



ANÁLISIS DE MOVIMIENTO INTRÍNSECO DE VEHÍCULOS EN SECUENCIA DE IMÁGENES (AMIVSI)

Autores:

Gerardo Abal Caldelas
Miguel Prieto Morán

Director del Proyecto:

Mario Mata Ortega

Proyecto Fin de Carrera, 2004 - 2005

**Ingeniería Informática
Universidad Europea de Madrid**

AGRADECIMIENTOS Y DEDICATORIAS

Este trabajo está dedicado a nuestras familias y a Maite por apoyarnos incondicionalmente.

Quisiéramos agradecer a Mario Mata, director del proyecto, su confianza y consejos en el desarollo de todo el trabajo realizado.

A todos nuestros amigos por su apoyo y paciencia. Gracias Javier, Agustín, Roberto, José, Esther, Juan, Pepe, Irina.

ÍNDICE DE CAPÍTULOS Y APÉNCIDES

Capítulo I.	Introducción.....	1
Capítulo II.	Antecedentes y estado actual del tema	3
Capítulo III.	Descripción del problema y objetivos	7
Capítulo IV.	Descripción de la solución propuesta	10
1	Introducción.....	10
2	Especificación de requisitos de software.....	11
2.1	Introducción.....	11
2.1.1	Propósito.....	11
2.1.2	Ámbito.....	11
2.1.3	Definiciones, siglas y abreviaciones.....	13
2.1.4	Apreciación global.....	15
2.2	Descripción global.....	16
2.2.1	Perspectiva del producto.....	16
1)	Interfaces del sistema.....	17
a)	Interfaces de entrada.....	17
b)	Interfaces de salida	19
2)	Interfaces con el usuario	20
a)	Interfaz de configuración.....	20
b)	Interfaz gráfico	21
3)	Interfaces con el Hardware	22
4)	Interfaces con el Software	22
a)	Nombre: Sistema Operativo	23
b)	Nombre: Matrox Imaging Library.....	23
c)	Nombre: Sistema SACAT	23
5)	Interfaces de comunicaciones.....	23
6)	Restricciones de memoria.....	23
7)	Funcionamientos.....	24

8) Requisitos de adaptación al sistema.....	24
2.2.2 Funciones del producto	24
2.2.3 Características del usuario	25
2.2.4 Restricciones	25
2.2.5 Supuesto y dependencias	26
2.3 Requisitos específicos	26
2.3.1 Interfaces externas.....	26
2.3.2 Requisitos Funcionales	27
1) Velocidad	27
2) Profundidad.....	27
3) Etiquetado	27
2.3.3 Requisitos de rendimiento.....	28
2.3.4 Requisitos del banco de datos lógicos.....	28
2.3.5 Aceptación de las normas	29
2.3.6 Atributos del software del sistema.....	30
2.3.7 Fiabilidad	30
2.3.8 Seguridad	30
2.3.9 Mantenimiento	30
2.3.10 Portabilidad	31
2.3.11 Organizar los requisitos específicos.....	31
2.3.12 Modo del sistema	31
2.3.13 Clases de usuarios	31
2.3.14 Objetos	32
2.3.15 Característica / Rasgo.....	32
2.3.16 Estímulo	32
2.3.17 Contestación.....	32
1) Análisis y generación del Optical Flow	32
2) Obtención de la velocidad global del objetivo.....	32
3) Cálculo del espacio libre, mapa de profundidades.....	33
4) Predicción de la posición de los objetos reconocidos en futuras imágenes	33

2.3.18 Jerarquía Funcional	33
3 Casos de Uso	34
3.1 Diagrama de Casos de Uso.....	35
3.2 Caso de Uso Iniciar Sistema AMIVSI.....	36
3.3 Caso de Uso Leer imagen de entrada	37
3.4 Caso de Uso Calcular Optical Flow	38
3.5 Caso de Uso Calcular velocidad.....	39
3.6 Caso de Uso Calcular Profundidad.....	40
3.7 Caso de Uso Estimar Posición de un Objeto	41
3.8 Caso de Uso Etiquetar un objeto	42
3.9 Caso de Uso Refrescar interfaz	43
4 Diseño detallado de Clases	44
4.1 Enumeración de clases.....	44
4.2 Diagrama de Clases	45
4.3 Clase OpticalFlow	46
4.4 Clase TMatriz	49
4.5 Clase TListaEtiquetas	51
4.6 Clase Navegación	53
4.7 Clase ToolPaint	54
4.8 Clase Etiquetado	56
4.9 Clase Secuenciador.....	58
4.10 Clase TMainForm.....	59
4.11 Clase TImagen	60
4.12 Clase TBaseMil	62
5 Diagrama de Secuencia	63
6 Diseño y análisis de módulos del sistema	65
6.1 Introducción.....	65
6.2 Generación de secuencia de imágenes.....	66
6.3 Diagrama de módulos del sistema	68
6.4 Cálculo del Optical Flow	69

6.4.1	Primera Fase: Adquisición de Imágenes	69
6.4.2	Segunda Fase: Técnicas de Filtrado.....	70
1)	Filtrado de Bordes	70
2)	Selección del punto de análisis	72
3)	Creación de una rejilla uniforme.....	72
4)	Discriminación de puntos entorno a un umbral	73
6.4.3	Tercera Fase: Procesado y Almacenamiento	76
1)	Crear Patrón de Búsqueda.....	76
2)	PatternMatching	79
3)	Evaluar Resultado	80
4)	Almacenar en matriz	80
6.5	Estimar el vector dirección	81
6.6	Estimación de la velocidad de movimiento	82
6.6.1	Zona de análisis.....	82
6.6.2	Calibración de la zona de análisis	83
6.6.3	Tratamiento del Optical Flow	85
6.6.4	Uso del Optical Flow	86
6.6.5	Cálculo de la velocidad instantánea	87
6.6.6	Cálculo de la velocidad promediada	88
6.7	Estimación del Mapa de Profundidades.....	89
6.7.1	Depth from motion: Con respecto a la posición de un punto y su centro óptico	89
6.8	Seguimiento y Etiquetado de Objetos	93
6.8.1	Lectura de objetos	93
6.8.2	Estimación de la posición futura.....	94
6.8.3	Asignar etiqueta a un objeto.....	95
6.8.4	Generar el archivo de registro de etiquetas	96
6.9	Análisis de Tiempo Real	97
6.10	Análisis de Estructuras de Datos.....	98
6.10.1	Matriz Optical Flow	98

6.10.2	Lista de Etiquetas	99
6.11	Alcance y limitaciones generales	99
6.11.1	Optical Flow	99
6.11.2	Estimación de la velocidad de movimiento.....	100
6.11.3	Estimación del mapa de profundidades	101
6.11.4	Seguimiento y etiquetado de objetos	101
6.12	Técnicas Alternativas	102
6.12.1	Filtrado Prewitt.....	102
6.12.2	Umbralizado + Adelgazamiento.....	102
6.12.3	Incremento de alturas	103
6.12.4	Calculo del Foco Óptico de Expansión	106
6.12.5	Variación del umbral	108
6.13	Tabla de Tareas, Técnicas y Estructuras de Datos	108
7	Planificación y Presupuesto.....	109
7.1.1	Distribución de tareas	109
7.1.2	Presupuesto por Recursos.....	111
7.1.3	Presupuesto por Fases.....	112
7.1.4	Resumen	112
Capítulo V.	Plan de Verificación	114
1	Plan de Pruebas.....	114
2	Ítems Probados	115
3	Características Probadas	116
4	Características No Probadas	118
5	Especificación de procedimientos de prueba.....	119
5.1	Procedimiento Unitario: Hay_Vector.....	119
5.2	Procedimiento Unitario: Crear_Matriz.....	120
5.3	Procedimiento Unitario: Buscar_Patron.....	122
5.4	Procedimiento Unitario: Calcular_Optical_Flow.....	123
5.5	Procedimiento Unitario: Calcular_Velocidad	125
5.6	Procedimiento Unitario: Calcular_Ventana_Búsqueda.....	126

5.7	Procedimiento Unitario: Calcular_Posicion_Búsqueda.....	127
5.8	Procedimiento Unitario: Calcular_Profundidad.....	128
5.9	Procedimiento Unitario: Pintar_Optical_Flow	129
5.10	Procedimiento Unitario: Pintar_Vector	130
5.11	Procedimiento Unitario: Pintar_Velocidad_Analógica	131
5.12	Procedimiento Unitario: Calcular_Posición_Futura	132
5.13	Procedimiento Unitario: Asignar_Etiqueta.....	134
5.14	Procedimiento Integración: Resolución_de_las_imágenes.....	135
5.15	Procedimiento Integración: Fichero_de_configuración.....	136
5.16	Procedimiento Sistema: Combinación_de_escena_de_carretera.....	138
5.17	Procedimiento Sistema: Escena_de_carretera_velocidad.....	139
5.18	Procedimiento Sistema: Escena_de_carretera_profundidad.....	140
5.19	Procedimiento Sistema: Escena_de_carretera_etiquetado.....	141
Capítulo VI. Evaluación de los resultados obtenidos		142
1	Introducción	142
1.1	Escena_de_carretera_velocidad.....	145
1.2	Escena_de_carretera_profundidad	145
1.3	Escena_de_carretera_etiquetado.....	145
1.4	Combinación_de_escena_de_carretera.....	145
2	Pruebas de Velocidad con Secuencias Cortas.....	147
2.1	Primera escena_de_carretera_velocidad	147
2.2	Segunda Escena_de_carretera_velocidad	149
2.3	Tercera Escena_de_carretera_velocidad.....	151
2.4	Evaluación de los tres procedimientos Escena_de_carretera_velocidad	154
3	Pruebas de Profundidad con Secuencias Cortas	155
3.1	Primera Escena_de_carretera_profundidad	155
3.2	Segunda Escena_de_carretera_profundidad	157
3.3	Tercera Escena_de_carretera_profundidad.....	159
3.4	Evaluación de los tres procedimientos Escena_de_carretera_profundidad	161
4	Pruebas de Etiquetado con Secuencias Cortas.....	163

4.1	Primera Escena_de_carretera_etiquetado.....	163
4.2	Segunda Escena_de_carretera_etiquetado.....	163
4.3	Tercera Escena_de_carretera_etiquetado	166
4.4	Evaluación de los tres procedimientos Escena_de_carretera_etiquetado.....	167
5	Primera Combinación_de_escena_de_carretera.....	169
5.1	Evaluación de velocidad en primera secuencia global	169
5.2	Evaluación de profundidad en primera secuencia global	172
5.3	Evaluación del etiquetado en primera secuencia global.....	174
6	Segunda Combinación_de_escena_de_carretera.....	175
6.1	Evaluación de velocidad en segunda secuencia global	175
6.2	Evaluación de profundidad en segunda secuencia global	177
6.3	Evaluación del etiquetado en segunda secuencia global	179
7	Informe Global de Resultados Obtenidos.....	181
Capítulo VII. Conclusiones y Líneas Futuras		184
APENDICE I: Bibliografía.....		186
APENDICE II: Diario de Experimentos		189
APENDICE III: Manual de Usuario.....		196
APENDICE IV: Guía de Solucionado de Errores		206

ÍNDICE DE FIGURAS

Figura II-1 Etapas del procesamiento digital de imágenes	4
Figura IV-1 Diagrama de ínter actuación con el sistema SACAT.....	17
Figura IV-2 Formato del archivo de lista de objetos reconocidos	18
Figura IV-3 Formato del archivo de registro de etiquetas	19
Figura IV-4 Fichero de configuración	21
Figura IV-5 Interfaz preliminar del sistema AMIVSI.....	22
Figura IV-6 Diagrama de Casos de Uso.	35
Figura IV-7 Diagrama de clases.....	45
Figura IV-8 Diseño de clase OpticalFlow.....	46
Figura IV-9 Diseño de clase TMatriz	49
Figura IV-10 Diseño de clase TListaEtiquetas	51
Figura IV-11 Diseño de clase Navegación	53
Figura IV-12 Diseño de clase ToolPaint.....	54
Figura IV-13 Diseño de clase Etiquetado	56
Figura IV-14 Diseño de clase Secuenciador	58
Figura IV-15 Diseño de clase TForm	59
Figura IV-16 Diseño de clase TImagen	60
Figura IV-17 Diseño de clase TBaseMil.....	62
Figura IV-18 Diagrama de secuencia del sistema AMIVSI	63
Figura IV-19 Diferencias entre entrelazado y desentrelazado.	67
Figura IV-20 Diagrama de módulos del sistema	68
Figura IV-21 Máscaras de convolución SOBEL.	70
Figura IV-22 Imagen original sobre la que se aplicará un filtro SOBEL	71
Figura IV-23 Imagen tras aplicar un filtro SOBEL	71
Figura IV-24 Superposición de imagen con rejilla uniforme.	73
Figura IV-25 Imagen original antes de realizarle una umbralización.....	74
Figura IV-26 Imagen tras aplicarle un umbral igual a 75.....	75

Figura IV-27 Distribución de puntos en zonas de interés mediante técnicas de filtrado	76
Figura IV-28 Representación de vectores de desplazamiento con respecto al FOE	78
Figura IV-29 Representación de la ventana de búsqueda.....	79
Figura IV-30 Representación del algoritmo de correlación de bloques	80
Figura IV-31 Optical Flow calculado sobre una secuencia (1)	81
Figura IV-32 Optical Flow calculado sobre una secuencia (2)	81
Figura IV-33 Zona de análisis de velocidad.....	83
Figura IV-34 Calibración del sistema.....	85
Figura IV-35 Variación del umbral para el cálculo de velocidad.....	86
Figura IV-36 Proceso de cálculo de velocidad instantánea.....	88
Figura IV-37 Cono de visión de un punto respecto de un eje óptico de movimiento.....	89
Figura IV-38 Mapa de profundidades estimadas para un Optical Flow.....	91
Figura IV-39 Imagen del mapa de Profundidades sin imagen.	92
Figura IV-40 Escala de profundidades	92
Figura IV-41 Identificación de objetos por medio de archivos de coordenadas.....	94
Figura IV-42 Estimación de la posición futura de un objeto.....	95
Figura IV-43 Asignación de etiqueta	96
Figura IV-44 Archivo de registro de etiquetas generado durante el análisis.....	97
Figura IV-45 Problema de la apertura	100
Figura IV-46 Imagen umbralizada y esqueletizada.....	103
Figura IV-47 Técnica del incremento de alturas.	104
Figura IV-48 Filtrado de vectores lineales con dos Optical Flow consecutivos.	105
Figura IV-49 Calculo de distancia a una altura determinada.	106
Figura IV-50 Calculo del foco óptico de expansión.....	107
Figura VI-1 Mediana del un histograma.....	143
Figura VI-2 Gráfico de luminosidades.	143
Figura VI-3 Gráfico de Puntos de análisis.	144
Figura VI-4 Comparativa velocidad secuencia 30-1.avi.	148
Figura VI-5 Características de la secuencia 30-1.avi.	148
Figura VI-6 Comparativa velocidad secuencia 40-1.avi.	150

ÍNDICE DE FIGURAS

Figura VI-7 Características de la secuencia 40-1.avi.....	151
Figura VI-8 Comparativa velocidad secuencia 10-1.avi.....	153
Figura VI-9 Características de la secuencia 10-1.avi.....	153
Figura VI-10 Características y porcentaje de acierto de las secuencias de velocidad.	154
Figura VI-11 Comparativa velocidad secuencia 30-6.avi.....	156
Figura VI-12 Características de la secuencia 30-1.avi.....	156
Figura VI-13 Comparativa velocidad secuencia 10-Persona1.avi.....	158
Figura VI-14 Características de la secuencia 10-Persona1.avi.....	158
Figura VI-15 Comparativa velocidad secuencia 50-2.avi.....	160
Figura VI-16 Características de la secuencia 50-2.avi.....	160
Figura VI-17 Características y porcentaje de acierto de las secuencias de profundidad.	161
Figura VI-18 Comparativa velocidad secuencia bosque.avi.....	171
Figura VI-19 Características de la secuencia bosque.avi en los frames analizados.	171
Figura VI-20 Comparativa profundidad secuencia bosque.avi.....	173
Figura VI-21 Características de la secuencia bosque.avi en los frames analizados.	173
Figura VI-22 Comparativa velocidad secuencia zarzal.avi.	176
Figura VI-23 Características de la secuencia zarzal.avi en los frames analizados.	177
Figura VI-24 Comparativa profundidad secuencia zarzal.avi.	178
Figura VI-25 Características de la secuencia zarzal.avi en los frames analizados.	179
Figura VI-26 Porcentaje de acierto de velocidad dividido en rangos.	181
Figura VI-27 Porcentajes de acierto de la profundidad según luminosidad y velocidad.....	182
Figura VI-28 Porcentaje de acierto del etiquetado según movimiento.....	182

ÍNDICE DE TABLAS

Tabla IV-1 Caso de Uso Iniciar Sistema AMIVSI.....	36
Tabla IV-2 Caso de Uso Leer imagen de entrada.....	37
Tabla IV-3 Caso de Uso Calcular Optical Flow.....	38
Tabla IV-4 Caso de Uso Calcular velocidad.....	39
Tabla IV-5 Caso de Uso Calcular Profundidad.....	40
Tabla IV-6 Caso de Uso Estimar Posición de un Objeto	41
Tabla IV-7 Caso de Uso Etiquetar un objeto.....	42
Tabla IV-8 Caso de Uso Refrescar Interfaz.....	43
Tabla IV-9 Combinación de tareas, técnicas y estructura de datos.....	108
Tabla IV-10 Duración de fases y tareas del proyecto.....	111
Tabla IV-11 Presupuesto de recursos.....	112
Tabla IV-12 Presupuesto por fases del proyecto.....	112
Tabla IV-13 Presupuesto global del proyecto.....	113
Tabla V-1 Procedimiento Unitario: Hay_Vector.....	119
Tabla V-2 Procedimiento Unitario: Crear_Matriz.....	120
Tabla V-3 Procedimiento Unitario: Buscar_Patron.....	122
Tabla V-4 Procedimiento Unitario: Calcular_Optical_Flow.....	123
Tabla V-5 Procedimiento Unitario: Calcular_Velocidad.....	125
Tabla V-6 Procedimiento Unitario: Calcular_Ventana_Búsqueda	126
Tabla V-7 Procedimiento Unitario: Calcular_Posición_Búsqueda	127
Tabla V-8 Procedimiento Unitario: Calcular_Profundidad.....	128
Tabla V-9 Procedimiento Unitario: Pintar_Optical_Flow	129
Tabla V-10 Procedimiento Unitario: Pintar_Vector.....	130
Tabla V-11 Procedimiento Unitario: Pintar_Velocidad_Analógica.....	131
Tabla V-12 Procedimiento Unitario: Calcular_Posición_Futúra	132
Tabla V-13 Procedimiento Unitario: Asignar_Etiqueta.....	134
Tabla V-14 Procedimiento Integración: Resolución_de_las_imágenes.....	135

Tabla V-15 Procedimiento Integración: Fichero_de_configuración	137
Tabla V-16 Procedimiento Sistema: Combinación_de_escena_de_carretera..	138
Tabla V-17 Procedimiento Sistema: Escena_de_carretera_velocidad.....	139
Tabla V-18 Procedimiento Sistema: Escena_de_carretera_profundidad.....	140
Tabla V-19 Procedimiento Sistema: Escena_de_carretera_etiquetado.	141
Tabla VI-1 Resultados 1º Escena_de_carretera_velocidad.....	147
Tabla VI-3 Resultados 2º Escena_de_carretera_velocidad.....	150
Tabla VI-4 Resultados 3º Escena_de_carretera_velocidad.....	152
Tabla VI-5 Resumen procedimientos Escena_de_carretera_velocidad.....	154
Tabla VI-6 Resultados 1º Escena_de_carretera_profundidad.....	155
Tabla VI-7 Resultados 2º Escena_de_carretera_profundidad.....	157
Tabla VI-8 Resultados 3º Escena_de_carretera_profundidad.....	159
Tabla VI-9 Resumen procedimientos Escena_de_carretera_velocidad.....	161
Tabla VI-10 Resultados 1ª Escena_de_carretera_etiquetado.....	163
Tabla VI-11 Resultados 1ª señal, 2ª Escena_de_carretera_etiquetado.	164
Tabla VI-12 Resultados 2ª señal, 2ª Escena_de_carretera_etiquetado.	164
Tabla VI-13 Resultados 3ª señal, 2ª Escena_de_carretera_etiquetado.	164
Tabla VI-14 Resultados 4ª señal, 2ª Escena_de_carretera_etiquetado.	165
Tabla VI-15 Resultados 5ª señal, 2ª Escena_de_carretera_etiquetado.	165
Tabla VI-16 Resultados 6ª señal, 2ª Escena_de_carretera_etiquetado.	166
Tabla VI-17 Resultados 7ª señal, 2ª Escena_de_carretera_etiquetado.	166
Tabla VI-18 Resultados 1ª señal, 3ª Escena_de_carretera_etiquetado.	167
Tabla VI-19 Resultados 2ª señal, 3ª Escena_de_carretera_etiquetado.	167
Tabla VI-20 Influencia del movimiento en el seguimiento y etiquetado.....	168
Tabla VI-21 Resultados velocidad 1ª Combinación_de_escena_de_carretera.....	170
Tabla VI-22 Resultados profundidad 1ª Combinación_de_escena_de_carretera	172
Tabla VI-23 Resultados 1º señal, 1ª Combinación_de_escena_de_carretera	174
Tabla VI-24 Resultados 2º señal, 1ª Combinación_de_escena_de_carretera	174
Tabla VI-25 Resultados velocidad 2ª Combinación_de_escena_de_carretera	176
Tabla VI-26 Resultados profundidad 2ª Combinación_de_escena_de_carretera	178

Tabla VI-27 Resultados 1 ^a señal, 2 ^a Combinación_de_escena_de_carretera.....	180
Tabla VI-28 Resultados 2 ^a señal, 2 ^a Combinación_de_escena_de_carretera.....	180
Tabla VI-29 Resultados 3 ^a señal, 2 ^a Combinación_de_escena_de_carretera.....	180
Tabla VI-30 Fiabilidad de resultados proporcionados por el sistema AMIVSI.....	183

*“El hombre encuentra a Dios detrás de
cada puerta que la ciencia logra abrir”*

(Albert Einstein)

Capítulo I. Introducción

Con el presente trabajo se persigue el estudio y desarrollo de un sistema de análisis de movimiento basado en el campo de la visión artificial. El área de los estudios de visión artificial es un campo que siempre ha intrigado a los ingenieros. Sin embargo las grandes dificultades tecnológicas y computacionales han retrasado los avances en ella. Si bien la base teórica era conocida desde mitad del siglo XX, la gran complejidad asociada a los cálculos ha postergado los estudios en profundidad hasta la actualidad.

El área de la visión artificial se centra en la captura, filtrado y procesado de imágenes teniendo como objetivo último la interpretación del mundo real, tal y como lo haría un ser humano. Por lo tanto se establecen numerosos paralelismos con la visión humana, adquiriendo imágenes mediante cámaras digitales, el equivalente a los ojos, y analizando mediante un computador dichas imágenes para la adquisición de información del entorno, cuyo paralelismo sería el cerebro.

Unas de las tareas más interesantes en la que se centra el cerebro es la capacidad de interpretación de la profundidad. Desde tempranas edades aprendemos a movernos en un espacio tridimensional que a priori nos es desconocido. Puede decirse que el cerebro humano se vale de la visión estereoscópica y de la percepción de formas en movimiento para estimar distancias. Además, esta última cualidad dota al ser humano de la capacidad de determinar la velocidad y trayectoria de su propio movimiento.

La visión artificial trata de imitar este comportamiento, con el fin de automatizar actividades que involucren análisis de movimiento. Nuestros esfuerzos están encaminados a investigar y profundizar en las técnicas que la visión artificial proporciona para dicho fin.

Existen incontables aplicaciones destinadas a dar solución a problemas de la vida cotidiana. Nuestro enfoque va destinado al campo de la automoción que debido a sus características dinámicas ofrece una amplia gama de líneas de investigación.

La sociedad actual demanda soluciones de seguridad vial y los beneficios más importantes que proporciona la visión artificial son ayudas a la conducción, aumentando la seguridad y previniendo accidentes. Cabe señalar que, el poder aportar mejoras en este campo, supone una motivación añadida en nuestros estudios.

El análisis intrínseco del movimiento de un vehículo se centra en el estudio de su velocidad, dirección y profundidad. Para ello es necesario introducir una cámara en el interior del vehículo que capture imágenes frontales con un ángulo de visión similar al que dispone el conductor. Valiéndonos de las imágenes capturadas y de las técnicas de visión artificial, disponemos de toda la información y recursos necesarios para interpretar el movimiento intrínseco del vehículo.

La técnica fundamental en la que se basa nuestro proyecto para el análisis de movimiento es el estudio de campo de velocidades o flujo óptico, que será detallada en profundidad en los capítulos siguientes y no es objetivo de esta introducción, sin embargo consideramos relevante mencionar a sus precursores *Horn & Schunck* [HORN 1980] y *Lucas & Kanade* [LUCAS 1981], que introdujeron a principios de los ochenta las primeras hipótesis acerca de esta técnica.

Una de las restricciones a las que nos enfrentamos, es lograr cumplir las exigencias de un sistema en tiempo real. Los procesos de análisis son computacionalmente muy costosos, y siempre consumen el máximo disponible de los recursos del hardware que estemos utilizando. Para controlar el tiempo de cálculo, es preciso optimizar la asignación de tiempos en función de las cargas de trabajo.

Quizá con todo lo expuesto se puede pensar en la visión artificial como el gran paradigma de un futuro reciente. La situación actual invita a pensar que los resultados esperados a corto y medio plazo son esperanzadores, sin embargo somos conscientes del largo camino que queda por recorrer.

Capítulo II. Antecedentes y estado actual del tema

La visión artificial es una disciplina cuya finalidad es dotar del sentido de la vista a un ordenador con el objetivo de mejorar la forma de interactuar en ambientes complejos.

La visión artificial se remonta a los años cincuenta con la aparición de los primeros computadores. En su comienzo tuvieron unos resultados muy esperanzadores que produjeron unas expectativas demasiado ambiciosas para más tarde caer en una crisis debido a los escasos resultados obtenidos, por culpa de las deficiencias tecnológicas del momento.

La actividad se retomó con el desarrollo de la informática en los años ochenta, y se obtuvieron las primeras aplicaciones industriales de visión por computador.

El mundo real en el que se interactúa, es tridimensional y dinámico. Es necesario captarlo a través de cámaras digitales, e interpretar los datos obtenidos utilizando los principios de la geometría. Una característica importante de los sistemas actuales se basa en la comprensión espacial del mundo. Los verdaderos fundamentos de esta disciplina son el dominio de la relación espacial, la interpretación del espacio tridimensional, la propagación de la luz a través de una lente y el modelo matemático de formas y tamaños de objetos.

La metodología de representación visual ha estado presente desde la época de los clásicos griegos hasta nuestros días; se han formulado teorías sobre cómo percibe el ser humano su realidad exterior y de qué manera usa y representa la información captada por sus sentidos. Mas recientes son los estudios de Kepler sobre el análisis geométrico de la formación de la imagen en el ojo, o los de Newton sobre visión en color.

El procesamiento y análisis de imágenes digitales nacen en el momento en que se dispone de recursos técnicos para captar y manipular, en forma de matrices bidimensionales, gran cantidad de información espacial. El procesamiento digital de imágenes se divide en 5 etapas [WOODS 1996], desde la adquisición de las imágenes hasta la obtención de los resultados, la siguiente figura representa la interacción de las distintas etapas:

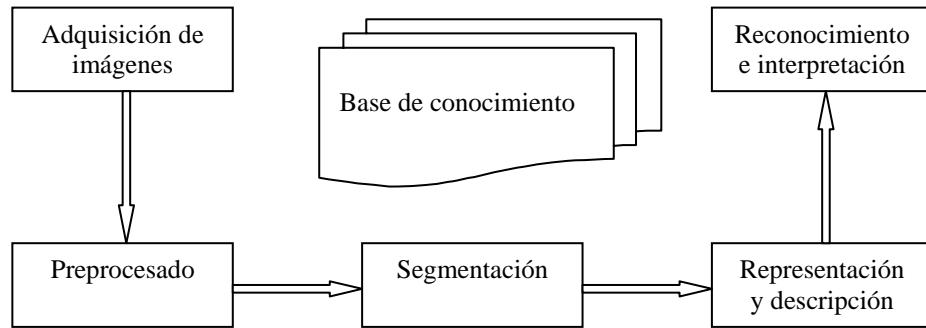


Figura II-1 Etapas del procesamiento digital de imágenes

La mejora de la calidad de la imagen deriva en el problema de la selección del método de impresión y la distribución de los niveles de gris. Los primeros sistemas de los años veinte, eran capaces de codificar cinco niveles de gris distintos. En 1929 ya se utilizaban quince. Lógicamente estábamos en lo que podría considerarse el comienzo de la formación, captación, muestreo, cuantificación, codificación y visualización de imágenes.

Las aplicaciones iniciales del campo de la visión artificial fueron [BRANCH 2001]:

- 1.- OCR (Optical Character Recognition): Reconocimiento de caracteres. Fundamental para los procesos de digitalización de bibliotecas.
- 2.- Medicina. Numerosas imágenes médicas eran examinadas por especialistas con el fin de clasificar cromosomas, contar células, detectar patologías, etc. El uso de técnicas tridimensionales para el análisis de imágenes radiológicas ha sido un gran logro.
- 3.- Teledetección. Desde los años sesenta una enorme cantidad de imágenes de la Tierra y la atmósfera han sido proporcionadas por satélites y necesitan de filtrados automáticos.

En la actualidad los esfuerzos de la visión artificial están encaminados a la percepción 3D de una escena, y a la obtención de parámetros de movimiento. Relacionando el primer enfoque con la percepción humana observamos la capacidad del observador para determinar la estructura 3D de los objetos a partir de patrones bidimensionales de luz.

Consultando la Real Academia Española [RAE 2005] tendremos que percepción es: acción y efecto de percibir y también sensación interior que resulta de una impresión material hecha en nuestros sentidos. Las definiciones de percibir son, entre otras: recibir por uno de los sentidos las imágenes, impresiones o sensaciones externas y comprender o conocer una cosa.

En el siglo XIX Wheatstone proporcionó la primera evidencia experimental de que la profundidad se percibe cuando se le presenta a cada ojo imágenes dispares. Desde entonces la disparidad binocular, diferencia angular entre las imágenes del mismo rasgo en los ojos izquierdo y derecho, ha sido considerada una de las fuentes más importantes para la recuperación de información 3D. Entramos entonces en lo que es conocido dentro de la visión artificial como *Visión Estéreo*, es decir, la habilidad para inferir información de la estructura 3D y la distancia de una escena a partir de dos o más imágenes tomadas desde diferentes puntos de vista.

Otra manera de percepción 3D de la que dispone el ser humano es a partir del movimiento, aunque no está claro como realizamos nuestras deducciones a partir de escenas en movimiento y obviamente existen varias teorías. Dichas teorías derivan en el análisis de la estructura tridimensional del escenario, que se basa en calcular el mapa de profundidades de una imagen, su estudio es el objetivo de la técnica *Depth from motion* que consiste en extraer información de las profundidades de los objetos y lograr así una visión espacial a partir de una secuencia de imágenes en 2D.

Después de comentar la extracción de la forma a partir de la visión estéreo y el movimiento, podemos describir otro tipo de información básica disponible para cualquier sistema visual. Se trata de la cantidad de luz que le llega a diferentes regiones de la retina procedentes de superficies iluminadas en la escena. Si entendemos que las sombras son una suave degradación de la luz, y queremos extraer la mayor cantidad de información 3-D posible a partir de esta información, estamos ante el problema llamado extracción de formas a partir de la sombra o sombreado (*Shape Form Shading*).

Para la obtención de parámetros de movimiento, la visión artificial proporciona dos metodologías:

La primera está basada en las características geométricas de los objetos de las escenas visualizadas y se divide en tres fases: Extracción de elementos significativos, buscar correspondencias entre elementos vistos en distintas tomas monoculares, y estimar el movimiento y el entorno de la escena.

La segunda introduce el concepto de flujo óptico o campo de velocidades, que recibe su nombre del vocablo inglés *Optical Flow*.

El *Optical Flow* puede ser definido como el movimiento aparente de los puntos de la imagen que se produce cuando la cámara se mueve en relación con los objetos que se enfocan. El reconocimiento de la velocidad (relativa) de los objetos que nos rodean ofrece mucha información sobre nuestra propia velocidad.

El concepto más importante relacionado con el movimiento es el de campo de movimiento, que está formado por el campo de vectores bidimensionales de velocidades de

los puntos de la imagen, inducido por el movimiento relativo entre la cámara y el objeto observado. El flujo óptico, por su parte, es también un campo de vectores que aproxima, bajo determinadas circunstancias, el campo de movimiento y que puede obtenerse a partir de secuencias de imágenes que varían con el tiempo.

En un estudio de la “*British Machine Vision Conference*” [GALVIN 1998] se compararon 8 algoritmos de *Optical Flow* y las conclusiones generales de dicho estudio, sitúan a los algoritmos de “*Lucas & Kanade*” y “*Horn & Schunck*” como los mas eficientes.

Actualmente la visión por computador se emplea en el desarrollo de aplicaciones relacionada con campos como la industria [GONZALEZ 1999] (control de calidad, identificación de piezas, ensamblaje, guiado de robots), medicina (pruebas automáticas de laboratorios, diagnósticos por computador), tele-medición (exploración geográfica, meteorología), defensa militar (vigilancia por satélite, armas inteligentes), así como la automoción, que es el sector en el que se engloba este proyecto.

Capítulo III. Descripción del problema y objetivos

En este capítulo se abordará la descripción detallada del problema propuesto y de los objetivos concretos que se pretenden conseguir, intentando sentar las bases necesarias para comprender la futura explicación de la solución propuesta.

El problema planteado se puede describir en breves palabras suscribiendo el título del proyecto “*Desarrollo de una aplicación para análisis de movimiento intrínseco de vehículos en secuencia de imágenes*”. Del cual se extrae el acrónimo que da nombre al proyecto **AMIVSI**.

El problema propuesto se engloba dentro del ámbito de desarrollo de la visión artificial, un extenso campo que admite múltiples definiciones de las cuales cabe destacar la que enunció V. S. Nalwa [NALWA 1993]: “*La Visión Artificial describe la deducción automática de las estructuras y propiedades de un mundo tridimensional, posiblemente dinámico, a partir de una o varias imágenes bidimensionales de él*”. Esta definición se ajusta correctamente a la esencia del problema propuesto y describe la dificultad de los objetivos propuestos, la extracción de información tridimensional a partir de imágenes bidimensionales, todo ello en movimiento.

La aplicación final del estudio y desarrollo del proyecto está dirigida hacia el campo de la automoción, una de las principales áreas de expansión de la visión artificial. Se trata de desarrollar un sistema de análisis en tiempo real del movimiento intrínseco de un vehículo, dotado de una cámara de video capaz de capturar imágenes digitales, que serán la fuente de análisis. Se entiende por movimiento intrínseco el desplazamiento que genera el vehículo, dicho movimiento es posible describirlo mediante una medida de velocidad y una trayectoria; ambos forman parte de los objetivos a desarrollar.

Una característica destacable del proyecto es la disponibilidad futura para interactuar con un proyecto externo desarrollado por profesores de la *Universidad Europea de Madrid* (2004-2005) denominado SACAT “*Sistema Automático de Control y Alerta en caso de accidente para vehículos de Transporte por carretera*”. Dicho proyecto cuenta con financiación interna de la universidad y se enmarca dentro del plan nacional PROFIT: P.N. de Medios de transporte: Subprogramas de Automoción.

Se describe una colaboración directa entre ambos proyectos, distribuyendo responsabilidades para un fin común. El sistema SACAT es responsable de la captura de imágenes desde el interior del vehículo y debe facilitarlas al sistema AMIVSI para su procesado. Por su parte el sistema AMIVSI debe proporcionar los resultados de sus análisis al sistema SACAT, que se valdrá de ellos para mejorar sus prestaciones.

Es importante resaltar que la integración de nuestro sistema con la adquisición de imágenes en tiempo real, y la comunicación real con el sistema SACAT queda fuera del

ámbito del proyecto, por lo que se trabajará en todo momento con una galería de secuencias de imágenes previamente capturadas, y se describirá una comunicación apropiada para un futuro acoplamiento con el sistema SACAT. Como resultado de esta característica el sistema desarrollado trabajará como una simulación, aunque no eximiéndose por ello de las restricciones de tiempo real, ni de proporcionar los resultados que espera el sistema externo.

Una vez establecidas las delimitaciones entre ambos sistemas se detallarán los objetivos concretos del sistema AMIVSI. El objetivo prioritario del proyecto es desarrollar el procesado de imagen necesario para extraer la información detallada acerca del movimiento intrínseco, trabajando bajo un entorno C/C++.

A partir del objetivo prioritario se desglosan tres objetivos concretos:

El primero consiste en estimar una medida de velocidad de desplazamiento del vehículo con respecto de la cámara, se debe proporcionar como resultado del cálculo una velocidad expresada en kilómetros por hora. No se pretende la exactitud de resultados, sino el desarrollo de una técnica que permita extraer a partir de una secuencia de imágenes la información necesaria para realizar el cálculo de la velocidad. Es importante resaltar que el estudio se basará en la técnica del campo de velocidades, o flujo óptico. La funcionalidad proporcionada servirá de apoyo para el cálculo del siguiente objetivo.

El segundo objetivo es la estimación del espacio libre frente a la cámara, es decir, se pretende calcular las profundidades de la escena que representan las imágenes, generando como resultado un mapa de profundidades. La fuente del análisis serán las imágenes capturadas por la cámara, dichas imágenes representan en dos dimensiones una realidad tridimensional; es nuestro objetivo obtener la tercera dimensión necesaria para representar el mapa de profundidades. Con una sola imagen no disponemos de la información necesaria, para lograrlo utilizaremos una secuencia de imágenes que representen movimiento, a partir del cual extraeremos la tercera dimensión. Conociendo la profundidad hasta algún punto de la imagen, y la velocidad y dirección de movimiento, es posible calcular la distancia o el tiempo de impacto hacia dicho punto; proporcionando así información adicional al conductor.

El tercer objetivo consiste en el etiquetado y seguimiento de objetos previamente reconocidos en la imagen. La tarea de reconocimiento de objetos es responsabilidad del sistema SACAT; debido a la disociación de ambos sistemas, se realizará por nuestra parte un reconocimiento visual de objetos (señales de tráfico) candidatos para el estudio de este último objetivo, y su debida señalización. El etiquetado consiste en asignarle un identificador único a cada objeto, de manera que se reconozca y diferencie claramente del resto de los objetos de la secuencia. El seguimiento se realizará para cada objeto individualmente, y consistirá en la predicción de la posición futura del objeto en la siguiente imagen, de manera que sea relativamente sencillo identificar si un objeto se corresponde con la representación futura del mismo objeto (previamente reconocido), o de lo contrario se trata de uno nuevo. La presentación de los resultados obtenidos será mediante el interfaz gráfico del sistema, indicando donde se encuentra el objeto, además de indicar su etiqueta; y también se generará

un archivo de etiquetas que sirva de intercambio de información con el sistema SACAT. De dicho objetivo se obtiene la principal aportación al sistema SACAT, con la información de las etiquetas asignadas el sistema externo posee una información complementaria al reconocimiento de objetos. El sistema externo se apoyará en los resultados proporcionados por nuestro proyecto para determinar si debe analizar o no una señal reconocida, de esta manera se evita que el sistema SACAT analice una señal repetidas veces, consumiendo tiempo innecesario y desperdiando capacidad de cálculo.

El sistema final debe calcular y representar los tres objetivos concretos y cumplir con la restricción de un sistema en tiempo real.

Capítulo IV. Descripción de la solución propuesta

1 Introducción

A continuación se detallará en profundidad las diferentes metodologías y técnicas, tanto de investigación como de desarrollo, encaminadas a proponer una solución coherente al problema expuesto. Se incluirá en este capítulo toda la información de análisis y diseño que se ha elaborado durante la investigación y desarrollo del proyecto.

Se incluyen una colección de documentos que reflejan diversos aspectos de la ingeniería de software, como un documento de especificación de requisitos útil para comprender el alcance del sistema; un diseño de casos de uso que representan las interacciones con el usuario y sus colaboraciones internas; un análisis y diseño exhaustivo de los módulos del sistema necesarios para comprender la combinación de técnicas empleadas; con respecto a la implementación se incluye un estudio de clases y diagrama de secuencia, ambos documentos serán imprescindibles para la comprensión integral de la aplicación.

Es importante señalar el ámbito técnico en el que se enarcan todos los documentos posteriores, destinados a un lector con conocimientos informáticos medios y con nociones de visión artificial y tratamiento de imágenes. Si bien son recomendados, no son obligatorios, debido a que nuestros esfuerzos han ido encaminados a elaborar una memoria de fácil lectura, accesible a cualquier tipo de público.

2 Especificación de requisitos de software

2.1 Introducción

2.1.1 Propósito

Este documento pretende describir las especificaciones para el desarrollo del proyecto *AMIVSI*, incluyendo requisitos funcionales y no funcionales, parámetros de calidad, etc.

La estructura de este documento cumple con el estándar IEEE 830 / 1998 de Especificación de Requisitos de Software.

Está dirigido tanto a desarrolladores del sistema como a usuarios que precisen conocer cualquier aspecto interno de la aplicación de una forma concreta y detallada.

2.1.2 Ámbito

Titulo

Desarrollo de una aplicación para análisis de movimiento intrínseco de vehículos en secuencia de imágenes.

Acrónimo

AMIVSI: Análisis de movimiento intrínseco de vehículos en secuencia de imágenes.

Objetivo prioritario

Desarrollar el procesado de imagen necesario para extraer la información detallada en el resumen de proyecto, utilizando un entorno en C/C++.

Códigos de la UNESCO

Tecnología de los ordenadores – Sistemas en tiempo real 3304.17, Ciencia de Los Ordenadores - Sistemas de Navegación y Telemetría del Espacio 1203.21.

Resumen

En español:

El proyecto se engloba dentro del ámbito de la visión artificial, y más concretamente en la estimación de parámetros de velocidad y espacio a partir de una secuencia de imágenes. Para obtener dichos objetivos se estudiará cuales de las técnicas actuales son las más efectivas, aplicándolas para la consecución de los objetivos.

Se trata de realizar una aplicación y una librería en C/C++ para manejar secuencias de imágenes captadas desde el interior de un vehículo en movimiento. A partir del tratamiento de la secuencia de imágenes se debe conseguir tres objetivos fundamentales:

Estimar la velocidad de movimiento respecto a la cámara, basándose en el estudio del campo de velocidades de una imagen trataremos de aplicarlo en el campo de la automoción. Cabe destacar la importancia de este cálculo como base para la resolución de los siguientes objetivos.

Estimar el espacio libre frente a la cámara, o mapa de profundidades de la escena. Con una sola imagen no se dispone del tamaño real de un objeto, por lo tanto no es posible calcular la distancia a dicho objeto. Es por ello por lo que se utilizará una secuencia de imágenes y una estimación de velocidad para conseguir nuestro objetivo.

Identificar una serie de objetos previamente reconocidos en imágenes anteriores, a partir de una lista detallada (generada por el sistema SACAT) de los ya identificados, y asignar una nueva etiqueta a los objetos nuevos.

In English:

The project is included within the scope of the artificial vision, and more concretely in the estimation of parameters of speed and distance using a sequence of images. In order to obtain these objectives, we will study the present techniques they are most effective, applying them for the attainment of the objectives.

One is to make an application and a library in C/C++ to handle sequences of images caught from the inside of a vehicle in movement. From the treatment of the sequence of images we have to obtain three main targets:

To calculate the speed of movement with respect to the camera, being based on the study of the optical flow of an image we will try to use it in the field of the automotion. This calculation is very important because the resolution of the following objectives are based on it.

To consider the free space front to the camera, or depth map from the scene. With a single image we have not the real size of an object, therefore it is not possible compute the distance to this object. It is for that reason why it will be used a sequence of images and an estimation of speed to obtain our objective.

To identify a series of objects previously recognized in previous images, from a detailed list (generated by the SACAT system) of already identified, and to assign a new label to the new objects.

Beneficios para el usuario

Al realizar el proyecto mejora el modo de operar para el cliente. Se trata de un proyecto de desarrollo que proporciona servicios prácticos al cliente de los que antes carecía: capacidad del cálculo de la velocidad de un vehículo y del espacio libre frente a él, a partir de una cámara, además del etiquetado y seguimiento de objetos (normalmente señales de tráfico).

Explotación de resultados (nuevos servicios)

La explotación del proyecto se centra en el campo de la automoción y permite entre otros:

Mejoras de seguridad y ayudas a la conducción, como por ejemplo: Detección de obstáculos y alerta de colisión. Interpretación y alerta de señales de tráfico y aviso de infracción.

También se podría aplicar al campo de guiado de robots en procesos industriales.

2.1.3 Definiciones, siglas y abreviaciones

AMIVSI: Análisis de movimiento intrínseco de vehículos en secuencia de imágenes.

Optical Flow: Campo de velocidades o flujo óptico sobre una secuencia de imágenes, representando mediante vectores el desplazamiento de los puntos de una imagen.

Depth from motion: concepto que define la extracción de información tridimensional a partir de dos o más imágenes monoculares en movimiento

SACAT: Sistema Automático de Control y Alerta en caso de accidente para vehículos de Transporte por carretera

Distancia focal: es la distancia existente desde la lente al CCD de la cámara.

CCD: Charge Couple Device. Parte de la cámara compuesta de chips de silicio sensibles a la luz, que convierten los fotones de luz a señales eléctricas que se usan para componer la imagen.

MIL: Matrox Imaging Library. Se trata de unas librerías para C++ desarrolladas por la empresa norteamericana Matrox para el tratamiento de imágenes digitales.

FOE (Foco Óptico de Expansión): se entiende por foco óptico de expansión al punto de la imagen desde el cual surge el movimiento. Se considera el centro o fuente de los vectores de desplazamiento.

Píxel: identifica la unidad mínima de información de una imagen digital, representa un punto en función de las intensidades de tres colores primarios, rojo, verde y azul.

Umbral: se entiende por umbral a un límite de intensidad que se utiliza para discriminar los puntos de una imagen que no superen, o sobrepasen dicha marca.

Mapa de profundidades: representa las profundidades de una imagen identificando cada punto de la imagen con una distancia desde el objetivo de la cámara hasta el punto analizado.

Vector de desplazamiento: vector que representa el desplazamiento de un punto de la imagen tras un incremento de tiempo. Con dicho vector se identifica la posición final e inicial del punto analizado.

Vector de dirección (giro): vector que representa la dirección de giro que describe el automóvil en un instante determinado.

Frame: en español fotograma, se identifican con cada una de las imágenes que componen una secuencia de movimiento.

Patrón de búsqueda: región de tamaño configurado por el desarrollador que comprende una porción de imagen, que se utilizará para la búsqueda de correspondencias en una imagen diferente a la fuente del patrón.

Ventana de búsqueda: comprende una región de la imagen, normalmente mayor que el patrón de búsqueda, donde se establece la zona por donde se intentarán encontrar equivalencias.

PatternMatching: técnica de reconocimiento de patrones que se encarga de buscar equivalencias o correspondencias mediante un patrón de búsqueda en una imagen o región delimitada de la misma.

Convolución: Operación matemática en la que se recorre un área de una imagen para que sus píxeles se vean influenciados por sus vecinos en función de unos pesos de entrada.

Mascara de convolución: Matriz de valores que representan los pesos en una operación de convolución.

Gradiente: [RAE 2005] razón entre la variación del valor de una magnitud en dos puntos próximos y la distancia que los separa.

Scan progresivo: propiedad de determinadas cámaras de video que les permite capturar imágenes sin entrelazado.

Entrelazado: característica de una imagen en la que la información de la totalidad de una escena se representa haciendo uso de la mitad de sus líneas horizontales.

Histograma: [RAE 2005] representación gráfica de una distribución de frecuencias por medio de rectángulos, cuyas anchuras representan intervalos de la clasificación y cuyas alturas representan las correspondientes frecuencias.

2.1.4 Apreciación global

Descripción global:

Esta sección del ERS se describirán los factores generales que afectan al producto y sus requisitos. Esta sección no declara requisitos específicos. En cambio, se mantienen en un plano secundario ya que más tarde se definen en detalle y resultan más fáciles de entender.

- Perspectiva del producto.
- Funciones del producto.
- Características del usuario.
- Restricciones.
- Supuestos y dependencias.

Requisitos específicos:

Esta sección del ERS contendrá todos los requisitos del software a un nivel de detalle suficiente para permitirles a los desarrolladores diseñar un sistema para satisfacer dichos requisitos, y a los auditores a probar que el sistema satisface esos requisitos. A lo largo de esta sección, cada requisito declarado debe ser externamente perceptible por los usuarios, operadores u otros sistemas externos.

Estos requisitos incluirán una descripción de cada entrada (el estímulo) en el sistema, cada salida (la contestación) del sistema, y todas las funciones realizadas por el sistema. Nos encontramos ante la sección más relevante del ERS.

- Interfaces externas.
- Requisitos Funcionales.
- Requisitos de rendimiento.
- Requisitos del banco de datos lógicos.
- Aceptación de las normas.
- Atributos del software del sistema.
- Fiabilidad.

Seguridad.
Mantenimiento.
Portabilidad.
Organizar los requisitos específicos.
Modo del sistema.
Clases de usuarios.
Objetos.
Característica / Rasgo.
Estímulo.
Contestación.
Jerarquía Funcional.

2.2 Descripción global

2.2.1 Perspectiva del producto

El producto especificado en este documento está relacionado con el sistema SACAT de identificación de señales de tráfico y mensajes de alerta, esta colaboración se ha tratado de manera que facilite su conexión final en un futuro, aunque no pertenece a la perspectiva del sistema AMIVSI realizar dicha conexión, tan solo facilitar un interfaz de comunicación entre ambos sistemas. El sistema AMIVSI se encarga de calcular la velocidad a la que se mueve el vehículo, el espacio libre frente a él y por último, interactuando con el proyecto SACAT, debe de predecir la posición de objetos previamente reconocidos en la siguiente secuencia de imágenes además de asignarles un identificador unívoco, y finalmente contrastar los resultados obtenidos con los esperados.

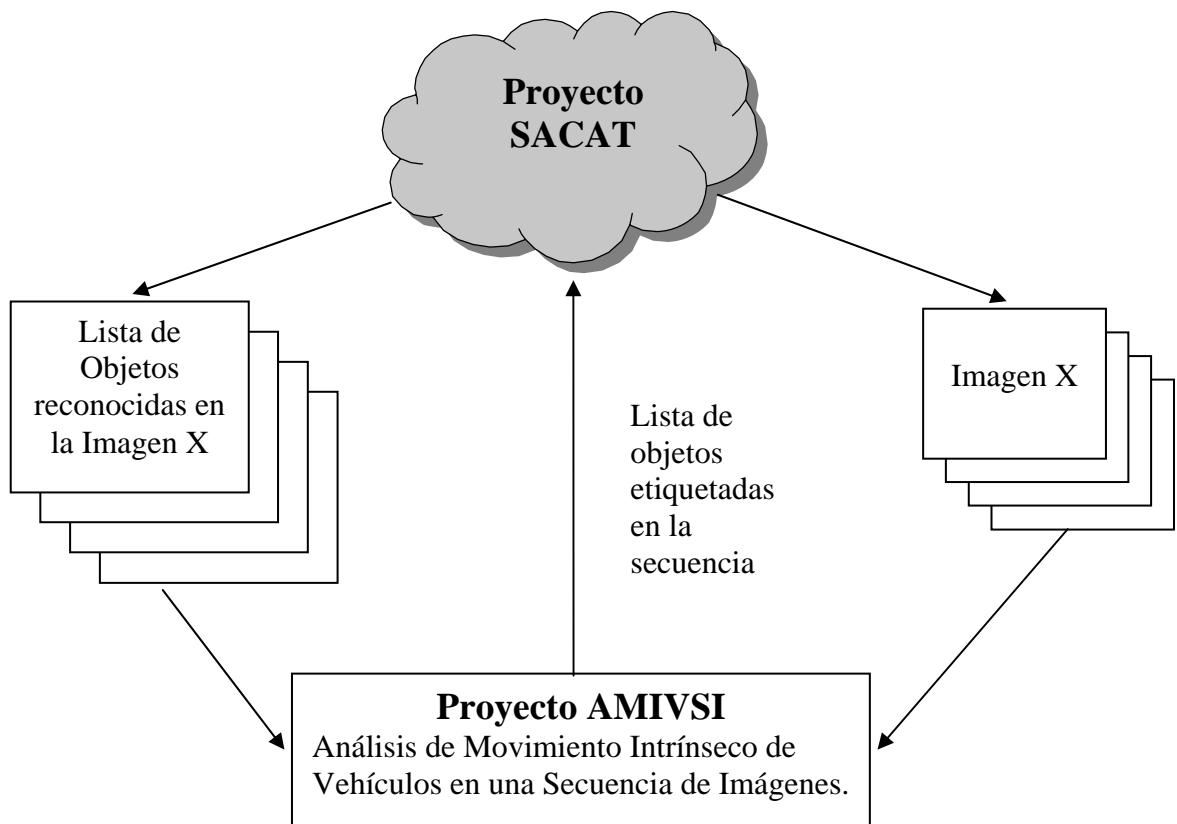


Figura IV-1 Diagrama de inter actuación con el sistema SACAT

1) Interfaces del sistema

Con el objetivo de facilitar las futuras interfaces de nuestro sistema con respecto al proyecto SACAT se describirán a continuación las interfaces de entrada y salida deseadas para una correcta interactuación de ambos sistemas. Dichos interfaces no se utilizan en la actualidad aunque se diseñan y detallan para un futuro uso.

a) Interfaces de entrada

Imagen: El proyecto SACAT captura el video desde la cámara situada en el vehículo para posteriormente obtener la secuencia de imágenes, y situar en un directorio compartido las imágenes capturadas; quedando a disposición del sistema AMIVSI. Las imágenes serán desentrelazadas, a color y con una resolución de 720 x 576 píxeles y en formato Bitmap

(bmp). El nombre de cada imagen seguirá una numeración empezado en uno y siguiendo un incremento de unidad, de manera que la primera imagen tendrá en nombre *1.bmp* y las sucesivas seguirán la numeración especificada: *2.bmp*, *3.bmp*, etc.

Nota: Como la comunicación con el sistema SACAT está simulada, es necesario generar una secuencia de imágenes por cuenta propia, y aplicarle un desentrelazado si fuera necesario.

Lista de Objetos (señales) reconocidas en la Imagen: el proyecto SACAT proporciona una lista de las coordenadas de los objetos reconocidos en cada imagen. El formato específico de dicha lista será en forma de archivo con la extensión *dat* y recibirá el nombre de la imagen a la que pertenece la lista; por tanto si estamos detallando la lista de objetos de la imagen 23 el archivo de coordenadas recibirá el nombre *23.dat*. La estructura de cada archivo de objetos seguirá el siguiente formato:

Cada línea del archivo identificará las coordenadas que describen el rectángulo que engloba a un objeto. De esta manera existe la posibilidad de que coexistan en una misma imagen numerosos objetos. Cada línea se divide en 4 números enteros expresados utilizando 3 caracteres numéricos para cada entero. Los enteros describen dos puntos representados en un sistema de coordenadas en el eje *x* e *y* utilizando como unidad los píxeles de la imagen, de manera que el primer entero se corresponde con la coordenada *x* de la esquina superior izquierda; el segundo a la coordenada *y* de la esquina superior izquierda; el tercero la coordenada *x* de la esquina inferior derecha y el cuarto la coordenada *y* de la esquina inferior derecha. Cada uno de estos enteros está separado del siguiente mediante un espacio. En el caso de que un entero se represente con menos de tres caracteres numéricos se sustituirán los restantes por espacios, de forma que para expresar el número 35 se introducirá [espacio|dígito_3|dígito_5]. Veamos a continuación un ejemplo de archivo de objetos:

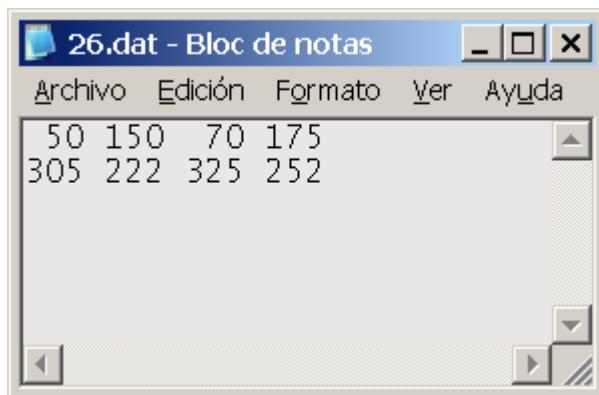


Figura IV-2 Formato del archivo de lista de objetos reconocidos

Se observa que en la imagen 26 han sido reconocidos dos objetos, el primero se presenta en la primera línea y describe un rectángulo de esquina superior izquierda (50,150) y de esquina inferior derecha (70,175). Entendiendo la representación de cada punto como un sistema de correderas en los ejes x e y , (eje X, eje Y).

b) Interfaces de salida

Lista de objetos etiquetados en la Secuencia: el sistema ha de elaborar un archivo con extensión *dat* que reciba el nombre de *registroEtiquetas*. Dicho archivo ha de contener la información de cada etiqueta asignada durante el análisis y que servirá al sistema SACAT. La estructura del archivo seguirá el siguiente formato:

Se debe detallar las etiquetas asignadas siempre que exista algún objeto en la lista de objetos reconocidos, y se hará indicando en qué imagen han sido encontrados, su etiqueta, posición inicial y posición futura. El campo de etiqueta mantiene una nomenclatura determinada, todas las etiquetas comienzan con el prefijo *Obj*, seguido de un guión y el número de objeto reconocido que irá en aumento a lo largo del análisis. La posición inicial y posición futura se representan de manera análoga a la lista de objetos reconocidos, mediante dos puntos expresados en sistemas de coordenadas x e y que representen la esquina superior izquierda e inferior derecha de un rectángulo. La posición inicial representa la posición en la que se encontró dicho objeto en la imagen indicada, y la posición futura deberá indicar la posición futura en la que se encontrará el mismo objeto en la siguiente imagen de la secuencia. Se representaran los distintos objetos de una imagen separados mediante una línea en blanco. Aunque conservando el mismo tabulado. A continuación se representa un ejemplo del formato a seguir:

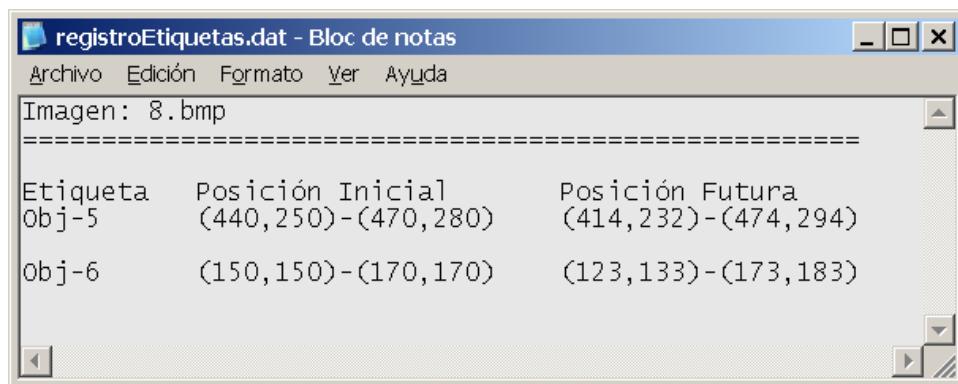


Imagen: 8.bmp		
Etiqueta	Posición Inicial	Posición Futura
Obj-5	(440,250)-(470,280)	(414,232)-(474,294)
Obj-6	(150,150)-(170,170)	(123,133)-(173,183)

Figura IV-3 Formato del archivo de registro de etiquetas

Se observa como primero se identifica la imagen a la que se hace referencia y a continuación separando mediante una línea de caracteres '=' se introducen las cabeceras de las columnas y en cada fila la representación de un objeto con todos sus atributos.

2) *Interfaces con el usuario*

a) Interfaz de configuración

Fichero de Configuración: Proporciona comunicación entre el sistema y un posible administrador. Se utiliza para almacenar parámetros de configuración del Optical Flow que luego serán leídos por el sistema. Este fichero recibe el nombre de CONFIG.DAT y posee los siguientes datos:

- Coeficiente del tamaño máximo que puede tener la ventana de búsqueda.
- Coeficiente del tamaño mínimo que puede tener la ventana de búsqueda.
- Coeficiente del tamaño del patrón de búsqueda.
- Umbral mínimo de luminosidad de un punto para ser analizado.
- Número de píxeles entre dos puntos de análisis de la imagen.
- Porcentaje de tolerancia a fallos en el PatternMatching.
- El contenido del fichero se debe especificar de la siguiente manera:

variable=valor

variable: [A..Z|_]
valor:[Entero|Decimal]

COEFICIENTE_VENTANA_MAX=Decimal
COEFICIENTE_VENTANA_MIN=Decimal
COEFICIENTE_PATRON=Decimal
UMBRAL=Entero
ESPACIO_ENTRE_VECTORES=Entero
PORCENTAJE_VALIDO=Decimal

En la siguiente figura se muestra un ejemplo con el contenido del fichero con valores por defecto.

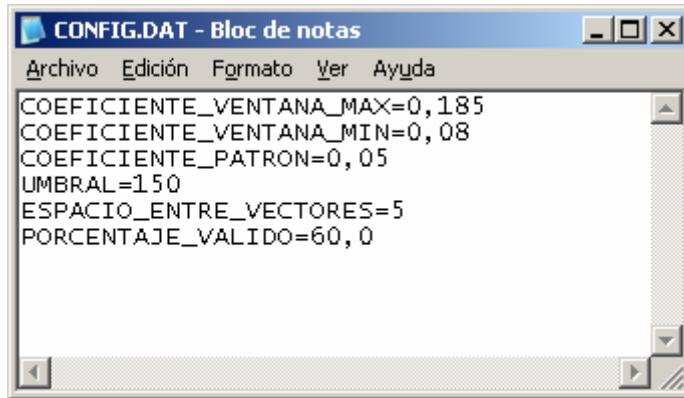


Figura IV-4 Fichero de configuración

b) Interfaz gráfico

El interfaz deseado para el usuario debe incluir tres apartados fundamentales que detallaremos a continuación:

Imagen fuente: es la sección principal del interfaz que incluye la imagen que ha sido analizada en la que se indica el espacio libre ante un objeto o puntos significativos, además de la localización y la etiqueta asignada a cada objeto previamente reconocido. La representación de la profundidad de la escena se deberá hacer siguiendo alguna técnica que no interfiera con la representación de los demás objetivos. Es necesario expresar dicha información mediante una ventana de profundidades, disociando dicha información de la imagen real.

Velocímetro (analógico - digital): el interfaz debe mostrar el resultado de la estimación de velocidad en formato digital y analógico.

Lista de etiquetas: se representará mediante una tabla cada una de las etiquetas asignadas a cada objeto junto con su posición inicial en la imagen, así como de su posición futura estimada. La representación de coordenadas se hará de manera análoga a lo documentado en el apartado de interfaces del sistema.

Este interfaz representa los resultados obtenidos en tiempo real por nuestro sistema, por lo tanto el interfaz se actualizará cada pocos segundos dependiendo de la frecuencia de los interfaces de entradas y el tiempo de cálculo de los resultados esperados.

A continuación se presenta el interfaz preeliminar:

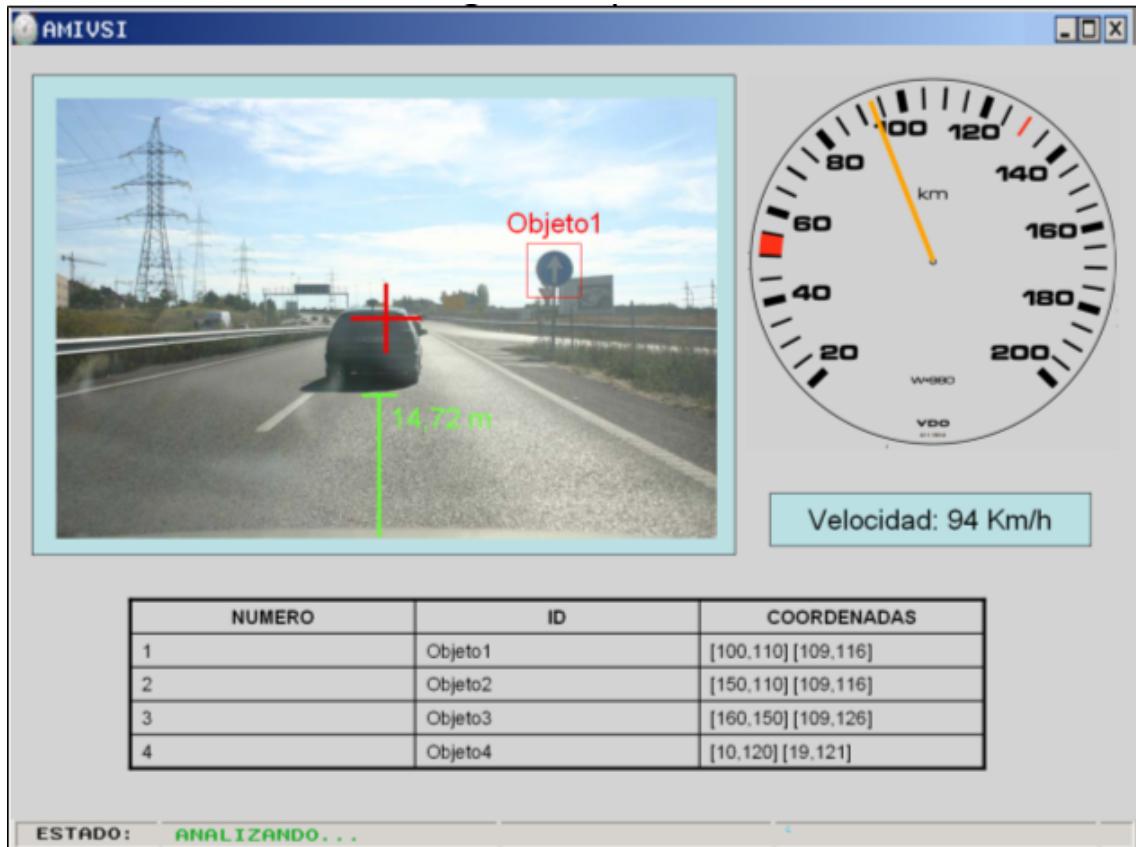


Figura IV-5 Interfaz preliminar del sistema AMIVSI

3) Interfaces con el Hardware

La consideración futura del sistema AMIVSI prevé una cooperación con el sistema SACAT, el cual describirá las interfaces con el Hardware de una forma más restrictiva. Para el proyecto AMIVSI:

Interfaz Gráfico - Pantalla: La aplicación deberá mostrar imágenes a color a una resolución de 720x576 píxeles. Cada cierto periodo de tiempo (variable dependiendo del incremento de tiempo entre las imágenes de una secuencia) se mostrará una imagen nueva en el interfaz gráfico y se actualizarán los valores de los resultados obtenidos, como el cálculo de velocidad, la profundidad de la escena y el etiquetado de objetos.

4) Interfaces con el Software

a) **Nombre: Sistema Operativo**

Siglas: Windows.

Numero de especificación: No Aplicable.

Numero de versión: 98, 2000, XP.

Fuente: Microsoft.

No hay que realizar ningún interfaz, puesto que es el sistema operativo quien realiza esta tarea.

b) **Nombre: Matrox Imaging Library**

Siglas: MIL.

Numero de especificación: ImagingClasses

Numero de versión: 5.12

Fuente: Matrox

Debe de haber un interfaz para facilitar el uso de estas librerías, el cual se llamará ImagingClasses.

c) **Nombre: Sistema SACAT**

Siglas: SACAT

Numero de especificación: DirectorioCompartido

Fuente: Universidad Europea de Madrid

Este Sistema almacena en un directorio compartido (SACAT-AMIVSI) los datos de entrada y leerá los datos resultantes en un formato fácilmente interpretable por los dos sistemas. De esta manera se trata de aislar lo más posible a los dos sistemas.

5) Interfaces de comunicaciones

No se contempla ninguna comunicación entre varios ordenadores por lo tanto este punto no es aplicable.

6) Restricciones de memoria

No existe una restricción acerca de la cantidad de memoria primaria o secundaria que se necesita.

7) *Funcionamientos*

Periodos Interactivos: El usuario inicia o cierra la aplicación.

Periodos de Espera: Se produce tras haber iniciado la aplicación y hasta que el sistema finaliza el análisis.

Funciones de procesamiento de datos: Una vez iniciado el sistema por el usuario el sistema se mantendrá realizando estas funciones.

Funcionamientos de Backup y Recuperación: Se trata de un sistema de tiempo real en el cual se almacenan los datos de entrada y los resultados obtenidos momentáneamente hasta que el sistema se actualiza. En ningún caso se contempla un requerimiento de backup o recuperación de datos.

8) *Requisitos de adaptación al sistema*

No debe de haber ningún requerimiento especial para la adaptación del sistema.

Debe de existir un software previamente instalado como es el sistema operativo, y tener accesibles las librerías Matrox. La instalación de este software no requiere instalaciones particulares.

2.2.2 Funciones del producto

Obtención de la velocidad global de la cámara: Se calculará el Optical Flow con una secuencia de imágenes y eligiendo puntos destacados. Se traza un vector de desplazamiento para cada punto correlacionado en las imágenes y posteriormente se determina un vector único para esa secuencia indicando la velocidad. Resultado de aplicar varios algoritmos

Obtención del espacio libre frente a la cámara: Se requiere calcular un mapa de profundidades de la escena que permita representar de manera intuitiva los puntos de la imagen que se encuentren más cercanos o alejados del objetivo, así como de valores intermedios.

Etiquetado de objetos previamente reconocidos: Como entrada tenemos una lista de objetos reconocidos por el sistema del proyecto SACAT. Nuestro objetivo es etiquetar cada objeto reconocido y estimar donde podría estar en la siguiente imagen. Si en la siguiente imagen la entrada del objeto es diferente a nuestra aproximación, lo etiquetamos como un nuevo objeto, de lo contrario se trata del mismo objeto.

2.2.3 Características del usuario

El principal usuario al que está destinado este producto es cualquier conductor que precise una asistencia a la conducción en tiempo real. No es necesario tener conocimientos avanzados de informática, simplemente unos conocimientos mínimos muy básicos, ya que la aplicación presenta una interfaz bastante intuitiva y el usuario no realiza comunicación interactiva con la aplicación salvo abrir y cerrar dicha aplicación. Al tratarse de un sistema de información en tiempo real, en el que el usuario no interactúa con el sistema, este se limitara a recibir información útil directamente desde el interfaz.

El nivel educativo, la experiencia previa y la especialización técnica necesarios para la comprensión y uso de esta aplicación son como mínimo las de un conductor con permiso de conducir valido en España.

2.2.4 Restricciones

Políticas de regulación o legislación

La información registrada en la aplicación debe tener el carácter meramente informativo. Cualquier uso de dicha aplicación en la práctica deberá estar autorizada y regulada previamente.

Restricciones de tiempo

Al ser un sistema de tiempo real, nuestro sistema deber dar un resultado antes de un tiempo que podrá ser configurable. Cuanto mas tiempo se disponga más fiable será dicho resultado.

Limitaciones del Hardware

Como disponemos de una restricción de tiempo cabe señalar que a mayor calidad y calidad del hardware se dispondrá de mejores resultados. Además, si el sistema se instalará en un vehículo dicho hardware deberá estar homologado.

Interfaces con otras aplicaciones

La característica de interconexión futura con el sistema SACAT establece que los datos tanto de entrada como de salida son manejados por la propia aplicación y el sistema externo.

Protocolos de comunicación

La aplicación no utiliza ningún protocolo de comunicación.

Restricciones del lenguaje

El sistema se implementará en C++ puesto que las librerías para el manejo de imágenes *MIL* que se disponen, están en dicho lenguaje. Además, en el campo de la visión por computador es el que mayoritariamente se utiliza gracias a su fiabilidad y rapidez frente a otros lenguajes como por ejemplo los interpretados.

Requisitos de Fiabilidad

Debe poseer un mínimo de fiabilidad para que el sistema sea de utilidad. En cualquier caso se debe realizar un estudio de fiabilidad una vez realizada la aplicación. En dicho estudio se podrá entender en qué casos el sistema se comporta mejor y en qué casos el sistema no es fiable.

Carácter crítico de la aplicación

El uso de la aplicación no debe tener un carácter crítico. Por lo tanto la posible perdida de los datos no supone riesgo importante. En todo caso debe evitarse cualquier posible fallo en las funciones.

Consideraciones de seguridad y de protección

El programa deberá ser usado como un asistente de ayuda a la conducción, y en ningún caso se pretende suplantar ninguna actividad que realice un conductor al utilizar un vehículo.

2.2.5 Supuesto y dependencias

Si no se dispone de ninguna secuencia de imágenes, el sistema no mostrará ningún tipo de información.

2.3 Requisitos específicos

2.3.1 Interfaces externas

El interfaz de usuario deberá ser muy sencillo y su aspecto se asemejará a la *Figura IV 5.*

De hardware

La aplicación en un principio esta pensada para funcionar sobre un sistema operativo Microsoft Windows 98 / 2000 / XP.

De software

Para que la aplicación funcione correctamente se deberá tener instalada las librerías Matrox 5.12 para el procesado de imágenes.

De comunicación

No se describe ninguno.

2.3.2 Requisitos Funcionales

1) *Velocidad*

Nombre: Calcular la velocidad global del vehículo

Descripción: El cálculo de la velocidad global del vehículo se realiza basándose en el resultado de cálculo del Optical Flow y extrayendo la información de desplazamiento necesario para estimar una medida de velocidad.

Razón: Se utiliza el resultado del Optical Flow ya que es la técnica mas apropiada para determinar el desplazamiento de los objetos dentro de la imagen.

2) *Profundidad*

Nombre: Calcular el espacio libre frente a la cámara, mapa de profundidades.

Descripción: A partir de una secuencia de imágenes se obtiene la distancia desde el objetivo de la cámara a un objeto o punto de la imagen. Para ello se hace uso del resultado del Optical Flow. Por medio de la técnica *Depth from motion* se calcula la profundidad relativa de la escena.

Razón: Es necesario trabajar con al menos dos imágenes ya que intentamos calcular una característica tridimensional a partir de unas imágenes en dos dimensiones.

3) *Etiquetado*

Nombre: Etiquetado de objetos previamente reconocidos.

Descripción: Con la posición del objeto en una imagen se estima mediante el Optical Flow su posición en la siguiente imagen y dependiendo de si la nueva posición existe o no lo etiquetaremos como un nuevo objeto.

Razón: El sistema SACAT es incapaz de determinar si un objeto es nuevo o ya fue reconocido, por lo tanto gracias al Optical Flow podemos estimar donde debería estar si es que existe el objeto a reconocer. Si la posición del objeto coincide con nuestra estimación podemos concluir diciendo que el objeto es el mismo y por tanto le daremos la misma etiqueta.

2.3.3 Requisitos de rendimiento

El sistema requiere un único terminal con un único usuario simultáneo. Se manejarán secuencias de imágenes sin compresión y se actualizaran dichas imágenes cada cierto tiempo.

El rendimiento relacionado con la gestión de tiempos es crítico por tratarse de un sistema en tiempo real.

En condiciones normales el sistema deberá usar el máximo disponible de CPU.

El 90% de los resultados deberán ser calculados antes de un tiempo máximo

2.3.4 Requisitos del banco de datos lógicos

Es necesario identificar un banco de datos lógicos que serán usados durante la ejecución de la aplicación. No conforman una base de datos pero si son variables o datos intermedios útiles para la consecución de los objetivos.

Tipo de Información utilizada

Imagen de entrada (objeto TImagen).

Lista de objetos previamente reconocidos identificados por coordenadas.

Velocidad actual del vehículo.

Intervalo de tiempo entre imágenes (variable real).

Lista de objetos reconocidos y etiquetados.

Lista de predicción de posición de los objetos ya etiquetados.

Espacio libre frente a la cámara, variables intermedias de tiempo para impacto que se utilizan para representar la profundidad relativa a la escena.

La frecuencia de uso

La frecuencia de uso es cíclica, a medida que se reciba una nueva imagen se trabajará con los datos ya calculados y almacenados para la anterior secuencia de imágenes.

Tipo de accesos

En la mayoría de las variables especificadas se realizarán operaciones de lectura al inicio de un ciclo y de escritura al finalizarlo. La variable u objetos que no sufren alteraciones son: Imagen de entrada, lista de objetos previamente reconocidos, intervalo de tiempo entre imágenes.

Entidades y relaciones

Al no tratarse de una base de datos relacional no es posible detallar este apartado.

Restricciones de Integridad

Al no tratarse de una base de datos relacional no es posible detallar este apartado.

Requerimientos en la retención de datos

Al no tratarse de una base de datos relacional no es posible detallar este apartado.

2.3.5 Aceptación de las normas

Formato de la documentación

Este informe ha sido realizado siguiendo el estándar 830/1998 publicado por la agencia IEEE, cubriendo los diferentes puntos detallados para la especificación de requisitos de software (ERS).

Nombre de los datos

No aplicable.

Procedimientos de contabilidad

En la elaboración del documento de planificación se ha realizado un seguimiento de la contabilidad.

Trazas de Auditoria

Al tratarse de un proyecto de fin de carrera no se realizará ninguna auditoria oficial, pero si está sujeto a revisiones del tutor del proyecto.

2.3.6 Atributos del software del sistema

Se detallarán los siguientes atributos de calidad del software, especificando lo deseable del producto final.

2.3.7 Fiabilidad

La fiabilidad del software desarrollado por el proyecto AMIVSI se puede valorar para cada uno de los tres objetivos a conseguir. La velocidad calculada debe coincidir como mínimo en un 60% con la velocidad real del vehículo así como el espacio libre frente a la cámara, por ultimo el etiquetado de objetos debe presentar una fiabilidad del 70%.

2.3.8 Seguridad

El sistema al no poseer una interfaz complicada con el usuario, ni excesiva interconexión con el mismo, no se proporciona de ninguna medida de seguridad para prever uso indebido o intento de manipulación malintencionada.

Con respecto a la seguridad interna se aplica la técnica de diferenciación de funcionalidades por módulos logrando una restricción entre los mismo con el fin de garantizar la estabilidad y seguridad del sistema.

El sistema realiza un control cíclico del estado de las variables pertenecientes al banco lógico de datos, acreditando su validez para el futuro uso.

2.3.9 Mantenimiento

El mantenimiento del sistema se facilita modularizando el sistema y detallando cada una de sus partes y funcionalidades mediante los documentos adjuntos, de manera que resulta sencillo para los desarrolladores progresar con el proyecto en evoluciones o modificaciones.

Cabe destacar que es necesario el conocimiento del campo de la visión por computador si se desea continuar con el proyecto, ya que su principal enfoque es hacia dicha materia.

2.3.10 Portabilidad

El sistema desarrollado no posee la cualidad de multiplataforma. Se implementa en Borland® C++ bajo sistema operativo Windows®, ambos componentes son propietarios y se adquieren bajo licencia.

Por último se utiliza una herramienta de tratamiento de imágenes de Matrox®, las librerías MIL® (Matrox Image Library) implementadas en C++, en su versión 5.12 de distribución libre.

Estos tres componentes utilizados restringen al máximo la portabilidad del sistema utilizado en términos de plataforma y lenguaje de programación.

El procesador del ordenador se deja a elección del usuario, siempre que dicho procesador soporte Windows®.

2.3.11 Organizar los requisitos específicos

A continuación se realizará una organización de requisitos específicos siguiendo distintos criterios de calificación, con el fin de proporcionar un entendimiento total de los requisitos del sistema. Cualquiera de las opciones expuestas proporciona una organización apropiada para el sistema *AMIVSI*, sin embargo la organización por Contestación proporciona la secuencia de funcionalidades y sus dependencias necesarias para el fin del sistema.

2.3.12 Modo del sistema

El sistema *AMIVSI* sólo posee un modo de operación. Al tratarse de un sistema de análisis y monitorización de resultados se observa un único modo de funcionamiento, constante e idéntico para todas las situaciones de uso.

El usuario inicia la aplicación y con ella el análisis y no interactúa más con el sistema hasta el cierre de la misma, por lo tanto es el sistema el que realiza todo el trabajo sin necesidad de una orden o acción del usuario. Esta característica es la que marca un único modo de operación.

2.3.13 Clases de usuarios

El sistema *AMIVSI* ofrece las mismas posibilidades a toda clase de usuario, los resultados del análisis no varían según el usuario, ya que no presenta diferentes modos de manipulación. En el resultado final del análisis tan solo influye la entrada de la cámara situada en el interior del vehículo.

2.3.14 Objetos

Véase diseño de clases punto 4.

2.3.15 Característica / Rasgo

La entrada del sistema *AMIVSI* es la secuencia de imágenes de la cámara y la lista de objetos reconocidos por el sistema *SACAT*, es deseable que las imágenes de entrada posean una serie de características de nitidez, visibilidad, integridad, etc. La lista de objetos previamente reconocidos debe identificar mediante un sistema de coordenadas los objetos dentro de la imagen.

2.3.16 Estímulo

Se entiende por estímulos los pequeños objetivos parciales que sirven de guía para un fin mayor.

El sistema *AMIVSI* se desglosa en la consecución de varios objetivos que forman el objetivo final del proyecto. Es posible identificar los siguientes estímulos dentro del sistema: análisis del *Optical Flow*, obtención de la velocidad de la cámara a partir de imágenes monoculares, estimación de espacio libre (en distancia), predicción de posición de objetos.

La totalidad de los estímulos descritos conforman el fin del proyecto.

2.3.17 Contestación

A continuación se organizará el sistema describiendo todas las funciones en el apoyo de la generación de una contestación. Es posible organizar la contestación final en las siguientes funciones:

1) Análisis y generación del *Optical Flow*

Estudiando y comparando las imágenes de la secuencia generamos el campo de velocidades de la imagen, señalando con vectores de desplazamiento cada uno de los puntos que se han desplazado. Esta funcionalidad será necesaria para los siguientes objetivos.

2) Obtención de la velocidad global del objetivo

Utilizando el campo de velocidades generados se interpreta la velocidad global de vehículo.

3) *Cálculo del espacio libre, mapa de profundidades*

A continuación de la obtención de la velocidad y con los datos conocidos de antemano como intervalo de tiempo entre dos imágenes podemos elegir puntos resaltados en la imagen y calcular la distancia hacia ellos.

4) *Predicción de la posición de los objetos reconocidos en futuras imágenes*

Necesitaremos el análisis de la velocidad y la lista de objetos reconocidos con su posición indicada en coordenadas dentro de la imagen.

La secuencia expuesta de contestaciones parciales conforma la contestación final del sistema.

2.3.18 Jerarquía Funcional

La organización de los requisitos específicos ha quedado correctamente especificada en los anteriores subapartados, por lo cual no es necesario emplear una organización funcional.

Éste es el párrafo que indica el contenido del capítulo tras la introducción. Sólo sirve de referencia para ver el contenido a primer nivel

3 Casos de Uso

En este apartado se aborda aspectos de análisis, para ello se ha hecho uso de una de las herramientas más potentes de las que disponen los desarrolladores. Se trata de los *Casos de uso* (Use Case). Los Casos de uso [JACOBSON 1992] describen bajo el marco de acciones y reacciones el comportamiento de un sistema desde el punto de vista del usuario; permitiendo definir los límites del sistema y las relaciones entre sistema y el entorno.

El sistema AMIVSI describe una interacción con el usuario nula o muy escasa, sin embargo no por ello está ausente de casos de uso, ya que se identifican como tal las colaboraciones entre los distintos módulos. Detallando estas interacciones se consigue alcanzar un nivel más profundo de entendimiento.

Tras analizar los requisitos se modelan los siguientes casos de uso:

1. Iniciar Sistema AMIVSI
2. Leer Imagen de Entrada
3. Calcular velocidad
4. Calcular profundidades
5. Etiquetar un objeto
6. Refrescar interfaz
7. Calcular Optical Flow
8. Estimar posición de un objeto

La totalidad de estos casos de uso y sus interacciones se pueden observar en el siguiente diagrama de casos de uso, que proporciona una visión gráfica de las colaboraciones.

Por último se adjunta una tabla por cada caso de uso donde se explican en detalle cada uno de sus pasos.

3.1 Diagrama de Casos de Uso

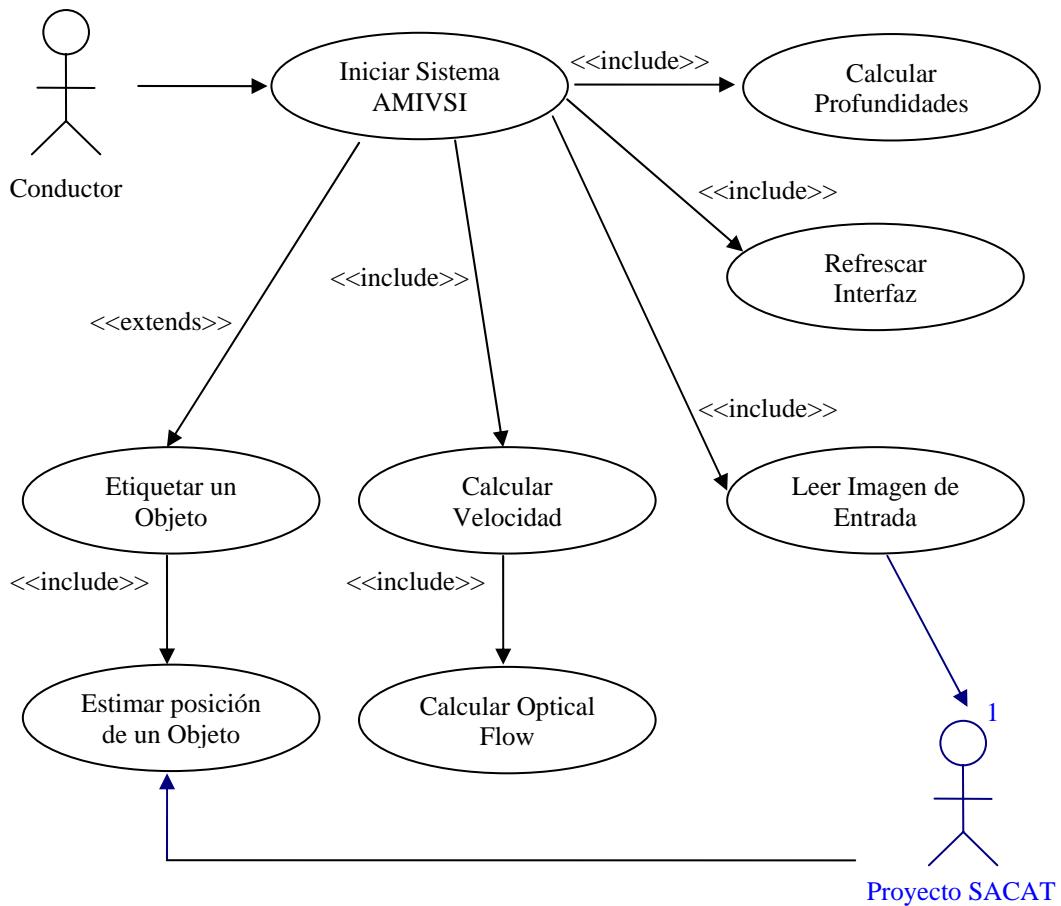


Figura IV-6 Diagrama de Casos de Uso.

¹ La interacción del Proyecto SACAT con el sistema AMIVSI no se contempla en la descripción de casos de uso.

3.2 Caso de Uso Iniciar Sistema AMIVSI

Nombre:	Iniciar Sistema AMIVSI
Actores:	Conductor
Precondiciones:	1. El directorio compartido entre nuestro sistema y el proyecto SACAT contiene una secuencia de imágenes.
Flujo Principal:	1. El <i>Conductor</i> inicia la aplicación <i>AMIVSI</i> 2. Punto de Inclusión 1: Caso de Uso “Leer Imagen de entrada” 3. [No es la primera imagen de la secuencia] Punto de Inclusión 2: Caso de Uso “Calcular velocidad” 4. [Se ha calculado la velocidad] Punto de Inclusión 3: Caso de Uso “Calcular profundidades”. 5. [Hay algún objeto reconocido] Punto de Extensión 1 : Caso de Uso “Etiquetar un objeto” 6. Punto de Inclusión 4 : Caso de Uso “Refrescar Interfaz” 7. Ir al paso 2.
Flujo Excepcional 1:	3. [Es la primera imagen de la secuencia] Ir al paso 5
Flujo Excepcional 2:	4. [No se ha calculado la velocidad] Ir al paso 5
Flujo Excepcional 3:	5. [No hay ningún objeto reconocido] Ir al paso 6.
Flujo Alternativo:	4. Puntos del 2 al 7. I. El Conductor cierra la aplicación <i>AMIVSI</i> II. El sistema <i>AMIVSI</i> se finaliza
Poscondiciones:	1. La secuencia de imágenes analizada se ha borrado del directorio compartido 2. Las listas de objetos reconocidos se han borrado del directorio compartido

Tabla IV-1 Caso de Uso Iniciar Sistema AMIVSI.

3.3 Caso de Uso Leer imagen de entrada

Nombre:	Leer imagen de entrada
Actores:	Proyecto SACAT
Precondiciones:	1. El proyecto SACAT esta iniciado.
	2. El proyecto SACAT está enviando imágenes al directorio compartido
	3. El proyecto SACAT está enviando listas de objetos reconocidos al directorio compartido
Flujo Principal:	1. El caso de Uso Iniciar sistema AMIVSI realiza una llamada a Leer imagen de entrada.
	2. [El proyecto SACAT guarda al menos dos imágenes en el directorio compartido] Se cargan en memoria dos imágenes de la secuencia, con sus correspondientes identificadores
	3. Se cargan en memoria las listas de los objetos reconocidos para cada imagen
	4. Se borran del directorio compartido las imágenes cargadas en memoria
	5. Se borran del directorio compartido las listas cargadas en memoria
Flujo Alternativo:	2. [El proyecto SACAT no guarda al menos dos imágenes en el directorio compartido] Se finaliza el caso de uso.
Poscondiciones:	1. Se han borrado las imágenes cargadas del directorio compartido
	2. Se han borrado las listas de objetos reconocidos cargadas del directorio compartido

Tabla IV-2 Caso de Uso Leer imagen de entrada.

3.4 Caso de Uso Calcular Optical Flow

Nombre:	Calcular Optical Flow
Actores:	
Precondiciones:	<ol style="list-style-type: none">1. Existe una secuencia nueva de imágenes cargadas en memoria
Flujo Principal:	<ol style="list-style-type: none">1. El caso de Uso Calcular velocidad realiza una llamada a calcular Optical Flow2. Se aplica un filtro de detección de bordes a las imágenes de la secuencia3. [Existen puntos destacados en la secuencia de imágenes] Se eligen puntos destacados de la primera imagen y se buscan las correspondencias en la segunda imagen4. Para cada correspondencia de puntos se calcula un vector de desplazamiento5. Se genera una lista con todos los vectores de desplazamiento
Flujo Alternativo:	<ol style="list-style-type: none">3. [No existen puntos destacados en la secuencia de imágenes] Se finaliza el caso de uso
Poscondiciones:	<ol style="list-style-type: none">1. Se ha generado una lista de vectores de desplazamiento para la secuencia de imágenes de entrada

Tabla IV-3 Caso de Uso Calcular Optical Flow.

3.5 Caso de Uso Calcular velocidad

Nombre:	Calcular velocidad
Actores:	
Precondiciones:	1. Existe una secuencia nueva de imágenes cargadas en memoria
Flujo Principal:	1. El caso de Uso Iniciar sistema AMIVSI realiza una llamada a calcular velocidad 2. Punto de inclusión 1: Caso de uso “Calcular Optical Flow” 3. [Se ha generado lista de vectores de desplazamiento] Se calcula a partir de los vectores de desplazamiento de la imagen la velocidad global del vehículo
Flujo Alternativo:	3. [No se ha generado lista de vectores de desplazamiento] Se finaliza el caso de uso
Poscondiciones:	1. Se ha calculado la velocidad global del vehículo

Tabla IV-4 Caso de Uso Calcular velocidad.

3.6 Caso de Uso Calcular Profundidad

Nombre:	Calcular Profundidad
Actores:	
Precondiciones:	<ol style="list-style-type: none">1. Existe una secuencia nueva de imágenes cargadas en memoria2. Se ha determinado un conjunto de vectores de desplazamiento
Flujo Principal:	<ol style="list-style-type: none">1. El Caso de Uso Iniciar Sistema AMIVSI realiza una llamada a Calcular Profundidades2. Se elige un vector del Optical Flow3. Se calcula la profundidad al punto que representa el vector4. [No es el ultimo vector] Volver al paso 2
Flujo Alternativo:	<ol style="list-style-type: none">4. [Es el ultimo vector] Se finaliza el caso de uso.
Poscondiciones:	<ol style="list-style-type: none">1. Se ha generado un mapa de profundidades

Tabla IV-5 Caso de Uso Calcular Profundidad.

3.7 Caso de Uso Estimar Posición de un Objeto

Nombre:	Estimar Posición de un Objeto
Actores:	
Precondiciones:	1. Existe una secuencia nueva de imágenes cargadas en memoria 2. Se ha determinado un conjunto de vectores de desplazamiento 3. Existe un objeto en la lista de objetos reconocidos
Flujo Principal:	1. El Caso de Uso Etiquetar un Objeto realiza una llamada a Estimar Posición de un Objeto. 2. Con la posición del objeto reconocido se buscan los vectores de desplazamiento más próximos a las esquinas superior izquierda e inferior derecha. 3. [Hay al menos un vector] Se calcula un vector de desplazamiento mediante la suma de los vectores seleccionados 4. La estimación se calcula desplazando la posición del objeto tantos píxeles en horizontal y en vertical como indique el vector.
Flujo Alternativo:	3. [No hay ningún vector] La estimación de la posición del objeto se sitúa en el mismo sitio donde se encuentra el objeto.
Poscondiciones:	1. Se ha determinado la posición del objeto en la siguiente imagen

Tabla IV-6 Caso de Uso Estimar Posición de un Objeto.

3.8 Caso de Uso Etiquetar un objeto

Nombre:	Etiquetar un objeto
Actores:	
Precondiciones:	<ol style="list-style-type: none"> 1. Existe una secuencia nueva de imágenes cargadas en memoria 2. Se ha determinado un conjunto de vectores de desplazamiento 3. Existe un objeto en la lista de objetos reconocidos
Flujo Principal:	<ol style="list-style-type: none"> 1. El Caso de Uso Iniciar Sistema AMIVSI realiza una llamada a Etiquetar un Objeto. 2. [Hay mas objetos] Se comprueba si la posición estimada de algún objeto de la imagen anterior coincide con la posición actual del objeto. 3. [No coincide con una predicción] Se asigna al objeto una etiqueta nueva. 4. Se introduce en la lista de etiquetas la nueva entrada. 5. Volver al paso 2
Flujo Alternativo:	<ol style="list-style-type: none"> 2. [No hay mas objetos] Finaliza el caso de uso.
Flujo Excepcional:	<ol style="list-style-type: none"> 3. [No coincide con una predicción] <ol style="list-style-type: none"> I. Se asigna al objeto una etiqueta nueva. II. Volver al paso 2
Poscondiciones:	<ol style="list-style-type: none"> 1. La lista de objetos etiquetados ha sido actualizada.

Tabla IV-7 Caso de Uso Etiquetar un objeto.

3.9 Caso de Uso Refrescar interfaz

Nombre:	Refrescar Interfaz
Actores:	
Precondiciones:	
Flujo Principal:	1. [Existe al menos una imagen cargada en memoria] Se representa la imagen cargada en memoria sobre el panel de imagen del interfaz.
	2. [Existe la velocidad calculada] Se representa el en velocímetro analógico y digital la velocidad del vehículo
	3. [Existe algún cálculo de profundidad] Se representa las profundidades calculadas
	4. [Existe algún objeto en la lista de objetos etiquetados] Se representa en la imagen el recuadro que engloba al objeto con su etiqueta correspondiente y se actualiza la lista de objetos etiquetados en la tabla del Interfaz
Flujo Alternativo 1:	1. [No existe ninguna imagen cargada en memoria] Finalizar Caso de uso
Flujo Excepcional 1:	2. [No existe la velocidad calculada] Ir al paso 3
Flujo Excepcional 2:	3. [No existe ningún cálculo de profundidad] Ir al paso 4
Flujo Alternativo 2:	4. [No existe ningún objeto en la lista de objetos etiquetados] Finalizar caso de uso.
Poscondiciones:	1. La lista de objetos etiquetados ha sido actualizada.

Tabla IV-8 Caso de Uso Refrescar Interfaz.

4 Diseño detallado de Clases

4.1 Enumeración de clases

1. Clase OpticalFlow:
2. Clase TMatriz:
3. Clase TListaEtiquetas
4. Clase Navegacion
5. Clase ToolPaint
6. Clase Etiquetado
7. Clase Secuenciador
8. Clase TMainForm
9. Clase TImagen
10. Clase TbaseMil

4.2 Diagrama de Clases

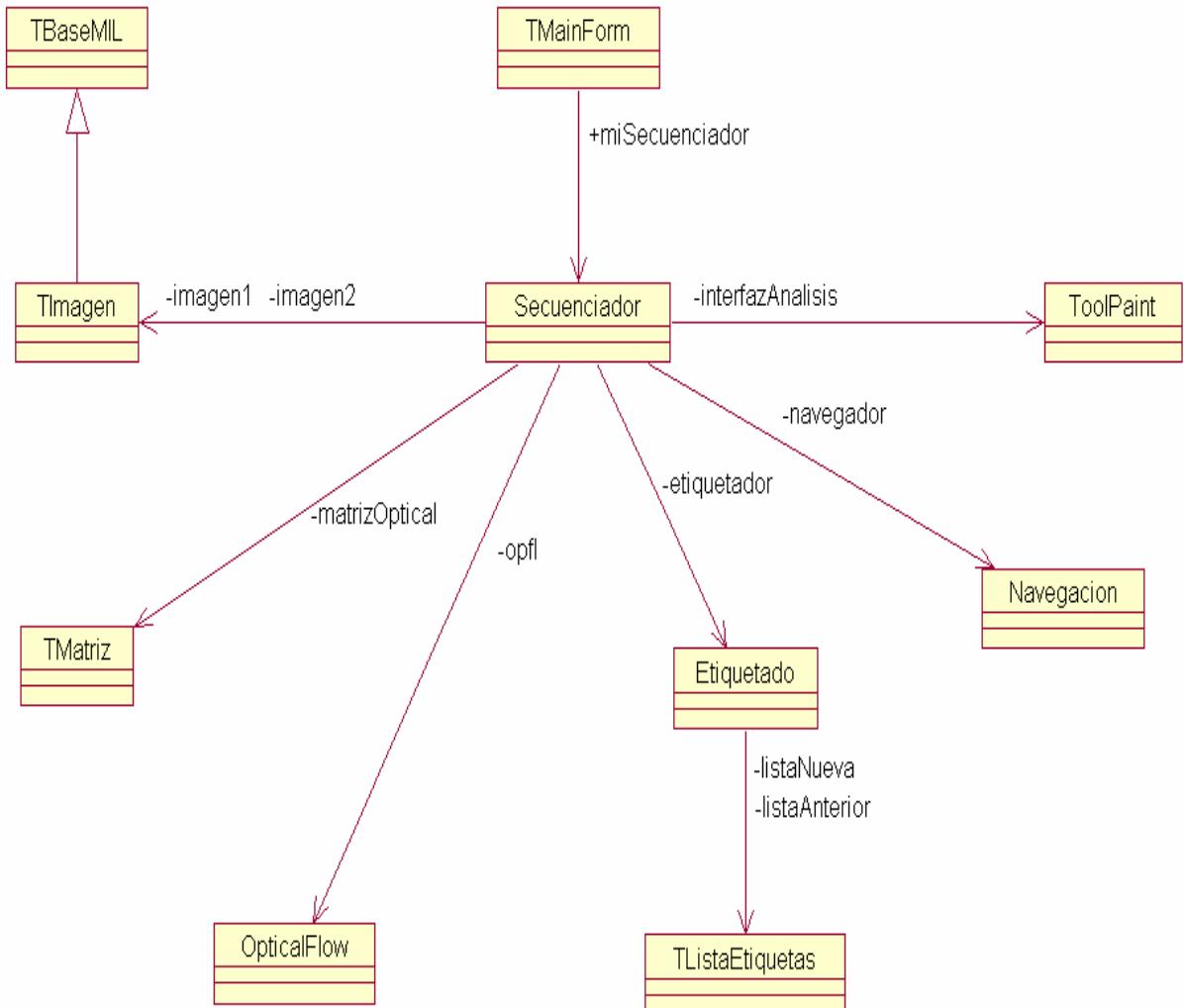


Figura IV-7 Diagrama de clases.

4.3 Clase OpticalFlow

Se encarga de calcular el Optical Flow (campo de velocidades entre dos imágenes)

La clase *Secuenciador* se encarga de crear un objeto de esta clase para utilizar sus funcionalidades con las distintas imágenes que carga del directorio compartido.

A continuación se observa el diseño de dicha clase.

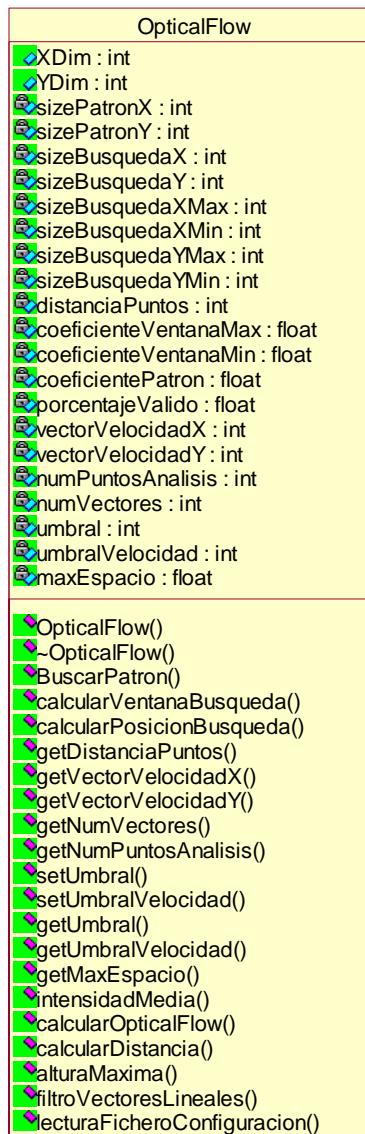


Figura IV-8 Diseño de clase OpticalFlow

Clase OpticalFlow, crea un objeto que existe durante toda la ejecución de la aplicación. Por cada par de imágenes de entrada de una secuencia dada se realiza el calculo de su Optical Flow.

OpticalFlow(): Método Constructor. Inicializa variables desde fichero o bien por defecto.

~OpticalFlow(): Método destructor.

getDistanciaPuntos(): Método que devuelve la variable privada distanciaPuntos.

getVectorVelocidadX(): Método que devuelve la variable privada vectorVelocidadX.

getVectorVelocidadY(): Método que devuelve la variable privada vectorVelocidadY.

getNumVectores(): Método que devuelve la variable privada numVectores.

getNumPuntosAnalisis(): Método que devuelve la variable privada numPuntosAnalisis.

setUmbral(): Método que establece el valor de la variable privada umbral.

setUmbralVelocidad(): Método que establece el valor de la variable privada umbralVelocidad.

getUmbral(): Método que devuelve la variable privada umbral.

getUmbralVelocidad(): Método que devuelve la variable privada umbralVelocidad.

getMaxEspacio(): Método que devuelve la variable privada maxExpacios.

calcularVentanaBusqueda(): Método que calcula el tamaño de la ventana de búsqueda dada una posición dentro de la imagen.

calcularPosicionBusqueda(): Método que calcula el desplazamiento de la ventana de búsqueda dada una posición dentro de la imagen.

BuscarPatron(): Método que determina el PatternMatching para un patrón de la primera imagen y una ventana de búsqueda de la segunda imagen.

intensidadMedia(): Método que calcula la intensidad media de una zona de una imagen.

calcularOpticalFlow(): Método que obtiene un Optical Flow en una estructura matrizOptical a partir de dos imágenes de entrada.

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

alturaMaxima(): Método que busca una máxima distancia en Y entre dos vectores. Además entre estos dos vectores debe haber todos los vectores posibles para poder determinar la altura de un mismo objeto.

filtroVectoresLineales(): Método que filtra vectores a partir de dos matrices de Optical Flow. Si no existe correspondencia lineal entre vectores próximos de ambas matrices se descartan.

calcularDistancia(): Método que a partir de incrementos de dos alturas de un mismo objeto, la velocidad, el tiempo, y la distancia focal de la cámara, estima una distancia al objeto.

lecturaFicheroConfiguracion(): Método que realiza una lectura de un fichero de configuración para establecer las variables iniciales del calculo del Optical Flow.

4.4 Clase TMatriz

Proporciona una estructura de almacenamiento de matriz, mediante punteros con doble indexación.

La clase *Secuenciador* se encarga de crear un objeto de esta clase para poder almacenar los resultados del cálculo del Optical Flow.

A continuación se observa el diseño de dicha clase.



Figura IV-9 Diseño de clase TMatriz

Clase Matriz, que maneja una estructura interna para almacenar un Optical Flow. La estructura utilizada es un array bidimensional de registros con dos enteros. Posee métodos públicos de lectura y escritura para acceder a la información.

TMatriz(): Método constructor, inicializa el tamaño de la matriz dependiendo de la distancia entre puntos y el tamaño de la imagen.

~TMatriz(): Método destructor, borra los punteros a objetos de la estructura.

getDimX(): Método que devuelve la variable privada dimX.

getDimY(): Método que devuelve la variable privada dimY.

getDistanciaPuntos(): Método que devuelve la variable privada distanciaPuntos.

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

getX(): Método que dada una posición en X y otra en Y devuelve el contenido de la coordenada X.

getY(): Método que dada una posición en X y otra en Y devuelve el contenido de la coordenada Y.

setPunto(): Método que dada una posición en X y otra en Y establece las coordenadas de un punto.

reset(): Método que reinicia a 0 todo el contenido de la matriz.

copiar(): Método que copia el contenido de una matriz a otra ya creada previamente.

hayVector(): Método que determina si existe un vector en una posición. Si una de las dos coordenadas contenidas es diferente de 0 existe vector.

4.5 Clase TListaEtiquetas

Proporciona una estructura de almacenamiento de etiquetas, mediante un array de 15 elementos.

La clase *Secuenciador* se encarga de crear dos objetos de esta clase para poder almacenar los resultados del cálculo del Optical Flow.

A continuación se observa el diseño de dicha clase.

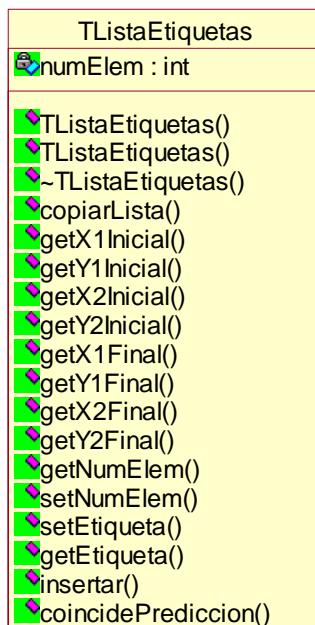


Figura IV-10 Diseño de clase TListaEtiquetas

Clase **TListaEtiquetas**, que implementa el tipo de datos lista de etiquetas sobre un array de dimensión estática, y almacena las etiquetas.

TListaEtiquetas(): Método constructor. Se crea la lista de objetos.

TListaEtiquetas(): Método constructor copia. Se crea una copia de si mismo.

~TListaEtiquetas(): Método destructor. Se libera la lista creada.

copiarLista(): Método que crea una copia de la lista de objetos.

setNumElem(): Método que devuelve la variable privada numElem.

getX1Inicial(): Método que devuelve la variable x1 inicial de la lista de objeto dada una posición.

getY1Inicial(): Método que devuelve la variable y1 inicial de la lista de objeto dada una posición.

getX2Inicial(): Método que devuelve la variable x2 inicial de la lista de objeto dada una posición.

getY2Inicial(): Método que devuelve la variable y2 inicial de la lista de objeto dada una posición.

getX1Final(): Método que devuelve la variable x1 final de la lista de objeto dada una posición.

getY1Final(): Método que devuelve la variable y1 final de la lista de objeto dada una posición.

getX2Final(): Método que devuelve la variable x2 final de la lista de objeto dada una posición.

getY2Final(): Método que devuelve la variable y2 final de la lista de objeto dada una posición.

insertar(): Método que inserta un objeto dada su posición inicial (definida por 4 enteros), su posición final (definida por 4 enteros) y su etiqueta.

setEtiqueta(): Método que establece la variable etiqueta.

coincidePrediccion(): Método que determina si la de un posición objeto (definida por 4 enteros) coincide con otro.

4.6 Clase Navegación

Proporciona la funcionalidad necesaria para implementar el objetivo de cálculo de velocidades

La clase *Secuenciador* se encarga de crear un objeto de esta clase para poder realizar los cálculos necesarios sobre las imágenes y las matrices Optical Flow.

A continuación se observa el diseño de dicha clase.



Figura IV-11 Diseño de clase Navegación

Clase Navegacion, utiliza el calculo del Optical Flow para extraer información de movimiento y lograr así cumplir el objetivo de análisis de la velocidad de movimiento.

Navegacion(): Método Constructor. Se inicializan las variables de velocidad.

~Navegacion(): Método destructor.

getVelocidad(): Método que devuelve la variable privada velocidad.

setVelocidad(): Método que establece el valor de la variable privada velocidad.

distanciaCalibrada(): Método que dado un valor en píxeles devuelve un valor n metros.

calcularVelocidadPromedio(): Método que calcula la velocidad promediada con sus 5 valores anteriores.

calcularVelocidad(): Método que a partir del tiempo y el espacio determina la velocidad y promedia el resultado.

4.7 Clase ToolPaint

Proporciona la funcionalidad necesaria para representar los resultados obtenidos de los cálculos de la aplicación. Realiza actividades de interfaz.

La clase *Secuenciador* se encarga de crear un objeto de esta clase para poder representar la información extraída.

A continuación se observa el diseño de dicha clase.

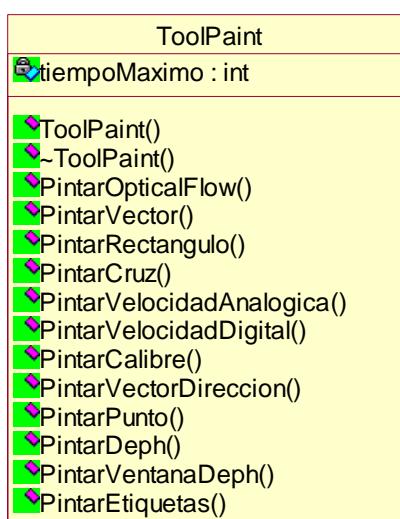


Figura IV-12 Diseño de clase ToolPaint

Clase ToolPaint, crea un objeto que implementa métodos para pintar sobre el interfaz grafico datos de análisis.

ToolPaint(): Método Constructor. Se crea un objeto de la clase ToolPaint que perdura durante toda la ejecución.

~ToolPaint(): Método Destructor.

PintarOpticalFlow(): Método PintarOpticalFLow. Llama al método PintarVector para cada vector que exista en la matriz de entrada.

PintarVector(): Método PintarVector. Dibuja un vector en una imagen. El vector de entrada se caracteriza por cuatro enteros. El punto inicial del vector se representa con un tamaño de punto 3 y así poder distinguir el inicio del fin.

PintarRectangulo(): Se dibuja un rectángulo de color blanco en la imagen dando el punto superior izquierdo, y el inferior derecho.

PintarCruz(): Se pinta una cruz en una posición de la imagen. Se utiliza el color rojo, grosor 3, ancho y alto 31 píxeles.

PintarPunto(): Dibuja en la imagen un punto de color blanco y grosor 5.

PintarVelocidadAnalogica(): Se dibuja una recta que va desde el centro de la imagen del cuentakilómetros hasta una posición que variara dependiendo de la velocidad de entrada.

PintarVelocidadDigital(): Se escribe en una etiqueta la velocidad de calculada dándole un formato.

PintarVectorDireccion(): Dado un vector dirección se determina su inclinación en la imagen.

PintarCalibre(): Se dibujan 4 líneas horizontales que corresponden a las distancias calibradas.

PintarVentanaDeph(): Se realiza el calculo de profundidad para cada vector de la matriz. Se almacena un valor máximo que actúa como escala. Se crean 6 tonalidades diferentes que se le asignaran a los vectores según su posición en la escala.

PintarDeph(): Es el mismo que el pintarVentanaDeph pero se muestran los resultados en la imagen principal.

PintarEtiquetas(): Dada una lista de etiquetas, se insertan los datos de la etiqueta en la tabla del interfaz grafico. Se escribe la etiqueta generada y se pinta un rectángulo en la posición del objeto.

4.8 Clase Etiquetado

Proporciona la funcionalidad necesaria para la consecución del objetivo de etiquetado de objetos previamente reconocidos por el proyecto SACAT.

La clase *Secuenciador* se encarga de crear un objeto de esta clase para poder utilizar la funcionalidad necesaria para conseguir el etiquetado dinámico de objetos.

A continuación se observa el diseño de dicha clase.

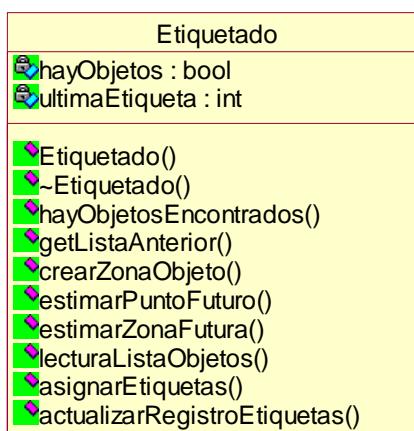


Figura IV-13 Diseño de clase Etiquetado

Clase Etiquetado que utiliza el calculo del Optical Flow calcular posiciones futuras de los objetos en la siguiente imagen y lograr así cumplir el objetivo de etiquetado de objetos.

Etiquetado(): Método Constructor. Se inicializan las variables y se crean dos listas de objetos.

~Etiquetado(): Método Destructor.

getListaAnterior(): Método que devuelve la lista de objetos etiquetados.

hayObjetosEncontrados(): Método que devuelve booleano diciendo si ha encontrado o no objetos

crearZonaObjeto(): Método para almacenar donde se ha encontrado un objeto.

estimarPuntoFuturo(): Método que dada una matriz Optical Flow y un punto del objeto determina su posición futura.

estimarZonaFutura(): Método que dada una matriz Optical Flow determina la posición futura de un objeto.

lecturaListaObjetos(): Método que lee de fichero objetos reconocidos y los inserta en una lista para su posterior etiquetado.

asignarEtiquetas(): Método que asigna a los objeto de una lista la misma etiqueta que tenían antes o una etiqueta nueva.

actualizarRegistroEtiquetas(): Método que registra en un fichero los objetos etiquetados.

4.9 Clase Secuenciador

Representa el cuerpo principal de la aplicación, donde se realizan las llamadas y creaciones de objetos necesarios para la obtención de resultados.

La clase TForm se encarga de lanzar este hilo de ejecución principal.

Posee dependencias con las siguientes clases: TImagen, TMatriz, OpticalFlow, Navegacion, ToolPaint, Etiquetado.

A continuación se observa el diseño de dicha clase.



Figura IV-14 Diseño de clase Secuenciador

Clase Secuenciador, crea un hilo de ejecución para realizar el bucle de análisis del sistema AMIVSI. Es necesario este hilo para ejecutar de forma independiente, el control de la aplicación, de su interfaz. Realiza la creación de objetos necesarias para la consecución de los tres objetivos principales. Análisis del Optical Flow, análisis de la velocidad de movimiento, análisis de la profundidad y etiquetado de objetos.

Secuenciador(): Método Constructor del Secuenciador heredando las características de un hilo.

Execute() : Método que se ejecuta al crear un objeto de tipo secuenciador. Este método contiene el bucle principal que recorre todas las imágenes de la carpeta *img* y realiza las llamadas a los métodos de cálculo y de representación de los objetivos propuestos.

4.10 Clase TMainForm

Clase “main”, se encarga de lanzar el hilo de ejecución principal Secuenciador, así como de crear los objetos de interfaz

La clase TForm se encarga de lanzar este hilo de ejecución principal.

A continuación se observa el diseño de dicha clase.



Figura IV-15 Diseño de clase TForm

Clase TMainForm, crea un objeto que contiene los atributos principales del interfaz. Tras la creación de este objeto se construye un objeto secuenciador para crear un hilo de ejecución diferente del interfaz.

TMainForm(): Método Principal. Se inicializa la tabla de objetos y se crea un hilo Secuenciador y se le pasan referencias a los elementos gráficos.

FormKeyPress(): Método reservado del Programador/Testeador

4.11 Clase TImagen

Proporciona la funcionalidad necesaria para trabajar de manera sencilla con imágenes y objetos de la librería MIL.

A continuación se observa el diseño de dicha clase.



Figura IV-16 Diseño de clase TImagen

Clase TImagen, proporciona métodos para facilitar el acceso a las librerías MIL ya que las imágenes se almacenan en estructuras internas y se necesitan convertir para poder ser visualizadas en el interfaz.

TImagen(): Método constructor, inicializa una imagen MIL.

~TImagen(): Método destructor, libera una imagen si no se libero antes.

Cargar(): Método que importa desde archivo una imagen y la carga en memoria.

convertir(): Método que convierte una imagen color a blanco y negro para su procesamiento en las MIL.

Sobel(): Método que realiza una convolución Sobel a la TImagen

Copiar(): Método que crea una TImagen desde MIL, copia contenido.

GetBitmap(): Método que convierte la imagen a un objeto TBitmap.

Raw2GreyBitmap(): Método que recibe un array con los píxel de la imagen, y su dimensiones. Devuelve referencia a un objeto TBitmap. Liberar después.

Raw2ColorBitmap(): Método que recibe tres arrays con los píxel de la imagen, y su dimensiones. Devuelve referencia a un objeto TBitmap. Liberar después.

TImagen2RawBN(): Método que saca de la TImagen una matriz 2d con los valores de los píxeles en bruto Liberar después la fila 0 y luego la total.

4.12 Clase TBaseMil

Proporciona la funcionalidad necesaria para inicializar correctamente los parámetros de la librería MIL.

A continuación se observa el diseño de dicha clase.



Figura IV-17 Diseño de clase TBaseMil

Clase TBaseMil, base para inicializar los recursos de las MIL. Las variables miembro son static para que las clases que deriven de ella tengan acceso inmediato a estos recursos.

TBaseMil(): Método Constructor. Se inicializa el sistema MIL si no se ha inicializado ya.

~TBaseMil(): Método Destructor. Se liberan el sistema MIL si no existen mas imágenes.

5 Diagrama de Secuencia

Diagrama de secuencia que muestra las colaboraciones entre los distintos objetos, siguiendo el hilo principal de ejecución del sistema AMIVSI

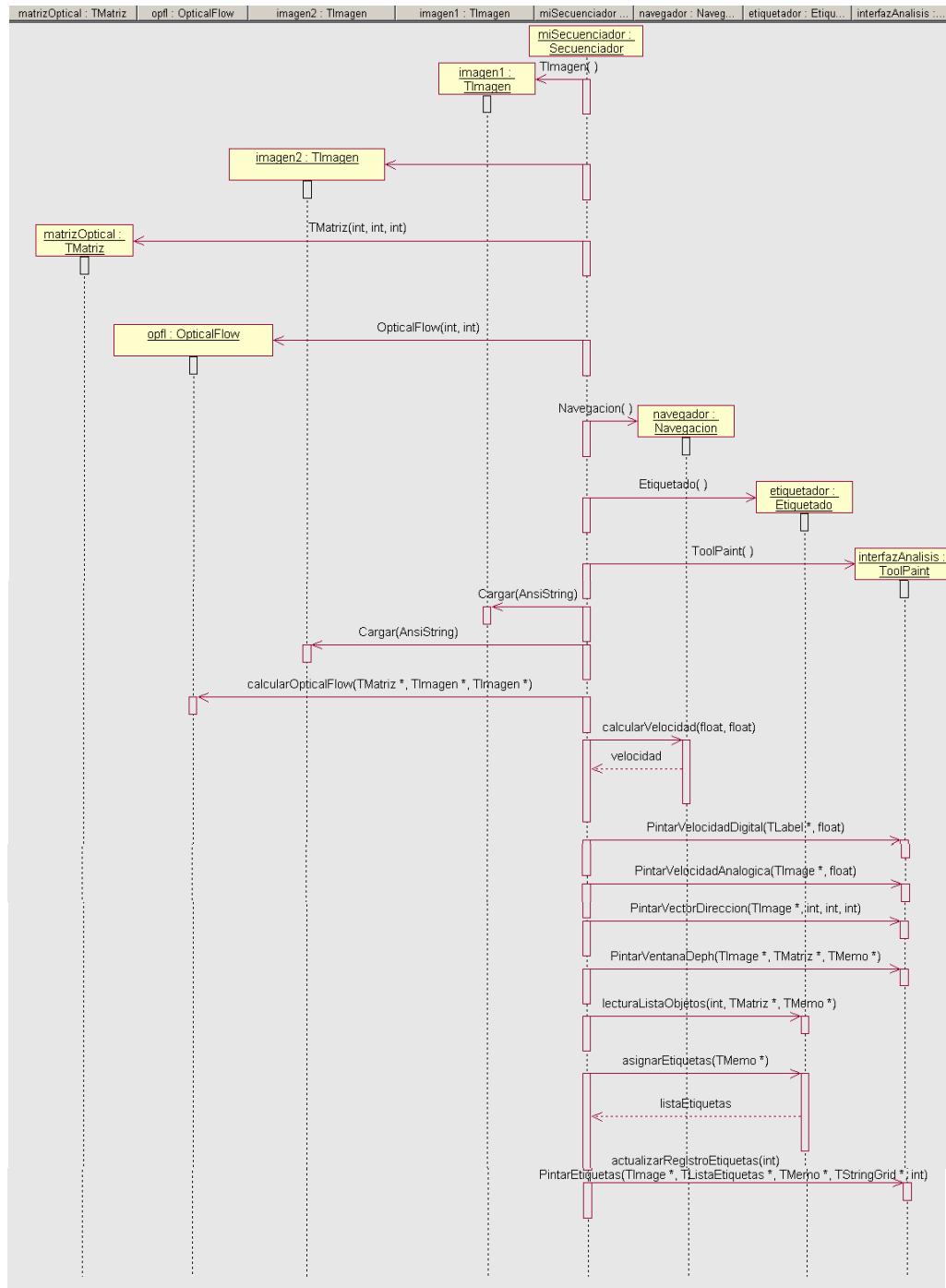


Figura IV-18 Diagrama de secuencia del sistema AMIVSI

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Observamos como se realizan las creaciones de objetos y la utilización de las propiedades que ofrece cada uno. La finalidad es comprender como colaboran los objetos entre si para conseguir los resultados esperados.

Téngase en cuenta que el hilo de ejecución contempla en la realidad un bucle de la secuencia descrita, salvo los pasos de creación de objetos. El bucle finaliza cuando se hayan analizado todas las imágenes de la secuencia.

6 Diseño y análisis de módulos del sistema

6.1 Introducción

El objetivo de este punto es el de dar una visión global del conjunto de las técnicas aplicadas en este proyecto. Más adelante se detallará la metodología, secuencia de pasos y estrategias empleadas para conseguir los objetivos.

Para una mayor compresión de los pasos seguidos, se puede consultar el diagrama de módulos del sistema que aparece en el siguiente apartado.

Como fuente de nuestros análisis crearemos una serie de secuencias de imágenes extraídas a partir de videos. La adquisición de los videos se realiza utilizando un vehículo con una cámara en su interior.

El primer paso tras iniciar la aplicación, es la carga en memoria de la primera y segunda imagen de la secuencia.

Se le aplica un filtro de convolución tipo Sobel a dichas imágenes para detectar solo los bordes de las formas que aparecen en la imagen. Con esto se consigue descartar mucha información que no vamos a necesitar para nuestros análisis, como puede ser el color, las texturas etc.

Hay que seleccionar puntos para realizar los posteriores análisis puesto que la carga de proceso es muy elevada. El sistema selecciona un numero configurable de puntos para ser analizados sobre la primera imagen. Dichos puntos están repartidos uniformemente por toda la imagen. Es necesario reducir aun más el número de puntos de análisis, por lo que se comprueba si cada punto seleccionado supera un umbral. Debido a que hemos filtrado la imagen en el paso anterior, solo serán los bordes de la imagen los que superen dicho umbral.

En el siguiente bloque se crea un patrón de búsqueda por cada uno de los puntos seleccionados. Este patrón almacena el punto seleccionado y los que le rodean hasta una distancia previamente definida. Estos píxeles forman el entorno de vecindad del píxel seleccionado y representa un área de la imagen.

El PatternMatching es el proceso más importante dentro del cálculo del Optical Flow. Este proceso debe ser optimizado al máximo debido a su enorme consumo de tiempo de cálculo. Dicha técnica se basa en buscar coincidencias en un área de búsqueda de la imagen dado un patrón. Para conseguir el mínimo número de comparaciones posibles entre el patrón y el área de búsqueda, esta última debe de tener un tamaño y un desplazamiento variable en función de la posición que ocupe el punto seleccionado en la imagen. En nuestro sistema se dice que el PatternMatching ha detectado una correspondencia, si se encuentran coincidencias

entre el patrón de la primera imagen y una zona de dentro del área de búsqueda de la segunda imagen.

En este caso el sistema almacena el punto seleccionado de la primera imagen y el punto encontrado en la segunda imagen. Las posiciones de estos puntos se almacenan en forma de vector en una matriz de correspondencias de puntos entre dos imágenes. Esta matriz debe de poseer el mínimo número de registros para que su recorrido sea lo más rápido posible.

Una vez que se ha terminado de procesar y registrar todos los puntos de una imagen, se realizan actividades de interpretación de los datos obtenidos. La primera es la de estimar el vector dirección, que no es mas que el modulo de la media de todos los vectores de la matriz para las coordenadas X.

La siguiente estimación es la velocidad, que se calcula a partir de datos calibrados previamente. Se determina una zona de análisis en la que hay calculadas distancias reales del suelo por donde el vehículo se desplaza. Un vector dentro de esta zona debe tener un tamaño proporcional a las medidas calibradas, y como el tiempo entre dos imágenes es conocido, es posible estimar la velocidad.

La estimación de profundidad, se realiza para cada vector almacenado, y se determina sobre un eje de coordenadas dividiendo la distancia de un vector al foco óptico de expansión y su incremento de tamaño. El resultado es el tiempo que falta para alcanzar un punto.

Para etiquetar los objetos, es necesario un fichero de entrada con la posición de los objetos reconocidos por el sistema SACAT. Dada una posición de un objeto se busca el vector más cercano almacenado y su desplazamiento indica donde se debe encontrar el punto en la siguiente imagen. Si el punto del nuevo objeto reconocido en la siguiente imagen se encuentra próximo al punto destino del vector, el objeto es el mismo y se le asigna la misma etiqueta.

6.2 Generación de secuencia de imágenes

La necesidad de generar una batería de imágenes propia surge debido a que la interconexión con el sistema SACAT no es real.

La generación de secuencias de imágenes se realiza utilizando una cámara de video doméstica situada en el interior de un vehículo en movimiento.

Una vez capturado el video se procede a su digitalización, utilizando para el efecto un programa de captura de video. Tras disponer del video en formato digital se realiza un tratamiento de desentrelazado, debido a que la cámara utilizada no dispone de Scan Progresivo.

El proceso de desentrelazado consigue mejorar la calidad de la imagen duplicando las líneas pares, rellenando así la información necesaria para representar la imagen correctamente.

Veamos a continuación una imagen que muestran el resultado



Figura IV-19 Diferencias entre entrelazado y desentrelazado.

Por ultimo se convierte el video en una secuencia de imágenes con las características detalladas en la ERS: imágenes a color, resolución de 720 x 576 píxeles, formato mapa de bits (bmp), a una frecuencia de 5 frames por segundo.

6.3 Diagrama de módulos del sistema

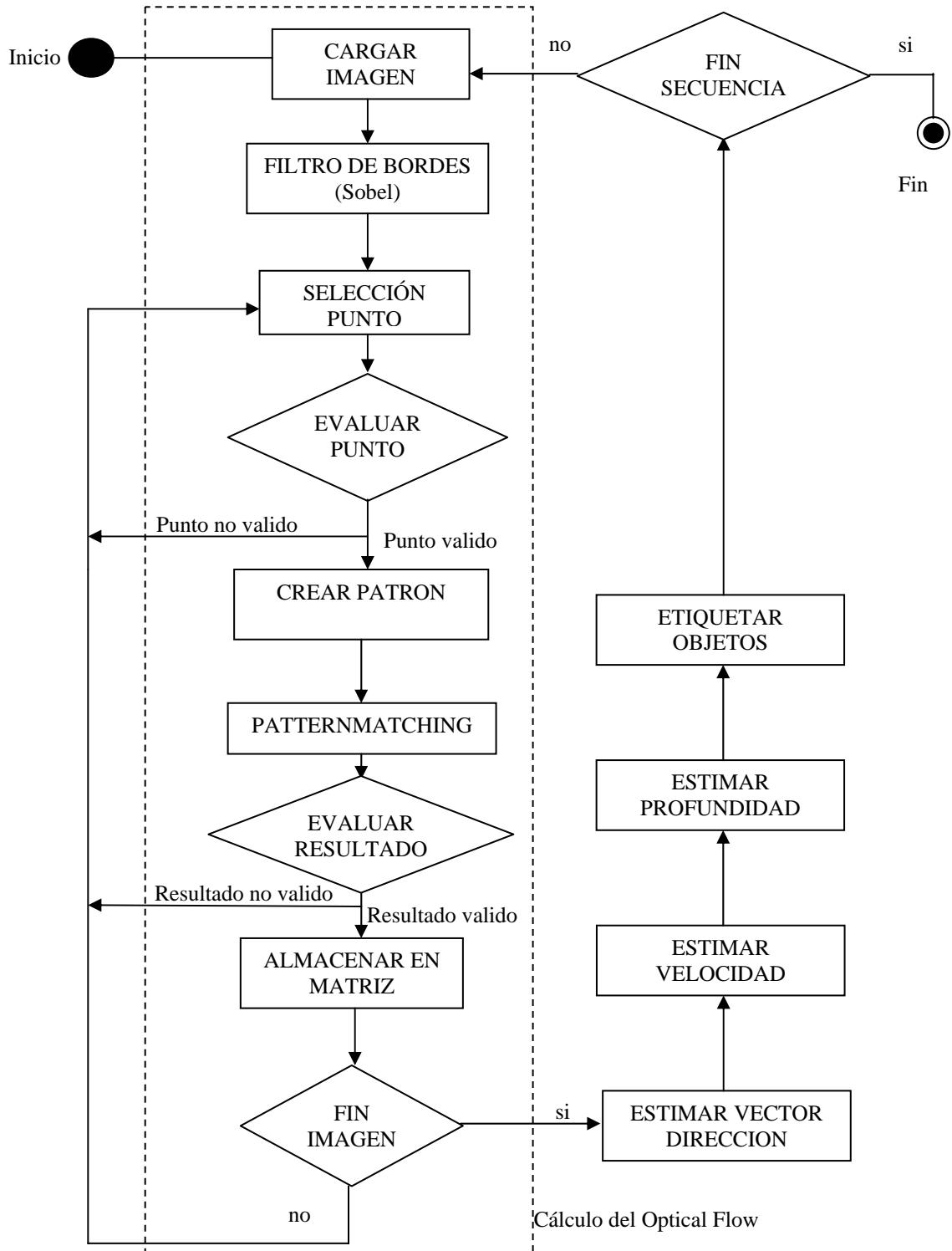


Figura IV-20 Diagrama de módulos del sistema

6.4 Cálculo del Optical Flow

La base del análisis de del movimiento es la técnica del estudio de velocidades, Optical Flow, que consiste en representar el desplazamiento de los puntos entre dos imágenes tomadas con un intervalo de tiempo. El campo de velocidades de dos imágenes se determina a partir de repetir la misma operación para un número considerable de puntos de la imagen.

En este apartado se detalla la solución propuesta para el problema del cálculo del Optical Flow, que será necesario para la consecución de los objetivos concretos. Como se puede apreciar en el diagrama de módulos del sistema el cálculo del flujo óptico se comprende de varios módulos relacionadas entre si. La colaboración de los distintos módulos permite elaborar un resultado.

Se ha establecido una clasificación de los módulos dependiendo de la tarea que desempeñan. Tenemos una primera fase de adquisición de imágenes, otra de técnicas de filtrado y por último el procesado y almacenamiento.

Antes de iniciar el análisis de la primera imagen de la secuencia se procede a la lectura del fichero de configuración, que carga en memoria parámetros relevantes para el cálculo del Optical Flow. Los parámetros son los siguientes: *coeficiente_ventana_max*, *coeficiente_ventana_min*, *coeficiente_patron*, *umbral*, *espacio_entre_vectores* y *porcentaje_valido*. El formato del fichero de configuración se ha especificado en la ERS, la utilización de cada uno de estos parámetros se explicará en los módulos pertinentes.

6.4.1 Primera Fase: Adquisición de Imágenes

Descripción por el módulo de Cargar Imagen, es en esta fase donde se recuperan las imágenes almacenadas en el disco duro y se cargan en memoria para trabajar con la matriz de puntos que las representan. Cada imagen posee 720 x 576 píxeles que se almacenan en forma de matriz bidimensional. Cada uno de los píxel contiene información de intensidad de tres colores primarios, rojo, verde y azul. El formato de las imágenes es Mapa de Bits (*bmp*).

Este módulo conoce en todo momento cual es la siguiente imagen que se espera recibir y procede a su carga cuando recibe la orden por parte del hilo principal de ejecución.

El proceso de cargar en memoria se realiza gracias a las funcionalidades que incorporan las librerías MIL, que permiten recuperar una imagen y almacenarla en un *Buffer* identificado mediante un *Mil_ID*, indicando la ruta y el nombre del archivo que se desea recuperar. De esta manera tendremos siempre accesible la imagen y lista para procesarla con cualquiera de las posibilidades que proporcionan las MIL.

6.4.2 Segunda Fase: Técnicas de Filtrado

El objetivo fundamental de esta fase es el tratamiento de la imagen para extraer la información relevante y eliminar todo aquello que no nos interese de cara al análisis. Es obvio que la manera de obtener los mejores resultados sería no desperdiciar ningún píxel de la imagen, sin embargo es necesario realizar un filtrado debido a las restricciones de tiempo y sobre todo a la excesiva carga de trabajo que requieren el procesado de imágenes.

1) *Filtrado de Bordes*

Es la primera técnica de filtrado empleada y debe permitirnos resaltar las zonas de la imagen que contengan un borde, se entiende por borde a una notable variación del gradiente de la imagen.

La imagen resultado que esperamos obtener estará representada mediante escala de grises. El fondo de la imagen será negro y todo lo que se considere un borde se representará utilizando colores próximos al blanco. Dependiendo del borde detectado se mostrará con un color mas intenso si se trata de uno muy pronunciado, o más tenue si se trata de un borde difuminado.

La metodología para alcanzar dichos resultados se denomina convolución de la imagen, y consiste en recorrer el área de la imagen con una máscara de pesos, la máscara describe unas operaciones matemáticas que irán transformando los valores de los píxeles en función de su entorno de vecindad.

El filtro utilizado se denomina SOBEL [GONZALEZ 1999] y consiste en la combinación de dos máscaras de convolución, una para la detección de bordes verticales, y otra para los horizontales. La combinación de ambas máscaras se establece como la suma de los valores absolutos de los resultados parciales. Cada una de las máscaras representa el gradiente y el filtrado de la media ponderada en el entorno. De esta manera se realiza al mismo tiempo la tarea de detección de bordes y un suavizado.

Veamos a continuación el contenido de dichas máscaras:

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Figura IV-21 Máscaras de convolución SOBEL.

Para comprender en profundidad el filtrado de bordes veamos a continuación el resultado obtenido tras aplicar dicho filtro a una imagen:



Figura IV-22 Imagen original sobre la que se aplicará un filtro SOBEL

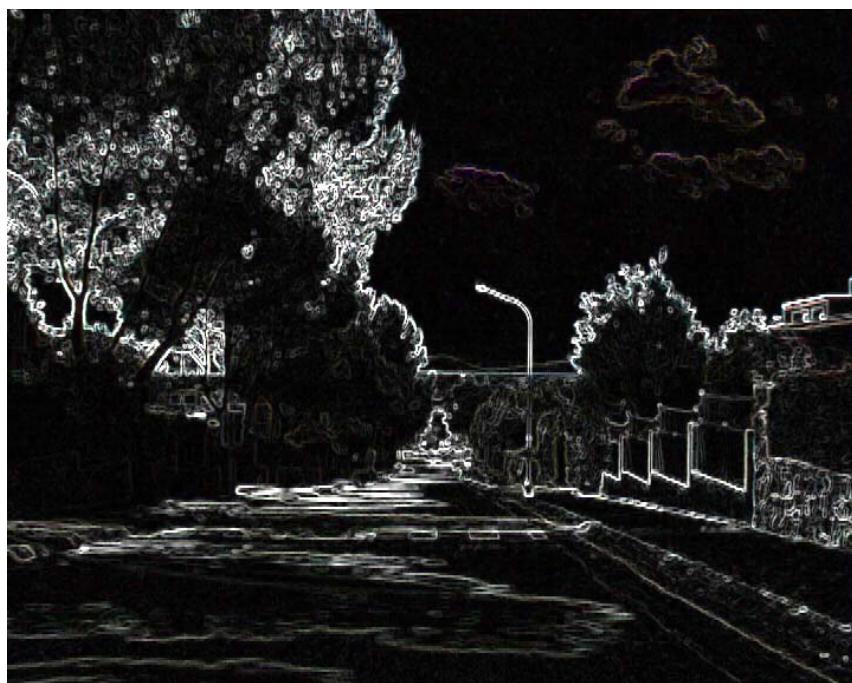


Figura IV-23 Imagen tras aplicar un filtro SOBEL

En la imagen *Figura IV 23* se puede apreciar con tonalidades claras los bordes que han sido detectados, y con fondo negro todo aquello que no es susceptible de análisis. Es importante resaltar la relevancia que el filtro tendrá a lo largo del análisis de las imágenes, es el primer filtro que aplicaremos con el fin de resaltar las zonas de análisis de la imagen. Consiguiéndose una reducción notable de la información a procesar.

2) *Selección del punto de análisis*

La siguiente técnica de filtrado empleada es la elección de los puntos de la imagen que serán analizados. Teniendo en cuenta que la imagen tiene una resolución de 720 x 576 píxeles, contamos con un total de 414720 puntos de análisis, realizar las operaciones sobre todos los puntos de la imagen dispararía el tiempo de cálculo. La técnica de selección de puntos se divide en dos fases: creación de rejilla uniforme de puntos, y discriminación de puntos entorno a un umbral.

3) *Creación de una rejilla uniforme*

Consiste en superponer a la imagen una rejilla uniforme de puntos, a modo de plantilla de selección. La creación de dicha rejilla se realiza comenzando por la esquina superior izquierda de la imagen, e incrementando un valor predeterminado llamado distancia entre puntos, en el eje *x* e *y*. El valor de la distancia entre puntos se lee a partir de fichero de configuración, en donde recibe el nombre de *espacio_entre_vectores*. El resultado sería la selección de puntos de la imagen separadas cada uno de ellos el valor de la distancia entre puntos.

A continuación superponemos la rejilla uniforme ante la imagen de análisis:

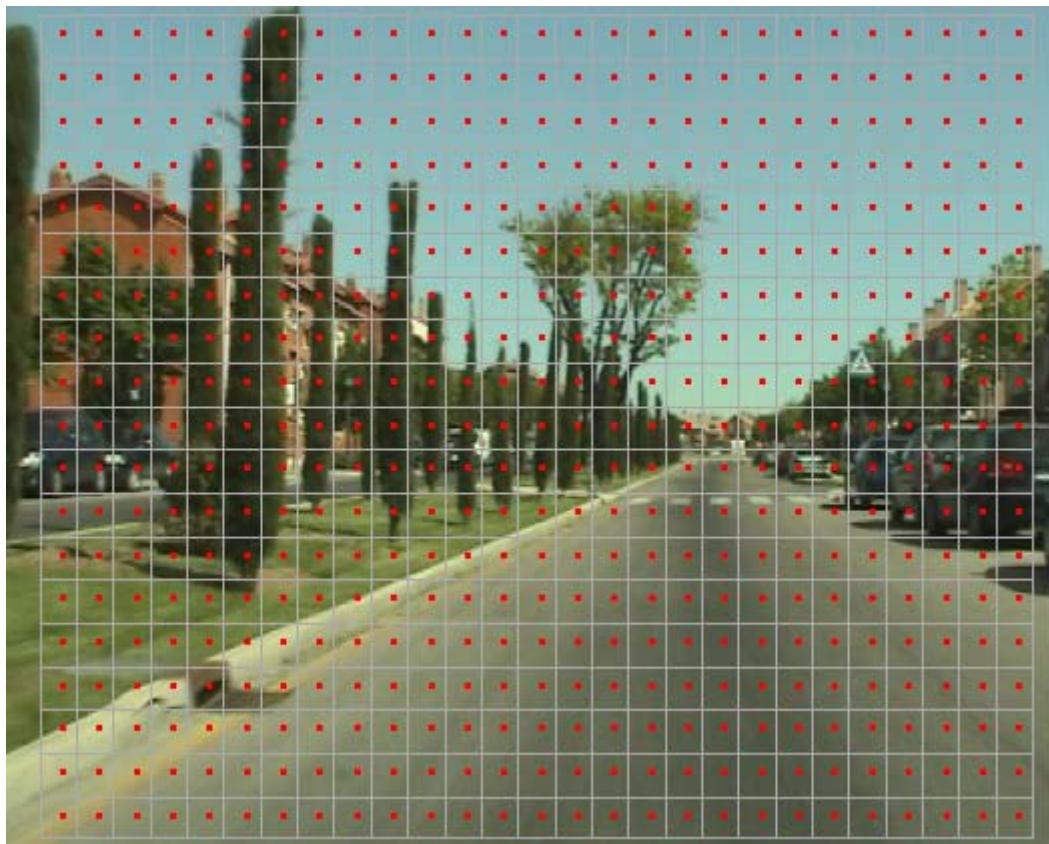


Figura IV-24 Superposición de imagen con rejilla uniforme.

Se observa como la cantidad de puntos de análisis se ha reducido considerablemente, ya que la rejilla actúa a modo de plantilla de elección, repartiendo de forma uniforme los puntos que serán analizados.

La distancia entre puntos, que inicialmente está configurada entre 5 y 7 píxeles, determina en gran medida el número de puntos de análisis.

4) Discriminación de puntos entorno a un umbral

Se realizará una evaluación de los puntos de análisis entorno a un umbral determinado, dicho umbral se lee desde el fichero de configuración. La discriminación se realiza individualmente para cada punto y consiste en comparar el nivel de intensidad del punto de análisis, con el umbral establecido. Si el punto supera el umbral se procesarán cálculos sobre él, de lo contrario será descartado.

La elección del umbral es una de las tareas críticas del sistema, ya que influye de forma directa en los resultados finales y en el tiempo de análisis. Con el fin de satisfacer una

necesidad para el objetivo de estimación de velocidad existen dos umbrales de discriminación, esta diferenciación se explicará en el modulo de cálculo de velocidad.

Para comprender mejor el comportamiento del umbralizado veamos una imagen a la cual le aplicaremos una umbralización:



Figura IV-25 Imagen original antes de realizarle una umbralización.



Figura IV-26 Imagen tras aplicarle un umbral igual a 75.

Combinando las técnicas de filtrado mediante rejilla uniforme de puntos y umbralización se consigue la elección de puntos interesantes para su posterior análisis, reduciendo así la cantidad de información de la imagen, y seleccionando aquellas zonas relevantes.

A continuación se muestra el resultado de combinar ambas técnicas y como su resultado se ajusta a las zonas de interés.



Figura IV-27 Distribución de puntos en zonas de interés mediante técnicas de filtrado

Se observa como se han descartados las zonas de la imagen que no ofrecen información relevante de la escena, y se han distribuido numerosos puntos de análisis en las secciones de interés.

6.4.3 Tercera Fase: Procesado y Almacenamiento

Una vez realizado el filtrado necesario nos disponemos a procesar la imagen con el fin de obtener los vectores de desplazamiento que conformen el Optical Flow. Esta fase se subdivide en cinco módulos: Crear patrón de búsqueda, PatternMatching, Evaluar resultado, Almacenamiento en matriz, Estimar Vector dirección.

1) *Crear Patrón de Búsqueda*

Se crea un patrón de búsqueda a partir de un punto que ha sido previamente filtrado y evaluado. El objetivo es crear un patrón que englobe al punto y a su entorno de vecindad, de manera que sirva para identificar las características de la zona en la que se encuentra, y posteriormente buscar una zona en la siguiente imagen con características similares.

El patrón resultante se describe mediante un rectángulo entorno al punto, de tamaño predeterminado por el parámetro de configuración *coeficiente_patron*, este parámetro se multiplica por las dimensiones de la imagen para fijar unas proporciones al patrón. El rectángulo selecciona una porción de la imagen y genera un patrón. Dicho proceso se realiza utilizando la funcionalidad *MpatAllocModel* que proporciona las librerías MIL.

El patrón de búsqueda se crea sobre la primera de las dos imágenes utilizadas para el cálculo del Optical Flow.

El siguiente paso es establecer una ventana de búsqueda sobre la segunda imagen de la secuencia. La ventana de búsqueda delimita el área sobre la imagen en la que se intentará encontrar correspondencias utilizando el patrón creado.

La ventana de búsqueda se delimita por medio de dos características, su tamaño y la posición en la imagen.

El tamaño de la ventana de búsqueda es variable dependiendo de la posición del punto y de los parámetros del fichero de configuración *coeficiente_ventana_max* y *coeficiente_ventana_min*; estos parámetros establecen el máximo y el mínimo de una ventana de búsqueda multiplicando sus valores por las dimensiones de la imagen.

Si consideramos el FOE como el centro de la imagen, podremos observar como los vectores de desplazamiento parten de él y se dispersan de manera circular alrededor de él.

Se ilustrará dicho fenómeno mediante la siguiente figura

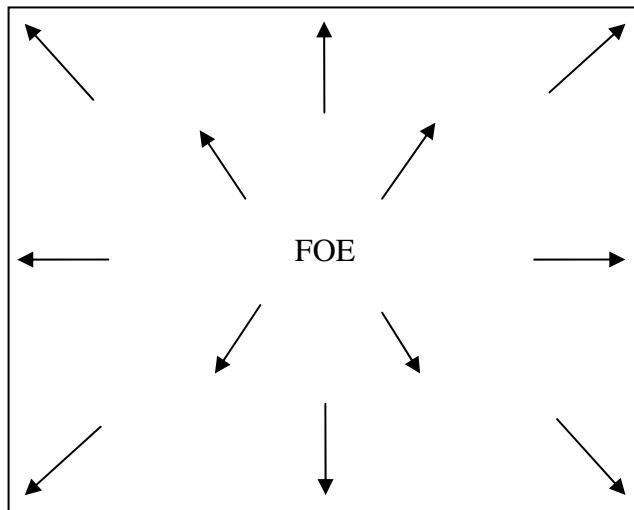


Figura IV-28 Representación de vectores de desplazamiento con respecto al FOE

Observamos como se puede predecir un la dirección desplazamiento según la posición de la imagen. Utilizaremos este concepto para elegir los tamaños de ventana, tanto en el eje x como en y . De esta manera establecemos el tamaño de la ventana de búsqueda será mayor cuanto más alejados nos encontrremos del FOE. Fijando como valor mínimo en x de 57 píxeles, un máximo de 133 píxeles; un valor máximo en y de 106 píxeles y un mínimo de 46 píxeles; obtendremos unos resultados adecuados.

Profundizando en la misma característica que presentan los vectores de desplazamiento podemos deducir que el centro de la ventana de búsqueda no tiene porque coincidir con el punto de análisis. Por lo contrario es lógico que exista un desplazamiento de la ventana de búsqueda en relación con el punto de análisis.

Mediante este mecanismo conseguiremos corregir posibles errores que se produzcan eliminando zonas hacia las que no se dirige el vector de desplazamiento. Utilizaremos un ejemplo concreto para explicar el fundamento de esta técnica.

Consideremos un vector situado en el cuadrante inferior derecho de la imagen, si analizamos el movimiento que a priori describirá dicho vector observaremos que en la inmensa mayoría de los casos la dirección de desplazamiento será hacia abajo y hacia la derecha. Es por ello que será lógico desplazar la ventana de búsqueda hacia la predicción de desplazamiento para obtener mejores resultado.

Veamos una ilustración de lo expuesto:

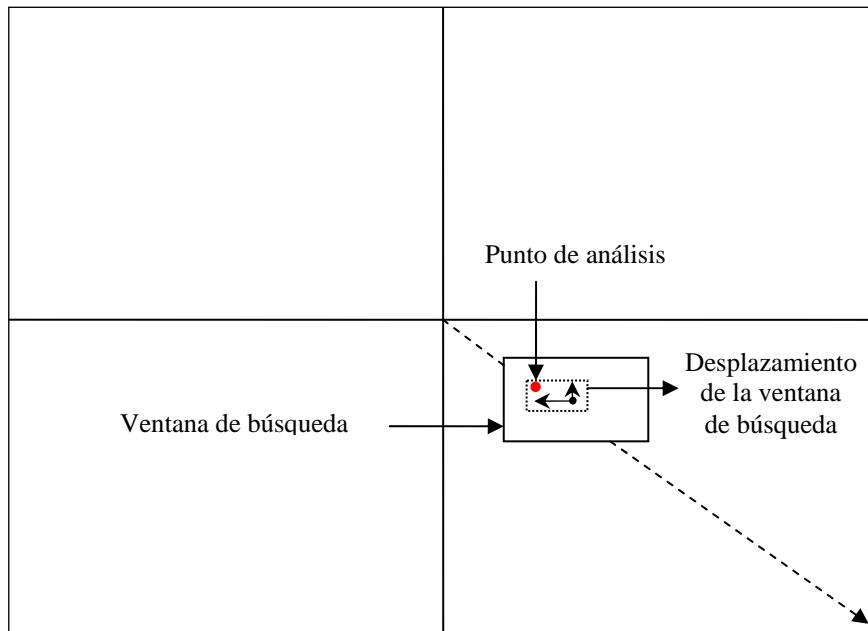


Figura IV-29 Representación de la ventana de búsqueda

2) *PatternMatching*

La técnica de *PatternMatching* se corresponde con el algoritmo de correlación de bloques el cual explicaremos a continuación:

Se parte de la suposición que se conserva la distribución de intensidad de la región que rodea al punto donde se desea evaluar el movimiento. Por lo tanto, para cada punto donde se desea medir el *Optical Flow* en un instante determinado, se crea una ventana de puntos que rodean al mismo (patrón de búsqueda).

Se busca la máxima correspondencia de esa ventana dentro de un conjunto de ventanas de igual tamaño y centradas en los puntos contenidos en una venta de posibles candidatos (ventana de búsqueda).

En la imagen se presenta un esquema ilustrativo del tipo de proceso involucrado en el método de correlación de bloques. En él, se pretende mostrar el procedimiento por el cual se desplaza la ventana de píxel sobre la ventana de búsqueda para obtener una correspondencia.

Como se observa en la *Figura IV 30*, en este ejemplo el patrón de búsqueda es de tamaño 3x3 y la ventana de búsqueda es de tamaño 10x10. Cada respuesta se corresponde a la comparación del patrón con una subventana 3x3 dentro de la ventana de búsqueda.

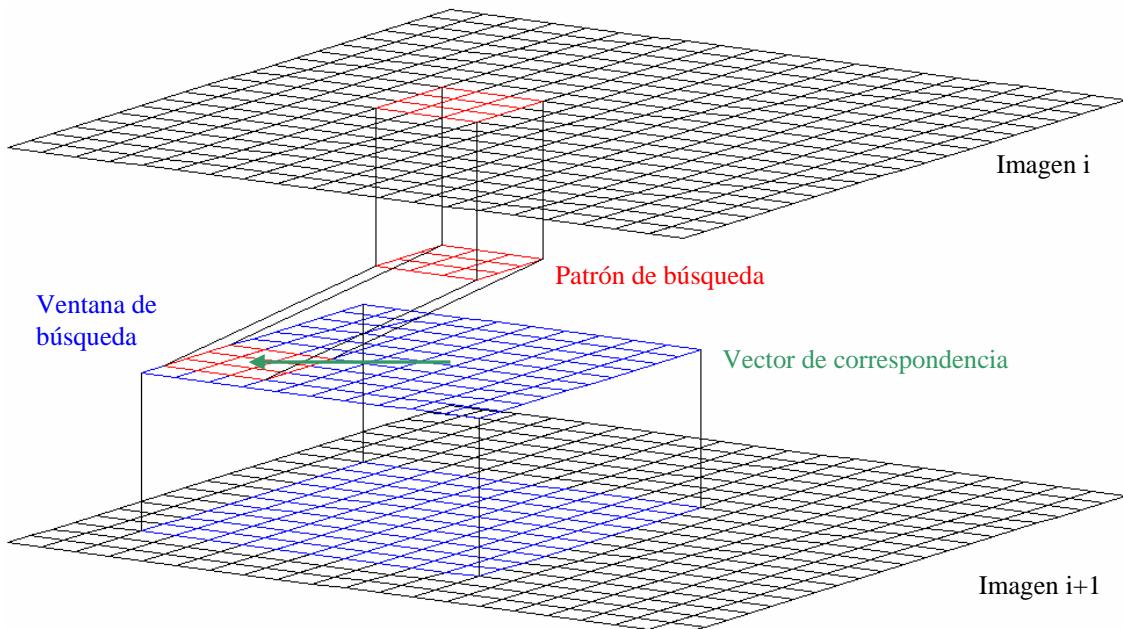


Figura IV-30 Representación del algoritmo de correlación de bloques

La funcionalidad de PatternMatching se realiza mediante las librerías MIL que proporcionan comandos de búsqueda de patrones dentro de una imagen, con la posibilidad de especificar la región de búsqueda y las propiedades de los resultados esperados.

La salida de dicha función especifica la fiabilidad del resultado calculado y la correspondencia determinada en coordenadas globales dentro de la imagen.

3) Evaluar Resultado

En este módulo se realiza una evolución del resultado obtenido por el PatternMatching. La finalidad es desestimar los vectores calculados que no posean un índice de confianza elevado. Una de las propiedades que facilita la función de PatternMatching mediante las librerías MIL es una medida de aceptación, se denomina *Acceptance* que indica el nivel de fiabilidad de la correspondencia obtenida. Estableciendo un límite determinado por el parámetro de configuración *porcentaje_valido* evitaremos incluir en nuestro Optical Flow vectores que no se ajusten a la realidad del desplazamiento.

4) Almacenar en matriz

Cada uno de los vectores de desplazamiento calculados se almacena en una estructura de matriz, en la que en cada posición se indica el origen y final de un vector de desplazamiento. La matriz está ordenada de tal manera que respete la posición de cada vector dentro de la imagen, de tal forma que dispongamos de una estructura de fácil manejo, útil para realizar los cálculos necesarios para la consecución de los objetivos.

Después de estudiar todas las fases de las que se compone el análisis del Optical Flow veamos los resultados obtenidos para un par de imágenes:



Figura IV-31 Optical Flow calculado sobre una secuencia (1)



Figura IV-32 Optical Flow calculado sobre una secuencia (2)

6.5 Estimar el vector dirección

La estimación de las coordenadas del vector dirección se obtienen mediante una función sencilla. Partiendo de la totalidad de vectores encontrados en el Optical Flow, se

realiza una suma de sus tamaños en la coordenada x , y se divide por el número de vectores encontrados.

$$\text{Vector dirección} = - \frac{\sum \text{matrizOptical.Vector}(i).\text{sizeX}}{\text{numeroVectores}}$$

Mediante esta fórmula se obtiene la representación del vector de dirección de giro.

6.6 Estimación de la velocidad de movimiento

La estimación de velocidad de movimiento del vehículo se apoya en el cálculo previamente explicado del Optical Flow, al cual se le introduce una ligera modificación que comentaremos durante este apartado. Con la información de desplazamiento que proporciona el Optical Flow se establece la teoría del movimiento del suelo (carretera, terreno, etc.) frente a la cámara que se refleja en los vectores de una cierta zona de análisis del Optical Flow.

Dicha zona de análisis ha sido previamente calibrada y es el referente para conocer la velocidad con la que se mueve el suelo, por lo tanto la velocidad con la que se desplaza el vehículo.

A continuación se detallan los pasos, técnicas y suposiciones en las que se ha basado la estimación de velocidad

6.6.1 Zona de análisis

La zona de la imagen útil para el análisis de la velocidad de movimiento se puede situar en la franja inferior que abarca el 30% de la imagen, la delimitación de dicha zona se ha establecido realizando un estudio de las imágenes captadas desde la cámara situada en el interior del vehículo, concluyendo que dicha franja engloba la sección de carretera por la que circulamos.

Como se puede apreciar en la siguiente imagen se delimita la zona de análisis al 30% de la franja inferior de la imagen, resultado la zona útil del píxel 400 (vertical) en adelante



Figura IV-33 Zona de análisis de velocidad

La imagen capturada es tan sólo una de una serie de 600 imágenes en las que se aprecia como la línea delimitadora abarca la zona de carretera más próxima al vehículo.

Si observamos detenidamente en la imagen podemos concluir que el 90% aproximadamente de la zona de análisis pertenece a la carretera, eliminando casi en su totalidad aceras, obstáculos, casas, árboles, etc. Disponemos de una superficie que conocemos plana (despreciando cambios de rasante e irregularidades), en la cual nos basaremos para analizar la velocidad.

6.6.2 Calibración de la zona de análisis

Como se ha explicado existe una zona de análisis de interés para el cálculo de la velocidad de movimiento, se ha delimitado dicha zona en la imagen y el siguiente paso consiste en calibrar el sistema para poder interpretar la información que se extraiga del movimiento.

La calibración consiste en la medición y anotación de distancias que refleja una imagen genérica de la secuencia; se supone que nos encontramos en una superficie plana libre de irregularidades.

El proceso de calibración del sistema se divide en los siguientes pasos:

1. Se sitúa el vehículo en una superficie plana y se posiciona la cámara.
2. Medir y marcar una serie de líneas paralelas a la horizontal de la imagen: se miden sobre el terreno 4 distancias paralelas a la horizontal de la imagen, a 5m , 7.5m, 10m y 12.5m. Y se señalan con una marca fácilmente identificable en la imagen.
3. Se captura una imagen de la escena.
4. Se observan correspondencias de las marcas de distancias con su posición vertical en píxeles.
5. Mediante las 4 correspondencias de distancias y píxeles se obtiene interpolando la función que acepte como dato de entrada la posición en píxeles y devuelva su correspondencia en metros.

Función de equivalencias entre píxeles y metros:

$$Y = 0.000188 \cdot X^2 - 0.21995 \cdot X + 69.143297;$$

Veamos una imagen que ilustra parte del proceso de calibración:

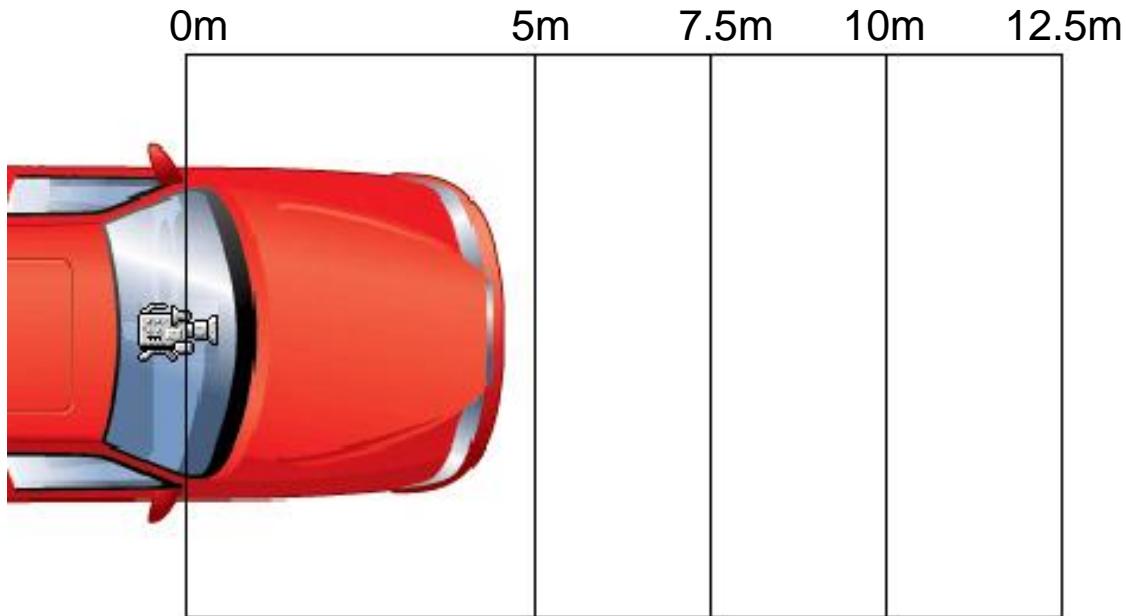


Figura IV-34 Calibración del sistema

6.6.3 Tratamiento del Optical Flow

Como se ha comentado la sensibilidad del análisis del Optical Flow se ve levemente modificada. Dicha modificación surge de la necesidad de poder obtener un número de vectores significativos en la zona de análisis. Debido al filtrado de los puntos de análisis del Optical Flow se eliminan los puntos que no poseen suficiente textura en la imagen; nos encontramos entonces con el problema frecuente de la falta de textura en la zona de la imagen que abarca la carretera, por ser una superficie plana, sin demasiados cambios de tonalidad ni objetos que sobresalgan. Resultando la pérdida de puntos analizados en la zona de interés.

Para aumentar la densidad de puntos analizados en la zona de interés se establece la modificación que afecta al cálculo del Optical Flow, introduciendo dos umbrales discriminatorios, que se corresponden con la zona de análisis y con el resto de la imagen. Como se puede apreciar en la figura se aplicará diferentes umbrales de discriminación en las dos zonas delimitadas de la imagen.

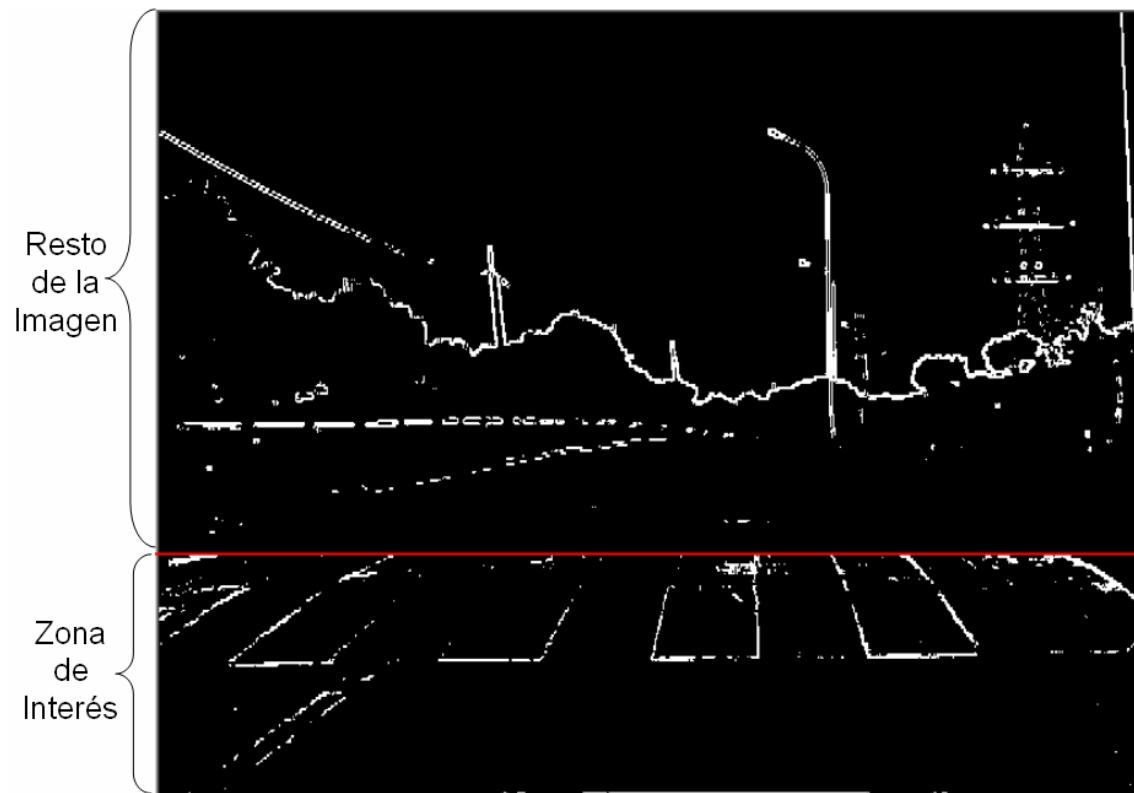


Figura IV-35 Variación del umbral para el cálculo de velocidad.

Para este caso en particular se ha aplicado un umbral de 100 en la zona superior y un umbral de 45 en la zona de interés, se puede apreciar como la densidad de puntos candidatos al análisis aumenta considerablemente al reducir el umbral.

Con esta técnica nos aseguramos la presencia de vectores pertenecientes a la zona de interés en la matriz que almacena el Optical Flow.

6.6.4 Uso del Optical Flow

El uso del Optical Flow se realiza accediendo a las posiciones de la matriz que lo almacena correspondientes con la zona de interés. De la totalidad de vectores encontrados en la zona de interés se selecciona el que posea un mayor desplazamiento en vertical, por entenderse que es el vector que mejor representa el desplazamiento del suelo en dicha secuencia.

6.6.5 Cálculo de la velocidad instantánea

Una vez conocemos el vector candidato que mejor representa el desplazamiento del terreno en la secuencia de imágenes se realiza con él el cálculo de la velocidad, partiendo de la siguiente teoría.

Dicho vector posee unas coordenadas verticales inicial y final, es decir una altura en píxeles, haciendo uso de la ecuación de interpolación descrita en el apartado de calibración se pudo obtener la posición inicial y final en metros de dicho punto. Con una simple operación de resta obtenemos el espacio que ha recorrido el punto que describe el vector. Conociendo la distancia vertical que ha recorrido un punto de la imagen situado en nuestra zona de interés, y el tiempo transcurrido entre la captura de las dos imágenes es posible determinar la velocidad de movimiento de dicho punto en relación con el vehículo. Es importante reseñar que la velocidad calculada es realmente la velocidad con la que se desplaza el vehículo, ya que se supone que no existen puntos móviles en el terreno.

A continuación se detallan los pasos seguidos para conseguir la velocidad instantánea:

1. Se selecciona el vector de mayor desplazamiento vertical dentro de la zona de interés.
2. Se calcula la distancia en metros que se ha desplazado.
3. Se calcula la velocidad instantánea mediante la ecuación $v = e/t$.

Veamos una imagen que ilustra dicho procedimiento

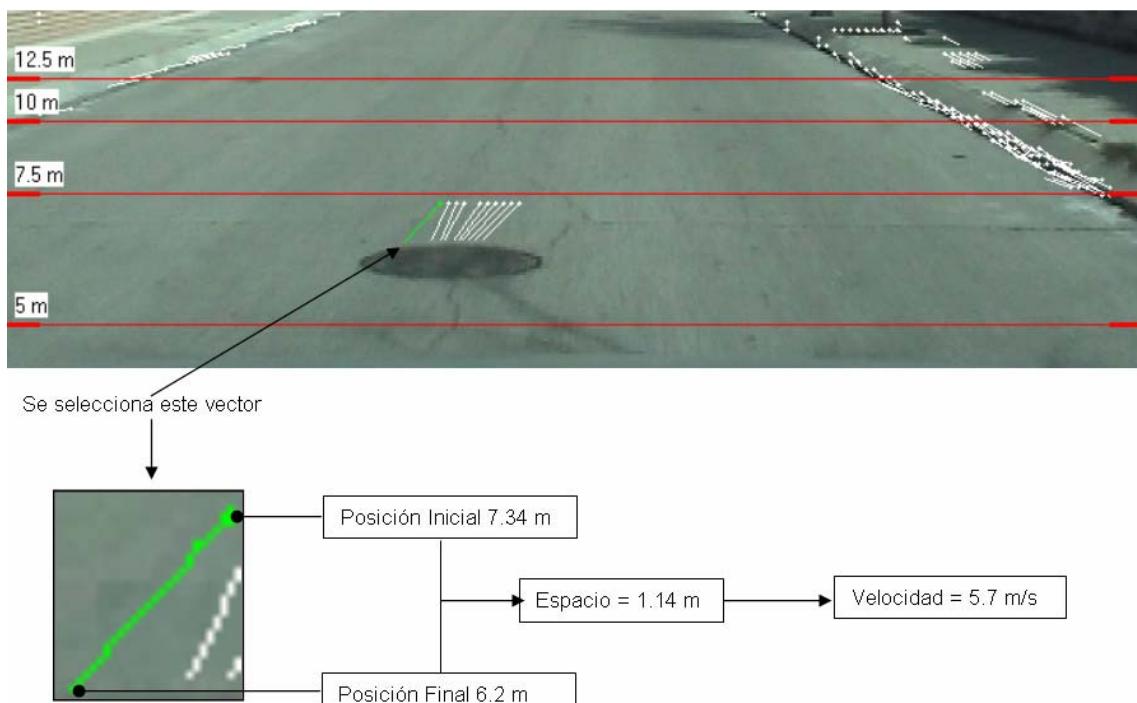


Figura IV-36 Proceso de cálculo de velocidad instantánea.

6.6.6 Cálculo de la velocidad promediada

Como se ha explicado en el apartado anterior ya conocemos la velocidad instantánea correspondiente a una secuencia de dos imágenes tomadas con un intervalo de tiempo determinado. Debido a la cantidad de errores que pueden producirse por diversos motivos en el análisis del Optical Flow es necesario realizar varios cálculos de velocidades durante una secuencia más amplia de imágenes y promediar el resultado final.

Para ello se ha propuesto una solución de promedio de las últimas 5 velocidades calculadas, lo que supone la estimación de la velocidad durante 1 segundo. Con esta técnica evitamos un salto excesivo del medidor de velocidad en un intervalo de 1 segundo.

Para filtrar mejor posibles errores se establece que una medida de velocidad instantánea de una determinada secuencia no puede presentar una variación de más del 30% con respecto a su medida anterior. Se supone que hemos obtenido una velocidad errónea y no se incluye en el cálculo de la velocidad promedio.

6.7 Estimación del Mapa de Profundidades

6.7.1 Depth from motion: Con respecto a la posición de un punto y su centro óptico

Vamos a tratar de interpretar el Optical Flow para calcular la distancia a un punto de la imagen. Para utilizar esta técnica es preciso partir de la base de que el vehículo se está moviendo a través de un eje óptico de expansión (FOE).

Este eje óptico de expansión es el punto donde nace el movimiento. Si la cámara está alineada con la trayectoria del vehículo, el FOE debe coincidir con el centro óptico de la imagen.

Se puede demostrar geométricamente, que el vector de un punto del Optical Flow es mayor cuanto mayor sea el radio al centro óptico.

La siguiente figura muestra la geometría óptica de un punto. El punto P es estacionario en el mundo real, y el vehículo se mueve con una velocidad de dZ/dt . El punto P es proyectado en las imágenes que toma la cámara del vehículo, de forma que esta proyección p se desplaza en cada imagen [HIMANSHU 2000].

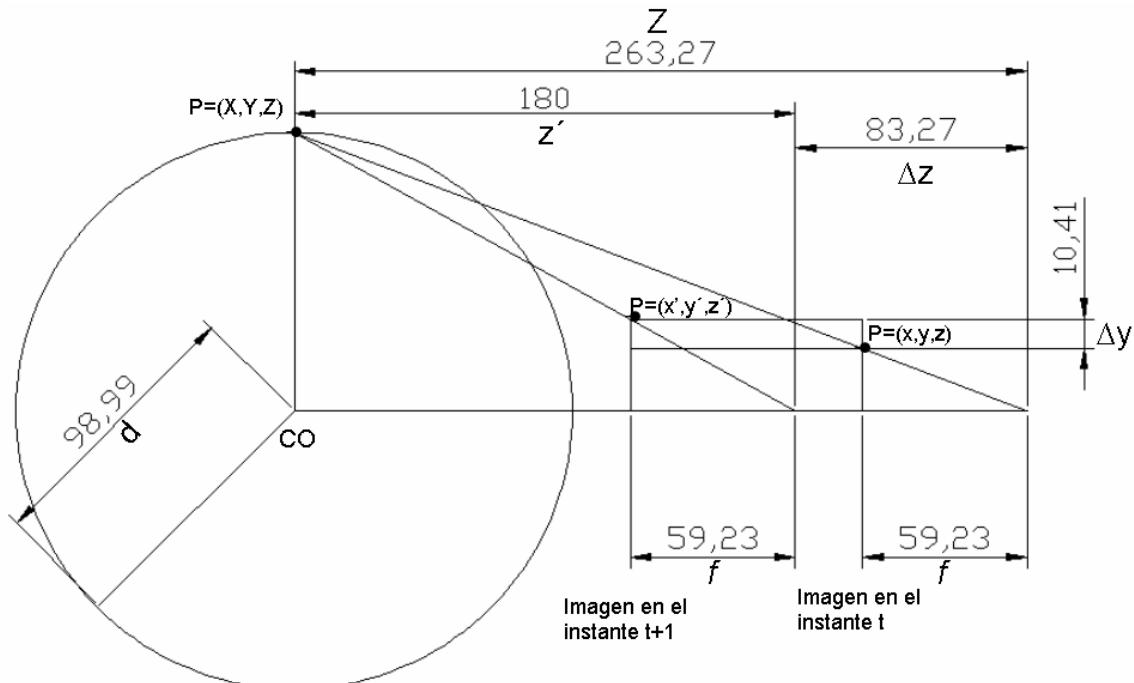


Figura IV-37 Cono de visión de un punto respecto de un eje óptico de movimiento.

Según la figura vamos a tener un Punto con unas coordenadas X, Y, Z que distan del centro Óptico (CO).

Pero en nuestra cámara disponemos de la proyección del punto x, y, z en la primera imagen, y x', y', z' para la proyección del punto en la segunda imagen (t+1).

Por triangulación tenemos:

$$1^{\text{a}} \text{ ecuación: } (x / X) = (y / Y) = (z / Z)$$

$$\text{Sustituimos... } y / z = Y / Z$$

$$yZ = Yz$$

Diferenciando con respecto al tiempo:

$$2^{\text{a}} \text{ ecuación: } yZ' + y'Z = Y'z + Yz'$$

z es la distancia focal de la cámara y permanece constante, luego su derivada es 0.

Como suponemos que el vehículo avanza a través del eje óptico, Y también es constante, luego su derivada es 0. Por lo tanto tendríamos:

$$3^{\text{a}} \text{ ecuación: } (Z / Z') = -(y / y')$$

$$\text{tiempo para contacto} = -(y / y').$$

Como sabemos la velocidad del vehículo Z', se puede determinar la profundidad a partir de la tercera ecuación.

Podemos mostrar un mapa de profundidades por cada vector del Optical Flow calculado, y para representar los diferentes tiempos para contacto con cada punto de la imagen, se utilizan colores de diferentes tonalidades.



Figura IV-38 Mapa de profundidades estimadas para un Optical Flow.

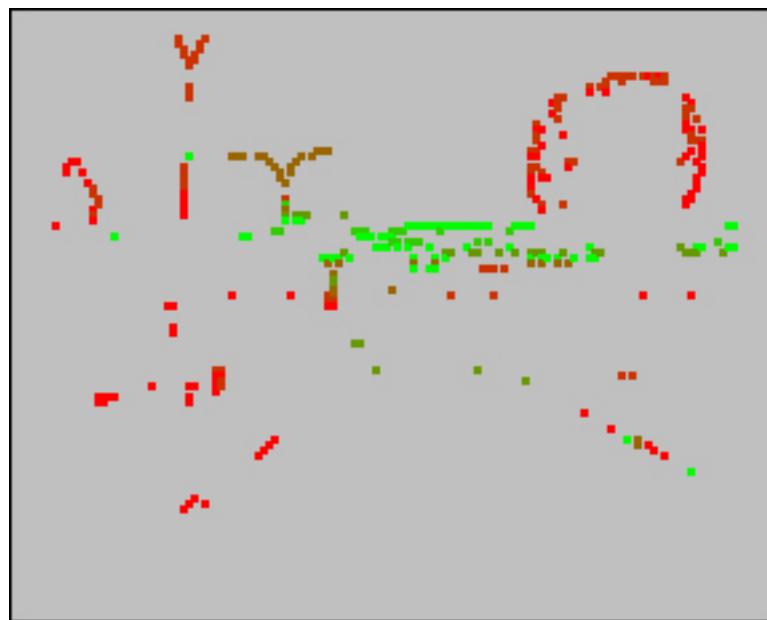


Figura IV-39 Imagen del mapa de Profundidades sin imagen.

Según la figura, las profundidades se indican según un gradiente de color que va desde el verde al rojo. Los puntos más cercanos de la imagen corresponden al color rojo, y los más lejanos al verde. El uso de este mapa de color se utiliza para dar una visión global de las profundidades de un Optical Flow. Se han elegido solo seis tonalidades diferentes entre el rojo y el verde para que a simple vista el usuario pueda comparar fácilmente las profundidades de un Optical Flow.



Zona más cercana

Zona más lejana

Figura IV-40 Escala de profundidades

En la imagen se puede distinguir distintas zonas en primer plano como por ejemplo una señal, la primera farola, un vehículo aparcado o un árbol, todos ellos con gradientes muy cercanos al rojo indicando proximidad. En un segundo plano podemos observar el horizonte con tonos mas verdes indicando lejanía.

Como veremos en el siguiente punto, el cálculo de la profundidad mediante esta técnica es muy recomendable, pero existen varios problemas que generan errores en la práctica.

6.8 Seguimiento y Etiquetado de Objetos

A continuación se detallará la técnica utilizada para el etiquetado de objetos previamente reconocidos por el sistema SACAT. El objetivo y su aplicación futura es la de proporcionar soporte de ayuda al sistema SACAT, ya que si se ha reconocido un objeto que ya ha sido analizado no nos interesa volver a analizarlo. Es por ello que nuestro sistema tratará de reconocer si un objeto es nuevo o se encuentra en la trayectoria de uno previamente reconocido y etiquetado.

Como se ha explicado la interconexión con el sistema SACAT se dejará preparada pero no se hará efectiva. Trabajamos bajo las suposiciones de la disponibilidad de los archivos que indiquen los objetos reconocidos en cada imagen.

6.8.1 Lectura de objetos

La lectura de los objetos reconocidos se realiza accediendo al archivo correspondiente a la imagen de análisis que contenga la lista de objetos. Dicho archivo está identificado con el número de la imagen a la que pertenece, siguiendo el siguiente formato:

Nombre de archivo: [numImage].dat

Cada archivo se corresponde con un objeto distinto, cada línea contiene 4 números enteros separados por un espacio, cada par de enteros representa un punto delimitado en el sistema de coordenadas. Se describen dos puntos que se corresponden con la esquina superior izquierda e inferior derecha de un rectángulo. Es dicho rectángulo el que establece la posición del objeto dentro de la imagen.

Veamos una imagen que representa este procedimiento:

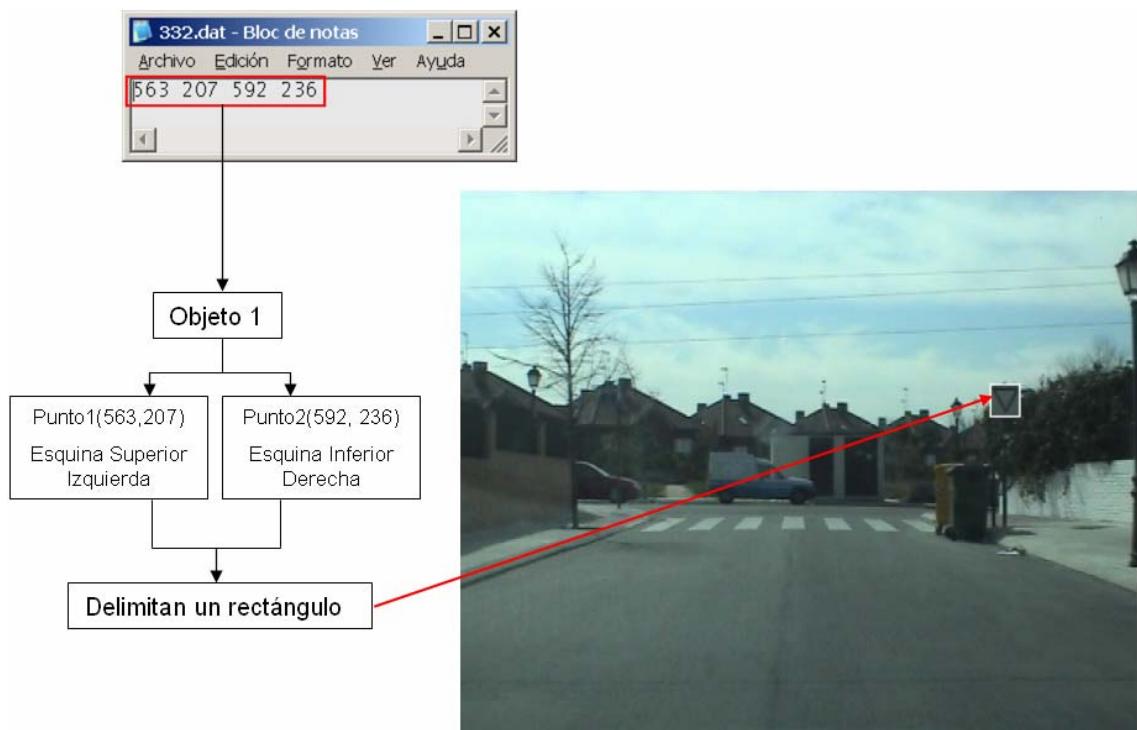


Figura IV-41 Identificación de objetos por medio de archivos de coordenadas.

6.8.2 Estimación de la posición futura

La estimación de la posición futura consiste en evaluar la posible posición que ocupará el objeto identificado en la siguiente imagen de la secuencia, con la finalidad de determinar si los objetos que se detallen en la siguiente imagen pertenecen a un mismo objeto ya identificado o son nuevos.

Para conseguir dicho objetivo se hace uso del cálculo del Optical Flow que proporciona información sobre el movimiento de la imagen, respecto a la imagen anterior. Es lógico establecer que el movimiento futuro tendrá gran similitud con el calculado anteriormente, por lo tanto se puede suponer que la posición futura del objeto seguirá la trayectoria que describía en las anteriores imágenes. Esta es la base fundamental de la técnica empleada que describiremos a continuación con mayor detalle.

El primer paso para la estimación de la posición futura de un objeto consiste en observar el entorno de vecindad que rodea a cada esquina del objeto y estimar la media de desplazamiento que posee dicho punto. Para ello se recorre la porción de vecindad de la matriz que almacena el Optical Flow para localizar los vectores próximos al punto de interés,

se suman sus desplazamientos y se realiza la media. Al punto original (esquinas) se le suma el desplazamiento calculado y se obtiene la posición futura de cada esquina. De esta manera ya tenemos un rectángulo que delimita la posición futura de nuestro objeto.

El segundo paso consiste en ampliar la zona futura delimitada en el apartado anterior para poder corregir en mayor parte algún error de cálculo o variación de la trayectoria de desplazamiento. En particular se añaden 15 píxeles tanto de ancho como de alto a la zona futura.

En caso de ausencia de vectores en la zona que delimita el objeto se establecerá como posición futura la posición inicial, aumentando en 15 píxeles su tamaño con el fin de incrementar las posibilidades de acierto.

Veamos una imagen que ilustre dicho procedimiento:

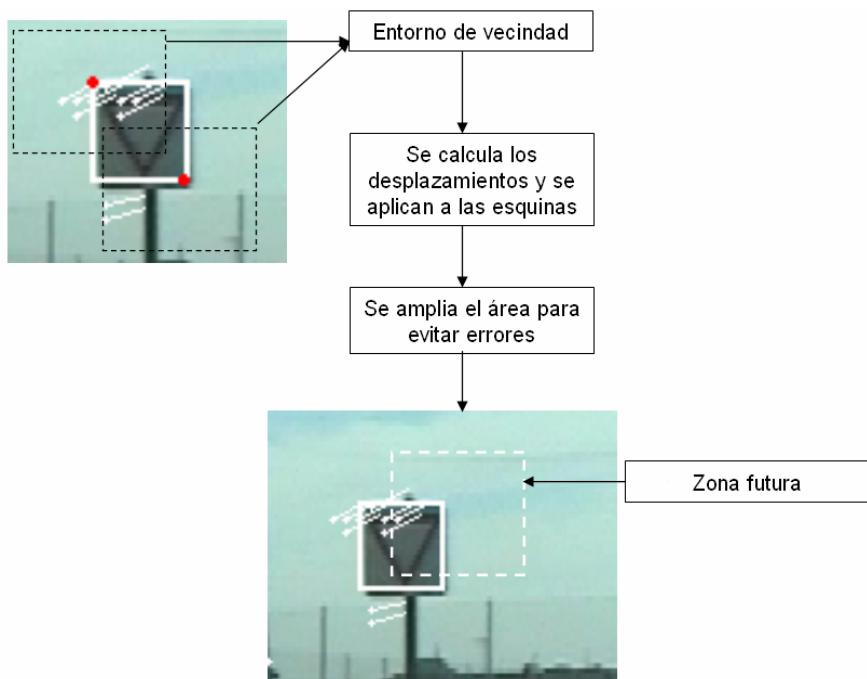


Figura IV-42 Estimación de la posición futura de un objeto.

6.8.3 Asignar etiqueta a un objeto

A cada objeto previamente reconocido se le ha de asignar una etiqueta que lo identifique únicamente con el fin de identificarlos y poder determinar si los siguientes objetos son nuevos o ya han sido etiquetados previamente, por tanto estaremos ante una situación de desplazamiento del mismo objeto por la imagen.

La técnica empleada para tal fin será verificar para cada objeto reconocido en la imagen, si su posición está contenida en la de alguna zona futura de los objetos reconocidos en la imagen anterior, si se produce alguna coincidencia se asignará la misma etiqueta que el objeto con el que coincide, de lo contrario se asignará una etiqueta totalmente nueva.

En el caso de que no exista ningún objeto reconocido en la imagen anterior se asignará una etiqueta nueva para cada uno de los objetos encontrados sin necesidad de realizar ninguna comprobación.

Veamos una imagen que ilustre dicho procedimiento:

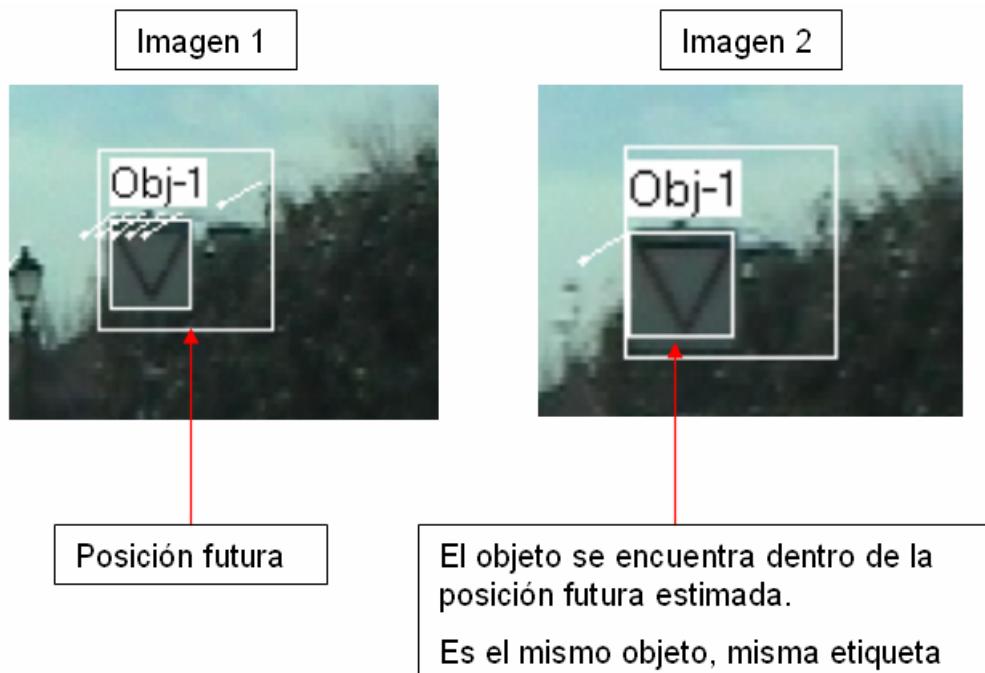


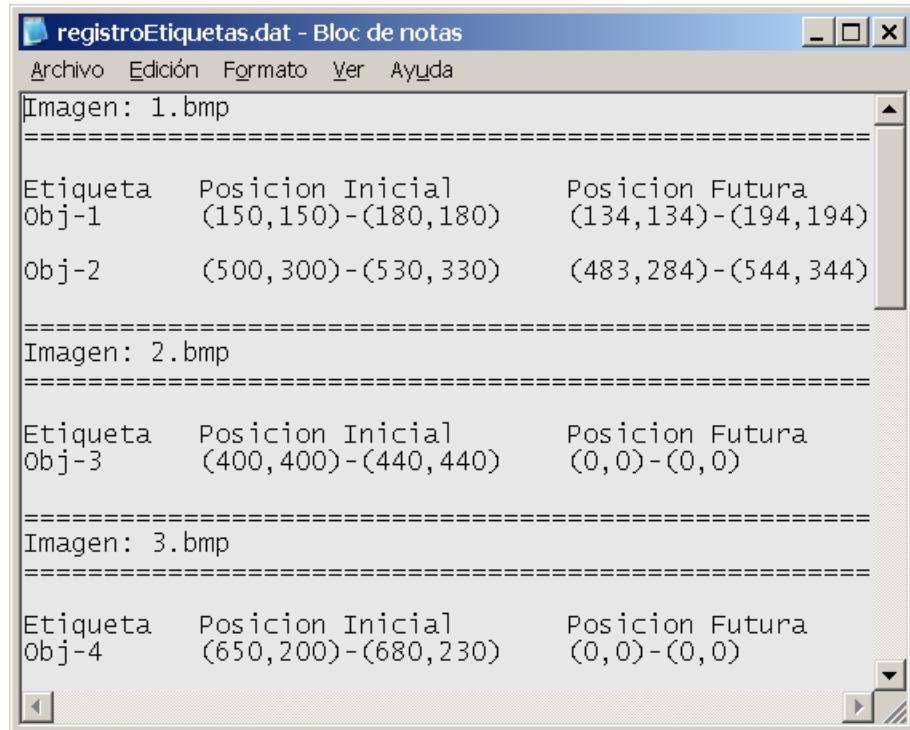
Figura IV-43 Asignación de etiqueta

6.8.4 Generar el archivo de registro de etiquetas

Con el fin de facilitar los resultados al sistema SACAT se generará un archivo durante el transcurso de la ejecución. En el transcurso del análisis, cada vez que se detecte un objeto y se asigne una etiqueta se añade una entrada al archivo especificado.

El archivo recibe el nombre de “listaEtiquetas.dat” y su formato sigue las especificaciones descritas en la ERS.

A continuación veremos un ejemplo del archivo generado al termino de un análisis.



The screenshot shows a Windows Notepad window with the title bar 'registroEtiquetas.dat - Bloc de notas'. The menu bar includes 'Archivo', 'Edición', 'Formato', 'Ver', and 'Ayuda'. The main content area contains the following text:

```
Imagen: 1.bmp
=====
Etiqueta Posicion Inicial Posicion Futura
Obj-1 (150,150)-(180,180) (134,134)-(194,194)

Obj-2 (500,300)-(530,330) (483,284)-(544,344)
=====
Imagen: 2.bmp
=====
Etiqueta Posicion Inicial Posicion Futura
Obj-3 (400,400)-(440,440) (0,0)-(0,0)
=====
Imagen: 3.bmp
=====
Etiqueta Posicion Inicial Posicion Futura
Obj-4 (650,200)-(680,230) (0,0)-(0,0)
```

Figura IV-44 Archivo de registro de etiquetas generado durante el análisis.

6.9 Análisis de Tiempo Real

El más importante de los requisitos no funcionales es la restricción de desarrollar un sistema de análisis en tiempo real. La disponibilidad de los resultados obtenidos carece de valor si se presentan desfasados en el tiempo. Es por ello que se ha perseguido cumplir al máximo dicha restricción.

La mayor carga de trabajo recae sobre el cálculo del Optical Flow, por ello se han ideado una serie de técnicas de filtrado y procesado que permitan minimizar al máximo el tiempo de cómputo de este módulo. Las técnicas empleadas han sido descritas en el análisis y diseño de módulos del sistema, y en cada una de ellas se puede apreciar los beneficios de tiempo que proporcionan al resultado final.

El sistema final analiza imágenes de dos en dos tomadas de una secuencia, y representa los resultados al término de los cálculos realizados. Existe entonces la posibilidad de que el tiempo de cálculo exceda al tiempo que ha transcurrido entre dos imágenes, produciéndose así un desfase que alteraría la ejecución del sistema. La solución propuesta realiza un control exhaustivo de tiempos, en el que se lleva la cuenta del tiempo transcurrido desde el comienzo del análisis, y lo compara con el tiempo real. Si el resultado de dicha comparación resulta desfavorable se procede a descartar la siguiente imagen de la secuencia, ganando así el tiempo de retraso acumulado.

El proceso de descartar la imagen siguiente se realiza siempre y cuando el sistema no se encuentre en fase de seguimiento y etiquetado de objetos, de lo contrario se produciría un error al saltarse la siguiente imagen, para la cual se han realizado estimaciones de posiciones futuras.

Descartar una imagen de la secuencia permite cumplir la restricción de tiempo sin alterar la simulación, ni producir un descenso del rendimiento de los resultados obtenidos.

6.10 Análisis de Estructuras de Datos

6.10.1 Matriz Optical Flow

El sistema debe almacenar en una estructura, un Optical Flow por cada par de imágenes para su posterior análisis y visualización. Esta estructura tiene un gran tamaño y debe contener el mínimo número de datos para que su acceso y recorrido sea eficiente.

El Optical Flow esta formado por vectores fácilmente representables por dos pares de coordenadas que representan la posición en la primera y segunda imagen respectivamente. La forma directa de almacenar dichos puntos, seria un array bidimensional estático con la misma dimensión que las imágenes de entrada. La posición de un vector en la matriz correspondería con sus coordenadas en la primera imagen y el contenido de dicha posición seria las coordenadas de la segunda imagen. El problema que surge es que la matriz almacenaría demasiados registros que nunca van a ser usados y resultaría del todo ineficiente. La solución es dividir el tamaño de la matriz tanto en X como en Y por la distancia entre puntos analizados:

Dimensión en X de la Matriz = dimensión X de la imagen / distancia entre puntos
Dimensión en Y de la Matriz = dimensión Y de la imagen / distancia entre puntos

Hay que tener en cuenta que para escribir en la matriz, hay que dividir entre la distancia entre puntos y una vez que se ha leído un dato hay que multiplicarlo por la distancia entre puntos.

Por ejemplo, si disponemos de un vector con origen en (100,50) y destino en (150, 120), y una distancia entre puntos de 5, este quedará almacenado en registro de la matriz (20,10) y su contenido sería (150,120).

6.10.2 Lista de Etiquetas

Debemos almacenar las posiciones iniciales y estimaciones futuras de todos los objetos reconocidos, así como la etiqueta asignada. La posición inicial de cada objeto se represente mediante dos pares de coordenadas, las dos primeras coordenadas sirven para determinar la esquina superior izquierda, y las dos últimas la esquina inferior derecha. De forma análoga se almacena la posición futura en otros dos pares de coordenadas. Por ultimo se utiliza una cadena de caracteres para almacenar la etiqueta.

Se establece un array estático de quince registros que conforman la lista de etiquetas.

6.11 Alcance y limitaciones generales

En este apartado se introducirán el alcance y limitaciones generales que poseen las técnicas empleadas para la consecución de los objetivos.

6.11.1 Optical Flow

Debido a problemas de diverso índole el Optical Flow posee errores que influirán al resto de cálculos.

El principal inconveniente del flujo óptico se conoce con el nombre de problema de la apertura [GONZALEZ 1999]. Surge de la imposibilidad de asignar un valor único al desplazamiento de un punto debido a que el patrón que se va a analizar es demasiado pequeño y posee muchas coincidencias.

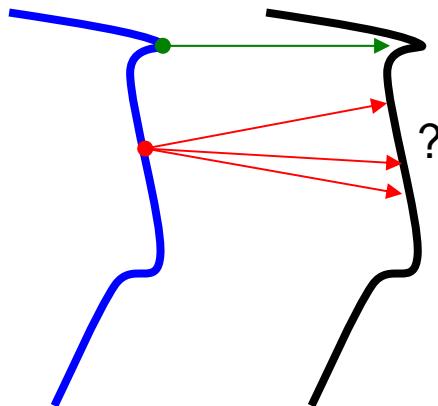


Figura IV-45 Problema de la apertura

Como se aprecia en la imagen, el primer punto analizado tiene una única correspondencia debido a su trazado unívoco, sin embargo para el siguiente punto es fácil encontrar muchos puntos con el mismo trazado.

Otra limitación que condiciona el Optical Flow es la aceptación como válidos de resultados del PatternMatching que sobrepasen el nivel de aceptación, pero incurren en un error de correspondencia.

6.11.2 Estimación de la velocidad de movimiento

La técnica que se ha explicado presenta diversas limitaciones sujetas a varios factores de influencia, todos los inconvenientes que describiremos a continuación repercuten en el resultado del Optical Flow y por consiguiente en la estimación de la velocidad.

El primer inconveniente es la ausencia de textura en la zona de interés, por estar compuesta en un 90% de superficie plana, carente de cambios de tonalidades y relieves. Estas características presentan en primer lugar un gran problema a la hora de seleccionar puntos candidatos para el análisis de correspondencias en la siguiente imagen, y en segundo lugar muchos de los puntos seleccionados presentan correspondencias que no superan el porcentaje de confianza (establecido a priori en un 60%).

Un segundo inconveniente se presenta por las posibles irregularidades del terreno (cambios de rasante, baches, imperfecciones, etc.) que afectan considerablemente al Optical Flow e introducen vectores que se corresponden con la realidad, pero sin embargo no válidos para la técnica explicada, por invalidar la suposición de la superficie plana de desplazamiento.

Un tercer inconveniente se presenta con los puntos que se desplazan en exceso (velocidad elevada) y que no quedan reflejados en el Optical Flow por salirse de la ventana de búsqueda establecida.

Un cuarto inconveniente lo presenta la naturaleza de la cámara utilizada, se trata de una cámara de video doméstica que no procesa *scan progresivo* y por lo tanto captura imágenes entrelazadas que disminuye la calidad de la imagen, afectando a la apreciación de movimiento a partir de 50 Km/h. Por lo tanto las imágenes que se capturen a una velocidad superior a 50 Km/h presentan un descenso de nitidez notable que afecta directamente sobre el cálculo del Optical Flow.

Debido a los inconvenientes expuestos se condiciona en gran medida la estimación de la velocidad de movimiento, llegando a perder eficiencia o presentar ausencia de resultados en velocidades superiores a 45 Km/h.

6.11.3 Estimación del mapa de profundidades

La naturaleza de esta técnica establece que nos encontramos ante un movimiento continuo, con ausencia de aceleración y siguiendo una misma trayectoria; basándonos en esta suposición podemos concluir que se obtendrán los mejores resultados en los instantes en los que se cumplan dichas características. Sin embargo el movimiento que describe un vehículo no está ausente de aceleraciones ni de cambios de trayectoria, por lo que nos encontraremos con análisis deficientes en determinados intervalos del análisis. Todas estas variaciones repercuten en el foco óptico de expansión (FOE), que es una de las bases de la técnica.

El problema surge puesto que el vehículo no sigue una trayectoria recta, por tanto, el FOE varía su posición para cada secuencia de imágenes influyendo negativamente en las estimaciones de profundidad.

Si bien las características deseadas de aceleración y trayectoria no son constantes, es importante resaltar que promediando los intervalos de tiempo en los que no se ajustan a las suposiciones, con la duración total del análisis, nos encontramos con un porcentaje elevado de tiempo en el que se producen resultados correctos.

6.11.4 Seguimiento y etiquetado de objetos

Los resultados proporcionados dependen directamente del resultado del cálculo del Optical Flow, y no suponen una carga de tiempo excesiva. Por lo tanto sus limitaciones se establecen según la fiabilidad del flujo óptico. Con el fin de ofrecer soluciones a las carencias

del Optical Flow, la técnica de seguimiento y etiquetado propone metodologías que consiguen minimizar los errores y mejoran la fiabilidad de este tercer objetivo.

6.12 Técnicas Alternativas

Se introducirán a continuación las técnicas que han sido estudiadas, desarrolladas y posteriormente descartadas, debido a un rendimiento deficiente para la aceptación del sistema final, o simplemente han sido superadas por las utilizadas.

6.12.1 Filtrado Prewitt

El filtro Prewitt [GONZALEZ 1999] se utiliza para la detección de bordes de una imagen, se basa en la convolución de dos máscaras:

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
-0	0	0
1	1	1

Detector de bordes verticales. Detector de bordes horizontales.

Este filtrado se ha descartado por no ofrecer un suavizado del resultado obtenido, característica que si posee el filtro SOBEL.

6.12.2 Umbralizado + Adelgazamiento

Técnica destinada al filtrado y selección de puntos de análisis [VAZQUEZ 1996]. Consiste en un tratamiento digital de la imagen que combina tres filtros independientes, primero un filtrado de bordes, a continuación se realiza un umbralizado binario, y por último se aplica un adelgazamiento, que proporciona como resultado una imagen binaria donde los bordes tienen dimensión de un píxel. Dicha técnica elimina de la imagen todo píxel que no se corresponda con un borde de un píxel de anchura, lo que supone más del 90 % de la imagen. Veamos como el resultado sobre una imagen en concreto:



Figura IV-46 Imagen umbralizada y esqueletizada.

Esta técnica ha sido descartada por consumir demasiado tiempo de cálculo, aunque proporciona resultados muy buenos, no supera en rendimiento a la técnica utilizada en la solución propuesta.

6.12.3 Incremento de alturas

Técnica destinada al cálculo del mapa de profundidades, implementada y posteriormente descartada.

La estimación de las distancias hasta objetos o puntos relevantes de la imagen se determina gracias al siguiente sistema de ecuaciones:

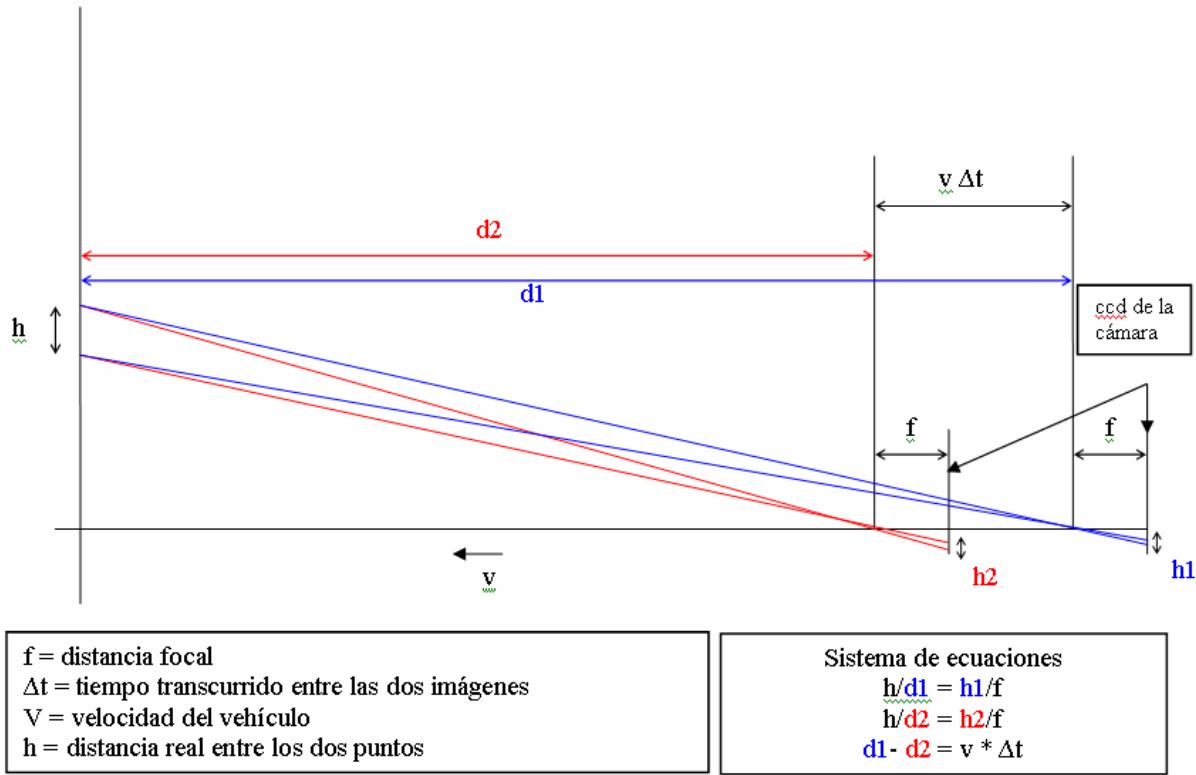


Figura IV-47 Técnica del incremento de alturas.

Como se aprecia en la figura se necesitan como datos de entrada además de la velocidad a las que se mueve el vehículo, las alturas de dos puntos significativos de la imagen.

Para seleccionar dichas alturas se realiza un filtrado al Optical-Flow del instante t con el del instante $t+1$. El objetivo de dicho filtrado es conservar una misma dirección de desplazamiento en las dos imágenes, ya que de lo contrario no sería un objeto válido para el análisis.



Optical Flow 1: Imagen 1 con imagen 2.

Optical Flow 2: Imagen 2 con imagen 3.

Filtro: Correspondencia de vectores entre los dos Optical Flow.

Figura IV-48 Filtrado de vectores lineales con dos Optical Flow consecutivos.

Los vectores resultantes poseen menos probabilidad de error y al reducir el número de vectores, determinamos más rápidamente el par de vectores candidatos para calcular su altura.



Figura IV-49 Calculo de distancia a una altura determinada.

El principal problema de esta técnica es el reconocimiento de objetos o zonas de la imagen. Es necesario disponer de la altura de un objeto o zona en dos imágenes para poder calcular el incremento de altura de dicho objeto.

Es muy probable que los vectores determinados no se correspondan al un mismo objeto, aun valiéndose del filtro de vectores lineales. Los experimentos realizados mediante esta técnica incurrián en un gran número de errores, es por ello por lo que fue descartada de cara al sistema final.

6.12.4 Calculo del Foco Óptico de Expansión

Para calcular el mapa de profundidades es necesario conocer la posición del Foco Óptico de Expansión (FOE) que se define como la dirección de la componente de translación del movimiento de aproximación, es decir, el punto donde emergen los vectores de un Optical Flow.

En nuestro sistema este foco se establece por defecto como el centro óptico de la cámara porque se considera que este centro esta alineado con el eje de movimiento del

vehículo. Pero esto no siempre sucede así, puesto que la trayectoria que describe el vehículo puede no ser recta.

Una técnica estudiada para resolver este problema es el uso de un FOE variable calculado a partir del Optical Flow. Cada vector que lo conforma, tiene una dirección dependiendo de la posición que ocupa en la imagen. Si calculamos la intersección entre la proyección de dos vectores, obtenemos su FOE.

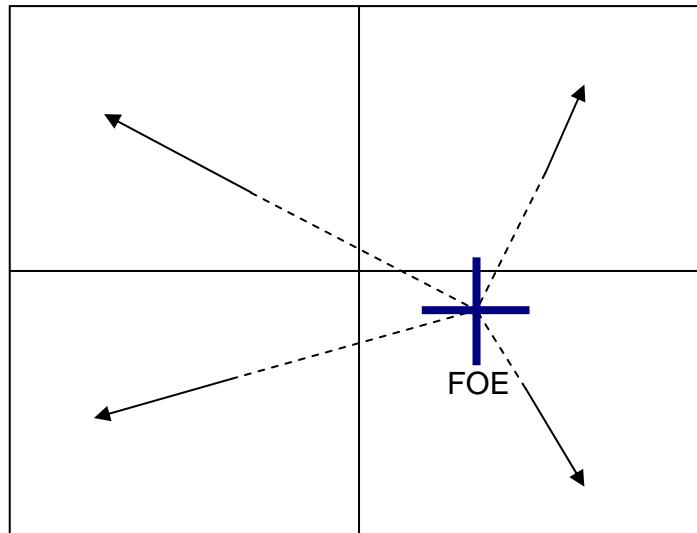


Figura IV-50 Calculo del foco óptico de expansión

Como se aprecia en la figura, todos los vectores confluyen en un mismo FOE, pero esto solo es posible teóricamente, ya que en la práctica el Optical Flow generado siempre posee errores como ya se explicó anteriormente.

Una solución sería hacer la intersección entre el promedio del vectores de la mitad izquierda de la imagen con los de la mitad derecha. El resultado del FOE variable mediante esta técnica es algo más preciso que un FOE estático, sin embargo, cuando el movimiento que se produce posee además de una componente traslacional, una rotacional, ambas aparecen combinadas en el Optical Flow y sería necesario descomponerlas para determinar con mayor exactitud su FOE [GONZALEZ 1999].

6.12.5 Variación del umbral

Técnica desarrollada para cumplir la restricción de tiempo, consiste en aumentar o disminuir el umbral del Optical Flow, con el fin de ajustar el número de puntos de análisis de cada imagen conforme a las restricciones de tiempo.

La teoría de dicha técnica consiste en realizar un control de tiempos del ultimo análisis, y compararlo con el máximo permitido (el intervalo transcurrido entre la captura de las dos imágenes), dependiendo del resultado de la comparación se aumentaba o disminuía el umbral.

Finalmente se observaba como el numerosas situaciones el umbral se disminuía drásticamente desembocando en un filtrado excesivo, ausencia de puntos de análisis. Perjudicando de ese modo al cálculo del Optical Flow y en consecuencia a la consecución de los tres objetivos concretos. Por dicho motivo se ha optado por la técnica de descartar imágenes.

6.13 Tabla de Tareas, Técnicas y Estructuras de Datos

Tarea	Técnicas Empleadas	Técnicas descartadas	Estructura de datos
Optical Flow	Filtrado Sobel, Rejilla uniforme, Umbralizado, PatternMatching	Filtrado Prewitt, Umbralizado y Adelgazamiento.	Matriz Optical Flow
Velocidad de movimiento	Movimiento del terreno		Matriz Optical Flow
Mapa de profundidades	Depth from motion	Incremento de alturas, Calculo del FOE	Matriz Optical Flow
Seguimiento y etiquetado	Estimación de la posición futura		Matriz Optical Flow, Lista de etiquetas
Restricción de tiempo real	Descartado de imágenes	Variación del umbral	Lista de etiquetas

Tabla IV-9 Combinación de tareas, técnicas y estructura de datos.

7 Planificación y Presupuesto

La metodología de desarrollo empleada a lo largo de todo el proyecto sigue el modelo RUP (Rational Unified Process), que contempla la existencia de cuatro fases: comienzo, elaboración, construcción y transición. En cada fase se realiza un proceso iterativo que genera como resultado un prototipo del sistema final.

7.1.1 Distribución de tareas

Tarea	Comienzo	Fin
APRENDIZAJE	lun 18/10/04	vie 11/02/05
Comienzo del proyecto (KICK OFF)	lun 18/10/04	lun 18/10/04
Aprendizaje C++	lun 18/10/04	jue 24/01/05
Aprendizaje Visión por computador	lun 18/10/04	vie 11/02/05
Aprendizaje de manejo de librerías para tratamiento de imágenes (MIL)	lun 08/11/04	jue 25/11/04
COMIENZO	jue 21/10/04	lun 22/11/04
Requisitos	jue 21/10/04	lun 25/10/04
Planificación de proyecto	jue 21/10/04	jue 21/10/04
Entrevista con el Cliente (Mario Mata)	lun 25/10/04	lun 25/10/04
Análisis	lun 25/10/04	vie 05/11/04
Desarrollo de propuesta de proyecto	lun 25/10/04	vie 05/11/04
Estudio del estado del arte	lun 25/10/04	vie 29/10/04
Diseño	vie 05/11/04	lun 22/11/04
Entrega de propuesta de proyecto	vie 05/11/04	vie 05/11/04
Presentación de propuesta de proyecto "Power Point"	lun 22/11/04	lun 22/11/04
ELABORACIÓN	lun 22/11/04	vie 11/02/05
Requisitos	lun 22/11/04	mar 30/11/04
Planificación de proyecto	lun 22/11/04	lun 22/11/04
Entrevista con el Cliente (Mario Mata)	lun 22/11/04	lun 22/11/04
Generar un conjunto de	lun 22/11/04	lun 29/11/04

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

imágenes de prueba		
Recogida de Informacion	lun 22/11/04	mar 30/11/04
Análisis	mié 01/12/04	lun 10/01/05
Especificación de requisitos de software ERS	mié 01/12/04	mar 14/12/04
Casos de Uso	mié 15/12/04	lun 20/12/04
Modelo conceptual de clases	mar 21/12/04	lun 10/01/05
Diseño	mar 11/01/05	mar 18/01/05
Diseño de clases	mar 11/01/05	vie 14/01/05
Diseño de pruebas	lun 17/01/05	mar 18/01/05
Implementación	mié 19/01/05	mié 09/02/05
"Implementacion del ""Optical Flow"""	mié 19/01/05	jue 27/01/05
Implementacion prototipo velocidad	vie 28/01/05	mar 01/02/05
Implementacion calculo de distancias	mié 02/02/05	vie 04/02/05
Implementacion prototipo espacio libre	lun 07/02/05	mié 09/02/05
Pruebas	jue 10/02/05	vie 11/02/05
Verificación de pruebas	jue 10/02/05	vie 11/02/05
Entrega de prototipo de velocidad	vie 11/02/05	vie 11/02/05
Entrega de prototipo de espacio libre	vie 11/02/05	vie 11/02/05
CONSTRUCCIÓN	vie 11/02/05	vie 08/04/05
Requisitos	vie 11/02/05	mar 08/03/05
Entrevista con el Cliente (Mario Mata)	vie 11/02/05	vie 11/02/05
Planificacion de proyecto	lun 28/02/05	lun 28/02/05
Recogida de Informacion	lun 28/02/05	mar 08/03/05
Análisis	mié 09/03/05	jue 17/03/05
Revision de Especificacion Requisitos	mié 09/03/05	lun 14/03/05
Revisión de Casos de Uso	mar 15/03/05	jue 17/03/05
Diseño	vie 18/03/05	mié 23/03/05
Revisión de Diseño de clases	vie 18/03/05	lun 21/03/05
Revisión de Diseño de pruebas	mar 22/03/05	mié 23/03/05
Implementación	jue 24/03/05	mié 06/04/05
Implementación del	jue 24/03/05	mié 06/04/05

Prototipo de Etiquetado		
Pruebas	jue 07/04/05	vie 08/04/05
Verificación de Pruebas	jue 07/04/05	vie 08/04/05
Entrega de Prototipo de Etiquetado	vie 08/04/05	vie 08/04/05
TRANSICIÓN	lun 11/04/05	vie 27/05/05
Requisitos	lun 11/04/05	mar 12/04/05
Planificación de proyecto	lun 11/04/05	mar 12/04/05
Implementación	mié 13/04/05	mar 17/05/05
Mejora de los prototipos	mié 13/04/05	mar 26/04/05
Implementación del Interface de Usuario	mié 27/04/05	mar 03/05/05
Implementación de la Aplicación	mié 04/05/05	mar 17/05/05
Pruebas	mié 18/05/05	vie 27/05/05
Estimación de los porcentajes de error de los resultados obtenidos	mié 18/05/05	lun 23/05/05
Verificación de Pruebas	mar 24/05/05	mié 25/05/05
Impresión y maquetación	jue 26/05/05	vie 27/05/05
Entrega de la Aplicación	vie 27/05/05	vie 27/05/05
Entrega de Documentación	vie 27/05/05	vie 27/05/05

Tabla IV-10 Duración de fases y tareas del proyecto.

A continuación se realiza un estudio del presupuesto estimado, desglosándolo en recursos consumidos y por fases.

7.1.2 Presupuesto por Recursos

Nombre del Recurso	Costo	Trabajo Estandar	Tasa
Jefe de proyecto	11.520,00 €	144 horas	80,00 €/hora
Analista	28.320,00 €	472 horas	60,00 €/hora
Programador	12.240,00 €	408 horas	30,00 €/hora
Testeador	1.920,00 €	96 horas	20,00 €/hora
Ordenador	3.000,00 €	1	3.000,00 €
Camara de fotos	1.000,00 €	1	1.000,00 €
Internet	467,20 €	1.168 horas	0,40 €/hora
Electricidad	163,20 €	1.088 horas	0,15 €/hora

Licencia de Borland C++	600,00 €	1	600,00 €
Licencia de Microsoft Office	100,00 €	1	100,00 €
Licencia Windows XP	150,00 €	1	150,00 €

Tabla IV-11 Presupuesto de recursos.

7.1.3 Presupuesto por Fases

Fase	Coste
APRENDIZAJE	4.850,00 €
COMIENZO	6.208,80 €
Requisitos	1.280,00 €
Análisis	4.832,00 €
Diseño	0,00 €
ELABORACIÓN	23.735,60 €
Requisitos	7.840,00 €
Análisis	8.640,00 €
Diseño	2.880,00 €
Implementación	3.840,00 €
Pruebas	320,00 €
CONSTRUCCIÓN	15.972,00 €
Requisitos	7.840,00 €
Análisis	3.360,00 €
Diseño	1.920,00 €
Implementación	2.400,00 €
Pruebas	320,00 €
TRANSICIÓN	8.714,00 €
Requisitos	1.280,00 €
Implementación	6.000,00 €
Pruebas	1.280,00 €

Tabla IV-12 Presupuesto por fases del proyecto.

7.1.4 Resumen

Como resumen del presupuesto se puede elaborar la siguiente tabla:

Comienzo	Fin	Duración	Trabajo	Coste Total
18/10/04	27/05/05	139 días	3376 horas	59.480,60 €

Tabla IV-13 Presupuesto global del proyecto.

Capítulo V. Plan de Verificación

1 Plan de Pruebas

Se ha elaborado un plan de verificación, compuesto por un conjunto de pruebas que pretenden verificar de manera formal los resultados del sistema, y poder determinar así, el grado de veracidad en la resolución de los objetivos concretos, así como de las técnicas desarrolladas.

El plan elaborado se divide en tipos bien diferenciados de pruebas. Se consideran como pruebas de bajo nivel las realizadas a módulos o funciones específicas del sistema, y reciben el nombre de pruebas de unidad; el siguiente nivel engloba las denominadas pruebas de integración que persiguen demostrar el funcionamiento y colaboración de los diferentes módulos; por último se analiza el sistema como un conjunto indivisible y se verifican los resultados proporcionados, es lo que se conoce como pruebas de sistema.

Pruebas de unidad: Se trata de verificar una serie de módulos clave de la aplicación. El objetivo es comprobar que las salidas de dichos módulos sean las correctas en función de las entradas. Se utilizará en muchos casos código auxiliar para probar el método y se introducirán datos de entrada que sean posibles y representativos.

Pruebas de integración: Se pretende probar que los diferentes módulos de los que se componen el sistema interactúan correctamente entre ellos obteniendo resultados coherentes en función de la entrada.

Pruebas del sistema: Están destinadas a determinar el grado de veracidad de los cálculos resultantes. Para ello es necesario que las imágenes de entrada sean escenas lo más representativas posibles de las combinaciones de los valores siguientes:

Movimiento: Curvas, rectas, cuestas arriba, cuestas abajo, etc.

Velocidad: 50, 40, 30, 20, 10 Km./h.

Visibilidad: Escasa visibilidad, buena visibilidad (Condiciones medioambientales adversas, contra luz, etc.).

Formas: Objetos de la naturaleza (árboles, montañas, nubes, etc.) y estructuras (túneles, puentes, etc.).

El resultado de las pruebas de sistema servirá para la elaboración de un informe de los resultados obtenidos en comparación con los esperados, estimando porcentajes de acierto en función de la cuantía de pruebas realizadas.

2 Ítems Probados

Se prueba el software del sistema AMIVSI con una secuencia de imágenes y un fichero de configuración predeterminado. El archivo que lanza el sistema se denomina *AMIVSI.exe*.

3 Características Probadas

1. Hay Vector

Corresponde al método Boolean hayVector(x,y) de la Clase Matriz. Se va a probar que se determina si hay un vector o no en una posición de la matriz.

2. Crear Matriz

Corresponde al método TMatriz(x, y, dPuntos) de la Clase Matriz. Se va a probar que se ha creado con éxito una matriz del tamaño específico.

3. Buscar Patrón

Corresponde al método BuscarPatron(ImagenID1, ImagenID2, x1,y1,x2,y2) de la Clase OpticalFlow. Se busca el patrón de la primera imagen en la segunda imagen con los límites de la ventana.

4. Calcular Optical Flow

Corresponde al método calcularOpticalFlow(matrizOptical,imgPrewit,imgPrewit2) de la Clase OpticalFlow. Se comprueba si se determinó un Optical Flow valido.

5. Calcular Velocidad

Corresponde al método CalcularVelocidad(matrizOptical,tiempo) de la Clase Navegacion. Se estima la velocidad de un Optical Flow.

6. Calcular Tamaño Ventana

Corresponde al método calcularVentanaBusqueda(x,y) de la Clase Optical Flow. Se determina el tamaño de la ventana de búsqueda en función de la posición del punto a tratar.

7. Calcular Posición Ventana

Corresponde al método CalcularPosicionBusqueda (posX, posY, origenVentanaX, origenVentanaY) de la Clase Optical Flow. Se determina la posición de la ventana de búsqueda en función de la posición del punto a tratar.

8. Calcular Profundidad

Corresponde al método PintarDeph(imagen,matriz) de la Clase ToolPaint. Se calcula y pinta la profundidad de un Optical Flow.

9. Pintar Optical Flow

Corresponde al método PintarOpticalFlow(imagen, matriz) de la Clase ToolPaint. Se va a probar que en la imagen se han pintado correctamente los vectores contenidos en la matriz.

10. Pintar Vector

Corresponde al método PintarVector(imagen, x1,y1,x2,y2) de la Clase ToolPaint. Se va a probar que un vector se pinta correctamente en la imagen.

11. Pintar Velocidad Analógica

Corresponde al método PintarVelocidadAnaloga(velocimetroDigital,vel) de la Clase ToolPaint. Se va a probar que la velocidad se muestra correctamente en el velocímetro.

12. Calcular Posición Futura

Corresponde al método Booleano estimarZonaFutura (matrizOptical) de la Clase Etiquetado. Se determina la posición de un punto en la siguiente imagen según su Optical Flow.

13. Asignar Etiqueta

Corresponde al método asignarEtiqueta(listaEtiquetas) de la Clase Etiquetado. Se asigna una etiqueta con un identificador generado.

14. Resolución de las imágenes

Se prueban diferentes resoluciones de imágenes de entrada.

15. Fichero de configuración

Se cambian los parámetros del fichero de configuración que proporcionarán diferentes salidas.

16. Combinación de escena de carretera.

Las imágenes de entrada representan una combinación de una situación concreta para una escena de donde se analizará en conjunto los objetivos propuestos.

17. Escena de carretera velocidad

Las imágenes de entrada representan una combinación de una situación concreta para una escena de donde se analizará en detalle la estimación de velocidad.

18. Escena de carretera profundidad

Las imágenes de entrada representan una combinación de una situación concreta para una escena de donde se analizará en detalle la estimación del mapa de profundidades.

19. Escena de carretera etiquetado

Las imágenes de entrada representan una combinación de una situación concreta para una escena de donde se analizará en detalle el etiquetado de objetos.

4 Características No Probadas

No se probará el análisis sobre imágenes con otros formatos.

No se probaran métodos que han sido reutilizados o de librerías externas al proyecto.

No se realizarán pruebas de hardware como adquisición de imágenes desde el interior del vehículo.

5 Especificación de procedimientos de prueba

5.1 Procedimiento Unitario: Hay_Vector

PROPOSITO

Se pretende probar que se determina correctamente si hay un vector o no en una posición de matriz. El método responde mediante una variable booleana.

REQUISITOS ESPECIALES

No aplicable.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se llama al método <i>hayVector(int x, int y)</i> de la clase TMatriz, que toma como parámetros de entrada las coordenadas x e y del punto en donde se desea comprobar si hay vector.	OK	
El método devuelve un booleano	Devuelve <i>true</i> , el método encontró un vector en la posición indicada.	
Se llama al método <i>hayVector(int x, int y)</i> de la clase TMatriz, que toma como parámetros de entrada las coordenadas x e y del punto en donde se desea comprobar si hay vector.	OK	
El método devuelve un booleano	Devuelve <i>false</i> , el método no encontró vectores	

Tabla V-1 Procedimiento Unitario: Hay_Vector.

Dicho procedimiento se realiza dos veces, una sobre una posición en la que se conoce que existe algún vector, y otra en la que se conoce que no existe ningún vector. El procedimiento indicado es crítico por ser un método que será llamado en cada acceso a la matriz Optical Flow, y agiliza el recorrido de la misma.

5.2 Procedimiento Unitario: Crear_Matriz

PROPÓSITO

Se pretende probar que se crea correctamente una estructura matriz, que almacenará el Optical Flow. La matriz se crea con un tamaño específico y un parámetro de distancia entre puntos, que indica la separación existente entre los de análisis. Además se quiere verificar que las posiciones de la matriz pueden ser consultadas y modificadas desde fuera.

REQUISITOS ESPECIALES

No aplicable.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se invoca el constructor de la clase TMatriz, pasándole como parámetros las dimensiones de la imagen y la distancia entre puntos de análisis.	OK	
Se ejecuta el método <i>setPunto(int posX, int posY,int valorX,int valorY)</i> de la clase TMatriz, indicando como parámetros de entrada dos coordenadas correctas, y sus correspondientes valores.	OK	
Se ejecuta el método <i>hayVector(int x, int y)</i> de la clase TMatriz, que toma como parámetros de entrada las coordenadas x e y del punto donde se ha introducido previamente un punto.	Devuelve <i>true</i> , el método encontró un vector en la posición indicada.	
Se ejecuta el método <i>getY(int posX, int posY)</i> de la clase TMatriz, indicando como parámetros de entrada dos coordenadas correctas.	La matriz devuelve la coordenada y del punto consultado.	

Tabla V-2 Procedimiento Unitario: Crear_Matriz.

El procedimiento verifica que se ha creado correctamente la matriz, haciendo uso del constructor, y posteriormente se introduce un dato para consultarla inmediatamente y comprobar que es el mismo.

5.3 Procedimiento Unitario: Buscar_Patron

PROPÓSITO

Se pretende buscar un patrón generado para una imagen sobre otra. Este procedimiento verifica el correcto funcionamiento del método que implemente el PatternMatching proporcionado por las librerías *MIL*.

REQUISITOS ESPECIALES

No aplicable.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se llama al método <i>BuscarPatron(MIL_ID ImagenID1,MIL_ID ImagenID2, int posX, int posY, int *x2, int *y2)</i> de la clase OpticalFlow, pasándole como parámetros la imagen origen y destino y las coordenadas del punto sobre el que se creará el patrón.	OK	
El método valiéndose del tamaño de patrón especificado, y de las librerías <i>MIL</i> devuelve una posición.	La posición esta dentro del tamaño de la ventana de búsqueda y supera el porcentaje mínimo de acierto especificado.	

Tabla V-3 Procedimiento Unitario: Buscar_Patron.

Es importante señalar como internamente se hace uso de las variables globales que especifican el tamaño del patrón, así como de la ventana de búsqueda y el porcentaje de acierto mínimo requerido. Esas variables determinan la solución, pudiéndose producir en numerosos casos ausencia de resultado.

5.4 Procedimiento Unitario: Calcular_Optical_Flow

PROPÓSITO

Se pretende probar que se calcula un Optical Flow coherente con las imágenes de entrada, y que se almacena correctamente en la estructura de datos destinada a tal efecto. Además el método debe proporcionar las medidas del vector más representativo para el cálculo futuro de la velocidad y el valor de dirección de giro.

REQUISITOS ESPECIALES

No aplicable.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se llama al método <i>calcularOpticalFlow(TMatriz *matrizOptical, TImagen *img1, TImagen *img2)</i> , pasándole por parámetros las dos imágenes de la secuencia, y la matriz donde se almacenan los resultados.	OK	
El método devuelve la matriz Optical con todos sus valores actualizados.	El Optical Flow es coherente con las imágenes de entrada. Los vectores coinciden en modulo y argumento con la correlación de puntos en las imágenes.	
El método modifica las variables globales que corresponden al vector más representativo para el cálculo de la velocidad.	Las coordenadas del vector de velocidad se corresponden con el vector de desplazamiento de mayor longitud en y, dentro de la zona de análisis de la velocidad.	
El método modifica la variable dirección de giro.	La dirección de giro se corresponde a la suma de los desplazamientos en x dividido por el número de vectores.	

Tabla V-4 Procedimiento Unitario: Calcular_Optical_Flow.

La comprobación de dicho procedimiento se realiza por parte de los desarrolladores, que deben representar la matriz Optical generada, utilizando el método *pintarOpticalFlow()*, para determinar de manera visual la corrección de los resultados.

5.5 Procedimiento Unitario: Calcular_Velocidad

PROPÓSITO

Se pretende probar que este método devuelve una velocidad coherente, a partir del vector más representativo calculado por el Optical Flow, y el tiempo transcurrido entre las dos imágenes.

REQUISITOS ESPECIALES

No aplicable.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se llama al método <i>calcularVelocidad(float tiempo, float maxEspacio)</i> de la clase Navegación , pasándole como parámetros el espacio recorrido y el intervalo de tiempo transcurrido.	OK	
Internamente se calcula una velocidad instantánea.	La velocidad instantánea es correcta siguiendo la fórmula velocidad = espacio / tiempo.	
Se invoca al método <i>calcularVelocidadPromedio(vel)</i> , pasándole la velocidad recién calculada.	Devuelve una velocidad promediada con las últimas 5 calculadas, siguiendo una media aritmética.	

Tabla V-5 Procedimiento Unitario: Calcular_Velocidad.

5.6 Procedimiento Unitario: Calcular_Ventana_Búsqueda

PROPÓSITO

Se pretende determinar el tamaño de la ventana de búsqueda en función de la posición del punto a tratar. Los tamaños de las ventanas varían según la posición del punto de análisis.

REQUISITOS ESPECIALES

No aplicable.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se llama al método <i>calcularVentanaBusqueda(int offX, int offY)</i> de la clase OpticalFlow, pasándole como parámetros las coordenadas del punto de análisis.	OK	
El método actualiza las variables globales que indican el tamaño de la ventana de búsqueda.	El tamaño de la ventana de búsqueda se ajusta correctamente a la posición en la que se encuentra el punto de análisis en la imagen.	

Tabla V-6 Procedimiento Unitario: Calcular_Ventana_Búsqueda

Con el fin de comprobar el funcionamiento de este método será preciso generar un código auxiliar por parte de los desarrolladores, que permita representar un punto y un rectángulo dentro de una imagen, pasándole por parámetros el origen y las dimensiones de los mismos. De esta manera será posible apreciar si los resultados son correctos o no.

5.7 Procedimiento Unitario: Calcular_Posicion_Búsqueda

PROPOSITO

Se pretende determinar la posición de la ventana de búsqueda en función del punto de análisis, y observar el desplazamiento del centro del rectángulo con respecto el punto a tratar.

REQUISITOS ESPECIALES

No aplicable.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se llama al método <i>calcularPosicionBusqueda(int posX, int posY,int *origenVentanaX, int *origenVentanaY)</i> de la clase OpticalFlow , pasándole como parámetros la posición del punto de análisis.	OK	
El método devuelve el origen de la ventana de búsqueda mediante dos coordenadas.	La posición de la ventana de búsqueda y su desplazamiento con respecto al punto de análisis es coherente con su posición dentro de la imagen.	

Tabla V-7 Procedimiento Unitario: Calcular_Posición_Búsqueda.

Al igual que en el procedimiento anterior para comprobar el funcionamiento de este método, será preciso generar un código auxiliar por parte de los desarrolladores, que permita representar un punto y un rectángulo dentro de una imagen, pasándole por parámetros el origen y las dimensiones de los mismos. De esta manera será posible apreciar si los resultados son correctos o no.

5.8 Procedimiento Unitario: Calcular_Profundidad

PROPÓSITO

Se pretende probar que la profundidad estimada es coherente con los datos de entrada introducidos. Es necesario comprobar que los cálculos son correctos para cualquier vector de la imagen.

REQUISITOS ESPECIALES

No aplicable.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se llama al método calcular <i>PintarVentanaDeph(TImage *img, TMatriz *mat)</i> de la clase ToolPaint, pasándole como parámetros la matriz del Optical Flow calculada y la imagen sobre la que debe presentar los resultados.	OK	
El método representa mediante cuadrados de distintas tonalidades las profundidades calculadas, sobre la imagen destino.	Comprobamos que las profundidades son coherentes con las calculadas a mano según ecuaciones de profundidad.	

Tabla V-8 Procedimiento Unitario: Calcular_Profundidad.

Para poder apreciar y comprobar la corrección de los cálculos realizados será necesario generar código alternativo, por parte de los desarrolladores para verificar los resultados aplicando a mano las ecuaciones correspondientes.

5.9 Procedimiento Unitario: Pintar_Optical_Flow

PROPOSITO

Se pretende probar que se representa correctamente la totalidad de los vectores de desplazamiento que conforman el Optical Flow. Los vectores se extraen de uno en uno de la matriz y se pintan sobre la imagen establecida.

Este procedimiento servirá para comprobar por parte de los desarrolladores la corrección del Optical Flow, y no se utilizará para ningún fin del sistema final.

REQUISITOS ESPECIALES

No aplicable.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se llama al método <i>PintarOpticalFlow(TImage *img, TMatriz *mat)</i> de la Clase ToolPaint, pasándole como parámetros la imagen y la matriz del Optical Flow calculado.	OK	
Internamente se invoca al método <i>PintarVector(TImage *img, int x1,int y1,int x2,int y2)</i> pasándole como parámetros la imagen y las coordenadas iniciales y finales, para cada vector encontrado en la matriz	OK	
Se genera la imagen resultado	La imagen tiene todos los vectores dibujados en las posiciones correspondientes.	

Tabla V-9 Procedimiento Unitario: Pintar_Optical_Flow

Por tratarse de una representación gráfica, los probadores no disponen de ninguna herramienta matemática para comprobar los resultados. Será necesario interpretar de forma visual los resultados, buscando errores de las zonas extremas de la imagen, o la existencia de alguna desviación común a todos los vectores.

5.10 Procedimiento Unitario: Pintar_Vector

PROPOSITO

Se pretende probar el método pinta correctamente un vector, indicando sus componentes iniciales y finales.

REQUISITOS ESPECIALES

No aplicable.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se llama al método <i>PintarVector(TImage *img, int x1,int y1,int x2,int y2)</i> de la clase ToolPaint, pasándole como parámetros la imagen y las coordenadas iniciales y finales de un vector.	OK	
El método devuelve la imagen resultado.	La imagen posee un vector dibujado en la posición indicada.	

Tabla V-10 Procedimiento Unitario: Pintar_Vector.

La metodología para comprobar si el vector se ha dibujado correctamente es simple, tan sólo es necesario comprobar por medio de algún programa de tratamiento de imágenes, las coordenadas del vector dentro de la imagen, y verificar que son las mismas que las introducidas.

5.11 Procedimiento Unitario: Pintar_Velocidad_Analógica

PROPOSITO

Se pretende probar que se muestran correctamente la velocidad analógica en el interfaz gráfico, es decir, la aguja apunta a la velocidad indicada.

REQUISITOS ESPECIALES

No aplicable.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se llama al método calcular <i>PintarVelocidadAnalogica(TImage *img, float vel)</i> de la clase ToolPaint, pasándole como parámetros la imagen del velocímetro y la velocidad estimada.	OK	
El método calcula y dibuja la aguja del cuentakilómetros sobre el interfaz del sistema.	La imagen posee un vector que tiene como origen el centro del cuentakilómetros y la velocidad introducida coincide con la marcada en la imagen.	

Tabla V-11 Procedimiento Unitario: Pintar_Velocidad_Analógica.

Es importante señalar que las velocidades en el cuentakilómetros analógico se representan con un intervalo de 2,5 Km./h. Por lo tanto si la velocidad estimada es 23 Km./h la aguja apuntará aproximadamente a 22,5 Km./h. Los probadores deberán tener en cuenta esta característica para determinar el correcto funcionamiento del método.

5.12 Procedimiento Unitario: Calcular_Posición_Futura

PROPÓSITO

Se pretende probar que se calcula la posición futura de un objeto previamente reconocido, tratando de adivinar donde se encontrará dicho objeto en la siguiente imagen.

REQUISITOS ESPECIALES

Existe una lista de objetos previamente reconocidos con al menos un objeto para alguna imagen en el directorio compartido.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se llama al método <i>estimarZonaFutura(TMatriz *matrizOptical)</i> de la clase Etiquetado, pasándole como parámetro la matriz.	El método lee la posición actual del objeto.	
Internamente se invoca el método <i>estimarPuntoFuturo(TMatriz *matrizOptical,int x1,int y1, int *x2, int *y2)</i> pasándole como parámetros las coordenadas actuales de la esquina superior izquierda del objeto.	El método devuelve las coordenadas futuras de la esquina superior izquierda.	
Internamente se invoca el método <i>estimarPuntoFuturo(TMatriz *matrizOptical,int x1,int y1, int *x2, int *y2)</i> pasándole como parámetros las coordenadas actuales de la esquina inferior derecha del objeto.	El método devuelve las coordenadas futuras de la esquina inferior derecha.	
El método calcula la posición futura de dicho objeto en la siguiente secuencia de imágenes, devolviendo sus coordenadas.	Se verifica que la posición futura es a priori correcta.	

Tabla V-12 Procedimiento Unitario: Calcular_Posición_Futura.

Con el fin de comprobar el correcto funcionamiento de este método, los probadores deberán crear un código auxiliar que les permita dibujar en la imagen los rectángulos de las posiciones actual y futura del objeto, y comprobar así si la estimación es correcta.

5.13 Procedimiento Unitario: Asignar_Etiqueta

PROPÓSITO

Se pretende comprobar que se asigna una etiqueta única a cada objeto de la lista de objetos, además de comprobar que un objeto que ha aparecido en la imagen anterior mantiene su etiqueta, salvo error.

REQUISITOS ESPECIALES

Existe una lista de objetos previamente reconocidos con al menos un objeto para alguna imagen en el directorio compartido.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se llama al método <i>asignarEtiquetas()</i> de la clase Etiquetado.	El método lee la lista de etiquetas anterior y la compara con la actual, para saber si un objeto nuevo recibe la etiqueta de uno antiguo.	
El método asigna una etiqueta única a cada objeto de la lista.	Cada objeto se identifica con su etiqueta.	

Tabla V-13 Procedimiento Unitario: Asignar_Etiqueta.

Para comprobar si el método ha actuado correctamente será necesario verificar que no existe ningún objeto que reciba la misma etiqueta, y que no se corresponda con el mismo objeto.

5.14 Procedimiento Integración: Resolución_de_las_imágenes

PROPÓSITO

Se pretende comprobar que la aplicación funciona correctamente trabajando a distintas resoluciones, siempre que sean inferiores a 720 x 576 píxeles, obteniendo resultados coherentes.

REQUISITOS ESPECIALES

Existe una lista de objetos previamente reconocidos con al menos un objeto para alguna imagen en el directorio compartido.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se inicia el análisis del sistema AMIVSI ejecutando el archivo <i>Amivsi.exe</i> .	Comienza el análisis de imágenes	
Se muestra el Optical Flow	Se analiza el Optical Flow en busca de resultado coherentes y ajustados a la resolución de entrada.	
Se muestra la velocidad estimada	Se analiza que tanto la velocidad analógica como la digital se corresponden con los cálculos realizados, salvo error de cálculo.	
Se muestra la dirección de giro.	Se analiza la dirección de giro es coherente, salvo error de cálculo.	
Se muestran las profundidades.	Se analizan las profundidades y se determinan si son coherentes, salvo errores de cálculo.	
Se muestran las etiquetas	Se analizan las etiquetas asignadas y se verifica que se representan correctamente.	

Tabla V-14 Procedimiento Integración: Resolución_de_las_imágenes.

Los probadores tendrán que generar un código auxiliar que les permita representar el Optical Flow con el fin de estimar si es correcto. El resto de comprobaciones se podrán seguir teniendo en cuenta los correspondientes procedimientos de pruebas unitarios.

5.15 Procedimiento Integración: Fichero_de_configuración

PROPÓSITO

Se pretende probar como afecta a los resultados obtenidos las variaciones sobre el fichero de configuración. Con la finalidad de analizar cuales son los valores óptimos que se adaptan a la mayoría de las secuencias.

Parámetros de configuración:

COEFICIENTE_VENTANA_MAX
COEFICIENTE_VENTANA_MIN
COEFICIENTE_PATRON
UMBRAL
ESPACIO_ENTRE_VECTORES
PORCENTAJE_VALIDO

REQUISITOS ESPECIALES

No aplicable.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se inicia el análisis del sistema AMIVSI ejecutando el archivo <i>Amivsi.exe</i> .	Comienza el análisis de imágenes	
Se calcula el Optical Flow	OK, se anotan los resultados	
Se Aumenta COEFICIENTE_VENTANA_MAX y se recalcula el Optical Flow	El sistema debe tardar más y obtener mejores resultados que los anotados previamente.	
Se Aumenta COEFICIENTE_VENTANA_MIN y se recalcula el Optical Flow	El sistema debe tardar más y obtener mejores resultados que los anotados previamente.	
Se Aumenta COEFICIENTE_PATRON y se recalcula el Optical Flow	El sistema debe tardar más y obtener mejores resultados que los anotados previamente.	
Se Aumenta MARCO y se recalcula el Optical Flow	El sistema debe tardar más y obtener mejores resultados que los anotados previamente.	

Se Aumenta ESPACIO_ENTRE_VECTORES y se recalcula el Optical Flow	El sistema debe tardar más y obtener mejores resultados que los anotados previamente.	
Se Aumenta PORCENTAJE_VALIDO y se recalcula el Optical Flow	El sistema debe tardar más y obtener mejores resultados que los anotados previamente.	

Tabla V-15 Procedimiento Integración: Fichero_de_configuración.

5.16 Procedimiento Sistema: Combinación_de_escena_de_carretera

PROPÓSITO

Se pretende evaluar la fiabilidad de los resultados obtenidos para una secuencia de imágenes determinadas. De las cuales se conocen parámetros reales de velocidad, profundidades y objetos.

REQUISITOS ESPECIALES

Existen dentro del directorio compartido la secuencia de imágenes detallada, con sus listas de objetos reconocidos.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se inicia el análisis del sistema AMIVSI ejecutando el archivo <i>Amivsi.exe</i> .	Comienza el análisis de imágenes	
Se muestra la velocidad	Se analiza el porcentaje de error de la estimación de velocidad con respecto a la del cuentakilómetros del vehículo, en una serie de imágenes de la secuencia.	
Se muestra la profundidad	Se analiza el porcentaje de error de la profundidad en una serie de imágenes de la secuencia.	
Se muestran las etiquetas	Se analiza el porcentaje de error de las etiquetas.	

Tabla V-16 Procedimiento Sistema: Combinación_de_escena_de_carretera..

Debido a la multitud de imágenes que componen una secuencia los probadores tendrán que seleccionar determinados frames, y descartar el resto para no prolongar en exceso la duración de la prueba. Este tipo de consideración serán tenidas en cuenta para el análisis de la velocidad y la profundidad, en cambio, la verificación del etiquetado deberá realizarse sobre todos los frames que contengan al menos un objeto.

Para cada imagen analizada los probadores deberán anotar las características de luminosidad, movimiento, formas y la velocidad de referencia, dichos datos serán necesarios para elaborar un buen informe de resultados.

5.17 Procedimiento Sistema: Escena_de_carretera_velocidad

PROPOSITO

Se pretende evaluar la fiabilidad de los resultados obtenidos con respecto a la velocidad de movimiento para una secuencia de imágenes determinadas. De las cuales se conoce la velocidad real de movimiento. Las pruebas se realizarán con secuencias de imágenes a velocidades de 10, 20, 30, 40 y 50 Km./h.

REQUISITOS ESPECIALES

Existen dentro del directorio compartido la secuencia de imágenes que se desea analizar.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se inicia el análisis del sistema AMIVSI ejecutando el archivo <i>Amivsi.exe</i> .	Comienza el análisis de imágenes	
Para cada imagen de la secuencia se muestra la velocidad calculada.	Se analiza el porcentaje de error de la estimación de velocidad con respecto a la del cuentakilómetros del vehículo, tomando como referencia la media de los valores obtenidos a lo largo de toda la secuencia.	

Tabla V-17 Procedimiento Sistema: Escena_de_carretera_velocidad.

La secuencia elegida deberá presentar una velocidad constante para poder determinar la validez de los resultados.

Para cada imagen analizada los probadores deberán anotar las características de luminosidad, movimiento, formas y la velocidad de referencia, dichos datos serán necesarios para elaborar un buen informe de resultados.

5.18 Procedimiento Sistema: Escena_de_carretera_profundidad

PROPÓSITO

Se pretende evaluar la fiabilidad de los resultados obtenidos con respecto a la profundidad de la escena para unas imágenes determinadas. De las cuales se conocen las relaciones de profundidad entre distintos objetos o formas de la imagen. El conocimiento de las profundidades será mediante percepción visual, dejando a la interpretación del probador la referencia de comparación.

REQUISITOS ESPECIALES

Existen dentro del directorio compartido la secuencia de imágenes que se desea analizar.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se inicia el análisis del sistema AMIVSI ejecutando el archivo <i>Amivsi.exe</i> .	Comienza el análisis de imágenes	
Se muestra las profundidad calculadas para cada imagen.	Se analiza el porcentaje de error de la profundidad en una serie de imágenes de la secuencia. Comparando punto a punto la estimación con la interpretación del probador, obteniendo así evaluaciones favorables o desfavorables para cada punto.	

Tabla V-18 Procedimiento Sistema: Escena_de_carretera_profundidad.

Debido a la multitud de imágenes que componen una secuencia los probadores tendrán que seleccionar determinados frames, y descartar el resto para no prolongar en exceso la duración de la prueba.

Para cada imagen analizada los probadores deberán anotar las características de luminosidad, movimiento, formas y la velocidad de referencia, dichos datos serán necesarios para elaborar un buen informe de resultados.

5.19 Procedimiento Sistema: Escena_de_carretera_etiquetado

PROPÓSITO

Se pretende evaluar la fiabilidad de los resultados obtenidos con respecto al etiquetado de objetos reconocidos previamente, para una secuencia de imágenes. De las cuales se conocen las posiciones de los objetos en cada imagen, el número objetos totales de la secuencia y el número de imágenes en las que aparece cada objeto.

REQUISITOS ESPECIALES

Existen dentro del directorio compartido la secuencia de imágenes detallada, con sus listas de objetos reconocidos.

PASOS DE PROCEDIMIENTO E INFORME DE PRUEBAS

Paso del procedimiento	Criterio de paso	Paso/Fallo
Se inicia el análisis del sistema AMIVSI ejecutando el archivo <i>Amivsi.exe</i> .	Comienza el análisis de imágenes	
Se muestran las etiquetas asignadas para cada imagen.	Se analiza el número de etiquetas asignadas y se compara con el número de objetos reconocidos, además se compara el numero de etiquetas del mismo objeto asignadas en distintas imágenes con el numero de imágenes en las que aparece cada objeto.	

Tabla V-19 Procedimiento Sistema: Escena_de_carretera_etiquetado.

La verificación del etiquetado deberá realizarse sobre todos los frames que contengan al menos un objeto.

Para cada imagen analizada los probadores deberán anotar las características de luminosidad, movimiento, formas y la velocidad de referencia, dichos datos serán necesarios para elaborar un buen informe de resultados.

Capítulo VI. Evaluación de los resultados obtenidos

1 Introducción

Este capítulo realiza un análisis exhaustivo de los resultados obtenidos, valiéndose de procedimientos de pruebas especificados en el plan de verificación, con el fin de determinar el alcance del sistema de análisis desarrollado y poder elaborar un informe detallado de fiabilidad de los resultados obtenidos.

El citado informe se nutre de los resultados de procedimientos de pruebas concernientes a pruebas de sistema, con las que se persigue analizar cada uno de los objetivos propuestos. Tanto las pruebas de unidad como las de integración no se contemplan en el informe porque su finalidad es comprobar la corrección del sistema desde el punto de vista de análisis y diseño de la solución propuesta, y no de los resultados obtenidos.

Las secuencias y los frames son clasificados por el juicio de los probadores atendiendo a las siguientes características de movimiento, velocidad, visibilidad y formas. Dicha clasificación genera diferentes informes que permiten discutir los resultados obtenidos.

Cada una de las cuatro características admite tres posibles valores (bajo, medio, alto), que se evaluarán según las siguientes apreciaciones:

El Movimiento trata de describir el tipo de desplazamiento que sufre el vehículo, se clasifica en tres categorías que se describen a continuación:

- Bajo: Desplazamiento en línea recta, sin cambios de rasante. Se representará en las tablas mediante el valor cero.
- Medio: Desplazamiento en curva no muy pronunciada (cambios de carril, etc.). Se representará en las tablas mediante el valor cincuenta.
- Alto: Desplazamiento en curva pronunciada, cambio de rasante (cruces, rotondas, etc.). Se representará en las tablas mediante el valor cien.

La Visibilidad trata de describir las características de luminosidad, efectos de luz y sombra que posee la escena. La unidad de medida en la que se basa esta apreciación la determina el valor de la mediana del histograma [GALER 2003] de la escena.

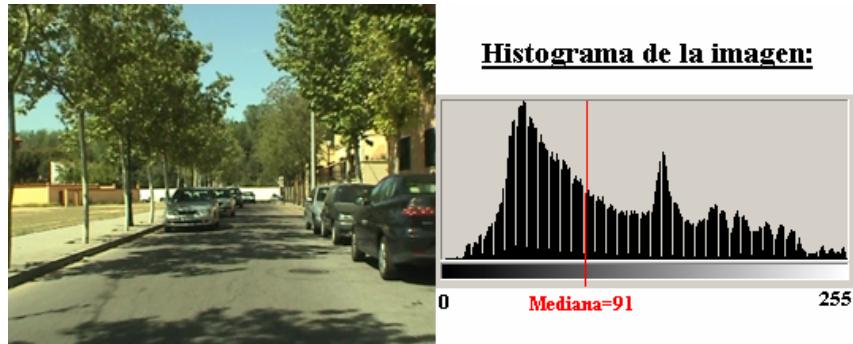


Figura VI-1 Mediana del un histograma.

Para establecer una clasificación se ha analizado la luminosidad media de 322 frames, las mediadas obtenidas representan el siguiente gráfico:

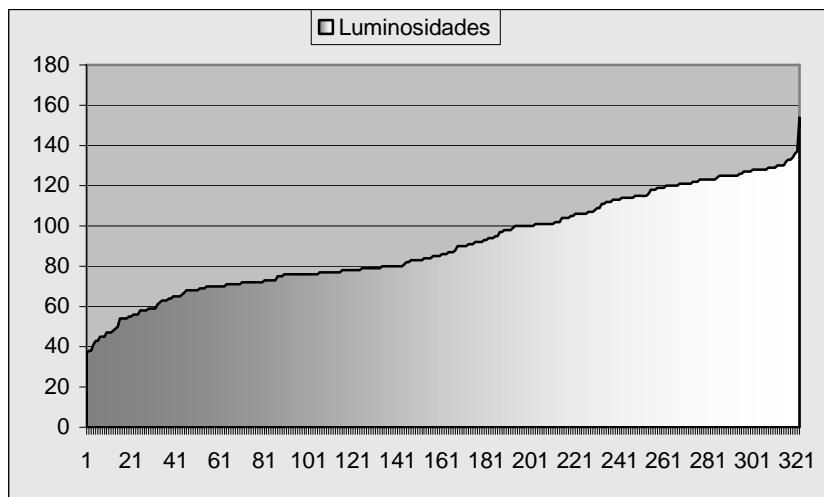


Figura VI-2 Gráfico de luminosidades.

Extraemos de la *Figura VI 2* las tres categorías que se describen a continuación:

- Bajo: Efecto de contraluz, Iluminación escasa, sombra casi total. Valores de luminosidad de 37 a 77.
- Medio: Iluminación moderada, sombra parcial. Valores de luminosidad de 78 a 102.
- Alto: Iluminación total, ausencia de sombras. Valores de luminosidad de 103 a 154.

La Velocidad se clasifica en:

- Bajo: de 0 a 19 Km./h.
- Medio: de 20 a 39 Km./h.
- Alto: de 40 a 50 Km./h.

Las Formas tratan de describir la cantidad de bordes de la escena. La unidad de medida en la que se basa esta apreciación la determinan los puntos de análisis del Optical Flow; que como hemos visto, se corresponden con los puntos en los que se ha intentado determinar el PatternMatching. Para establecer una clasificación se han contabilizado los puntos de análisis de 322 frames, los resultados obtenidos se representan en el siguiente gráfico:

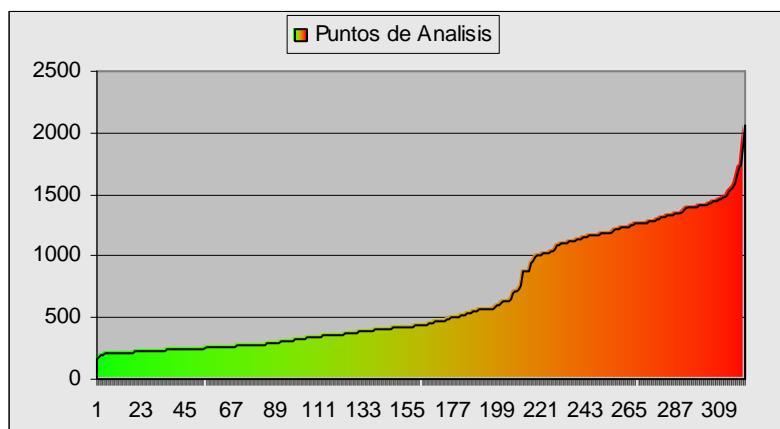


Figura VI-3 Gráfico de Puntos de análisis.

Estos puntos han sido filtrados de tal manera que representan los bordes de la escena. Se clasifica en tres categorías que se describen a continuación:

- Bajo: Ausencia de bordes, la escena no posee coches, casas, muros, etc. Existen grandes zonas uniformes. Valores de 169 a 337.
- Medio: Se ven algunas formas como coches, casas, muros, etc. Existen algunas zonas uniformes. Valores de 338 a 878.
- Alto: Presencia casi total de formas como coches, casas, muros, etc. No existen zonas uniformes. Valores de 879 a 2054.

Las pruebas de sistema realizadas se pueden dividir en cuatro grupos:

1.1 Escena_de_carretera_velocidad

Se ha pasado el procedimiento número 5.17 del plan de verificación, escena_de_carretera_velocidad, con tres secuencias de imágenes independientes, a velocidad constante.

Para medir la velocidad de una escena se compara la velocidad del cuentakilómetros del vehículo con la estimación proporcionada por el sistema de cada frame, y se determina el porcentaje de error.

1.2 Escena_de_carretera_profundidad

Se ha pasado el procedimiento número 5.18 del plan de verificación, escena_de_carretera_profundidad, con 11 frames de tres secuencias de imágenes independientes.

Para medir la profundidad de una escena, se determina el número de puntos donde se ha calculado la profundidad, el probador estima si la tonalidad (que representa distintas profundidades) del punto se corresponde con la real. El mapa de profundidades se restringe a tres tonalidades para que el probador pueda contar con facilidad los puntos erróneos.

1.3 Escena_de_carretera_etiquetado

Se ha pasado el procedimiento número 5.19 del plan de verificación, escena_de_carretera_etiquetado, con tres secuencias de imágenes independientes, en la que figura al menos un objeto.

Para el etiquetado se seleccionan los frames donde existe algún objeto. Para cada objeto se cuenta el número de frames en los que aparece, y el número de etiquetas diferentes que se ha asignado.

1.4 Combinación_de_escena_de_carretera

Se ha pasado el procedimiento número 5.16 del plan de verificación, combinación_de_escena_de_carretera, con dos secuencias de imágenes independientes. Se pretende analizar de manera global los objetivos bajo condiciones diferentes, para ello se deben recoger las muestras más representativas que combinen valores dispares de velocidad, formas, carretera, condiciones ambientales, etc.

Para medir la velocidad de una escena global se seleccionan frames de 20 en 20 a lo largo de toda la secuencia. Se compara la velocidad del cuentakilómetros del vehículo con la estimación proporcionada por el sistema, y se determina el porcentaje de error para cada frame.

La profundidad de una escena analiza de manera análoga al procedimiento escena_de_carretera_profundidad, con la diferencia de seleccionar frames de 40 en 40 a lo largo de toda la secuencia.

El etiquetado se realiza de manera análoga al procedimiento de escena_de_carretera_etiquetado.

2 Pruebas de Velocidad con Secuencias Cortas

2.1 Primera escena_de_carretera_velocidad

Se ha seleccionado la secuencia de imágenes extraídas del video de pruebas *30-1.avi*. Se analiza del frame 20 al 60, que transcurren en 8 segundos a velocidad constante de 30 Km./h. La secuencia posee en conjunto una luminosidad de valor medio, los puntos de análisis toman un valor alto, y el movimiento se cataloga como bajo-medio.

La secuencia transcurre por una carretera de una zona residencial, el vehículo se desplaza en línea recta con una ligera rectificación de la trayectoria en la mitad de la secuencia. La carretera es de doble sentido sin tráfico y con abundante presencia de árboles y sombras. Se observa una ligera pendiente negativa.

La siguiente tabla de datos expresa los resultados obtenidos para cada frame:

Frame	Vel. Estimada	Vel. Referencia	Acierto (%)	Movimiento	Luminosidad	Ptos. de Análisis
20	28,34	30	94,46	0	77	1021
22	28,56	30	95,2	0	68	1021
24	20,13	30	67,1	0	77	1148
26	25,36	30	84,53	0	85	1150
28	25,36	30	84,53	0	127	1212
30	25,36	30	84,53	0	119	991
32	25,36	30	84,53	50	104	1011
34	25,36	30	84,53	50	98	949
36	25,36	30	84,53	50	90	1019
38	25,36	30	84,53	0	79	1137
40	37,02	30	76,6	0	76	1098
42	35,48	30	81,73	0	78	1140
44	35,48	30	81,73	0	71	1236
46	31,85	30	93,83	0	65	1366
48	31,85	30	93,83	0	59	1392
50	36,79	30	77,36	0	90	1350
52	36,79	30	77,36	0	80	1241
54	36,79	30	77,36	0	54	1355
56	36,79	30	77,36	0	76	1422
58	36,79	30	77,36	50	72	1184
Vel. Media	30,509	Acierto total	83,15		82	1172,15

Tabla VI-1 Resultados 1º Escena_de_carretera_velocidad.

En la tabla se observa como los frames aparecen de dos en dos, esto se debe al mecanismo de restricción de tiempo, que ha ido saltando un frame, para poder alcanzar la condición de análisis en tiempo real.

De la tabla VI 2 se extraen los dos gráficos siguientes, que permite comparar los resultados de velocidad obtenidos a lo largo de toda la secuencia, con la evolución de las características de la secuencia:

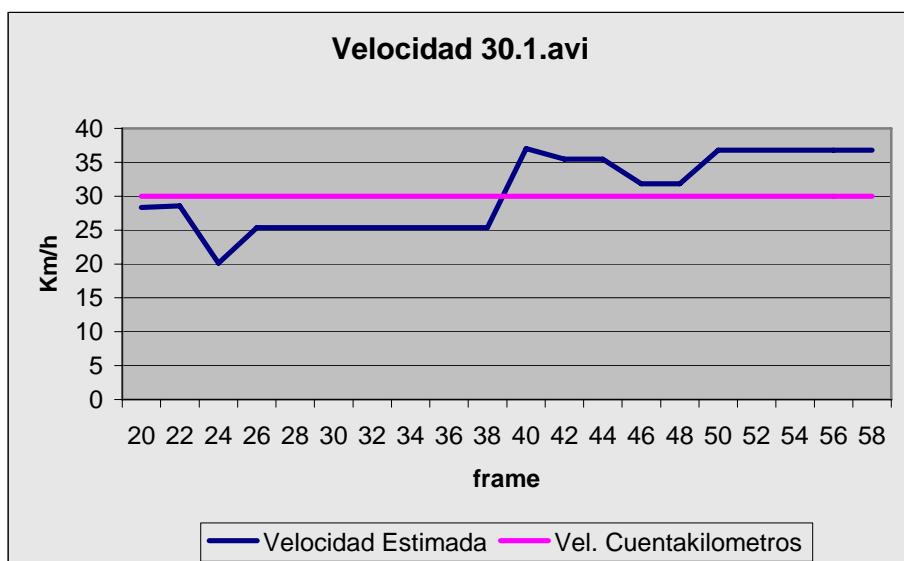


Figura VI-4 Comparativa velocidad secuencia 30-1.avi.

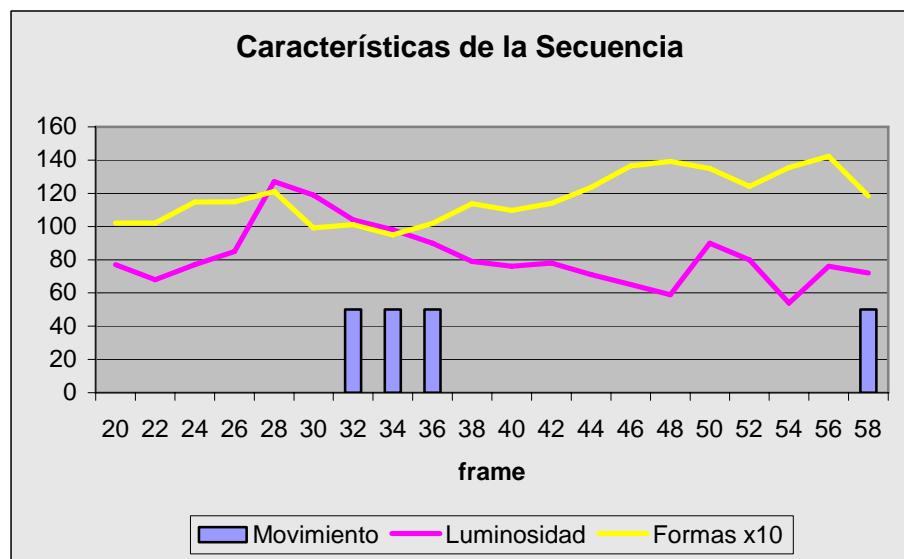


Figura VI-5 Características de la secuencia 30-1.avi.

De estos gráficos se deduce que las estimaciones de velocidad a 30 Km./h. se realizan con un notable porcentaje de acierto (83.5%), bajo condiciones medias de luminosidad, gran presencia de formas en la secuencia y siguiendo un movimiento bajo-medio. Un dato importante es que las estimaciones que se suceden en un frame que presente movimiento medio, describen desviaciones similares; resaltando la importancia de esta característica para el cálculo de la velocidad.

2.2 Segunda Escena_de_carretera_velocidad

Se ha seleccionado la secuencia de imágenes extraídas del video de pruebas *40-1.avi*. Se analiza del frame 1 al 50, que transcurren en 10 segundos a velocidad constante de 40 Km./h. La secuencia posee en conjunto una luminosidad de valor medio, los puntos de análisis toman un valor alto, y el movimiento se cataloga como medio-alto.

La secuencia transcurre por una carretera de una zona residencial, el vehículo se desplaza por una curva no muy pronunciada, con una ligera pendiente negativa. Es un trazado de doble sentido sin tráfico y presencia total de árboles a ambos lados de la carretera que cubren casi por completo la parte superior de la imagen. Se aprecia como el asfalto está bacheado en las vibraciones que sufre la cámara. La secuencia presenta variaciones de luminosidad debido al efecto de la claros y sobras que produce la vegetación.

La siguiente tabla de datos expresa los resultados obtenidos para cada frame:

Frame	Vel. Estimada	Vel. Referencia	Acierto (%)	Movimiento	Luminosidad	Ptos. de Análisis
2	33,34	40	83,35	100	79	1452
4	35,43	40	88,57	50	83	1413
6	33,03	40	82,57	50	83	1307
8	41,92	40	95,2	50	87	1662
10	41,92	40	95,2	50	84	1561
12	41,92	40	95,2	50	85	1527
14	41,92	40	95,2	50	86	1499
16	41,92	40	95,2	50	80	1445
18	38,2	40	95,5	100	83	1463
20	43,18	40	92,05	100	77	1585
22	43,15	40	92,12	50	94	1397
24	44,17	40	89,57	50	77	1460
26	43,47	40	91,32	50	75	1474
28	40,76	40	98,1	50	72	1729
30	40,76	40	98,1	100	83	1907
32	40,76	40	98,1	50	79	2054
34	40,76	40	98,1	50	78	1376

36	40,76	40	98,1	50	76	1544
38	40,76	40	98,1	50	73	1452
40	29,15	40	72,87	50	72	1394
42	33,96	40	84,9	50	76	1716
44	38	40	95	50	70	1422
46	46,31	40	84,22	50	91	1389
48	46,18	40	84,55	50	71	1479
50	45,15	40	87,12	50	91	1265
Vel. Media	40,27	Acierto total	91,53		80	1518

Tabla VI-2 Resultados 2º Escena_de_carretera_velocidad.

En la tabla se observa como los frames aparecen de dos en dos, esto se debe al mecanismo de restricción de tiempo, que ha ido saltando un frame, para poder alcanzar la condición de análisis en tiempo real.

De la tabla VI 2 se extraen los dos gráficos siguientes, que permite comparar los resultados de velocidad obtenidos a lo largo de toda la secuencia, con la evolución de las características de la secuencia:

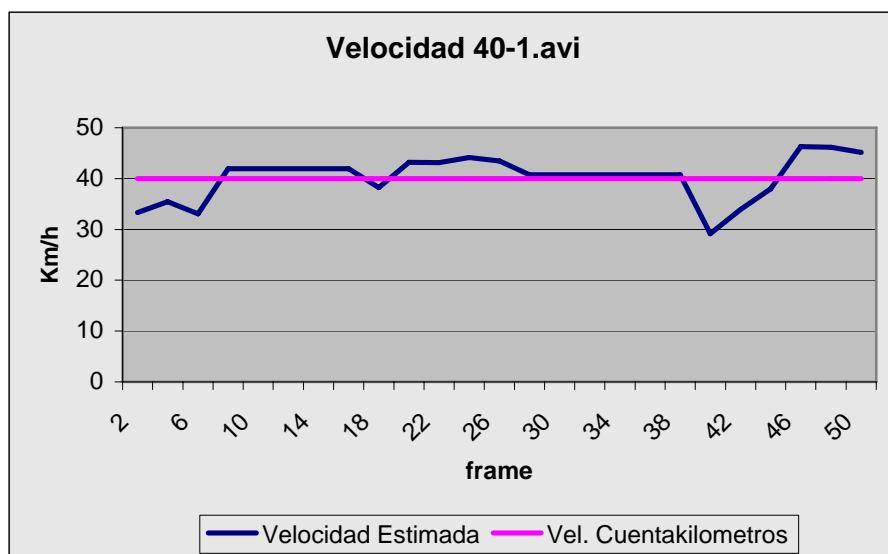


Figura VI-6 Comparativa velocidad secuencia 40-1.avi.

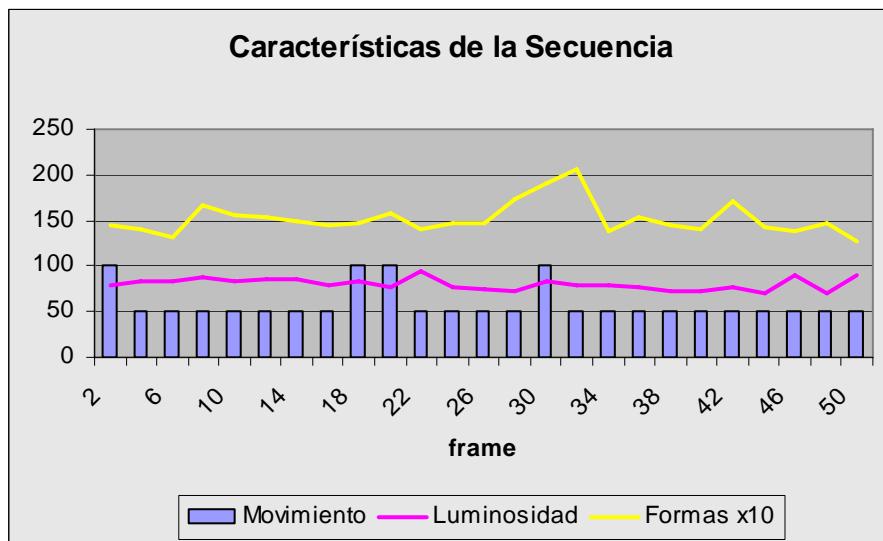


Figura VI-7 Características de la secuencia 40-1.avi.

De estos gráficos se deduce que las estimaciones de velocidad a 40 Km./h. se realizan con un elevado porcentaje de acierto (91.5%), bajo condiciones medias de luminosidad, numerosa presencia de formas en la secuencia y siguiendo un movimiento medio-alto. Es importante resaltar la escasa variabilidad que presenta la luminosidad en toda la secuencia, que proporciona estabilidad en las estimaciones, ya que no se producen cambios bruscos de visibilidad.

2.3 Tercera Escena_de_carretera_velocidad

Se ha seleccionado la secuencia de imágenes extraídas del video de pruebas 10-1.avi. Se analiza del frame 1 al 50, que transcurren en 10 segundos a velocidad constante de 10 Km./h. La secuencia posee en conjunto una luminosidad de valor alto, los puntos de análisis toman un valor alto, y el movimiento se cataloga como bajo-medio.

La secuencia transcurre a velocidad muy baja por una carretera de una zona residencial, el vehículo se desplaza en línea recta por un terreno llano. Existe vegetación alta a ambos lados de la carretera.

La siguiente tabla de datos expresa los resultados obtenidos para cada frame:

Frame	Vel. Estimada	Vel. Referencia	Acierto (%)	Movimiento	Luminosidad	Ptos. de Análisis
2	6,69	10	66,9	50	90	1419

4	6,69	10	66,9	50	87	1336
6	6,69	10	66,9	50	83	1174
8	6,69	10	66,9	50	86	1185
10	6,69	10	66,9	50	91	1157
12	6,69	10	66,9	50	94	1164
14	12,32	10	76,8	50	100	1189
16	12,99	10	70,1	0	93	1175
18	11,49	10	85,1	0	76	1261
20	12,57	10	74,3	0	79	1299
22	12,92	10	70,8	0	90	1415
24	13,44	10	65,6	0	105	1237
26	13,29	10	67,1	0	100	1317
28	12,52	10	74,8	0	98	1287
30	12,52	10	74,8	0	102	1290
32	17,73	10	22,7	0	112	1243
34	21,33	10	0	0	121	1214
36	21,33	10	0	0	121	1163
38	21,42	10	0	0	122	1114
40	17,92	10	20,8	0	122	1113
42	13,57	10	64,3	0	121	1192
44	13,57	10	64,3	0	119	1131
46	13,57	10	64,3	0	121	1046
48	13,57	10	64,3	0	127	878
Vel. Media	12,84	Acierto total	56,72		103	1208

Tabla VI-3 Resultados 3º Escena_de_carretera_velocidad.

En la tabla se observa como los frames aparecen de dos en dos, esto se debe al mecanismo de restricción de tiempo, que ha ido saltando un frame, para poder alcanzar la condición de análisis en tiempo real.

De la tabla VI 3 se extraen los dos gráficos siguientes, que permite comparar los resultados de velocidad obtenidos a lo largo de toda la secuencia, con la evolución de las características de la secuencia:

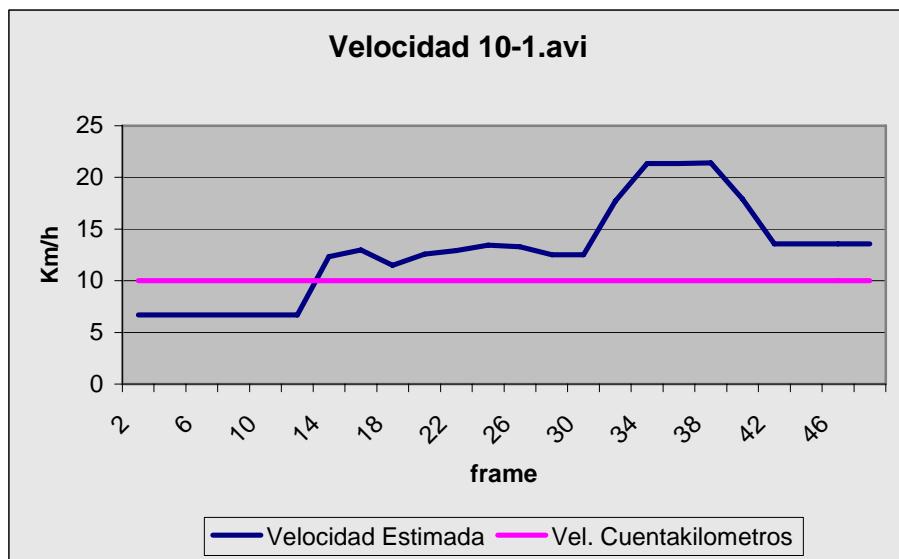


Figura VI-8 Comparativa velocidad secuencia 10-1.avi.

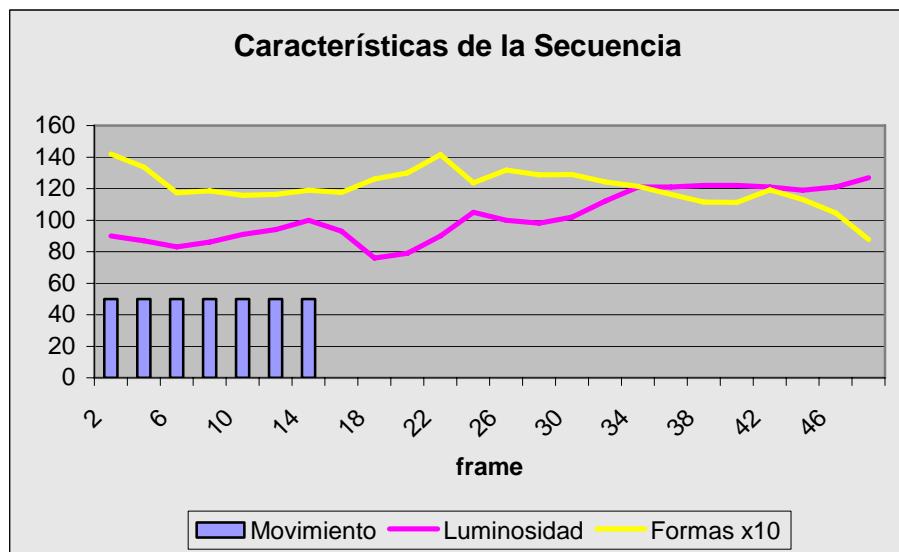


Figura VI-9 Características de la secuencia 10-1.avi.

De estos gráficos se deduce que las estimaciones de velocidad a 10 Km./h. se realizan con un moderado porcentaje de acierto (56.7%), bajo condiciones altas de luminosidad y formas, describiendo un movimiento bajo-medio. Se puede destacar como a medida que aumenta la luminosidad y disminuyen los puntos de análisis, los resultados obtenidos presentan mayor porcentaje de error.

2.4 Evaluación de los tres procedimientos Escena_de_carretera_velocidad

Como resumen de los tres procedimientos realizados obtenemos la siguiente tabla, con su correspondiente gráfico:

Secuencia	Vel. Estimada	Vel. Referencia	Acierto (%)	Movimiento	Luminosidad	Ptos. de Análisis
30.1.avi	30,5	30	83,1	Bajo	Medio	Alto
40-1.avi	40,2	40	91,5	Medio – Alto	Medio	Alto
10-1.avi	12,8	10	56,7	Bajo – Medio	Alto	Alto

Tabla VI-4 Resumen procedimientos Escena_de_carretera_velocidad.

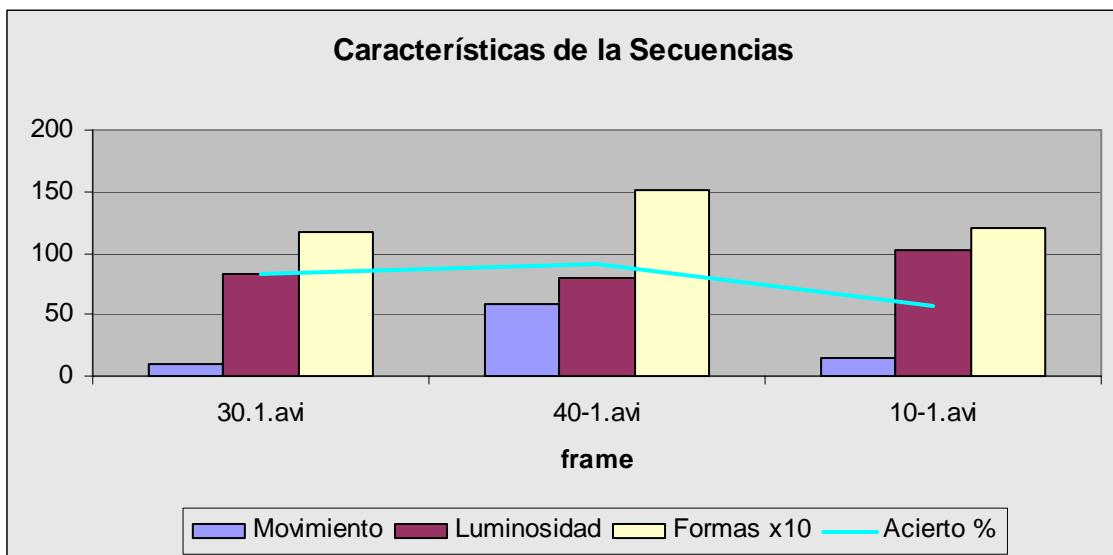


Figura VI-10 Características y porcentaje de acierto de las secuencias de velocidad.

Los mejores resultados se obtienen a unas condiciones de velocidad y luminosidad medias, sin demasiados cambios de movimiento y a velocidad media.

Los peores resultados se obtienen bajo una luminosidad variable y elevada, además de las secuencias en las que se produzcan numerosos cambios de movimiento.

No se aprecia ninguna influencia del número de puntos de análisis en el cálculo de la velocidad.

3 Pruebas de Profundidad con Secuencias Cortas

3.1 Primera Escena_de_carretera_profundidad

Se ha seleccionado la secuencia de imágenes extraídas del video de pruebas *30-6.avi*. Se analizan 11 frames, comenzando en el 20 con un incremento de 40, la secuencia dura 1 minuto y 20 segundos, a una velocidad de 30 Km./h, con unas características de luminosidad media-alta, formas con valor medio y movimiento variable.

La secuencia trascurre a velocidad media por una carretera de doble carril y con mediana, el vehículo se desplaza sobre una curva ligera de pendiente positiva. A lo largo de la secuencia se realizan dos cambios de carril y una rotonda en la que se toma la primera salida. Existe vegetación baja a ambos lados de la carretera, y sobre la mediana una serie de farolas de alumbrado muy altas.

La siguiente tabla de datos expresa los resultados obtenidos para cada frame:

Frame	Ptos. Profundidad	Ptos. de error	Acierto	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
20	197	62	68,52	30	0	88	430
40	149	28	81,2	30	50	76	307
80	231	63	72,72	30	0	65	362
120	259	51	80,3	30	0	79	391
160	199	65	67,33	30	50	86	370
200	191	48	74,86	30	0	92	380
240	176	77	56,25	30	50	107	426
280	169	51	69,82	30	0	130	367
320	212	57	73,11	30	0	134	707
360	169	32	81,06	30	100	128	515
400	64	11	82,81	30	0	112	483
Total	183	49	73,45			100	431

Tabla VI-5 Resultados 1º Escena_de_carretera_profundidad.

De la tabla VI 5 se extraen los dos gráficos siguientes, que permite comparar los resultados de profundidad obtenidos a lo largo de toda la secuencia, con la evolución de las características de la secuencia:

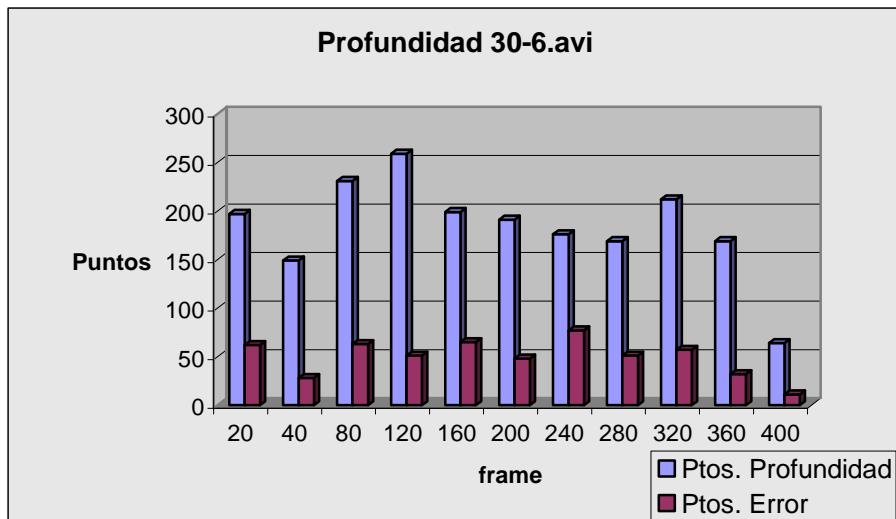


Figura VI-11 Comparativa velocidad secuencia 30-6.avi.

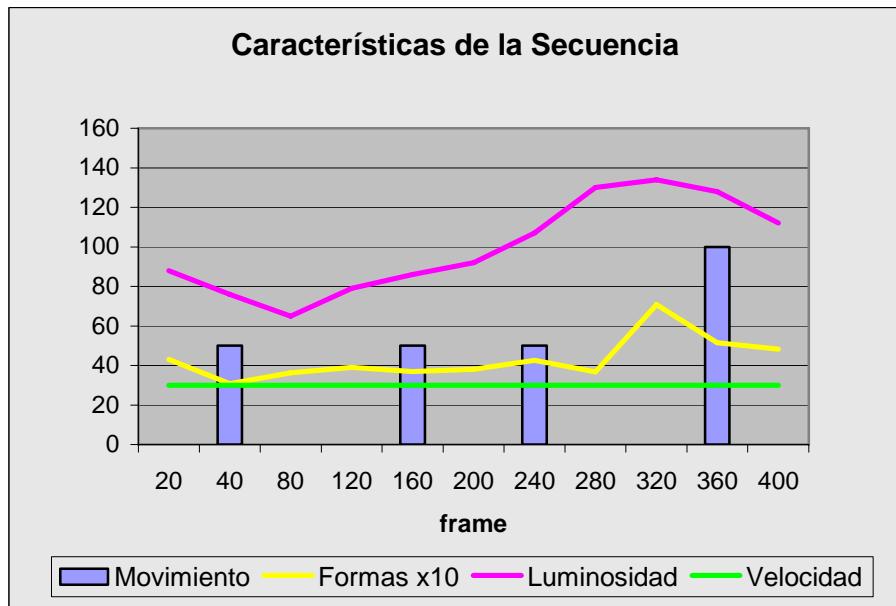


Figura VI-12 Características de la secuencia 30-1.avi.

De estos gráficos se deduce que las estimaciones de profundidad a 30 Km./h. se realizan con un notable porcentaje de acierto (73.4%), bajo condiciones altas de luminosidad y formas de valor medio, describiendo un movimiento con numerosos cambios. Se observa como no existe una relación entre el número de puntos de análisis y los puntos de profundidad.

3.2 Segunda Escena_de_carretera_profundidad

Se ha seleccionado la secuencia de imágenes extraídas del video de pruebas *10-Personal.avi*. Se analizan 11 frames consecutivos, del 50 al 60, la secuencia dura 2 segundos, a una velocidad de 10 Km./h, con unas características de luminosidad media-alta, formas con valor alto y movimiento bajo.

La secuencia recoge como el vehículo se acerca a velocidad baja a una persona en el medio de la carretera.

La siguiente tabla de datos expresa los resultados obtenidos para cada frame:

Frame	Ptos. Profundidad	Ptos. de error	Acierto	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
50	345	106	69,27	10	0	114	1089
51	357	103	71,14	10	0	109	1055
52	405	94	76,79	10	0	106	1110
53	369	103	72,08	10	0	104	1109
54	362	131	63,81	10	0	101	1109
55	431	105	75,63	10	0	98	1184
56	466	98	78,96	10	0	95	1226
57	416	96	76,92	10	0	92	1173
58	406	88	78,32	10	0	87	1241
59	312	87	72,11	10	0	84	1176
60	310	138	55,48	10	0	85	1269
Total	380	104	71,86	10		98	1158

Tabla VI-6 Resultados 2º Escena_de_carretera_profundidad.

De la tabla VI 6 se extraen los dos gráficos siguientes, que permite comparar los resultados de profundidad obtenidos a lo largo de toda la secuencia, con la evolución de las características de la secuencia:

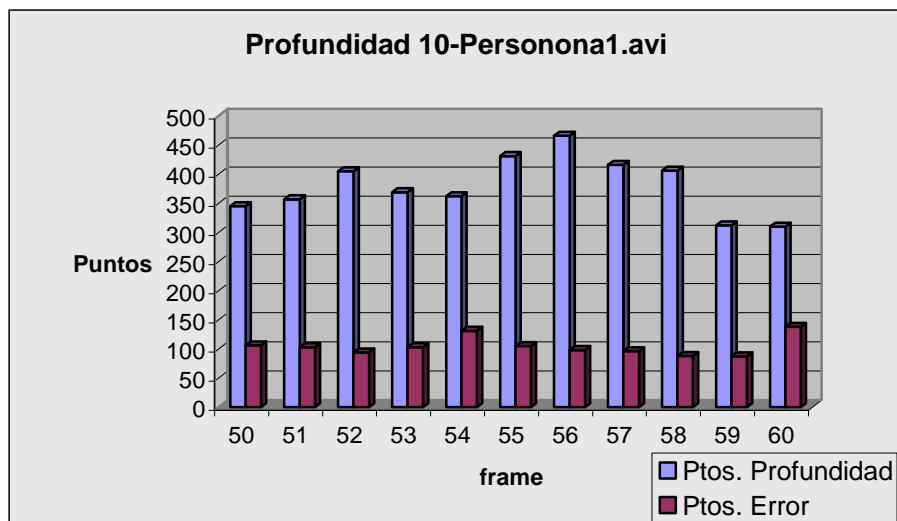


Figura VI-13 Comparativa velocidad secuencia 10-Persona1.avi.

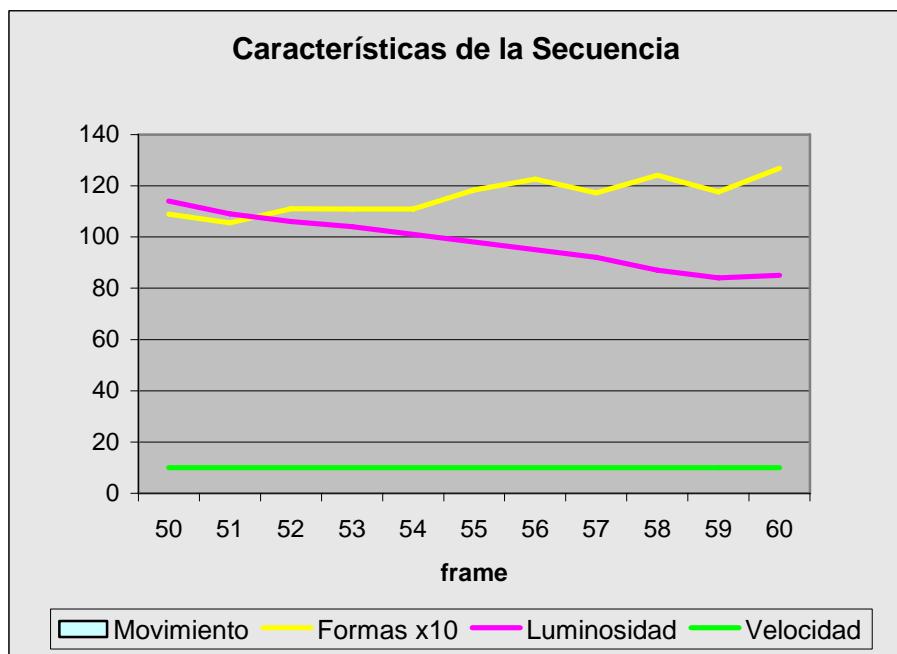


Figura VI-14 Características de la secuencia 10-Persona1.avi.

De estos gráficos se deduce que las estimaciones de profundidad a 10 Km./h. se realizan con un notable porcentaje de acierto (71.8), bajo condiciones de luminosidad medio-alto y formas de valor medio, describiendo un movimiento bajo.

3.3 Tercera Escena_de_carretera_profundidad

Se ha seleccionado la secuencia de imágenes extraídas del video de pruebas 50-2.avi. Se analizan 11 frames aleatorios, la secuencia dura 13,2 segundos, a una velocidad de 50 Km./h, con unas características de luminosidad alta, formas con valor medio y movimiento bajo-medio.

La secuencia transcurre a 50 Km./h por una carretera sentido único. El desplazamiento se realiza en línea recta sobre una ligera pendiente positiva. Existe vegetación baja a ambos lados de la carretera.

La siguiente tabla de datos expresa los resultados obtenidos para cada frame:

Frame	Ptos. Profundidad	Ptos. de error	Acierto	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
6	143	48	66,43	50	50	121	614
12	122	50	59,01	50	50	123	645
18	136	58	57,35	50	50	121	626
24	161	54	66,45	50	50	120	881
30	115	56	51,30	50	50	109	573
36	63	14	77,77	50	0	115	477
42	75	43	42,66	50	0	119	557
48	120	44	63,33	50	0	120	569
54	79	35	55,69	50	0	115	477
60	85	54	36,47	50	0	115	628
66	90	59	34,44	50	0	119	530
Total	108	47	55,54			118	598

Tabla VI-7 Resultados 3º Escena_de_carretera_profundidad.

De la tabla VI 7 se extraen los dos gráficos siguientes, que permite comparar los resultados de profundidad obtenidos a lo largo de toda la secuencia, con la evolución de las características de la secuencia:

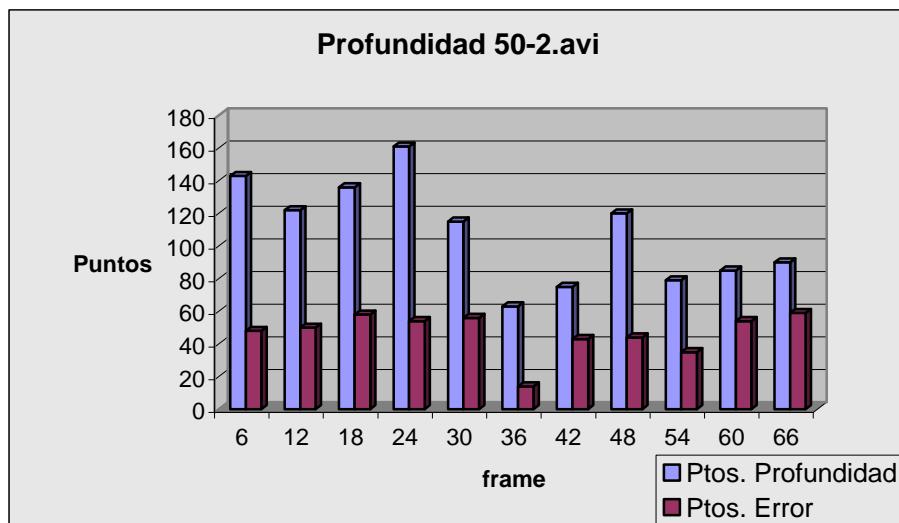


Figura VI-15 Comparativa velocidad secuencia 50-2.avi.

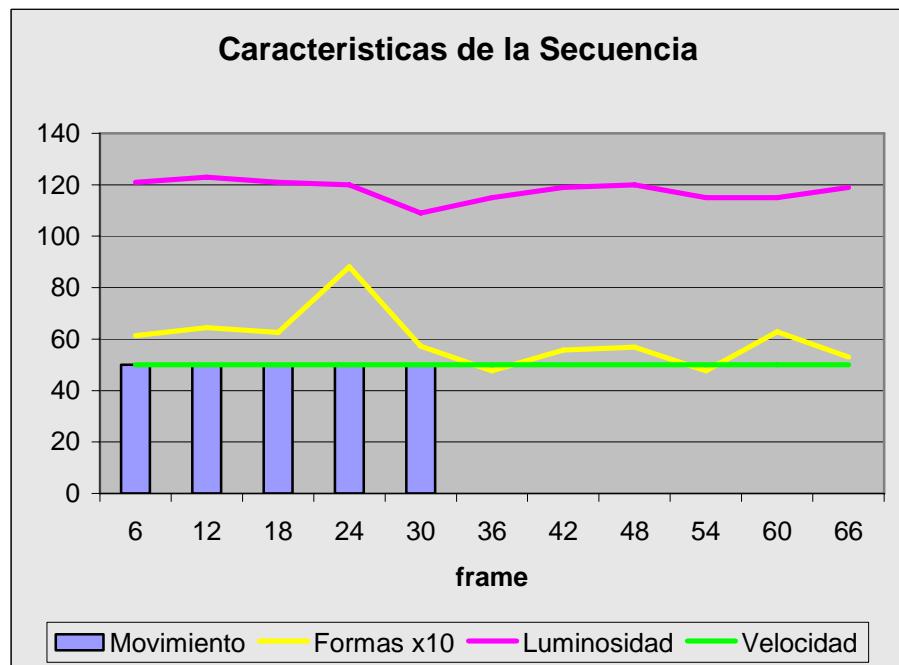


Figura VI-16 Características de la secuencia 50-2.avi.

De estos gráficos se deduce que las estimaciones de profundidad a 50 Km./h. se realizan con un moderado porcentaje de acierto (55.5), bajo condiciones de luminosidad alta y formas de valor medio, describiendo un movimiento bajo-medio.

3.4 Evaluación de los tres procedimientos Escena_de_carretera_profundidad

Como resumen de los tres procedimientos realizados obtenemos la siguiente tabla, con su correspondiente gráfico:

Secuencia	Ptos. Profundidad	Ptos. de error	Acierto (%)	Velocidad	Movimiento	Luminosidad	Ptos. Análisis
30-6.avi	183	49	73,45	Medio	Bajo - Medio	Medio	Alto
10-Persona1.avi	380	104	71,86	Bajo	Bajo	Medio	Alto
50-2.avi	108	47	55,54	Alto	Bajo - Medio	Alto	Medio

Tabla VI-8 Resumen procedimientos Escena_de_carretera_velocidad.

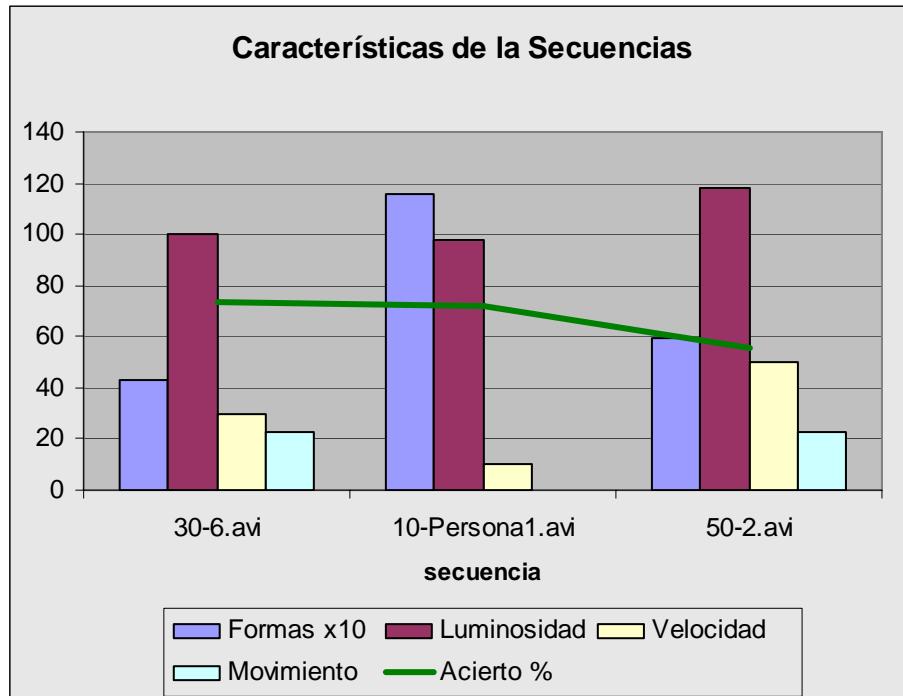


Figura VI-17 Características y porcentaje de acierto de las secuencias de profundidad.

EVALUACIÓN DE LOS RESULTADOS OBTENIDOS

Los mejores resultados se obtienen a unas condiciones de velocidad bajo-medio y luminosidad media, sin demasiados cambios de movimiento.

Los peores resultados se obtienen bajo una luminosidad alta y una característica de movimiento alto.

No se aprecia ninguna influencia del número de puntos de análisis en el cálculo de la profundidad.

4 Pruebas de Etiquetado con Secuencias Cortas

4.1 Primera Escena_de_carretera_etiquetado

Se ha seleccionado la secuencia de imágenes extraídas del video de pruebas *20-1.avi*. Se realiza el seguimiento de 1 señal a lo largo de 14 frames. La secuencia dura 2,8 segundos a una velocidad de 20 Km./h, con unas características de luminosidad medio, formas con valor alto y movimiento bajo.

La secuencia transcurre a 20 Km./h por una carretera de doble sentido. El desplazamiento se realiza en línea recta sobre una ligera pendiente negativa. Existe vegetación abundante a la izquierda de la carretera.

La siguiente tabla de datos expresa los resultados obtenidos para cada frame:

Objeto	Frame	Etiqueta correcta	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
Objeto 1	100	SI	20	0	100	1206
Objeto 1	101	SI	20	0	101	1223
Objeto 1	102	SI	20	0	104	1284
Objeto 1	103	SI	20	0	106	1329
Objeto 1	104	SI	20	0	113	1323
Objeto 1	105	SI	20	0	118	1336
Objeto 1	106	SI	20	0	113	1190
Objeto 1	107	SI	20	0	101	1347
Objeto 1	108	SI	20	0	90	1289
Objeto 1	109	SI	20	0	83	1392
Objeto 1	110	SI	20	0	78	1296
Objeto 1	111	SI	20	0	80	1402
Objeto 1	112	SI	20	0	80	1260
Objeto 1	113	SI	20	0	81	1353
Total	14	100%			96	1302

Tabla VI-9 Resultados 1ª Escena_de_carretera_etiquetado.

De la tabla VI 9 se deduce que el seguimiento y etiquetado a 20 Km./h. se realizan con un acierto del 100%, bajo condiciones de luminosidad media y formas de valor alto, describiendo un movimiento bajo.

4.2 Segunda Escena_de_carretera_etiquetado

EVALUACIÓN DE LOS RESULTADOS OBTENIDOS

Se ha seleccionado la secuencia de imágenes extraídas del video de pruebas *60-1.avi*. Se realiza el seguimiento de 7 señales a lo largo de 60 frames. La secuencia dura 12 segundos a una velocidad de 60 Km./h, con unas características de luminosidad bajo, formas con valor bajo y movimiento variable.

La secuencia transcurre a 60 Km./h por una vía de servicio de sentido único. El desplazamiento se realiza la mayor parte del tiempo en línea recta, y presenta en algunos tramos curvas muy ligeras y pendientes suaves. Ausencia total de vegetación a ambos lados de la carretera, se observa una serie de farolas altas a la derecha.

Los resultados obtenidos se expresan mediante 7 tablas, una para cada objeto:

Objeto	Frame	Etiqueta correcta	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
Objeto 1	6	SI	60	0	68	326
Objeto 1	7	SI	60	0	68	314
Total	2	100%			68	320

Tabla VI-10 Resultados 1ª señal, 2ª Escena_de_carretera_etiquetado.

Objeto	Frame	Etiqueta correcta	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
Objeto 2	6	SI	60	0	68	326
Objeto 2	7	SI	60	0	68	314
Objeto 2	8	SI	60	0	68	273
Objeto 2	9	SI	60	0	67	275
Objeto 2	10	SI	60	0	66	277
Objeto 2	11	SI	60	0	68	252
Total	6	100%			68	286

Tabla VI-11 Resultados 2ª señal, 2ª Escena_de_carretera_etiquetado.

Objeto	Frame	Etiqueta correcta	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
Objeto 3	6	SI	60	0	68	326
Objeto 3	7	SI	60	0	68	314
Objeto 3	8	SI	60	0	68	273
Objeto 3	9	SI	60	0	67	275
Total	4	100%			68	297

Tabla VI-12 Resultados 3ª señal, 2ª Escena_de_carretera_etiquetado.

Objeto	Frame	Etiqueta correcta	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis

Objeto 4	10	SI	60	0	66	277
Objeto 4	11	SI	60	0	68	252
Objeto 4	12	SI	60	0	69	257
Objeto 4	13	SI	60	0	70	247
Objeto 4	14	SI	60	0	70	235
Objeto 4	15	SI	60	0	70	219
Objeto 4	16	SI	60	0	70	216
Objeto 4	17	SI	60	0	70	178
Objeto 4	18	SI	60	0	70	241
Objeto 4	19	SI	60	0	70	241
Objeto 4	20	SI	60	0	71	252
Objeto 4	21	SI	60	0	71	214
Objeto 4	22	SI	60	0	71	241
Objeto 4	23	SI	60	0	71	220
Objeto 4	24	SI	60	0	72	232
Objeto 4	25	SI	60	0	72	227
Objeto 4	26	SI	60	0	72	243
Objeto 4	27	SI	60	0	72	230
Objeto 4	28	SI	60	0	72	256
Objeto 4	29	SI	60	0	72	241
Total	20	100%			70	236

Tabla VI-13 Resultados 4^a señal, 2^a Escena_de_carretera_etiquetado.

Objeto	Frame	Etiqueta correcta	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
Objeto 5	37	SI	60	50	80	213
Objeto 5	38	SI	60	50	80	238
Objeto 5	39	SI	60	50	80	216
Objeto 5	40	SI	60	50	79	232
Objeto 5	41	SI	60	50	78	211
Objeto 5	42	SI	60	50	77	248
Objeto 5	43	SI	60	50	78	253
Objeto 5	44	SI	60	50	78	262
Objeto 5	45	SI	60	50	78	257
Objeto 5	46	SI	60	50	78	277
Objeto 5	47	SI	60	50	77	246
Objeto 5	48	SI	60	50	76	249
Total	12	100%			78	242

Tabla VI-14 Resultados 5^a señal, 2^a Escena_de_carretera_etiquetado.

Objeto	Frame	Etiqueta correcta	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
Objeto 6	47	SI	60	50	77	246
Objeto 6	48	SI	60	50	76	249

Objeto 6	49	SI	60	50	76	229
Objeto 6	50	SI	60	50	76	228
Total	4	100%			76	238

Tabla VI-15 Resultados 6ª señal, 2ª Escena_de_carretera_etiquetado.

Objeto	Frame	Etiqueta correcta	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
Objeto 7	49	SI	60	50	76	229
Objeto 7	50	SI	60	50	76	228
Objeto 7	51	SI	60	50	76	233
Objeto 7	52	SI	60	50	76	212
Objeto 7	53	SI	60	50	77	232
Objeto 7	54	SI	60	50	76	210
Objeto 7	55	SI	60	100	76	221
Objeto 7	56	NO	60	100	76	211
Objeto 7	57	NO	60	100	75	201
Objeto 7	58	NO	60	100	76	214
Objeto 7	59	NO	60	100	79	249
Objeto 7	60	NO	60	100	80	223
Total	12	58,33%			77	222

Tabla VI-16 Resultados 7ª señal, 2ª Escena_de_carretera_etiquetado.

De las tablas previas se deduce que el seguimiento y etiquetado a 60 Km./h. se realizan con un acierto del 90,56%, bajo condiciones de luminosidad baja y formas de valor bajo, describiendo un movimiento variable.

Los peores resultados se obtienen cuando se aprecia un movimiento alto, donde se han producido 5 errores en tan solo 6 frames.

4.3 Tercera Escena_de_carretera_etiquetado

Se ha seleccionado la secuencia de imágenes extraídas del video de pruebas *30-7.avi*. Se realiza el seguimiento de 2 señales a lo largo de 25 frames. La secuencia dura 5 segundos a una velocidad de 30 Km./h, con unas características de luminosidad alta, formas con valor medio y movimiento alto.

La secuencia transcurre a 30 Km./h por una rotonda. El desplazamiento se realiza en curva con ausencia de pendientes. No presente de vegetación ni demasiadas formas.

Los resultados obtenidos se expresan mediante 2 tablas, una para cada objeto:

Objeto	Frame	Etiqueta correcta	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
Objeto 1	5	SI	30	100	127	527
Objeto 1	6	SI	30	100	128	561
Objeto 1	7	SI	30	100	128	638
Objeto 1	8	SI	30	100	128	544
Objeto 1	9	SI	30	100	129	734
Objeto 1	10	SI	30	100	129	695
Objeto 1	11	SI	30	100	129	605
Objeto 1	12	SI	30	100	130	528
Total	8	100%			128	604

Tabla VI-17 Resultados 1ª señal, 3ª Escena_de_carretera_etiquetado.

Objeto	Frame	Etiqueta correcta	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
Objeto 2	39	SI	30	100	120	421
Objeto 2	40	SI	30	100	123	392
Objeto 2	41	NO	30	100	123	387
Objeto 2	42	NO	30	100	125	402
Objeto 2	43	NO	30	100	125	404
Objeto 2	44	NO	30	100	125	417
Objeto 2	45	NO	30	100	126	383
Objeto 2	46	SI	30	100	126	385
Objeto 2	47	NO	30	100	125	410
Objeto 2	48	SI	30	100	125	407
Objeto 2	49	SI	30	100	125	441
Objeto 2	50	SI	30	100	125	410
Objeto 2	51	SI	30	100	123	442
Objeto 2	52	SI	30	100	123	380
Objeto 2	53	SI	30	100	123	430
Objeto 2	54	SI	30	100	124	403
Total	16	62,5%			124	407

Tabla VI-18 Resultados 2ª señal, 3ª Escena_de_carretera_etiquetado.

De las tablas previas se deduce que el seguimiento y etiquetado a 30 Km./h. y movimiento alto se realizan con un acierto del 75 %, bajo condiciones de luminosidad alta y formas de valor medio, describiendo un movimiento alto.

4.4 Evaluación de los tres procedimientos Escena_de_carretera_etiquetado

Como resumen de los tres procedimientos realizados, se extrae la conclusión de que la característica de movimiento es la que influye de manera notable sobre el etiquetado. En los frames en los que el movimiento ha sido alto se han registrado un número elevado de errores.

Veamos la siguiente tabla se muestra la influencia del movimiento en el etiquetado de objetos:

Movimiento	Nº de frames	Acierto (%)
Bajo	39	100
Medio	22	100
Alto	21	64,5

Tabla VI-19 Influencia del movimiento en el seguimiento y etiquetado.

5 Primera Combinación_de_escena_de_carretera

Se ha seleccionado la secuencia de imágenes extraídas del video de pruebas *bosque.avi*. Se analizan 50 frames, comenzando en el 20 con un incremento de 20, la secuencia dura 3 minutos y 20 segundos, a una velocidad de entre 0 y 50 Km./h, con unas características de luminosidad baja, formas con valor medio y movimiento variable.

La secuencia transcurre por una carretera de doble sentido. A lo largo de la secuencia se realiza un giro hacia la derecha y una rotonda en la que se toma la tercera salida. Existe vegetación alta a ambos lados de la carretera, y sobre la mediana una serie de farolas de alumbrado muy altas.

5.1 Evaluación de velocidad en primera secuencia global

La siguiente tabla de datos expresa los resultados obtenidos de velocidad para cada frame analizado:

Frame	Vel. Estimada	Vel. Referencia	Acierto (%)	Movimiento	Luminosidad	Ptos. de Análisis
20	0	5	0	50	50	364
40	4,2	5	84	0	48	317
60	16,5	15	90	100	45	268
80	20,47	20	97,65	50	59	573
100	30,77	30	97,43	50	97	414
120	27,93	30	93,1	50	93	561
140	31,71	30	94,3	50	79	878
160	36,94	20	15,3	100	92	456
180	24,8	25	99,2	50	85	285
200	28,62	28	97,78	0	77	425
220	24,8	30	82,66	0	78	350
240	24,15	30	80,5	0	58	345
260	16,37	15	90,86	100	71	460
280	18,75	15	75	0	84	466
300	25,11	20	74,45	0	82	373
320	23,13	20	84,35	0	69	422
340	22,35	20	88,25	0	75	344
360	23,02	20	84,9	0	73	282
380	20,15	20	99,25	0	49	334
400	20,15	30	67,16	50	47	570
420	27,64	30	92,13	50	56	354
440	20,12	30	67,06	0	58	244

460	27,07	25	91,72	100	154	258
480	28,56	30	95,2	50	70	344
500	16,36	30	54,53	50	73	329
520	14,57	40	36,425	50	72	386
540	21,84	40	54,6	50	77	246
560	27,01	40	67,52	50	84	294
580	28,51	35	81,45	0	76	252
600	35,6	35	98,28	0	80	251
620	25,8	20	71	100	125	169
640	30,65	25	77,4	100	63	259
660	21,65	25	86,6	100	69	357
680	26,02	30	86,73	0	64	356
700	27,51	35	78,6	0	45	203
720	28,2	40	70,5	50	47	253
740	32,99	35	94,25	50	47	264
760	32,36	40	80,9	0	38	208
780	28,99	40	72,47	0	38	282
800	28,99	40	72,47	50	41	337
820	32,1	40	80,25	50	45	227
840	26,01	40	65,02	0	77	328
860	31,67	40	79,17	0	37	261
880	24,71	40	61,77	0	43	316
900	25,87	35	73,91	0	43	345
920	25,22	30	84,06	0	73	499
940	38,46	30	71,8	100	79	302
960	33	35	94,28	50	82	499
980	26,35	50	52,7	0	80	413
1000	37,93	50	75,86	0	92	245
	25,43	29,66	77,29	34	69	351

Tabla VI-20 Resultados velocidad 1^a Combinación_de_escena_de_carretera.

El resultado de velocidad fue del 77,29% respecto de la velocidad que marcaba el cuentakilómetros sobre 50 frames analizados a lo largo de toda la secuencia.

De la tabla VI 20 se extraen los dos gráficos siguientes, que permite comparar los resultados de velocidad obtenidos a lo largo de toda la secuencia, con la evolución de las características de la secuencia:

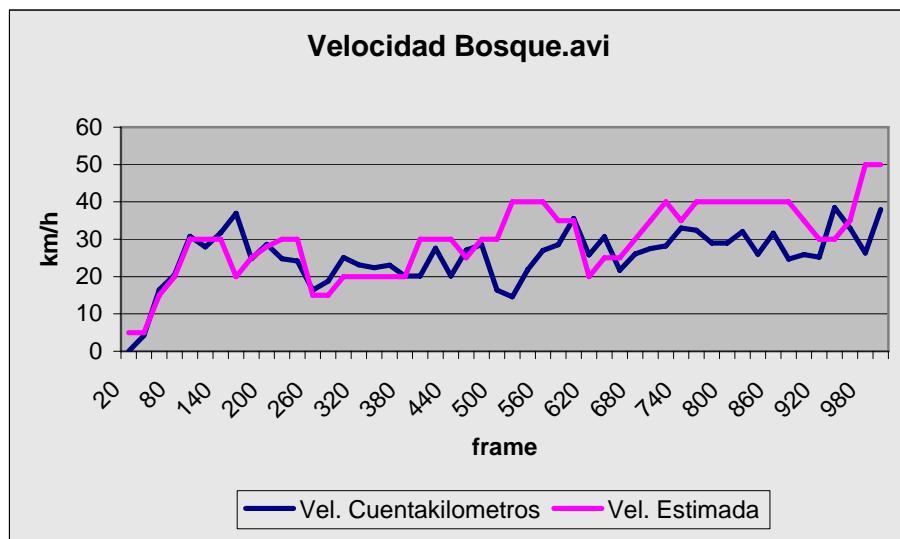


Figura VI-18 Comparativa velocidad secuencia bosque.avi.

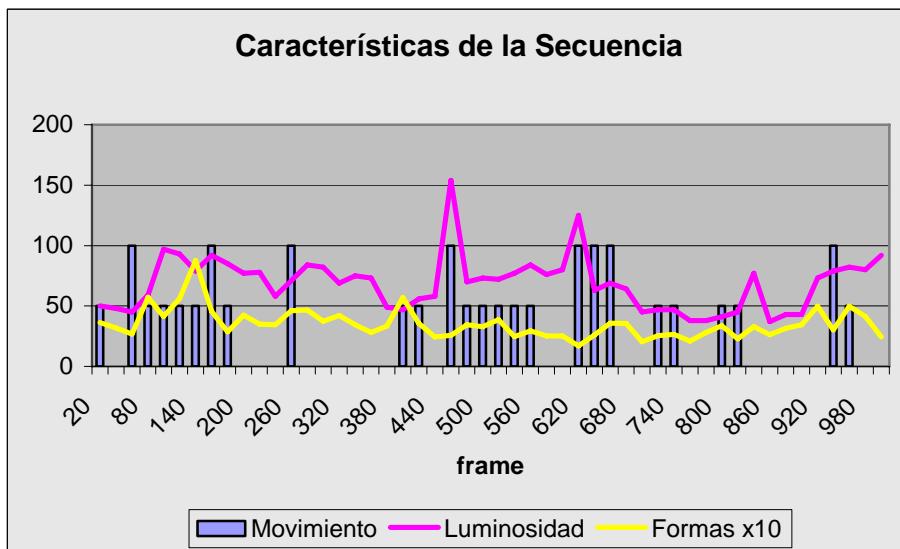


Figura VI-19 Características de la secuencia bosque.avi en los frames analizados.

De estos gráficos se deduce que las estimaciones de velocidad en global se realizan con un notable porcentaje de acierto (77.29%), bajo condiciones bajas de luminosidad, presencia media de formas en la secuencia y siguiendo un movimiento bajo-medio. En los frames 500, 520, 540 o 980 existe un menor acierto, esto puede explicarse porque existen bajas formas, movimiento medio y alta luminosidad.

5.2 Evaluación de profundidad en primera secuencia global

La siguiente tabla de datos expresa los resultados obtenidos de profundidad para cada frame:

Secuencia	Ptos. Profundidad	Ptos. de error	Acierto (%)	Velocidad	Movimiento	Luminosidad	Ptos. Análisis
20	178	94	47,19	5	50	50	364
60	173	53	69,36	15	100	45	268
100	128	81	36,71	30	0	97	414
140	419	146	65,15	30	0	79	878
180	153	82	46,40	25	50	85	285
220	209	29	86,12	30	0	78	350
260	122	41	66,39	15	100	71	460
300	213	74	65,25	20	0	82	373
340	232	98	57,75	20	0	75	344
380	200	42	79	20	0	49	334
420	142	87	38,73	30	50	56	354
460	70	43	38,57	25	100	154	258
500	210	64	69,52	30	50	73	329
540	101	21	79,20	40	50	77	246
580	116	35	69,82	35	0	76	252
620	46	26	43,47	20	100	125	169
660	196	95	51,53	25	100	69	357
700	98	13	86,73	35	0	45	203
740	143	45	68,53	35	50	47	264
780	166	37	77,71	40	0	38	282
820	102	47	53,92	40	50	45	227
860	109	28	74,31	40	0	37	261
900	106	35	66,98	35	0	43	345
940	161	77	52,17	30	100	79	302
980	120	22	81,66	50	0	80	413
Total	156	57	62,89	28,8		70	333

Tabla VI-21 Resultados profundidad 1ª Combinación_de_escena_de_carretera.

De la tabla VI 21 se extraen los dos gráficos siguientes, que permite comparar los resultados de profundidad obtenidos a lo largo de toda la secuencia, con la evolución de las características de la secuencia:

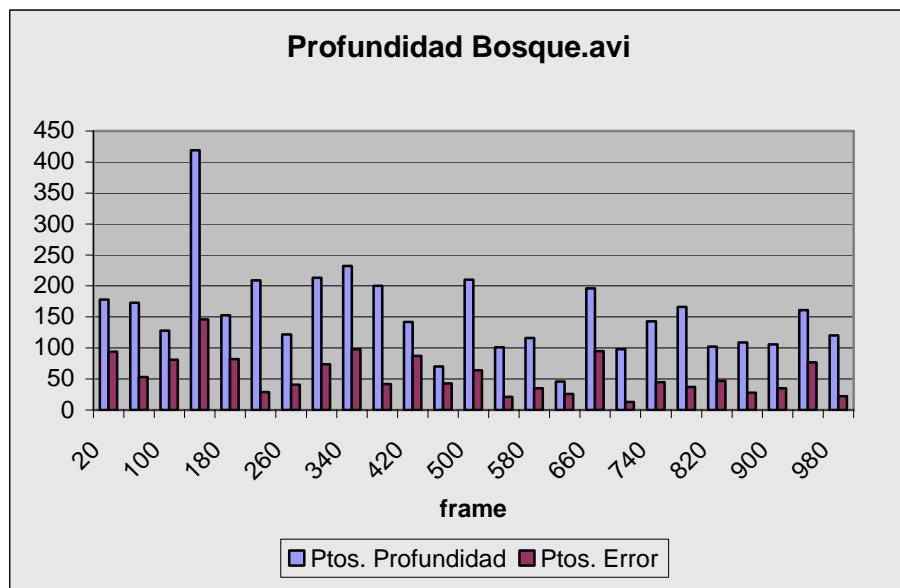


Figura VI-20 Comparativa profundidad secuencia bosque.avi.

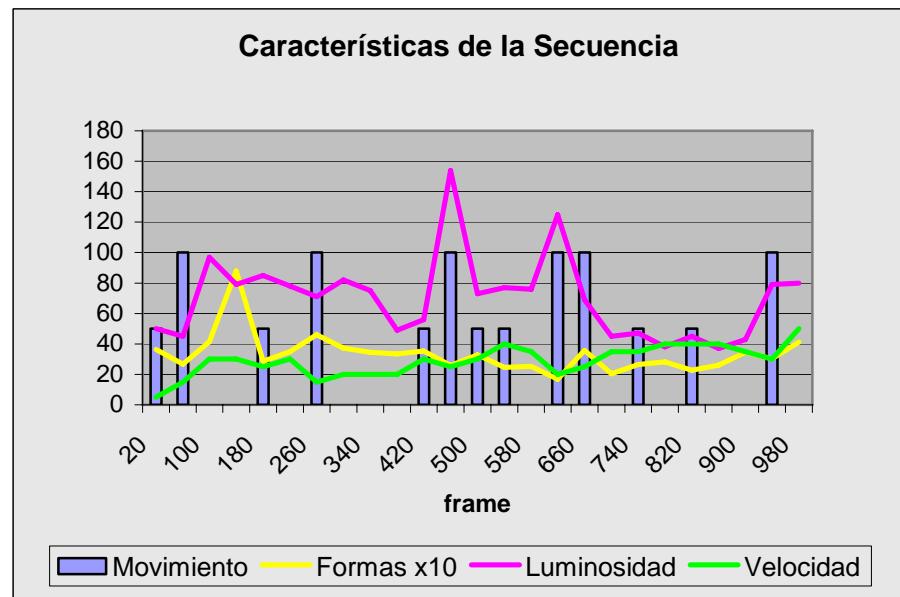


Figura VI-21 Características de la secuencia bosque.avi en los frames analizados.

De estos gráficos se deduce que las estimaciones de profundidad se realizan con un buen porcentaje de acierto (62.89%), bajo condiciones bajas de luminosidad y formas de valor bajo, describiendo un movimiento muy variable. Se observa en los frames 420 y 460 se alcanza el mínimo acierto debido a movimientos muy altos y escasa existencia de formas.

5.3 Evaluación del etiquetado en primera secuencia global

En la secuencia se realiza un seguimiento a 2 señales en un total de 21 frames. Las siguientes tablas de datos expresan los resultados obtenidos de etiquetado en la secuencia:

Objeto	Frame	Etiqueta correcta	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
Objeto 1	422	SI	30	50	54	273
Objeto 1	423	SI	30	50	55	364
Objeto 1	424	SI	30	50	58	312
Objeto 1	425	SI	30	50	63	391
Objeto 1	426	SI	30	50	62	338
Objeto 1	427	SI	30	50	61	281
Objeto 1	428	SI	30	50	59	250
Objeto 1	429	SI	30	50	59	277
Objeto 1	430	SI	30	50	58	249
Objeto 1	431	SI	30	50	54	283
Objeto 1	432	SI	30	50	56	235
Objeto 1	433	SI	30	50	56	281
Objeto 1	434	SI	30	50	55	267
Total	13	100%			57	292

Tabla VI-22 Resultados 1º señal, 1ª Combinación_de_escena_de_carretera.

Objeto	Frame	Etiqueta correcta	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
Objeto 2	914	SI	35	0	65	413
Objeto 2	915	SI	35	0	54	437
Objeto 2	916	SI	35	0	63	458
Objeto 2	917	SI	35	0	64	426
Objeto 2	918	NO	35	0	65	431
Objeto 2	919	NO	35	0	68	499
Objeto 2	920	SI	30	0	73	499
Objeto 2	921	SI	30	0	73	603
Total	8	75%			66	471

Tabla VI-23 Resultados 2º señal, 1ª Combinación_de_escena_de_carretera.

6 Segunda Combinación_de_escena_de_carretera

Se ha seleccionado la secuencia de imágenes extraídas del video de pruebas *zarzal.avi*. Se analizan 29 frames, comenzando en el 20 con un incremento de 20, la secuencia dura 1 minuto y 56 segundos, a una velocidad de entre 5 y 40 Km./h, con unas características de luminosidad alta, formas con valor medio y movimiento variable.

La secuencia transcurre alrededor de una serie de calles que rodean una urbanización. A lo largo de la secuencia se realizan giros hacia la izquierda a poca velocidad. Existe una vegetación baja, y muchos bloques de casas que proporcionan sobras a la secuencia.

6.1 Evaluación de velocidad en segunda secuencia global

La siguiente tabla de datos expresa los resultados obtenidos de velocidad para cada frame:

Frame	Vel. Estimada	Vel. Referencia	Acierto (%)	Movimiento	Luminosidad	Ptos. de Análisis
20	21,23	20	93,85	50	128	585
40	24,82	20	75,9	0	133	576
60	24,5	20	77,5	0	130	710
80	25,91	20	70,45	0	120	870
100	17,57	15	82,86	0	107	1006
120	18,18	5	0	100	95	1273
140	15,55	15	96,33	50	111	1408
160	32,34	20	38,3	50	125	526
180	22,8	20	86	100	123	627
200	31,48	20	42,6	100	133	489
220	26,91	30	89,7	50	113	568
240	37,63	40	94,07	0	112	326
260	17,36	40	43,4	50	107	323
280	40,02	40	99,95	50	115	293
300	20,62	35	58,91	0	101	535
320	31,07	40	77,67	0	94	466
340	34,26	35	97,88	0	106	401
360	18,44	15	77,06	100	105	425
380	18,79	15	74,73	50	100	548
400	26,78	20	66,1	100	108	364
420	16,74	15	88,4	100	114	209
440	14,13	15	94,2	50	137	287
460	20,18	20	99,1	0	128	346
480	17,36	20	86,8	0	128	297

500	24,62	25	98,48	0	127	380
520	18,25	20	91,25	0	130	295
540	17,98	15	80,13	50	136	435
560	18,53	20	92,65	100	132	757
580	12,6	10	74	0	115	506
	22,98	22,24	77,52		118	546

Tabla VI-24 Resultados velocidad 2^a Combinación_de_escena_de_carretera.

El resultado de velocidad fue del 77,52% respecto de la velocidad que marcaba el cuentakilómetros sobre 29 frames analizados a lo largo de toda la secuencia.

De la tabla VI 24 se extraen los dos gráficos siguientes, que permite comparar los resultados de velocidad obtenidos a lo largo de toda la secuencia, con la evolución de las características de la secuencia:

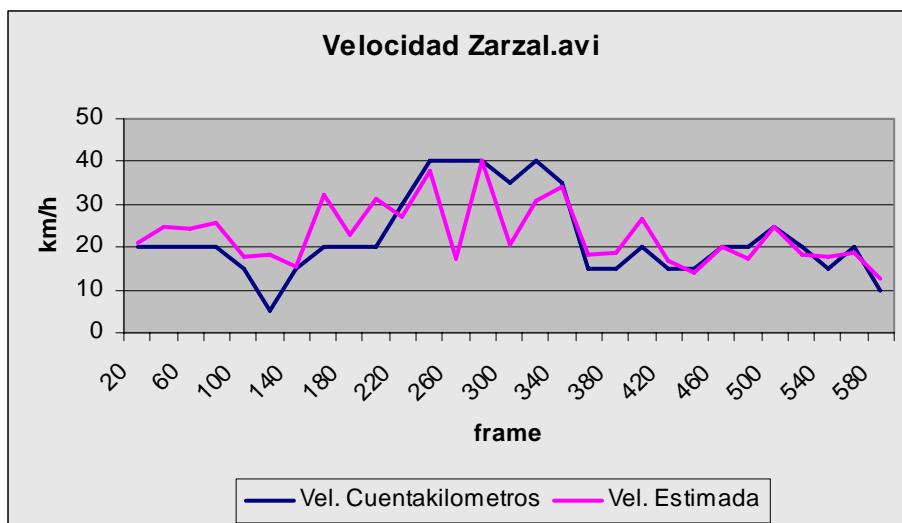


Figura VI-22 Comparativa velocidad secuencia zarzal.avi.

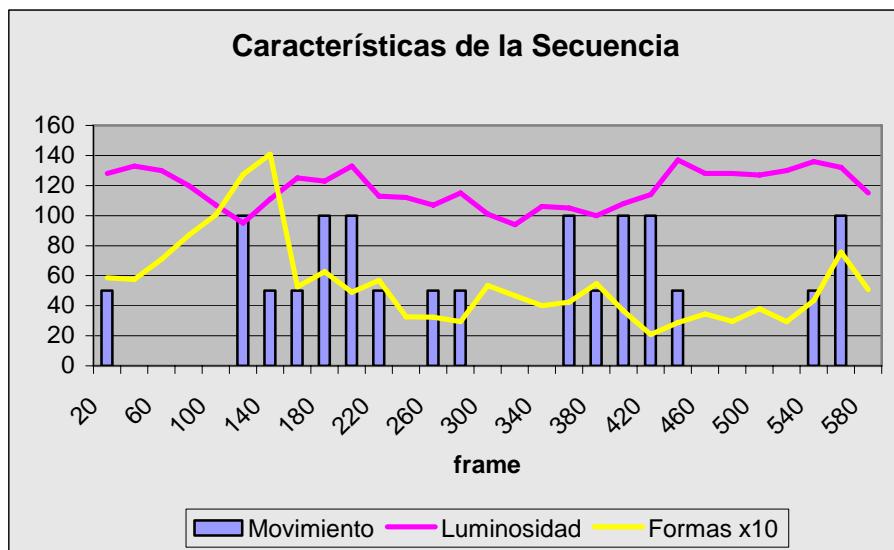


Figura VI-23 Características de la secuencia zarzal.avi en los frames analizados.

De estos gráficos se deduce que las estimaciones de velocidad variable se realizan con un notable porcentaje de acierto (77,52%), bajo condiciones altas de luminosidad, presencia de formas media en la secuencia y siguiendo un movimiento bajo-medio. En el frame 260 se alcanza un acierto muy bajo debido a la escasa existencia de formas.

6.2 Evaluación de profundidad en segunda secuencia global

La siguiente tabla de datos expresa los resultados obtenidos de profundidad para cada frame:

Secuencia	Ptos. Profundidad	Ptos. de error	Acierto (%)	Velocidad	Movimiento	Luminosidad	Ptos. Análisis
20	590	126	78,64	20	50	128	585
60	526	184	65,01	20	0	130	710
100	599	222	62,93	15	0	107	1006
140	44	15	65,90	15	50	111	1408
180	186	74	60,21	20	100	123	627
220	161	63	60,86	30	50	113	568
260	120	57	52,5	40	50	107	323
300	132	51	61,36	35	0	101	535
340	207	37	82,12	35	0	106	401
380	211	92	56,39	15	50	100	548
420	0	0	0	15	100	114	209

460	96	55	42,70	20	0	128	346
500	123	24	80,48	25	0	127	380
540	235	38	83,82	15	50	136	435
580	78	57	26,92	10	0	115	506
221	73	58,66		22		116	572

Tabla VI-25 Resultados profundidad 2ª Combinación_de_escena_de_carretera.

De la tabla VI 25 se extraen los dos gráficos siguientes, que permite comparar los resultados de profundidad obtenidos a lo largo de toda la secuencia, con la evolución de las características de la secuencia:

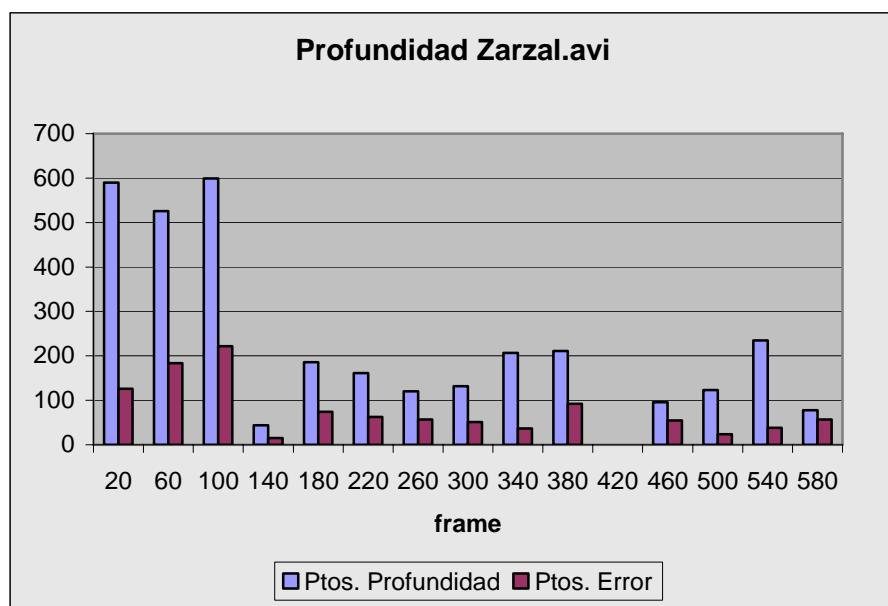


Figura VI-24 Comparativa profundidad secuencia zarzal.avi.

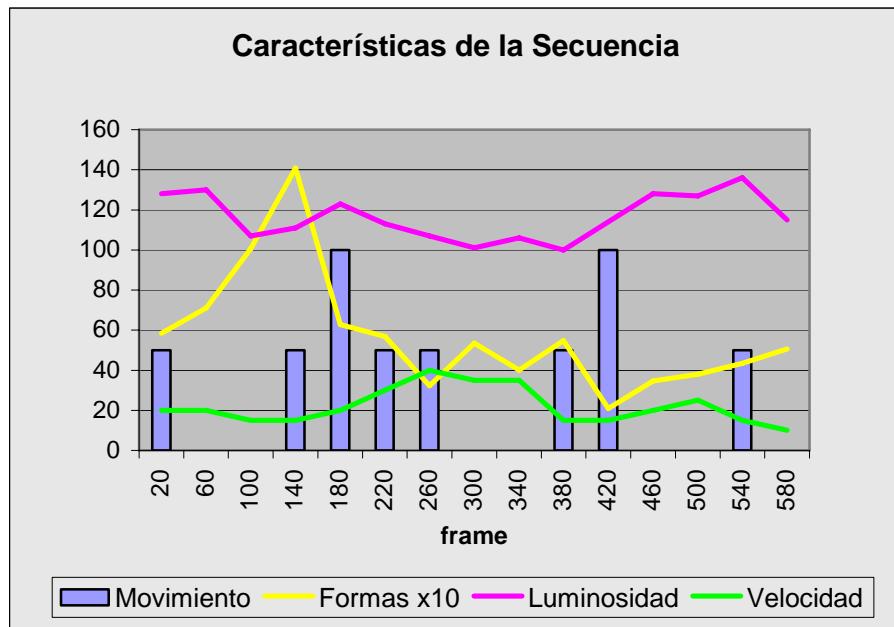


Figura VI-25 Características de la secuencia zarzal.avi en los frames analizados.

De estos gráficos se deduce que las estimaciones de profundidad se realizan con un buen porcentaje de acierto (58.66%), bajo condiciones altas de luminosidad y formas de valor medio, describiendo un movimiento variable. En los frames 380, 420 y 460 se observa un bajo acierto debido a la escasa existencia de formas y alto movimiento.

6.3 Evaluación del etiquetado en segunda secuencia global

En la secuencia se realiza un seguimiento a 3 señales en un total de 34 frames. Las siguientes tablas de datos expresan los resultados obtenidos de etiquetado en la secuencia:

Objeto	Frame	Etiqueta correcta	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
Objeto 1	79	SI	20	0	129	1037
Objeto 1	80	SI	20	0	120	950
Objeto 1	81	SI	20	0	120	1006
Objeto 1	82	SI	20	0	118	1030
Objeto 1	83	SI	20	0	116	1122
Objeto 1	84	SI	20	0	115	1084
Objeto 1	85	SI	20	0	114	1115
Objeto 1	86	SI	20	0	113	1268
Objeto 1	87	SI	20	0	114	1254

EVALUACIÓN DE LOS RESULTADOS OBTENIDOS

Objeto 1	88	NO	20	0	114	1324
Objeto 1	89	NO	20	0	114	1411
Total	11	81,81%			117	114

Tabla VI-26 Resultados 1ª señal, 2ª Combinación_de_escena_de_carretera.

Objeto	Frame	Etiqueta correcta	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
Objeto 2	329	SI	40	0	97	361
Objeto 2	330	SI	40	0	101	367
Objeto 2	331	SI	40	0	102	228
Objeto 2	332	SI	40	0	101	236
Objeto 2	333	SI	40	0	101	248
Objeto 2	334	SI	40	0	100	262
Objeto 2	335	SI	40	0	98	272
Total	7	100%			100	282

Tabla VI-27 Resultados 2ª señal, 2ª Combinación_de_escena_de_carretera.

Objeto	Frame	Etiqueta correcta	Velocidad	Movimiento	Luminosidad	Ptos. de Análisis
Objeto 3	380	SI	15	50	100	314
Objeto 3	381	NO	15	50	99	364
Objeto 3	382	SI	15	50	102	354
Objeto 3	383	SI	15	50	111	436
Objeto 3	384	SI	15	50	123	478
Objeto 3	385	SI	15	50	122	472
Objeto 3	386	SI	15	50	118	384
Objeto 3	387	SI	15	50	106	425
Objeto 3	388	SI	15	50	100	315
Objeto 3	389	SI	15	50	101	276
Objeto 3	390	SI	15	50	100	194
Objeto 3	391	SI	15	50	100	283
Objeto 3	392	SI	15	50	101	207
Objeto 3	393	SI	15	50	104	235
Objeto 3	394	SI	15	50	106	218
Objeto 3	395	SI	15	50	106	290
Total	16	93,75%			106	328

Tabla VI-28 Resultados 3ª señal, 2ª Combinación_de_escena_de_carretera

7 Informe Global de Resultados Obtenidos

Los resultados de todas las pruebas de sistema realizadas son la base para generar el siguiente informe. Se pretende estimar la fiabilidad de los resultados obtenidos, dividiendo en tres categorías: velocidad, profundidad y etiquetado de objetos.

Los resultados obtenidos con respecto al objetivo de estimación de velocidad se pueden clasificar en 5 rangos de velocidades. La siguiente figura expresa los porcentajes de acierto para cada rango:

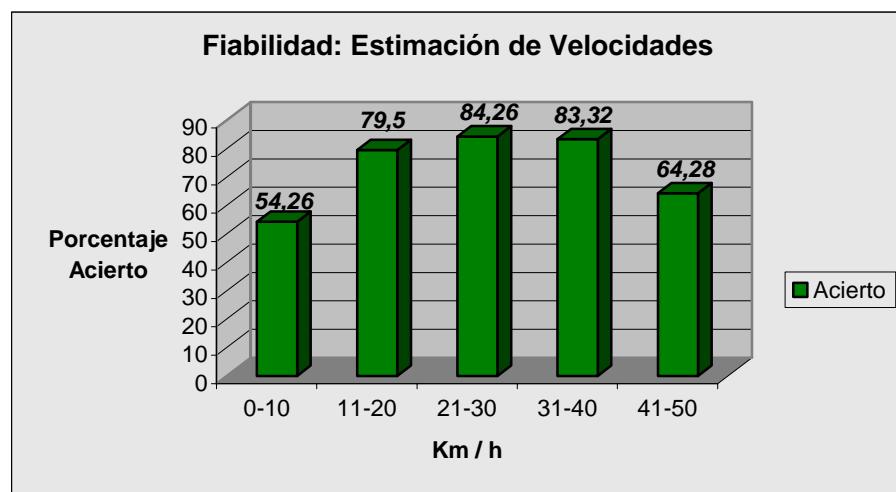


Figura VI-26 Porcentaje de acierto de velocidad dividido en rangos.

Las pruebas de profundidad revelan, que las características de la escena que más influyen en los resultados son la luminosidad y la velocidad. Veamos una figura que muestra la influencia de cada una de ellas en los porcentajes de acierto de la profundidad:

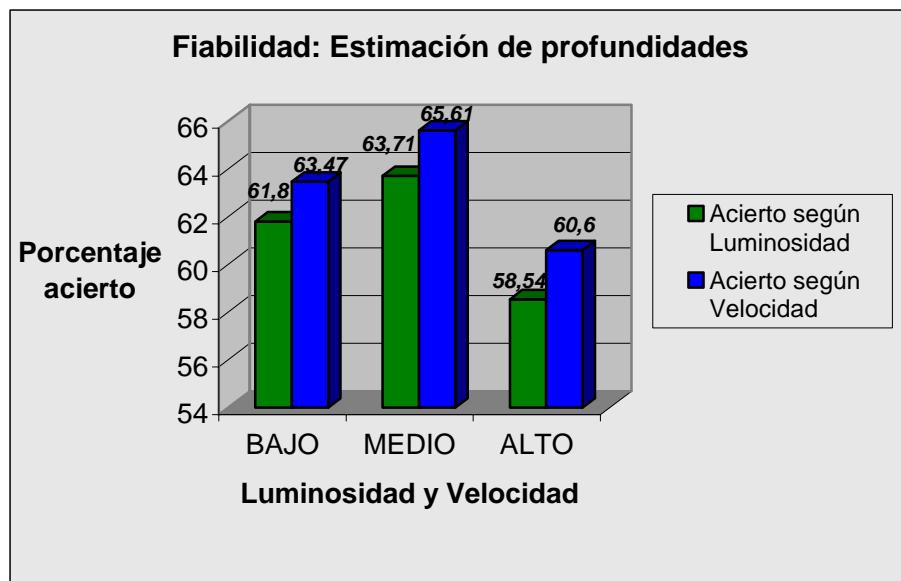


Figura VI-27 Porcentajes de acierto de la profundidad según luminosidad y velocidad.

Como se aprecia los mejores resultados se obtienen para condiciones de luminosidad y velocidad media, y los peores bajo luminosidad y velocidad alta.

Por ultimo los resultados obtenidos en el etiquetado se ven claramente influenciados por las características de movimiento. El siguiente gráfico muestra la influencia del movimiento en el etiquetado:

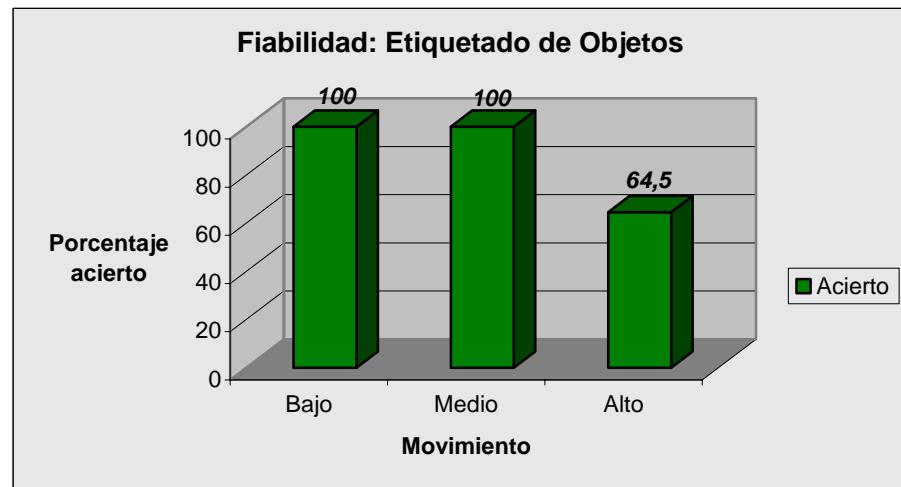


Figura VI-28 Porcentaje de acierto del etiquetado según movimiento.

Se observa que el rendimiento se reduce en un 35.5 % cuando la escena presenta un movimiento alto.

Con el fin de proporcionar un porcentaje de acierto global para cada uno de los objetivos, se ha contabilizado los resultados parciales, y posteriormente se ha obtenido un porcentaje global.

La siguiente tabla proporciona los resultados de fiabilidad para cada objetivo:

	Porcentaje de acierto
<i>Velocidad</i>	77,20 %
<i>Profundidad</i>	63,85 %
<i>Etiquetado</i>	89,54 %

Tabla VI-29 Fiabilidad de resultados proporcionados por el sistema AMIVSI.

Capítulo VII. Conclusiones y Líneas Futuras

La elaboración del presente proyecto ha supuesto para nosotros un gran reto, en el que hemos encontrado numerosas dificultades que han sido resueltas gracias al esfuerzo continuo, una dedicación diaria y la compenetración del equipo de trabajo.

Después de más de siete meses de trabajo podemos afirmar que hemos adquirido un gran enriquecimiento personal, fundamentalmente en dos aspectos: comprender la dinámica de un proyecto de investigación y aprender en el campo de la visión artificial, desconocido hasta entonces para nosotros.

La mayoría de las investigaciones del movimiento en secuencia de imágenes que actualmente se llevan a cabo, se realizan sobre robots que se desplazan en interiores, y que reciben coloquialmente el nombre de robots pasilleros, porque cuando salen al exterior no se comportan como es debido. Es por tanto una dificultad añadida tener que realizar forzosamente nuestras investigaciones contando con la multitud de factores que intervienen en el mundo real.

Podemos decir que el Optical Flow es el motor de nuestro proyecto. Hemos elegido esta técnica para el cálculo de movimiento en una secuencia de imágenes ya que es ampliamente aceptada y posee cantidad de aplicaciones en la práctica. Existen muchos estudios sobre esta técnica y nos hemos centrado en el que mejor se adaptaba a nuestro dominio dentro de la automoción.

En el estudio del movimiento intrínseco de un vehículo la estimación de velocidad ha supuesto una gran complicación. La escasez de información encontrada nos ha motivado a desarrollar una técnica específica, capaz de extraer el espacio recorrido en un intervalo de tiempo basándose en el Optical Flow, y proporcionar resultados satisfactorios. Es innegable que dicha técnica contempla numerosas limitaciones que le impiden estimar una velocidad con la precisión y fiabilidad de la que disponen métodos como el sistema GPS o un tradicional cuentakilómetros.

El segundo objetivo propuesto correspondía al cálculo de profundidad. Debemos decir que, si bien es cierto que teníamos mayores expectativas en los resultados de las pruebas, estamos satisfechos con el trabajo realizado. Hay que tener en cuenta que partíamos de unos sólidos fundamentos teóricos lo que permite dejar una buena base de cara a líneas futuras. Creemos que seguir investigando en este objetivo para aumentar los resultados obtenidos, permitirá beneficios tangibles para la seguridad en el campo automoción.

Una de las mayores satisfacciones llegó con el tercer y último objetivo. La solución propuesta para el seguimiento y etiquetado de objetos previamente reconocidos ha presentado resultados excelentes. La motivación extra de crear una solución para un problema de un

proyecto externo, que se centra en ayudas a la conducción, nos ha impulsado en la consecución de este objetivo. La técnica apoyada en el Optical Flow y en la continuidad de la trayectoria de un objeto se posiciona como una de las más eficientes para realizar seguimiento de señales de tráfico en una secuencia de imágenes.

Por último y como conclusión general, podemos destacar que siempre tuvimos la convicción de culminar con éxito este proyecto de investigación, del cual podemos asegurar que hemos cumplido con los objetivos perseguidos.

En lo referente a las líneas futuras de investigación es lógico pensar que el campo de la visión artificial estará cada día más presente en la vida cotidiana. Son conocidas las aplicaciones industriales como el control de calidad, vigilancia, sistemas de clasificación, etc. Uno de los ámbitos en los que más se está trabajando actualmente de visión artificial es el campo de la automoción.

Actuar cuando el hombre no actúa y ver cuando el ojo humano no ve, debe ser el futuro inmediato de la visión por computador en dicho campo.

Perfeccionando la estimación de profundidades se podrá detectar con precisión obstáculos y distancias de seguridad con el vehículo que le precede, con el fin de alertar al conductor y prevenir accidentes.

El reconocimiento e interpretación de la señales de tráfico es una de las aplicaciones de visión artificial más interesantes y de mayor beneficio para el usuario. La evolución de estas técnicas debe llegar hasta la integración de sistemas de seguridad, que proporcionen avisos de alerta, y en casos extremos tomen alguna medida para evitar situaciones conflictivas, tanto en vehículos destinados al transporte como al uso personal.

El Optical Flow es la técnica más utilizada en las tareas que involucren movimiento. Su evolución ha de mejorar la precisión de los resultados, reducir el tiempo computacional y proporcionar herramientas que faciliten la interpretación del movimiento. Con el avance tecnológico se puede lograr un aumento de información de movimiento, será posible analizar como se han desplazado un mayor número de puntos, consiguiendo así una representación más completa de la realidad del movimiento.

APENDICE I: Bibliografía

- [BRANCH 2001] J. W. Branch, G. Olague. *La Visión por Computador. Una aproximación al estado del arte.* [Documento en línea]. 2001 [Departamento de Ciencias de la Computación, Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California - México]
<<http://www.minas.unalmed.edu.co/facultad/publicaciones/dyna/133/vision.pdf>> [Consulta: 12/01/2005]
- [ESCALERA 2001] Escalera Hueso, Arturo de la. *Visión por computador: fundamentos y métodos.* Madrid: Prentice-Hall, D.L. 2001.
- [GALER 2003] Galer Mark. *Tratamiento digital de imágenes.* Madrid: Anaya, D.L. 2003.
- [GALVIN 1998] B. Galvin, B. McCane, K. Novins, D. Mason & S. Mills. *Recovering Motion Fields: An Evaluation of Eight Optical Flow Algorithm.* [Documento en línea]. 1998 British Machine Vision Conference. [University of Otago, New Zealand].
<www.bmva.ac.uk/bmvc/1998/pdf/p087.pdf> [Consulta: 29/10/2004]
- [GARCIA 1998] J. García de Jalón, J. Ignacio Rodríguez, J.M. Sarriegui, A. Brazález. *Aprenda C++ como si estuviera en primer.* [Documento en línea]. 1998 [Escuela Superior de Ingenieros Industriales de San Sebastián Universidad de Navarra].
<<http://mat21.etsii.upm.es/ayudainf/aprendainf/Cpp/manualcpp.pdf>> [Consulta: 18/10/2004]
- [GONZALEZ 1999] González Jiménez J. *Visión por computador.* Madrid: Paraninfo, D.L. 1999.
- [HIMANSHU 2000] Himanshu Arora, Abhishek Tiwari and Tirthankar Bandhyopadhyay. *Optical Flow based navigation usign an onboard camera.* [Documento en línea]. 2000 [Indian Institute of Technology,

Kampur Department of Computer Science & Engineering]
<<http://www.cse.iitk.ac.in/~amit/courses/768/00/atiwari>>
[Consulta: 14/02/2005]

- [HORN 1980] B. K. P. Horn, B. G. Schunck. *Determining optical flow*. Massachusetts Institute of Technology: Addison Wesley. 1980.
- [JACOBSON 1992] Ivar Jacobson. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison Wesley. 1992.
- [LEINECKER 1999] Leinecker Richard C. *Biblia de Visual C++*. Madrid: Anaya , D.L. 1999.
- [LUCAS 1981] B.D. Lucas, T. Kanade. *An Interactive Image Registration Technique with an Application to Stereo Vision*. DARPA: Image Understanding workshop. 1981.
- [MARAVALL 1993] Maravall, Gomez Allende. *Reconocimiento de formas y visión artificial*. Madrid: RAMA 1993.
- [NALWA 1993] V. S. Nalwa. *A Guided Tour of Computer Vision*. Boston: Addison Wesley.1993.
- [RAE 2005] *Real Academia Española*. [Base de datos en línea]. España 2005
<www.rae.es> [Consulta: 23/12/2004].
- [VAZQUEZ 1996] Fernando Vázquez Núñez. *Segmentación de Imágenes en grafos de contorno. Aplicación a la estimación de la profundidad y el movimiento relativo para un robot móvil autónomo*. [Documento en línea]. 1996
Tesis doctoral. [Universidad de Vigo departamento de ingeniería de sistemas y automática]
<<http://www.aisa.uvigo.es/fvazquez/research.html#PhD>>
[Consulta: 31/01/2005]

BIBLIOGRAFÍA

- [WOODS 1996] González Rafael C., Woods Richard E. *Tratamiento digital de imágenes*. 2^a edición. Madrid: Díaz de Santos, 1996.

APENDICE II: Diario de Experimentos

Este anexo trata de dar una visión general sobre los diferentes métodos utilizados para llevar a cabo tanto las técnicas aplicadas, como pruebas del sistema.

El sistema se ha programado en el entorno Borland C++. Es recomendable utilizar una herramienta de este tipo, para facilitar la programación del interfaz grafico y el manejo de imágenes.

Las librerías MIL poseen bastante documentación asociada, pero para poder entender toda su funcionalidad, se debe aprender a manejar el programa MATROX Inspector, una herramienta visual que permite aplicar técnicas de visión artificial a imágenes.

Hemos utilizado una cámara de video estándar colocada en el interior del vehículo y se han grabado videos de diferentes trayectos. Para lograr una secuencia de imágenes valida para nuestro sistema, se realizaron los siguientes pasos:

Grabar videos de diferentes trayectos: Debe Existir una planificación previa sobre lo que se va a grabar.

Adquisición del video al ordenador: La cámara de video utilizada debe poder conectarse al ordenador. Para adquirir y cortar el video se puede utilizar un programa de edición de video como por ejemplo Adobe Premier.

Captura de imágenes: Una vez tenemos el video cortado, procedemos a la extracción de imágenes. Hay que tener en cuenta, que nuestra aplicación esta pensada para procesar 5 imágenes por segundo y el video normalmente graba 25 imágenes por segundo, por tanto solo capturaremos 1 imagen de cada 5. Además debemos desentrelazar el video, para obtener una mejor calidad de imagen. La aplicación que hemos utilizado para realizar este proceso es el Virtual Dub.

Las imágenes capturadas, deben estar en formato bmp y consecutivas empezando desde la imagen *1.bmp* hasta *n.bmp*

Pruebas del sistema

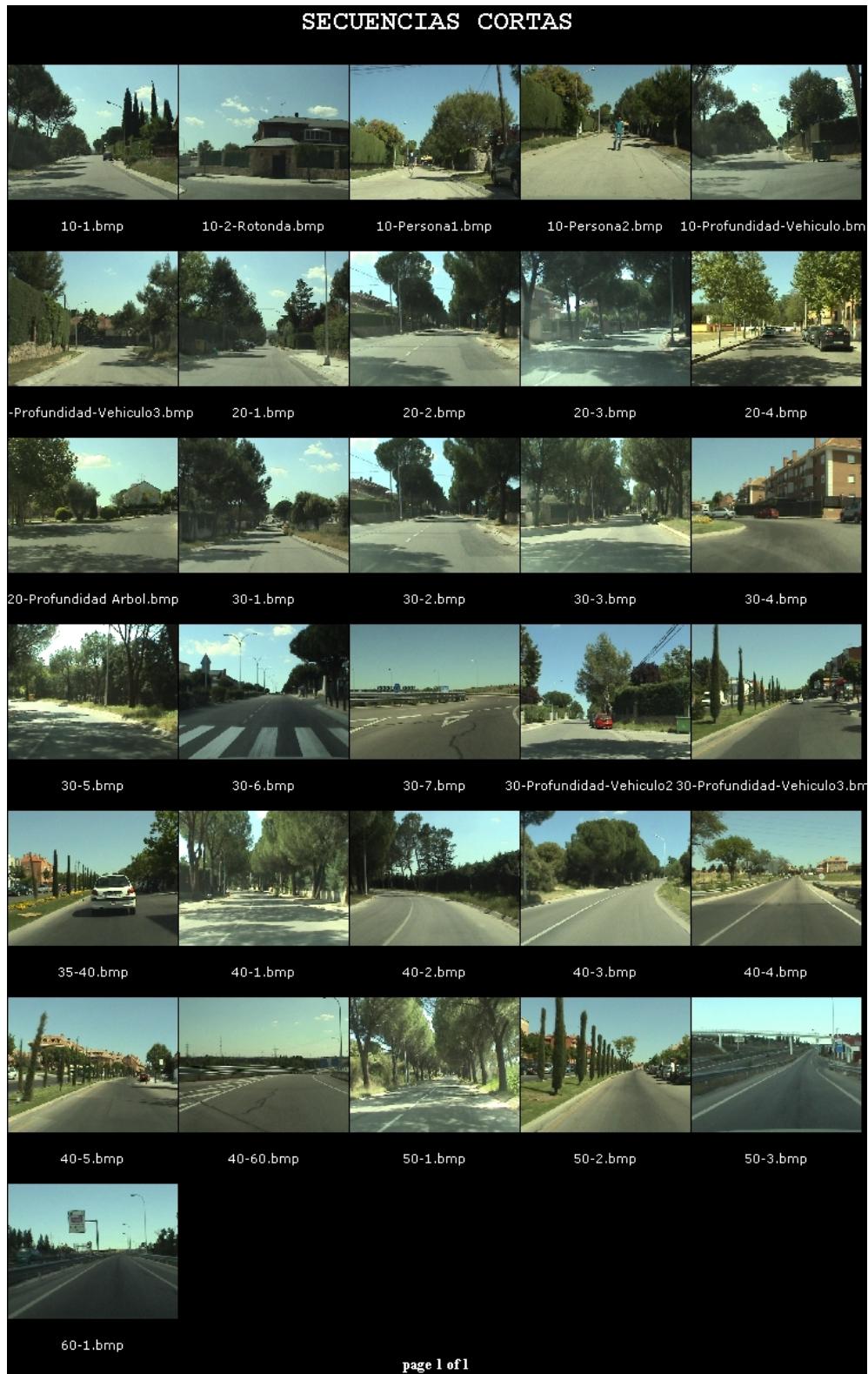
Para realizar las pruebas del sistema se capturaron diferentes secuencias según el plan de verificación. A continuación se muestran índices con imágenes de las secuencias utilizadas en las pruebas:



Copia índice de la Secuencia Elbosque.avi



Copia índice de la Secuencia zarzal.avi



Copia índice de diferentes secuencias concretas.

En la siguiente tabla se muestra un listado de los videos utilizados en las pruebas así como su tamaño y fecha en la que fue capturado:

Video:	Tamaño (bytes):	Fecha de Grabación:
elbosque.avi	2.782.850.668	21/01/2005
zarzal.avi	802.620.924	16/03/2005
10-1.avi	176.133.388	05/05/2005
10-2-Rotonda.avi	76.795.444	05/05/2005
10-Persona1.avi	53.228.784	05/05/2005
10-Persona2.avi	46.516.748	05/05/2005
10-Profundidad-Vehiculo.avi	115.426.852	05/05/2005
15-Profundidad-Vehiculo3.avi	54.720.340	05/05/2005
20-1.avi	112.443.732	05/05/2005
20-2.avi	159.577.056	05/05/2005
20-3.avi	110.355.548	05/05/2005
20-4.avi	64.862.956	05/05/2005
20-Profundidad-Arbol.avi	33.391.012	05/05/2005
20-Profundidad-Vehiculo2.avi	21.011.068	05/05/2005
30-1.avi	90.816.112	05/05/2005
30-2.avi	167.929.804	05/05/2005
30-3.avi	75.154.712	05/05/2005
30-4.avi	28.021.392	05/05/2005
30-5.avi	33.689.320	05/05/2005
30-6.avi	314.251.924	05/05/2005
30-7.avi	116.470.964	05/05/2005
30-Profundidad-Vehiculo3.avi	31.302.828	05/05/2005
35-40-1.avi	33.689.320	05/05/2005
40-1.avi	39.953.880	05/05/2005
40-2.avi	142.871.572	05/05/2005
40-3.avi	29.214.640	05/05/2005
40-4.avi	55.615.280	05/05/2005
40-5.avi	41.296.296	05/05/2005
40-60-1.avi	99.914.628	05/05/2005
50-1.avi	94.246.692	05/05/2005
50-2.avi	52.482.988	05/05/2005
50-3.avi	68.144.388	05/05/2005
60-1.avi	88.429.616	05/05/2005

Tabla de videos capturados

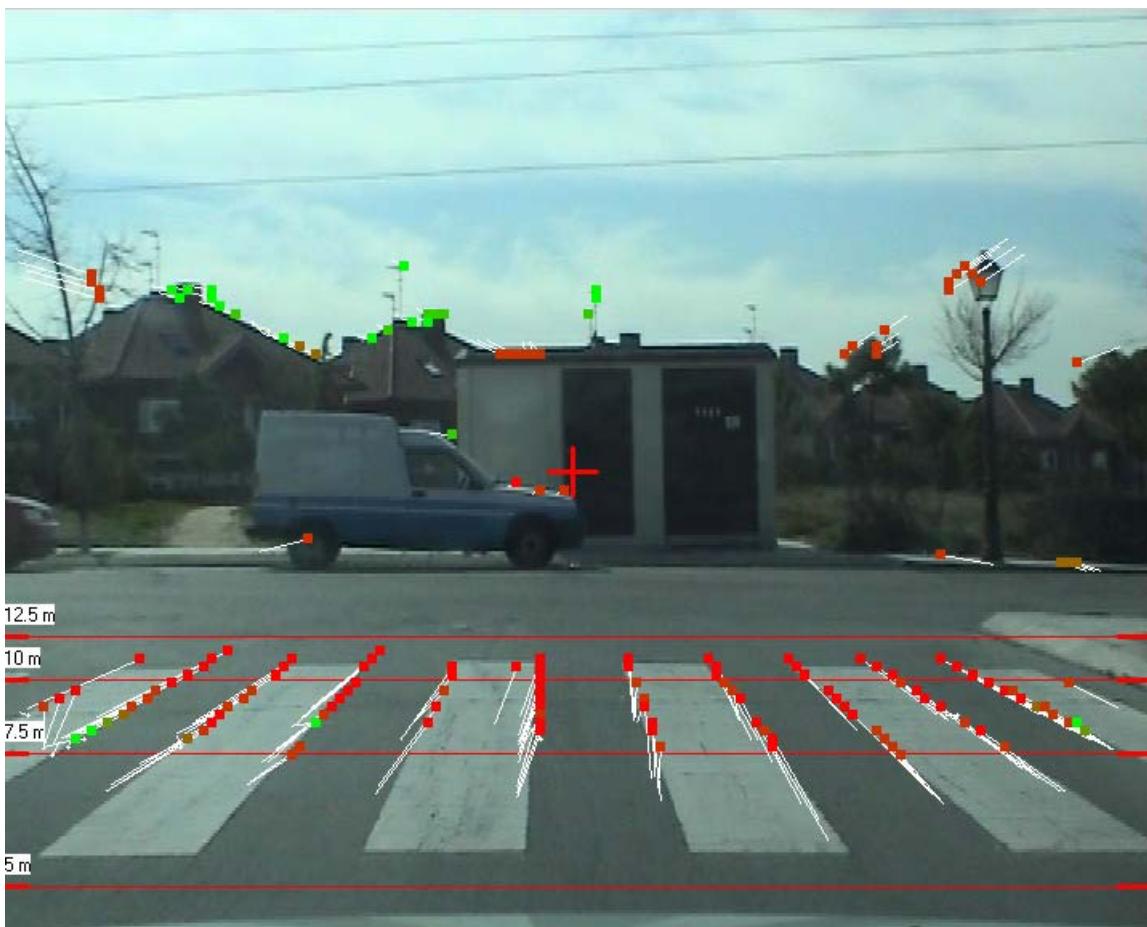
Para poder pasar las pruebas correctamente y capturar fácilmente datos se puede programar una consola que muestre entre otros: el número de puntos calculados, nombre de la imagen, restricciones de tiempos etc.

```
[> Analizando imagen:  
img\279.bmp] Puntos  
analisis: 504|TVuelta:  
9,3730001449585TReal:  
60,8001594543457 TTotal:  
69,9070129394531
```

Consola de Pruebas

Además existen funcionalidades que se desarrollaron en las pruebas, como teclas para pausar y continuar la simulación.

En las pruebas sobre el objetivo de la profundidad, al no ser fácilmente cuantificable el número de puntos de error en la ventana de profundidades del interfaz gráfico, se prefirió dibujar sobre la imagen como se muestra en la figura.



Mapa de profundidades sobre imagen para la realización de las pruebas

Estas modificaciones no aparecen en la aplicación final porque no entran dentro de los requerimientos, aunque los códigos desarrollados para tal efecto están incluidos en el soporte digital.

APENDICE III: Manual de Usuario

Este apéndice incluye un manual completo de usuario que describirá todos los pasos necesarios para instalar el sistema AMIVSI, además de explicar su funcionamiento e interpretación de los resultados presentados.

Manual de Instalación:

La instalación de la aplicación del sistema AMIVSI es muy sencilla:

El usuario debe copiar la carpeta *Sistema AMIVSI* del soporte digital del proyecto AMIVSI a un destino que seleccione el usuario en su disco duro.

Manual de Uso:

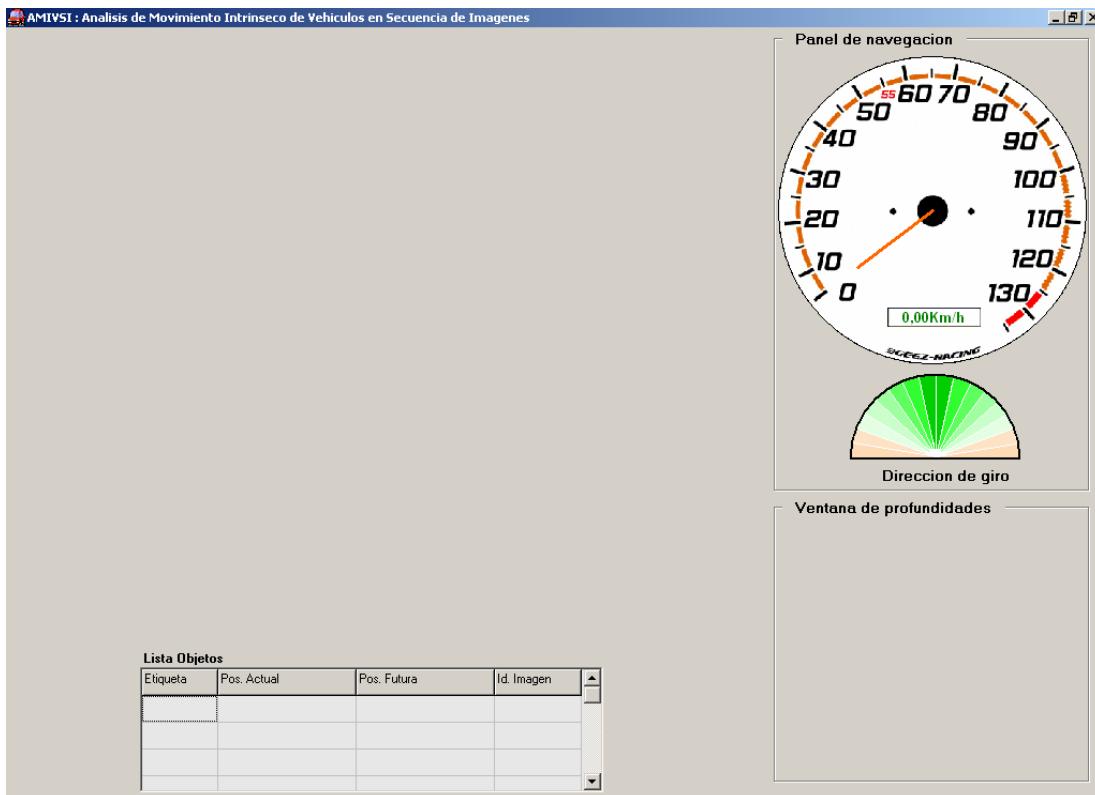
La utilización de la aplicación AMIVSI por tratarse de un sistema de análisis no presenta interacciones con el usuario, lo que simplifica este manual.

- Iniciar el análisis:

Para iniciar el análisis de la secuencia de imágenes el usuario tan sólo ha ejecutar el archivo *AMIVSI.exe* que se encuentra en la carpeta *Sistema AMIVSI*, que el usuario ha copiado previamente en su disco duro. (Véase Manual de Instalación).

Existen dos maneras de lanzar la ejecución, haciendo doble clic sobre el archivo, o seleccionándolo y posteriormente pulsar la tecla *Intro*.

Tras lanzar la ejecución se abrirá la ventana de la aplicación que inicialmente presentará este aspecto:



Interfaz inicial del sistema AMIVSI

Posteriormente se irán presentando los resultados del análisis y la imagen analizada en cada instante. En el siguiente punto se explica como serán presentados los resultados.

- Interpretación de los resultados del análisis:

El interfaz de la aplicación del sistema AMIVSI se puede dividir en cuatro grupos donde se representa toda la información del análisis realizado.

Ventana de video:

Es la sección más grande del interfaz, está alineada con la esquina superior izquierda de la ventana de la aplicación. Es donde se representa la imagen que ha sido analizada, en color para poder apreciar la escena con claridad. La ventana de video presenta una cruz de color rojo que indica el centro de la imagen. Es en esta ventana donde se podrá apreciar el resultado del seguimiento y etiquetado de objetos.

A continuación se muestra una captura de la ventana de video a lo largo del análisis.



Ventana de video

En esta imagen vemos el contenido de la escena así como los objetos previamente reconocidos y etiquetados.

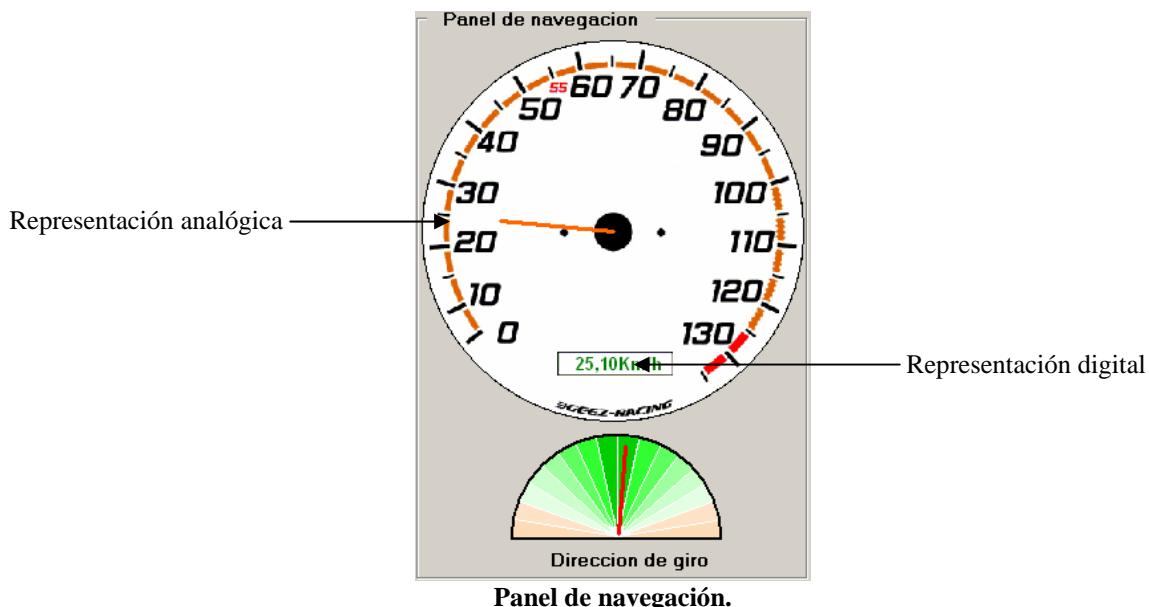
Panel de navegación:

Situado en la esquina superior derecha, proporciona información a cerca del movimiento que describe el vehículo en cada instante, consta de dos componentes, un velocímetro y un indicador de dirección de giro.

El velocímetro indica la velocidad estimada por el sistema para la imagen analizada y que aparece en la ventana de video. La velocidad estimada se presenta en formato digital y analógico dentro de una misma representación.

El indicador de dirección de giro representa la trayectoria que posee el vehículo en cada instante, si se está sucediendo algún giro a izquierda o derecha, o por lo contrario se desplaza en línea recta.

Veamos un ejemplo:



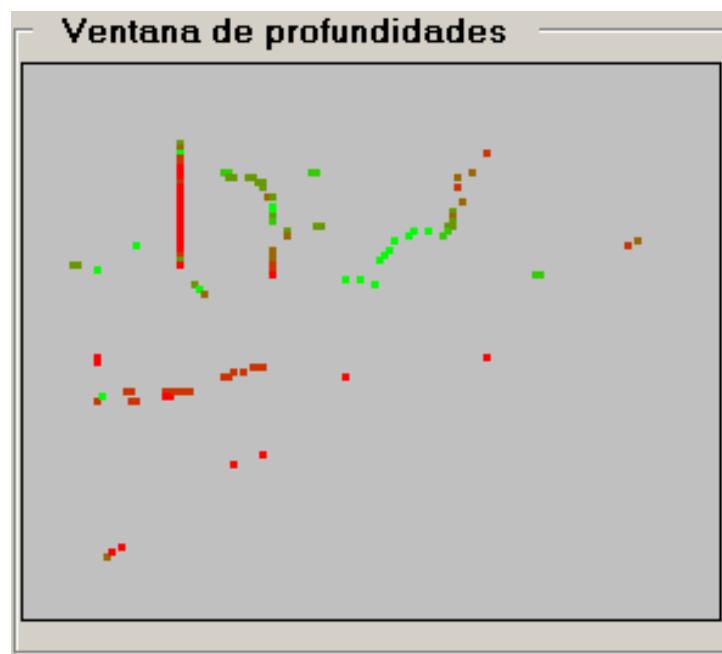
Desplazamiento en línea recta, con una ligera desviación a la derecha, a una velocidad de 25,1 Km./h.

Ventana de profundidades:

Se encuentra debajo del panel de navegación y representa la profundidad de la escena analizada, valiéndose de cuadrados de distintas tonalidades que expresan profundidades. La escala de tonalidades va desde el verde claro para expresar lejanía, pasando por tonos marrones que indican distancia media, hasta el rojo intenso que expresa cercanía. La siguiente imagen representa la escala de profundidades:



Veamos un ejemplo de un análisis en concreto:



Ventana de profundidades.

Esta es la ventana de profundidad generada para la siguiente imagen:



Imagen de profundidad analizada.

Se aprecia como las tonalidades en rojo pertenecen a la farola más próxima o a la porción de carretera inmediata. Los tonos marrones se corresponden con la segunda farola que se aprecia o con la copa de los árboles de la derecha, mientras que los tonos verdes intenso pertenecen a los árboles que se observan al final de la imagen.

Lista de objetos:

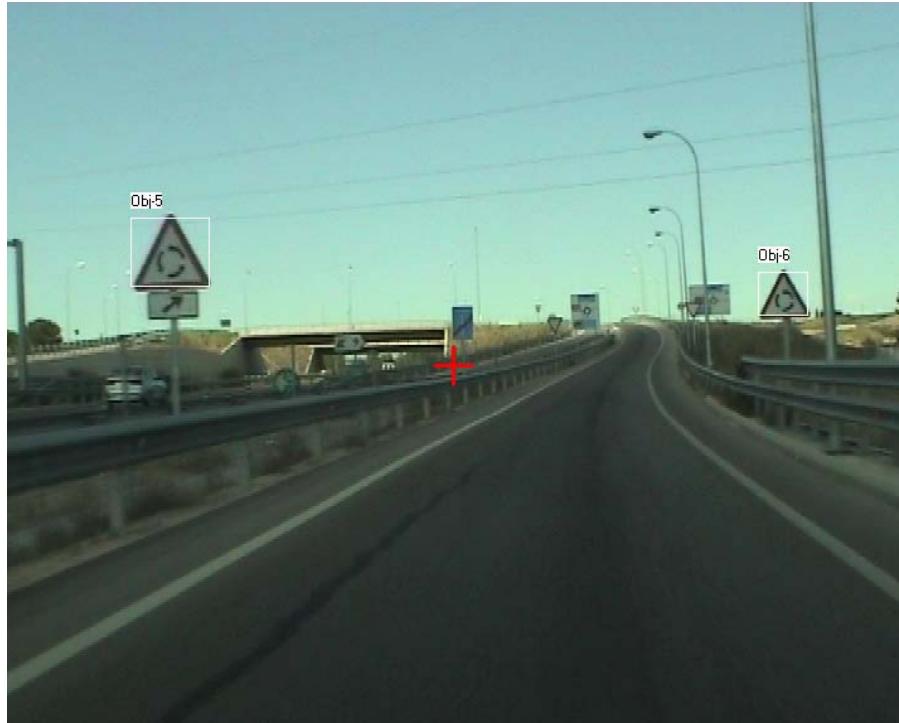
Situada en la parte inferior de la ventana proporciona una lista detallada de los diez últimos objetos que se han encontrado en la secuencia. Detallando de cada uno la imagen en la que se encontró, así como de las posiciones actuales y futuras de cada objeto y la etiqueta asignada. A continuación se presenta una captura de la simulación que recoge lo explicado.

Lista de objetos encontrados y etiquetados:

Lista Objetos				
Etiqueta	Pos. Actual	Pos. Futura	Id. Imagen	
Obj-5	(104,171) - (166,225)	(35,139) - (137,233)	47.bmp	
Obj-6	(602,214) - (641,248)	(585,188) - (664,262)	47.bmp	
Obj-5	(154,185) - (208,233)	(97,154) - (191,242)	46.bmp	
Obj-5	(194,199) - (240,240)	(145,170) - (231,251)	45.bmp	

Lista de objetos etiquetados.

Como se ve en la siguiente imagen aparecen los objetos 5 y 6 con su posición actual y futura, así como cada una de sus etiquetas:



Ventana de video con objetos etiquetados.

- Finalizar el análisis:

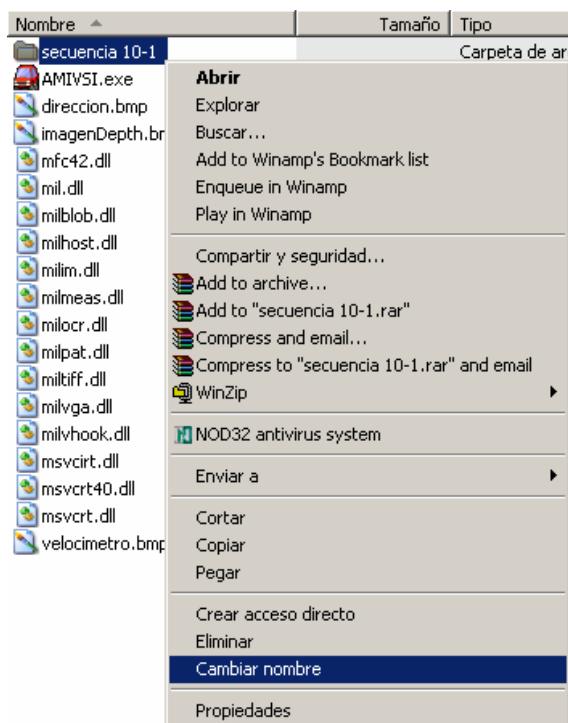
La aplicación finaliza cuando se han analizado la totalidad de las imágenes situadas en el directorio *img*. Es entonces cuando se puede cerrar la aplicación con seguridad, pulsando sobre la cruz superior derecha, o mediante las teclas *Alt+F4*.

- Cambiar secuencia de imágenes:

El usuario tiene la posibilidad de cambiar la secuencia de imágenes que se analizan. Para ello debe seguir los siguientes pasos:

1. Renombrar la carpeta *img* actual (última secuencia de análisis) por otro nombre cualquiera.
2. Se seleccionar una nueva carpeta de imágenes, del soporte digital del proyecto AMIVSI, y se copia en el directorio *Sistema AMIVSI*.
3. Renombrar la carpeta que se acaba de copiar con el nombre *img*.

Para cambiar el nombre de una carpeta debemos hacer clic con el botón derecho del ratón sobre la carpeta, seleccionar *Cambiar nombre* en el menú que se despliega, y escribir el nuevo nombre. Véase la siguiente imagen:



Cambiar nombre a un directorio.

- Añadir objetos previamente reconocidos:

El usuario tiene la posibilidad de crear archivos de lista de objetos previamente reconocidos, para comprobar como se realiza el seguimiento y etiquetado de los mismos por el sistema AMIVSI. El procedimiento a seguir consta de 5 pasos:

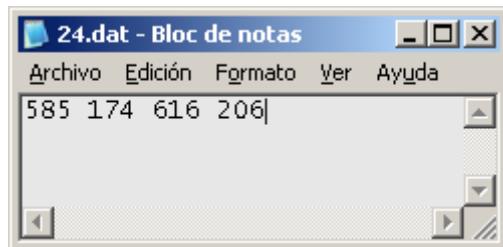
1. Identificar visualmente el objeto que se desea analizar dentro de una imagen: el usuario debe abrir la imagen deseada, y señalizar y anotar las coordenadas que delimitan el rectángulo que engloba al objeto.

2. Generar el archivo de lista de objetos: el usuario debe crear un archivo con extensión *.dat*, que tenga como nombre el mismo que la imagen de la que procede el objeto. Si queremos identificar un objeto en la imagen *24.bmp* el archivo generado tendrá por nombre *24.dat*.
3. Introducir mediante un editor de textos el archivo generado: el usuario una vez abierto el archivo, debe introducir las coordenadas del rectángulo que ha apuntado previamente, siguiendo el formato descrito en los interfaces del sistema, que encontrará en el capítulo IV, dentro de la especificación de requisitos.
4. En cada archivo se pueden especificar tantos objetos como se deseen. (El sistema contempla un máximo de 15 por imagen).
5. Si se desea realizar un seguimiento a un objeto se deben repetir los pasos del 1 al 4 en las sucesivas imágenes de la secuencia.

La imagen que se presenta a continuación ilustra el proceso de añadir objetos previamente reconocidos, a la imagen número 24 de una secuencia determinada:



Archivo generado tendrá el siguiente aspecto:



Archivo de lista de objetos.

- Contenidos del Soporte Digital (DVD):

Carpeta Código Fuente:

- Códigos desarrollados y librerías necesarias para compilar el sistema AMIVSI bajo Borland C++ Builder 6.

Carpeta Documentación:

- Cuaderno de Proyecto en el que se detalla día a día las tareas realizadas con su duración correspondiente.
- Plan de Proyecto en el que se describe la planificación, recursos empleados y presupuesto previstos.
- Memoria AMIVSI (este documento).
- Propuesta de Proyecto.

Carpeta Secuencias:

- Conjunto de directorios con diferentes secuencias de imágenes.

Carpeta Sistema AMIVSI:

- Archivos necesarios para la ejecución del sistema. Además contiene una secuencia de imágenes válida, en el directorio *img*.

APENDICE IV: Guía de Solucionado de Errores

Este apéndice contempla los posibles errores o anomalías de funcionamiento que se puedan producir en el sistema AMIVSI, proporcionando la información necesaria que describa el error y una guía para su correcta solución.

Los posibles errores que pueden suceder son dos:

- No aparece ninguna imagen en la ventana de video.
- El sistema no indica un objeto previamente reconocido e indicado mediante su fichero *.dat* correspondiente.

No aparece ninguna imagen en la ventana de video:

Sucede por dos motivos bien diferenciados y ambos tienen el mismo efecto: el sistema se inicia correctamente pero no aparece ninguna imagen en la ventana de video.

1. No existe ningún directorio en el interior de la carpeta *Sistema AMIVSI* con nombre *img*. Puede ser debido a una mala manipulación en el cambio de secuencia de análisis. Pasos para solucionar el problema:

- I. Localizar la carpeta que contiene la secuencia de imágenes que se desea analizar.
- II. Renombrar con *img*, la carpeta que contiene la secuencia de imágenes que se desea analizar.
- III. Si no se encuentra dentro de la carpeta que contiene los archivos del sistema AMIVSI, copiarla al directorio *Sistema AMIVSI*.

Nota: Para manipular las carpetas de imágenes se recomienda seguir el manual de usuario, apéndice III.

2. En el interior de la carpeta *img* no están nombrados correctamente los archivos de las imágenes. Los archivos deben ser nombrados siguiendo la numeración que comience en 1 y se incremente de unidad en unidad, si ceros por la izquierda, y con extensión *.bmp*. Si alguna de estas características no se cumplen el sistema no analizará ninguna imagen y dará por terminada la ejecución. Pasos para solucionar el problema:

- I. Localizar la primera imagen de la secuencia, y renombrarla a *1.bmp*.
- II. Para cada una de las imágenes siguientes, cambiarles el nombre si es necesario, siguiendo las normas de nombrado.

El sistema no indica un objeto previamente reconocido:

Este problema sucede cuando se ha especificado de manera errónea un objeto, dentro del archivo correspondiente de extensión *.dat*. Como resultado se obvia dicho objeto y se continúa procesando con los siguientes de la lista, en el caso de que existan. El sistema AMIVSI comprueba que se hayan introducido correctamente las coordenadas que delimitan al objeto, y en caso contrario se captura una excepción que garantiza el funcionamiento del análisis, descartando el objeto erróneo. Pasos para solucionar el problema:

- I. Localizar el archivo de lista de objetos *.dat* que se corresponda con la imagen donde se esperaba identificar el objeto.
- II. Abrir el archivo de lista de objetos
- III. Para cada línea verificar que existen 4 números enteros que ocupen 3 caracteres, y estén separados por un espacio. Importante verificar que si un entero se expresa con menos de 3 caracteres se sustituyan los restantes por espacios.
- IV. Al principio y al final de cada línea no debe existir ningún carácter o caracteres adicionales.

Nota: Para conocer en profundidad el formato de los archivos *.dat* de lista de objetos, se recomienda leer las interfaces del sistema del documento de especificación de requisitos, incluido en el capítulo IV de la memoria.