4.7)$$

Para cada par fijo  $\{x_s, w\}$ , el significado de cada función  $E(u)$  o  $O(u)$  viene dado por el contenido de energía. Sin embargo, se observa un problema cuando comparamos la energía de la media de la función impar que es siempre cero con la media de la función par que siempre tiene valores positivos. Para compensar este problema, se normaliza la función par para obtener media cero:

$$E_n(u) = E(u) - \frac{1}{w} \int_{-\frac{w}{2}}^{\frac{w}{2}} E(v) dv \quad (4.8)$$

Con las definiciones de las funciones  $E_n(u)$  y  $O(u)$  podemos determinar, para cada fila de la imagen, el valor de simetría dado por:

$$s = \frac{\int E_n(u)^2 du - \int O(u)^2 du}{\int E_n(u)^2 du + \int O(u)^2 du} \quad (4.9)$$

donde  $s$  indica el valor de simetría respecto a un eje y un ancho dados. Esta medida sigue la propiedad:

$$-1 \leq s \leq 1 \quad (4.10)$$

Por lo tanto,  $s = 1$  indica simetría ideal,  $s = 0$  indica asimetría y  $s = -1$  indica antisimetría. Para obtener un nivel de simetría entre  $[0, 1]$  se normalizan estos valores:

$$s = \frac{1}{2}(s + 1) \quad (4.11)$$

Finalmente el valor de simetría de la imagen completa,  $S_F$ , es el sumatorio del nivel de simetría de cada fila dividido por el número total de filas de nuestra imagen:

$$S_F = \frac{1}{N} \sum_{n=1}^N S_n \quad (4.12)$$

donde  $S = [s_1, \dots, s_N]^T$  y  $N$  es el número de filas.

### 4.3.3. Método 2

Al igual que en el método anterior vamos a partir de los valores de intensidad de grises de la imagen,  $I(x)$ . El eje de simetría potencial,  $x_p$ , se mueve a lo largo de un intervalo que definimos como posible intervalo de los píxeles de la imagen donde puede encontrarse el eje,  $\frac{w_{max}}{4} \leq x_p \leq (w_{max} - \frac{w_{max}}{4})$ . Además se consideran todos los posibles anchos correspondientes a ese eje, siendo el menor ancho posible,  $\frac{w_{max}}{2}$ . Para mover el origen de una función  $I(u)$  a cualquier eje potencial, se usa la sustitución:  $u = x - x_p$ . Ahora se puede definir la función par de  $I(x) = I(x_p + u)$  para un intervalo dado de ancho  $w$ , sobre el eje de simetría  $x_p$  como:

$$E(u, x_p, w) = \begin{cases} \frac{1}{2}(I(x_p + u) + I(x_p - u)) & -\frac{w}{2} \leq u \leq \frac{w}{2} \\ 0 & \text{resto} \end{cases} \quad (4.13)$$

De la misma manera se puede definir la función impar:

$$O(u, x_p, w) = \begin{cases} \frac{1}{2}(I(x_p + u) - I(x_p - u)) & -\frac{w}{2} \leq u \leq \frac{w}{2} \\ 0 & \text{resto} \end{cases} \quad (4.14)$$

Al igual que en el caso anterior se normaliza la función par para obtener media cero:

$$E_n = E(u, x_p, w) - \frac{1}{w} \int_{-\frac{w}{2}}^{\frac{w}{2}} E(v, x_p, w) dv \quad (4.15)$$

Con las definiciones de las funciones  $E_n(u, x_p, w)$  y  $O(u, x_p, w)$  podemos determinar, para cada fila de la imagen, la matriz de los valores de simetría dada por:

$$S_1(x_p, w) = \frac{\int E_n(u, x_p, w)^2 du - \int O(u, x_p, w)^2 du}{\int E_n(u, x_p, w)^2 du + \int O(u, x_p, w)^2 du} \quad (4.16)$$

donde la dependencia con la fila ha sido eliminada de la ecuación para una mayor simplicidad. Ahora, se realiza la media de los valores de simetría de  $S_1$  a lo largo de la dirección vertical,  $y$ . El rango de valores de simetría resultante se encuentra entre  $[-1, 1]$ , por lo tanto para obtener una escala apropiada, tendremos que normalizar estos valores al rango  $[0, 1]$ , obteniendo:

$$S_2(x_p, w) = \frac{1}{R} \sum_{y=1}^R \frac{1}{2} (1 + S_1(x_p, w, y)) \quad (4.17)$$

donde  $R$  es el número total de filas.

Como por cada eje potencial,  $x_p$ , hemos tenido en cuenta todos los posibles anchos y recorrido cada fila de la imagen,  $S_2$  es un vector cuya longitud va a ser igual a todas las posibles combinaciones entre eje y ancho y donde los valores de este vector van a ser todos los posibles valores de simetría que tiene la imagen. Así, el valor final de simetría de la imagen será el máximo



Figura 4.2: Elección del eje dependiendo del ancho de ventana.

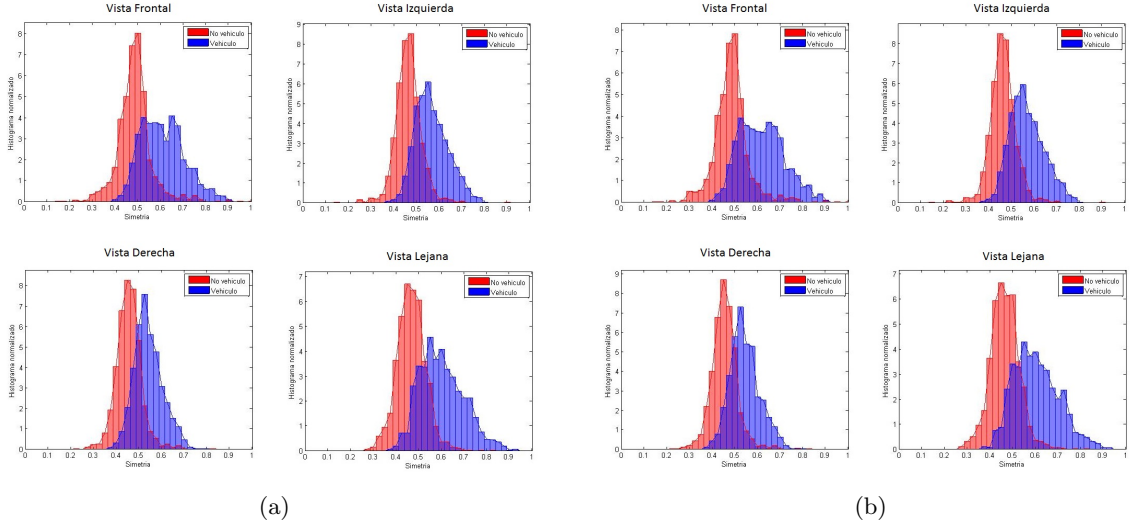


Figura 4.3: Representación de las distribuciones de los valores de simetría de las clases vehículo y no-vehículo. (a) Eje constituido por un solo píxel en todos los casos (b) Eje constituido por uno o dos píxeles.

valor del vector  $S_2$ :

$$S = \max(S_2) \quad (4.18)$$

Para hacer más robusto el sistema y dado que nuestras imágenes son de 64x64 píxeles, se propone añadir al método anterior una mejora por la cual se va a tener en cuenta el ancho del eje, es decir, si el ancho es un número impar de píxeles, el eje estará constituido por un píxel, mientras que si el ancho es par, el eje constará de dos píxeles. Esto se puede ver en la Figura 4.2.

Para entender mejor el efecto que se produce al usar un eje compuesto por un solo píxel y la variación entre la elección de un píxel o dos para determinar el eje, se presenta en la Figura 4.3, la representación de la distribución de las clases, vehículo y no-vehículo, para las diferentes vistas. Como se puede apreciar, las diferencias son muy pequeñas, aunque se consigue, con este pequeño cambio, que disminuya el área de solape, reduciendo el número de imágenes clasificadas erróneamente. No obstante, mediante esta representación de la distribución de las clases dadas



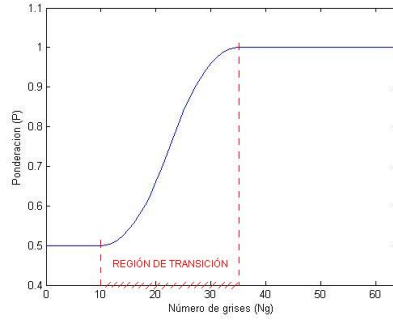


Figura 4.4: Sigmoide utilizada para la penalización del valor de simetría.

por el valor de simetría, queda patente que existe una gran región tanto de la clase vehículo como de la clase no vehículo que comparten el mismo valor de simetría. Para intentar mejorar estas distribuciones y obtener mejores resultados a la hora de la clasificación se propone incluir una penalización en el valor de simetría mediante una sigmoide creada a partir de la media del número de valores de gris diferentes que existen en una imagen. Basándose esta mejora en la suposición de que una imagen tiene una mayor probabilidad de ser un vehículo cuanto mayor es la variabilidad de grises en ella. Por lo tanto, esta penalización se propone porque existen imágenes que no siendo vehículos, dada su alta homogeneidad (fondo, típicamente carretera o cielo) tienen un alto valor de simetría.

Para cada fila de la imagen se calcula el número de niveles de intensidad que contiene y el valor medio de la variación de grises se define como la media de estos niveles; a partir de estos valores se define la sigmoide que se va a utilizar para modelar el comportamiento que queremos que tenga el valor de simetría, teniendo en cuenta la variación de grises en la imagen. La sigmoide se definirá como se expone en la Figura 4.4. Como se puede apreciar, tenemos que definir la región de transición, para ello debemos decidir el número mínimo y máximo de grises que vamos a utilizar para penalizar, estos valores se han determinado empíricamente analizando el valor máximo y mínimo de intensidad de los conjuntos de imágenes vehículos y no-vehículos por separado. Definida la función sigmoide, ponderaremos el valor de simetría, multiplicando este valor por el valor resultante de introducir el número medio de variación de grises en la sigmoide.

$$S_p = S \cdot P \quad (4.19)$$

Con este nuevo valor de simetría obtenemos la distribución de las clases mostrada en la Figura 4.5. Como podemos observar, con respecto a las distribuciones de las clases antes de introducir la penalización, a priori hay una mayor separación entre clases, por lo que se espera obtener mejores resultados en clasificación. No obstante, la vista lejana sigue teniendo una gran

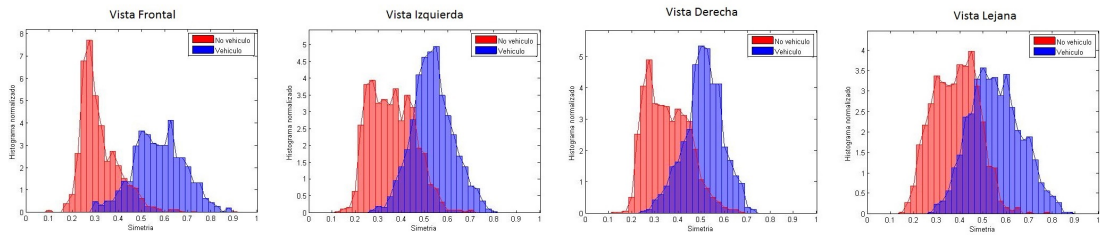


Figura 4.5: Representación de las distribuciones de simetría para cada una de las clases haciendo uso de la penalización mediante sigmoide.

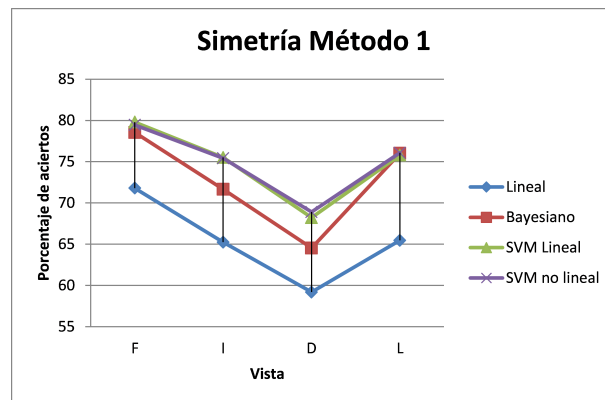


Figura 4.6: Rendimiento del sistema de simetría haciendo uso del método 1 para cada una de las vistas, utilizando diferentes clasificadores.

región común para ambas clases lo que impedirá un alto rendimiento en su clasificación.

#### 4.3.4. Clasificación

En esta sección se realiza un estudio del rendimiento de nuestro sistema mediante el uso de diferentes clasificadores, por una parte se hace uso de un clasificador generativo como es un clasificador de mezcla de gaussianas y por otra parte se van a probar tres clasificadores discriminativos diferentes: un clasificador lineal basado en perceptrón, un clasificador SVM con núcleo lineal y un clasificador SVM con núcleo no lineal.

Primero se realiza la clasificación para el método 1 explicado con anterioridad. La Figura 4.6 muestra la comparación del uso de estos cuatro clasificadores en nuestro sistema de simetría. Como vemos el clasificador que mejor rendimiento obtiene es la SVM no lineal, pero con resultados muy similares empleando el núcleo lineal, lo que nos indica que la distribución de los datos en el espacio de características pueden separarse mediante una frontera cuasilineal. Por otro lado, el espacio original indica que los datos no pueden ser separados linealmente debido a que existen zonas donde los datos de ambas clases se encuentran mezclados, por ello el clasificador lineal utilizado ofrece peor rendimiento a favor del clasificador generativo bayesiano. También

Simetría Método 1				
Vista	Bayesiano		SVM no lineal	
	TA	R	TA	R
F	78,54	85,56	79,46	78,64
I	71,66	81,32	75,42	70,04
D	64,54	80,68	68,90	62,04
L	76,02	65,36	76,08	63,02

Tabla 4.1: Comparación clasificador bayesiano y SVM no lineal.

hay que destacar que las vistas frontal y lejana obtienen un rendimiento similar del clasificador bayesiano y el de la SVM no lineal, lo cuál puede ser una ventaja en términos de rapidez del sistema.

En la Tabla 4.1 se exponen los resultados de la clasificación mediante la SVM no lineal y el clasificador bayesiano para realizar un análisis más exhaustivo. Los valores de la tasa de acierto de las vistas frontal y lejana, como observábamos en el gráfico, son muy parecidos para los dos clasificadores. Sin embargo, el valor del recall para ambas vistas es más bajo en la SVM no lineal, lo que nos indica que hay un mayor número de coches mal clasificados, por lo tanto la mejor opción en estos casos sería el clasificador bayesiano. En el caso de las vistas laterales, la tasa de acierto obtenida mediante el clasificador bayesiano es notablemente menor, sin embargo el valor del recall es muy superior al obtenido por el clasificador SVM no lineal, esto nos indica que el clasificador bayesiano hace una clasificación de la clase coche mejor, pero incluye en ésta muchas imágenes que no contienen vehículos, por otra parte el clasificador SVM no lineal está más equilibrado en los fallos de ambas clases. En este tipo de sistemas, como ya se ha explicado con anterioridad, prima obtener el mayor número de aciertos en la clasificación de la clase vehículos. Así podemos concluir que, haciendo uso del método 1 de simetría, el clasificador que se adapta mejor sería el clasificador bayesiano.

Para el método 2, los resultados de la clasificación se exponen en la Figura 4.7. La realización del sistema de verificación de vehículos mediante la extracción de simetría por el método 2 nos muestra en primer lugar como, sin aplicarle ninguna penalización, se obtienen rendimientos muy similares para las cuatro vistas haciendo uso de los clasificadores bayesiano, SVM lineal y SVM no lineal, obteniendo sus valores más elevados con el uso del clasificador SVM no lineal. Los bajos resultados obtenidos mediante el clasificador lineal nos indican que el espacio de entrada no puede separarse linealmente a diferencia del espacio de características producido por el clasificador SVM, que distribuye las clases de manera que la frontera de decisión se aproxime a una frontera lineal. Por otro lado, el uso de la penalización mejora globalmente los resultados, de manera específica los de la vista frontal y lateral izquierda, consiguiendo los mejores resultados con el clasificador SVM no lineal.

Al igual que en el caso del método 1, en la Tabla 4.2 se muestran los valores más destacados

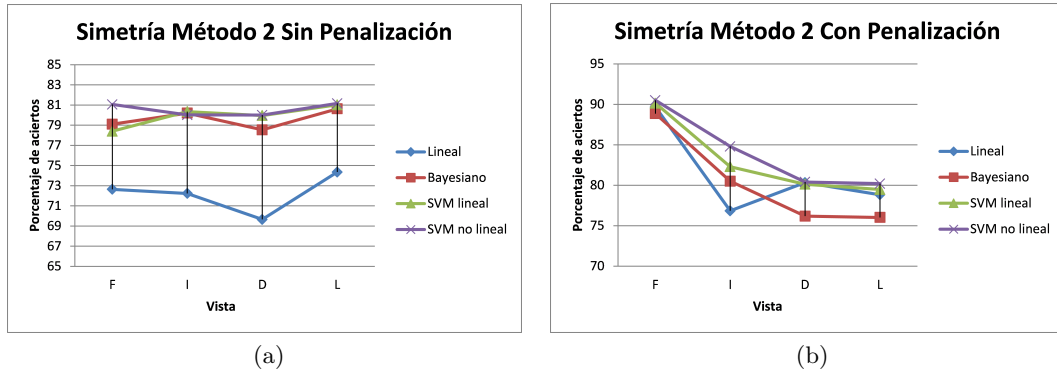


Figura 4.7: Rendimiento del sistema de simetría haciendo uso del método 2 para cada una de las vistas, utilizando diferentes clasificadores. (a) Método 2 sin aplicar la penalización. (b) Método 2 aplicando la penalización.

Vista	Método 2 Sin Penalización				Método 2 Con Penalización			
	Bayesiano		SVM no lineal		Bayesiano		SVM no lineal	
	TA	R	TA	R	TA (%)	R	TA	R
F	79,10	79,10	81,06	78,50	88,85	93,80	90,50	89,80
I	80,20	78,90	80,02	77,00	80,50	90,30	84,80	84,00
D	78,54	80,50	80,00	77,40	76,18	91,17	80,40	80,00
L	80,62	78,70	81,16	74,90	76,02	88,40	80,20	73,40

Tabla 4.2: Comparación entre los diferentes clasificadores para el método 2 con y sin penalización.

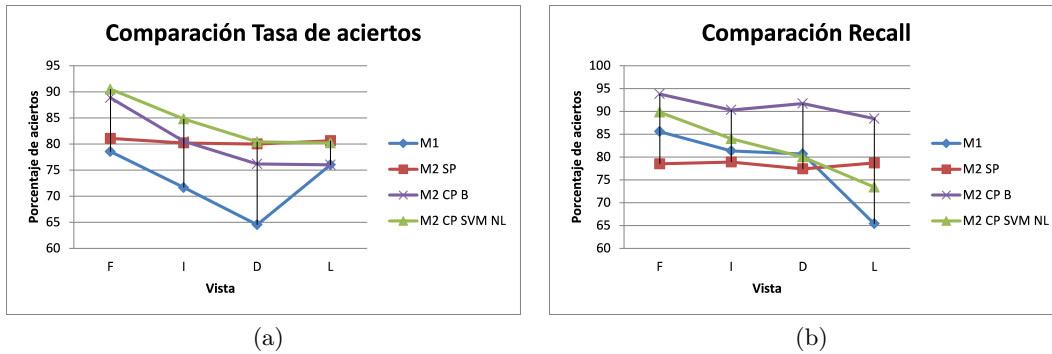


Figura 4.8: Comparación entre los métodos de extracción de simetría.

de estos métodos para hacer un estudio más en profundidad. Como se puede ver, para las vistas frontal y lateral izquierda el uso de la penalización nos ofrece una tasa de aciertos en clasificación mucho más alta mediante el clasificador SVM no lineal, mientras que el clasificador bayesiano sólo aumenta su tasa de aciertos para la vista frontal. Sin embargo, nos tenemos que fijar en el valor del recall, donde el clasificador SVM no lineal aumenta este valor para las vistas frontal y lateral izquierda, mientras que el clasificador bayesiano pese a bajar en algunos casos la tasa de acierto, aumenta para todas las vistas de manera considerable, es decir, consigue una mejor clasificación de la clase vehículo. Por lo tanto, se puede concluir que el uso de la penalización introducida tiene un efecto positivo sobre el método de extracción de simetría 2. Así, el uso del método sin penalización utilizaría para las vistas frontal y derecha el clasificador SVM no lineal y para las vistas izquierda y lejana el clasificador bayesiano. Por otra parte, si utilizamos la penalización, el uso del clasificador bayesiano es interesante debido a los altos valores de recall que obtiene para todas las vistas, mientras que el clasificador SVM no lineal obtiene valores superiores para la tasa de aciertos.

La Figura 4.8, presenta una comparación entre los diferentes métodos utilizados para la verificación de vehículos con el uso de simetría. Mediante esta comparación queda demostrado que el método 2 funciona mucho mejor para la verificación de vehículos utilizando simetría. Si sólo se tiene en cuenta la tasa de aciertos, el método 2 con penalización utilizando el clasificador SVM no lineal es el que obtiene mejores resultados, siendo para las vistas derecha y lejana muy similares a los obtenidos mediante el método 2 sin penalización. Por otra parte, fijándonos solamente en el valor de recall el método 2, con penalización con el clasificador bayesiano, consigue unos resultados mucho mejores. Por lo tanto, el uso de un clasificador u otro dependerá del tipo de aplicación del sistema.

Por último, comparando nuestro sistema basado en simetría con la propuesta en [7], se obtiene un rendimiento superior como se muestra en la Tabla 4.3. La mejora es clara en todas las vistas, siendo especialmente notable en la frontal. En media (haciendo la media de la tasa

Tasa de Acierto				
	F	I	D	L
Método [7]	80,18	80,46	78,54	80,50
Nuestro Método	90,50	84,80	80,40	81,16

Tabla 4.3: Comparación de nuestro sistema con la propuesta [7].

de aciertos en todas las vistas), la mejora que ofrece nuestra propuesta es de 4,3 %.

## 4.4. Desarrollo del sistema basado en bordes

### 4.4.1. Extracción de gradientes de bordes

El uso de bordes es uno de los métodos más utilizados en la verificación de vehículos mediante características explícitas. Los métodos presentados a continuación se basan en la extracción de bordes únicamente verticales, como ya se explicó en el capítulo anterior, que debido a la naturaleza de los vehículos, la extracción de este tipo de bordes es muy útil para la caracterización de éstos.

#### 4.4.1.1. Uso de los 64 valores de bordes

Para este método se van a utilizar todos los valores resultantes de la extracción de bordes verticales sobre la imagen a nivel de gris. En cada uno de los métodos se estudian dos alternativas para la extracción de bordes. La primera es la extracción de los valores de los gradientes verticales mediante el operador de Sobel. La segunda alternativa realiza la misma extracción de gradientes seguida de una binarización, es decir, a los píxeles en los que el gradiente supera un cierto umbral se les asigna el valor '1' y los píxeles que no superan el umbral se les asigna '0'. Para realizar la clasificación de las imágenes, se genera un vector de 64 valores, que son el valor medio de los gradientes de cada columna de la imagen. Así obtenemos una distribución de cómo se encuentran distribuidos los bordes verticales en la imagen, como se muestra en la Figura 4.9, donde se observan las distribuciones de ambas clases, vehículos y no vehículos, obtenidas mediante el uso de los dos métodos expuestos. Es esperable la existencia de picos más pronunciados relativos a los extremos del vehículo, que no tienen porqué darse en imágenes de la clase no-vehículo.

#### 4.4.1.2. Reducción del vector de bordes

Obtenido el vector de bordes de 64 valores, se ha buscado conseguir un vector de bordes reducido, con el objetivo de mejorar el rendimiento de la clasificación. Para ello la primera aproximación que se ha llevado a cabo es la de agrupar los valores del vector de 64 valores en

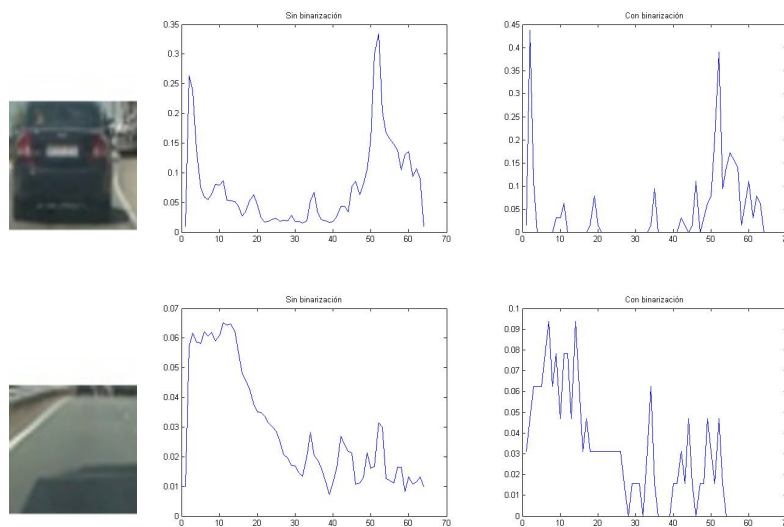


Figura 4.9: Distribuciones de los bordes verticales en las clases vehículo y no vehículo para los métodos de extracción de bordes sin binarización y con binarización.



Figura 4.10: Ejemplo de la agrupación de los valores del vector completo de bordes de la imagen en bins. En este caso se pasa a un vector compuesto por 16 valores o bins.

bins, como se puede ver en la Figura 4.10, buscando la configuración óptima y el vector más reducido posible para obtener una clasificación satisfactoria.

Las pruebas realizadas consisten en la exploración de vectores de tamaño 32, 16, 8 y 4. A pesar de obtener un coste computacional menor en la fase de clasificación, estas aproximaciones añaden un coste computacional en la fase de extracción de características, debido a que se añade el paso de la agrupación en bins. Por lo tanto, también se ha buscado mejorar la eficiencia, mediante el uso únicamente de los datos más representativos del vector de bordes, como son el valor máximo de la parte izquierda y derecha del vector, correspondientes a los bordes verticales del contorno del vehículo. Así, partimos de la hipótesis de que los vehículos son simétricos y que tienen dos bordes muy pronunciados correspondientes a los extremos del vehículo, mientras que los valores máximos de las imágenes que no son vehículos, correspondientes a los bordes más pronunciados en cada una de las partes, no serán tan altos.

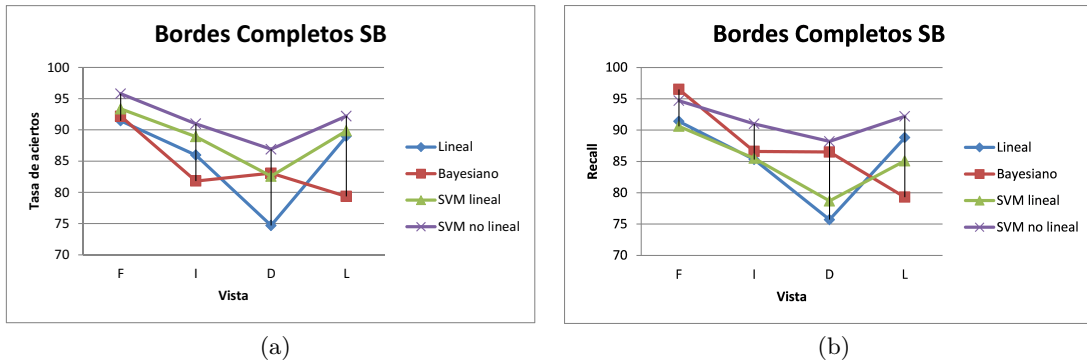


Figura 4.11: Rendimiento de los diferentes clasificadores sobre el vector de 64 componentes. (a) Tasa de aciertos (b) Recall.

#### 4.4.2. Clasificación

En esta subsección se ha realizado el estudio para obtener el método de extracción de bordes que mejor rendimiento consiga de los anteriormente expuestos, así como la elección del clasificador que mejor se adapte a este tipo de características. Primero, el vector compuesto de los 64 valores, sin utilizar la binarización, se estudia para determinar qué clasificador se adapta mejor a los datos haciendo uso de la tasa de aciertos y del valor recall, y luego se realiza el mismo estudio para el vector binarizado. Los resultados del clasificador con mejor rendimiento se comparan entre los dos métodos.

La Figura 4.11 muestra las gráficas del rendimiento del vector completo sin binarizar con los cuatro clasificadores. Como se puede observar, el clasificador SVM no lineal es el que obtiene un mayor rendimiento, ya que ofrece los mayores valores de tasa de acierto y recall, es decir, que además de realizar una mejor clasificación de todas las imágenes de ambas clases, consigue la mejor clasificación de la clase vehículo que en el caso de nuestro sistema es la clase más importante. Bien es cierto que para la vista frontal el clasificador bayesiano obtiene un mayor recall, pero debido a la mejora que proporciona la tasa de acierto del clasificador SVM no lineal respecto al bayesiano y a la pequeña diferencia producida en el valor de recall, el mejor clasificador para todas las vistas es el clasificador SVM no lineal. Cabe destacar que los comportamientos de los clasificadores lineal y SVM no lineal son similares para ambas clases en lo que respecta a tasa de aciertos, mientras que el clasificador bayesiano clasifica mejor la clase vehículo a diferencia del clasificador SVM lineal que obtiene mejores tasas de acierto para la clase no vehículo. Esto indica que el clasificador SVM no lineal obtiene una frontera de decisión que trata de diferenciar lo máximo posible una clase de otra, no como los otros clasificadores que dan prioridad a una de las clases.

En la Figura 4.12 se observa el comportamiento del rendimiento del sistema de verificación que utiliza el vector completo de bordes de la imagen binarizada. Al igual que el caso anterior el



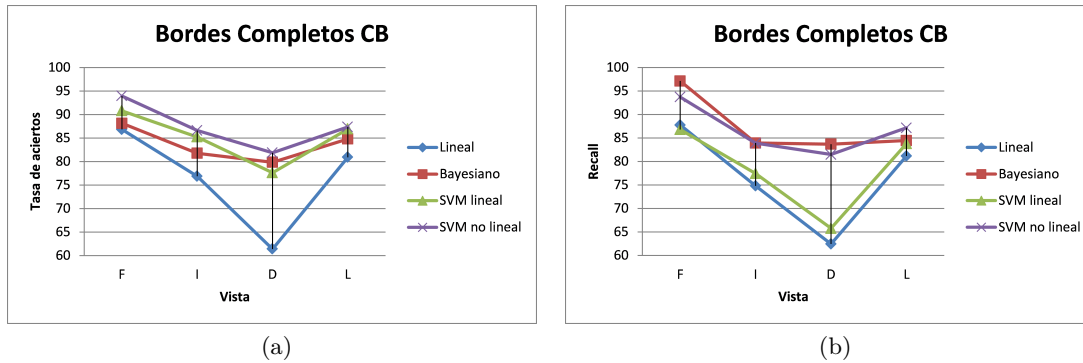


Figura 4.12: Rendimiento de los diferentes clasificadores sobre el vector de 64 componentes extraídas de la imagen binarizada. (a) Tasa de aciertos (b) Recall.

Vista	Sin Binarización		Con Binarización	
	TA	R	TA	R
F	95,78	94,70	93,94	93,72
I	90,98	91,00	86,60	83,88
D	86,90	88,20	79,84	83,68
L	92,20	92,20	87,34	87,16

Tabla 4.4: Comparación entre los métodos de extracción de bordes mediante la imagen sin binarizar y la imagen binarizada.

clasificador que obtiene la mayor tasa de aciertos es el clasificador SVM no lineal, aunque en este caso el rendimiento en cuanto a tasa de aciertos es muy similar con el uso de los clasificadores bayesiano, SVM lineal y no lineal. Fijándonos en el valor de recall, es fácil descartar el clasificador SVM lineal ya que obtiene valores muy por debajo de lo esperado, llegando a los valores del clasificador lineal. En cuanto a la elección del mejor clasificador, podemos asegurar que para las vistas lateral izquierda y lejana el que mejor comportamiento consigue es el clasificador SVM no lineal. En cuanto a las vistas frontal y derecha la elección del clasificador adecuado resulta más difícil. Para la vista frontal se elegirá el clasificador SVM no lineal debido a que la diferencia en la tasa de acierto es considerablemente mejor, mientras que el valor de recall es ligeramente menor. Por el contrario el clasificador elegido para la vista derecha es el clasificador bayesiano ya que nuestro sistema da preferencia al valor de recall cuando la diferencia en la tasa de acierto es mínima.

Comparando ambos métodos para saber si el uso de la binarización es adecuado cuando se trata de la extracción de bordes, obtenemos la Tabla 4.4 que resume los valores obtenidos utilizando el mejor clasificador en cada caso. Se obtiene un mejor rendimiento utilizando la imagen sin binarizar ya que esta aporta más información, quedando demostrado como la utilización del histograma generado a partir de los gradientes verticales de la imagen consigue unos buenos

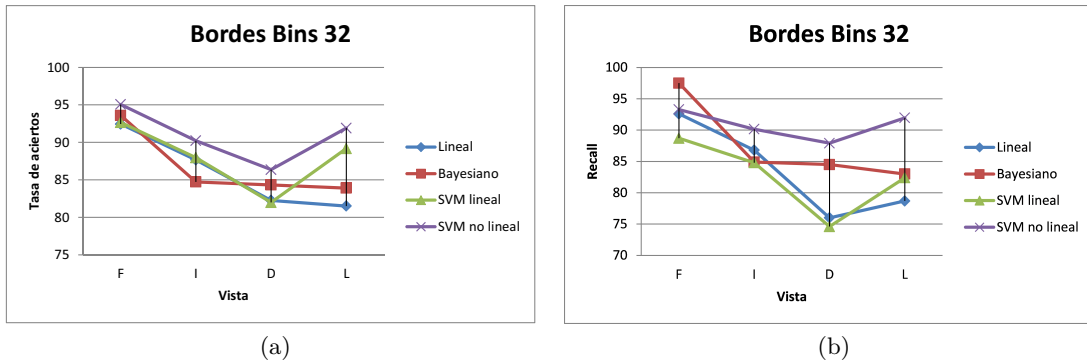


Figura 4.13: Rendimiento de los diferentes clasificadores sobre el vector de 32 bins. (a) Tasa de aciertos (b) Recall.

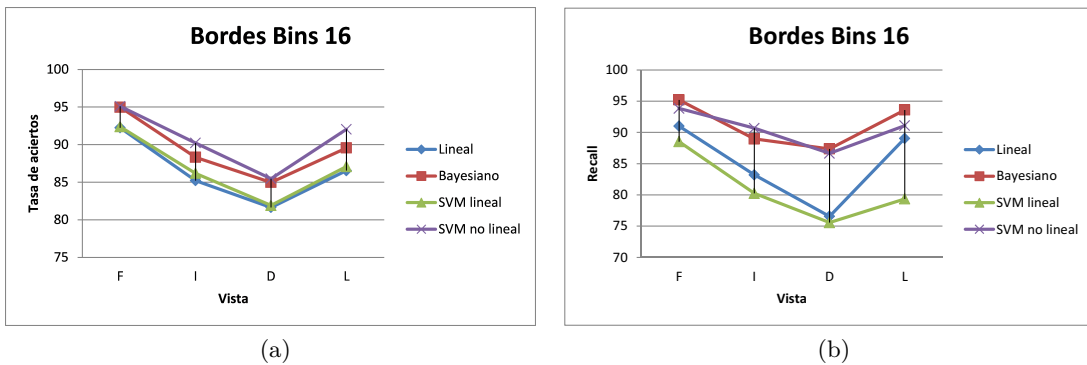


Figura 4.14: Rendimiento de los diferentes clasificadores sobre el vector de 16 bins. (a) Tasa de aciertos (b) Recall.

resultados. Así, la búsqueda de vectores reducidos se realizará sobre la imagen sin binarizar.

El primer paso que se ha dado para la reducción del vector de características, es la reducción del vector a la mitad, es decir, la caracterización de los bordes en un vector de 32 bins. La Figura 4.13 nos muestra los resultados obtenidos utilizando los diferentes clasificadores. Como vemos, con la reducción a 32 bins seguimos obteniendo rendimientos elevados. Esto se debe a que no es necesaria tanta cantidad de información de bordes para caracterizar cada una de las clases. Por otro lado, también cabe destacar que el clasificador SVM no lineal sigue siendo el clasificador que obtiene mejores resultados. Sin embargo, en la vista frontal, el valor de recall es mayor mediante el uso del clasificador bayesiano y debido a que la diferencia en la tasa de aciertos es mínima respecto a la SVM no lineal, el mejor clasificador para esta vista cuando se utilizan 32 bins es el bayesiano.

El siguiente paso es volver a reducir a la mitad, por lo tanto se forma un vector compuesto por 16 bins, dando como resultados en clasificación los mostrados en la Figura 4.14. Se puede

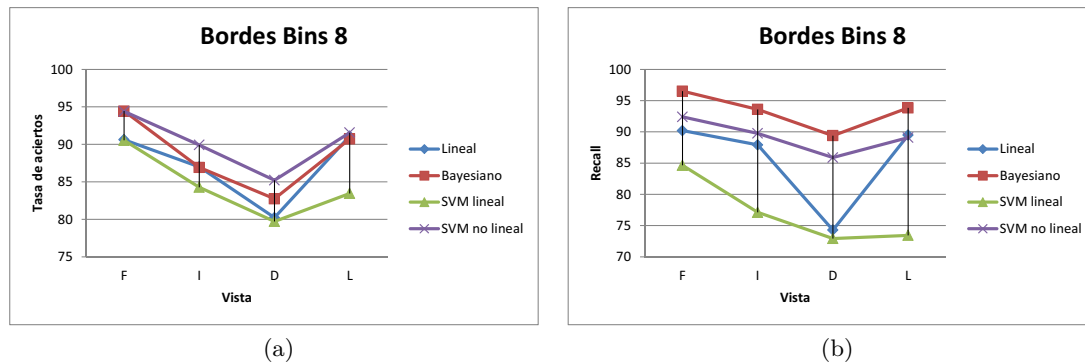


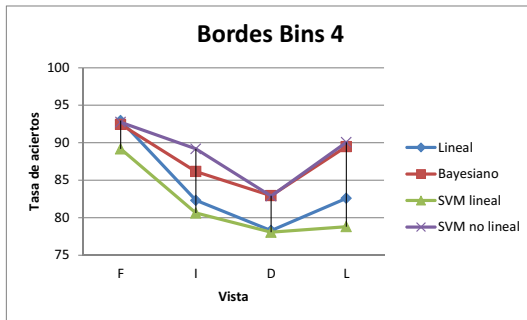
Figura 4.15: Rendimiento de los diferentes clasificadores sobre el vector de 8 bins. (a) Tasa de aciertos (b) Recall.

observar como el rendimiento de los clasificadores bayesiano y SVM no lineal es más similar que en el caso de 32 bins. Este hecho cambia la tendencia del uso del clasificador SVM no lineal hacia el uso del clasificador bayesiano, siendo la vista lateral izquierda y lejana las únicas que obtiene claramente mejores resultados con el clasificador SVM no lineal. Por lo tanto, teniendo en cuenta la prioridad que se le da al valor de recall, el clasificador elegido para las vistas restantes es el clasificador bayesiano.

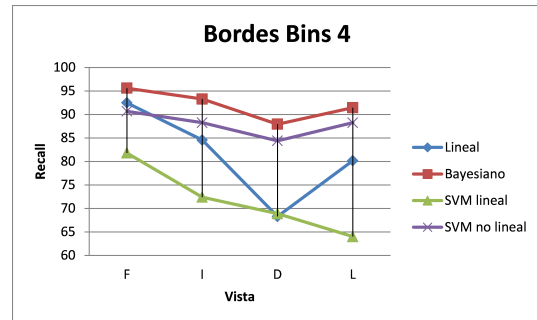
Siguiendo los pasos anteriores, se vuelve a reducir el vector a la mitad para ver el impacto que genera, obteniendo un vector de 8 bins, cuyo resultado en clasificación se puede ver en la Figura 4.15. Se observa como el clasificador SVM ya no consigue los mejores resultados en la tasa de aciertos para todas las vistas, donde para las vistas frontal y lejana el clasificador bayesiano obtiene valores similares. En el caso del valor de recall el cambio es más significativo, ya que para todas las vistas el clasificador bayesiano obtiene mejores resultados. Por lo tanto, se puede asegurar que el clasificador adecuado para las vistas frontal y lejana en el caso del vector compuesto por 8 bins es el bayesiano a diferencia de las vistas laterales donde la selección del clasificador depende de lo que necesite el sistema.

Por último, se realiza la reducción hasta obtener un vector de 4 bins, los resultados obtenidos mediante este vector están expuesto en la Figura 4.16. Vemos como el clasificador SVM no lineal baja su rendimiento hasta valores del clasificador bayesiano en la tasa de aciertos. Por otro lado, al igual que ocurría en 8 bins, el valor de recall obtenido por el clasificador bayesiano es mejor que el obtenido por la SVM no lineal.

En la Figura 4.17 se muestra la evolución que se produce al ir reduciendo el tamaño del vector para los clasificadores SVM no lineal y bayesiano. Se puede observar como el clasificador SVM no lineal reduce los valores de tasa de acierto y recall a medida que se reduce el vector de características utilizado, esto indica que este clasificador diferencia mejor entre clases cuanto mayor es la información utilizada. Sin embargo, el clasificador bayesiano tiene un comporta-



(a)



(b)

Figura 4.16: Rendimiento de los diferentes clasificadores sobre el vector de 4 bins. (a) Tasa de aciertos (b) Recall.

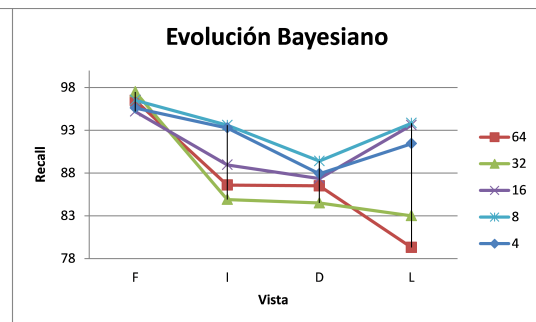
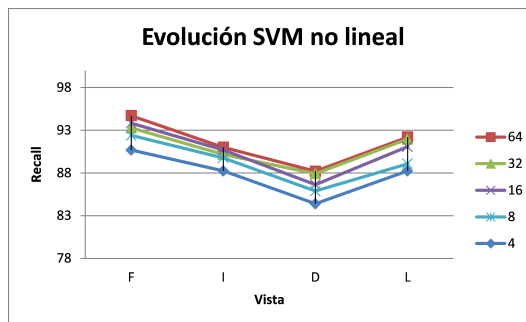
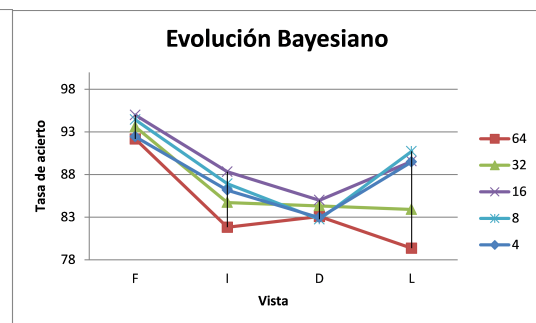
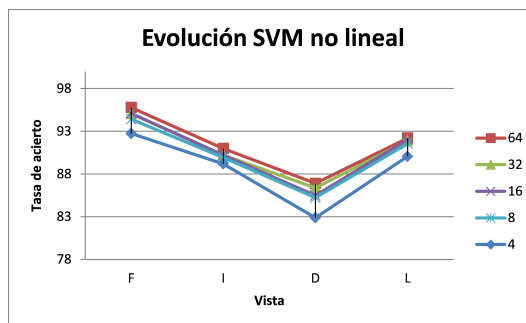


Figura 4.17: Evolución de los clasificadores SVM no lineal y Bayesiano al ir reduciendo la dimensionalidad del vector de características.

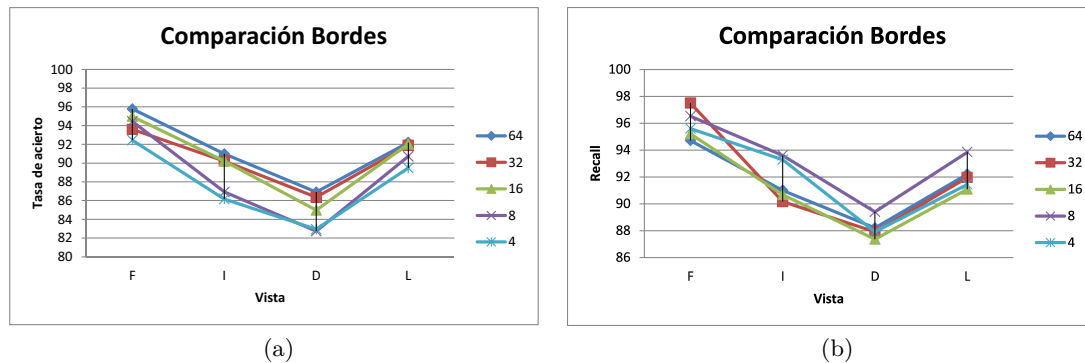


Figura 4.18: Comparación entre el rendimiento de las diferentes configuraciones del vector de bordes. (a) Tasa de aciertos (b) Recall.

miento diferente, como vemos este consigue mejores resultados para la tasa de aciertos cuando reducimos el número de componentes del vector hasta llegar a un límite (en éste caso, 16 bins) donde el rendimiento empieza a decrecer. Para comprender este comportamiento tenemos que fijarnos en el valor de recall, el cuál es mayor cuanto menor es la dimensión del vector de características, lo que indica que el uso de un número menor de información dificulta la distinción entre las dos clases, por lo que el clasificador da prioridad a la clase vehículo, formando una frontera de decisión más flexible con los falsos positivos y así obtiene un número mayor de vehículos clasificados correctamente.

Para ver qué aproximación de bordes se comporta mejor, en la Figura 4.18 se muestra una comparación de los mejores resultados obtenidos en cada una de las aproximaciones, junto al rendimiento obtenido del vector de bordes completo sin binarizar. Aunque el uso del vector completo del histograma formado por los gradientes verticales consigue los mejores resultados para la tasa de acierto, éste consigue peores resultados comparados con las otras aproximaciones para el valor de recall. Por lo tanto, la reducción del vector, además de conseguir una mejora en el coste computacional de la fase de clasificación, también mejora el número de vehículos correctamente clasificados, aunque ofrece tasas de acierto inferiores. Esta situación hace que la elección de la aproximación a utilizar sea complicada. Así, se presentan tres estrategias a seguir para la elección de ésta. Si se quiere obtener la mayor tasa posible de aciertos del sistema, la opción escogida será el vector completo de 64 valores. Por otro lado, si lo que se busca es reducir la dimensión del vector, reduciendo en la menor medida posible la tasa de acierto, el vector elegido será el constituido por 16 valores. Por último, si se quiere obtener el mayor número de vehículos correctamente clasificados la elección es el vector compuesto por 8 valores.

Con el objetivo de conseguir un vector de características lo más reducido posible, consiguiendo así reducir el procesamiento del sistema y obteniendo una mayor eficiencia, se ha estudiado el impacto generado por el valor máximo de la parte izquierda y derecha del histograma. Por lo

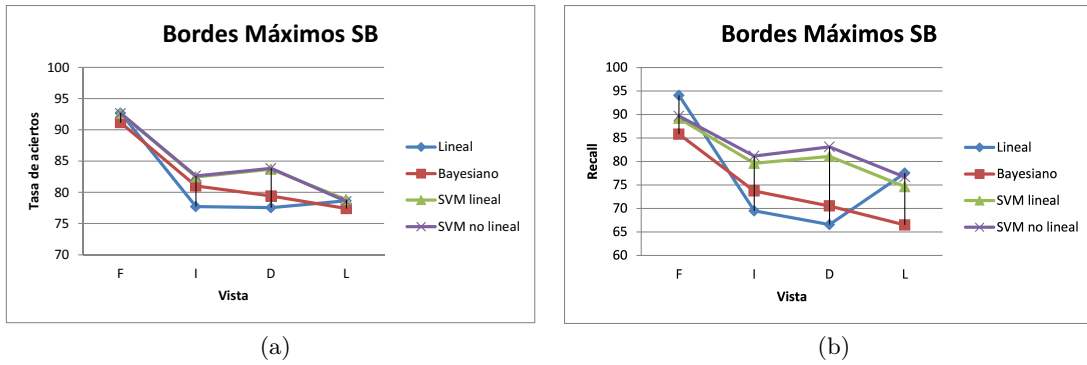


Figura 4.19: Rendimiento de los diferentes clasificadores sobre el vector de máximos sin binarización. (a) Tasa de aciertos (b) Recall.

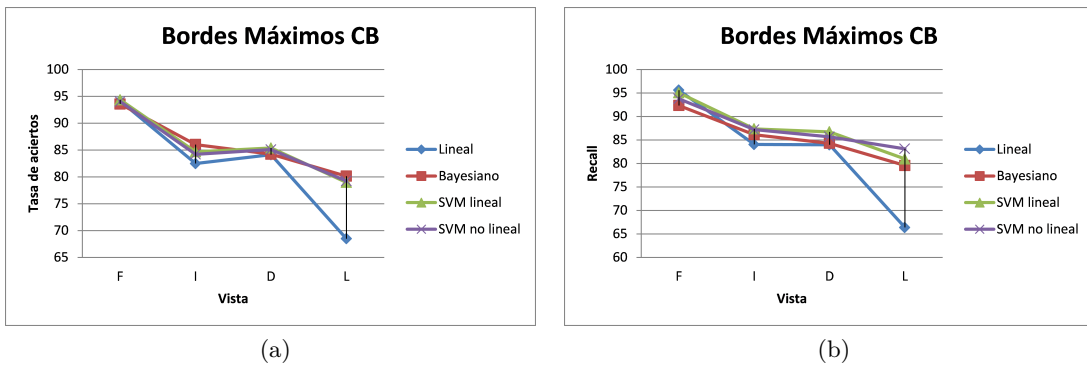


Figura 4.20: Rendimiento de los diferentes clasificadores sobre el vector de máximos con binarización. (a) Tasa de aciertos (b) Recall.

tanto, este vector estará formado por los valores que corresponden a los extremos del vehículo. Dado que sólo se tienen en cuenta dos valores del histograma, esta aproximación se ha realizado tanto sobre la imagen binarizada como sobre la imagen sin binarizar.

En la Figura 4.19 se muestra el comportamiento de la aproximación sin binarizar haciendo uso de los diferentes clasificadores. El uso exclusivamente de los máximos sigue ofreciendo rendimientos altos. Para las vistas laterales los clasificadores que ofrecen el mejor rendimiento son los clasificadores SVM, por otro lado las vistas frontal y lejana obtienen rendimientos similares, tanto usando los clasificadores SVM como el lineal, incluso consiguiendo este último una ligera mejora en el rendimiento.

Para el caso de la extracción de los máximo sobre la imagen binarizada, la Figura 4.20 ofrece una visión del comportamiento de los clasificadores. A diferencia de la aproximación anterior, los clasificadores, en este caso, obtienen un rendimiento muy similar para todas las vistas salvo el clasificador lineal en la vista lejana, el cuál obtiene un rendimiento bastante inferior al resto.

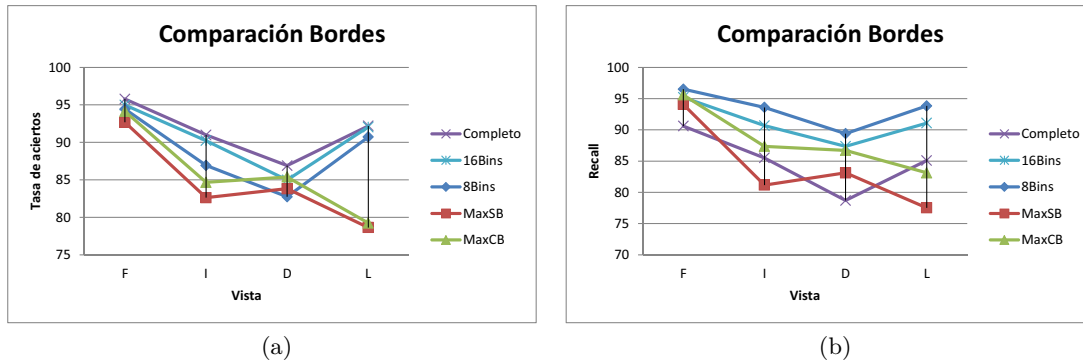


Figura 4.21: Comparación entre el rendimiento de las diferentes configuraciones del vector de bordes. (a) Tasa de aciertos (b) Recall.

Esto se debe a que al realizar la binarización los bordes verticales con menor nitidez no sean correctamente seleccionados. Como la vista lejana contiene imágenes de menor calidad, es más difícil conseguir bordes relevantes o con la intensidad suficiente para la caracterización de los vehículos, y por lo tanto una frontera de decisión lineal no consigue rendimientos aceptables. Por lo tanto para la vista frontal se escogería el clasificador lineal, para las vistas laterales el clasificador SVM lineal y para la vista lejana el clasificador SVM no lineal.

En la Figura 4.21 se muestra una comparación entre las aproximaciones de máximos descritas y las aproximaciones elegidas anteriormente (compuestas por: 64, 16 y 8 valores). La aproximación por máximos, haciendo uso de la imagen binarizada, nos ofrece valores similares en las vistas frontal y derecha a la mejor aproximación conseguida para la extracción de bordes. Así en aplicaciones donde queramos conseguir un sistema rápido, el uso de la aproximación por máximos utilizando la binarización nos ofrece rendimientos muy buenos para las vistas frontal y derecha, siendo aceptables los valores obtenidos para la vista izquierda. Sin embargo para la vista lejana, al estar compuesta de imágenes con poca definición, es necesario usar un mayor número de valores para la correcta clasificación. El mejor balance obtenido para nuestro sistema entre tasa de acierto, recall y dimensionalidad del vector de características siguiendo los resultados obtenido en la Figura 4.21, será para la vista frontal y lateral derecha el uso de la aproximación de máximos con binarización, para la vista lateral izquierda el vector compuesto por 16 bins y para la vista lejana el vector compuesto por 8 bins.

Por último, comparando la media de la tasa de aciertos para todas las vistas, que en nuestro sistema es un 91,47 %, podemos ver como obtenemos rendimientos superiores a los encontrados en propuestas como [41] (88,13 %) donde también se trabaja sobre propuestas basadas en bordes orientados, o propuestas como [90] que consiguen un rendimiento en torno al 90 %.

## 4.5. Desarrollo del sistema basado en componentes principales

El desarrollo del sistema basado en PCA para la extracción de características se ha realizado sobre las imágenes de la base de datos GTI. En esta sección se estudia la efectividad de la aplicación de PCA sobre la imagen completa, es decir, sobre los valores de intensidad que componen la hipótesis a verificar. Así tendremos, en primer lugar, vectores de características de la misma longitud que las imágenes de las que se compone la base de datos, en nuestro caso de imágenes de 64x64 si utilizamos las imágenes en nivel de gris o 64x64x3 si utilizamos las imágenes en el espacio de color RGB. Debido a la gran variabilidad de las características, como al gran número de éstas, se hace necesario aplicar PCA para reducir la dimensionalidad. Por lo tanto, se espera conseguir subespacios reducidos para cada una de las vistas que componen la base de datos y un alto rendimiento en la clasificación entre clases. Este análisis se ha realizado mediante el uso de un clasificador SVM lineal, siguiendo la propuesta de [1], donde se muestra que este es el clasificador que se adapta mejor a este tipo de sistemas.

### 4.5.1. Análisis de Componentes Principales

El principal objetivo de PCA es reducir la dimensionalidad de un conjunto de variables aleatorias manteniendo la mayor cantidad de información posible. El desarrollo de nuestro sistema que utiliza PCA se ha basado en [91]. Para plantear el problema suponemos  $\mathbf{X} = [X_1, \dots, X_p]^t$ , donde  $p$  es el número de variables (número de imágenes) y cada variable  $X_i$  está compuesta por  $n$  observaciones, en nuestro caso la dimensionalidad de la imagen,  $X_i = [x_1, \dots, x_n]$ . A partir de este vector queremos obtener otro conjunto de variables  $\mathbf{Y} = [Y_1, \dots, Y_p]$ , donde  $Y_i$  está compuesto por  $m$  observaciones, cumpliendo la restricción  $m \leq n$ .

Para la aplicación de PCA se hace necesario centrar la matriz de datos originales  $\mathbf{X}$  a partir de la media de sus componentes:

$$X_i^c = X_i - E\{X_i\} \quad (4.20)$$

por lo que  $E\{\mathbf{x}\} = 0$ . La diferencia entre nuestro sistema basado en PCA y otros trabajos de la literatura como [73], es el uso de la matriz de correlaciones en vez de la matriz de covarianza. No obstante estas matrices están relacionadas de la manera:

$$R(i, j) = \frac{C(i, j)}{\sqrt{C(i, i)C(j, j)}} \quad (4.21)$$

Por lo tanto, sea  $\mathbf{R}_x = E\{\mathbf{X}\mathbf{X}^t\}$  la matriz de correlaciones de la matriz  $\mathbf{X}$  centrada, se puede demostrar que posee  $n$  autovalores reales y diferentes que garantizan que  $\mathbf{R}_x$  sea diagonalizable, puesto que es una matriz simétrica y definida positiva, es decir, se puede encontrar una matriz  $\mathbf{T}$  invertible tal que:



	Gris				RGB			
Vista	I90		I95		I90		I95	
	TA	CP	TA	CP	TA	CP	TA	CP
F	93,37	80	92,48	165	95,76	91	95,10	185
I	92,30	80	90,12	163	92,94	93	93,90	189
D	89,02	65	87,08	140	91,80	78	91,00	164
L	90,88	47	90,40	98	91,50	57	92,00	117
MAX CP	4096				12288			

Tabla 4.5: Rendimiento de PCA para la imagen en nivel de gris y en el espacio de color RGB.

$$\mathbf{R}_x = \mathbf{T}\mathbf{A}\mathbf{T}^{-1} \quad (4.22)$$

donde  $\mathbf{T} = [T_1, \dots, T_n]^t$  es una matriz cuadrada formada por los autovectores de  $\mathbf{R}_x$  y  $\mathbf{\Lambda} = \text{diag}[\lambda_1, \dots, \lambda_n]$  una matriz diagonal formada por sus autovalores. A continuación se normalizan los autovectores de la matriz  $\mathbf{T}$ , redistribuyendo los valores  $\lambda_i$  de la matriz  $\mathbf{\Lambda}$  de acuerdo con su magnitud, de forma que  $\lambda_i > \lambda_j$  para todo  $i < j$ , y rotamos de la misma manera las columnas de  $\mathbf{T}$ . A partir de esta matriz podemos obtener las Componentes Principales. Seleccionando las  $m$  primeras columnas de la matriz  $\mathbf{T}$  y multiplicando por la matriz de datos originales centrada  $\mathbf{X}$  obtenemos la matriz  $\mathbf{Y}$ :

$$\mathbf{Y} = \mathbf{T}_{(m)}^t \mathbf{X} = [Y_1, \dots, Y_p]^t, \quad Y_i = [y_1, \dots, y_m] \quad (4.23)$$

Cada  $Y_i$  se denomina Componente Principal y es una combinación lineal de las variables originales  $\mathbf{X}$ .

Este estudio también se ha basado en trabajos como [73], donde definen un marco de comparación basado en la medida de distorsión de la aproximación PCA, que es la suma de los autovalores de los autovectores descartados. En otras palabras, el factor  $\frac{\sum_{i=1}^M \lambda_i}{\sum_{i=1}^D \lambda_i}$  ofrece una idea del porcentaje de varianza después de la aproximación, que es la cantidad de información retenida. Algunas aproximaciones determinan la dimensionalidad del subespacio principal mediante la retención de un cierto porcentaje de información, en vez de tomar un número fijo de componentes. En [59], por ejemplo, realizan los experimentos mediante la retención del 90% y 95% de la varianza. Este estudio, ha seguido el mismo criterio para la realización del sistema.

La Tabla 4.5 muestra los resultados cuando se mantiene un 90% y 95% de varianza (I90 e I95 respectivamente en la tabla), así como el número de componentes utilizadas para conseguir estos resultados. Como se puede observar, el rendimiento de PCA es superior al conseguido por los métodos de extracción de características basado en características explícitas. A su vez, éste es mayor empleando el espacio de color RGB, ya que éste ofrece una mayor cantidad de información,

Gris												
Vista	Número de Componentes											
	10	20	30	40	50	60	70	80	90	100	110	120
F	93,7	<b>96,1</b>	94,3	95,0	95,1	93,7	92,1	92,3	92,8	90,8	90,2	91,6
I	88,0	90,3	91,9	<b>93,0</b>	92,2	92,7	92,1	91,0	91,3	92,1	90,5	88,9
D	86,6	89,1	89,0	88,7	89,6	90,0	89,9	<b>90,1</b>	88,9	88,9	88,1	86,2
L	88,1	87,9	90,9	<b>91,3</b>	90,5	90,6	90,6	91,0	90,2	89,4	88,9	89,8

RGB												
Vista	Número de Componentes											
	10	20	30	40	50	60	70	80	90	100	110	120
F	94,2	<b>97,1</b>	95,8	95,6	97,0	94,8	95,6	95,1	95,8	95,5	95,1	96,5
I	88,7	90,7	92,9	<b>93,8</b>	93,4	93,7	93,2	92,8	93,2	93,6	93,3	91,9
D	87,5	89,9	91,5	91,8	92,2	91,7	90,4	<b>93,1</b>	89,7	91,5	90,3	89,9
L	89,4	92,9	90,6	90,7	91,6	90,9	90,6	92,4	<b>93,5</b>	92,0	88,0	91,0

Tabla 4.6: Rendimiento para diferente dimensionalidad del subespacio principal.

que permite una mejor distinción entre clases. El número de componentes principales utilizadas por PCA en ambos espacios son similares, sin embargo hay que destacar que el espacio de color RGB tiene el triple de componentes, lo que nos indica que la información está más agrupada en las primeras componentes del subespacio principal. Este hecho pone de manifiesto que, aunque este espacio contenga más información, no toda ella es relevante, siendo solamente una pequeña parte de las componentes la que aportan información útil para la clasificación a diferencia de la imagen en gris, donde la información está más distribuida entre las componentes. Haciendo una comparación entre el porcentaje de varianza retenido, observamos cómo para alguna de las vistas como la frontal o la derecha, no es necesario retener un número elevado de información, ya que el exceso de información puede generar situaciones de clasificación errónea. A diferencia de otras vistas como la izquierda donde se necesita retener más información para realizar una clasificación satisfactoria.

Otras aproximaciones fijan a priori el número de componentes principales empleadas. En la Tabla 4.6 se ha estudiado el comportamiento de PCA para cada una de las vistas, fijando el número de componentes principales utilizadas, y a partir de los resultados se determina el número de componentes adecuado para obtener una buena clasificación en cada una de las vistas. Cabe destacar que los mejores resultados se consiguen tanto para las imágenes en nivel de gris como en el espacio RGB, para el mismo número de componentes, salvo en el caso de la vista lejana. También es interesante como el número de componentes crece en función de la vista, lo que nos indica que aquellas vistas que contienen imágenes con mayor calidad necesitan menos información para su correcta clasificación. Fijando el número de componentes a utilizar nos aseguramos conseguir el máximo rendimiento para cada vista. Por otro lado, si queremos obtener

Tasa de Acierto				
	F	I	D	L
Método [73]	96,22	93,32	91,04	91,54
Nuestro Método	97,10	93,80	93,10	93,50

Tabla 4.7: Comparación de nuestro sistema basado en PCA con otros encontrados en la literatura.

un sistema genérico basado en PCA, la mejor opción es utilizar un número de componentes que dependa del porcentaje de información que queremos retener.

Por último, si comparamos estos resultados con propuestas como [73], podemos observar como se ha conseguido un aumento en el rendimiento del sistema, como se muestra en la Tabla 4.7.

## 4.6. Desarrollo del sistema basado en Histogramas de Gradientes Orientados

El descriptor HOG involucra muchos parámetros, por lo que se va a estudiar la influencia de estos sobre la clasificación. En primer lugar se va a realizar el descriptor Standard HOG (SHOG) (Subsección 4.6.1). A partir de este descriptor se estudia la influencia de los parámetros sobre el proceso de clasificación, proponiendo el uso de diferentes parámetros, que consideramos aquellos que maximizan el rendimiento, por lo que llamaremos a este nuevo descriptor, HOG óptimo (OHOG) (Subsección 4.6.2). Por último, con el objetivo de conseguir un sistema eficiente que pueda permitir su incorporación en sistemas de tiempo real, se han estudiado tres estrategias diferentes (Subsección 4.6.3 y Subsección 4.6.4).

### 4.6.1. SHOG

Se entiende como SHOG, el descriptor que cumple las características del descriptor propuesto por Dalal en [74]. Para la fase de extracción de gradientes, Dalal propone el operador  $[-1, 0, 1]$  y opta por utilizar un rango para los ángulos reducido,  $[0, \pi]$ , esto se debe a que se ignora el signo del gradiente porque considera que no obtiene ninguna mejora en clasificación. En la fase de cálculo de histogramas, se han de definir los parámetros de número de celdas ( $\eta \times \eta$ ) y número de bins de orientación empleados ( $\beta$ ). Para los estudios realizados en este trabajo se van a considerar los valores,  $\eta = 2, 4, 8, 16$  y  $\beta = 8, 12, 16, 32$ . También hay que definir en esta etapa el método para el cálculo de los histogramas. Para llevar a cabo este cálculo partimos de los valores del módulo y ángulo de cada píxel, para cada una de las celdas dividimos el rango de los ángulos en  $\beta$  intervalos. Para cada píxel se parte del valor del ángulo y se busca la contribución que tiene sobre sus intervalos más cercanos, añadiendo a cada intervalo el valor proporcional

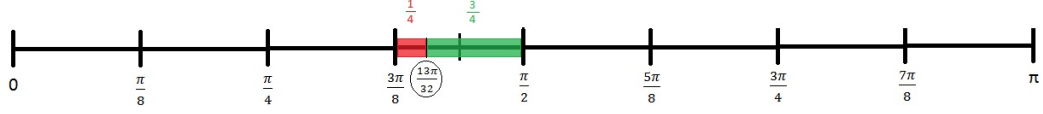


Figura 4.22: Ejemplo de la interpolación de los gradientes en el descriptor HOG.

F	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$
$\eta = 2$	98,60	96,28	98,20	92,62
$\eta = 4$	<b>99,48</b>	98,52	99,18	98,28
$\eta = 8$	98,68	98,54	99,16	98,40
$\eta = 16$	98,82	98,98	98,48	99,02

I	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$
$\eta = 2$	93,04	91,52	94,30	88,76
$\eta = 4$	<b>97,64</b>	96,96	97,04	95,70
$\eta = 8$	97,34	96,56	96,96	96,04
$\eta = 16$	96,10	96,02	95,98	96,02

D	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$
$\eta = 2$	92,78	91,02	93,66	88,32
$\eta = 4$	96,00	95,46	<b>96,22</b>	95,10
$\eta = 8$	94,70	95,12	94,54	94,46
$\eta = 16$	93,64	94,42	93,56	95,10

L	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$
$\eta = 2$	91,96	87,52	91,52	84,32
$\eta = 4$	<b>97,76</b>	96,54	97,04	95,56
$\eta = 8$	97,28	96,70	96,44	96,18
$\eta = 16$	97,38	97,16	97,40	97,34

Tabla 4.8: Tasa de acierto del descriptor SHOG.

del ángulo de ese píxel. Un ejemplo de este proceso se muestra en la Figura 4.22. Para  $\beta = 8$ : si el valor del ángulo del píxel actual es  $ang = \frac{13\pi}{32}$ , tenemos que los intervalos adyacentes son:  $a_1 = \frac{3\pi}{8}$ ,  $a_2 = \frac{\pi}{2}$  y el tamaño del intervalo  $t = \frac{\pi}{\beta} = \frac{\pi}{8}$ . Si  $r_1 = ang - a_1 = \frac{\pi}{32}$  y  $\alpha_1 = \frac{t-r_1}{t} = \frac{3}{4}$ , obtenemos que la contribución del módulo al primer intervalo será:  $Mod * \alpha_1$ . Para el segundo intervalo tendríamos  $r_2 = ang - a_2 = \frac{3\pi}{32}$  y  $\alpha_2 = \frac{t-r_2}{t} = \frac{1}{4}$  y por lo tanto la contribución a este intervalo será:  $Mod * \alpha_2$ .

Al final de esta etapa nos quedaría un vector compuesto de  $\eta \times \eta \times \beta$  componentes, el cuál representa el histograma de la imagen completa. Por último como se indica en la Figura 3.3, se realiza una etapa de normalización. Para ello se van a utilizar bloques solapados de  $2 \times 2$  celdas y el método utilizado para la normalización será L2-norm. Para la clasificación se emplea un clasificador SVM con núcleo lineal.

La Tabla 4.8 muestra los resultados obtenidos para cada una de las vistas utilizando esta configuración. Los mejores resultados se obtienen para  $\eta = 4$ , es decir, 16 celdas en todas las vistas, lo que nos sirve como configuración de partida para las propuestas de mejoras en las secciones posteriores. Es destacable también como los mejores resultados se obtienen para  $\beta = 8$  en tres de las cuatro vistas. Como se esperaba, la vista frontal es la que obtiene un mayor rendimiento, debido a que los vehículos presentan unos patrones geométricos más estables que se adaptan perfectamente a la topología del descriptor HOG. Los peores valores obtenidos en la vista derecha siguen siendo suficientes para obtener una clasificación satisfactoria. En las siguientes subsecciones se estudia por un lado, mejorar el rendimiento de este descriptor buscando

$\eta = 4$				$\eta = 8$				
$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$	
[-1 0 1]								
F	<b>99,48</b>	98,52	99,18	98,28	98,68	98,54	99,16	98,40
I	<b>97,64</b>	96,96	97,04	95,70	97,34	96,56	96,96	96,04
D	96,00	95,46	<b>96,22</b>	95,10	94,70	95,12	94,54	94,46
L	<b>97,76</b>	96,54	97,04	95,56	97,28	96,70	96,44	96,18
Sobel								
F	99,34	98,90	<b>99,44</b>	98,70	99,14	98,64	98,50	98,20
I	<b>97,18</b>	96,22	96,42	95,88	96,88	96,36	96,02	96,14
D	<b>96,18</b>	95,50	95,60	94,94	94,78	94,92	93,94	94,98
L	<b>97,62</b>	96,28	96,90	95,98	96,80	96,84	96,28	96,36

Tabla 4.9: Porcentaje de aciertos en la verificación usando diferentes operadores.

los parámetro que se adapten mejor para el caso particular de los vehículos, y por otro lado se han buscado configuraciones reducidas que faciliten la incorporación de estos descriptores en sistemas de tiempo real.

#### 4.6.2. OHOG

En esta subsección se expone la importancia de una buena elección de los parámetros en el descriptor cuando se clasifican objetos utilizando HOG. Las diferentes configuraciones del descriptor HOG se han evaluado siguiendo los pasos de la Figura 3.3. Los resultados obtenidos en este apartado se han realizado sobre  $\eta = 4$  y  $\eta = 8$  debido a que, como podemos ver en los resultados del descriptor SHOG  $\eta = 2$  obtiene rendimientos bajos y  $\eta = 16$  aunque obtiene rendimientos altos, el coste computacional se hace muy pesado, por lo tanto en este apartado buscamos un balance entre rendimiento y coste computacional. No obstante la configuración óptima elegida se representará con todos los valores.

##### 4.6.2.1. Extracción de gradientes

En primer lugar se han realizado pruebas para ver el impacto del operador de extracción de gradientes. Dalal propone el operador [-1,0,1], mientras que en otros trabajos centrados en la verificación de vehículos [35] se propone el uso del operador de Sobel. La Tabla 4.9 nos ofrece una perspectiva clara del rendimiento del descriptor con el uso de cada uno de estos operadores. El operador de Sobel sólo produce mejoras marginales cuando se usa un mayor número de bins de orientación en la vista frontal. Para el resto de casos este operador no sólo no proporciona ninguna mejora, sino que causa un ligero decremento del rendimiento en todos los casos, siendo más considerable para aquellos donde se utiliza un menor número de bins. Por lo tanto el operador [-1,0,1] demuestra una mayor estabilidad para todas las configuraciones, siendo

$\eta = 4$				$\eta = 8$				
$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$	
$[0, \pi]$								
F	<b>99,48</b>	98,52	99,18	98,28	98,68	98,54	99,16	98,40
I	<b>97,64</b>	96,96	97,04	95,70	97,34	96,56	96,96	96,04
D	96,00	95,46	<b>96,22</b>	95,10	94,70	95,12	94,54	94,46
L	<b>97,76</b>	96,54	97,04	95,56	97,28	96,70	96,44	96,18
$[-\pi, \pi]$								
F	99,12	99,34	<b>99,70</b>	99,52	99,26	99,56	99,54	99,70
I	98,40	<b>99,00</b>	98,92	98,80	98,58	98,82	98,90	98,96
D	98,14	98,04	98,54	98,48	98,22	98,34	98,40	<b>98,62</b>
L	97,56	98,46	98,46	98,38	97,84	98,54	98,44	<b>98,56</b>

Tabla 4.10: Porcentaje de aciertos en la verificación usando diferentes rangos para los ángulos.

notable el mayor rendimiento para los valores más bajos de los parámetros, lo cual proporciona vectores más cortos que se traducen en un descriptor más eficiente.

El segundo cambio propuesto para esta etapa es ver el impacto que causa obviar el signo del gradiente en el descriptor HOG sobre la verificación de vehículos, como se propone en trabajos como [35], que siguen las especificaciones de [74] motivado por la idea de obtener un descriptor de tamaño menor. La Tabla 4.10 muestra los resultados de la clasificación para el rango  $[0, \pi]$  y los resultados para el rango expandido  $[-\pi, \pi]$  donde se tiene en cuenta el signo del gradiente como menciona Dalal en [78]. Los resultados obtenidos demuestran cómo el uso del signo del gradiente es importante, consiguiendo una gran mejora en el rendimiento para todas las configuraciones, siendo más significativa para las vistas más críticas como son las vistas laterales y lejana. También hay que destacar que usando este aumento de rango la tendencia cambia, obteniendo mejores rendimientos para los valores más altos de los bins de orientación ( $\beta$ ). A diferencia de lo que se podía pensar en un primer lugar, con la ampliación del rango no aumenta la longitud del vector de características, sino el número de ángulos que se tienen en cuenta para cada bin, como se puede observar en la Figura 4.23.

Para los mismo valores de  $\beta$ , el uso de un rango mayor, es decir, si se tiene en cuenta el ángulo de los gradientes, se obtiene un mayor rendimiento, lo que indica claramente que para algunos objetos, en nuestro caso vehículos, la información que aporta el signo del gradiente es más importante que la resolución en la orientación. Por lo que podemos concluir que para aplicaciones donde se busca un balance entre rendimiento y longitud del vector de características, como es nuestro caso, la mejor elección sería tener en cuenta el signo del gradiente y usar un valor de  $\beta$  adecuado para la aplicación.

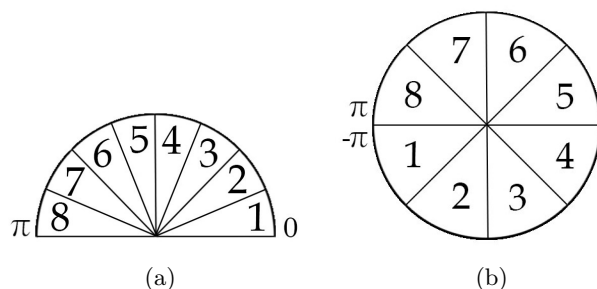


Figura 4.23: Configuraciones posibles del descriptor HOG cambiando el rango de los ángulos. Esta imagen es un ejemplo de la configuración para  $\beta = 8$ . El rango de los ángulos está dividido en  $\beta$  secciones distribuidas uniformemente. (a) Configuración del rango  $[0, \pi]$ . (b) Configuración del rango  $[-\pi, \pi]$ .



Figura 4.24: Ejemplo de la eliminación de la interpolación de los gradientes en el descriptor HOG.

#### 4.6.2.2. Cálculo de Histogramas

El cambio que se propone para esta etapa está enfocado a la reducción del coste computacional en la fase del cálculo de los histogramas. Para ello se propone la eliminación del proceso de interpolación, por lo que el cálculo de los histogramas quedaría reducido, debido a que para cada píxel se buscaría en qué bin de orientación se encuentra su ángulo y se le aplicaría a este bin la contribución completa del píxel. Un ejemplo de esto lo podemos ver en la siguiente Figura 4.24.

El rendimiento producido por este cambio se presenta en la Tabla 4.11. La eliminación de la interpolación nos sigue proporcionando resultados estables, los cuales mejoran cuanto mayor es  $\beta$ . A pesar de los buenos resultados, el rendimiento sufre un pequeño decremento respecto al uso de la interpolación en todos los casos. El uso de este cambio se debe hacer dependiendo de la aplicación que vaya a tener nuestro sistema. Si queremos un sistema que ofrezca el mayor rendimiento posible tendremos que usar la interpolación. Si lo que buscamos es la reducción del coste computacional manteniendo un alto rendimiento para conseguir un sistema más rápido deberemos eliminar la interpolación. En nuestro caso, como lo que buscamos es obtener los parámetros óptimos para el descriptor HOG, no vamos a tener en cuenta este cambio.

$\eta = 4$				$\eta = 8$				
$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$	
Con Interpolación								
F	99,12	99,34	<b>99,70</b>	99,52	99,26	99,56	99,54	99,70
I	98,40	<b>99,00</b>	98,92	98,80	98,58	98,82	98,90	98,96
D	98,14	98,04	98,54	98,48	98,22	98,34	98,40	<b>98,62</b>
L	97,56	98,46	98,46	98,38	97,84	98,54	98,44	<b>98,56</b>
Sin Interpolación								
F	99,06	99,26	99,38	99,30	99,26	99,42	<b>99,64</b>	99,56
I	98,36	98,64	98,60	<b>99,04</b>	98,82	98,98	98,78	98,96
D	97,62	98,10	98,30	98,48	97,96	98,06	98,28	<b>98,58</b>
L	97,56	98,56	98,42	<b>98,72</b>	97,64	98,08	98,52	98,60

Tabla 4.11: Porcentaje de aciertos en la verificación usando diferentes métodos para el cálculo de histogramas.

#### 4.6.2.3. Normalización

En esta etapa se han explorado dos cambios con respecto al descriptor SHOG. En [74] o [35], se sugiere el uso de bloques solapados de tamaño 2x2 celdas para conseguir los mejores resultados, adaptándose así a los cambios de iluminación que se producen en las imágenes. En este trabajo se propone un cambio en el tamaño del bloque. Para ver el impacto que tiene el uso de diferentes tamaños de bloques, se han probado bloques formados por 2x2, 2x1 y 1x1 celdas.

En la Tabla 4.12 se muestran los resultados obtenidos. Los resultados muestran cómo el tamaño del bloque no afecta de manera significativa al resultado de la clasificación. El uso de tamaño de bloque 1x1 muestra resultados más estables y proporciona un vector resultante más corto, lo que se traduce en un coste computacional menor y mayor rapidez del sistema. Esto se debe a que con el uso de bloques de tamaño 1x1 no es necesaria la realización de un segundo histograma para la normalización del vector resultante, ya que la normalización se produce para cada una de las celdas, teniéndose en cuenta así solamente los cambios de iluminación para los píxeles que las constituyen. Esto se traduce en una gran mejora con respecto al uso de bloques de tamaño mayor.

El segundo parámetro que vamos a estudiar en esta etapa es el método de normalización. Dalal en [74] propone diferentes métodos de normalización, si bien trabajos como [35] utilizan directamente la norma L2 para vehículos. Sin embargo, Dalal en [78] propone como el mejor método para el descriptor HOG sobre vehículos, la raíz cuadrada de la norma L1.

La Tabla 4.13 muestra el rendimiento de estos métodos sobre la clasificación. Los experimentos confirman que el uso de la norma L2 proporciona un mejor rendimiento para todas las situaciones frente a la raíz cuadrada de la norma L1, obteniendo pérdidas en algunos casos entorno al 1%. Además la norma L2 muestra resultados más estables, como se puede observar,



	$\eta = 4$				$\eta = 8$			
	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$
<b>2x2</b>								
F	99,12	99,34	<b>99,70</b>	99,52	99,26	99,56	99,54	99,70
I	98,40	<b>99,00</b>	98,92	98,80	98,58	98,82	98,90	98,96
D	98,14	98,04	98,54	98,48	98,22	98,34	98,40	<b>98,62</b>
L	97,56	98,46	98,46	98,38	97,84	98,54	98,44	<b>98,56</b>
<b>2x1</b>								
F	99,36	99,34	99,56	<b>99,72</b>	99,38	99,54	99,60	99,62
I	98,62	<b>99,04</b>	98,98	98,96	98,82	98,88	98,74	98,90
D	97,96	98,20	<b>98,56</b>	98,52	98,14	98,40	98,22	98,48
L	98,04	98,44	<b>98,78</b>	98,68	97,76	98,22	98,70	98,46
<b>1x1</b>								
F	99,16	99,60	99,66	<b>99,76</b>	99,42	99,50	99,46	99,72
I	98,76	99,06	<b>99,20</b>	99,10	98,32	98,92	98,72	98,98
D	98,32	<b>98,56</b>	98,30	98,56	98,04	98,36	98,16	98,50
L	97,90	98,12	98,18	97,88	97,72	97,82	98,36	<b>98,64</b>

Tabla 4.12: Porcentaje de aciertos en la verificación usando diferentes tamaños de bloque.

	$\eta = 4$				$\eta = 8$			
	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$
<b>L2-norm</b>								
F	99,16	99,60	99,66	<b>99,76</b>	99,42	99,50	99,46	99,72
I	98,76	99,06	<b>99,20</b>	99,10	98,32	98,92	98,72	98,98
D	98,32	<b>98,56</b>	98,30	98,56	98,04	98,36	98,16	98,50
L	97,90	98,12	98,18	97,88	97,72	97,82	98,36	<b>98,64</b>
<b>L1-sqrt</b>								
F	99,40	99,38	99,62	<b>99,70</b>	99,24	99,56	99,56	99,50
I	98,70	<b>98,82</b>	98,74	98,64	98,28	98,82	98,82	98,70
D	97,68	97,84	98,14	<b>98,24</b>	97,90	96,00	97,90	98,24
L	97,00	97,96	98,12	97,66	97,58	98,30	98,32	<b>98,42</b>

Tabla 4.13: Porcentaje de aciertos en la verificación usando diferentes métodos de normalización.

$\eta = 4$				$\eta = 8$				
$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$	
SVM lineal								
F	99,16	99,60	99,66	<b>99,76</b>	99,42	99,50	99,46	99,72
I	98,76	99,06	<b>99,20</b>	99,10	98,32	98,92	98,72	98,98
D	98,32	<b>98,56</b>	98,30	98,56	98,04	98,36	98,16	98,50
L	97,90	98,12	98,18	97,88	97,72	97,82	98,36	<b>98,64</b>
SVM no lineal (polinomial de orden 2)								
F	99,56	99,78	<b>98,80</b>	99,80	99,56	99,70	99,54	99,46
I	99,14	99,54	99,54	99,40	99,18	<b>99,62</b>	99,52	99,50
D	98,80	98,70	98,90	98,98	98,86	98,98	99,02	<b>99,20</b>
L	98,40	<b>98,98</b>	98,64	98,88	98,08	98,16	98,76	98,70

Tabla 4.14: Porcentaje de aciertos en la verificación usando diferentes clasificadores.

por ejemplo, para los valores  $\eta = 8$  y  $\beta = 12$ .

## Clasificación

En la mayoría de los trabajos presentes en la literatura donde se utiliza el descriptor HOG sobre vehículos, el clasificador utilizado es un clasificador SVM con un núcleo lineal. Sin embargo, para este trabajo, se parte de la idea de que para conseguir los mejores resultados es necesario, además de explorar los parámetros óptimos del descriptor HOG, hacer un estudio del impacto que genera el uso de diferentes núcleos en el clasificador y cuál es el que se adapta mejor a los vectores resultantes del descriptor OHOG. Para ello, se evalúa además del núcleo lineal, el núcleo no lineal de carácter polinomial de orden 2. La Tabla 4.14 muestra los resultados de clasificación obtenidos. Como se puede observar el núcleo polinomial de orden 2 mejora los resultados en todos los casos si se comparan con los obtenidos mediante el núcleo lineal. Se aprecia una notable mejora para las vistas más críticas como son la vista derecha, izquierda y lejana. Esto se traduce en una mayor flexibilidad del núcleo polinomial de grado 2, el cual se adapta mejor a todas las vistas, consiguiendo una mejor discriminación entre clases. Por lo tanto se consigue un rendimiento más estable y balanceado en la tasa de verificación para todas las categorías.

Los cambios realizados y la adaptación del descriptor, al caso particular de la verificación de vehículos, nos proporciona un aumento considerable en el rendimiento de este descriptor. En la Figura 4.25 se muestra una comparación entre el rendimiento obtenido para SHOG y OHOG. Los cambios que se han realizado con respecto al descriptor SHOG son el aumento del rango de los ángulos ( $[-\pi, \pi]$ ), el uso de bloques más pequeños (de tamaño  $1 \times 1$ ) y el cambio del núcleo en el clasificador SVM (núcleo polinomial de orden 2). Se demuestra claramente como el descriptor OHOG tiene un rendimiento muy superior al SHOG. También cabe destacar cómo para todos

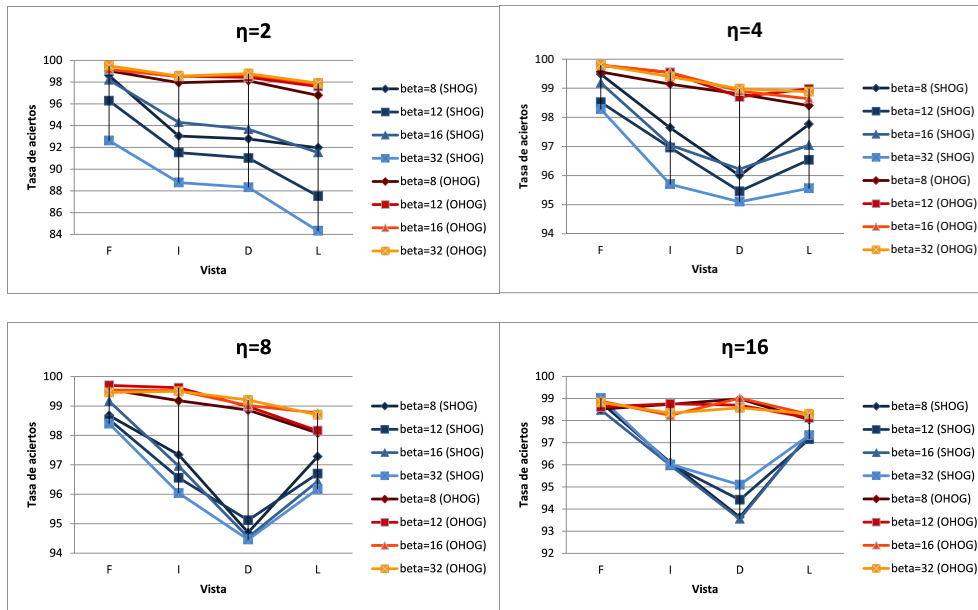


Figura 4.25: Comparación entre los descriptores SHOG y OHOG.

los casos el descriptor OHOG tiene un comportamiento más estable, como muestran sus curvas. No obstante, este estudio también nos ha proporcionado información relevante que nos será de gran utilidad en la búsqueda de un descriptor eficiente, manteniendo en la medida de lo posible el rendimiento.

#### 4.6.3. PCA sobre OHOG

Trabajos como [92] o [93], introducen el análisis de componentes principales (PCA) sobre el descriptor HOG para la clasificación de personas. Basándonos en estas ideas vamos a llevar a cabo un estudio del uso de PCA sobre nuestro descriptor OHOG, adaptado al caso particular de la verificación de vehículos. Lo que se quiere conseguir mediante el uso de PCA es la reducción del vector de características, consiguiendo que éste sea más compacto y así reducir la intensidad computacional en la clasificación.

PCA aplica una transformación lineal sobre el espacio de características, en nuestro caso sobre el vector resultante de aplicar el descriptor OHOG, obteniendo un nuevo espacio de características, en el cuál la información es agrupada en las primeras componentes. Al igual que en la sección 4.5 la suma de los autovalores de los autovectores descartados nos proporciona una medida de la distorsión de la aproximación PCA, es decir, el factor  $\frac{\sum_{i=1}^M \lambda_i}{\sum_{i=1}^D \lambda_i}$ , donde  $M$  es la dimensión del nuevo espacio de características y  $D$  la dimensión del espacio de características original. Este factor nos proporciona una idea del porcentaje de varianza que se mantiene después de la aproximación, que representa la cantidad de información retenida. En la Tabla 4.15

$\beta$	$\eta = 4$				$\eta = 8$			
	8	12	16	32	8	12	16	32
F	99.42	99.72	99.70	<b>99.82</b>	99.68	99.62	99.64	99.56
I	99.08	99.40	99.40	99.40	99.36	99.54	<b>99.56</b>	99.50
D	98.36	98.68	98.66	99.14	98.78	99.06	99.08	<b>99.32</b>
L	98.36	98.78	98.74	<b>98.84</b>	98.10	98.32	98.72	98.72

Dimensionalidad del subespacio principal								
F	47	72	94	<b>177</b>	184	262	324	496
I	51	78	101	185	191	270	<b>331</b>	491
D	52	79	103	184	193	272	333	<b>490</b>
L	55	83	105	<b>175</b>	199	278	335	472
MAX	128	192	256	512	512	768	1024	2048

Tabla 4.15: Tasa de aciertos y número de componentes para PCA-OHOG con un 90 % de varianza retenida.

Vista	I95		I90		I75		I50		I25		I10	
	TA	NC	TA	NC	TA	NC	TA	NC	TA	NC	TA	NC
F	99,78	102	99,72	72	99,60	31	99,22	9	98,16	2	97,18	1
I	99,46	108	99,40	78	99,32	36	98,46	11	97,86	3	97,00	1
D	98,84	109	98,68	79	98,54	37	98,46	11	97,42	3	94,56	1
L	99,00	112	98,78	83	98,34	41	97,72	13	96,60	4	94,26	1

Tabla 4.16: Rendimiento del sistema PCA-OHOG cambiando el porcentaje de varianza retenida.

se ve el impacto de la aplicación de PCA sobre el descriptor OHOG cuando se retiene un 90 % de información. Observamos, cómo el rendimiento se mantiene reduciendo a más de la mitad la dimensionalidad del vector de características que genera el descriptor OHOG.

Para obtener una visión de cuánto de agrupada está la información relevante para la clasificación, se ha realizado un estudio del rendimiento que se obtiene variando el porcentaje de información retenida. La realización de estos experimentos se han llevado a cabo sobre el descriptor OHOG y fijando los parámetros  $\eta = 4$  y  $\beta = 12$  para cada una de las vistas. Los resultados se pueden observar en la Tabla 4.16 y están gráficamente representados en la Figura 4.26. Se observa que en las vistas donde existe una mayor calidad de imagen, como las vistas frontal o derecha, la información relevante para la clasificación está agrupada en la primeras componentes de manera más significativa que en las otras vistas, llegando a conseguir buenos resultados manteniendo incluso el 50 % de la información, lo que conlleva a utilizar aproximadamente el 5 % de las componentes. Para el resto de vistas, el uso del 75 % de información es suficiente para obtener buenos resultados, lo que implica el uso de solamente el 20 % de componentes, que

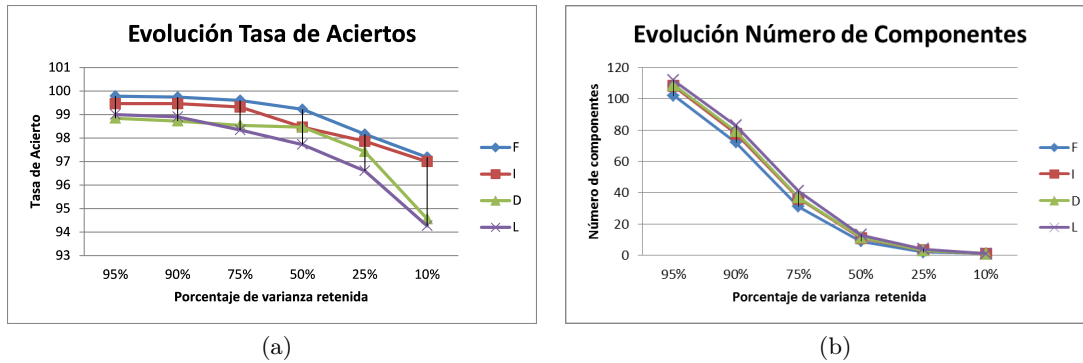


Figura 4.26: Evolución del rendimiento para cada vista. (a) en función del rendimiento y el porcentaje de varianza retenido; (b) en función del número de componentes y el porcentaje de varianza retenido.

Vista	Dimensionalidad del subespacio principal									
	10	20	30	40	50	60	70	80	90	100
F	99,18	99,32	99,34	99,24	99,50	99,72	99,64	99,78	99,68	<b>99,92</b>
I	98,58	98,50	98,54	98,48	98,56	99,26	99,40	99,38	<b>99,48</b>	99,26
D	98,38	98,24	98,18	98,56	98,40	98,74	98,94	<b>98,98</b>	98,84	98,80
L	97,60	97,40	97,38	97,46	97,34	98,58	98,54	98,80	<b>98,98</b>	98,80

Tabla 4.17: Rendimiento del sistema PCA-OHOG cambiando la dimensionalidad del espacio de características principal.

supone una reducción muy importante. No obstante, la vista lejana debido a su peor resolución de imagen sufre un mayor impacto al verse reducidas sus componentes, y aunque el uso del 75 % de información es suficiente para obtener buenos resultados, la retención del 90 % de varianza es más adecuado para esta vista.

En este punto, queremos ver el comportamiento del rendimiento del sistema como función de la dimensión del subespacio de características principal. Para ello se fija el número de componentes,  $M$ , y se obtiene el rendimiento para cada una de las vistas, buscando así la dimensión que maximice este rendimiento. La Tabla 4.17, representa la evolución del rendimiento cambiando la dimensión del subespacio de características principal. Cuando fijamos el número de componentes a utilizar, se puede observar que los valores máximos obtenidos corresponden al uso de prácticamente la mitad de las componentes. El uso de 60 componentes es el punto de inflexión a partir del cual empieza a decrecer significativamente el rendimiento, con más intensidad para las vistas izquierda y lejana.

Sin embargo, a pesar de los buenos resultados obtenidos y de reducir notablemente la dimensionalidad, lo que en principio se puede observar como un avance en la reducción del coste computacional y hacer de estos sistemas aptos para el uso en sistemas de tiempo real, en la

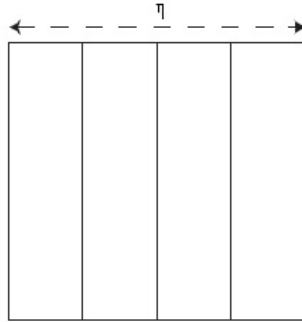


Figura 4.27: Configuración vertical de las celdas.

práctica no es así. El uso de PCA nos proporciona el uso de vectores de características mucho más reducido, lo que supone un procesado más rápido de la fase de clasificación. No obstante, la transformación del vector a este nuevo espacio de características, implica tiempos similares a los que conseguimos reducir utilizando una dimensionalidad del vector menor. Por lo que en la siguiente subsección se indagará en la reducción de la dimensionalidad más en profundidad.

#### 4.6.4. Descriptores eficientes basados en OHOG

El descriptor HOG conlleva un alto coste computacional lo que le hace poco eficiente a la hora de aplicarlo a sistemas en tiempo real. Lo que se busca en esta sección es reducir la dimensionalidad del vector y por ende la simplificación de algunos de los cálculos derivados, intentando conseguir el máximo rendimiento. Para ello se han estudiado dos posibilidades:

Una adaptación del descriptor V-HOG introducido en [35], y la exploración de topologías simplificadas del descriptor asociadas a cada vista.

##### 4.6.4.1. V-OHOG

El descriptor V-HOG [1], está basado en el descriptor SHOG, pero considerándose únicamente celdas verticales (Figura 4.27). Esta configuración de las celdas permite obtener vectores más cortos, lo que conlleva un menor coste computacional. Esta configuración ha sido elegida frente a otras configuraciones donde se consideraban solo celdas horizontales o concéntricas, por ser la que mejor rendimiento obtiene [35]. Esto se debe a la estructura rectangular de los coches, los cuales pueden ser claramente identificados por sus bordes verticales.

En este apartado se ha estudiado la configuración de este descriptor sobre el descriptor OHOG, obteniendo así el descriptor V-OHOG. No obstante, como ya se comentó en la subsección 4.6.2 cuando buscamos mejorar la eficiencia, la simplificación de no considerar la interpolación en la fase de cálculo de histogramas debe tenerse en cuenta. Por lo tanto, para evaluar el rendimiento de V-OHOG, se eliminará la interpolación en el cálculo de los histogramas. El rendimiento de

F	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$
$\eta = 2$	97,38	98,04	98,34	98,66
$\eta = 4$	98,48	99,06	99,30	99,58
$\eta = 8$	98,76	99,18	99,36	<b>99,62</b>
$\eta = 16$	98,56	99,52	99,38	99,56

I	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$
$\eta = 2$	94,84	95,50	96,06	97,58
$\eta = 4$	96,72	97,48	97,64	<b>98,62</b>
$\eta = 8$	97,74	97,86	98,20	98,38
$\eta = 16$	97,22	97,80	97,96	98,44

D	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$
$\eta = 2$	96,02	96,50	96,52	97,34
$\eta = 4$	96,86	97,38	97,36	97,64
$\eta = 8$	96,92	97,18	97,20	<b>97,80</b>
$\eta = 16$	96,38	96,94	96,68	97,54

L	$\beta = 8$	$\beta = 12$	$\beta = 16$	$\beta = 32$
$\eta = 2$	93,54	94,88	94,90	95,58
$\eta = 4$	96,52	97,20	97,44	97,48
$\eta = 8$	96,86	97,34	97,46	97,42
$\eta = 16$	96,38	97,02	<b>97,50</b>	97,48

Tabla 4.18: Rendimiento del descriptor V-OHOG.

este descriptor se puede ver en la Tabla 4.18. Si comparamos los resultados obtenidos por el descriptor V-OHOG con los obtenidos por el descriptor SHOG, vemos como el rendimiento de ambos es similar, siendo el descriptor V-OHOG más estable cuando variamos el número de celdas y el número de bins de orientación, con el beneficio de ser más eficiente que el descriptor estándar. Por otro lado, si comparamos este descriptor con el descriptor OHOG se observa como ha decrementado ligeramente el rendimiento. Cabe destacar que los mejores resultados se consiguen para valores altos de  $\beta$ .

En la Tabla 4.19a se muestra una comparativa de los mejores resultados obtenidos en los diferentes descriptores basados en HOG para cada una de las vistas, así como su número de componentes. La tabla muestra como el descriptor propuesto por Arróspide obtiene rendimientos inferiores en cuanto a la tasa de acierto, pero también con una importante disminución en la longitud del vector. Por el contrario el descriptor OHOG consigue el mejor rendimiento pero con vectores significativamente más largos. El descriptor propuesto basado en estos dos anteriores, el V-OHOG, tiene sus rendimientos máximos mejores que los del descriptor V-HOG. Para ver el rendimiento real del descriptor V-OHOG comparado con el V-HOG, en la Tabla 4.19b, se muestra el rendimiento del descriptor V-OHOG con vectores de longitud similar al V-HOG. Comparando estos resultados se comprueba que, al partir del descriptor óptimo para la realización del descriptor V-HOG, se consiguen resultados mejores debido a la adaptación de los parámetros al caso particular de los vehículos. Por otro lado, esta tabla también nos muestra el comportamiento del descriptor OHOG reduciendo el número de celdas con el fin de reducir la longitud del vector a valores similares a los del descriptor V-HOG. Los resultados obtenidos muestran rendimientos muy similares a los obtenidos por el descriptor V-OHOG, lo que nos sugiere que no es necesaria la construcción de un nuevo descriptor para obtener un sistema válido para situaciones en tiempo real.

	SHOG			V-HOG [1]			OHOG			V-OHOG		
Vista	TA	NC	$(\eta, \beta)$	TA	NC	$(\eta, \beta)$	TA	NC	$(\eta, \beta)$	TA	NC	$(\eta, \beta)$
F	99,48	128	(4,8)	97,68	64	(4,16)	99,80	256	(4,16)	99,62	256	(8,32)
I	97,64	128	(4,8)	97,02	144	(4,36)	99,62	768	(8,12)	98,62	128	(4,32)
D	96,22	256	(4,16)	95,54	64	(4,16)	99,20	2048	(8,32)	97,80	256	(8,32)
L	97,76	128	(4,8)	95,60	48	(4,12)	98,98	192	(4,12)	97,50	256	(16,16)

(a)

	OHOG			V-OHOG		
Vista	TA	NC	$(\eta, \beta)$	TA	NC	$(\eta, \beta)$
F	99,16	64	(2,16)	99,30	64	(4,16)
I	98,50	64	(2,16)	98,62	128	(4,32)
D	98,70	64	(2,16)	97,38	48	(4,12)
L	97,82	64	(2,16)	97,20	48	(4,12)

(b)

Tabla 4.19: Comparación entre los diferentes descriptores basados en HOG. (a) Comparativa de los mejores resultados; (b) Resultados de los métodos OHOG y V-OHOG correspondientes a valores similares del número de componentes de los mejores resultados de V-HOG.

#### 4.6.4.2. Influencias de las celdas en el descriptor OHOG

El segundo estudio realizado para encontrar un descriptor eficiente que se base en OHOG, es el análisis del impacto de las diferentes celdas sobre dicho descriptor en la clasificación. El objetivo es la eliminación de celdas que proveen información de carácter no significativo o que llevan a la confusión entre clases en clasificación. Por lo tanto, con esto conseguimos vectores más cortos que conllevan una clasificación más rápida y un menor coste computacional en la fase de extracción de características. Como en el caso de V-OHOG, la configuración del descriptor OHOG de la que partiremos, no hace uso de la interpolación en la fase de cálculo de histogramas para conseguir una mayor reducción del coste computacional.

La Figura 4.28 muestra para cada vista, la evolución del rendimiento de la verificación cuando las celdas menos significativas son eliminadas del vector (tablas adjuntas en la imagen) y la topología de las celdas eliminadas (celdas en gris). El proceso llevado a cabo es un proceso iterativo en el cual, en cada iteración, se identifica la celda con el impacto menor en los resultados y se elimina del descriptor. Por lo tanto, el número de celdas que componen el descriptor (NC en las tablas) se va reduciendo, las celdas eliminadas están indicadas en las tablas en la columna CE. La última columna de las tablas ( $\Delta$ ) muestra la desviación del rendimiento con respecto al rendimiento del descriptor completo. La primera conclusión que podemos extraer es que para todas las vistas, la eliminación de las celdas menos significativas mejoran la verificación. Eliminando la mitad de las celdas, es decir, quedándonos con 8 celdas obtenemos rendimientos similares



Frontal				Izquierda				Derecha				Lejana			
TA	NC	CE	$\Delta$	TA	NC	CE	$\Delta$	TA	NC	CE	$\Delta$	TA	NC	CE	$\Delta$
99,80	16	-	-	99,54	16	-	-	98,90	16	-	-	98,64	16	-	-
99,84	15	7	+0,04	99,54	15	11	+0,00	99,12	15	8	+0,22	98,78	15	3	+0,14
99,86	14	10	+0,06	99,62	14	1	+0,08	99,12	14	5	+0,22	98,80	14	1	+0,16
99,84	13	11	+0,04	99,54	13	15	+0,00	99,12	13	3	+0,22	98,84	13	2	+0,20
99,82	12	14	+0,02	99,58	12	8	+0,04	99,10	12	14	+0,20	98,84	12	5	+0,20
99,90	11	2	+0,10	99,52	11	9	-0,02	99,06	11	7	+0,16	98,76	11	8	+0,12
99,86	10	3	+0,06	99,58	10	3	+0,04	99,06	10	2	+0,16	98,72	10	11	+0,08
99,88	9	6	+0,08	99,54	9	2	+0,00	98,96	9	9	+0,06	98,60	9	4	-0,04
99,82	8	16	+0,02	99,50	8	5	-0,04	98,92	8	4	+0,02	98,52	8	15	-0,12

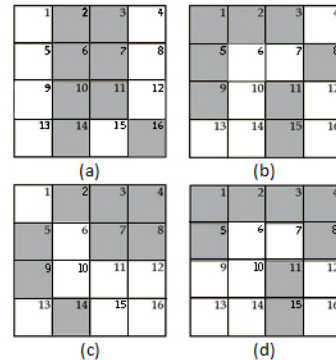


Figura 4.28: Evolución del rendimiento en clasificación eliminando las celdas menos significativas del descriptor para cada una de las vistas. En gris se presentan las celdas eliminadas del vector. Los parámetros escogidos para llevar a cabo este estudio son  $\eta = 4$  y  $\beta = 16$ .

al descriptor OHOG para todas las vistas, siendo las vistas más afectadas, la vista izquierda y lejana. Además, se consigue un descriptor suficientemente rápido para su implementación en sistemas de tiempo real. Otra conclusión relevante es que la topología de las celdas que tienen un impacto negativo en la verificación está relacionada con las particularidades de la posición del vehículo en la imagen. Las celdas más importantes para la vista frontal (Figura 4.28 a) son las exteriores, ya que típicamente contienen los bordes más relevantes de la parte trasera de los vehículos cuando se les observa frontalmente, y por lo tanto proporcionan información más discriminativa. Para la vista izquierda (Figura 4.28 b), la información más relevante se encuentra tanto en las celdas de la diagonal de derecha a izquierda como en las celdas que se encuentran debajo de ésta, lo cuál concuerda con las áreas de la imagen que proporcionan más información del vehículo cuando es observada desde la derecha. Conclusiones similares se pueden obtener a partir de la distribución de las celdas para la vista derecha (Figura 4.28 c), pero teniendo en cuenta la diagonal de izquierda a derecha. Por último, para las imágenes de vista lejana (Figura 4.28 d), las cuáles muestran bordes suavizados debido a la interpolación (las imágenes están escaladas a 64x64 píxeles), las celdas de la parte inferior son las más discriminativas, ya que contienen información de bordes más estable y contrastada.

En conclusión, el estudio de la influencia de las celdas en los descriptores OHOG ayuda a entender que partes de la imagen son más decisivas para la clasificación. Además, podemos asegurar que el uso del descriptor completo OHOG, que tiene un coste computacional elevado, es innecesario. El uso de una parte del descriptor OHOG es suficiente para la correcta clasificación. No obstante, el descriptor obtenido realiza la selección de las celdas que mejor se comportan para las imágenes de la base de datos que estamos utilizando. Así, a partir de la topología que siguen las celdas eliminadas y de la posición de los vehículos en las imágenes, se proponen una máscaras estándar para cada una de las vistas, como se muestra en la Figura 4.29. Como ya

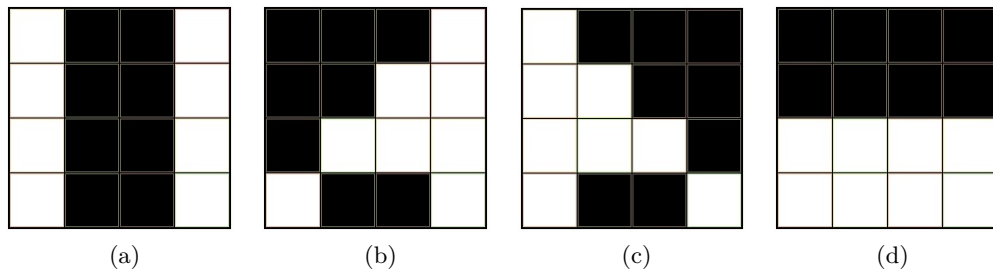


Figura 4.29: Máscaras propuestas para cada una de las vistas de la base de datos. (a) Frontal (b) Izquierda (c) Derecha (d) Lejana. Donde solo se tienen en cuenta las celdas de color blanco.

	Vista				NC
	F	I	D	L	
V-OHOG	99,36	98,20	97,20	97,46	128
T-OHOG	99,50	99,42	98,82	98,18	128
OHOG	99,52	99,06	98,38	98,16	128
T-OHOG	99,62	99,34	98,44	98,30	96

Tabla 4.20: Comparación entre los descriptores eficientes propuestos.

se ha mencionado, en la vista frontal las celdas más significativas corresponden a los laterales del vehículo, por lo que se decidió el uso de estas celdas para la clasificación. En relación a las vistas laterales, se realizó el mismo análisis para ambas, manteniendo las celdas de la diagonal correspondiente, como las celdas de debajo de ésta, eliminando además las celdas que típicamente contienen las sombras proyectadas por los vehículos. Para la vista lejana, sólo se consideran las celdas de la mitad inferior que corresponden a los bordes de las ruedas, luces o sombras proyectadas de los vehículos, que en esta vista son los más importantes.

La Tabla 4.20, realiza una comparación entre los resultados de las máscaras propuestas para el descriptor OHOG, denominadas en la tabla T-OHOG (Topology OHOG), el descriptor V-OHOG y el descriptor OHOG. Los resultados indican cómo T-OHOG, consiguen resultados superiores a V-OHOG. Sin embargo, con la configuración adecuada de los parámetros del descriptor OHOG podemos conseguir un rendimiento y una longitud del vector muy similar al descriptor T-OHOG. Por último, en la tabla se muestra el uso del descriptor T-OHOG con un valor de  $\beta = 12$ , lo que consigue mantener el rendimiento para la mayoría de las vistas reduciendo aún más la longitud del vector. El descriptor más adecuado para sistemas en tiempo real es T-OHOG, ya que consigue resultados similares a los obtenidos por OHOG, siendo más eficiente, debido a la eliminación de la interpolación y al uso de vectores más cortos.

La Figura 4.30 muestra algunos ejemplos de vehículos mal clasificados haciendo uso del descriptor T-HOG, que ilustran la dependencia en clasificación con la calidad de las hipótesis generadas.

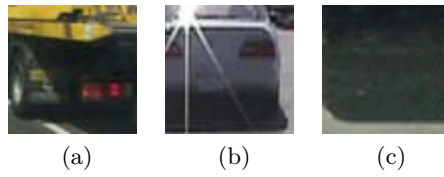


Figura 4.30: Hipótesis con (a) vista parcial, (b) bajo contraste, y (c) distorsión.

Por último, en términos de la tasa de verificación media, es decir la media de la tasa de verificación para todas las vistas, nuestro sistema consigue un 98,98 %, un resultado excelente que ampliamente supera los resultados presentados en [94] (92,9 %), y [62](94 %), que proponen una cascada de clasificadores potenciados (*boosted classifiers*) combinados con características Haar-like y HOG.

## Capítulo 5

# Conclusiones y trabajo futuro

### 5.1. Conclusiones

En este trabajo se ha llevado a cabo una aproximación de un sistema de verificación de vehículos basado en visión monocular. El objetivo es obtener un sistema que minimice la complejidad del problema, especialmente las variaciones extrínsecas e intrínsecas de los patrones de los vehículos y las restricciones dadas por realizar un sistema en tiempo real. En particular se han evaluado métodos de extracción de características, encontrados en el estado del arte, y se han propuesto nuevos descriptores con el objetivo de incrementar el rendimiento y de obtener una mayor eficiencia. Una característica importante de nuestro sistema es que se ha realizado dependiente de la vista, es decir, las imágenes a verificar se clasifican dependiendo de la posición del vehículo en ella, consiguiendo así un incremento del rendimiento en estos sistemas. Gracias a los experimentos se ha podido comprobar qué tipo de características son las más adecuadas para resolver el problema de la verificación de vehículos. Los descriptores basados en HOG han resultado ser los que mejores rendimientos ofrecen. Estos descriptores ya se han utilizado en el estado del arte para la verificación de vehículos, no obstante la configuración de éstos no era la más adecuada para el problema en cuestión. Como se demuestra, estos descriptores con la configuración adecuada de sus parámetros ofrecen un rendimiento excelente, si bien llevan asociado un alto coste computacional. Con el fin de mejorar la eficiencia, se proponen configuraciones del descriptor simplificadas, manteniendo un alto rendimiento en la clasificación, lo que hace de estos descriptores una excelente opción para establecer sistemas de verificación de vehículos basados exclusivamente en una única característica. En relación con los otros métodos explorados, queda demostrado que las características explícitas no consiguen llegar al rendimiento ofrecido por las características implícitas. Bien es cierto que las características explícitas tienen asociado un menor coste computacional, por lo que si se desea un sistema rápido, donde el rendimiento pueda quedar en segundo plano, los descriptores propuestos de bordes, en concreto los basados en máximos, resultan una opción válida con la que se consigue un rendimiento medio superior

al 85 %. Por otro lado los sistemas basados en simetría obtienen los rendimientos más bajos, además su tiempo de ejecución es mayor al ofrecido por otro tipo de características como bordes. Esta situación hace que la simetría vea relegado su uso a sistemas que combinan diferentes características. En relación con el entrenamiento y el test de los clasificadores, es necesario el uso de una base de datos amplia y con alta variabilidad con el fin de obtener una aproximación lo más fiel posible a situaciones reales. Este hecho se ha perfilado como uno de los grandes problemas en el diseño de este tipo de sistemas, ya que los trabajos encontrados utilizan para su evaluación diferentes datos (bases de datos de imágenes o secuencias de vídeo) que además no suelen estar disponibles para la comunidad científica. Con la aparición de la base de datos GTI, se ofrece un conjunto de datos público que permite analizar comparativamente estrategias de verificación. La base de datos, que además tiene previsiones de ser ampliada en un futuro cercano, ofrece un muy significativo número de muestras con gran variabilidad, lo que hace de ella una de las mejores opciones para establecer un marco comparativo de diferentes estrategias.

En resumen, en este trabajo se han presentado las siguientes contribuciones: en el campo de la extracción de características explícitas, simetría y bordes, diferentes estrategias de verificación que si bien mejoran algunos de los trabajos encontrados en la literatura, no ofrecen todavía un rendimiento suficiente como para ser utilizadas de manera individual. El desarrollo de un sistema basado en el análisis de componentes principales que consigue mediante el uso de la matriz de correlaciones aumentar el rendimiento de aproximaciones similares. No obstante, las contribuciones más importantes se han llevado a cabo en el desarrollo de un sistema basado en HOG con dos propuestas diferenciadas. La primera dirigida a conseguir un aumento del rendimiento adaptando la configuración del descriptor completo así como las etapas de extracción de la información necesaria para su obtención. La segunda propone descriptores eficientes, donde se elimina la interpolación y se reduce el número de celdas en configuraciones definidas en función del tipo de vista, teniendo un impacto mínimo en el rendimiento global del sistema de verificación. Estas dos contribuciones han dado lugar a dos artículos aceptados para su presentación en dos conferencias internacionales: ICIP'2014 y EUSIPCO'2014, mostrados en el Apéndice B.

Por lo tanto, este trabajo ofrece una aproximación de los métodos de verificación de vehículos más utilizados en la literatura, buscando la mejora de estos. Se ha conseguido dar una visión detallada de estos métodos, mostrando sus limitaciones al igual que sus puntos fuertes, esto puede resultar de gran ayuda para investigaciones futuras en este campo.

## 5.2. Trabajo Futuro

El trabajo descrito en este documento es un paso más en la dirección para la mejora de los rendimientos obtenidos en sistemas de verificación de vehículos. No obstante, el problema propuesto no está completamente resuelto. En esta sección se presentan algunas líneas de trabajo

futuro que surgen como una extensión natural de los trabajos presentados.

En relación a las características estudiadas, concretamente las características explícitas, trabajos futuros pueden realizar el estudio de éstas más en profundidad, buscando vectores de características más completos, como en el caso de la simetría donde se ha evaluado un vector de características formado por una dimensión relativa al valor de simetría de la imagen. Puede resultar interesante que estos vectores estén compuestos por más valores, como puede ser la posición del eje o el ancho del vehículo. Al igual que en simetría, puede resultar interesante en bordes añadir más información al vector de características. Valores como el ancho esperado de un vehículo o la posición del eje de simetría respecto a los máximos (posiciones de los bordes verticales de la estructura del vehículo), pueden formar parte del vector de características, también se pueden utilizar estos valores como parámetros de una función que genere un valor que ayude a una mejor caracterización de los vehículos o que penalice el valor de los bordes extraídos, consiguiendo una mayor diferencia entre clases.

La combinación de los descriptores propuestos podría aumentar el rendimiento obtenido, en concreto se propone realizar un estudio en profundidad, poniendo como punto de partida el descriptor T-OHOG y buscando la combinación adecuada para conseguir un clasificador robusto que clasifique de manera satisfactoria aquellas muestras que mediante el uso de un único clasificador se clasifican erróneamente. Además, se propone la realización de los descriptores OHOG y T-OHOG mediante el método de histograma integral descrito en [95].

Por otro lado, en el campo de la verificación de vehículos encontramos otros descriptores como los filtros de Gabor, o características Haar-like que podrían ser evaluados. En trabajos del estado del arte se proponen sistemas que combinan las características Haar-like con HOG mediante el uso de clasificadores potenciados en cascada, lo que podría ser también una nueva línea de investigación haciendo uso de los descriptores HOG propuestos.

En este trabajo también se han probado diferentes clasificadores, con especial interés en las SVM. De hecho éstas se presentan en la literatura como la mejor elección, por ofrecer mejor generalización que otros clasificadores, tales como redes neuronales, y conseguir un mejor rendimiento general para la verificación de vehículos. No obstante se propone como trabajo futuro la investigación de la configuración óptima de las SVM y la evaluación de otras aproximaciones de aprendizaje.

Al igual que ocurre al realizar la división para las diferentes vistas, se propone la división de la base de datos en diferentes subgrupos, en los que se consideren las condiciones meteorológicas y de iluminación. Con esto se pretende conseguir un aumento de rendimiento, ya que se consigue una mejor interpretación de las clases. De hecho, muchos vehículos actualmente incorporan sensores que permiten identificar situaciones de lluvia, oscurecimiento, etc., de manera que la selección en cada caso del clasificador más adecuado se presenta como una opción atractiva para mejorar el rendimiento. Por otro lado, también podría ser muy interesante realizar una etapa

de reconocimiento, posterior a la verificación de hipótesis, en la cuál se clasifiquen las imágenes en diferentes categorías, por ejemplo: coches, furgonetas, camiones y autobuses. Esto requiere la definición de nuevos descriptores, al igual que el incremento de la base de datos. Por último, se propone la posibilidad de incluir en el sistema la verificación de vehículos de dos ruedas.







# Bibliografía

- [1] Jon Arróspide Laborda. Vision-based vehicle detection and tracking with a mobile camera using statistical framework. Master's thesis, 2012. v, v, 3, 4, 5, 6, 11, 13, 14, 20, 69, 83, 85
- [2] Zehang Sun, G. Bebis, and R. Miller. On-road vehicle detection: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(5):694–711, May 2006. v, 1, 6, 7, 9, 10
- [3] Gwang Yul Song, Ki Yong Lee, and Joon Woong Lee. Vehicle detection by edge-based candidate generation and appearance-based classification. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 428–433, June 2008. v, 6, 19, 20
- [4] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science+Business Media, 2006. VI, VI, VI, 24, 25, 31, 35, 36, 39, 40, 41, 43
- [5] Gerardo Colmenares. Inteligencia artificial. capítulo iv. máquinas de vectores de soporte, 2000. VI, 37, 38
- [6] Asa Ben-Hur, Jason Weston, Department of Computer, Science NEC, and Labs America. A user's guide to support vector machines. 1992. VI, VII, 42, 45
- [7] J. Arrospide, L. Salgado, and J. Marinas. Hog-like gradient-based descriptor for visual vehicle detection. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 223–228, June 2012. IX, 58, 59
- [8] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta. Artificial vision in road vehicles. *Proceedings of the IEEE*, 90(7):1258–1271, Jul 2002. 1
- [9] M. Bertozzi, A. Broggi, A. Fascioli, and S. Nichele. Stereo vision-based vehicle detection. In *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, pages 39–44, 2000. 4, 18
- [10] D. Alonso, L. Salgado, and M. Nieto. Robust vehicle detection through multidimensional classification for on board video based systems. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 4, pages IV – 321–IV – 324, Sept 2007. 6, 13, 16
- [11] Andreas Kuehnl. Symmetry-based recognition of vehicle rears. *Pattern Recognition Letters*, 12(4):249 – 258, 1991. 6, 17
- [12] T. Schamm, C. von Carlowitz, and J.M. Zollner. On-road vehicle detection during dusk and at night. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 418–423, June 2010. 6, 15

- [13] A. Broggi, M. Bertozzi, A. Fascioli, C.G.L. Bianco, and A. Piazzzi. Visual perception of obstacles and vehicles for platooning. *Intelligent Transportation Systems, IEEE Transactions on*, 1(3):164–176, Sep 2000. 6, 12
- [14] F. Woelk, S. Gehrig, and R. Koch. A monocular collision warning system. In *Computer and Robot Vision, 2005. Proceedings. The 2nd Canadian Conference on*, pages 220–227, May 2005. 7
- [15] T.N. Tan and K.D. Baker. Efficient image gradient based vehicle localization. *Image Processing, IEEE Transactions on*, 9(8):1343–1356, Aug 2000. 7, 12
- [16] S. Gupte, O. Masoud, R.F.K. Martin, and N.P. Papanikolopoulos. Detection and classification of vehicles. *Intelligent Transportation Systems, IEEE Transactions on*, 3(1):37–47, Mar 2002. 7
- [17] S. Kubota, T. Nakano, and Y. Okamoto. A global optimization algorithm for real-time on-board stereo obstacle detection systems. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 7–12, June 2007. 8, 9
- [18] Van Hulle M. M. Chumerin, N. Cue and sensor fusion for independent moving objects detection and description in driving scenes. *Signal Processing Techniques for Knowledge Extraction and Information Fusion. Springer, Boston, USA*, pages 161–180, 2008. 8
- [19] N. Simond. Reconstruction of the road plane with an embedded stereo-rig in urban environments. In *Intelligent Vehicles Symposium, 2006 IEEE*, pages 70–75, 2006. 9
- [20] U. Franke and I. Kutzbach. Fast stereo based object detection for stop and go traffic. In *Intelligent Vehicles Symposium, 1996., Proceedings of the 1996 IEEE*, pages 339–344, Sep 1996. 9
- [21] Bühlhoff H. H. Little J. J. Bohrer S. Mallot, H. A. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological Cybernetics*, 64(3):177–185, 1991. 9, 10
- [22] M. Bertozzi and A. Broggi. Gold: a parallel real-time stereo vision system for generic obstacle and lane detection. *Image Processing, IEEE Transactions on*, 7(1):62–81, Jan 1998. 10
- [23] Zisserman A. Hartley, R. I. *Multiple View Geometry in Computer Vision*. 2nd Edition. Cambridge University Press, 2004. 10
- [24] Guo-Wei Zhao and S. Yuta. Obstacle detection by vision system for an autonomous vehicle. In *Intelligent Vehicles '93 Symposium*, pages 31–36, Jul 1993. 10
- [25] Parent M. Simond, N. Obstacle detection by vision system for autonomous vehicle. *Proc. International Conference on Intelligent Robots and Systems*, pages 4283–4288, 2007. 10
- [26] W. Kruger, W. Enkelmann, and S. Rossle. Real-time estimation and tracking of optical flow vectors for obstacle detection. In *Intelligent Vehicles '95 Symposium., Proceedings of the*, pages 304–309, Sep 1995. 11
- [27] T. Naito, T. Ito, and Y. Kaneda. The obstacle detection method using optical flow estimation at the edge image. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 817–822, June 2007. 11
- [28] Koch R. Woelk, F. Robust monocular detection of independent motion by a moving observer. In *Proc. International Conference on Complex Motion*, pages 209–222, 2004. 11

- [29] K. Yamaguchi, A. Watanabe, T. Naito, and Y. Ninomiya. Road region estimation using a sequence of monocular images. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, Dec 2008. 11
- [30] Jianguang Lou, Tieniu Tan, Weiming Hu, Hao Yang, and S.J. Maybank. 3-d model-based vehicle tracking. *Image Processing, IEEE Transactions on*, 14(10):1561–1569, Oct 2005. 12
- [31] Luo-Wei Tsai, Jun-Wei Hsieh, and Kao-Chin Fan. Vehicle detection using normalized color and edge map. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages II–598–601, Sept 2005. 12
- [32] Daniilidis K. Nagel H.-H. Koller, D. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3):257–281, 1993. 13
- [33] J.M. Collado, C.H. Hilario, A. de la Escalera, and J.M. Armingol. Model based vehicle detection for intelligent vehicles. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 572–577, June 2004. 13
- [34] Zhaoxiang Zhang, Tieniu Tan, Kaiqi Huang, and Yunhong Wang. Three-dimensional deformable-model-based localization and recognition of road vehicles. *Image Processing, IEEE Transactions on*, 21(1):1–13, Jan 2012. 13
- [35] Jon Arróspide, Luis Salgado, and Massimo Camplani. Image-based on-road vehicle detection using cost-effective histograms of oriented gradients. *Journal of Visual Communication and Image Representation*, 24(7):1182 – 1190, 2013. 13, 23, 74, 75, 77, 83
- [36] I. Cabani, G. Toulminet, and A. Bensrhair. Color-based detection of vehicle lights. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 278–283, June 2005. 14
- [37] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001. 15
- [38] J.M. Mossi, A. Albiol, A. Albiol, and V.N. Ornedo. Real-time traffic analysis at night-time. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2941–2944, Sept 2011. 15
- [39] M. Casares, A. Almagambetov, and S. Velipasalar. A robust algorithm for the detection of vehicle turn signals and brake lights. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 386–391, Sept 2012. 15
- [40] R. O’Malley, E. Jones, and M. Glavin. Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):453–462, June 2010. 15
- [41] T. Fang L. Gao, C. Li and Z. Xiong. Vehicle detection based on color and edge information. *Image Analysis and Recognition, ser. Lecture Notes in Computer Science*, 5112:142–150, 2008. 15, 68
- [42] H. Mori and N.M. Charkari. Shadow and rhythm as sign patterns of obstacle detection. In *Industrial Electronics, 1993. Conference Proceedings, ISIE’93 - Budapest., IEEE International Symposium on*, pages 271–277, 1993. 15
- [43] C. Hoffman, T. Dang, and C. Stiller. Vehicle detection fusing 2d visual features. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 280–285, June 2004. 15

- [44] Kah Phooi SENG King Hann LIM, Li-Minn ANG and Siew Wen CHIN. Lane-vehicle detection and tracking. *Proceedings of the International MultiConference of Engineers and Computer Scientists, IMECS*, 2:18–20, March 2009. 16
- [45] L. Salgado J. Arróspide. A survey and on computer and vision techniques and for and on-road vehicle and detection and tracking. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS* , VOL. XXX, NO. YYY, OCTOBER, 2012. 16
- [46] T. Graf C. Rotaru and J. Zhang. Color image segmentation in hsi space for automotive applications. *Journal of Real-Time Image Processing*, 3(4):311–322, 2008. 16
- [47] Luo-Wei Tsai, Jun-Wei Hsieh, and Kuo-Chin Fan. Vehicle detection using normalized color and edge map. *Image Processing, IEEE Transactions on*, 16(3):850–864, March 2007. 16
- [48] A. Russell and Ju Jia Zou. Vehicle detection based on color analysis. In *Communications and Information Technologies (ISCIT), 2012 International Symposium on*, pages 620–625, Oct 2012. 16
- [49] A. Broggi M. Bertozzi and S. Castelluccio. A real-time oriented system for vehicle detection. *Journal of Systems Architecture*, 43(1-5):317–325, 1997. 17
- [50] Yixin Chen, M. Das, and D. Bajpai. Vehicle tracking and distance estimation based on multiple image features. In *Computer and Robot Vision, 2007. CRV '07. Fourth Canadian Conference on*, pages 371–378, May 2007. 17
- [51] Y. Du and N.P. Papanikolopoulos. Real-time vehicle following through a novel symmetry-based approach. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 4, pages 3160–3165 vol.4, Apr 1997. 18, 29, 49
- [52] T. K. Ten Kate, M. B. Van Leewen, S. E. Moro-Ellenberger, B. J F Driessen, A. H G Versluis, and F. C A Groen. Mid-range and distant vehicle detection with a mobile camera. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 72–77, June 2004. 18, 29, 50
- [53] Wei Liu, Xuezhi Wen, Bobo Duan, Huai Yuan, and Nan Wang. Rear vehicle detection and tracking for lane change assist. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 252–257, June 2007. 18
- [54] Rafael C. Gonzales and Richard E. Woods. *Digital Image Processing. 2nd ed. Englewood Cliffs*. Prentice-Hall, 2002. 19, 30
- [55] N. Srinivasa. Vision-based vehicle detection and tracking method for forward collision warning in automobiles. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 2, pages 626–631 vol.2, June 2002. 19
- [56] Jingwei Liu, Yong Zhao, Yule Yuan, Wei Luo, and Kai Liu. Vehicle capturing and counting using a new edge extraction approach. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 62–66, Oct 2011. 19
- [57] J. Xin T. Wang, N. Zheng and Z. Ma. Integrating millimeter wave radar with a monocular vision sensor for on-road obstacle detection applications. *Sensors*, 11(9):8992–9008, 2011. 19
- [58] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 746–751 vol.1, 2000. 20

- [59] Zehang Sun, G. Bebis, and R. Miller. Monocular precrash vehicle detection: features and classifiers. *Image Processing, IEEE Transactions on*, 15(7):2019–2034, July 2006. 20, 22, 23, 26, 70
- [60] Poggio T. Papageorgiou, C. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000. 21
- [61] Jones M. J. Viola, P. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2001. 21
- [62] Clady X. Hanif S. M. Prevost L. Negri, P. A cascade of boosted generative and discriminative classifiers for vehicle detection. *EURASIP Journal on Advances in Signal Processing*, 782432:12, 2008. 21, 88
- [63] T.T. Son and S. Mita. Car detection using multi-feature selection for varying poses. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 507–512, June 2009. 21
- [64] D. Gabor. Theory of communication. *Journal of Institute for Electrical Engineering 93, part III (26)*, pages 429–457, 1946. 22
- [65] Thomas P. Weldon, William E. Higgins, and Dennis F. Dunn. Gabor filter design for multiple texture segmentation. *Optical Engineering*, 35:2852–2863, 1996. 22
- [66] Biswas P. K. Chatterji B. N. Manthalkar, R. Rotation invariant texture classification using even symmetric gabor filters. *Pattern Recognition Letters*, 24(12):2061–2068, 2003. 22
- [67] Ma K.-K. Han, J. Rotation-invariant and scale-invariant gabor features for texture image retrieval. *Image and Vision Computing*, 25(9):1474–1481, 2007. 22
- [68] Czyzewski A. Dalka, P. Vehicle classification based on soft computing algorithms. in: rough sets and current trends in computing. *Lecture Notes in Computer Science*, 6086:70–79, 2010. 22, 26
- [69] Zehang Sun, G. Bebis, and R. Miller. On-road vehicle detection using evolutionary gabor filter optimization. *Intelligent Transportation Systems, IEEE Transactions on*, 6(2):125–137, June 2005. 22
- [70] Thiran J.-P. Popovici, V. Face detection using an svm trained in eigenfaces space. in: Audio- and video-based biometric person authentication. *Lecture Notes in Computer Science*, 2688:190–198, 2003. 22
- [71] Junwen Wu and Xuegong Zhang. A pca classifier and its application in vehicle detection. In *Neural Networks, 2001. Proceedings. IJCNN '01. International Joint Conference on*, volume 1, pages 600–604 vol.1, 2001. 23
- [72] Chengcui Zhang, Xin Chen, and Wei-Bang Chen. A pca-based vehicle classification framework. In *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*, pages 17–17, 2006. 23
- [73] J. Arrospide and L. Salgado. Region-dependent vehicle classification using pca features. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 453–456, Sept 2012. 23, 26, 69, 70, 72

- [74] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005. 23, 26, 32, 72, 75, 77
- [75] O. Daniz, G. Bueno, J. Salido, and F. De la Torre. Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 32(12):1598 – 1603, 2011. 23
- [76] Pengpeng Liang, Gregory Teodoro, Haibin Ling, Erik Blasch, Genshe Chen, and Li Bai. Multiple kernel learning for vehicle detection in wide area motion imagery. pages 1629–1636, 2012. 23
- [77] Seonyoung Lee, Haengseon Son, Jong Chan Choi, and Kyoungwon Min. Hog feature extractor circuit for real-time human and vehicle detection. In *TENCON 2012 - 2012 IEEE Region 10 Conference*, pages 1–5, Nov 2012. 23
- [78] Bill Triggs Navneet Dalal. Object detection using histograms of oriented gradients. In *Pascal VOC 2006 Workshop, ECCV 2006, Graz, Austria, 2006*. 23, 75, 77
- [79] Yang Lv, B. Yao, Yongtian Wang, and Song-Chun Zhu. Reconfigurable templates for robust vehicle detection and classification. In *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, pages 321–328, Jan 2012. 23, 26
- [80] Xianbin Cao, Changxia Wu, Pingkun Yan, and Xuelong Li. Linear svm classification using boosting hog features for vehicle detection in low-altitude airborne videos. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2421–2424, 2011. 23, 26
- [81] Ilkay Ulusoy and Christopher M. Bishop. Comparison of generative and discriminative techniques for object detection and classification. In *In Proc. Sicily Workshop on Object Recognition, 2006*. 24
- [82] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. 2002. 24
- [83] Guillaume Bouchard and Bill Triggs. The trade-off between generative and discriminative classifiers. In *Proceedings in Computational Statistics, 16th Symposium of IASC*, pages 721–728. Physica-Verlag, 2004. 25
- [84] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery 2*, pages 121–167, 1998. 26
- [85] Chi-Chen Raxle Wang and J.-J.J. Lien. Automatic vehicle detection using local features: A statistical approach. *Intelligent Transportation Systems, IEEE Transactions on*, 9(1):83–96, March 2008. 26
- [86] Welling M.-Perona P. Weber, M. Unsupervised learning of models for recognition. *Computer Vision - ECCV 2000, Lecture Notes in Computer Science*, 1842:18–32, 2000. 26
- [87] Thomas Zielke, Michael Brauckmann, and Werner von Seelen. Intensity and edge-based symmetry detection with an application to car-following. *CVGIP: Image Underst.*, 58(2):177–190, September 1993. 29
- [88] J. Mercer. Functions of Positive and Negative Type, and Their Connection with the Theory of Integral Equations. *Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences*, 83:69–70, 1909. 38

- [89] Gti vehicle image database, image processing group at upm, <<http://www.gti.ssr.upm.es/data>>, 2011. 47
- [90] M. Boumediene, A. Ouamri, and M. Keche. Vehicle detection algorithm based on horizontal/vertical edges. In *Systems, Signal Processing and their Applications (WOSSPA), 2011 7th International Workshop on*, pages 396–399, May 2011. 68
- [91] Universidad de Sevilla. Métodos de descomposición lineal de las observaciones. 69
- [92] Jiafu Jiang and Hui Xiong. Fast pedestrian detection based on hog-pca and gentle adaboost. In *Computer Science Service System (CSSS), 2012 International Conference on*, pages 1819–1822, Aug 2012. 80
- [93] Wei-Lwun Lu and J.J. Little. Simultaneous tracking and action recognition using the pca-hog descriptor. In *Computer and Robot Vision, 2006. The 3rd Canadian Conference on*, pages 6–6, June 2006. 80
- [94] D. Balcones, D.F. Llorca, M.A. Sotelo, M. Gavilán, S. Álvarez, I. Parra, and M. Ocaña. Real-time vision-based vehicle detection for rear-end collision mitigation systems. In Roberto Moreno Díaz, Franz Pichler, and Alexis Quesada-Arencia, editors, *Computer Aided Systems Theory - EUROCAST 2009*, volume 5717 of *Lecture Notes in Computer Science*, pages 320–325. Springer Berlin Heidelberg, 2009. 88
- [95] F. Porikli. Integral histogram: a fast way to extract histograms in cartesian spaces. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 829–836 vol. 1, June 2005. 91





# Apéndice A

## Resultados

### A.1. Simetría

#### A.1.1. Método 1

Clasificador Lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	71,80	0,6674	0,8692	0,1308	0,4332	0,8692	0,5668
I	65,24	0,6686	0,6044	0,3956	0,2996	0,6044	0,7004
D	59,20	0,5821	0,6524	0,3476	0,4684	0,6524	0,5316
L	65,46	0,6977	0,5456	0,4544	0,2364	0,5456	0,7636

Clasificador Bayesiano							
Vista	TA	P	R	FN	FP	TP	TN
F	78,54	0,7503	0,8556	0,1444	0,2848	0,8556	0,7152
I	71,66	0,6815	0,8132	0,1868	0,3800	0,8132	0,6200
D	64,54	0,6099	0,8068	0,1932	0,5160	0,8068	0,4840
L	76,02	0,8307	0,6536	0,3464	0,1332	0,6536	0,8668

Clasificador SVM lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	79,80	0,7980	0,7980	0,2020	0,2020	0,7980	0,7980
I	75,52	0,7976	0,6840	0,3160	0,1736	0,6840	0,8264
D	68,22	0,6800	0,6884	0,3116	0,3240	0,6884	0,6760
L	75,80	0,8595	0,6168	0,3832	0,1008	0,6168	0,8992

Clasificador SVM no lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	79,46	0,7995	0,7864	0,2136	0,1972	0,7864	0,8028
I	75,42	0,7848	0,7004	0,2996	0,1920	0,7004	0,8080
D	68,90	0,7191	0,6204	0,3796	0,2424	0,6204	0,7576
L	76,08	0,8517	0,6316	0,3684	0,1100	0,6316	0,8900

### A.1.2. Método 2 sin mejora

Clasificador Lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	72,64	0,6850	0,8370	0,1630	0,3840	0,8370	0,6160
I	72,23	0,7030	0,7690	0,2310	0,3240	0,7690	0,6760
D	69,65	0,7000	0,6870	0,3130	0,2940	0,6870	0,7060
L	74,35	0,7100	0,8240	0,1760	0,3370	0,8240	0,6630

Clasificador Bayesiano							
Vista	TA	P	R	FN	FP	TP	TN
F	79,10	0,7910	0,7910	0,2090	0,2090	0,7910	0,7910
I	80,20	0,8100	0,7890	0,2110	0,1850	0,7890	0,8150
D	78,54	0,7750	0,8050	0,1950	0,2340	0,8050	0,7660
L	80,62	0,8180	0,7870	0,2130	0,1750	0,7870	0,8250

Clasificador SVM lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	78,40	0,8640	0,6740	0,3260	0,1060	0,6740	0,8940
I	80,35	0,8520	0,7340	0,2660	0,1270	0,7340	0,8730
D	79,96	0,8320	0,7510	0,2490	0,1520	0,7510	0,8480
L	81,02	0,8620	0,7380	0,2620	0,1180	0,7380	0,8820

Clasificador SVM no lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	81,06	0,8270	0,7850	0,2150	0,1640	0,7850	0,8360
I	80,02	0,8200	0,7700	0,2300	0,1690	0,7700	0,8310
D	80,00	0,8200	0,7740	0,2260	0,1740	0,7740	0,8260
L	81,16	0,8560	0,7490	0,2510	0,1260	0,7490	0,8740

### A.1.3. Método 2 con mejora

Clasificador Lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	89,67	0,8890	0,9060	0,0940	0,1130	0,9060	0,8870
I	76,84	0,7780	0,7510	0,2490	0,2140	0,7510	0,7860
D	80,33	0,7940	0,8190	0,1810	0,2130	0,8190	0,7870
L	78,83	0,7810	0,8010	0,1990	0,2250	0,8010	0,7750

Clasificador Bayesiano							
Vista	TA	P	R	FN	FP	TP	TN
F	88,85	0,8540	0,9380	0,0620	0,1610	0,9380	0,8390
I	80,50	0,7550	0,9030	0,0970	0,2930	0,9030	0,7070
D	76,18	0,7000	0,9170	0,0830	0,3940	0,9170	0,6060
L	76,02	0,7090	0,8840	0,1160	0,3640	0,8840	0,6360

Clasificador SVM lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	90,14	0,8920	0,9130	0,0870	0,1100	0,9130	0,8900
I	82,30	0,8040	0,8550	0,1450	0,2090	0,8550	0,7910
D	80,12	0,7850	0,8300	0,1700	0,2280	0,8300	0,7720
L	79,48	0,8060	0,7770	0,2230	0,1880	0,7770	0,8120

Clasificador SVM no lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	90,5	0,9108	0,8980	0,1020	0,0880	0,8980	0,9120
I	84,8	0,8537	0,8400	0,1600	0,1440	0,8400	0,8560
D	80,4	0,8065	0,8000	0,2000	0,1920	0,8000	0,8080
L	80,2	0,8495	0,7340	0,2660	0,1300	0,7340	0,8700

## A.2. Bordes

### A.2.1. Bordes completos

#### A.2.1.1. Sin binarización

Clasificador Lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	91,45	0,9150	0,9140	0,0860	0,0850	0,9140	0,9150
I	85,95	0,8640	0,8530	0,1470	0,1340	0,8530	0,8660
D	74,69	0,7420	0,7570	0,2430	0,2630	0,7570	0,7370
L	88,97	0,8910	0,8880	0,1120	0,1090	0,8880	0,8910

Clasificador Bayesiano							
Vista	TA	P	R	FN	FP	TP	TN
F	92,17	0,8880	0,9650	0,0350	0,1210	0,9650	0,8790
I	81,81	0,7900	0,8660	0,1340	0,2300	0,8660	0,7700
D	83,07	0,8100	0,8650	0,1350	0,2030	0,8650	0,7970
L	79,35	0,7940	0,7930	0,2070	0,2060	0,7930	0,7940

Clasificador SVM lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	93,36	0,9590	0,9060	0,0940	0,0380	0,9060	0,9620
I	88,92	0,9180	0,8550	0,1450	0,0770	0,8550	0,9230
D	82,58	0,8530	0,7870	0,2130	0,1350	0,7870	0,8650
L	89,84	0,9400	0,8510	0,1490	0,0540	0,8510	0,9460

Clasificador SVM no lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	95,78	0,9680	0,9470	0,0530	0,0310	0,9470	0,9690
I	90,98	0,9090	0,9100	0,0890	0,0910	0,9100	0,9090
D	86,90	0,8600	0,8820	0,1180	0,1440	0,8820	0,8560
L	92,20	0,9220	0,9220	0,0780	0,0780	0,9220	0,9220

### A.2.1.2. Con binarización

Clasificador Lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	86,82	0,8614	0,8776	0,1224	0,1412	0,8776	0,8588
I	76,86	0,7801	0,7480	0,2520	0,2108	0,7480	0,7892
D	61,38	0,6114	0,6244	0,3756	0,3968	0,6244	0,6032
L	80,90	0,8074	0,8116	0,1884	0,1936	0,8116	0,8064

Clasificador Bayesiano							
Vista	TA	P	R	FN	FP	TP	TN
F	88,14	0,8233	0,9712	0,0288	0,2084	0,9712	0,7916
I	81,76	0,8044	0,8392	0,1608	0,2040	0,8392	0,7960
D	79,84	0,7771	0,8368	0,1632	0,2400	0,8368	0,7600
L	84,80	0,8505	0,8444	0,1556	0,1484	0,8444	0,8516

Clasificador SVM lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	90,78	0,9431	0,8680	0,1320	0,0524	0,8680	0,9476
I	85,22	0,9171	0,7744	0,2256	0,0700	0,7744	0,9300
D	77,60	0,8616	0,6576	0,3424	0,1056	0,6576	0,8944
L	86,92	0,8938	0,8380	0,1620	0,0996	0,8380	0,9004

Clasificador SVM no lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	93,94	0,9413	0,9372	0,0628	0,0584	0,9372	0,9416
I	86,60	0,8871	0,8388	0,1612	0,1068	0,8388	0,8932
D	81,86	0,8210	0,8148	0,1852	0,1776	0,8148	0,8224
L	87,34	0,8747	0,8716	0,1284	0,1248	0,8716	0,8752

### A.2.2. Bordes 32 bins

Clasificador Lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	92,44	0,9230	0,9260	0,0740	0,0770	0,9260	0,9230
I	87,66	0,8830	0,8680	0,1320	0,1150	0,8680	0,8850
D	82,26	0,8680	0,7600	0,2400	0,1150	0,7600	0,8850
L	81,50	0,8340	0,7870	0,2130	0,1570	0,7870	0,8430

Clasificador Bayesiano							
Vista	TA	P	R	FN	FP	TP	TN
F	93,60	0,9050	0,9750	0,0250	0,1030	0,9750	0,8970
I	84,72	0,8460	0,8490	0,1510	0,1550	0,8490	0,8450
D	84,32	0,8420	0,8450	0,1550	0,1590	0,8450	0,8410
L	83,90	0,8450	0,8300	0,1700	0,1520	0,8300	0,8480

Clasificador SVM lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	92,66	0,9630	0,8870	0,1130	0,0340	0,8870	0,9660
I	87,96	0,9060	0,8480	0,1520	0,0880	0,8480	0,9120
D	81,98	0,8750	0,7460	0,2540	0,1070	0,7460	0,8930
L	89,18	0,9540	0,8240	0,1760	0,0400	0,8240	0,9600

Clasificador SVM no lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	95,06	0,9672	0,9328	0,0672	0,0316	0,9328	0,9684
I	90,22	0,9027	0,9016	0,0984	0,0972	0,9016	0,9028
D	86,36	0,8526	0,8792	0,1208	0,1520	0,8792	0,8480
L	91,90	0,9185	0,9196	0,0804	0,0816	0,9196	0,9184

### A.2.3. Bordes 16 bins

Clasificador Lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	92,24	0,9331	0,9100	0,0900	0,0652	0,9100	0,9348
I	85,22	0,8670	0,8320	0,1680	0,1276	0,8320	0,8724
D	81,64	0,8522	0,7656	0,2344	0,1328	0,7656	0,8672
L	86,54	0,8480	0,8904	0,1096	0,1596	0,8904	0,8404

Clasificador Bayesiano							
Vista	TA	P	R	FN	FP	TP	TN
F	94,98	0,9478	0,9520	0,0480	0,0524	0,9520	0,9476
I	88,32	0,8784	0,8896	0,1104	0,1232	0,8896	0,8768
D	84,98	0,8339	0,8736	0,1264	0,1740	0,8736	0,8260
L	89,56	0,8660	0,9360	0,0640	0,1448	0,9360	0,8552

Clasificador SVM lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	92,38	0,9593	0,8852	0,1148	0,0376	0,8852	0,9624
I	86,18	0,9106	0,8024	0,1976	0,0788	0,8024	0,9212
D	81,92	0,8657	0,7556	0,2444	0,1172	0,7556	0,8828
L	87,08	0,9385	0,7936	0,2064	0,0520	0,7936	0,9480

Clasificador SVM no lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	95,10	0,9627	0,9384	0,0616	0,0364	0,9384	0,9636
I	90,22	0,8985	0,9068	0,0932	0,1024	0,9068	0,8976
D	85,48	0,8468	0,8664	0,1336	0,1568	0,8664	0,8432
L	92,04	0,9286	0,9108	0,0892	0,0700	0,9108	0,9300

#### A.2.4. Bordes 8 bins

Clasificador Lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	90,62	0,9096	0,9020	0,0980	0,0896	0,9020	0,9104
I	87,00	0,8633	0,8792	0,1208	0,1392	0,8792	0,8608
D	80,22	0,8429	0,7428	0,2572	0,1384	0,7428	0,8616
L	91,10	0,9248	0,8948	0,1052	0,0728	0,8948	0,9272

Clasificador Bayesiano							
Vista	TA	P	R	FN	FP	TP	TN
F	94,44	0,9267	0,9652	0,0348	0,0764	0,9652	0,9236
I	86,92	0,8257	0,9360	0,0640	0,1976	0,9360	0,8024
D	82,74	0,7889	0,8940	0,1060	0,2392	0,8940	0,7608
L	90,72	0,8833	0,9384	0,0616	0,1240	0,9384	0,8760

Clasificador SVM lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	90,54	0,9596	0,8464	0,1536	0,0356	0,8464	0,9644
I	84,28	0,9001	0,7712	0,2288	0,0856	0,7712	0,9144
D	79,72	0,8440	0,7292	0,2708	0,1348	0,7292	0,8652
L	83,46	0,9185	0,7344	0,2656	0,0652	0,7344	0,9348



Clasificador SVM no lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	94,40	0,9625	0,9240	0,0760	0,0360	0,9240	0,9640
I	89,92	0,9005	0,8976	0,1024	0,0992	0,8976	0,9008
D	85,22	0,8476	0,8588	0,1412	0,1544	0,8588	0,8456
L	91,56	0,9373	0,8908	0,1092	0,0596	0,8908	0,9404

### A.2.5. Bordes 4 bins

Clasificador Lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	92,96	0,9338	0,9248	0,0752	0,0656	0,9248	0,9344
I	82,32	0,8093	0,8456	0,1544	0,1992	0,8456	0,8008
D	78,28	0,8535	0,6828	0,3172	0,1172	0,6828	0,8828
L	82,58	0,8424	0,8016	0,1984	0,1500	0,8016	0,8500

Clasificador Bayesiano							
Vista	TA	P	R	FN	FP	TP	TN
F	92,44	0,8992	0,9560	0,0440	0,1072	0,9560	0,8928
I	86,16	0,8165	0,9328	0,0672	0,2096	0,9328	0,7904
D	82,94	0,7996	0,8792	0,1208	0,2204	0,8792	0,7796
L	89,48	0,8799	0,9144	0,0856	0,1248	0,9144	0,8752

Clasificador SVM lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	89,18	0,9596	0,8180	0,1820	0,0344	0,8180	0,9656
I	80,62	0,8664	0,7240	0,2760	0,1116	0,7240	0,8884
D	78,06	0,8437	0,6888	0,3112	0,1276	0,6888	0,8724
L	78,80	0,9091	0,6400	0,3600	0,0640	0,6400	0,9360

Clasificador SVM no lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	92,74	0,9458	0,9068	0,0932	0,0520	0,9068	0,9480
I	89,18	0,8993	0,8824	0,1176	0,0988	0,8824	0,9012
D	82,88	0,8191	0,8440	0,1560	0,1864	0,8440	0,8136
L	90,06	0,9157	0,8824	0,1176	0,0812	0,8824	0,9188

## A.2.6. Bordes máximos (2 valores)

### A.2.6.1. Sin binarización

Clasificador Lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	92,68	0,9152	0,9408	0,0592	0,0872	0,9408	0,9128
I	77,70	0,8315	0,6948	0,3052	0,1408	0,6948	0,8592
D	77,54	0,8529	0,6656	0,3344	0,1148	0,6656	0,8852
L	78,64	0,7930	0,7752	0,2248	0,2024	0,7752	0,7976

Clasificador Bayesiano							
Vista	TA	P	R	FN	FP	TP	TN
F	91,14	0,9606	0,8580	0,1420	0,0352	0,8580	0,9648
I	81,00	0,8628	0,7372	0,2628	0,1172	0,7372	0,8828
D	79,40	0,8575	0,7052	0,2948	0,1172	0,7052	0,8828
L	77,40	0,8506	0,6648	0,3352	0,1168	0,6648	0,8832

Clasificador SVM lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	92,62	0,9579	0,8916	0,1084	0,0392	0,8916	0,9608
I	82,46	0,8443	0,7960	0,2040	0,1468	0,7960	0,8532
D	83,68	0,8553	0,8108	0,1892	0,1372	0,8108	0,8628
L	78,80	0,8141	0,7464	0,2536	0,1704	0,7464	0,8296

Clasificador SVM no lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	92,62	0,9528	0,8968	0,1032	0,0444	0,8968	0,9556
I	82,62	0,8360	0,8116	0,1884	0,1592	0,8116	0,8408
D	83,82	0,8430	0,8312	0,1688	0,1548	0,8312	0,8452
L	78,52	0,7961	0,7668	0,2332	0,1964	0,7668	0,8036

### A.2.6.2. Con binarización

Clasificador Lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	94,14	0,9289	0,9560	0,0440	0,0732	0,9560	0,9268
I	82,48	0,8150	0,8404	0,1596	0,1908	0,8404	0,8092
D	84,12	0,8423	0,8396	0,1604	0,1572	0,8396	0,8428
L	68,48	0,6930	0,6636	0,3364	0,2940	0,6636	0,7060

Clasificador Bayesiano							
Vista	TA	P	R	FN	FP	TP	TN
F	93,56	0,9467	0,9232	0,0768	0,0520	0,9232	0,9480
I	86,02	0,8595	0,8612	0,1388	0,1408	0,8612	0,8592
D	84,22	0,8418	0,8428	0,1572	0,1584	0,8428	0,8416
L	80,12	0,8046	0,7956	0,2044	0,1932	0,7956	0,8068

Clasificador SVM lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	94,36	0,9376	0,9504	0,0496	0,0632	0,9504	0,9368
I	84,66	0,8288	0,8736	0,1264	0,1804	0,8736	0,8196
D	85,38	0,8446	0,8672	0,1328	0,1596	0,8672	0,8404
L	78,98	0,7788	0,8096	0,1904	0,2300	0,8096	0,7700

Clasificador SVM no lineal							
Vista	TA	P	R	FN	FP	TP	TN
F	94,02	0,9425	0,9376	0,0624	0,0572	0,9376	0,9428
I	84,20	0,8226	0,8720	0,1280	0,1880	0,8720	0,8120
D	85,08	0,8466	0,8568	0,1432	0,1552	0,8568	0,8448
L	79,24	0,7713	0,8312	0,1688	0,2464	0,8312	0,7536

## A.3. PCA

### A.3.1. Imagen en nivel de gris

Retención del 90 % de la varianza								
Vistas	TA	CP	P	R	FN	FP	TP	TN
F	93,37	80,30	0,9560	0,9090	0,0910	0,0420	0,9090	0,9580
I	92,30	79,80	0,9240	0,9220	0,0780	0,0760	0,9220	0,9240
D	89,02	65,20	0,8980	0,8800	0,1200	0,1000	0,8800	0,9000
L	90,88	47,30	0,9160	0,9010	0,0990	0,0830	0,9010	0,9170

Retención del 95 % de la varianza								
Vistas	TA	CP	P	R	FN	FP	TP	TN
F	92,48	165,00	0,9566	0,8900	0,1100	0,0404	0,8900	0,9596
I	90,12	163,20	0,9183	0,8808	0,1192	0,0784	0,8808	0,9216
D	87,08	140,20	0,8815	0,8568	0,1432	0,1152	0,8568	0,8848
L	90,40	98,40	0,9163	0,8892	0,1108	0,0812	0,8892	0,9188

Cambio de la dimensionalidad del subespacio principal												
Vista	Número de Componentes											
	10	20	30	40	50	60	70	80	90	100	110	120
F	93,7	96,1	94,3	95,0	95,1	93,7	92,1	92,3	92,8	90,8	90,2	91,6
I	88,0	90,3	91,9	93,0	92,2	92,7	92,1	91,0	91,3	92,1	90,5	88,9
D	86,6	89,1	89,0	88,7	89,6	90,0	89,9	90,1	88,9	88,9	88,1	86,2
L	88,1	87,9	90,9	91,3	90,5	90,6	90,6	91,0	90,2	89,4	88,9	89,8

### A.3.2. Imagen en el espacio de color RGB

Retención del 90 % de la varianza								
Vistas	TA	CP	P	R	FN	FP	TP	TN
F	95,76	90,80	0,9670	0,9480	0,0520	0,0320	0,9480	0,9680
I	92,94	93,20	0,9320	0,9260	0,0740	0,0670	0,9260	0,9330
D	91,80	78,20	0,9180	0,9180	0,0820	0,0820	0,9180	0,9180
L	91,50	57,00	0,9210	0,9080	0,0920	0,0780	0,9080	0,9220

Retención del 95 % de la varianza								
Vistas	TA	CP	P	R	FN	FP	TP	TN
F	95,10	185,00	0,9688	0,9320	0,0680	0,0300	0,9320	0,9700
I	93,90	189,00	0,9526	0,9240	0,0760	0,0460	0,9240	0,9540
D	91,00	164,00	0,9343	0,8820	0,1180	0,0620	0,8820	0,9380
L	92,00	117,00	0,9393	0,8980	0,1020	0,0580	0,8980	0,9420

Cambio de la dimensionalidad del subespacio principal												
Vista	Número de Componentes											
	10	20	30	40	50	60	70	80	90	100	110	120
F	94,2	97,1	95,8	95,6	97,0	94,8	95,6	95,1	95,8	95,5	95,1	96,5
I	88,7	90,7	92,9	93,8	93,4	93,7	93,2	92,8	93,2	93,6	93,3	91,9
D	87,5	89,9	91,5	91,8	92,2	91,7	90,4	93,1	89,7	91,5	90,3	89,9
L	89,4	92,9	90,6	90,7	91,6	90,9	90,6	92,4	93,5	92,0	88,0	91,0

## A.4. HOG

### A.4.1. SHOG

		$\eta = 2$							
		Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	98,60	0,9806	0,9916	0,0084	0,0196	0,9916	0,9804	
	I	93,04	0,9138	0,9504	0,0496	0,0896	0,9504	0,9104	
	D	92,78	0,9125	0,9464	0,0536	0,0908	0,9464	0,9092	
	L	91,96	0,9019	0,9416	0,0584	0,1024	0,9416	0,8976	
$\beta = 12$	F	96,28	0,9688	0,9564	0,0436	0,0308	0,9564	0,9692	
	I	91,52	0,8968	0,9384	0,0616	0,1080	0,9384	0,8920	
	D	91,02	0,9081	0,9128	0,0872	0,0924	0,9128	0,9076	
	L	87,52	0,8788	0,8704	0,1296	0,1200	0,8704	0,8800	
$\beta = 16$	F	98,20	0,9767	0,9876	0,0124	0,0236	0,9876	0,9764	
	I	94,30	0,9288	0,9596	0,0404	0,0736	0,9596	0,9264	
	D	93,66	0,9184	0,9584	0,0416	0,0852	0,9584	0,9148	
	L	91,52	0,9023	0,9312	0,0688	0,1008	0,9312	0,8992	
$\beta = 32$	F	92,62	0,9587	0,8908	0,1092	0,0384	0,8908	0,9616	
	I	88,76	0,9024	0,8692	0,1308	0,0940	0,8692	0,9060	
	D	88,32	0,8962	0,8668	0,1332	0,1004	0,8668	0,8996	
	L	84,32	0,8834	0,7908	0,2092	0,1044	0,7908	0,8956	

		$\eta = 4$							
		Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,48	0,9920	0,9976	0,0024	0,0080	0,9976	0,9920	
	I	97,64	0,9737	0,9792	0,0208	0,0264	0,9792	0,9736	
	D	96,00	0,9503	0,9708	0,0292	0,0508	0,9708	0,9492	
	L	97,76	0,9780	0,9772	0,0228	0,0220	0,9772	0,9780	
$\beta = 12$	F	98,52	0,9776	0,9932	0,0068	0,0228	0,9932	0,9772	
	I	96,96	0,9648	0,9748	0,0252	0,0356	0,9748	0,9644	
	D	95,46	0,9445	0,9660	0,0340	0,0568	0,9660	0,9432	
	L	96,54	0,9759	0,9544	0,0456	0,0236	0,9544	0,9764	
$\beta = 16$	F	99,18	0,9908	0,9928	0,0072	0,0092	0,9928	0,9908	
	I	97,04	0,9704	0,9704	0,0296	0,0296	0,9704	0,9704	
	D	96,22	0,9537	0,9716	0,0284	0,0472	0,9716	0,9528	
	L	97,04	0,9742	0,9664	0,0336	0,0256	0,9664	0,9744	
$\beta = 32$	F	98,28	0,9759	0,9900	0,0100	0,0244	0,9900	0,9756	
	I	95,70	0,9475	0,9676	0,0324	0,0536	0,9676	0,9464	
	D	95,10	0,9399	0,9636	0,0364	0,0616	0,9636	0,9384	
	L	95,56	0,9567	0,9544	0,0456	0,0432	0,9544	0,9568	

		$\eta = 8$							
		Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	98,68	0,9895	0,9840	0,0160	0,0104	0,9840	0,9896	
	I	97,34	0,9793	0,9672	0,0328	0,0204	0,9672	0,9796	
	D	94,70	0,9461	0,9480	0,0520	0,0540	0,9480	0,9460	
	L	97,28	0,9770	0,9684	0,0316	0,0228	0,9684	0,9772	
$\beta = 12$	F	98,54	0,9879	0,9828	0,0172	0,0120	0,9828	0,9880	
	I	96,56	0,9649	0,9664	0,0336	0,0352	0,9664	0,9648	
	D	95,12	0,9548	0,9472	0,0528	0,0448	0,9472	0,9552	
	L	96,70	0,9740	0,9596	0,0404	0,0256	0,9596	0,9744	
$\beta = 16$	F	99,16	0,9916	0,9916	0,0084	0,0084	0,9916	0,9916	
	I	96,96	0,9757	0,9632	0,0368	0,0240	0,9632	0,9760	
	D	94,54	0,9492	0,9412	0,0588	0,0504	0,9412	0,9496	
	L	96,44	0,9633	0,9656	0,0344	0,0368	0,9656	0,9632	
$\beta = 32$	F	98,40	0,9840	0,9840	0,0160	0,0160	0,9840	0,9840	
	I	96,04	0,9641	0,9564	0,0436	0,0356	0,9564	0,9644	
	D	94,46	0,9398	0,9500	0,0500	0,0608	0,9500	0,9392	
	L	96,18	0,9627	0,9608	0,0392	0,0372	0,9608	0,9628	

		$\eta = 16$							
		Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	98,82	0,9872	0,9892	0,0108	0,0128	0,9892	0,9872	
	I	96,10	0,9888	0,9916	0,0084	0,0112	0,9916	0,9888	
	D	93,64	0,9378	0,9348	0,0652	0,0620	0,9348	0,9380	
	L	97,38	0,9774	0,9700	0,0300	0,0224	0,9700	0,9776	
$\beta = 12$	F	98,98	0,9877	0,9920	0,0080	0,0124	0,9920	0,9876	
	I	96,02	0,9652	0,9548	0,0452	0,0344	0,9548	0,9656	
	D	94,42	0,9454	0,9428	0,0572	0,0544	0,9428	0,9456	
	L	97,16	0,9762	0,9668	0,0332	0,0236	0,9668	0,9764	
$\beta = 16$	F	98,48	0,9860	0,9836	0,0164	0,0140	0,9836	0,9860	
	I	95,98	0,9698	0,9492	0,0508	0,0296	0,9492	0,9704	
	D	93,56	0,9356	0,9356	0,0644	0,0644	0,9356	0,9356	
	L	97,40	0,9798	0,9680	0,0320	0,0200	0,9680	0,9800	
$\beta = 32$	F	99,02	0,9888	0,9916	0,0084	0,0112	0,9916	0,9888	
	I	96,02	0,9645	0,9556	0,0444	0,0352	0,9556	0,9648	
	D	95,10	0,9490	0,9532	0,0468	0,0512	0,9532	0,9488	
	L	97,34	0,9770	0,9696	0,0304	0,0228	0,9696	0,9772	

## A.4.2. OHOG

### A.4.2.1. SHOG + operador de Sobel

		$\eta = 4$							
		Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,34	0,9916	0,9952	0,0048	0,0084	0,9952	0,9916	
	I	97,18	0,9746	0,9688	0,0312	0,0252	0,9688	0,9748	
	D	96,18	0,9576	0,9664	0,0336	0,0428	0,9664	0,9572	
	L	97,62	0,9810	0,9712	0,0288	0,0188	0,9712	0,9812	
$\beta = 12$	F	98,90	0,9834	0,9948	0,0052	0,0168	0,9948	0,9832	
	I	96,22	0,9583	0,9664	0,0336	0,0420	0,9664	0,9580	
	D	95,50	0,9400	0,9720	0,0280	0,0620	0,9720	0,9380	
	L	96,28	0,9643	0,9612	0,0388	0,0356	0,9612	0,9644	
$\beta = 16$	F	99,44	0,9924	0,9964	0,0036	0,0076	0,9964	0,9924	
	I	96,42	0,9693	0,9588	0,0412	0,0304	0,9588	0,9696	
	D	95,60	0,9509	0,9616	0,0384	0,0496	0,9616	0,9504	
	L	96,90	0,9669	0,9712	0,0288	0,0332	0,9712	0,9668	
$\beta = 32$	F	98,70	0,9837	0,9904	0,0096	0,0164	0,9904	0,9836	
	I	95,88	0,9534	0,9648	0,0352	0,0472	0,9648	0,9528	
	D	94,94	0,9343	0,9668	0,0332	0,0680	0,9668	0,9320	
	L	95,98	0,9560	0,9640	0,0360	0,0444	0,9640	0,9556	

		$\eta = 8$							
		Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,14	0,9896	0,9932	0,0068	0,0104	0,9932	0,9896	
	I	96,88	0,9749	0,9624	0,0376	0,0248	0,9624	0,9752	
	D	94,78	0,9598	0,9348	0,0652	0,0392	0,9348	0,9608	
	L	96,80	0,9718	0,9640	0,0360	0,0280	0,9640	0,9720	
$\beta = 12$	F	98,64	0,9876	0,9852	0,0148	0,0124	0,9852	0,9876	
	I	96,36	0,9673	0,9596	0,0404	0,0324	0,9596	0,9676	
	D	94,92	0,9550	0,9428	0,0572	0,0444	0,9428	0,9556	
	L	96,84	0,9699	0,9668	0,0332	0,0300	0,9668	0,9700	
$\beta = 16$	F	98,50	0,9887	0,9812	0,0188	0,0112	0,9812	0,9888	
	I	96,02	0,9600	0,9604	0,0396	0,0400	0,9604	0,9600	
	D	93,94	0,9413	0,9372	0,0628	0,0584	0,9372	0,9416	
	L	96,28	0,9677	0,9576	0,0424	0,0320	0,9576	0,9680	
$\beta = 32$	F	98,20	0,9793	0,9848	0,0152	0,0208	0,9848	0,9792	
	I	96,14	0,9612	0,9616	0,0384	0,0388	0,9616	0,9612	
	D	94,98	0,9450	0,9552	0,0448	0,0556	0,9552	0,9444	
	L	96,36	0,9666	0,9604	0,0396	0,0332	0,9604	0,9668	

### A.4.2.2. SHOG + Ampliación del rango de ángulos a $[-\pi, \pi]$

$\eta = 4$								
	Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,12	0,9916	0,9908	0,0092	0,0084	0,9908	0,9916
	I	98,40	0,9798	0,9884	0,0116	0,0204	0,9884	0,9796
	D	98,14	0,9740	0,9892	0,0108	0,0264	0,9892	0,9736
	L	97,56	0,9730	0,9784	0,0216	0,0272	0,9784	0,9728
$\beta = 12$	F	99,34	0,9936	0,9932	0,0068	0,0064	0,9932	0,9936
	I	99,00	0,9888	0,9912	0,0088	0,0112	0,9912	0,9888
	D	98,04	0,9736	0,9876	0,0124	0,0268	0,9876	0,9732
	L	98,46	0,9852	0,9840	0,0160	0,0148	0,9840	0,9852
$\beta = 16$	F	99,70	0,9944	0,9996	0,0004	0,0056	0,9996	0,9944
	I	98,92	0,9900	0,9884	0,0116	0,0100	0,9884	0,9900
	D	98,54	0,9810	0,9900	0,0100	0,0192	0,9900	0,9808
	L	98,46	0,9883	0,9808	0,0192	0,0116	0,9808	0,9884
$\beta = 32$	F	99,52	0,9944	0,9960	0,0040	0,0056	0,9960	0,9944
	I	98,80	0,9884	0,9876	0,0124	0,0116	0,9876	0,9884
	D	98,48	0,9821	0,9876	0,0124	0,0180	0,9876	0,9820
	L	98,38	0,9895	0,9780	0,0220	0,0104	0,9780	0,9896

$\eta = 8$								
	Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,26	0,9936	0,9916	0,0084	0,0064	0,9916	0,9936
	I	98,58	0,9872	0,9844	0,0156	0,0128	0,9844	0,9872
	D	98,22	0,9801	0,9844	0,0156	0,0200	0,9844	0,9800
	L	97,84	0,9803	0,9764	0,0236	0,0196	0,9764	0,9804
$\beta = 12$	F	99,56	0,9952	0,9960	0,0040	0,0048	0,9960	0,9952
	I	98,82	0,9892	0,9872	0,0128	0,0108	0,9872	0,9892
	D	98,34	0,9821	0,9848	0,0152	0,0180	0,9848	0,9820
	L	98,54	0,9895	0,9812	0,0188	0,0104	0,9812	0,9896
$\beta = 16$	F	99,54	0,9924	0,9984	0,0016	0,0076	0,9984	0,9924
	I	98,90	0,9927	0,9852	0,0148	0,0072	0,9852	0,9928
	D	98,40	0,9828	0,9852	0,0148	0,0172	0,9852	0,9828
	L	98,44	0,9871	0,9816	0,0184	0,0128	0,9816	0,9872
$\beta = 32$	F	99,70	0,9948	0,9992	0,0008	0,0052	0,9992	0,9948
	I	98,96	0,9865	0,9928	0,0072	0,0136	0,9928	0,9864
	D	98,62	0,9860	0,9864	0,0136	0,0140	0,9864	0,9860
	L	98,56	0,9903	0,9808	0,0192	0,0096	0,9808	0,9904



### A.4.2.3. SHOG + $[-\pi, \pi]$ + No interpolación

$\eta = 4$								
	Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,06	0,9900	0,9912	0,0088	0,0100	0,9912	0,9900
	I	98,36	0,9813	0,9860	0,0140	0,0188	0,9860	0,9812
	D	97,62	0,9704	0,9824	0,0176	0,0300	0,9824	0,9700
	L	97,56	0,9756	0,9756	0,0244	0,0244	0,9756	0,9756
$\beta = 12$	F	99,26	0,9901	0,9952	0,0048	0,0100	0,9952	0,9900
	I	98,64	0,9852	0,9876	0,0124	0,0148	0,9876	0,9852
	D	98,10	0,9755	0,9868	0,0132	0,0248	0,9868	0,9752
	L	98,56	0,9848	0,9864	0,0136	0,0152	0,9864	0,9848
$\beta = 16$	F	99,38	0,9928	0,9948	0,0052	0,0072	0,9948	0,9928
	I	98,60	0,9868	0,9852	0,0148	0,0132	0,9852	0,9868
	D	98,30	0,9771	0,9892	0,0108	0,0232	0,9892	0,9768
	L	98,42	0,9867	0,9816	0,0184	0,0132	0,9816	0,9868
$\beta = 32$	F	99,30	0,9908	0,9952	0,0048	0,0092	0,9952	0,9908
	I	99,04	0,9892	0,9916	0,0084	0,0108	0,9916	0,9892
	D	98,48	0,9791	0,9908	0,0092	0,0212	0,9908	0,9788
	L	98,72	0,9864	0,9880	0,0120	0,0136	0,9880	0,9864

$\eta = 8$								
	Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,26	0,9940	0,9912	0,0088	0,0060	0,9912	0,9940
	I	98,82	0,9888	0,9876	0,0124	0,0112	0,9876	0,9888
	D	97,96	0,9785	0,9808	0,0192	0,0216	0,9808	0,9784
	L	97,64	0,9753	0,9776	0,0224	0,0248	0,9776	0,9752
$\beta = 12$	F	99,42	0,9944	0,9940	0,0060	0,0056	0,9940	0,9944
	I	98,98	0,9896	0,9900	0,0100	0,0104	0,9900	0,9896
	D	98,06	0,9785	0,9828	0,0172	0,0216	0,9828	0,9784
	L	98,08	0,9851	0,9764	0,0236	0,0148	0,9764	0,9852
$\beta = 16$	F	99,64	0,9960	0,9968	0,0032	0,0040	0,9968	0,9960
	I	98,76	0,9931	0,9820	0,0180	0,0068	0,9820	0,9932
	D	98,28	0,9813	0,9844	0,0156	0,0188	0,9844	0,9812
	L	98,52	0,9829	0,9876	0,0124	0,0172	0,9876	0,9828
$\beta = 32$	F	99,56	0,9936	0,9976	0,0024	0,0064	0,9976	0,9936
	I	98,96	0,9932	0,9860	0,0140	0,0068	0,9860	0,9932
	D	98,58	0,9833	0,9884	0,0116	0,0168	0,9884	0,9832
	L	98,60	0,9848	0,9872	0,0128	0,0152	0,9872	0,9848

**A.4.2.4. SHOG +  $[-\pi, \pi]$  + tamaño de bloque 2x1**

$\eta = 4$								
	Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,36	0,9928	0,9944	0,0056	0,0072	0,9944	0,9928
	I	98,62	0,9837	0,9888	0,0112	0,0164	0,9888	0,9836
	D	97,96	0,9702	0,9896	0,0104	0,0304	0,9896	0,9696
	L	98,04	0,9819	0,9788	0,0212	0,0180	0,9788	0,9820
$\beta = 12$	F	99,34	0,9901	0,9968	0,0032	0,0100	0,9968	0,9900
	I	99,04	0,9881	0,9928	0,0072	0,0120	0,9928	0,9880
	D	98,20	0,9763	0,9880	0,0120	0,0240	0,9880	0,9760
	L	98,44	0,9903	0,9784	0,0216	0,0096	0,9784	0,9904
$\beta = 16$	F	99,56	0,9952	0,9960	0,0040	0,0048	0,9960	0,9952
	I	98,98	0,9877	0,9920	0,0080	0,0124	0,9920	0,9876
	D	98,56	0,9806	0,9908	0,0092	0,0196	0,9908	0,9804
	L	98,78	0,9911	0,9844	0,0156	0,0088	0,9844	0,9912
$\beta = 32$	F	99,72	0,9944	1,0000	0,0000	0,0056	1,0000	0,9944
	I	98,96	0,9896	0,9896	0,0104	0,0104	0,9896	0,9896
	D	98,52	0,9821	0,9884	0,0116	0,0180	0,9884	0,9820
	L	98,68	0,9888	0,9848	0,0152	0,0112	0,9848	0,9888

$\eta = 8$								
	Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,38	0,9960	0,9916	0,0084	0,0040	0,9916	0,9960
	I	98,82	0,9900	0,9864	0,0136	0,0100	0,9864	0,9900
	D	98,14	0,9808	0,9820	0,0180	0,0192	0,9820	0,9808
	L	97,76	0,9776	0,9776	0,0224	0,0224	0,9776	0,9776
$\beta = 12$	F	99,54	0,9956	0,9952	0,0048	0,0044	0,9952	0,9956
	I	98,88	0,9888	0,9888	0,0112	0,0112	0,9888	0,9888
	D	98,40	0,9852	0,9828	0,0172	0,0148	0,9828	0,9852
	L	98,22	0,9851	0,9792	0,0208	0,0148	0,9792	0,9852
$\beta = 16$	F	99,60	0,9948	0,9972	0,0028	0,0052	0,9972	0,9948
	I	98,74	0,9927	0,9820	0,0180	0,0072	0,9820	0,9928
	D	98,22	0,9828	0,9816	0,0184	0,0172	0,9816	0,9828
	L	98,70	0,9876	0,9864	0,0136	0,0124	0,9864	0,9876
$\beta = 32$	F	99,62	0,9944	0,9980	0,0020	0,0056	0,9980	0,9944
	I	98,90	0,9912	0,9868	0,0132	0,0088	0,9868	0,9912
	D	98,48	0,9848	0,9848	0,0152	0,0152	0,9848	0,9848
	L	98,46	0,9852	0,9840	0,0160	0,0148	0,9840	0,9852

**A.4.2.5. SHOG +  $[-\pi, \pi]$  + tamaño de bloque 1x1**

$\eta = 4$								
	Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,16	0,9936	0,9896	0,0104	0,0064	0,9896	0,9936
	I	98,76	0,9845	0,9908	0,0092	0,0156	0,9908	0,9844
	D	98,32	0,9794	0,9872	0,0128	0,0208	0,9872	0,9792
	L	97,90	0,9796	0,9784	0,0216	0,0204	0,9784	0,9796
$\beta = 12$	F	99,60	0,9952	0,9968	0,0032	0,0048	0,9968	0,9952
	I	99,06	0,9888	0,9924	0,0076	0,0112	0,9924	0,9888
	D	98,56	0,9821	0,9892	0,0108	0,0180	0,9892	0,9820
	L	98,12	0,9808	0,9816	0,0184	0,0192	0,9816	0,9808
$\beta = 16$	F	99,66	0,9960	0,9972	0,0028	0,0040	0,9972	0,9960
	I	99,20	0,9932	0,9908	0,0092	0,0068	0,9908	0,9932
	D	98,30	0,9801	0,9860	0,0140	0,0200	0,9860	0,9800
	L	98,18	0,9832	0,9804	0,0196	0,0168	0,9804	0,9832
$\beta = 32$	F	99,76	0,9952	1,0000	0,0000	0,0048	1,0000	0,9952
	I	99,10	0,9877	0,9944	0,0056	0,0124	0,9944	0,9876
	D	98,56	0,9814	0,9900	0,0100	0,0188	0,9900	0,9812
	L	97,88	0,9777	0,9800	0,0200	0,0224	0,9800	0,9776

$\eta = 8$								
	Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,42	0,9944	0,9940	0,0060	0,0056	0,9940	0,9944
	I	98,32	0,9851	0,9812	0,0188	0,0148	0,9812	0,9852
	D	98,04	0,9816	0,9792	0,0208	0,0184	0,9792	0,9816
	L	97,72	0,9810	0,9732	0,0268	0,0188	0,9732	0,9812
$\beta = 12$	F	99,50	0,9948	0,9952	0,0048	0,0052	0,9952	0,9948
	I	98,92	0,9904	0,9880	0,0120	0,0096	0,9880	0,9904
	D	98,36	0,9836	0,9836	0,0164	0,0164	0,9836	0,9836
	L	97,82	0,9819	0,9744	0,0256	0,0180	0,9744	0,9820
$\beta = 16$	F	99,46	0,9952	0,9940	0,0060	0,0048	0,9940	0,9952
	I	98,72	0,9915	0,9828	0,0172	0,0084	0,9828	0,9916
	D	98,16	0,9824	0,9808	0,0192	0,0176	0,9808	0,9824
	L	98,36	0,9879	0,9792	0,0208	0,0120	0,9792	0,9880
$\beta = 32$	F	99,72	0,9956	0,9988	0,0012	0,0044	0,9988	0,9956
	I	98,98	0,9920	0,9876	0,0124	0,0080	0,9876	0,9920
	D	98,50	0,9844	0,9856	0,0144	0,0156	0,9856	0,9844
	L	98,64	0,9891	0,9836	0,0164	0,0108	0,9836	0,9892

**A.4.2.6. SHOG +  $[-\pi, \pi]$  + bloque 1x1 + L1-sqrt**

$\eta = 4$								
	Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,40	0,9924	0,9956	0,0044	0,0076	0,9956	0,9924
	I	98,70	0,9868	0,9872	0,0128	0,0132	0,9872	0,9868
	D	97,68	0,9757	0,9780	0,0220	0,0244	0,9780	0,9756
	L	97,00	0,9615	0,9792	0,0208	0,0392	0,9792	0,9608
$\beta = 12$	F	99,38	0,9909	0,9968	0,0032	0,0092	0,9968	0,9908
	I	98,82	0,9845	0,9920	0,0080	0,0156	0,9920	0,9844
	D	97,84	0,9773	0,9796	0,0204	0,0228	0,9796	0,9772
	L	97,96	0,9842	0,9748	0,0252	0,0156	0,9748	0,9844
$\beta = 16$	F	99,62	0,9964	0,9960	0,0040	0,0036	0,9960	0,9964
	I	98,74	0,9857	0,9892	0,0108	0,0144	0,9892	0,9856
	D	98,14	0,9789	0,9840	0,0160	0,0212	0,9840	0,9788
	L	98,12	0,9804	0,9820	0,0180	0,0196	0,9820	0,9804
$\beta = 32$	F	99,70	0,9952	0,9988	0,0012	0,0048	0,9988	0,9952
	I	98,64	0,9860	0,9868	0,0132	0,0140	0,9868	0,9860
	D	98,24	0,9767	0,9884	0,0116	0,0236	0,9884	0,9764
	L	97,66	0,9749	0,9784	0,0216	0,0252	0,9784	0,9748

$\eta = 8$								
	Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,24	0,9940	0,9908	0,0092	0,0060	0,9908	0,9940
	I	98,28	0,9840	0,9816	0,0184	0,0160	0,9816	0,9840
	D	97,90	0,9761	0,9820	0,0180	0,0240	0,9820	0,9760
	L	97,58	0,9737	0,9780	0,0220	0,0264	0,9780	0,9736
$\beta = 12$	F	99,56	0,9956	0,9956	0,0044	0,0044	0,9956	0,9956
	I	98,82	0,9861	0,9904	0,0096	0,0140	0,9904	0,9860
	D	96,00	0,9675	0,9520	0,0480	0,0320	0,9520	0,9680
	L	98,30	0,9828	0,9832	0,0168	0,0172	0,9832	0,9828
$\beta = 16$	F	99,56	0,9948	0,9964	0,0036	0,0052	0,9964	0,9948
	I	98,82	0,9919	0,9844	0,0156	0,0080	0,9844	0,9920
	D	97,90	0,9807	0,9772	0,0228	0,0192	0,9772	0,9808
	L	98,32	0,9817	0,9848	0,0152	0,0184	0,9848	0,9816
$\beta = 32$	F	99,50	0,9920	0,9980	0,0020	0,0080	0,9980	0,9920
	I	98,70	0,9888	0,9852	0,0148	0,0112	0,9852	0,9888
	D	98,24	0,9786	0,9864	0,0136	0,0216	0,9864	0,9784
	L	98,42	0,9817	0,9868	0,0132	0,0184	0,9868	0,9816

**A.4.2.7. SHOG +  $[-\pi, \pi]$  + bloque 1x1 + clasificador SVM (núcleo polinomial de orden 2)**

$\eta = 2$								
	Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,04	0,9838	0,9972	0,0028	0,0164	0,9972	0,9836
	I	97,94	0,9762	0,9828	0,0172	0,0240	0,9828	0,9760
	D	98,12	0,9714	0,9916	0,0084	0,0292	0,9916	0,9708
	L	96,78	0,9570	0,9796	0,0204	0,0440	0,9796	0,9560
$\beta = 12$	F	99,14	0,9842	0,9988	0,0012	0,0160	0,9988	0,9840
	I	98,54	0,9840	0,9868	0,0132	0,0160	0,9868	0,9840
	D	98,44	0,9787	0,9904	0,0096	0,0216	0,9904	0,9784
	L	97,60	0,9692	0,9832	0,0168	0,0312	0,9832	0,9688
$\beta = 16$	F	99,16	0,9862	0,9972	0,0028	0,0140	0,9972	0,9860
	I	98,50	0,9813	0,9888	0,0112	0,0188	0,9888	0,9812
	D	98,70	0,9799	0,9944	0,0056	0,0204	0,9944	0,9796
	L	97,82	0,9712	0,9856	0,0144	0,0292	0,9856	0,9708
$\beta = 32$	F	99,50	0,9909	0,9992	0,0008	0,0092	0,9992	0,9908
	I	98,56	0,9829	0,9884	0,0116	0,0172	0,9884	0,9828
	D	98,78	0,9803	0,9956	0,0044	0,0200	0,9956	0,9800
	L	97,90	0,9746	0,9836	0,0164	0,0256	0,9836	0,9744

$\eta = 4$								
	Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,56	0,9928	0,9984	0,0016	0,0072	0,9984	0,9928
	I	99,14	0,9920	0,9908	0,0092	0,0080	0,9908	0,9920
	D	98,80	0,9857	0,9904	0,0096	0,0144	0,9904	0,9856
	L	98,40	0,9821	0,9860	0,0140	0,0180	0,9860	0,9820
$\beta = 12$	F	99,78	0,9976	0,9980	0,0020	0,0024	0,9980	0,9976
	I	99,54	0,9952	0,9956	0,0044	0,0048	0,9956	0,9952
	D	98,70	0,9826	0,9916	0,0084	0,0176	0,9916	0,9824
	L	98,98	0,9880	0,9916	0,0084	0,0120	0,9916	0,9880
$\beta = 16$	F	99,80	0,9964	0,9996	0,0004	0,0036	0,9996	0,9964
	I	99,54	0,9944	0,9964	0,0036	0,0056	0,9964	0,9944
	D	98,90	0,9845	0,9936	0,0064	0,0156	0,9936	0,9844
	L	98,64	0,9810	0,9920	0,0080	0,0192	0,9920	0,9808
$\beta = 32$	F	99,80	0,9960	1,0000	0,0000	0,0040	1,0000	0,9960
	I	99,40	0,9916	0,9964	0,0036	0,0084	0,9964	0,9916
	D	98,98	0,9850	0,9948	0,0052	0,0152	0,9948	0,9848
	L	98,88	0,9869	0,9908	0,0092	0,0132	0,9908	0,9868

$\eta = 8$								
	Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,56	0,9960	0,9952	0,0048	0,0040	0,9952	0,9960
	I	99,18	0,9924	0,9912	0,0088	0,0076	0,9912	0,9924
	D	98,86	0,9861	0,9912	0,0088	0,0140	0,9912	0,9860
	L	98,08	0,9774	0,9844	0,0156	0,0228	0,9844	0,9772
$\beta = 12$	F	99,70	0,9960	0,9980	0,0020	0,0040	0,9980	0,9960
	I	99,62	0,9960	0,9964	0,0036	0,0040	0,9964	0,9960
	D	98,98	0,9861	0,9936	0,0064	0,0140	0,9936	0,9860
	L	98,16	0,9778	0,9856	0,0144	0,0224	0,9856	0,9776
$\beta = 16$	F	99,54	0,9924	0,9984	0,0016	0,0076	0,9984	0,9924
	I	99,52	0,9956	0,9948	0,0052	0,0044	0,9948	0,9956
	D	99,02	0,9892	0,9912	0,0088	0,0108	0,9912	0,9892
	L	98,76	0,9818	0,9936	0,0064	0,0184	0,9936	0,9816
$\beta = 32$	F	99,46	0,9901	0,9992	0,0008	0,0100	0,9992	0,9900
	I	99,50	0,9944	0,9956	0,0044	0,0056	0,9956	0,9944
	D	99,20	0,9920	0,9920	0,0080	0,0080	0,9920	0,9920
	L	98,70	0,9788	0,9956	0,0044	0,0216	0,9956	0,9784

$\eta = 16$								
	Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	98,52	0,9891	0,9812	0,0188	0,0108	0,9812	0,9892
	I	98,74	0,9971	0,9776	0,0224	0,0028	0,9776	0,9972
	D	98,98	0,9908	0,9888	0,0112	0,0092	0,9888	0,9908
	L	98,04	0,9755	0,9856	0,0144	0,0248	0,9856	0,9752
$\beta = 12$	F	98,62	0,9852	0,9872	0,0128	0,0148	0,9872	0,9852
	I	98,76	0,9980	0,9772	0,0228	0,0020	0,9772	0,9980
	D	98,70	0,9907	0,9832	0,0168	0,0092	0,9832	0,9908
	L	98,12	0,9785	0,9840	0,0160	0,0216	0,9840	0,9784
$\beta = 16$	F	98,88	0,9872	0,9904	0,0096	0,0128	0,9904	0,9872
	I	98,24	0,9988	0,9660	0,0340	0,0012	0,9660	0,9988
	D	99,02	0,9920	0,9884	0,0116	0,0080	0,9884	0,9920
	L	98,28	0,9797	0,9860	0,0140	0,0204	0,9860	0,9796
$\beta = 32$	F	98,82	0,9888	0,9876	0,0124	0,0112	0,9876	0,9888
	I	98,34	0,9975	0,9692	0,0308	0,0024	0,9692	0,9976
	D	98,58	0,9907	0,9808	0,0192	0,0092	0,9808	0,9908
	L	98,30	0,9790	0,9872	0,0128	0,0212	0,9872	0,9788

### A.4.3. PCA-OHOG

	I95		I90		I75		I50		I25		I10	
Vista	TA	NC	TA	NC	TA	NC	TA	NC	TA	NC	TA	NC
F	99,78	102	99,72	72	99,60	31	99,22	9	98,16	2	97,18	1
I	99,46	108	99,40	78	99,32	36	98,46	11	97,86	3	97,00	1
D	98,84	109	98,68	79	98,54	37	98,46	11	97,42	3	94,56	1
L	99,00	112	98,78	83	98,34	41	97,72	13	96,60	4	94,26	1

Vista	Dimensionalidad del subespacio principal									
	10	20	30	40	50	60	70	80	90	100
F	99,18	99,32	99,34	99,24	99,50	99,72	99,64	99,78	99,68	99,92
I	98,58	98,50	98,54	98,48	98,56	99,26	99,40	99,38	99,48	99,26
D	98,38	98,24	98,18	98,56	98,40	98,74	98,94	98,98	98,84	98,80
L	97,60	97,40	97,38	97,46	97,34	98,58	98,54	98,80	98,98	98,80

### A.4.4. VOHOG

		$\eta = 2$						
	Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	97,38	0,9710	0,9768	0,0232	0,0292	0,9768	0,9708
	I	94,84	0,9448	0,9524	0,0476	0,0556	0,9524	0,9444
	D	96,02	0,9485	0,9732	0,0268	0,0528	0,9732	0,9472
	L	93,54	0,9214	0,9520	0,0480	0,0812	0,9520	0,9188
$\beta = 12$	F	98,04	0,9743	0,9868	0,0132	0,0260	0,9868	0,9740
	I	95,50	0,9491	0,9616	0,0384	0,0516	0,9616	0,9484
	D	96,50	0,9469	0,9852	0,0148	0,0552	0,9852	0,9448
	L	94,88	0,9332	0,9668	0,0332	0,0692	0,9668	0,9308
$\beta = 16$	F	98,34	0,9741	0,9932	0,0068	0,0264	0,9932	0,9736
	I	96,06	0,9564	0,9652	0,0348	0,0440	0,9652	0,9560
	D	96,52	0,9511	0,9808	0,0192	0,0504	0,9808	0,9496
	L	94,90	0,9293	0,9720	0,0280	0,0740	0,9720	0,9260
$\beta = 32$	F	98,66	0,9769	0,9968	0,0032	0,0236	0,9968	0,9764
	I	97,58	0,9652	0,9872	0,0128	0,0356	0,9872	0,9644
	D	97,34	0,9625	0,9852	0,0148	0,0384	0,9852	0,9616
	L	95,58	0,9474	0,9652	0,0348	0,0536	0,9652	0,9464

		$\eta = 4$							
		Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	98,48	0,9871	0,9824	0,0176	0,0128	0,9824	0,9872	
	I	96,72	0,9598	0,9752	0,0248	0,0408	0,9752	0,9592	
	D	96,86	0,9571	0,9812	0,0188	0,0440	0,9812	0,9560	
	L	96,52	0,9682	0,9620	0,0380	0,0316	0,9620	0,9684	
$\beta = 12$	F	99,06	0,9877	0,9936	0,0064	0,0124	0,9936	0,9876	
	I	97,48	0,9733	0,9764	0,0236	0,0268	0,9764	0,9732	
	D	97,38	0,9676	0,9804	0,0196	0,0328	0,9804	0,9672	
	L	97,20	0,9694	0,9748	0,0252	0,0308	0,9748	0,9692	
$\beta = 16$	F	99,30	0,9928	0,9932	0,0068	0,0072	0,9932	0,9928	
	I	97,64	0,9730	0,9800	0,0200	0,0272	0,9800	0,9728	
	D	97,36	0,9672	0,9804	0,0196	0,0332	0,9804	0,9668	
	L	97,44	0,9755	0,9732	0,0268	0,0244	0,9732	0,9756	
$\beta = 32$	F	99,58	0,9940	0,9976	0,0024	0,0060	0,9976	0,9940	
	I	98,62	0,9845	0,9880	0,0120	0,0156	0,9880	0,9844	
	D	97,64	0,9678	0,9856	0,0144	0,0328	0,9856	0,9672	
	L	97,48	0,9775	0,9720	0,0280	0,0224	0,9720	0,9776	

		$\eta = 8$							
		Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	98,76	0,9955	0,9796	0,0204	0,0044	0,9796	0,9956	
	I	97,74	0,9768	0,9780	0,0220	0,0232	0,9780	0,9768	
	D	96,92	0,9707	0,9676	0,0324	0,0292	0,9676	0,9708	
	L	96,86	0,9688	0,9684	0,0316	0,0312	0,9684	0,9688	
$\beta = 12$	F	99,18	0,9944	0,9892	0,0108	0,0056	0,9892	0,9944	
	I	97,86	0,9811	0,9760	0,0240	0,0188	0,9760	0,9812	
	D	97,18	0,9701	0,9736	0,0264	0,0300	0,9736	0,9700	
	L	97,34	0,9774	0,9692	0,0308	0,0224	0,9692	0,9776	
$\beta = 16$	F	99,36	0,9944	0,9928	0,0072	0,0056	0,9928	0,9944	
	I	98,20	0,9859	0,9780	0,0220	0,0140	0,9780	0,9860	
	D	97,20	0,9720	0,9720	0,0280	0,0280	0,9720	0,9720	
	L	97,46	0,9763	0,9728	0,0272	0,0236	0,9728	0,9764	
$\beta = 32$	F	99,62	0,9964	0,9960	0,0040	0,0036	0,9960	0,9964	
	I	98,38	0,9832	0,9844	0,0156	0,0168	0,9844	0,9832	
	D	97,80	0,9712	0,9852	0,0148	0,0292	0,9852	0,9708	
	L	97,42	0,9763	0,9720	0,0280	0,0236	0,9720	0,9764	



		$\eta = 16$							
		Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	98,56	0,9907	0,9804	0,0196	0,0092	0,9804	0,9908	
	I	97,22	0,9777	0,9664	0,0336	0,0220	0,9664	0,9780	
	D	96,38	0,9655	0,9620	0,0380	0,0344	0,9620	0,9656	
	L	96,38	0,9647	0,9628	0,0372	0,0352	0,9628	0,9648	
$\beta = 12$	F	99,52	0,9972	0,9932	0,0068	0,0028	0,9932	0,9972	
	I	97,80	0,9776	0,9784	0,0216	0,0224	0,9784	0,9776	
	D	96,94	0,9711	0,9676	0,0324	0,0288	0,9676	0,9712	
	L	97,02	0,9693	0,9712	0,0288	0,0308	0,9712	0,9692	
$\beta = 16$	F	99,38	0,9964	0,9912	0,0088	0,0036	0,9912	0,9964	
	I	97,96	0,9804	0,9788	0,0212	0,0196	0,9788	0,9804	
	D	96,68	0,9661	0,9676	0,0324	0,0340	0,9676	0,9660	
	L	97,50	0,9737	0,9764	0,0236	0,0264	0,9764	0,9736	
$\beta = 32$	F	99,56	0,9964	0,9948	0,0052	0,0036	0,9948	0,9964	
	I	98,44	0,9863	0,9824	0,0176	0,0136	0,9824	0,9864	
	D	97,54	0,9722	0,9788	0,0212	0,0280	0,9788	0,9720	
	L	97,48	0,9752	0,9744	0,0256	0,0248	0,9744	0,9752	

#### A.4.5. T-OHOG

		$\eta = 4$							
		Vista	TA	P	R	FN	FP	TP	TN
$\beta = 8$	F	99,30	0,9901	0,9960	0,0040	0,0100	0,9960	0,9900	
	I	99,30	0,9901	0,9960	0,0040	0,0100	0,9960	0,9900	
	D	97,10	0,9758	0,9660	0,0340	0,0240	0,9660	0,9760	
	L	97,90	0,9743	0,9840	0,0160	0,0260	0,9840	0,9740	
$\beta = 12$	F	99,62	0,9980	0,9920	0,0080	0,0020	0,9920	0,9980	
	I	99,34	0,9900	0,9940	0,0060	0,0100	0,9940	0,9900	
	D	98,44	0,9762	0,9840	0,0160	0,0240	0,9840	0,9760	
	L	98,30	0,9782	0,9880	0,0120	0,0220	0,9880	0,9780	
$\beta = 16$	F	99,50	0,9960	0,9940	0,0060	0,0040	0,9940	0,9960	
	I	99,42	0,9956	0,9928	0,0072	0,0044	0,9928	0,9956	
	D	98,82	0,9834	0,9932	0,0068	0,0168	0,9932	0,9832	
	L	98,18	0,9770	0,9868	0,0132	0,0232	0,9868	0,9768	
$\beta = 32$	F	99,80	0,9980	0,9980	0,0020	0,0020	0,9980	0,9980	
	I	99,30	0,9901	0,9960	0,0040	0,0100	0,9960	0,9900	
	D	98,30	0,9745	0,9920	0,0080	0,0260	0,9920	0,9740	
	L	98,70	0,9822	0,9920	0,0080	0,0180	0,9920	0,9820	

## Apéndice B

# Publicaciones

Parte de este trabajo ha producido las siguientes publicaciones:

- Gonzalo Ballesteros, Luis Salgado, “*Optimized HOG for on-road video based vehicle verification*”, in Proceedings of the 2014 IEEE European Signal Processing Conference, EU-SIPCO’2014, Lisbon, Portugal, September 2014.
- Gonzalo Ballesteros, Luis Salgado, “*Histograms of oriented gradients for fast on-board vehicle verification*”, in Proceedings of the 2014 IEEE International Conference on Image Processing, ICIP’2014, Paris, Francia, October 2014.

# OPTIMIZED HOG FOR ON-ROAD VIDEO BASED VEHICLE VERIFICATION

Gonzalo Ballesteros<sup>1</sup>, Luis Salgado<sup>1,2</sup>

<sup>1</sup>Visual Processing and Understanding Lab, Universidad Autónoma de Madrid

<sup>2</sup>Grupo de Tratamiento de Imágenes, Universidad Politécnica de Madrid

## ABSTRACT

Vision-based object detection from a moving platform becomes particularly challenging in the field of advanced driver assistance systems (ADAS). In this context, on-board vision-based vehicle verification strategies become critical, facing challenges derived from the variability of vehicles appearance, illumination, and vehicle speed.

In this paper, an optimized HOG configuration for on-board vehicle verification is proposed which not only considers its spatial and orientation resolution, but descriptor processing strategies and classification. An in-depth analysis of the optimal settings for HOG for on-board vehicle verification is presented, in the context of SVM classification with different kernels. In contrast to many existing approaches, the evaluation is realized in a public and heterogeneous database of vehicle and non-vehicle images in different areas of the road, rendering excellent verification rates that outperform other similar approaches in the literature.

**Index Terms**— HOG, feature extraction, feature classification, video-based vehicle verification, O-HOG.

## 1. INTRODUCTION

Regarding in-vehicle video-based systems to identify the possibility of collision in the context of ADAS, the detection of vehicles is presented as the fundamental element in the processing chain. The methodology mostly used for on-board video-based detection of vehicles, so that it can perform in real time, consists of two stages. First, the whole image is analyzed to identify regions potentially containing vehicles. For this vehicle hypothesis generation stage, knowledge-based methods keep track of specific features present in the vehicle and its environment [1][2][3], while others resort to stereovision [4] or motion analysis [5]. Second, the hypothesized candidates are typically verified using features relating to their appearance. Template or model-based verification methods have been used, although recently learning-based methods are being adopted, in which the characteristics of vehicles are learned from a training set, and new candidates are dynamically classified according to the learned patterns.

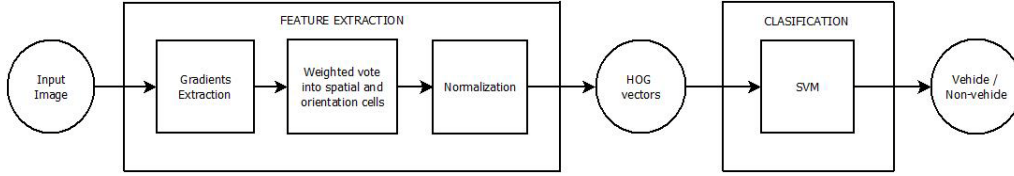
Learning-based hypothesis verification is typically approached as a two-class classification problem: feature vector is extracted from the image, and the sample is classified as vehicle or non-vehicle. The election of the feature space is critical to achieve a successful classification. Explicit features such as symmetry, edges or shadows, although intuitive, fast to compute and classify, show limited discriminating capabilities that bound the accuracy of the results. On the contrary, implicit features

do not make assumptions about vehicles appearance or shape, demanding higher computational resources but rendering significantly better classification results. Among the most commonly used implicit features [6][7], Histograms of Oriented Gradients (HOG) stand out for their excellent results in the object verification.

Dalal and Triggs [8] proposed HOG for human detection obtaining excellent results, and explore their application to different objects [9]. In recent studies, HOG is gaining attention for vehicle detection and classification, being used on images taken under different camera settings. In [10] vehicle top views taken from low-attitude air borne videos are considered, while most works consider more lateral and frontal views, either from wide area monitoring imaging [11] [15] or from on-board-like camera views [13][14][16][18]. Proposals range from straightforward applications of standard HOG and its variations, customized to the particular applications in combination with different classifiers, to combinations with other features as Haar-like [11][18] to improve detection performance. The reduction of the computation requirements for HOG-based drives many contributions, as in [12], where the use of masks adapted to the vehicle shape is proposed to speed up classification. Other works use an AdaBoost classifier on HOG features, but as in [10] they only deal with vehicle top views, or in [13] where only rear views of vehicles in front of the own vehicle (in the same lane) are considered. In other works such as [14], larger databases are considered, proposing cost-effective HOG descriptors where computational cost is alleviated considering only horizontal or vertical cells.

In almost all existing studies that use HOG, the configuration of the descriptor is not fully explored. Direct assumptions of the parameterization for people and other objects detection are applied. Moreover, the variability of the camera settings and databases used bound the impact of the solutions proposed as no direct comparison is possible. Particularly, except for [14], databases are mostly ad-hoc generated, non-public or very limited both, in the number of positive and negative samples and their heterogeneity. HOG configurations in terms of spatial and binning resolution have been analyzed [14]. However, the impact analysis of choices regarding the descriptor computation and the classification strategy is still required. Therefore, extensive quantitative experiments are lacking on the effectiveness of HOG descriptors for vehicle verification.

In this paper, an optimal configuration for HOG is proposed which not only considers spatial and orientation resolution, but descriptor processing strategies and classification. A deep analysis of the optimal settings for HOG for on-board vehicle verification is presented, in the context of SVM classification with different kernels. In contrast to many existing approaches, the evaluation is



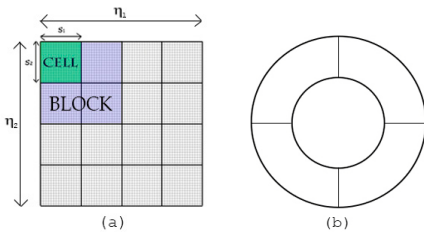
**Fig. 1.** Feature extraction and classification system based on HOG.

realized in a large and heterogeneous database of vehicle and non-vehicle images in different areas of the road. Results demonstrate a very significant improvement of the verification accuracy, outperforming other strategies in the state of the art [14].

## 2. HOG DESCRIPTOR

HOGs [8] result from the computation of local histograms of the orientation of the image gradients in a grid. The underlying idea is that objects appearance can be characterized by the local distribution of its edges orientation. A diagram of the HOG feature extraction and classification strategy is shown in Figure 1.

Gradients computation is the first step to generate the HOG descriptor: the application of the selected gradient operator results on edge intensity and orientation values for each pixel. The image is divided into regions called cells, as shown in Figure 2 (a), and the second step involves creating the cell histograms. Orientation binning is applied using an even number of intervals that span the range  $[0, \pi]$  or  $[0, 2\pi]$ . Each pixel in the cell contributes with a vote to the two closest histogram channels, weighted according to the gradient magnitude and the distance to the channel centers. The last step accounts for changes in illumination and contrast through locally normalizing gradient responses. As proposed in [8], cells are grouped into larger structures named blocks as shown in Figure 2 (a). For each block, the non-normalized vector holding all histograms of its cells is normalized using any standard norm such as the L1-norm, L2-norm, etc. Block overlap is also suggested to make this step more robust. The final HOG descriptor is the vector resulting from the concatenation of the normalized vectors of all the blocks.



**Fig. 2.** Cell configurations: rectangular (a)/ circular (b) HOG.

Rectangular (R-HOG) and circular (C-HOG) cell configurations have been both proposed (Figure 2 (b)). However, R-HOG is the most widely used for vehicles as it naturally adapts to the dominant vehicle geometry. Thus it is the one considered in this study.

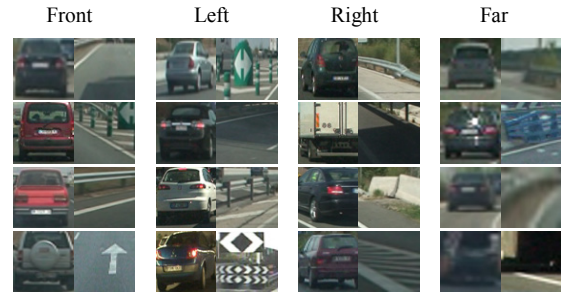
## 3. VEHICLE VERIFICATION USING STANDARD HOG

In order to classify the input samples into vehicle or non-vehicle, discriminative approaches largely outnumber generative methods in the literature. Indeed, discriminative

models usually give better performance on discriminative tasks than generative models. Among the discriminative approaches, Support Vector Machines (SVMs) have been extensively proposed for HOG-based vehicle verification, providing good generalization [21] and rendering excellent results, and rendering excellent results. Therefore, SVMs is the learning method adopted in this work.

### Classification Methodology

The evaluation is carried out in the most extensive public database for vehicle verification proposed in the literature [17]. The GTI vehicle database is a complete data set with



**Fig. 3.** GTI Database: vehicle (left) and non-vehicle (right) samples for front, left, right and far views.

4000 vehicles and 4000 non-vehicles images of size  $64 \times 64$  pixels. They have been taken from videos acquired with on-board forward looking cameras under different illumination and weather conditions with variable quality. Samples include partial views of vehicle rears, views misaligned with the image, shifted horizontally and/or vertically. Therefore, training and evaluation can be carried out considering a large variability of situations typical from on-line hypothesis generation systems. Additionally, samples are further classified into four sets according to the relative distance and position to the camera: left, front and right in the close-middle distance, and far distance. This allows exploring classification taken into account the relative pose of the vehicles. Figure 3 shows example images from this database.

Experiments are carried out based on 5-fold 50% cross-validation methodology, and results are evaluated as the average percentage of correctly classified samples.

### Experiments on HOG

When using HOG, it is necessary to define the descriptor parameterization. Experiments have been performed to obtain the most suitable set of parameters and thus get the best possible classification accuracy following some of the recommendations in [8]. These configurations, referred in some works as standard HOG (S-HOG), are explored evaluating the impact of different parameter settings.

Configuration parameters include the number of cells,  $\eta_1 \times \eta_2$ , block size,  $b$ , and number of orientation bins,  $\beta$ . Square cells are considered, as they are the most used in the literature and adapt naturally to the square images size ( $64 \times 64$  pixels). Therefore,  $\eta_1 = \eta_2 = \eta$  and  $\eta = \{2, 4, 8, 16\}$  that correspond to  $\{4, 16, 64, 256\}$  cells. The number of orientation bins explored are  $\beta = \{8, 12, 16, 32\}$ .

For the gradients extraction phase, as suggested in [8], the mask used is  $[-1 \ 0 \ 1]$  and the range of angles is  $[0, \pi]$  so the sign of gradients is ignored. The L2-norm is used to normalize square blocks of side  $b=2$ , ( $2 \times 2$  cells). Finally, to classify the HOG vectors, a linear SVM is trained.

Front	$\beta=8$	$\beta=12$	$\beta=16$	$\beta=32$
$\eta=2$	98.64	95.36	98.30	92.78
$\eta=4$	<b>99.48</b>	98.52	99.18	98.28
$\eta=8$	98.68	98.54	99.16	98.40
$\eta=16$	98.62	98.80	98.62	98.76

Left	$\beta=8$	$\beta=12$	$\beta=16$	$\beta=32$
$\eta=2$	93.28	91.68	94.14	90.04
$\eta=4$	<b>97.64</b>	96.96	97.04	95.70
$\eta=8$	97.34	96.56	96.96	96.04
$\eta=16$	96.46	96.26	96.02	95.94

Right	$\beta=8$	$\beta=12$	$\beta=16$	$\beta=32$
$\eta=2$	92.60	90.22	93.40	88.66
$\eta=4$	96.00	95.46	<b>96.22</b>	95.10
$\eta=8$	94.70	95.12	94.54	94.46
$\eta=16$	93.96	95.36	93.72	95.64

Far	$\beta=8$	$\beta=12$	$\beta=16$	$\beta=32$
$\eta=2$	92.10	87.12	91.92	83.28
$\eta=4$	<b>97.76</b>	96.54	97.04	95.56
$\eta=8$	97.28	96.70	96.44	96.18
$\eta=16$	97.66	97.62	96.98	97.54

**Table 1.** Accuracy of Standard HOG.

Experiments were conducted training individual classifiers for each relative pose of the vehicles. The classification rates (%) are presented in Table 1 where the highest scores are highlighted in bold. We can draw some conclusions from these results, starting first with the close-middle front view. As expected, this is the view that gets the best results in classification. Hypothesis generated for vehicles located in front of the own vehicle show well defined and quite stable geometrical patterns that adapt perfectly to the HOG topology. Analyzing the cells' granularity,  $\eta=2$ , *i.e.* four cells, results in unstable and poor classification results. However with  $\eta=4$  we get the same or even better results than for higher granularities. This means that 16 cells get sufficient detail of the edges orientation distribution for optimal classification: a higher number of cells do not add relevant information for classification, although that information overload does not significantly underperform. Regarding the number of orientation bins, it can also be seen that with the lowest number,  $\beta=8$ , is enough for a good rating, reaching the maxima of 99.48%, and that higher values do not guarantee better results. For the right and left views,  $\eta=2$  underperforms dramatically. The best results – significantly lower than for those for the front view - are obtained for  $\eta=4$ , and there is a clear trend to lose accuracy for higher

number of cells. Lateral vehicle views do show much higher edge pattern variability than vehicle rears, and also the non-vehicle samples in those areas show richer and similarly oriented edge patterns to those of the vehicles (particularly noticeable for right views). Increasing the granularity in cells or orientation bins cause that for some vehicle images, their geometrical structures are diluted in a long descriptor, becoming very similar to that of images where there are no cars but also have a high level of detail.

Finally, far rear views are taken from vehicles at longer distances from the camera, thus showing highly softened details and lower quality than those in the close-middle range. Gradients extraction is not optimal and affects classification, but less than may be expected: these far distance views show very similar structure and stability to that in the close-middle front views that suit perfectly R-

$\eta=4$	[-1 0 1]			Sobel		
	$\beta=8$	$\beta=12$	$\beta=16$	$\beta=8$	$\beta=12$	$\beta=16$
Front	99.48	98.52	99.18	98.28	98.54	99.28
Left	97.64	96.96	97.04	95.70	96.22	96.66
Right	96.00	95.46	96.22	95.10	95.36	95.38
Far	97.76	96.54	97.04	95.56	96.34	97.02
$\eta=8$						
Front	98.68	98.54	99.16	98.68	98.34	99.00
Left	97.34	96.56	96.96	96.74	96.04	96.24
Right	94.70	95.12	94.54	94.78	94.76	93.94
Far	97.28	96.70	96.44	96.92	96.48	96.56

**Table 2.** Verification rate using different detectors.

HOGs. Although the HOG descriptor provides very good results in view-dependent vehicle verification, its potentiality is not fully exploited in the explored standard configuration. In the next section, how to improve the descriptor and classification modifying some of its parameters is studied, trying to find the optimum configuration for vehicle verification.

#### 4. OPTIMIZING HOG BASED VEHICLE VERIFICATION

Classification performance not only depends on HOG configuration, but on the classifier and strategies used to compute the descriptor. Different alternatives are here evaluated following the steps described in Figure 1.

##### A. Feature extraction phase

###### 1) Gradients extraction

In several works [14][18], Sobel detector is proposed for vehicle detection and verification. Results shown in Table 2 demonstrate that, overall, it underperforms, showing a trend to narrow the differences with  $[-1 \ 0 \ 1]$  when the binning resolution increases. The mask used in this study demonstrates more stability in the results and significantly higher accuracy rates for lower parameter values, thus resulting in shorter but more efficient descriptors.

Expanding the range of angles is now evaluated. For the application to vehicles [9], considering the sign is suggested, although it also significantly impacts the descriptor size and therefore, the cost of classification. This motivates in some works such as [14] the decision to keep the original positive range as good results are still obtained.

	[0, $\pi$ ]			[- $\pi$ , $\pi$ ]		
	$\beta=8$	$\beta=12$	$\beta=16$	$\beta=8$	$\beta=12$	$\beta=16$
$\eta=4$						
Front	99.48	98.52	99.18	99.00	99.32	99.28
Left	97.64	96.96	97.04	98.72	98.82	99.04
Right	96.00	95.46	96.22	98.14	98.34	98.36
Far	97.76	96.54	97.04	97.38	98.72	98.50
$\eta=8$						
Front	98.68	98.54	99.16	99.20	99.52	99.76
Left	97.34	96.56	96.96	98.70	99.00	99.10
Right	94.70	95.12	94.54	98.32	97.98	98.32
Far	97.28	96.70	96.44	97.54	98.34	98.46

**Table 3.** Verification rate extending the orientation range.

Table 3 shows the comparative results between using  $[0, \pi]$  and  $[-\pi, \pi]$  ranges. The impact of sign is very significant. In almost all configurations, considering  $[-\pi, \pi]$  renders better results for the same number of bins, being particularly relevant in the most critical right and far views. The improved results obtained for the same number of bins when sign is considered suggests that, for vehicles, sign information may be even more relevant than orientation resolution. Although the trend has now changed and better performance is generally obtained for higher values of  $\beta$ , sign information should prevail to orientation resolution if the descriptor size is an issue for the application.

## 2) Normalization

In [8] or [14] the use of overlapping blocks of 2x2 cells is proposed to account for changes in illumination.

	2x2			1x1		
	$\beta=8$	$\beta=12$	$\beta=16$	$\beta=8$	$\beta=12$	$\beta=16$
$\eta=4$						
Front	99.00	99.32	99.28	99.30	99.54	99.60
Left	98.72	98.82	99.04	98.60	99.04	99.18
Right	98.14	98.34	98.36	97.90	98.36	98.50
Far	97.38	98.72	98.50	98.04	98.40	98.32
$\eta=8$						
Front	99.20	99.52	99.76	99.50	99.56	99.60
Left	98.70	99.00	99.10	98.50	98.86	98.80
Right	98.32	97.98	98.32	98.24	98.34	98.40
Far	97.54	98.34	98.46	97.54	98.06	98.20

**Table 4.** Verification rate using different block sizes.

Alternatives are here evaluated, going down to blocks of 1 cell, i.e. take only into account changes in illumination between pixels of the same cell. As shown in Table 4, the performance of 1x1 blocks is very similar to 2x2 overlapped blocks. However, the former renders more stable values with shorter descriptors, reducing the cost of the descriptors computation and that of classification.

Regarding the norm used, experiments (Table 5) confirm L2 outperforming L1 (suggested in [9]) for all situations. The classification accuracy degrades here particularly for the far and right views reaching in some combinations around a 1-2% loss.

## B. Classification phase

A linear SVM is used as baseline for classification in this study. However, higher order polynomials Kernels have been also evaluated. As shown in Table 6, the degree-2 polynomial kernel outperforms for all categories and descriptor configurations. The highest gains correspond to the critical right view in all  $(\eta, \beta)$  configurations but (4, 16),

	L2			L1		
	$\beta=8$	$\beta=12$	$\beta=16$	$\beta=8$	$\beta=12$	$\beta=16$
$\eta=4$						
Front	99.30	99.54	99.60	99.26	99.40	99.50
Left	98.60	99.04	99.18	98.42	98.84	98.62
Right	97.90	98.36	98.5	97.50	97.70	97.48
Far	98.04	98.40	98.32	96.92	97.94	98.24
$\eta=8$						
Front	99.50	99.56	99.60	98.78	99.26	99.52
Left	98.50	98.86	98.80	98.40	98.76	98.78
Right	98.24	98.34	98.40	97.86	98.26	97.82
Far	97.54	98.06	98.20	97.54	95.44	98.20

**Table 5.** Verification rate for different norms.

followed closely by the left and far ones. The higher flexibility of the degree-2 polynomial kernel adapts to the views showing lower discrimination, thus rendering a more balanced and stable verification rates in all categories.

	linear			degree-2		
	$\beta=8$	$\beta=12$	$\beta=16$	$\beta=8$	$\beta=12$	$\beta=16$
$\eta=4$						
Front	99.30	99.54	99.60	<b>99.82</b>	99.72	99.80
Left	98.60	99.04	99.18	99.18	99.48	<b>99.64</b>
Right	97.90	98.36	98.58	98.58	98.88	98.78
Far	98.04	98.40	98.32	98.16	<b>98.84</b>	98.80
$\eta=8$						
Front	99.50	99.56	99.60	99.68	99.68	99.58
Left	98.50	98.86	98.80	99.14	99.48	99.40
Right	98.24	98.34	98.40	98.94	99.04	<b>99.28</b>
Far	97.54	98.06	98.20	97.92	98.66	98.44

**Table 6.** Verification rate with SVM kernels.

## 5. DISCUSSION

The proper configuration of the HOG descriptor and the classifier used demonstrates to be fundamental for on-board vehicle verification. Sign information is critical for classification. Substantially higher scores than unsigned HOGs are obtained even for the same binning resolution, being particularly relevant for the complex lateral vehicle views. The simple  $[-1,0,1]$  mask for gradients computation render better results than others proposed in the literature at a lower computational cost. L2 normalization outperforms, but regarding overlapped multi-cell approaches (block based), their cost in terms of descriptor length and computation barely justifies the marginal classification improvement achieved only for some descriptor combinations. Therefore the proposed descriptor computation choices are:  $[-1,0,1]$  filter for gradient extraction;  $[-\pi, \pi]$  as orientation range, and individual L2-norm cell normalization (blocks of 1 cell).

Regarding the number of cells (Table 6),  $\eta=4$  provides an excellent performance regardless on the kernel used, outperforming for all vehicle poses but for some combinations in the right view. Particularly in this case, higher spatial and binning granularities are required to well overpass 99%. Moving from  $\beta=8$  to 12 is justified for the lateral and far views as it improves very significantly the results regardless on the kernel used. Going further benefits verification for vehicles in lateral views, worthy to be considered especially for the right one.

Considering average classification values, i.e. computing for each  $\beta$  the average verification rate of the four classifiers,  $\eta=4$  outperforms  $\eta=8$  for all values of  $\beta$ ,

confirming it as the best choice for vehicle verification. The best average results are 99.25% for  $(\eta, \beta)=(4, 16)$  and a 99.23% for  $(\eta, \beta)=(4, 12)$ , confirming that the degree-2 polynomial adapts better to vehicle verification. Moreover, one of the advantages of considering independent classifiers for each view is the possibility to adapt the  $(\eta, \beta)$  operational point to that offering the best and more stable results. In Table 7, verification rates (VR) for the optimal configuration (O-HOG) for each view are summarized.

Recently, in [14], a new HOG descriptor, V-HOG, was

	S-HOG [18]			O-HOG		
	$(\eta, \beta)$	VR(%)	NCe	$(\eta, \beta)$	VR(%)	NCe
<i>Front</i>	(16,8)	99.18	7200	(4,8)	99.82	128
<i>Left</i>	(8,12)	98.32	2352	(4,16)	99.64	256
<i>Right</i>	(8,12)	97.44	2352	(8,16)	99.28	1024
<i>Far</i>	(8,12)	98.40	2352	(4,12)	98.84	192
	V-HOG[18]			O-HOG		
<i>Front</i>	(4,16)	97.68	64	(2,16)	99.44	64
<i>Left</i>	(4,36)	97.02	144	(2,16)	98.56	64
<i>Right</i>	(4,16)	95.54	64	(2,16)	98.64	64
<i>Far</i>	(4,12)	95.60	48	(2,16)	97.80	64

**Table 7.** HOG, V-HOG and O-HOG.

proposed which only considers vertical cells to achieve high computational efficiency. In [14], a throughout study on the standard HOG (S-HOG) configuration is performed; the best results are presented in Table 7 (S-HOG[14]), together with the descriptors size for each view (NCe column). As it can be observed, the O-HOG here proposed, with much shorter descriptors, outperforms S-HOG in all categories. In average, the number of components required for O-HOG is about a 10% of S-HOG (400 vs 3564 components). In the same table, the results of the V-HOG proposed in [14] are compared with O-HOG configurations requiring similar number of components. O-HOG largely outperform V-HOG with gains ranging between 2-3% for all categories. Compared with the results in [20], where average detection rates of 92.2% are achieved, an average 6% gain is obtained using the efficient O-HOG configuration (64 components).

## 6. CONCLUSION

In this paper, the application of HOG for video-based vehicle verification is explored. The impact of different processing approaches and descriptor configurations is in depth analyzed, proposing an optimized HOG that provides excellent results reducing the computational cost with respect to standard HOG. By selecting an appropriate classifier, results are further improved. Nonlinear kernels on SVMs are the most suitable choice, with the degree-2 polynomial rendering the highest verification rates. Training independent classifiers for the different views allows proposing the combination of spatial and orientation resolution that better adapt to each view, yielding excellent results that outperform other approaches in the literature.

## REFERENCES

[1] J. Hwang, K. Huh, and D. Lee. Vision based vehicle detection and tracking algorithm design. *Optical Engineering*, 48(12):127201–127201–10, 2009.

[2] G. Y. Song, K. Y. Lee, and J. W. Lee. Vehicle detection by edge-based candidate generation and appearance-based

classification. In *Intelligent Vehicles Symposium*, 2008 IEEE, pages 428–433, 2008.

[3] T. Schamm, C. Carlowitz, and J.M. Zollner. On-road vehicle detection during dusk and at night. *IEEE Intelligent Vehicles Symposium (IV)*, 2010., pp. 418–423, 2010.

[4] F. Oniga and S. Nedeveschi. Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection. *Vehicular Technology, IEEE Transactions on*, 59(3):1172–1182, 2010.

[5] F. Woelk and R. Koch. Robust monocular detection of independent motion by a moving observer. In *Proceedings of the 26th DAGM Symposium*, ser. *Lecture Notes in Computer Science*, pp. 27–35. Springer, 2004.

[6] J. Zhou, D. Gao, and D. Zhang. Moving vehicle detection for automatic traffic monitoring. *IEEE Transactions on Vehicular Technology*, 56(1):51–59, Jan 2007.

[7] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection using evolutionary Gabor filter optimization. *Intelligent Transportation Systems, IEEE Transactions on*, 6(2):125–137, 2005.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for man detection. In *Computer Vision and Pattern Recognition, CVPR. IEEE Conference on*, vol. 1, pp. 886–893, 2005.

[9] N. Dalal and B. Triggs. Object detection using histograms of oriented gradients. *Pascal VOC Workshop. ECCV*, 2006.

[10] X. Cao, C. Wu, P. Yan, and X. Li. Linear svm classification using boosting hog features for vehicle detection in low-altitude airborne videos. In *Image Processing (ICIP), IEEE International Conference on*, pp. 2421–2424, 2011.

[11] P. Liang, G. Teodoro, H. Ling, E. Blasch, G. Chen, and L. Bai. Multiple kernel learning for vehicle detection in wide area motion imagery. In *Information Fusion (FUSION), International Conference on*, pages 1629–1636, 2012.

[12] Y. Lv, B. Yao, Y. Wang, and S.-C. Zhu. Reconfigurable templates for robust vehicle detection and classification. In *Applications of Computer Vision (WACV), IEEE Workshop on*, pages 321–328, 2012.

[13] B. Southall, M. Bansal, and J. Eledath. Real-time vehicle detection for highway driving. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 541–548, 2009.

[14] J. Arróspide, L. Salgado, and M. Camplani. Image-based on-road vehicle detection using cost-effective histograms of oriented gradients. *Journal of Visual Communication and Image Representation*, 24(7):1182 – 1190, 2013.

[15] Z. Chen and T. Ellis. Multi-shape Descriptor Vehicle Classification for Urban Traffic, pages 456–461. Institute of Electrical and Electronics Engineers, Dec 2011.

[16] P. Rybski, D. Huber, D. Morris, R. Hoffman. Visual classification of coarse vehicle orientation using histogram of oriented gradients features. *IEEE Intelligent Vehicles Symposium (IV)*. 921-928. 2010

[17] GTI vehicle image database, image processing group at upm, <<http://www.gti.ssr.upm.es/data>>, 2011.

[18] P. Negri, X. Clady, S. M. Hanif, and L. Prevost. A cascade of boosted generative and discriminative classifiers for vehicle detection. *EURASIP Journal on Advances in Signal Processing*, 2008(1):782432.

[19] J. Jiang and H. Xiong. Fast Pedestrian Detection Based on HOG-PCA and Gentle AdaBoost, pages 1819–1822. Institute of Electrical and Electronics Engineers, Aug 2012.

[20] D. Balcones, D.F. Llorca, M.A. Sotelo, M. Gavilán, S. Álvarez, I. Parra, and M. Ocaña. Real-Time Vision-Based Vehicle Detection for Rear-End Collision Mitigation Systems. *EUROCAST*, vol. 5717, pp. 320–325, 2009.

[21] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Science+Business Media, 2006.

Dear Mr. Gonzalo Ballesteros,

The review process of EUSIPCO 2014 is completed and we are pleased to announce that your paper #1569924423 ('Optimized HOG for On-Road Video Based Vehicle Verification') has been ACCEPTED for publication. Congratulations!

The reviews are included below and can alternatively be found at <http://edas.info/showPaper.php?m=1569924423>, using your EDAS username [gonzalo.ballestero@estudiante.uam.es](mailto:gonzalo.ballestero@estudiante.uam.es).

Full information regarding EUSIPCO'2014 is posted on the conference web site (<http://www.eusipco2014.org>) which will list the complete technical program as it becomes available.

CRITICAL INFORMATION:

1) Included at the end of this email are the reviewers' comments on your paper. Please make all the necessary changes based on the reviewers' comments and suggestions. We recall that the revised paper has a maximum length of 5 pages and must follow the format detailed in the "paper submission" section of "Authors" tab in the EUSIPCO'2014 conference webpage.

2) Your final, publication-ready manuscript MUST be uploaded until June 23, 2014, in the form of a PDF that is fully compliant with IEEE Xplore. In order to ensure Xplore-compliance, you must use IEEE PDF eXpress to either (1) convert your manuscript from its original source form into an Xplore-compliant PDF, or (2) check your manuscript's PDF to verify compliance with IEEE Xplore. Follow the instructions at EUSIPCO'2014 conference webpage for further details.

3) You MUST upload your final Xplore-compatible PDF to EDAS. The upload process will involve completion of a copyright form (also provided through EDAS); every manuscript MUST have a completed copyright form. Again, follow the instructions at EUSIPCO'2014 conference webpage for further details.

4) At least one author MUST BE REGISTERED for the conference at a non-student rate by June 23, 2014. Failure to have such registration will result in removal of the paper from the Technical Program. A single, full registration may cover up to three (3) papers.

5) Note that papers which are not presented at the conference will be excluded from the official proceedings and will not appear on IEEE Xplore. It is therefore very important that at least one author of the paper attends the conference to present the work.

Please, note also that the EUSIPCO organizers cannot assist the authors in solving specific problems related to the formatting of the paper. All the information that the organizers can provide is already available in the conference website. In addition, the authors can also refer to the EDAS help tool and the FAQ's available at <http://edas.info/listFAQ.php?c=9281>. We encourage authors to prepare well in advance the camera-ready papers to avoid last-minute problems with the formatting close to the deadline.

We are looking forward to seeing you in Lisbon in September.

Best regards,  
Leonel Sousa, General Chair  
Mário Figueiredo, António Rodrigues, TPC Chairs



# HISTOGRAMS OF ORIENTED GRADIENTS FOR FAST ON-BOARD VEHICLE VERIFICATION

*Gonzalo Ballesteros<sup>1</sup>, Luis Salgado<sup>1,2</sup>*

<sup>1</sup>Visual Processing and Understanding Lab, Universidad Autónoma de Madrid

<sup>2</sup>Grupo de Tratamiento de Imágenes, Universidad Politécnica de Madrid

## ABSTRACT

Histograms of Oriented Gradients (HoGs) provide excellent results in object detection and verification. However, their high computational cost bound their applicability in some critical real-time scenarios, such as for video-based on-board vehicle detection systems. In this work, an efficient HOG configuration for pose-based on-board vehicle verification is proposed, which alleviates both the computational cost and required feature vector length without reducing classification performance. The impact on classification of some critical configuration and processing parameters is in depth analyzed to propose a baseline efficient descriptor. Based on the analysis of its cells contribution to classification, new view-dependent cell-configuration patterns are proposed, resulting in reduced descriptors which provide an excellent balance between performance and computational cost, rendering higher verification rates than other works in the literature.

**Index Terms**— HOGs, vehicle verification, efficient descriptor configuration, view-dependent classification.

## 1. INTRODUCTION

Vision-based object detection from a moving platform becomes particularly challenging in the field of advanced driver assistance systems (ADAS). On-board vision systems are an excellent source of information where real-time detection of vehicles becomes a critical task, facing challenges derived from the variability of vehicles appearance, illumination, shadows, and vehicle speed.

The most used methodology for vehicles detection consists of two stages. First, the whole image is fast analyzed using knowledge-based [1][2], motion [3][4] or stereovision [5] methods to identify regions potentially containing vehicles. Second, these candidates are verified using features related to their appearance through model-based approaches or, more recently, learning-based methods [6] in which the vehicles characteristics are learned from a training set, and new candidates are classified according to learned patterns.

Learning-based hypothesis verification is approached as a two-class classification problem: a feature vector is extracted from the image, and the sample is classified as vehicle or non-vehicle. Among the most commonly used

descriptors, such as those based on PCA [7] or Gabor filters [8], HOGs stands out for their excellent performance in object detection. However, their performance is bounded by a tradeoff between complexity of descriptor configuration and real-time operation.

In this work, an efficient HOG configuration for pose-based on-board vehicle verification is proposed, which alleviates both the computational cost and required feature vector length without reducing classification performance. The impact on classification of some critical configuration and processing parameters is in depth analyzed to propose a baseline efficient descriptor. Furthermore, subsets of cells corresponding to view-dependent patterns are investigated, resulting in reduced descriptors where only the most significant cells are considered for classification. The hypothesis that there are areas in the images which do not contain useful information or even gather misleading information for classification is verified. The classification accuracy is demonstrated on a large public database, outperforming other approaches recently proposed.

## 2. PREVIOUS WORK

Although originally proposed for people detection [9], HOGs were rapidly expanded to other fields such as face recognition [10]. In recent studies, HOGs have been adopted for video-based vehicle detection and verification, although with a limited exploration of the descriptor configuration as typically that for people and other objects detection are directly applied. This so-called standard HOG is present in works such as [11] for vehicle detection in aerial views, [12] for preceding vehicle detection, [13] for rear collision avoidance, or [14] for view-dependent vehicle verification. Although works either provide only qualitative results [12], or use limited non-public sequences and databases [13][11], in [14] a large public database of vehicle hypothesis obtained from an on-board forward looking camera is considered. Computational efficiency is addressed differently in the literature: from standard HOG ad-hoc hardware implementations [15], to descriptor simplifications [12][14][16] reducing the orientation range considered, modifying the weighted contributions to adjacent bins, or proposing alternative cell and blocks configurations to alleviate the cost of classification. The use of different block

sizes is explored in [17], with fairly low accuracy results, while in [18] the use of masks adapted to the vehicle shape is proposed to speed up classification with good results in the classification between different types of vehicles. In [14], different reduced configurations are evaluated and the V-HOG, that uses only vertical cells, is proposed: it provides better verification accuracy than other approaches with a reduction of the computational cost that allows real-time operation. Combination with other features, such as Haar-like [11][16], is also proposed to speed up detection.

### 3. FEATURE EXTRACTION AND CLASIFICATION BASED ON HOG

#### 3.1. Feature extraction phase

The main idea of using HOG is that objects appearance can be characterized by the local distribution of its edges orientation. The HOG descriptor results from the computation of local histograms of orientation of the image gradients in a grid. A scheme summarizing the feature extraction system using HOG is shown in Figure 1.

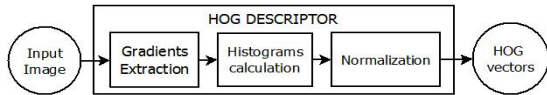


Figure 1. HOG descriptor generation

The gradients extraction phase computes, for each image pixel, the edge magnitude and orientation. The image is divided into cells, and for each cell a histogram of its pixels orientation is obtained. The last step is the normalization of the histograms to account for uneven illumination and shadows. As proposed in [9], cells are grouped into larger structures called blocks. For each block the non-normalized vector holding the histograms of its cells is normalized using any standard norm. The overlap of blocks is also suggested to make this step more robust. The final HOG descriptor is the resulting vector of the concatenation of the normalized blocks. Among the cell configurations proposed, the rectangular (R-HOG) geometry is assumed in this work, as it naturally adapts to the dominant vehicle geometry.

#### 3.2. Classification phase

For HOG based classification of the input samples into vehicles and non-vehicles, Support Vector Machines (SVMs) have been extensively proposed: they render excellent results, and provide better generalization involving lower number of parameters than other discriminative approaches such as Neural Networks [20]. A linear SVM is adopted in this work.

Evaluation is carried out in the only extensive public database for vehicle verification proposed in the literature [18]. The GTI vehicle database is a complete data set with 4000 vehicles and 4000 non-vehicles images of size 64x64 pixels. Images were obtained from videos taken with an on-

board forward looking camera, considering a large variability of situations typical from on-line hypothesis generation systems. Furthermore, to explore classification taking into account the vehicles pose, images are also organized into four classes according to their relative position and distance to the camera: frontal (CF), left (L) and right (R) views in the middle-close distance, and far distance (F). Examples of vehicle and non-vehicle images for the four classes are shown in Figure 2. Experiments are carried out based on 5-fold 50% cross-validation methodology, and the classification accuracy, the average percentage of correctly classified samples is evaluated.



Figure 2. GTI Database images. For frontal (CF), left (L), right (R), and far (F) views: two vehicle (left) and two non-vehicle (right) samples.

#### 3.3. Experiments on HOG

The configuration of the parameters of HOG descriptor is important to perform a good classification. Table I shows the best results obtained with the standard HOG (S-HOG) training individual classifiers for each view. Different values of the number of cells ( $\eta \times \eta = 4, 16, 64, 256$ ) and the number of orientation bins ( $\beta = 8, 12, 16, 32$ ) are evaluated. The L2-norm is used to normalize blocks of 2x2 cells.

View	Acc (%)	H	$\beta$
CF	99.48	4	8
L	97.64	4	8
R	96.22	4	16
F	97.76	4	8

Table I: Best accuracy results for standard HOG

The best results are obtained for  $\eta=4$  (16 cells), the lowest spatial resolution considered, which serve as a starting point to reduce the computational cost in this work. As expected, the highest scores correspond to the frontal view: hypothesis generated for vehicles located in front of the own vehicle show well defined and quite stable geometrical patterns that adapt perfectly to the HOG topology. Regarding the orientation resolution, low values (the lowest ones explored for three of the four views) are enough for a good rating. Larger granularity does not imply a better outcome.

In the following sections, first alternative configurations of the parameters or the HOG descriptor to reduce the computational cost without reducing performance are proposed. Then, based on the hypothesis that there are cells in the descriptor that are not relevant for classification, we analyze their influence for each view. Based on the results, new view-dependent cell configurations are proposed.

$\eta=4$ View	S-HOG		[- $\pi$ , $\pi$ ]		NCCG		2x1		1x1	
	$\beta=8$	$\beta=16$	$\beta=8$	$\beta=16$	$\beta=8$	$\beta=16$	$\beta=8$	$\beta=16$	$\beta=8$	$\beta=16$
CF	99.48	99.18	99.00	99.28	99.06	99.14	99.14	99.52	99.20	99.56
L	97.64	97.04	98.72	99.04	98.40	99.00	98.50	98.84	98.42	98.86
R	96.00	96.22	98.14	98.36	97.44	98.28	97.96	98.02	97.94	98.22
F	97.76	97.04	97.38	98.50	97.54	98.02	97.70	98.16	97.60	97.98

Table II: Accuracy rates for efficient configuration of parameters

#### 4. EFFICIENT CONFIGURATION PARAMETERS

The impact on the performance of alternative processing and configuration parameters of the descriptor is studied.

##### A. Gradients extraction

Although orientation sign is typically disregarded for efficiency purposes [13][14][16], Table II demonstrates its high impact (columns [- $\pi$ ,  $\pi$ ]) particularly on non-frontal vehicle views. Compared with S-HOG (no sign considered), better results are obtained for the same granularity, being more significant for  $\beta=16$  in all views. It seems more important to have a complete orientation of the gradient than smaller bins orientation to describe the structure of the vehicles. Improved results are achieved keeping the same computational cost.

##### B. Histograms Calculation

The next improvement focuses on reducing the computational cost of the histograms computation in the S-HOG. According to the edge orientation, weighted contributions are provided to the corresponding bin and the closest one. This results on a costly interpolation process whose impact is evaluated. Column NCCG in Table II shows the classification results achieved when interpolation is removed: each gradient contributes only to the bin that corresponds to its orientation. Results are stable, improving for most configurations while slightly underperforming for the others. The non-interpolation causes better accuracy values for higher values of  $\beta$ , and it has a significant impact on computational saving.

##### C. Normalization

As proposed in [9], the use of overlapped 2x2 blocks is typically assumed [14], resulting on a new histogram generation and longer feature vectors. Using lower size configurations speed up the computation of the descriptor so 2x1 and 1x1 blocks are explored. Table II shows that for 1x1 blocks, the performance of the descriptor is still maintained, not requiring the computation of a second histogram, with significantly better accuracy than S-HOG. Therefore, the descriptor with the proposed modifications improves the performance with respect to S-HOG while significantly reducing the computational cost. In average (i.e. computing the average accuracy of the four classifiers

for each  $\beta$ ), there is a 0.57% and 1.3% gain for  $\beta=8$  and 16 respectively. As a conclusion, the values of  $\eta$  and  $\beta$  chosen to provide a good balance between performance and computational cost are  $\eta=4$  and  $\beta=16$ .

#### 5. INFLUENCE OF HOG CELLS

In this section we carried out a study to analyze the impact of the different cells of the HOG descriptor in the classification. The goal is to eliminate cells that provide less or misleading information for classification, thus reducing both, the length of the descriptor – faster classification - and the cost of feature extraction.

Figure 3 shows for each view, the evolution of the verification accuracy when cells are removed from the descriptor (left tables), and the corresponding topology of the removed cells (right images, grey colored). Iterative experiments were conducted. In each iteration, the cell that has the lowest impact on the results is identified and eliminated from the feature vector. Therefore, the number of cells (NC column in the left tables) composing the descriptor is reduced, removing the cell with the number indicated in the DC column. The last column of the tables ( $\Delta\%$ ) shows the accuracy deviation with respect to the performance of the entire descriptor.

The first conclusion is that for all views, removing the less significant cells do improve verification. Deviations are positive, with a gain in classification which is less significant for the frontal view (as expected) than for the more complex lateral and far views. Therefore, with a much smaller histogram, a good discrimination is still reached, improving of performance through the elimination of cells.

Another relevant conclusion is that the topology of the cells having negative impact in verification is related to the vehicle view particularities. Most important cells for the frontal view (Figure 3 (a)) are the outer ones, as they typically hold the main edges of vehicle rear when frontally observed, and thus provide more discriminating information for the system.

For the left view (Figure 3 (b)), most of the relevant information is within and below the right to left diagonal cells, which agrees with the image areas likely holding vehicle information when observed from its right. Similar conclusions can be obtained from the cells distribution for the right view, but considering the left to right diagonal. Finally, for the far view images (Figure 3 (d)), which show

Frontal View				Left View				Right View				Far View			
A(%)	NC	DC	$\Delta$ (%)	A(%)	NC	DC	$\Delta$ (%)	A(%)	NC	DC	$\Delta$ (%)	A(%)	NC	DC	$\Delta$ (%)
99.56	16	-	-	98.86	16	-	-	98.22	16	-	-	97.98	16	-	-
99.78	15	3	+0.22	99.12	15	5	+0.26	98.38	15	4	+0.16	98.32	15	2	+0.34
99.62	14	14	+0.08	99.16	14	2	+0.30	98.48	14	15	+0.26	98.28	14	3	+0.30
99.70	13	11	+0.14	99.08	13	9	+0.22	98.34	13	8	+0.12	98.36	13	6	+0.38
99.68	12	2	+0.12	99.20	12	6	+0.36	98.50	12	3	+0.28	98.20	12	10	+0.22
99.70	11	15	+0.14	99.26	11	11	+0.40	98.44	11	5	+0.22	98.16	11	4	+0.18
99.62	10	10	+0.08	99.14	10	15	+0.28	98.36	10	7	+0.14	98.34	10	5	+0.36
99.56	9	12	0	99.06	9	1	+0.20	98.30	9	10	+0.08	98.34	9	11	+0.36
99.48	8	9	-0.08	98.90	8	3	+0.04	98.38	8	13	+0.16	98.36	8	1	+0.38

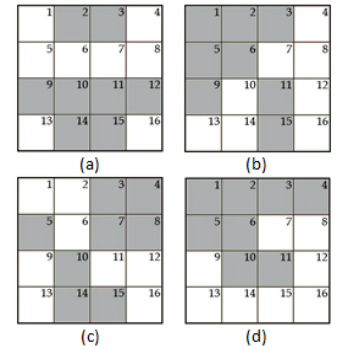


Figure 3: For the different views, classification accuracy evolution removing less significant cells from the descriptor

smoothed edges due to interpolation (images are scaled to 64x64 pixels), the lower cells are more discriminative as they hold more stable and contrasted edge information.

In conclusion, the study of the influence of the cells in the HOG descriptor helps us to understand which parts of the image are more decisive for the classification. In addition, we can assure that the use of full HOG descriptors, which conveys a high computational cost, is unnecessary. Using a part of the HOG descriptor is sufficient for the correct classification, greatly reducing the computational cost.

## 6. DISCUSSION

Firstly in this work changes in processing and parameters of the S-HOG descriptor are proposed: modifying the gradient extraction phase improves performance while maintaining the computational cost, and simplifying the histogram generation and normalization steps manage to reduce the computational cost while marginally affecting verification. However, motivated from the conclusions of the previous section, view-dependent cell configurations are here proposed which result in faster to compute and shorter HOG descriptors. Figure 4 shows the HOG cells, in white, considered for each view.

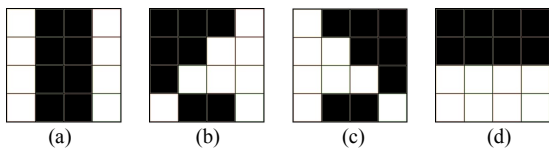


Figure 4. HOG cells – in white - considered for (a) frontal view; (b) left view; (c) right view; (d) far view.

As already mentioned, in the middle close frontal view the most significant cells are those corresponding to the sides of the vehicle, so it was decided to use these cells only for classification. Regarding the left and right views the same analysis was made, and the cells in and below each view diagonal are kept, removing also those that typically hold shadows cast by the vehicles. For the far view only the lower-half cells are considered: edges corresponding to vehicle wheels, lights or underneath shadows are here more important.

S-HOG		Our method		V-HOG[14]	
CF	99.48	CF	99.40	CF	97.68
L	97.64	L	98.96	L	97.02
R	96.22	R	98.14	R	95.54
F	97.76	F	98.24	F	95.60

Table III: Comparison of accuracy (%) between different methods

Table III compares the performance of S-HOG, our view dependent HOG descriptors and V-HOG [14]. Our method outperforms S-HOG at a much smaller computational cost: no interpolation and no multi-cell normalization are applied, halving the number of cells to compute. Furthermore, in terms of descriptor length, the average 360 components required for S-HOG go down to 128 with our proposal. In [14], an efficient HOG descriptor using only vertical cells, V-HOG, is proposed to carry out view dependent vehicle verification. Compared with the best results achieved with V-HOG, our proposal also largely outperforms for all views with a similar cost. In terms of average verification rate, i.e. averaging the accuracy for all views, our proposal reaches 98.69%, an excellent score that largely outperforms the results reported in [13] (92.9%), and [16] (94%), that proposes a cascade of boosted classifiers combining Haar-like and HOG features.

## 7. CONCLUSIONS

In this work, the adaptation of the HOG descriptor for real time vehicle verification is proposed. An adequate configuration of the descriptor and the simplification of some processing steps demonstrate improved computational efficiency bounding the loss in verification accuracy. A study of the influence of cells in classification is carried out, showing that a significant number of the cells do not contribute positively to verification, and that their spatial configuration relates to the view of the vehicles to be verified. This information has been used to propose new view-dependent HOG cells configurations which provide a suitable balance between performance and computational cost, rendering higher verification rates than other works in the literature.

## 11. REFERENCES

- [1] Junyeon Hwang, Kunsoo Huh, and Donghwi Lee. Vision based vehicle detection and tracking algorithm design. *Optical Engineering*, 48(12):127201–127201–10, 2009.
- [2] Gwang Yul Song, Ki Yong Lee, and Joon Woong Lee. Vehicle detection by edge-based candidate generation and appearance-based classification. In *Intelligent Vehicles Symposium*, 2008 IEEE, pages 428–433, 2008.
- [3] F. Woelk and R. Koch. Robust monocular detection of independent motion by a moving observer. In *Proceedings of the 26th DAGM Symposium*, ser. *Lecture Notes in Computer Science*, pages 27–35.
- [4] J. Arróspide, L. Salgado, M. Nieto and R. Mohedano, "Homography based ground plane detection using single on-board camera", *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 149–160, 2010.
- [5] F. Oniga and S. Nedeveschi. Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection. *Vehicular Technology, IEEE Transactions on*, 59(3):1172–1182, 2010.
- [6] Sivaraman Sayanan and Trivedi, Mohan Manubhai, "A General Active-Learning Framework for On-Road Vehicle Recognition and Tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, pp. 267-276, 2010.
- [7] J. Zhou, D. Gao and D. Zhang, "Moving Vehicle Detection for Automatic Traffic Monitoring," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 51-59, 2007.
- [8] S. Zehang, G. Bebis and R. Miller, "On-road vehicle detection using evolutionary Gabor filter optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, pp. 125-137, 2005.
- [9] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 886–893, Apr. 2005.
- [10] O. Daniz and G. Bueno and J. Salido and F. De la Torre, "Face recognition using Histograms of Oriented Gradients," *Pattern Recognition Letters*, vol. 32, pp. 1598 - 1603, 2011.
- [11] Pengpeng Liang and Gregory Teodoro and Haibin Ling and Erik Blasch and Genshe Chen and Li Bai, "Multiple Kernel Learning for Vehicle Detection in Wide Area Motion Imagery," *Information Fusion (FUSION), 2012 15th International Conference on*, pp. 1629-1636, 2012.
- [12] L. Mao, M. Xie, Y. Huang, and Y. Zhang, "Preceding vehicle detection using Histograms of Oriented Gradients," in *Proc. International Conference on Communications, Circuits and Systems*, 2010, pp. 354–358.
- [13] D. Balcones, D. F. Llorca, M. A. Sotelo, M. Gavilán, S. Álvarez, I. Parra, and M. Ocaña, "Real-time vision-based vehicle detection for rear-end Collision mitigation systems," in *Computer Aided Systems Theory - EUROCAST, LNCS*, 2009, vol. 5717, pp. 320–325.
- [14] Jon Arrospide, Luis Salgado, and Massimo Camplani, "Image-based on-road vehicle detection using cost-effective Histograms of Oriented Gradients," *Journal of Visual Communication and Image Representation*, vol. 24, pp. 1182 - 1190, 2013.
- [15] Seonyoung Lee and Haengseon Son and Jong Chan Choi and Kyoungwon Min, "HOG feature extractor circuit for real-time human and vehicle detection," *ENCON 2012 - 2012 IEEE Region 10 Conference*, pp. 1-5, 2012.
- [16] Pablo Negri, Xavier Clady, Shehzad Muhammad Hanif, and Lionel Prevost. A cascade of boosted generative and discriminative classifiers for vehicle detection. *EURASIP Journal on Advances in Signal Processing*, 2008(1):782432.
- [17] Xianbin Cao and Changxia Wu and Pingkun Yan and Xuelong Li, "Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos," *Image Processing (ICIP), 2011 18th IEEE International Conference on*, 2011.
- [18] Yang Lv, Benjamin Yao, Yongtian Wang, and Song-Chun Zhu, "Reconfigurable Templates for Robust Vehicle Detection and Classification," *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, pp. 321-328, 2012.
- [19] GTI Vehicle Image Database, Image Processing Group at UPM, <<http://www.gti.ssr.upm.es/data>>, 2011.
- [20] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Science+Business Media, 2006.

Dear Mr. Gonzalo Ballesteros:

Congratulations! We are pleased to inform you that your manuscript 'Histograms of Oriented Gradients for fast on-board vehicle verification' has been accepted for presentation at ICIP 2014.

Over the past few months, the Technical Program Committee has worked hard to thoroughly review each submission to ICIP 2014. Reviewers were carefully selected and assigned papers for review in their areas of expertise. Most valid papers received at least three reviews, and most were assigned at least one highly experienced reviewer.

The reviewers' comments on your paper can be found at the end of this email, or can be found at <http://edas.info/showPaper.php?m=1569911281> (using your EDAS login name [gonzalo.ballesteros@estudiante.uam.es](mailto:gonzalo.ballesteros@estudiante.uam.es)), where you can also rate the reviewers' feedbacks. If appropriate, an additional feedback from the Technical Program Committee (TPC) is also included after the reviewers' comments. These comments served as part of the basis for the TPC's decision on your manuscript. In many cases, the reviewer comments and TCP feedback offer important suggestions for strengthening the final paper to be published in the Proceedings of ICIP 2014, which has to be uploaded between May 28, 2014 and June 8, 2014.

Full information regarding ICIP 2014 is posted on the conference web site (<http://www.icip2014.org/>) which will list the complete technical program as it becomes available.

CRITICAL INFORMATION:

1) Included at the end of this email are the reviewers' comments on your paper. You may make minor revisions to your paper in light of these comments prior to the upload of your final, publication-ready manuscript. However, please note that you CANNOT make substantive text changes to the body of the original reviewed paper. We reserve the option to use the accepted version of your paper if we feel that the updated version differs substantially from what the reviewers accepted. In any event, the revised paper has a maximum length of 4 pages, plus an optional fifth page for references only, and must follow the format detailed in the Paper Kit (<http://www.icip2014.org/paper-kit/>).

2) Your final, publication-ready manuscript MUST be uploaded between May 28, 2014 and June 8, 2014, in the form of a PDF that is fully compliant with IEEE Xplore. In order to ensure Xplore-compliance, you must use IEEE PDF eXpress to either (1) convert your manuscript from its original source form into an Xplore-compliant PDF, or (2) check your manuscript's PDF to verify compliance with IEEE Xplore. Follow the instructions at <http://www.icip2014.org/upload-final-manuscript/>

3) You MUST upload your final Xplore-compatible PDF to EDAS. The upload process will involve completion of an online copyright form; every manuscript MUST have a completed copyright form. Again, follow the instructions at <http://www.icip2014.org/upload-final-manuscript/>

4) At least one author MUST BE REGISTERED for the conference at a non-student rate by June 15, 2014. Failure to have such registration will result in removal of the paper from the Technical Program. A single, full registration may cover up to three (3) papers. When registering, please enter the ID number(s) of the paper(s) that you will be covering with your full registration. Register at <http://www.icip2014.org/registration/>

5) Note that papers which are not presented at the conference will be excluded from the official proceedings and will not appear on IEEE Xplore. It is therefore very important that at least one author of the paper attend the conference to present the work.

The IEEE Signal Processing Society (SPS) is pleased to sponsor travel grants to student and non-student authors from developing countries. The grants will

be awarded on a highly competitive basis to a limited number of applicants who have a genuine need for financial support to attend ICIP 2014. SPS membership is required, and the paper quality will be taken into consideration. The grants may only be used to cover the travel expenses to attend ICIP 2014. The grants, if approved, will be distributed via a reimbursement during or shortly after the conference. The SPS travel grant selection and oversight committee reserves the right to make the final decision regarding all applications. For more details, please visit the SPS Travel Grants Page of the IEEE Signal Processing Society (<http://www.signalprocessingsociety.org/membership/sps-travel-grants/>), or email [students@icip2014.org](mailto:students@icip2014.org).

If you are traveling from outside France, you may need to apply for a VISA to come to ICIP 2014. See <http://www.icip2014.org/registration/> for information on obtaining a letter of invitation.

Congratulations on the acceptance of your paper at ICIP 2014!  
We look forward to welcoming you to Paris in October.

Sincerely,

Pascal Frossard and Marc Antonini  
ICIP 2014 Technical Program Chairs

# Apéndice C

## Presupuesto

### 1. Ejecucion Material

- Compra de ordenador personal (Software incluido) ..... 2.000 €
- Alquiler de impresora láser durante 6 meses ..... 260 €
- Material de oficina ..... 150 €
- Total de ejecución material ..... 2.400 €

### 2. Gastos generales

- 16 % sobre Ejecucion Material ..... 352 €

### 3. Beneficio Industrial

- 6 % sobre Ejecucion Material ..... 132 €

### 4. Honorarios Proyecto

- 1800 horas a 15 €/ hora ..... 27.000 €

### 5. Material fungible

- Gastos de impresión ..... 280 €
- Encuadernación ..... 200 €

### 6. Subtotal del presupuesto

- Subtotal Presupuesto ..... 32.774 €

### 7. I.V.A. aplicable



- 21 % Subtotal Presupuesto ..... 6.882,54 €

**8. Total presupuesto**

---

- Total Presupuesto ..... 39.656,54 €

Madrid, Julio de 2014

El Ingeniero Jefe de Proyecto

Fdo.: Gonzalo Ballesteros Villarreal

Ingeniero Superior de Telecomunicación

# Apéndice D

## Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de “VERIFICACIÓN DE VEHÍCULOS MEDIANTE TÉCNICAS DE VISIÓN ARTIFICIAL” para ser visto en pantallas de baja resolución. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

### Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.
2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.
3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.
4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.
6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.
7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.
8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.
9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.
10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.
11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.
13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.
14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.
15. La garantía definitiva será del 4 % del presupuesto y la provisional del 2 %.
16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.
17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.
18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.
19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.
20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.
21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.
23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

### **Condiciones particulares**

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.
2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.
3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.
4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.
5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.
6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.
8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.
9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.
10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.
11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.
12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.

